TÓPICOS EN
álgebra aplicada

CARLOS ARAUJO
MIGUEL CARO
ITALO DEJTER

Universidad del Atlántico
VIGILADA MINEDUCACIÓN

Sello Editorial
UNIVERSIDAD DEL ATLÁNTICO

TÓPICOS EN
álgebra aplicada

CARLOS ARAUJO
MIGUEL CARO
ITALO DEJTER

Universidad del Atlántico
VIGILADA MINEDUCACIÓN

Sello Editorial
UNIVERSIDAD DEL ATLÁNTICO

Araujo, Carlos -- Caro Candezano, Miguel Antonio -- Dejter, Italo

Tópicos en álgebra aplicada / Carlos Araujo, Miguel Antonio Caro Candezano, Italo Dejter. – 1 edición. – Puerto Colombia, Colombia: Sello Editorial Universidad del Atlántico, 2021.

158 páginas. 17x24 centímetros. Incluye bibliografía. Ilustraciones

ISBN: 978-958-5173-40-8 (Tapa dura)

1. Algebra. 2. Álgebras lineales. 2. Álgebra – Problemas, ejercicios. 3. Polinomios. I. Autor. II. Título.

CDD: 512 A663

Cómo citar este libro: Araujo, C., Caro, M. & Dejter, I. (2021). *Tópicos en álgebra aplicada.* Barranquilla, Colombia: Editorial Universidad del Atlántico.

Universidad del Atlántico
VIGILADA MINEDUCACIÓN

Sello Editorial
UNIVERSIDAD DEL ATLÁNTICO

www.uniatlantico.edu.co
Kilómetro 7, Antigua Vía a Puerto Colombia.
Barranquilla, Colombia.

ISBN 978-958-5173-41-5

Coordinación editorial
Javier Alfonso Ramírez Durán.

Asistencia editorial
Estefanía Calderón Potes.

Diseño y diagramación
Joaquín Camargo Valle.

Revisión y corrección
Estefanía Calderón Potes.

Impreso y hecho en Barranquilla, Colombia.
Ditar S.A. www.ditar.co
Kilómetro 7, Vía a Juan Mina.
Parque Industrial Clavería.

Printed and made in Barranquilla, Colombia.

La presente obra es posible gracias a las siguientes autoridades académicas de la Universidad del Atlántico:

José Rodolfo Henao Gil
Rector

Leonardo Niebles Núñez
Vicerrector de Investigaciones, Extensión y Proyección Social

Danilo Hernández Rodríguez
Vicerrector de Docencia

Mariluz Stevenson
Vicerrectora Financiera

Josefa Cassiani Pérez
Secretaria General

Miguel Caro Candezano
Jefe del Departamento de Investigaciones

Agradecimientos especiales

Facultad de Ciencias Básicas

Decano Alberto Moreno Rossi

2021

Esta obra ha sido publicada gracias a la iniciativa liderada por la Vicerrectoría de Investigaciones, Extensión y Proyección Social de la Universidad del Atlántico, en su esfuerzo por promover la divulgación de los avances en la investigación y generación de conocimiento en el Caribe colombiano.

CONTENIDO

PREFACIO

En los códigos que se usan para el envío de telecomunicaciones, se agregan dígitos redundantes a los mensajes que se pretende enviar con el fin de corregir los errores que se producen en su transmisión electrónica por causa del 'ruido' producido por fuentes variadas que incluyen la misma radiación cósmica.

Estas notas versan sobre conceptos elementales del álgebra aplicada, especialmente códigos correctores de errores, de tanto uso en telecomunicaciones hoy en día, y están acompañadas de numerosos ejemplos y ejercicios de aplicación. Cubrimos inclusive códigos lineales, códigos c íclicos, y códigos sobre cuerpos finitos y sobre álgebras booleanas, así como los esquemas de codificación y decodificación de los mismos, presentando y justificando las nociones básicas necesarias para la comprensión del material. De esta forma, creemos contribuir en forma no trivial con la presentación de conceptos educativos y técnicos de actualidad en castellano, manteniendo el rigor matemático adecuado, dentro de las posibilidades limitadas de la educación en nuestro medio.

Usamos letra itálica para indicar conceptos que se van definiendo en el texto, así como en los enunciados de los teoremas, proposiciones, lemas, corolarios y otras observaciones que se quieran resaltar para su uso en la exposición. En ese sentido, algunos hechos elementales de álgebra básica y lineal, incluso de polinomios sobre un cuerpo finito $\mathbb{F}(2^r)$, son citados cuando sean requeridos en el texto, siguiendo la filosofía industrial denominada 'justo a tiempo', que consiste en presentar los conceptos teóricos para el desarrollo de las aplicaciones cuando sea absolutamente necesario. El manejo práctico de tales conceptos es lo que nos importa en tal tipo de exposición, de ahí que ofrezcamos abundantes ejercicios para permitir que el lector tenga la oportunidad de entrenarse eficazmente en el uso de los conceptos y de adquirir una comprensión de sus importantes aplicaciones.

Este trabajo está dividido en capítulos indicados con número arábigos e, igualmente las secciones internas de los capítulos están numeradas con números arábigos y en forma lineal a lo largo de toda la presentación. Lo mismo acontece, por su parte, con el conjunto de los lemas, proposiciones, teoremas y corolarios del trabajo. Por otra parte, los ejercicios de cada capítulo se numeran linealmente (de nuevo empleando números arábigos) solo dentro del capítulo.

El apéndice contiene soluciones a algunos de los ejercicios del texto para beneficio del lector.

C. Araujo, M. Caro, I. Dejter.
Barranquilla - Colombia, Octubre 2020

CAPÍTULO 1
De cómo se codifican los mensajes electrónicos

La teoría de códigos es el estudio de métodos eficaces para la transferencia certera de la información de un local a otro. Esta teoría ha sido desarrollada en relación a diversas aplicaciones como son la minimización del ruido en grabaciones de discos compactos, la transmisión de información financiera a través de líneas telefónicas, la transferencia de datos de una computadora a otra o de la memoria de un procesador central y la transmisión desde una fuente distante como un satélite de comunicaciones o las naves espaciales Voyager que enviaron fotos desde Júpiter, Saturno, y desde distancias superiores, más allá de los límites de nuestro sistema solar.

El medio físico a través del cual la información es transmitida dícese ser un *canal* de información. Las líneas telefónicas y la atmósfera son ejemplos de canales de información. Perturbaciones indeseables recogidas en el concepto de *ruido* pueden causar que la información recibida difiera de la que fue transmitida. El ruido puede ser causado por manchas solares, rayos, dobleces en una cinta magnética, lluvias de meteoros, mensajes computacionales de la competencia, perturbaciones de radio aleatorias, mala escritura a maquinilla, lenguaje defectuoso o pobre, entre muchas otras causas.

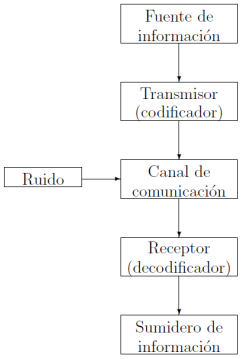

Figura 1.1: Idea de un sistema de transmisión de información

La teoría de códigos trabaja con el problema de detectar y corregir errores de transmisión causados por el ruido de un canal de información. El diagrama de la Figura 1.1 ofrece un esquema del sistema de transmisión de información. La parte más importante de este diagrama para nosotros es el ruido, pues sin él no habría necesidad de esta teoría.

En la práctica, el control que tenemos sobre este ruido es la posible elección de un buen canal para usar en la transmisión, y el uso de varios filtros de ruido para combatir ciertos tipos de interferencia que puedan ser encontrados. Estos son problemas de ingeniería. Una vez que hayamos establecido el mejor sistema mecánico para resolver estos problemas, podemos enfocar nuestra atención en la construcción del codificador y el decodificador. Nuestro deseo es construirlos de modo para alcanzar:

1. una codificación rápida de la información;
2. una fácil transmisión de los mensajes codificados;
3. una decodificación rápida de los mensajes recibidos;
4. la corrección de errores introducidos en el canal;
5. la transferencia máxima de la información por unidad de tiempo.

Nuestra meta primaria es el cuarto de estos ítems. El problema es que este no es generalmente compatible con el quinto y puede no ser tampoco compatible con los otros tres. Por lo tanto, cualquier solución es necesariamente un intercambio de los cinco objetivos.

En nuestras comunicaciones diarias, usamos palabras habladas o escritas, construidas a partir de un alfabeto limitado. Tenemos información para comunicar, y la codificamos en cadenas de palabras que luego decimos o escribimos; luego estas, a su vez, son enviadas por un canal que, generalmente, es el espacio que va de la boca de uno al oído de otro, o del lápiz al papel, y del papel al ojo. El ruido puede ser causado por un habla confusa, mala audición, un uso gramatical incorrecto, un aparato de música estereofónica, conversaciones superpuestas, faltas de ortografía o una maquinilla defectuosa. El decodificador es nuestra audición (o lectura) y la capacidad de comprender los mensajes recibidos.

Existen dispositivos correctores de errores en los que ni siquiera pensamos. Si en un mensaje corto hay letras cambiadas que hacen que el mensaje sea incomprensible, -por ejemplo, por el cambio de dos o tres letras-, podemos modificar esas letras y recobrar el mensaje original, (Ej:"Aptue natural. Tengo un arba"), que en realidad sería "Actúe naturalmente. Tengo un arma", lo cual ocurre en una película de 1969 dirigida por un actor y clarinetista norteamericano.

Para estos tipos de errores podemos probablemente operar con lo siguiente: elegir la palabra más factible o susceptible de haber sido transmitida. El método patrón para combatir errores es a través de la redundancia. Es así como muchos negocios hoy en día agregan de forma rutinaria controles de dígitos para identificar números; estos son conocidos como extra-dígitos y son usados para chequear que los datos son correctos en los números de cuentas. Este es, probablemente, el método más comúnmente reconocido para codificar en la vida real. Trabajaremos, pues, con ideas similares, pero más sofisticadas.

1.1. Suposiciones básicas

Enunciemos algunas definiciones básicas y suposiciones que serán aplicadas en la teoría de códigos.

En muchos casos, la información a ser enviada es transmitida por una sucesión de ceros y unos. Decimos que 0 y 1 son *dígitos*. Una *palabra* es una sucesión de dígitos. La *longitud* o *largo* es el número de dígitos en la palabra. Luego, 0110101 es una palabra de largo 7. Una palabra es transmitida al enviar sus dígitos, uno detrás del otro, a través de un *canal binario*. La palabra "binario" se refiere al hecho de que solo los dígitos 0 y 1 son usados. Cada dígito es transmitido mecánicamente, eléctricamente, magnéticamente o de otra forma que diferencie dos tipos fácilmente distinguibles de pulsos.

Un *código binario* es un conjunto C de palabras. El código que consiste de todas las palabras de largo 2 es

$$C = \{00, 10, 01, 11\}.$$

Un *código-bloque* es un código que tiene todas las palabras del mismo largo; este número es el *largo* del código-bloque. Así, el término *código* querrá decir para nosotros un código-bloque binario. Las palabras que pertenecen a un código C_0 serán denominadas *palabras-código*. Denotamos el número de palabras de un código C por medio de $|C|$.

Ejercicios.

1. Liste todas las palabras de largo 3; de largo 4; de largo 5.
2. Halle una fórmula para el número total de palabras de largo n.
3. Sea C el código consistente de todas las palabras de largo 6 que tienen un número par de unos. Liste las palabras de C.

También precisamos hacer ciertas suposiciones básicas en relación al canal de comunicación. Estas suposiciones darán forma a la teoría que formulamos.

La primera suposición es que una palabra-código de largo n consistente de varias entradas de valor 0 y 1 sea recibida como una palabra de largo n, aunque no sea necesariamente la misma palabra que fue transmitida.

La segunda suposición es que no hay ninguna dificultad en identificar el comienzo de la primera palabra transmitida. Luego, si usamos palabras de largo 3 y recibimos 011011001,

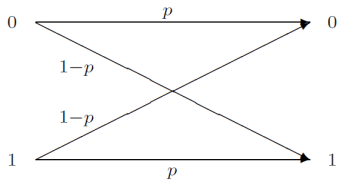

Figura 1.2: Diagrama que clarifica cómo opera un CSB

sabemos que las palabras recibidas son 011, 011 y 001. Esta suposición quiere decir, de nuevo usando largo 3, que el canal no puede enviar 01101 al destinatario, pues un dígito fue perdido aquí.

La suposición final es que el ruido es desparramado aleatoriamente, no en racimos o montones, llamados *ráfagas*. O sea, la probabilidad de que un dígito sea afectado en la transmisión es la misma que la de que cualquier otro dígito vecino lo sea. Esta no es una suposición muy realista para muchos tipos de ruidos tales como rayos o rayaduras de discos compactos. Este es un capítulo más avanzado en la teoría de códigos, ("Códigos correctores de ráfagas de errores").

En un canal *perfecto*, o sin ruido, el dígito enviado, 0 ó 1, es siempre recibido. Si todos los canales son perfectos, no habría necesidad de la teoría de códigos. Pero, afortunadamente (o desafortunadamente, tal vez) no todos los canales son perfectos; todo canal es ruidoso. Algunos canales son menos ruidosos o más confiables que otros.

Un canal binario es *simétrico* si 0 y 1 son transmitidos con igual puntualidad, esto es si la probabilidad de recibir el dígito correcto es independiente de qué dígito, 0 ó 1, está siendo transmitido. La *confiabilidad* de un canal simétrico binario (CSB) es un número real p, $(0 \leq p \leq 1)$, donde p es la probabilidad de que el dígito enviado sea el dígito recibido.

Si p es la probabilidad de que el dígito recibido es el mismo que el dígito enviado, entonces $1 - p$ es la probabilidad de que el dígito recibido no sea el dígito enviado. El diagrama de la Figura 1.2 clarifica cómo opera un CSB.

En la mayoría de los casos puede ser difícil estimar el valor real de p para un canal dado. Sin embargo, el valor real de p no tiene influencia significativa en la forma de la teoría.

Diremos que un canal es más confiable que otro si su confiabilidad es mayor. Nótese que si $p = 1$, entonces no hay chance de que un dígito sea alterado en la transmisión. Así pues, el canal es perfecto y no tiene interés para nosotros. Tampoco ofrece interés un canal con $p = 0$. Cualquier canal con $0 < p \leq 1/2$ puede ser convertido en otro con $1/2 \leq p < 1$. De aquí, supondremos que estamos usando un CSB con probabilidad p satisfaciendo $1/2 < p < 1$. (El caso $p = 1/2$ está citado en un ejercicio más abajo).

Ejercicios.

4. Explique por qué un canal con $p = 0$ no es interesante.

5. Explique cómo convertir un canal con $0 < p \leq 1/2$ en otro canal con $1/2 \leq p < 1$.

6. ¿Qué puede decirse al respecto de un canal con $p = 1/2$?

1.2. Patrones correctores y detectores de errores

Consideremos ahora las posibilidades de corregir y detectar errores. En esta sección intentamos desarrollar un entendimiento intuitivo de los conceptos asociados a la corrección y detección de errores; además, un enfoque formal es adoptado más adelante.

Supongamos que una palabra sea recibida y que no sea una palabra-código. Claramente algún error ha ocurrido durante el proceso de transmisión, de modo que hemos *detectado* que un error (tal vez varios errores) han acontecido. Sin embargo, si una palabra-código es recibida, entonces tal vez no hubo errores durante la transmisión, de modo que no podemos detectar un error.

El concepto de corrección de un error es más complejo. Cuando en la primera sección pensamos en corregir "arba" por "arma", (y no por ejemplo "arpa"), recurrimos a la intuición para sugerir que cualquier palabra recibida debería ser *corregida* a una palabra-código que requiera tan pocos cambios como sea factible. (Habrá que probar que la probabilidad de que tal palabra-código haya sido enviada es al menos tan grande como la probabilidad de que cualquier otra palabra fuese enviada). Para consolidar estas ideas, discutiremos algunos códigos en particular. Tener presente, pues, nuestra suposición de que ningún dígito es perdido o creado en la transmisión.

Ejemplo. Sea $C_1 = \{00, 01, 10, 11\}$. Cada palabra recibida es una palabra-código y así C_1 no puede detectar ningún error. Además C_1 no corrige errores, pues ninguna palabra recibida requiere cambios para tornarse en palabra-código.

Ejemplo. Modifiquemos C_1 para repetir cada palabra-código tres veces. El nuevo código es

$$C_2 = \{000000, 010101, 101010, 111111\}.$$

Este es un ejemplo de un *código de repetición*. Supongamos que la palabra 110101 sea recibida. Como esta no es una palabra-código, podemos detectar que al menos un error ha ocurrido. La palabra 010101 puede ser formada cambiando un dígito, pero todas las otras palabras son formadas al cambiar más de un dígito. Luego, esperamos que 010101 fuese la palabra-código más factible transmitida, de modo que corregimos 110101 a 010101. (Una palabra que puede ser formada de una palabra w con el menor número de dígitos siendo cambiados es denominada la palabra-código más próxima, idea que se formalizará más adelante). De hecho, si cualquiera de las palabras $c \in C_2$ es transmitida y un error ocurre durante la transmisión, entonces la única palabra-código más próxima a la palabra recibida es c; así, cualquier error individual resulta en una palabra que corregimos a la palabra que fue transmitida.

Ejemplo. Modifiquemos C_1 agregando un tercer dígito a cada palabra-código de modo que

el número de 1 (unos) en cada palabra-código sea par. El código resultante es

$$C_3 = \{000, 011, 101, 110\}.$$

El dígito agregado es denominado dígito de *control* de *paridad*. Supongamos que la palabra 010 sea recibida; entonces, como 010 no es una palabra-código, podemos detectar que un error ha ocurrido. Cada una de las palabras 110, 000 y 011 puede ser formada al cambiar un dígito en la palabra recibida. Más adelante distinguiremos cómo tratar palabras recibidas más próximas de una sola palabra-código o igualmente próximas de varias. Basta ahora observar que parece más adecuado corregir 010 como alguna palabra 110, 000 ó 011, más bien que a 101.

Ejercicios.

7. Sea C un código cuyas palabras sean todas las de largo 3. Determine cuál palabra fue la más factible enviada si 001 es recibida.

8. Agregar un control de paridad a las palabras en el código anterior y usar el código resultante C para responder las siguientes preguntas:

   a) Si la palabra 1101 es recibida, ¿podemos detectar un error?

   b) Supóngase que recibimos la palabra 1101. ¿Cuáles son las más factibles palabras enviadas?

   c) ¿Es alguna palabra de largo 4 que no esté en el código más próxima a una palabra única?

9. Repetir cada palabra del código C del ejercicio 1 tres veces para formar un código de repetición de largo 9. Hallar las palabras-código más próximas a las siguientes palabras recibidas

   a) 001000001

   b) 011001011

   c) 101000101

   d) 100000010

10. Hallar el máximo número de palabras de largo $n = 4$ en un código en que cada error individual puede ser detectado.

11. Repetir el ejercicio 10 para $n = 5$, $n = 6$, y para n general.

1.3. Razón de información

Es claro ahora que el agregado de dígitos a palabras-código puede mejorar las capacidades de corrección y detección del código. Sin embargo, cuanto más largas las palabras, más demora en ser enviado el mensaje. La *razón de información* (o solo razón) de un código es un número que se designa para medir la proporción de cada palabra-código que está cargada en el mensaje. La razón de información de un código C de largo n se define para códigos binarios como

$$\frac{1}{n} \log_2 |C|.$$

Puesto que podemos suponer que $1 \leq |C| \leq 2^n$, es claro que la razón de información corre entre 0 y 1; es 1 si cada palabra es palabra-código, y 0 si $|C| = 1$.

Por ejemplo, la razón de información de los códigos C_1, C_2 y C_3 tratados arriba es respectivamente 1, 1/3 y 2/3. Cada una de estas razones de información parece sensiblemente relacionada a los respectivos códigos, puesto que los primeros dos dígitos de los 6 en cada palabra de C_2 pueden ser vistos como que llevan el mensaje, lo que pasa también con los dos primeros dígitos de C_3.

1.4. Efectos de la corrección y detección de errores

Para ejemplificar los efectos dramáticos que el agregado de un control de paridad a un código puedan tener en reconocer cuando ocurren errores, consideremos los siguientes códigos.

Supongamos que todas las 2^{11} palabras de largo 11 son palabras-código; luego, no se detecta ningún error. Sea la confiabilidad del canal $p = 1 - 10^{-8}$ y supongamos que los dígitos son transmitidos a una razón de 10^7 dígitos por segundo. Luego, la probabilidad de que una palabra sea transmitida incorrectamente es aproximadamente $11p^{10}(1-p)$, que está cerca de $11/10^8$. Así:

$$\frac{11}{10^8} \cdot \frac{10^7}{11} = ,1 \text{ palabras por segundo}$$

son transmitidas incorrectamente sin ser detectadas. O sea, una palabra errada cada 10 segundos, 6 por minuto, 360 por hora, ¡8640 por día!: no demasiado bueno...

Ahora supongamos que un dígito de control de paridad es agregado a cada palabra, de modo que el número de 1 (unos) en cada una de las 2048 palabras-código sea par. Luego cualquier error individual es siempre detectado, de modo que al menos 2 errores deben ocurrir si la palabra ha de ser transmitida incorrectamente sin nuestro conocimiento. La probabilidad de que al menos 2 errores ocurran es $1-p^{12}-12p^{11}(1-p)$, que puede aproximarse a $\binom{12}{2}p^{10}(1-p)^2$, lo cual para $p = 1 - 10^{-8}$ está cerca de $\frac{66}{10^{16}}$. Ahora bien, aproximadamente $\frac{66}{10^{16}} \cdot \frac{10^7}{12} = 5{,}5 \times 10^{-9}$ palabras por segundo son transmitidas incorrectamente sin nosotros haberlo detectado. O sea ¡un error cada 2000 días!

Así, si estamos dispuestos a reducir la razón de información alargando el código de 11 a 12 dígitos, muy probablemente conoceremos cuándo ocurren errores. Para decidir dónde esos errores realmente ocurren, podemos precisar requerir la retransmisión del mensaje. Físicamente, esto quiere decir que o bien la retransmisión debe hacerse hasta que se confirme la recepción, o bien el mensaje se debe guardar temporalmente hasta que la retransmisión se requiera. Ambas alternativas deben ser costosas en tiempo o en memoria. Puede ser también que la retransmisión sea impráctica, como en el caso del Voyager y con discos compactos. Por lo tanto, sería mejor aumentar las capacidades de corrección de errores. Introducir esas capacidades hace que la codificación y la decodificación sean más costosas, pero evitará los costos escondidos en tiempo y en espacio mencionados arriba.

Un esquema simple para introducir la corrección de errores es formar un código de repetición donde cada palabra sea transmitida tres o más veces en sucesión. Luego, si al menos un error se produce para palabras-código de largo 33, al menos dos de las tres transmisiones estarán correctas. Como la comparación de las tres palabras de largo 11 es relativamente simple, el único cambio para ser capaces de corregir un error es una razón de información de 1/3, en lugar de 1.

1.5. Hallando la más factible palabra-código transmitida

Supongamos que tenemos una visión del proceso de transmisión, conociendo la palabra-código v transmitida y la palabra w recibida. Para cualquier v y w dadas, sea $\phi_p(v,\ w)$ la probabilidad de que si v es enviada por un CSB con confiabilidad p, entonces w es recibida. Como suponemos que el ruido está distribuido aleatoriamente, podemos tratar la transmisión de cada dígito como un evento independiente. Así, si v y w difieren en d posiciones, entonces, tenemos $n-d$ dígitos correctamente transmitidos y d incorrectamente transmitidos. Luego,

$$\phi_p(v, w) = p^{n-d}(1-p)^d.$$

Ejemplo. Sea C un código de largo 5. Luego, para todo $v \in C$, la probabilidad de que v sea recibida correctamente es

$$\phi_p(v, v) = p^5.$$

Sea 10101 un elemento de C. Entonces,

$$\phi_p(10101, 01101) = p^3(1-p)^2$$

y si $p = 0{,}9$ entonces

$$\phi_p(10101, 01101) = (0{,}9)^3(0{,}1)^2 = 0{,}00729$$

Ejercicios.

12. Calcular $\phi_{0{,}97}(v, w)$ para cada uno de los pares de v y w:

    a) $v = 01101101$, $w = 10001110$

    b) $v = 1110101$, $w = 1110101$

    c) $v = 00101$, $w = 11010$

    d) $v = 00000$, $w = 00000$

    e) $v = 1011010$, $w = 0000010$

    f) $v = 10110$, $w = 01001$

    g) $v = 111101$, $w = 000010$.

En la práctica, conocemos w, la palabra recibida, pero no la palabra-código v que fue enviada. Sin embargo, cada palabra v determina una asignación de probabilidades $\phi_p(v, w)$ a las palabras w. Cada tal asignación es un modelo matemático y elegimos el modelo (es decir, la palabra v) que concuerda más con la observación –en este caso, que hace la palabra recibida más factible. O sea, supongamos que v haya sido enviada cuando w es recibida si

$$\phi_p(v, w) = \text{máx}\{\phi_p(u, w) : u \in C\}.$$

El siguiente teorema provee un criterio para hallar tal palabra v.

Teorema 1. *Supongamos que tenemos un CSB con* $1/2 < p < 1$. *Sean* v_1 *y* v_2 *palabras-código y* w *una palabra, todas de largo* n. *Supongamos además que* v_1 *y* w *difieren en* d_1 *posiciones y que* v_2 *y* w *difieren en* d_2 *posiciones. Luego,*

$$\phi_p(v_1, w) \leq \phi_p(v_2, w) \text{ si y solo si } d_1 \geq d_2.$$

Demostración. Tenemos ya establecido que $\phi_p(v_1, w) \leq \phi_p(v_2, w)$ si y solo si $p^{n-d_1}(1-p)^{d_1} \leq p^{n-d_2}(1-p)^{d_2}$ si y solo si $\left(\frac{p}{1-p}\right)^{d_2-d_1} \leq 1$ si y solo si $d_2 \leq d_1$, (puesto que $\frac{p}{1-p} > 1$). □

Esto establece formalmente el procedimiento para corregir palabras que hasta ahora adoptamos como un procedimiento intuitivamente sensible: corregir w a una palabra-código que discuerda con w en tan pocas posiciones como sea factible, ya que tal palabra-código es la más factible palabra enviada, dado que w fue recibida.

Ejemplo. Si $w = 00110$ es recibida en un CSB con $p = ,98$, ¿cuál de las palabras-código 01101, 01001, 10100, 10101 fue la más factible palabra enviada?

v	d (número de desacuerdos con w)
01001	3
01001	4
10100	$2 \leftarrow d$ (el más pequeño)
10101	3

Usando la tabla de arriba, el Teorema 1 dice que 10100 fue la más factible palabra-código transmitida. Notar que no precisamos saber el valor exacto de p para poder aplicar el teorema; solo precisamos saber que $p > 1/2$.

Ejercicios.

13. Supongamos que $w = 0010110$ sea recibida en un CSB con confiabilidad $p = ,90$. ¿Cuáles de las siguientes palabras-código fueron las más factibles palabras enviadas?

$$1001011, 1111100, 0001110, 0011001, 1101001.$$

14. ¿Cuál de las 8 palabras-código en el código del Ejercicio 9 es la más factible de haber sido enviada si $w = 101000101$ es recibida?

15. Si $C = \{01000, 01001, 00011, 11001\}$ y la palabra $w = 10101$ es recibida, ¿cuál palabra-código es la más factible de haber sido enviada?

16. Repita el Ejercicio 15 luego de reemplazar C por
$\{010101, 110110, 101101, 100110, 011001\}$ y w por 101010.

17. ¿Cuál de las palabras-código 110110, 110101, 000111,
100111, 101000 fue la más factible palabra enviada si $w = 011001$ es recibida?

18. En el Teorema 1 supusimos que $1/2 < p < 1$. ¿Qué cambiaría en el enunciado del Teorema 1 si se reemplaza esa suposición?

    a) $0 < p < 1/2$

    b) $p = 1/2$.

1.6. Álgebra básica para códigos

Un problema que necesitamos tratar es el de hallar una manera eficiente de encontrar la palabra-código más próxima de la palabra recibida. Si el código tiene muchas palabras, entonces es impráctico comparar cada palabra w recibida a cada palabra-código para hallar que palabra-código está en desacuerdo en el menor número de posiciones posibles. Por ejemplo, si el código contiene 2^{12} palabras (como en el caso de la misión Voyager), entonces tal procedimiento de decodificación no podría jamás mantenerse con la transmisión entrante. Para vencer este problema, precisamos introducir alguna estructura en nuestros códigos.

Sea $\mathbb{K} = \{0, 1\}$ y sea $\mathbb{K}^n$ el conjunto de todas las palabras binarias de largo n. Definamos suma y multiplicación de los elementos de $\mathbb{K}$ como sigue:

$$0 + 0 = 0,\ \ 0 + 1 = 1,\ \ 1 + 0 = 1,\ \ 1 + 1 = 0;$$

$$0 \cdot 0 = 0,\ \ 0 \cdot 1 = 0,\ \ 1 \cdot 0 = 0,\ \ 1 \cdot 1 = 1.$$

Definamos la suma de los elementos de $\mathbb{K}^n$ componente por componente usando la suma definida en $\mathbb{K}$ para sumar cada componente. Por ejemplo, sea

$$v = 01101 \text{ y } w = 11001; \text{ luego } v + w = 10100.$$

Claramente, la suma de dos palabras binarias de largo n resulta en una palabra binaria de largo n, de modo que $\mathbb{K}^n$ es cerrado bajo suma.

Usando la terminología del álgebra lineal, nos referimos a un elemento de $\mathbb{K}$ como *escalar*. Entonces la multiplicación escalar de $\mathbb{K}^n$ se define componente por componente. Como los únicos escalares son 0 y 1, los únicos múltiplos escalares de una palabra w son $0 \cdot w$, que es el elemento de $\mathbb{K}^n$ con cero en cada componente, y $1 \cdot w$, que es w. Nos referimos al elemento de $\mathbb{K}^n$ con cero en cada componente como la *palabra nula*. Claramente, $\mathbb{K}^n$ es cerrada bajo la multiplicación escalar.

Con estas definiciones de suma y multiplicación escalar puede ser mostrado que $\mathbb{K}^n$ es un espacio vectorial. O sea, para cualquier par de palabras u, v de largo n y para escalares a, b, vale que

1. $v + w \in \mathbb{K}^n$
2. $(u + v) + w = u + (v + w)$
3. $v + 0 = 0 + v = v$, donde 0 es la palabra nula
4. para algún $v' \in \mathbb{K}^n$, $v + v' = v' + v = 0$
5. $v + w = w + v$
6. $av \in \mathbb{K}^n$
7. $a(v + w) = av + aw$
8. $(a + b)v = av + aw$
9. $(ab)v = a(bv)$
10. $1v = v$.

Ejercicios.

19. Mostrar que si v es una palabra en $\mathbb{K}^n$, entonces $v + v = 0$.

20. Mostrar que si v y w son palabras en $\mathbb{K}^n$ y $v + w = 0$, entonces $v = w$.

21. Mostrar que si u, v y w son palabras en $\mathbb{K}^n$ y $u + v = w$, entonces $u + w = v$.

Nótese que si v fue enviada en un CSB y w es recibida, entonces 0 ocurre en una componente de $v + w$ si la componente correspondiente de v fue correctamente transmitida, y 1 ocurre si la componente fue incorrectamente transmitida. Entonces, $v + w$ es denominado el *patrón de error*, o el *error*. Por ejemplo, si la palabra $v = 10101$ fue transmitida y la palabra $w = 01100$ es recibida, entonces errores ocurren en las primera, segunda y quinta componentes. El patrón de error es $v + w = 11001$.

1.7. Peso y distancia de Hamming

Introducimos dos términos importantes. Sea v una palabra de largo n. El *peso de Hamming*, o simplemente *peso*, de v es el número de veces que el dígito 1 ocurre en v. Denotamos el peso de Hamming de v por medio de $\mathrm{w}(v)$. Por ejemplo, $\mathrm{w}(10101) = 3$ y $\mathrm{w}(00000) = 0$.

Sean v y w dos palabras de largo n. La *distancia de Hamming*, o simplemente *distancia*, de v es el número de posiciones en las que v y w discuerdan. Denotamos la distancia entre v y w por medio de $d(v, w)$. Por ejemplo, $d(01011, 00111) = 2$ y $d(10110, 10110) = 0$.

Notar que la distancia entre v y w es igual al peso del patrón de error $u = v + w$:

$$d(v, w) = \mathrm{w}(v + w).$$

Por ejemplo, si $v = 11010$ y $w = 01101$, tenemos que $d(v, w) = d(11010, 01101) = 4$ y $\mathrm{w}(v+w) = \mathrm{w}(11010+01101)\mathrm{w}(10111) = 4$. Luego, la fórmula probabilística en la sección 6 puede expresarse como

$$\phi_p(v, w) = p^{n-\mathrm{W}(u)}(1 - p)^{\mathrm{W}(u)},$$

donde u es el patrón de error $u = v + w$. Nos referimos a $\phi_p(v, w)$ como *probabilidad del patrón de error* $u = v + w$.

Ejercicios.

22. Comparar los pesos de cada una de las siguientes palabras, y la distancia entre cada par de ellas: $v_1 = 1001010$, $v_2 = 0110101$, $v_3 = 0011110$ y $v_4 = v_2 + v_3$.

23. Sean $u = 01011$, $v11010$ y $w = 01100$. Comparar cada uno de los siguientes pares de cantidades:

    a) $\mathrm{w}(v + w)$ y $\mathrm{w}(v) + \mathrm{w}(w)$,

    b) $d(v, w)$ y $d(v, u)$ y $d(u, w)$.

Listamos ahora un número de hechos que se refieren a peso y distancia. Aquí, u, v y w son palabras de largo n y a es un dígito.

1. $0 \leq \mathrm{w}(v) \leq n$.

2. $\mathrm{w}(v) = 0$ si y solo si $v = 0$.
3. $0 \leq d(v, w) \leq n$.
4. $d(v, w) = 0$ si y solo si $v = w$.
5. $d(v, w) = d(w, v)$.
6. $\mathrm{w}(v + w) \leq \mathrm{w}(v) + \mathrm{w}(w)$.
7. $d(v, w) \leq d(v, u) + d(u, w)$.
8. $\mathrm{w}(av) = a \cdot \mathrm{w}(v)$.
9. $d(av, aw) = a \cdot d(v, w)$.

La mayoría de estos hechos son inmediatamente claros de las definiciones de peso y distancia. En el último ejercicio el lector construyó ejemplos de los hechos (6) y (7). Para elaborar pruebas, trate de usar la relación básica $d(v, w) = \mathrm{w}(v+w)$ y los tres últimos ejercicios de la sección 6.

Ejercicios.

24. Construir un ejemplo en $\mathbb{K}^5$ de cada una de las nueve leyes arriba enunciadas.
25. Probar cada una de esas nueve leyes.

Estos hechos serán usados cuando sea preciso sin mayores comentarios ni referencias, dada su ubicuidad.

1.8. Decodificación de máxima verosimilitud

Estamos ya preparados para dar una formulación más precisa de dos problemas básicos de la teoría de códigos. Supongamos que estamos en el receptor (destinatario) de un CSB y que queremos recibir un mensaje del transmisor. Por supuesto, el transmisor es uno que habíamos diseñado previamente. De hecho, el diseño del transmisor es uno de los problemas básicos.

Hay dos cantidades sobre las que no tenemos ningún control. Una de ellas es la probabilidad p de que nuestro CSB transmita un dígito correctamente. La segunda cantidad es la cardinalidad $|M|$ del conjunto de mensajes posibles que pueden ser transmitidos. Los mensajes en cuestión no son tan importantes como lo es el número $|M|$ de mensajes posibles.

Recordemos que para cualquier conjunto S, denotamos por S el número de elementos de S. Luego, $|\mathbb{K}^n| = 2^n$.

Los dos problemas básicos de códigos son los siguientes:

1.8.1. Codificación

Debemos determinar un código para emplear en el envío de nuestros mensajes, y algunas selecciones deben ser hechas para ello. Primero, seleccionamos un entero positivo k, que es el largo de cada palabra binaria correspondiente a un mensaje. Como a cada mensaje se le debe asignar una palabra binaria diferente de largo k, k debe ser elegido de modo que $|M| \leq |\mathbb{K}^n| = 2^k$. Luego decidimos cuántos dígitos precisamos agregar a cada palabra de largo k para asegurar

que tantos errores puedan ser corregidos o detectados como precisemos; esto es la selección de las palabras-código y el largo del código, n. Para transmitir un mensaje particular, el transmisor halla la palabra de largo k asignada a cada mensaje; luego, transmite la palabra-código de largo k correspondiente a cada palabra de largo k.

1.8.2. Decodificación

Digamos que la palabra w en K^n es recibida. Describiremos un procedimiento llamado *decodificación de máxima verosimilitud*, o DMV, para decidir qué palabra $v \in C$ fue enviada. Hay, de hecho, dos tipos de DMV:

1. *DMV completa*, o *DMVC*. Si hay una y solo una palabra $v \in C$ más próxima a w que cualquier otra palabra en C, decodificamos w como v. O sea, si $d(v, w) < d(v_1, w)$ para toda v_1 en C con $v_1 \neq v$, entonces decodificamos w como v. Si hay hay varias palabras en C más próximas a v, o sea que mantienen la misma distancia mínima a w, entonces seleccionamos arbitrariamente una de las palabras y concluimos que el resultante es el mensaje enviado.

2. *DMV incompleta*, o *DMVI*. De nuevo, si hay una única palabra $v \in C$ más próxima a w, entonces decodificamos w como v. Pero si hay varias palabras en C a la misma mínima distancia desde w, entonces requerimos una retransmisión. En algunos casos podemos aún pedir una retransmisión si la palabra recibida w está demasiado lejos de todas las palabras del código.

Usaremos DMVI para los ejemplos y ejercicios a menos de que se indique lo contrario. Enfaticemos, pues, que DMV no siempre funciona; en particular, si demasiados errores fueron hechos en la transmisión a lo largo de un CSB, entonces DMV falla.

La palabra $v \in C$ más próxima a la palabra recibida w es aquella v para la cual $d(v, w)$ es lo menor posible y por, el Teorema 1, tiene la mayor probabilidad $\phi_p(v, w)$ entre todas las palabras-código de modo que es la más factible palabra-código enviada. Esto es sea visto en el Ejemplo con una tablita que dimos más arriba. Como $d(v, w) = \mathrm{w}(v + w)$, el peso del patrón de error $u = v + w$, el Teorema 1 puede ser presentado como sigue:

$$\phi_p(v_1, w) \leq \phi_p(v_2, w) \text{ si y solo si } \mathrm{w}(v_1 + w) \geq \mathrm{w}(v_2 + w);$$

0 sea, *la más factible palabra-código enviada es la que tiene el patrón de error de menor peso.*

Luego, la estrategia en DMVI es examinar el patrón de error $v + w$ para todas las palabras-código v, y seleccionar la v que produce el patrón de error de menor peso.

Ejemplo. Supongamos$|M| = 2$, y seleccionemos $n = 3$ con $C = \{000, 111\}$. Si $v = 000$ es transmitida, ¿cuándo concluirá la DMVI esto correctamente? ¿y cuándo concluirá la DMVI incorrectamente que 111 fue enviada? Consideremos la siguiente tabla:

La primera columna de la tabla lista todas las posibles palabras que pueden ser recibidas. Esto da todo $\mathbb{K}^3$. La segunda y tercer columnas listan los patrones de error $v + w$ para cada palabra v en el código C. Como DMVI seleccionará el patrón de error de menor peso, pusimos un asterisco al lado de la entrada en la segunda o tercera columna que produzca ese menor peso. En la última columna indicamos la palabra v en el código C correspondiente a la columna en la que el asterisco quedó ubicado. Esta es la palabra v que la DMVI concluye que fue la más

Recibida w	Patrones $000 + w$	de error $111 + w$	Decodificada v
000	000∗	111	000
100	100∗	011	000
010	010∗	101	000
001	001∗	110	000
110	110	001∗	111
101	101	010∗	111
011	011	100∗	111
111	111	000∗	111

Cuadro 1.1.: **Tabla I**

factible palabra enviada. Luego, la DMVI concluye correctamente que 000 fue enviada si 000, 100, 010 ó 001 es recibida (las primeras cuatro filas de la tabla). Y la DMVI concluye incorrectamente que 111 fue enviada si 110, 101, 011 ó 111 fue recibida, (cuatro últimas filas de la tabla).

Ejemplo. Supongamos que $|M| = 3$ y seleccionemos

$$C = \{0000, 1010, 0111\}$$

con $n = 4$. Construimos una tabla para DMVI como arriba, excepto que si dos o más entradas en las columnas de patrones de error tienen el mismo menor peso, entonces no colocamos un asterisco en tal fila y no colocamos nada significativo en la columna de decodificación para v, (apenas indicado por "−"). Esto quiere decir para DMVI que se requiere una retransmisión siempre y cuando hay un empate para el menor peso de un patrón de error.

Ejercicios.

26. Sea $|M| = 2$, $n = 3$ y $C = \{001, 101\}$. Si $v = 001$ es enviada, ¿cuándo concluirá la DMVI esto correctamente y cuándo concluirá incorrectamente que 101 fue enviada?

27. Sea $|M| = 3$ y $n = 3$. Para cada palabra w en K^3 que puede ser recibida, hallar la palabra w en el código $C = \{000, 001, 110\}$ que la DMVI concluye que fue enviada.

28. Construir una tabla de DMVI para cada uno de los siguientes códigos:

    a) $C = \{101, 111, 011\}$

    b) $C = \{000, 001, 010, 011\}$

    c) $C = \{0000, 0001, 1100\}$

    d) $C = \{0000, 1001, 0110, 1111\}$

    e) $C = \{00000, 11111\}$

    f) $C = \{00000, 11100, 00111, 11011\}$

    g) $C = \{00000, 11110, 01111, 10001\}$

    h) $C = \{000000, 101010, 010101, 111111\}$

Recibida w	Patrones $0000 + w$	de $1010 + w$	error $0111 + w$	Decodificada v
0000	0000∗	1010	0111	0000
1000	1000	0010	1111	–
0100	0100∗	1110	0011	0000
0010	0010	1000	0101	–
0001	0001∗	1011	0110	0000
1100	1100	0110	1011	–
1010	1010	0000∗	1101	1010
1001	1001	0011	1110	–
0110	0110	1100	0001∗	0111
0101	0101	1111	0010∗	0111
0011	0011	1001	0100∗	0111
1110	1110	0100∗	1001	1010
1101	1101	0111	1010∗	0111
1011	1011	0001∗	1100	1010
0111	0111	1101	0000∗	0111
1111	1111	0101	1000∗	0111

Cuadro 1.2.: **Tabla II**

Recordemos que que tenemos que seleccionar n y C. Algunas selecciones son mejores que otras. Listamos tres criterios importantes para evaluar buenas selecciones.

1. Palabras más largas toman más tiempo en ser transmitidas, de modo que n debería no ser demasiado grande. O sea, la razón de información debería ser tan próxima a 1 como sea posible.

2. Con muchos mensajes siendo recibidos por segundo, si $|C|$ es grande, digamos unos tantos miles o así, el procedimiento de DMVI consumiría demasiado tiempo en ser implementado. Afortunadamente, ciertas selecciones ingeniosas de C admiten métodos muy rápidos para DMVI.

3. Si se cometen muchos errores en la transmisión, DMV no funcionará. O sea, la palabra que la DMV concluye que fue enviada no será la misma palabra que fue enviada. Así, C debería ser elegido de modo que la probabilidad de que la DMV funcione sea bien alta, (ver la siguiente sección).

Por lo tanto, afirmamos que *la meta principal de la teoría de códigos es hallar conjuntos C de palabras que son adecuados cuando se juzgan a partir de los tres criterios arriba mencionados.* Nuestros esfuerzos se dirigen, pues, en esa dirección.

1.9. Confiabilidad de la DMV

Supongamos que n y C fueron seleccionados. Damos un procedimiento para determinar la probabilidad $\theta_p(C, v)$ que si v fue enviada en un CSB de probabilidad p, entonces la DMVI concluye correctamente que v fue enviado.

Hallemos el conjunto $L(v)$ de todas las palabras en K^n que estén más cerca de v que cualquier otra palabra de C. Luego, la $\theta_p(C, v)$ es la suma de todas las probabilidades $\phi_p(v, w)$ cuando w varía en $L(v)$. O sea,

$$\theta_p(C, v) = \sum_{w \in L(v)} \phi_p(v, w).$$

Nótese que $L(v)$ es precisamente el conjunto de palabras v de K^n para las cuales, si recibidas, la DMVI concluye correctamente que v fue enviada. Podemos hallar $L(v)$ a partir de la tabla de DMVI, construida como lo fue en las Tablas I y II. En cada fila de tal tabla, donde v es decodificada en la última columna, la palabra w en la primera columna de esa fila está en $L(v)$. Y así se obtienen todas las palabras de $L(v)$.

También, obsérvese que $\theta_p(C,\ v)$ es la suma sobre todas las palabras w en $L(v)$ de las probabilidades de patrones de error $v + w$ que ocurren durante la transmisión.

θ_p puede ser usada para comparar dos códigos, juzgándolos por el tercer criterio dado arriba. Sin embargo, debe notarse que $\theta_p(C, v)$ está definida ignorando la posibilidad de retransmisión cuando la palabra recibida es equidistante de dos palabras-código. Esto conduce a algunas anomalías (tales como $\theta_p(\mathbb{K}^n, v) > \theta_p(C, u)$, para cualquier v en K^n y $u \in C$, donde C es el código de control de paridad formado a partir de $\mathbb{K}^n$), pero es una primera aproximación razonable para una medida de confiabilidad. Ciertamente, $\theta_p(C, v)$ es una cota inferior para la probabilidad de que v sea decodificada correctamente.

Ejemplo. Supongamos $p = ,9$, $|M| = 2$, $n = 3$ y $C = \{000, 111\}$, como en un ejemplo anterior. Si la palabra $v = 000$ es enviada, computamos la probabilidad de que la DMVI concluya correctamente esto mismo luego de una transmisión. De la Tabla I sale que v es decodificada en las primeras cuatro filas, de modo que el conjunto $L(000)$, (palabras de $\mathbb{K}^3$ más próximas a $v = 000$ que a 111) es

$$L(000) = \{000, 100, 010, 001\}.$$

Luego,

$$\begin{aligned}\theta_p(C, 000) &= \phi_p(000, 000) + \phi_p(000, 100) + \phi_p(000, 010) + \phi_p(000, 001)\\ &= p^3 + p^2(1-p) + p^2(1-p) + p^2(1-p)\\ &= p^3 + 3p^2(1-p)\\ &= 0{,}972 \text{ (suponiendo que } p = ,9).\end{aligned}$$

Si $v = 111$ es transmitida, computamos la probabilidad de que la DMVI lo concluya correctamente luego de una transmisión. Primero,

$$L(111) = \{110, 101, 011, 111\},$$

de modo que

$$\begin{aligned}\theta_p(C, 111) &= \phi_p(111, 110) + \phi_p(111, 101) + \phi_p(111, 011) + \phi_p(111, 111)\\ &= p^2(1-p) + p^2(1-p) + p^3\\ &= 3p^2(1-p) + p^3\\ &= 0{,}972 \text{ (suponiendo que } p = ,9).\end{aligned}$$

Ejercicios.

29. Suponga que $p = 0{,}9$, $|M| = 2$, $n = 3$ y $C = \{001, 101\}$, como en el Ejercicio 28.

    a) Si $v = 001$ es enviada, hallar la probabilidad de que la DMVI concluya correctamente esto mismo luego de la transmisión.

    b) Repita la parte (a) para $v = 101$.

Ambas respuestas en este ejercicio son $\theta_p(C, v) = 0{,}9$. Comparando esto con los resultados del ejemplo anterior, concluimos que como $0{,}9 < 0{,}972$, el código $C = \{000, 111\}$ es mejor que el código $C = \{001, 101\}$, al menos cuando juzgados bajo el tercer criterio mencionado. Nuestro método provee un procedimiento (no obstante, algo ineficiente cuando n es grande) para determinar cuando la probabilidad de que la DMVI funciona es elevada. Afortunadamente, la mayoría de los códigos que se diseñan en un curso de teoría de códigos están estructurados de modo que el cálculo de esta probabilidad es más simple.

Ejemplo. Supongamos que $p = 0{,}9$, $|M| = 3$, $n = 4$ y

$$C = \{0000, 1010, 0111\},$$

como en la Tabla II. Para cada $v \in C$ computamos $\theta_p(C, v)$.

1.
$$v = 1000$$
$$L(0000) = \{0000, 0100, 0001\} \text{ (de la tabla II)}$$
$$\theta_p(C, v) = \phi_p(0000, 0000) + \phi_p(0000, 0100) + \phi_p(0000, 0001)$$
$$= p^4 + p^3(1-p) + p^3(1-p) + p^3(1-p)$$
$$= p^4 + 3p^3(1-p) = 0{,}8019$$

2.
$$v = 1010$$
$$L(1010) = \{1010, 1110, 1011\} \text{ (de la tabla II)}$$
$$\theta_p(C, v) = \phi_p(1010, 1010) + \phi_p(1010, 1110) + \phi_p(1010, 1011)$$
$$= p^4 + p^3(1-p) + p^3(1-p) + p^3(1-p)$$
$$= p^4 + 3p^3(1-p) = 0{,}8019$$

3.
$$v = 0111$$
$$L(0111) = \{0110, 0101, 0011, 1101, 0111, 1111\}$$
$$\theta_p(C, v) = \phi_p(0111, 0110) + \phi_p(0111, 0101) + \phi_p(0111, 0011)$$
$$+\phi_p(0111, 1101) + \phi_p(0111, 0111) + \phi_p(0111, 1111)$$
$$= p^3(1-p) + p^3(1-p) + p^3(1-p) + p^2(1-p)^2 + p^4 + p^3(1-p)$$
$$= p^4 + 4p^3(1-p) + p^2(1-p)^2 = 0{,}9558$$

Examinando las tres probabilidades vemos que la probabilidad de que la DMVI concluya correctamente que 0111 fue enviada no es tan mala. Sin embargo, la probabilidad de que la DMVI concluya correctamente que 0000 ó 1010 fueron enviadas es horrible. Por lo tanto, al menos por el tercer criterio de arriba, $C = \{0000, 1010, 0111\}$ no es una selección especialmente buena para un código.

Ejercicios.

30. Suponga que $p = 0{,}9$ y que $C = \{000, 001, 110\}$. Si $v = 110$ es enviada, halle la probabilidad de que la DMVI lo concluya correctamente, y la probabilidad de que la DMVI concluya incorrectamente que 000 fue enviada.

31. Para cada uno de los códigos C del Ejercicio 28 de la página 20, compute $\theta_p(C, v)$, para cada $v \in C$ usando $p = 0{,}9$. (Use las tablas que construyó en el Ejercicio 30).

1.10. Códigos detectores de errores

Ahora haremos precisa la noción de cuando un código C detecta errores. Recordar que si $v \in C$ fue enviada y w en $\mathbb{K}^n$ es recibida entonces $u = v + w$ es un patrón de error. Cualquier palabra u en $\mathbb{K}^n$ puede ocurrir como patrón de error, y deseamos saber qué patrones de error C detecta.

Decimos que el código C *detecta* un patrón de error u si y solo si $v + u$ no es una palabra-código para cada $v \in C$. En otras palabras, u es detectada si para toda palabra transmitida v, el decodificador al recibir $v + u$, puede reconocer que no es una palabra-código y de aquí que algún error ha ocurrido.

Ejemplo. Sea $C = \{001, 101, 110\}$. Para el patrón de error $u = 010$, computamos $v + 010$ para todo $v \in C$:

$$001 + 010 = 011, \quad 101 + 010 = 111, \quad 110 + 010 = 100.$$

Ninguna de las palabras $011, 111$ ó 100 está en C, de modo que C detecta el patrón de error 010. Por otra parte, para el patrón de error 100 hallamos:

$$001 + 100 = 101, \quad 101 + 100 = 001, \quad 110 + 100 = 010.$$

Como al menos una de estas sumas está en C, entonces C no detecta el patrón de error 100.

Ejercicios.

32. Sea $C = \{001, 101, 110\}$. Determine si C detecta el patrón de error (a) 011; (b) 000.

33. Para cada uno de los siguientes códigos C, determine si C detecta o no u, en cada caso:

    a) $C = \{00000, 10101, 00111, 11100\}$
    (i) $u = 10101$ (ii) $u = 01010$ (iii) $u = 11011$

    b) $C = \{1101, 0110, 1100\}$
    (i) $u = 0010$ (ii) $u = 0011$ (iii) $u = 1010$

c) $C = \{1000, 0100, 0010, 0001\}$
(i) $u = 1001$ (ii) $u = 1110$ (iii) $u = 0110$

34. ¿Cuáles patrones de error detectará $C = K^n$?

35. *a*) Sea C un código que contiene la palabra cero como palabra-código. Pruebe que si el patrón de error u es una palabra-código, entonces C no detecta u.

b) Pruebe que ningún código detecta el patrón de error cero.

La tabla construida para DMVI puede ser usada para determinar cuáles patrones de error un código C detecta. La primera columna lista cada palabra de $\mathbb{K}^n$. Entonces la primera columna puede reinterpretarse como constituida por todos los posibles patrones de error, en cuyo caso las columnas de "patrón de error" en la tabla de DMVI contienen las sumas $v + u$, para todo $v \in C$. Si en cualquier fila particular ninguna de las sumas son palabra-código de C, entonces C detecta el patrón de error en la primera columna de esa fila.

Ejemplo. Consideremos el código $C = \{000, 111\}$ con la Tabla I de DMVI. Todos los patrones de error u posibles están en la primera columna. Para un u dado, todas las sumas $v + u$ cuando v recorre C están en la segunda y tercera columnas de la fila rotulada con u. Si ninguna de estas entradas están en C (o sea, ni 000 ni 111), entonces C detecta u. Luego, C detecta los patrones de error 100, 010, 001, 110, 101 y 011, como puede verse al inspeccionar las filas 2 a 7 de la tabla, pero no detecta los patrones 000 y 111.

Ejercicios.

36. Detemine los patrones de error detectados por cada palabra del Ejercicio 28.

Un método alternativo y mucho más rápido para hallar los patrones de error que C puede detectar es primero hallar los patrones de error que C no detecta; luego todos los patrones de error remanentes pueden ser detectados por C. Claramente, para todo par de palabras-código v y w, si $e = v + w$, entonces e no puede ser detectado, pues $v + e = w$, que es palabra-código. Así, el conjunto de patrones de error que no puede ser detectado por C es el conjunto de todas las palabras que pueden ser escritas como la suma de dos palabras-código.

Ejemplo. Consideremos el código $C = \{000, 111\}$. Como

$$000 + 000 = 000, \quad 000 + 111 = 111 \quad \text{y } 111 + 111 = 000,$$

el conjunto de patrones de error que no pueden ser detectados es $\{000, 111\}$. Por lo tanto, todos los patrones en $K^3 \setminus \{000, 111\}$ pueden ser detectados.

Ejemplo. Sea $C = \{1000, 0100, 1111\}$. Como $1000 + 1000 = 0000$, $1000 + 0100 = 1100$, $1000 + 1111 = 0111$ y $0100 + 1111 = 1011$, el conjunto de patrones de errores que no puede ser detectado por C es $\{0000, 1100, 0111, 1011\}$. Por lo tanto, todos los patrones de error en $K^4 \setminus \{0000, 1100, 0111, 1011\}$ pueden ser detectados.

Ejercicios.

37. Halle los patrones de error detectados por cada uno de los códigos C del Ejercicio 28 y compare sus respuestas con las del Ejercicio 32.

Existe también una forma de determinar algunos patrones de error que el código C detecta sin ningún control manual. Primero, tenemos que introducir otro número asociado a C.

Para un código C que contiene al menos dos palabras, la *distancia* de C es el menor de los números $d(v, w)$, con v y w formando todos los pares de palabras diferentes de C. Nótese que como $d(v, w) = \mathrm{w}(v + w)$, la distancia de un código es el menor valor de $\mathrm{w}(u + w)$ con v y w recorriendo todas las palabras-código posibles y satisfaciendo $v \neq w$.

Ejemplo. Sea $C = \{0000, 1010, 0111\}$. Entonces, $d(0000, 1010) = 2$, $d(0000, 0111) = 3$ y $d(1010, 0111) = 3$. Luego, la distancia de C es 2.

Ejercicios.

38. Hallar la distancia de los 8 códigos del Ejercicio 28.

39. Hallar la distancia del código formado al agregar un dígito de control de paridad a las palabras de K^n.

Ahora podemos enunciar un teorema que ayuda a identificar muchos de los patrones de error que un código detecta.

Teorema 2. *Un código C de distancia d detecta al menos todos los patrones de error de peso $\leq d - 1$. Más aún, hay al menos un patrón de error de peso d que C no detecta.*

Nótese que C puede detectar algunos patrones de error de peso d o mayor, pero que no detecta todos los patrones de error de peso d.

Demostración. Sea u un patrón de error no nulo con $\mathrm{w}(u) \leq d - 1$, y sea $v \in C$. Luego

$$d(v, v + u) = \mathrm{w}(v + v + u) = \mathrm{w}(u) \leq d - 1.$$

Como C tiene distancia d, $v + u \notin C$. Por lo tanto, C detecta u. De la definición de d, hay palabras-código $v, w \in C$ con $d(v, w) = d$. Consideremos el patrón de error $u = v + w$. Luego, $w = v + u \in C$, de modo que C no detectará el patrón de error u de peso d. □

Un código es *t-detector*, o corrector de t errores, si detecta todos los patrones de error de peso a lo sumo t y no detecta al menos un patrón de error de peso $t + 1$. Así, en vista del Teorema 2, si un código tiene distancia d, entonces es un código $d - 1$-detector.

Ejemplo. El código $C = \{000, 111\}$ tiene distancia $d = 3$. Por el Teorema 2, C detecta todos los patrones de error de peso 1 ó 2, y C no detecta el único patrón de error de peso 3: 111. El único patrón de error no cubierto por el Teorema 2 es 000. Pero por el Ejercicio 34(b) sabemos que 000 no es detectado.

El Teorema 2 no previene que un código C detecte patrones de error de peso d o mayor, De hecho, C usualmente detecta algunos de esos patrones.

Ejemplo. El código $C = \{001, 101, 110\}$ tiene distancia $d = 1$. Como $d - 1 = 0$, el Teorema 2 no nos ayuda a determinar que patrones de error C detecta. Pero tampoco nos dice de que no haya al menos un patrón de error de peso $d = 1$ que C no detecta. Como se vió en el primer ejemplo de esta sección, tal patrón de error es 100. Notar sin embargo que C no detecta el patrón de error 010 de peso $d = 1$.

Ejercicios.

40. El código $C = \{0000, 1010, 0111\}$ tiene distancia $d = 2$. Usando el Ejercicio 34 muestre que el patrón de error 1010 no es detectado. Muestre que este es el único patrón de error de peso 2 que C no detecta. Halle los patrones de error que C detecta.

41. Halle los patrones de error que detecta el código C_3 ejemplificado en la Sección 2. Observe que C_3 es 1-corrector.

42. Para cada código C del Ejercicio 28, halle los patrones de error que C detecta, de acuerdo con el Teorema 2.

43. Sea C un código que consiste de todas las palabras de largo 4 que tengan peso par. Halle los patrones de error que C detecta.

1.11. Códigos correctores de errores

Si una palabra v en un código C fue transmitida en un CSB y si w es recibida, resultando en un patrón de error $u = v + w$, entonces la DMVI correctamente concluye que v fue enviada al suponer que w está más cerca de v que cualquier otra palabra de C. Si esto acontece cada vez que el patrón de error u ocurre, sin importar qué palabra-código es transmitida, entonces decimos que C *corrige* el patrón de error u. O sea, *un código C corrige el patrón de error u si, para toda $v \in C$, $v + u$ está más cerca de v que cualquier otra palabra de C*. También se dice que un código es *t-corrector*, o *corrector de t-errores*, si corrige todos los patrones de error de peso a lo sumo t, pero no corrige algún patrón de error de peso $t + 1$.

Ejemplo. Sea $C = \{000, 111\}$.

1. Tomemos el patrón de error $u = 010$. Para $v = 000$,

$$d(000, v + u) = d(000, 010) = 1,$$
$$d(111, v + u) = d(111, 010) = 2.$$

   Y para $v = 111$,

$$d(000, v + u) = d(000, 101) = 2,$$
$$d(111, v + u) = d(111, 101) = 1.$$

   Luego, C corrige el patrón de error 010.

2. Ahora tomemos el patrón de error $u = 110$. Para $v = 000$,

$$d(000, v + u) = d(000, 110) = 2,$$
$$d(111, v + u) = d(111, 110) = 1.$$

   Como $v + u$ no está más cerca de $v = 000$ que de 111, C no corrige el patrón de error 110.

La tabla de DMVI puede ser usada para determinar cuáles patrones de error un código C corrige. En cada columna de patrones de error de la tabla, cada patrón de error posible (o sea cada palabra de K^n) ocurre una y solo una vez (pues si el patrón de error u ocurre dos veces en una columna de una palabra-código v, entonces u ocurre en distintas filas correspondientes a palabras recibidas, digamos w_1 y w_2. Luego $u = v + w_1 = v + w_2$, lo cual es imposible para $w_1 \neq w_2$). Además colocamos un asterisco al lado del patrón de error, en la columna correspondiente a una palabra-código v en la tabla de DMVI, precisamente cuando $v + u$ está más

cerca de v que de cualquier otra palabra-código. Por lo tanto, un patrón de error u es corregido si un asterisco es colocado al lado de u en *cada* columna de la tabla de DMVI.

Ejemplo. Para el código $C = \{000, 111\}$, la tabla de DMVI es la Tabla I. En cada fila de la tabla en la que el patrón de error 010 ocurre (filas 3 y 6), la DMVI correctamente concluye que la palabra v fue enviada. Además, en al menos una fila (fila 4) donde el patrón de error 110 ocurre, si 111 fue enviada y 001 es recibida, la DMVI incorrectamente concluye que 000 fue enviada. Nótese que este código corrige los patrones de error 000, 100, 010 y 001, que reciben un asterisco cada vez que ocurren.

Ejemplo. Sea $C = \{0000, 1010, 0111\}$. La tabla de DMVI para C es la Tabla II. El código C no corrige el patrón de error $u = 1010$. Este patrón de error ocurre en las filas en las que $w = 0000$, 1010 y 1101. En un solo caso, con $w = 1101$, la DMVI correctamente concluye que la palabra-código v fue enviada. Nótese que el patrón de error $u = 1010$ recibe un asterisco sólo en la columna para $v = 0111$ y no en las otras dos columnas. C sí corrige los patrones de error 0000, 0100 y 0001.

Ejemplo. Sea $C = \{001, 101, 110\}$. ¿Corrige C el patrón de error $u = 100$? Construimos solo las tres filas de la tabla de DMVI en las que 100 aparece. Como $u = v + w$ y conocemos u y v, podemos hallar las palabras recibidas a partir de $w = u + v$. Nótese que $u = 100$ no recibe un asterisco en todas las columnas de la siguiente tabla, de modo que C no corrige 100.

Recibida w	Patrones $001 + w$	de $101 + w$	error $110 + w$	Decodificada v
101	100	000∗	011	101
001	000∗	100	111	001
010	011	111	100∗	110

Ejercicios.

44. Sea $C = \{001, 101, 110\}$. ¿Corrige C el patrón de error $u = 000$?

45. Pruebe que ese mismo patrón de error no puede ocurrir más de una vez en una fila dada de la tabla de DMVI.

46. Pruebe que el patrón de error nulo es siempre corregido.

47. ¿Cuáles patrones de error corregirá el código $C = \mathbb{K}^n$?

La distancia de un código puede ser usada para diseñar una prueba de corrección de errores que evite al menos algunos de los controles manuales en la tabla de DMVI. Recordar que el símbolo $\lfloor x \rfloor$ denota el mayor entero menor o igual que el número real x. Por ejemplo, $\lfloor 5/2 \rfloor = 2$, $\lfloor 3 \rfloor = 3$ y $\lfloor 1/2 \rfloor = 0$.

Teorema 3. *Un código de distancia d corrige todos los patrones de error de peso $\leq \lfloor (d-1)/2 \rfloor$. Más aún, hay al menos un patrón de error de peso $1 + \lfloor (d-1)/2 \rfloor$ que C no corrige.*

Demostración. Sea u un patrón de error con $\mathrm{w}(u) \leq (d-1)/2$. Sean v y w palabras en C con $w \neq v$. Queremos mostrar que $d(v, v+u) \leq d(w, v+u)$. Usando la desigualdad triangular y

continuando:

$$\begin{aligned} d(w, v+u) + d(v+u, v) &\geq d(w, v) \\ d(w, v+u) + \mathrm{w}(u) &\geq 2\mathrm{w}(u) + 1 \\ d(w, v+u) &\geq \mathrm{w}(u) + 1 \\ &\geq d(v, v+u) + 1. \end{aligned}$$

(La segunda desigualdad usó $\mathrm{w}(u) = d(v+u, v)$ y $2\mathrm{w}(u) + 1 \leq d$).

Por lo tanto, C corrige u. Ahora bien, sean v y w palabras-código con $d(v, w) = d$.

Formemos el patrón de error u cambiando $d-1-\lfloor (d-1)/2 \rfloor$ unos (1) cualesquiera en $v+w$ ceros (0) correspondientes. En este caso,

$$\begin{aligned} d(v, v+u) &= \mathrm{w}(u) = 1 + \lfloor (d-1)/2 \rfloor, \text{ y} \\ d(w, v+u) &= \mathrm{w}(w+v+u) = d(v+w, u) \\ &= d - (1 + \lfloor (d-1)/2 \rfloor). \end{aligned}$$

Si d es impar, digamos $d = 2t+1$, entonces

$$\begin{aligned} d(v, v+u) &= \mathrm{w}(u) = 1 + (2t)/2 = 1 + t, \text{ y} \\ d(w, v+u) &= 2t + 1 - (1+t) = t, \end{aligned}$$

de modo que $d(v, v+u) > d(w, v+u)$. Y si d es par, digamos $d = 2t$, entonces

$$\begin{aligned} d(v, v+u) &= 1 + \lfloor (t - 1/2) \rfloor = t, \text{ y} \\ d(w, v+u) &= 2t - t = t. \end{aligned}$$

En cualquier caso, $d(v, v+u) \geq d(w, v+u)$, y entonces $v+u$ no está más cerca de v que la palabra-código w. Luego, C no corrige el patrón de error u. □

En relación al Teorema 3, es claro que cualquier código de distancia d es $\lfloor (d-1)/2 \rfloor$-corrector, o sea corrector de $\lfloor (d-1)/2 \rfloor$ errores.

Ejemplo. El código $C = \{000, 111\}$ tiene distancia $d = 3$. Como $\lfloor (d-1)/2 \rfloor = 1$, el Teorema 3 asegura que C corrige todos los patrones de error de peso 0 ó 1. Como dijimos en un ejemplo anterior, C no corrige los patrones de error 000, 100, 010 y 001. El patrón de error 110 tiene peso $1 + \lfloor (d-1)/2 \rfloor = 2$ y vimos que C no corrige 110.

El Teorema 3 no previene que un código C de distancia d corrija patrones de error de peso mayor que $\lfloor (d-1)/2 \rfloor$.

Ejemplo. Sea $C = \{001, 101\}$. Su distancia es $d = 1$. El patrón de error $u = 011$ tiene peso 2, que es mayor que $1 + \lfloor (d-1)/2 \rfloor = 1$. Como la siguiente parte de la tabla de DMVI muestra, C corrige $u = 011$.

Recibida w	Patrones $001 + w$	de error $101 + w$	Decodificada v
010	011∗	111	001
110	111	011∗	101

Ejercicios.

48. Para cada uno de los códigos C del Ejercicio 28, (página 20): (i) determine los patrones de error que C corrige (las tablas de DMVI fueron construidas en el Ejercicio 28) y (ii) halle los patrones de error que el Teorema 3 garantiza que C corrige.

49. Use la técnica descrita en el último ejemplo para decidir si los siguientes patrones de error son corregidos por los códigos que les acompañan:

    (a) $C = \{000000, 100101, 010110, 001111, 110011, 101010,$

    $011001, 111000\}$

    (i) $u = 001000$ (ii) $u = 000010$ (iii) $u = 100100$

    (b) $C = \{1001011, 0110101, 1110010, 1111111\}$

    (i) $u = 0100000$ (ii) $u = 0101000$ (iii) $u = 1100000$

50. Para cada código del Ejercicio 28, hallar un patrón de error de peso $\lfloor (d-1)/2 \rfloor + 1$ que C no corrija.

51. Sea C el código que consiste de todas las palabras de largo 4 y peso par. Determine los patrones de error que C corrige.

52. Sean u_1 y u_2 patrones de error de largo n, y supongamos que u_1 y u_2 concuerdan al menos en las posiciones en las que 1 ocurre en u_1. Pruebe que si el código C corrige u_2, entonces también corrige u_1.

CAPÍTULO 2
De cómo se usa el álgebra lineal binaria

En este capítulo, introducimos una amplia clase de códigos. De hecho, virtualmente cada código que consideremos pertenecerá a esta clase. Podremos poner en juego algunas herramientas matemáticas poderosas para resolver algunos de los problemas de teoría de códigos discutidos en el Capítulo 1, cuando son aplicados a códigos de esta clase.

Se dice que un código C es *lineal* si $v + w$ es una palabra de C siempre que v y w están en C. O sea, un código lineal es un código que es cerrado bajo la adición de palabras. Por ejemplo, $C = \{000, 111\}$ es un código lineal, pues las cuatro sumas

$$\begin{aligned} 000 + 000 &= 000, \quad 111 + 000 = 111, \\ 000 + 111 &= 111, \quad 111 + 111 = 000, \end{aligned}$$

están en C. Pero $C_1 = \{000, 001, 101\}$ no es un código lineal, pues 001 y 101 están en C mientras que $001 + 101$ no lo está.

Un código lineal C debe contener la palabra nula. Pues si C es lineal y contiene la palabra v, entonces la suma $v + v = 0$ está en C por clausura de la adición. Sin embargo, como el código C_1 del capítulo 1 muestra, la palabra nula en un código no garantiza que este sea lineal.

Ejercicios.

1. Determine si cada uno de los 8 códigos del Ejercicio 28 de la página 20 son lineales.

Una ventaja que un código lineal tiene sobre los códigos no lineales es que la distancia es fácil de hallar. *La distancia de un código lineal es igual al peso mínimo de las palabras no nulas.* Un ejercicio abajo requiere una prueba elemental de este hecho.

Ejercicios.

2. Muestre que $\{0000, 1100, 0011, 1111\}$ es un código lineal y que su distancia es 2.

3. Halle la distancia de cada código lineal del Ejercicio 12.

4. Pruebe que la distancia de un código lineal C es el peso de una palabra de peso mínimo en C.

Como veremos en las secciones siguientes, los códigos lineales están altamente estructurados y ofrecen numerosas ventajas sobre los códigos arbitrarios discutidos hasta ahora. Aquí hay algunos problemas, tediosos de establecer en general, pero relativamente fáciles para códigos lineales:

1. Para cada código lineal existe un procedimiento de DMV más simple y rápido de usar que el descrito anteriormente (algunos códigos lineales con aún más estructura tienen algoritmos decodificadores bien simples).

2. Codificar un código lineal es más rápido y requiere menos espacio de almacenamiento que los códigos no lineales arbitrarios.

3. Las probabilidades $\theta_p(C, v)$ se computan directamente en códigos lineales.

4. Es fácil describir el conjunto de patrones de error que un código lineal puede detectar.

5. Es mucho más fácil describir el conjunto de patrones de error que un código lineal corrige que describirlo para códigos no lineales arbitrarios.

Las herramientas y técnicas más importantes para estudiar códigos lineales provienen del álgebra lineal. Revisaremos algunos hechos básicos del álgebra lineal e intentaremos mostrar su relevancia en la teoría de códigos. La mayoría de las pruebas que no dependan de productos escalares en K^n son réplicas exactas de pruebas en $\mathbb{R}^n$ y, por lo tanto, se omitirán.

Recordemos que habíamos definido un espacio vectorial (sobre K) como consistente de escalares ($\mathbb{K}$) y de un conjunto $\mathbb{K}^n$ de vectores, o palabras, junto con las operaciones de adición vectorial y multiplicación escalar, que satisfacían diez propiedades (ver Sección 6 del Capítulo 1). Un subconjunto no vacío U de un espacio vectorial V es un *subespacio* de V si U es cerrado bajo la adición vectorial y la multiplicación escalar. O sea, si v y w están en U, entonces $v + w$ y av también lo están, para cualquier a en K. En particular, como los únicos escalares en $\mathbb{K}$ son 0 y 1, U es un subespacio de $\mathbb{K}^n$ si y sólo si U es cerrado bajo la adición. Por lo tanto, C es un código lineal si y sólo si C es un subespacio de $\mathbb{K}^n$. En lo que sigue, usaremos nuestro conocimiento de subespacios para mejorar dramáticamente nuestra técnicas de codificación y decodificación.

2.1. Subespacios expansivo y dual

Dos subespacios del espacio vectorial $\mathbb{K}^n$ proveerán sendos ejemplos de interés en códigos lineales y serán vitales en nuestros desarrollos futuros. Definiciones y resultados serán enunciados para un espacio vectorial arbitrario y luego interpretados para $\mathbb{K}^n$.

Se dice que el vector w es *combinación lineal* de vectores $v_1, v_2, \cdots, v_k$ si existen escalares $a_1, a_2, \cdots, c_k$ tales que

$$w = a_1v_1 + a_2v_2 + \cdots + a_kv_k.$$

El conjunto de todas las combinaciones lineales de un subconjunto dado $S = \{v_1, v_2, \cdots, v_k\}$ de vectores de $\mathbb{K}^n$ es llamado *alcance lineal* de S, denotándose por $\langle S\rangle$. Si S es vacío, definimos $\langle S\rangle = \emptyset$.

El álgebra lineal muestra que para cualquier subconjunto S de un espacio vectorial V, el alcance lineal $\langle S\rangle$ es un subespacio *expandido* o *generado* por S. Para el espacio vectorial $V = \mathbb{K}^n$, poseemos una muy simple descripción de $\langle S\rangle$ que es enunciada en el siguiente teorema. Como $\langle S\rangle$ es un subespacio de $\mathbb{K}^n$, decimos que $\langle S\rangle$ es el código lineal generado por S.

Teorema 4. *Para cualquier subconjunto S de $\mathbb{K}^n$, el código $C = \langle S\rangle$ generado por S consiste precisamente de las siguientes palabras: la palabra nula, todas las palabras en S y todas las sumas de dos o más palabras en S.*

Ejemplo. Sea $S = \{0100, 0011, 1100\}$. El código $C = \langle S \rangle$ generado por S consiste de

$$0000,\ 0100,\ 0100 + 0011 = 0111,\ 0100 + 0011 + 1100 = 1011,$$

$$0011,\ 1100,\ 0100 + 1100 = 1000,\ 0011 + 1100 = 1111.$$

0 sea, $C = \langle S \rangle = \{0000, 0100, 0011, 1100, 0111, 1000, 1111, 1011\}$.

Ejercicios.

5. Para cada uno de los siguientes conjuntos S, liste los elementos del código lineal $\langle S \rangle$:

   a) $S = \{010, 011, 111\}$
   b) $S = \{1010, 0101, 1111\}$
   c) $S = \{0101, 1010, 1100\}$
   d) $S = \{1000, 0100, 0010, 0001\}$
   e) $S = \{11000, 01111, 11110, 01010\}$
   f) $S = \{10101, 01010, 11111, 00011, 10110\}$

Si $v = (a_1, a_2, \ldots, a_n)$ y $w = (b_1, b_2, \ldots, b_n)$ son vectores de $\mathbb{K}^n$, definimos el *producto escalar* $v \cdot w$ de v y w como

$$v \cdot w = a_1 b_1 + a_2 b_2 + \cdots a_n b_n.$$

Notar que $v \cdot w$ es un escalar, no un vector. Por ejemplo, en $\mathbb{K}^5$,

$$\begin{aligned} 11001 \cdot 01101 &= 1 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 \\ &= 0 + 1 + 0 + 0 + 1 \\ &= 0. \end{aligned}$$

Ejercicios.

6. Construya ejemplos en K^5 de cada una de las siguientes leyes:

   a) $u \cdot (v + w) = u \cdot v + u \cdot w$
   b) $a(v \cdot w) = a \cdot v + a \cdot w$

7. Pruebe que las dos leyes del Ejercicio 1 valen en $\mathbb{K}^n$.

Dos vectores v y w son *ortogonales* si $v \cdot w = 0$. El ejemplo anterior muestra que $v = 11001$ y $w = 01101$ son ortogonales en K^5. Para un conjunto dado S de vectores en K^n, decimos que un vector v es *ortogonal al conjunto* S si $v \cdot w = 0$ para todos los $w \in S$. O sea, v es ortogonal a cada vector en S. El conjunto de vectores ortogonales a S es denotado $S^\perp$ y denominado el *complemento ortogonal* de S.

En álgebra lineal se demuestra que para cualquier conjunto S dado en un espacio vectorial V, el complemento ortogonal $S^\perp$ es un subespacio de V. Si $V = \mathbb{K}^n$ y $C = \langle S \rangle$, entonces escribimos $C^\perp = S^\perp$, que es denominado el *código dual* de C.

Ejemplo. Para $S = \{0100, 0101\}$, computamos el código dual $C^\perp$. Debemos hallar todas las palabras $v = (x, y, z, w)$ en K^4 tales que ambas ecuaciones

$$v \cdot 0100 = 0$$

$$v \cdot 0101 = 0$$

valen. Computando el producto escalar, tenemos

$$y = 0 \text{ e } y + w = 0.$$

Luego, $y = w = 0$ pero x y z pueden valer 0 ó 1. Escribiendo todas estas posibilidades para v, obtenemos

$$C^\perp = S^\perp = \{0000, 0010, 1000, 1010\}.$$

Ejercicios.

8. Halle el código dual $C^\perp$ para cada uno de los códigos del Ejercicio 1.
9. Halle un ejemplo de palabra no nula v tal que $v \cdot v = 0$. ¿Qué puede decir respecto al peso de tal palabra?
10. ¿Será que para cualquier subconjunto S de un espacio vectorial V vale que $(S^\perp)^\perp = \langle S \rangle$? Use el ejercicio anterior para verificar su respuesta a esta pregunta en $\mathbb{K}^4$.
11. Pruebe que $\langle S \rangle \subseteq (S^\perp)^\perp$. (De hecho, $(S)^\perp)^\perp \supseteq \langle S \rangle$ para un código lineal C quiere decir $(C^\perp)^\perp \supseteq C$).

2.2. Independencia lineal, bases, dimensión

Revisemos conceptos importantes del álgebra lineal e ilustremos cómo se aplican estos conceptos a códigos lineales. El objetivo principal ahora es hallar una forma eficiente de describir un código lineal sin tener que listar todas sus palabras.

Un conjunto $S = \{v_1, v_2, \ldots, v_k\}$ de vectores es *linealmente dependiente* si existen escalares $a_1, a_2, \ldots, a_k$ no todos nulos tales que

$$a_1 v_1 + a_2 v_2 + \cdots + a_k v_k = 0.$$

Caso contrario, el conjunto S es *linealmente independiente*.

La prueba para independencia lineal, entonces, es formar la ecuación vectorial pasada usando escalares arbitrarios. Si esta cuestión fuerza todos los escalares $a_1, a_2, \ldots, a_k$ a ser nulos, entonces el conjunto S es linealmente independiente. Si *al menos un* a_i puede ser elegido no nulo, entonces S es linealmente dependiente.

Ejemplo. Probemos $S = \{1001, 1101, 1011\}$ para independencia lineal. Sean a, b y c escalares tal que

$$a(1001) + b(1101) + c(1011) = 0000.$$

Igualando ecuaciones en ambos lados, produce la ecuaciones escalares

$$a + b + c = 0, \;\; b = 0, \;\; c = 0, \;\; a + b + c = 0.$$

Estas ecuaciones fuerzan $a = b = c = 0$. Por lo tanto, S es un conjunto linealmente independiente de palabras de $\mathbb{K}^4$.

Ejemplo. Probemos $S = \{110, 011, 101, 111\}$ para independencia lineal. Consideremos

$$a(110) + b(011) + c(101) + d(111) = 000.$$

Esto produce el sistema de ecuaciones escalares

$$\begin{cases} a + c + d = 0 \\ a + b + d = 0 \\ b + c + d = 0 \end{cases}$$

Sumando estas tres ecuaciones se obtiene $d = 0$. Ahora tenemos $a+c = 0$, $a+b = 0$ y $b+c = 0$. Luego, podemos elegir $a = b = c = 1$. Por lo tanto, S es un conjunto linealmente dependiente.

En álgebra lineal, se demuestra que *cualquier conjunto de vectores* $S \neq \{0\}$ *contiene un subconjunto linealmente independiente máximo*. El próximo ejemplo muestra cómo puede hallarse tal subconjunto.

Ejemplo. Sea $S = \{110, 011, 101, 111\}$. El último ejemplo mostró que S es linealmente dependiente. De hecho, hallamos que

$$1(110) + 1(011) + 1(101) + 0(111) = 000,$$

de modo que podemos poner 101 como combinación lineal de las otras palabras de S:

$$101 = 1(110) + 1(011) + 0(111).$$

En el conjunto linealmente dependiente, si tomamos las palabras en el orden dado llegamos a 101 como la primera palabra que es dependiente de, o sea, es combinación lineal de las palabras precedentes, 110 y 011, en S. Descartando esta palabra, obtenemos un nuevo conjunto $S' = \{110, 011, 111\}$. Ahora S' puede ser probado para independencia lineal. Si S' fuese linealmente dependiente, descartaríamos la primera palabra que sea combinación lineal de las palabras precedentes, así obteniendo un nuevo conjunto S''. Este proceso puede ser repetido hasta que hallemos un nuevo conjunto que sea linealmente independiente; tal conjunto es siempre un conjunto linealmente dependiente máximo del conjunto dado S. En este ejemplo, ese conjunto es S'.

Ejercicios.

12. Verifique independencia lineal para cada uno de los siguientes conjuntos. Si el conjunto es linealmente dependiente, extraiga de S un subconjunto linealmente independiente máximo:

    a) $S = \{1101, 1110, 1011\}$
    b) $S = \{101, 011, 110, 010\}$
    c) $S = \{1101, 0111, 1100, 0011\}$
    d) $S = \{1000, 0100, 0010, 0001\}$
    e) $S = \{1000, 1100, 1110, 1111\}$
    f) $S = \{1100, 1010, 1001, 0101\}$

$g)$ $S = \{0110, 1010, 1100, 0011, 1111\}$

$h)$ $S = \{111000, 000111, 101010, 010101\}$

$i)$ $S = \{00000000, 10101010, 01010101, 11111111\}$

En el primer inciso de este ejercicio, se halla que S es un conjunto linealmente dependiente. Nótese que S contiene la palabra nula. Es siempre cierto que *cualquier conjunto de vectores que contenga al vector nulo es linealmente dependiente.*

Un conjunto no vacío B de vectores de un espacio vectorial es una *base* de V si valen las dos propiedades siguiente:

1. B expande V, (0 sea, $\langle B\rangle = V$).
2. B es un conjunto linealmente independiente.

Notar que *cualquier conjunto linealmente independiente B es automáticamente una base de* $\langle B\rangle$. Además, como un conjunto linealmente dependiente S de vectores que contiene una palabra no nula siempre contiene un conjunto linealmente independiente máximo, podemos extraer de S una base B para $\langle S\rangle$. Si $S = \{0\}$, entonces decimos que la base de S es el conjunto vacío $\emptyset$.

Ejemplo. Sea $S = \{1001, 1101, 1011\}$. En el primer ejemplo de esta sección, hallamos que S es linealmente independiente. Por lo tanto, S es una base para el código

$$C = \langle S\rangle = \{0000, 1001, 1101, 1011, 0100, 0010, 0110, 1111\},$$

que es un subespacio de $\mathbb{K}^4$.

Ejemplo. Sea $S = \{110, 011, 101, 111\}$. En el segundo ejemplo de esta sección hallamos que S es linealmente dependiente. En el siguiente ejemplo extraíamos un subconjunto linealmente independiente $B = S' = \{110, 011, 111\}$ de S. Por lo tanto, B es una base del código $C = \langle S\rangle$.

Estos ejemplos ilustran cómo obtener una base del código $C = \langle S\rangle$ generado por un subconjunto no vacío S de $\mathbb{K}^n$. Para hallar una base del código dual $C^\perp$, extraemos un subconjunto linealmente independiente máximo de $C^\perp$, siguiendo el procedimiento del tercer ejemplo de esta sección.

Ejercicios.

13. Para cada conjunto en el Ejercicio 1, halle una base B para el código $C = \langle S \rangle$ y una base de $B^{\perp}$ del código dual.

El conjunto $B = \{110, 011, 111\}$ no es el único conjunto linealmente independiente máximo de $S = \{110, 011, 101, 111\}$, (ver Ejercicio 8). También el conjunto $B_1 = \{110, 101, 111\}$ es otro tal conjunto de S. Luego, B_1 es también base de un código $C = \langle S \rangle$.

En general, un espacio vectorial tiene usualmente muchas bases. Sin embargo, *todas las bases de un espacio vectorial contienen el mismo número de elementos*. El número de elementos en una tal base de un espacio vectorial es denominado la *dimensión* del espacio.

La dimensión de $\mathbb{K}^n$ es n, pues el conjunto de todas las palabras de largo n y peso 1 forman una base de $\mathbb{K}^n$. En el otro extremo, la base del subespacio es $\emptyset$ y, por lo tanto, tiene dimensión 0.

Ejercicios.

14. Halle las dimensiones de cada código $C = \langle S \rangle$ y su dual $C^{\perp}$ en el Ejercicio 5.

Una base provee una forma eficiente de describir un código lineal. Para un espacio vectorial V, *si* $\{v_1, v_2, \ldots, v_k\}$ *es una base de* V*, entonces cada vector* $w \in V$ *puede ser expresado como una única combinación lineal de los vectores* $v_1, v_2, \ldots, v_k$ *de la base*. O sea, existen escalares bien determinados (únicos) $a_1, a_2, \ldots, a_k$ tal que $w = a_1 v_1 + a_2 v_2 + \cdots + a_k v_k$.

Ejemplo. Escribimos $w = 011$ como única combinación lineal de las palabras en la base $\{110, 001, 100\}$ de K^3. Procuramos dígitos a, b, c tales que

$$a(110) + b(001) + c(100) = 011.$$

Esto produce las ecuaciones escalares

$$a + c = 0, \quad a = 1, \quad b = 1,$$

que tienen la única solución $a = b = c = 1$. Luego, $011 = 1(110) + 1(001) + 1(100)$.

Ejercicios.

15. Escriba cada una de las siguientes palabras en la base $\{1000, 1100, 1110, 1111\}$:
    (a) 0011 (b) 1010 (c) 0111 (d) 0001 (e) 0000.

Ejemplo. El conjunto $S = \{110, 001\}$ es un subconjunto linealmente independiente de $\mathbb{K}^3$. Extendemos S a una base de $\mathbb{K}^3$. Primero, adjuntamos a S cualquier base conocida: $\{100, 010, 001\}$ es una base conveniente para adjuntar a S. La lista resultante de palabras

$$110, 001, 100, 010, 001$$

es entonces reducida a una base de K^3 de acuerdo con el procedimiento del tercer ejemplo de esta sección, resolviendo para 100, 010 ó 001.

Ejercicios.

16. (a) Halle una base de $\mathbb{K}^4$ que contenga $\{1001, 1111\}$. (b) Extienda $\{101010, 010101\}$ a una base de $\mathbb{K}^6$.

Arribamos a dos teoremas que se refieren a las dimensiones de códigos lineales. Si un código lineal C tiene dimensión k y si $\{v_1, v_2, \ldots, v_k\}$ es una base de C, entonces una palabra $w \in C$ puede ser escrita como

$$w = a_1v_1 + a_2v_2 + \cdots + a_kv_k$$

para una selección única de dígitos $a_1, a_2, \ldots, a_k$. Como cada a_i es 0 ó 1, hay 2^k selecciones de $a_1, a_2, \ldots, a_k$, y de aquí 2^k palabras en C.

Teorema 5. *Un código lineal de dimensión k contiene precisamente 2^k palabras.*

El siguiente teorema puede ser probado usando resultados elementales de la teoría de sistemas de ecuaciones lineales.

Teorema 6. *Sea $C = \langle S \rangle$ un código lineal generado por un subconjunto S de $\mathbb{K}^n$. Entonces dimensión(C)+dimensión$(C^\perp) = n$.*

Ejercicios.

17. Revise sus respuestas del Ejercicio 10 con las ecuaciones en el Teorema 6.

18. Sea S un subconjunto de $\mathbb{K}^7$, sea $C = \langle S \rangle$ y supongamos que $C^\perp$ tiene dimensión 3.

    a) Halle la dimensión de $C = \langle S \rangle$.

    b) Halle el número de palabras en C.

19. Sea S un subconjunto de $\mathbb{K}^8$ y supongamos que $\{111110000, 00001111, 10000001\}$ es una base de $C^\perp$. Halle el número de palabras en $C = \langle S \rangle$.

20. El Teorema 6 vale también en el el espacio vectorial $\mathbb{R}^n$ de dimensión n sobre los reales, donde cada vector puede ser escrito en forma única como la suma de un vector en $\langle S \rangle$ y un vector en $S^\perp$, siendo el vector nulo el único vector que $S^\perp$ y $\langle S \rangle$ tienen en común. (Por ejemplo, en $\mathbb{R}^n$ tome $\langle S \rangle$ como el plano xy y $S^\perp$ como el eje de las z). Use $S = \{000, 101\}$ en $\mathbb{K}^3$ para mostrar que esta no es la situación en general en $\mathbb{K}^n$.

2.3. Matrices

Una *matriz de tipo* $m \times n$ (o matriz $m \times n$) es un arreglo rectangular de escalares con m filas y n columnas. Supondremos que el lector está familiarizado con el álgebra de matrices sobre los números reales. En esta sección revisaremos las partes de la teoría elemental de matrices necesarias en teoría de códigos.

Si A es una matriz $m \times n$ y B es una matriz $n \times p$, entonces el *producto* AB de A por B es la matriz $m \times p$ que tiene en su (i, j)-entrada (o sea, la entrada en la fila i y la columna j), el producto escalar de la fila i de A y la columna j de B. Por ejemplo

$$\begin{bmatrix} 1101 \\ 0101 \end{bmatrix} \begin{bmatrix} 101 \\ 011 \\ 101 \\ 100 \end{bmatrix} = \begin{bmatrix} 010 \\ 111 \end{bmatrix}.$$

Nótese que el número de columnas de la primera matriz debe igualar el número de filas de la segunda matriz para que el producto quede definido.

Ejercicios.

21. Halle el producto de cada dos de las siguientes matrices, siempre que este esté bien definido:

$$A = \begin{bmatrix} 11011 \\ 00101 \\ 11011 \end{bmatrix}, \quad B = \begin{bmatrix} 0101 \\ 1001 \\ 1100 \end{bmatrix}, \quad C = \begin{bmatrix} 110110 \\ 011011 \\ 101101 \\ 101011 \end{bmatrix}, \quad D = \begin{bmatrix} 1111 \\ 0101 \\ 1010 \\ 1101 \end{bmatrix}.$$

Las leyes algebraicas usuales para matrices sobre los reales también valen para matrices sobre K. La $m \times n$-*matriz nula* es la matriz $m \times n$ cada una de cuyas entradas es nula. La matriz $n \times n$ (cuadrada) I en la cual la (i, j)-entrada es 1 si $i = j$ y es 0 si $i \neq j$ es llamada la *matriz identidad*. Para cualquier matriz A, vale que $AI = A$ y que $IA = A$. Los próximos tres ejercicios indican tres leyes algebraicas que fallan para matrices sobre $\mathbb{K}$.

Ejercicios.

22. Halle 2×2-matrices A y B sobre $\mathbb{K}$ tales que $AB \neq BA$.

23. Halle 2×2-matrices A y B sobre K, ambas diferentes de la matriz nula, tales que $AB = 0$.

24. Halle 2×2-matrices A, B y C sobre K tales que $AB = AC$, pero $B \neq C$.

Hay dos tipos de *operaciones elementales en filas* que pueden ser realizadas en una matriz sobre $\mathbb{K}$. Ellas son:

1. Intercambio de dos filas cualesquiera;

2. Reemplazo de una fila por la suma de ella misma con otra fila.

Dos matrices son fila-equivalentes si una de ellas puede ser obtenida de la otra por una sucesión de operaciones elementales en filas.

Un 1 en una matriz M (sobre $\mathbb{K}$) es *líder* (de su fila) si no hay ningún 1 a su izquierda. Una columna de M es una *columna líder* si contiene un 1 líder. M está en *forma escalonada* (o en FE) si las filas nulas de M (si es que hay alguna) están todas al fondo (abajo) en M y cada 1 líder está a la derecha de los 1 líderes de las filas anteriores (si hubiese alguna fila anterior). Si M está en FE, decimos que M es una matriz escalonada (o una ME). En ese caso, si cada columna líder contiene exactamente un 1, decimos que M está en *forma escalonada reducida* (o en FER), o que es una matriz escalonada reducida (o MER).

Cualquier matriz A sobre K puede ser puesta en FE o en FER por medio de una sucesión de operaciones elementales de filas. O sea, A es fila-equivalente a una ME o MER, y decimos que esta última matriz fue obtenida por *reducción de filas* de A. Para una A dada, su MER es única, pero A tiene muchas ME.

Ejemplo. Hallemos la MER de la siguiente matriz M:

$$
\begin{aligned}
M = \begin{bmatrix} 1011 \\ 1010 \\ 1101 \end{bmatrix} &\rightarrow \begin{bmatrix} 1011 \\ 0001 \\ 0110 \end{bmatrix} \quad \text{(suma la fila 1 a las filas 2 y 3)} \\
&\rightarrow \begin{bmatrix} 1011 \\ 0110 \\ 0001 \end{bmatrix} \quad \text{(intercambia las filas 2 y 3)} \\
&\rightarrow \begin{bmatrix} 1010 \\ 0110 \\ 0001 \end{bmatrix} \quad \text{(suma la fila 3 a la fila 1)}
\end{aligned}
$$

Ejercicios.

25. Halle las MER de cada una de las cuatro matrices del primer ejemplo de esta sección.

La matriz *transpuesta* de una matriz $m \times n$, A es una matriz $n \times m$, A^T, que tiene la i-ésima columna de A como i-ésima fila. Por ejemplo,

$$\text{si } A = \begin{bmatrix} 1011 \\ 0000 \\ 0110 \end{bmatrix}, \text{ entonces } A^T = \begin{bmatrix} 100 \\ 001 \\ 101 \\ 100 \end{bmatrix}.$$

Precisaremos dos hechos respecto de la transpuesta de matrices A y B: $(A^T)^T = A$ y $(AB)^T = B^T A^T$.

2.4. Bases para $C = \langle S \rangle$ y $C^\perp$

Desarrollamos algoritmos para hallar bases de un código lineal y de su código dual. Estos métodos serán de gran asistencia en nuestro estudio de los códigos lineales.

Sea S un subconjunto no vacío de $\mathbb{K}^n$. Los primeros dos algoritmos proveen una base de $C = \langle S \rangle$, el código lineal generado por S.

Algoritmo A. Formemos la matriz A cuyas filas son las palabras en S. Usemos las operaciones elementales en filas para hallar una FE de A. Luego, las filas no nulas de la ME forman una base de $C = \langle S \rangle$.

El algoritmo funciona, pues las filas de A generan C y las operaciones elementales en filas simplemente intercambian palabras o reemplazan una palabra (fila) por su suma con otra palabra en C. Claramente, las filas no nulas de una matriz en FE son linealmente independientes.

Ejemplo. Hallamos una base del código lineal $C = \langle S \rangle$, para

$$S = \{11101, 10110, 01011, 11010\}.$$

$$A = \begin{bmatrix} 11101 \\ 10110 \\ 01011 \\ 11010 \end{bmatrix} \rightarrow \begin{bmatrix} 11101 \\ 01011 \\ 01011 \\ 00111 \end{bmatrix} \rightarrow \begin{bmatrix} 11101 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix}.$$

La última matriz es una FE de A. Por el Algoritmo A, el conjunto de vectores $\{11101, 01011, 00111\}$ es una base de $C = \langle S \rangle$. Otra FE de A es

$$\begin{bmatrix} 11101 \\ 01100 \\ 00111 \\ 00000 \end{bmatrix}$$

de modo que $\{11101, 01100, 00111\}$ es también una base de $C = \langle S \rangle$. Nótese que el Algoritmo A no produce una base única para $C = \langle S \rangle$, ni las palabras en una base están necesariamente en el conjunto S dado.

Ejercicios.

26. Use el Algoritmo A para hallar una base de $C = \langle S \rangle$, para cada uno de los conjuntos S siguientes:

a) $S = \{010, 011, 111\}$

b) $S = \{1010, 0101, 1111\}$

c) $S = \{0101, 1010, 1100\}$

d) $S = \{1000, 0100, 0010, 0001\}$

e) $S = \{11000, 01111, 11110, 01010\}$

f) $S = \{10101, 01010, 11111, 00011, 10110\}$

g) $S = \{0110, 1010, 1100, 0011, 1111\}$

h) $S = \{111000, 000111, 101010, 010101\}$

i) $S = \{00000000, 10101010, 01010101, 11111111\}$

Algoritmo B. (Hallar una base para C) Formar la matriz A cuyas columnas son las palabras en S. Usar operaciones elementales en filas para colocar A en FE y localizar las columnas líderes en la ME. Luego, las columnas originales de A correspondientes a esas columnas líderes forman una base de $C = \langle S \rangle$.

En álgebra lineal elemental se demuestra que un conjunto linealmente independiente de columnas de una matriz es todavía linealmente independiente luego de aplicar a la matriz una sucesión de operaciones elementales en filas. Es fácil ver que las columnas líderes de la matriz en FE forman un conjunto linealmente independiente.

Ejemplo. Usamos el Algoritmo B para hallar una base de $C = \langle S \rangle$, donde S es como en el último ejemplo.

$$A = \begin{bmatrix} 1101 \\ 1011 \\ 1100 \\ 0111 \\ 1010 \end{bmatrix} \rightarrow \begin{bmatrix} 1101 \\ 0110 \\ 0001 \\ 0111 \\ 0111 \end{bmatrix} \rightarrow \begin{bmatrix} 1101 \\ 0110 \\ 0001 \\ 0000 \\ 0000 \end{bmatrix}, \text{ que está en FE.}$$

Como las columnas 1, 2 y 4 de la ME son columnas líderes, el Algoritmo B dice que las columnas 1, 2 y 4 de A forman una base de $C = \langle S \rangle$. Esta base es $\{11101, 10110, 11010\}$. Nótese que el Algoritmo B tiene la propiedad de producir una base de $C = \langle S \rangle$ cuyos elementos son palabras de un conjunto S dado.

Ejercicios.

27. Use el Algoritmo B para hallar una base de $C = \langle S \rangle$ para cada conjunto S del Ejercicio 26.

Ahora proveeremos un algoritmo para hallar una base del código dual $C^{\perp}$. Será este un algoritmo muy útil en la presentación sucesiva. Además, notar que este algoritmo provee una base para C (pues incluye al Algoritmo A).

Algoritmo C. (Hallar una base para $C^{\perp}$) Formar la matriz A cuyas filas son las palabras de S. Usar operaciones elementales en filas para colocar A en FER. Sea G la matriz $k \times n$ consistente de todas las filas no nulas de la MER. Sea X la matriz $k \times (n-k)$ obtenida de G al eliminar las columnas líderes de G. Formar una matriz $n \times (n-k)$, H, como sigue:

1. en las filas de H correspondientes a las columnas líderes de G, colocar, en orden, las filas de X;

2. en las restantes $n-k$ filas de H, colocar, en orden, las filas de la matriz $(n-k)\times(n-k)$ identidad I_{n-k}.

Luego, las columnas de H forman una base de $C^{\perp}$.

El algoritmo funciona, pues las $n-k$ columnas de H son linealmente independientes, $\dim C^{\perp} = n-k$, y a menos de una permutación de las columnas de G y de las filas de H, $GH = X + X = 0$.

La siguiente descripción del Algoritmo C puede ayudar a recordárnoslo. La matriz G contiene k columnas líderes. Permutemos las columnas de G de modo que las líderes sean puestas en primer lugar. Las demás columnas forman una matriz X. Denotemos la matriz obtenida $G' = [I_k, X]$. El Algoritmo C comienza así:

$$A \to \begin{bmatrix} G \\ O \end{bmatrix} \quad \text{(FER)}$$

donde cada entrada de la matriz O es 0. Permutemos las columnas de G para formar $G' = [I_k, X]$. Formemos una matriz H' como sigue:

$$H' = \begin{bmatrix} X \\ I_{n-k} \end{bmatrix}.$$

Y apliquemos la inversa de la permutación que fue aplicada a las columnas de G, a las filas de H' para formar H.

Ejemplo. Usamos el Algoritmo C para hallar una base de $C^{\perp}$ para el conjunto S del primer ejemplo de esta sección.

$$A = \begin{bmatrix} 11101 \\ 10110 \\ 01011 \\ 11010 \end{bmatrix} \to \begin{bmatrix} 11101 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix} \to \begin{bmatrix} 11010 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix} \to \begin{bmatrix} 10001 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix},$$

que está en FER. Ahora $G = \begin{bmatrix} 100|01 \\ 010|11 \\ 001|11 \end{bmatrix}$, $k = 3$, y $X = \begin{bmatrix} 01 \\ 11 \\ 11 \end{bmatrix}$. Las columnas líderes de G son las columnas 1, 2 y 3, de modo que las filas de X están colocadas en las filas 1, 2 y 3, respectivamente, de la matriz $5\times(5-3)$, H. Las restantes filas de H son completadas con la matriz 2×2 identidad. Luego,

$$H = \begin{bmatrix} 01 \\ 11 \\ \underline{11} \\ 10 \\ 01 \end{bmatrix}.$$

De acuerdo con el Algoritmo C, las columnas de H forman una base de $C^{\perp}$. Nótese, por el Algoritmo A, que las filas de G forman una base de $C = \langle S \rangle$.

Ejemplo. Supongamos $n = 10$ y que tenemos un conjunto S de palabras de $\mathbb{K}^{10}$. Supongamos que la FER de la matriz A del Algoritmo C no tiene filas nulas:

$$G = \begin{bmatrix} 1010010101 \\ 0001010001 \\ 0000100100 \\ 0000001001 \\ 0000000011 \end{bmatrix}.$$

Las columnas líderes de G son las columnas 1, 4, 5, 7 y 9. Permutamos las columnas de G en el nuevo orden 1, 4, 5, 7, 9, 2, 3, 6, 8 ,10 (de modo que las columnas líderes vayan primero) para formar la matriz.

$$G' = \begin{bmatrix} 10000|01111 \\ 01000|00101 \\ 00100|00010 \\ 00010|00001 \\ 00001|00001 \end{bmatrix}.$$

Luego, formamos la matriz H', y finalmente rearreglamos las filas de H en su orden natural para formar la matriz H':

$$H' = \begin{bmatrix} X \\ I \end{bmatrix} = \begin{bmatrix} 01111 \\ 00101 \\ 00010 \\ 00001 \\ 00001 \\ 10000 \\ 01000 \\ 00100 \\ 00010 \\ 00001 \end{bmatrix} \begin{matrix} 1 \\ 4 \\ 5 \\ 7 \\ 9 \\ 2 \\ 3 \\ 6 \\ 8 \\ 10 \end{matrix}; \quad H = \begin{bmatrix} 01111 \\ 10000 \\ 01000 \\ 00101 \\ 00010 \\ 00100 \\ 00001 \\ 00010 \\ 00001 \\ 00001 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix};$$

Por causa del Algoritmo C, las columnas de H forman una base de $C^{\perp}$.

Ejercicios.

28. Use el Algoritmo C para hallar una base de $C^{\perp}$ para cada uno de los códigos $C = \langle S \rangle$, donde los S son como en el Ejercicio 26 de la página 45.

29. Con la notación del Algoritmo C, explique por qué $GH = 0$.

30. Para cada uno de los conjuntos S siguientes, use el Algoritmo C para producir una base B para el código $C = \langle S \rangle$, y la base $B^{\perp}$ para el código dual $C^{\perp}$.

    a) $S = \{000000, 111000, 000111, 111111\}$.

    b) $S = \{1101000, 0110100, 0011010, 0001101, 1000110, 0100011, 1010001\}$.

    c) $S = \{1111000, 0111100, 0011110, 1000111, 1100011, 1110001\}$.

    d) $S = \{100100100, 010010010, 111111111, 000000000\}$.

    e) $S = \{001101, 001000, 001111, 000101, 000001\}$.

2.5. Matrices generadoras y codificación

Pondremos el material de las pasadas secciones a funcionar para hallar una matriz importante para un código lineal y ver cómo esta matriz es usada para transmitir mensajes.

Inicialmente, damos algunas definiciones. El *rango* de una matriz sobre K es el número de filas no nulas de cualquier FE de la matriz. La *dimensión* k del código C es la dimensión de C como subespacio de K^n. Si C también tiene largo (o longitud) n y distancia d, entonces decimos que C es un código (n, k, d)-*lineal*. Estos tres parámetros, (longitud, dimensión y distancia), proveen información vital sobre C.

Si C es un código lineal de largo n y dimensión k, entonces cualquier matriz cuyas filas forman una base de C es una *matriz generadora* para C. Nótese que una matriz generadora para C debe tener k filas y n columnas, y debe tener rango k.

Teorema 7. *Una matriz G es una matriz generadora de algún código lineal si y solo si las filas de G son linealmente independientes. O sea, si y solo si el rango de G iguala el número de filas de G.*

Como las matrices fila-equivalentes tienen el mismo rango, llegamos al siguiente teorema.

Teorema 8. *Si G es una matriz generadora de algún código lineal C, entonces cualquier matriz fila-equivalente a G es también una matriz generadora de G. En particular, todo código lineal tiene una matriz generadora en FER.*

Para hallar una matriz generadora de un código lineal C, formamos la matriz cuyas filas son las palabras de C. Como $C = \langle S \rangle$, uno de los Algoritmos A ó B puede ser usado para producir una base de C. La matriz cuyas filas son los vectores de esta base es una matriz generadora para C.

Ejemplo. Hallemos una matriz generadora para el código $C = \{0000, 1110, 0111, 1001\}$. Usando el Algoritmo A,

$$A = \begin{bmatrix} 0000 \\ 1110 \\ 0111 \\ 1001 \end{bmatrix} \to \begin{bmatrix} 1110 \\ 0111 \\ 1001 \\ 0000 \end{bmatrix} \to \begin{bmatrix} 1110 \\ 0111 \\ 0111 \\ 0000 \end{bmatrix} \to \begin{bmatrix} 1110 \\ 0111 \\ 0000 \\ 0000 \end{bmatrix},$$

de modo que $G = \begin{bmatrix} 1110 \\ 0111 \end{bmatrix}$ es una matriz generadora de C. Por el Algoritmo A, como la FER de A es $\begin{bmatrix} 1001 \\ 0111 \\ 0000 \\ 0000 \end{bmatrix}$, entonces $G_1 = \begin{bmatrix} 1001 \\ 0111 \end{bmatrix}$ es también una matriz generadora para C.

Ejercicios.

31. Determine si cada una de las siguientes es una matriz generadora para algún código lineal:

$$A = \begin{bmatrix} 010011101 \\ 100101101 \\ 101100110 \\ 101101101 \end{bmatrix} \quad B = \begin{bmatrix} 1001101001 \\ 1101000101 \\ 0111001011 \\ 1000010111 \\ 1010001110 \end{bmatrix}.$$

32. Halle una matriz generada en FER para cada uno de los siguientes códigos:

    a) $C = \{000, 001, 010, 011\}$
    b) $C = \{0000, 1001, 0110, 1111\}$
    c) $C = \{00000, 11111\}$
    d) $C = \{00000, 11100, 00111, 11011\}$
    e) $C = \{00000, 11110, 01111, 10000\}$
    f) $C = \{000000, 101010, 010101, 111111\}$

33. Halle una matriz generadora en FER para cada uno de los siguientes códigos:

    a) $C = \{000000, 001011, 010101, 011110, 100110, 101101, 110011, 111000\}$.
    b) $C = \{00000000, 01101111, 11011000, 11111101, 10010010, 00100101, 01001010, 10110111\}$.
    c) $C = \{0000000000, 1111100000, 0000011111, 1111111111\}$.

34. Halle la matriz generadora para el código lineal generado por cada uno de los siguientes conjuntos:

    a) $S = \{11111111, 11110000, 11001100, 10101010\}$.
    b) $S = \{11111100, 11110011, 11001111, 00111111\}$.
    c) $S = \{100100100, 010010010, 001001001, 111111111\}$.
    d) $S = \{10101, 01010, 11111, 00011, 10110\}$.
    e) $S = \{1010, 0101, 1111\}$.
    f) $S = \{101101, 011010, 110111, 000111, 110000\}$.
    g) $S = \{1001011, 0101010, 1001100, 0011001, 0000111\}$.

Sea C un código lineal de largo n y dimensión k. Si G es una matriz generadora para C y si u es una palabra de largo k escrita como fila, entonces $v = uG$ es una palabra en C, pues v es una combinación lineal de las filas de G, y estas forman una base de C. De hecho, si $u = (a_1, a_2, \ldots, a_k)$ y si

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix}.$$

donde $g_1, g_2, \ldots, g_k$ son las filas de G, entonces $v = a_1 g_1 + a_2 g_2 + \cdots + a_k g_k$. Por otra parte, como cada palabra v en C es una combinación lineal de las palabras de la base (formada por las filas de G), entonces $v = uG$, para alguna palabra $u \in K^k$. Más aún, si $u_1 G = u_2 G$, entonces $u_1 = u_2$, pues cada palabra de C es una *única* combinación lineal de las palabras de la base. Luego, ninguna palabra $v = uG$ es producida por más de una palabra $u \in K^k$.

Teorema 9. *Si G es una matriz generadora de un código lineal C de largo n y dimensión k, entonces $v = uG$ recorre todas las 2^k palabras de C a medida que u recorre todas las 2^k palabras de largo k. Luego, C es el conjunto de todas las palabras uG con $u \in K^k$. Más aún, $u_1 G = u_2 G$ si y sólo si $u_1 = u_2$.*

Asigne mensajes a las palabra en K^4 como sigue:

0000	1000	0100	0010	1100	1010	1010	1001
A	B	C	D	E	F	G	H
0110	0101	0011	1110	1101	1011	0111	1111
I	J	K	L	M	N	O	P

a) Codifique el mensaje "HÁGALO FÁCIL", ignorando los espacios y los acentos.

b) Transmita el mismo mensaje suponiendo que durante la transmisión la primera palabra recibe un error en la primera posición, la segunda palabra recibe errores en las quinta y sexta posiciones, la tercera palabra en la séptima posición, y las demás palabras no reciben errores.

38. Halle el número de mensajes que pueden ser enviados, y la razón de información r, para cada uno de los códigos lineales en los Ejercicios 33 y 34.

2.6. Matrices de control de paridad

Desarrollamos otra matriz asociada con un código lineal que está intimamente conectada con la matriz generadora. Esta nueva matriz será de gran valor para describir esquemas de decodificación.

Se dice que una matriz H es una *matriz de control de paridad* de un código lineal C si las columnas de H forman una base del código dual $C^{\perp}$. Si C tiene largo n y dimensión k, entonces ya que por el Teorema 6 la suma de las dimensiones de C y $C^{\perp}$ es n, tenemos que cualquier matriz de control de paridad para C debe poseer n filas, $n-k$ columnas y rango $n-k$. Comparar el siguiente teorema con el Teorema 7.

Teorema 10. *Una matriz H es matriz de control de paridad de un código lineal C si y solo si las columnas de H son linealmente independientes.*

El próximo teorema describe un código lineal en términos de su matriz de control de paridad.

Teorema 11. *Si H es una matriz de control de paridad de un código lineal C de largo n, entonces C consiste precisamente de todas las palabras v en K^n tales que $vH = 0$.*

Si se nos da una matriz generadora de un código lineal C, entonces podemos hallar una matriz de control de paridad para C usando el Algoritmo C. Tal matriz de control de paridad lo es la matriz H construida en aquel algoritmo, pues las columnas de H forman una base de $C^{\perp}$.

Ejemplo. Hallemos la matriz de control de paridad para el código $C = \{0000, 1110, 0111, 1001\}$ del primer ejemplo de la Sección 16. Allí hallamos que

$$G_1 = \begin{bmatrix} 10 & 01 \\ 01 & 11 \end{bmatrix} = [I_2, X]$$

es una matriz generadora para C, la cual está en FER. Por el Algoritmo C, construimos H:

$$H = \begin{bmatrix} X \\ I_2 \end{bmatrix} = \begin{bmatrix} 01 \\ 11 \\ 10 \\ 01 \end{bmatrix}$$

Asigne mensajes a las palabra en K^4 como sigue:

0000	1000	0100	0010	1100	1010	1010	1001
A	B	C	D	E	F	G	H
0110	0101	0011	1110	1101	1011	0111	1111
I	J	K	L	M	N	O	P

a) Codifique el mensaje "HÁGALO FÁCIL", ignorando los espacios y los acentos.

b) Transmita el mismo mensaje suponiendo que durante la transmisión la primera palabra recibe un error en la primera posición, la segunda palabra recibe errores en las quinta y sexta posiciones, la tercera palabra en la séptima posición, y las demás palabras no reciben errores.

38. Halle el número de mensajes que pueden ser enviados, y la razón de información r, para cada uno de los códigos lineales en los Ejercicios 33 y 34.

2.6. Matrices de control de paridad

Desarrollamos otra matriz asociada con un código lineal que está intimamente conectada con la matriz generadora. Esta nueva matriz será de gran valor para describir esquemas de decodificación.

Se dice que una matriz H es una *matriz de control de paridad* de un código lineal C si las columnas de H forman una base del código dual $C^{\perp}$. Si C tiene largo n y dimensión k, entonces ya que por el Teorema 6 la suma de las dimensiones de C y $C^{\perp}$ es n, tenemos que cualquier matriz de control de paridad para C debe poseer n filas, $n-k$ columnas y rango $n-k$. Comparar el siguiente teorema con el Teorema 7.

Teorema 10. *Una matriz H es matriz de control de paridad de un código lineal C si y solo si las columnas de H son linealmente independientes.*

El próximo teorema describe un código lineal en términos de su matriz de control de paridad.

Teorema 11. *Si H es una matriz de control de paridad de un código lineal C de largo n, entonces C consiste precisamente de todas las palabras v en K^n tales que $vH = 0$.*

Si se nos da una matriz generadora de un código lineal C, entonces podemos hallar una matriz de control de paridad para C usando el Algoritmo C. Tal matriz de control de paridad lo es la matriz H construida en aquel algoritmo, pues las columnas de H forman una base de $C^{\perp}$.

Ejemplo. Hallemos la matriz de control de paridad para el código $C = \{0000, 1110, 0111, 1001\}$ del primer ejemplo de la Sección 16. Allí hallamos que

$$G_1 = \begin{bmatrix} 10 & 01 \\ 01 & 11 \end{bmatrix} = [I_2, X]$$

es una matriz generadora para C, la cual está en FER. Por el Algoritmo C, construimos H:

$$H = \begin{bmatrix} X \\ I_2 \end{bmatrix} = \begin{bmatrix} 01 \\ 11 \\ 10 \\ 01 \end{bmatrix}$$

que es una matriz de control de paridad para C. Nótese que $vH = 00$, para toda palabra $v \in C$, lo cual era de esperar por el Teorema 11.

Ejercicios.

39. Hallar una matriz de control de paridad para cada uno de los códigos C del Ejercicio 32.

40. Hallar una matriz de control de paridad para cada uno de los códigos C construidos en los Ejercicios 33 y 34.

Caracterizamos ahora la relación entre las matrices generadora y de control de paridad de un código lineal, y la relación entre estas matrices para un código lineal y su código dual.

Teorema 12. *Las matrices G y H son la generadora y la de control de paridad, respectivamente, de un código lineal C, si y sólo si:*

1. *las filas de G son linealmente independientes*;
2. *las columnas de H son linealmente independientes*;
3. *el número de filas de G más el número de columnas de H iguala el número de columnas de G, que iguala a su vez, al número de filas de H*;
4. $GH = 0$.

Teorema 13. *H es una matriz de control de paridad de C si y solo si H^T es una matriz generadora para $C^\perp$.*

El Teorema 13 sigue del teorema 12 y del hecho de que

$$H^T G^T = (GH)^T = 0.$$

Dada una de estas cuatro matrices, la generadora o la de control de paridad de C ó $C^\perp$, el Algoritmo C y el Teorema 13 pueden ser usados para formar las otras tres matrices. El siguiente diagrama indica cómo se procede:

$$\begin{array}{ccccc} & & \text{Algoritmo C} & & \\ & H_{C^\perp} & \longleftarrow & G_{C^\perp} & \\ \text{Transpuesta} & \downarrow & & \uparrow & \text{Transpuesta} \\ & G_C & \longrightarrow & H_C & \\ & & \text{Algoritmo C} & & \end{array}$$

Ejemplo. Sea C un código lineal con matriz de control de paridad

$$H = \begin{bmatrix} 11 \\ 11 \\ 01 \\ 10 \\ 01 \end{bmatrix} = \begin{bmatrix} X \\ I_2 \end{bmatrix}.$$

1. Entonces la matriz generadora para $C^\perp$ es

$$H^T = \begin{bmatrix} 11010 \\ 11101 \end{bmatrix}.$$

2. La FER de H^T es $\begin{bmatrix}11010\\00111\end{bmatrix}$, de modo que por el Algoritmo C, una matriz de control de paridad para $C^\perp$ es
$$\begin{bmatrix}110\\100\\011\\010\\001\end{bmatrix}.$$

3. A partir de la forma de H, tenemos que
$$G = \begin{bmatrix}100 & 11\\010 & 11\\001 & 01\end{bmatrix} = [I_3, X]$$
es una matriz generadora para C. Esto se ve usando el Algoritmo C al revés. Luego, por el Teorema 13, G^T es también una matriz de control de paridad para $C^\perp$:
$$\begin{bmatrix}100\\010\\001\\110\\111\end{bmatrix}.$$

Ejercicios.

41. En cada parte de este ejercicio, damos una matriz de control de paridad H de un código lineal C. Halle (i) una matriz generadora para $C^\perp$; (ii) una matriz generadora para C:
$$\text{(a) } H = \begin{bmatrix}100\\100\\010\\001\\010\\001\end{bmatrix} \qquad \text{(b) } H = \begin{bmatrix}01\\10\\01\\10\\01\end{bmatrix} \qquad \text{(c) } H = \begin{bmatrix}111\\110\\101\\011\\100\\010\\001\end{bmatrix}$$

42. Liste todas las palabras en el código dual $C^\perp$, para el código $C = \{00000, 11111\}$. Luego halle matrices generadora y de control de paridad para $C^\perp$.

43. Para cada código C descrito abajo, halle la dimensión de C, la dimensión de $C^\perp$, el tamaño de las matrices generadora y de control de paridad para C y $C^\perp$, el número de palabras en C y en $C^\perp$, y las razones de información de C y $C^\perp$:

    a) C tiene largo $n = 2^r - 1$ palabras y dimensión t.

    b) C tiene largo $n = 23$ y dimensión 11.

    c) C tiene largo $n = 15$ y dimensión 8.

2.7. Códigos equivalentes

Cualquier matriz $k \times n$, G, tal que $k < n$ y cuyas primeras k columnas forman la matriz $k \times k$ identidad I_k, o sea. que es de la forma $G = [I_k, X]$, tiene automáticamente sus filas linealmente independientes y está en FER. Luego, G es una matriz generadora para algún código lineal de largo n y dimensión k. Se dice que una tal matriz generadora está en *forma estándar* y que el código C es un *código sistemático*.

No todos los códigos lineales tienen una matriz generadora en forma estándar. Por ejemplo, el código definido por la matriz generadora G en el ejercicio abajo tiene otras cinco matrices generadoras, ninguna de las cuales está en forma estándar, y tampoco lo está G.

Ejercicios.

44. Halle las otras cinco matrices generadoras para el código generado por

$$G = \begin{bmatrix} 100 \\ 001 \end{bmatrix}.$$

Sin embargo, es deseable usar códigos que tengan matrices generadoras que se hallen en forma estándar. Una razón para ello es que si un código lineal C tiene matriz generadora G en forma estándar $G = [I, X]$, entonces el Algoritmo C produce

$$H = \begin{bmatrix} X \\ I \end{bmatrix}$$

que es una matriz de control de paridad para C.

Por el Teorema 9, cada palabra v en un código lineal C de largo n y dimensión k puede escribirse como uG, donde u es una palabra única en K^k (que solo depende de v), y G es una matriz generadora de C. Consideramos la palabra u, de largo k, como el mensaje a ser enviado. Pero en lugar de transmitir solo u, transmitimos la palabra uG. Si la DMV nos lleva a concluir correctamente que $v = uG$ fue enviada, entonces el destinatario de la transmisión debe recobrar (o recuperar) de alguna forma el mensaje original u a partir de uG. Si G está en forma estándar, entonces recobrar u es trivial. Pues en este caso

$$v = uG = u[I_k, X] = [uI_k, uX] = [u, uX].$$

De modo que obtenemos el siguiente teorema, que indica la importante ventaja de poseer una matriz generadora en forma estándar.

Teorema 14. *Si C es un código lineal de largo n y dimensión k con matriz generadora G en forma estándar, entonces los primeros k dígitos en una palabra-código $v = uG$ forman la palabra u en K^k.*

Ejemplo. Si

$$G = \begin{bmatrix} 1000|101 \\ 0100|100 \\ 0010|110 \\ 0001|011 \end{bmatrix} = [I_4, X]$$

y si el mensaje es $u = 0111$, entonces $uG = 0111001 = [u001]$. Y si $u = 1011$, entonces $uG = 1011000$.

Ejercicios.

45. Sea C la matriz generadora del ejemplo anterior. Codifique cada uno de los mensajes siguientes y observe que los primeros 4 dígitos en la palabra-código resultante forman el mensaje u:

    (a) $u = 1111$ (b) $u = 1011$ (c) $u = 0000$.

46. Describa y justifique un método para recobrar u a partir de uG, si G no está en forma estándar.

47. Si un código lineal C tiene matriz generadora

$$G = \begin{bmatrix} 1100101 \\ 0110101 \\ 1011011 \\ 1100110 \\ 0110000 \end{bmatrix},$$

    recupere u a partir de $v = uG = 0000101$.

Bajo las hipótesis del Teorema 14, los primeros k dígitos de la palabra-código $v = uG$ son denominados *dígitos de información*, Pues ellos, de hecho, contienen el mensaje u, mientras que los últimos $n - k$ dígitos de $v = uG$ son llamados *dígitos de redundancia*.

Con todas las ventajas de disponer de un código lineal con matriz generadora en forma estándar, ¿qué se puede hacer si nos atascamos con un código C que no tiene matriz generadora en forma estándar? Consideremos el código C con matriz generadora G del Ejercicio 44. Como fue indicado en el ejercicio, C no tiene matriz generadora en forma estándar. Supongamos, en este ejemplo, que decidimos rearreglar los dígitos en el orden "primero, tercero, segundo", en lugar de "primero, segundo, tercero". Las cuatro palabras en C han sido entonces transformadas en un nuevo código C'. Comparemos:

$$\begin{array}{rcl} C & = & \{000, \quad 100, \quad 001, \quad 101\} \\ C' & = & \{000, \quad 100, \quad 010, \quad 110\}. \end{array}$$

Notar que C', aunque diferente de C, comparte muchas propiedades con C. Por ejemplo, ambos C y C' son lineales, ambos tiene largo 3, dimensión 2 y distancia 3. Pero C' tiene una ventaja sobre C, y es que C' tiene una matriz generadora en forma estándar. Observar que G' es obtenida de G permutando sus segunda y tercera columnas, justo como C' es obtenido de C al intercambiar consistentemente los segundos con los terceros dígitos:

$$G = \begin{bmatrix} 100 \\ 001 \end{bmatrix}, \quad G' = \begin{bmatrix} 100 \\ 010 \end{bmatrix}.$$

Si C es un código-bloque de largo n podemos siempre obtener un nuevo código-bloque C' de largo n eligiendo una permutación σ particular de los n dígitos y, luego. consistentemente reordenando los dígitos de cada palabra de C por medio de σ. Decimos que el código resultante es *equivalente* a C.

Ejemplo. Si $n = 5$ y elegimos un rearreglo de los dígitos en el orden 2, 1, 4, 5, 3, o sea con $\sigma = \begin{pmatrix} 12345 \\ 21453 \end{pmatrix}$, entonces el código

$$C = \{11111, 01111, 00111, 00011, 00001\}$$

es equivalente al código

$$C' = \{11111, 10111, 00111, 00110, 00010\}.$$

(Nótese que C y C' no son lineales).

Teorema 15. *Todo código lineal C es equivalente a un código lineal C' con matriz generadora en forma estándar.*

Demostración. Si G es una matriz generadora de C, colocamos G en FER. Reordenamos las columnas de la FER de modo que las columnas líderes vengan en primer lugar y formen una matriz identidad. El resultado es una matriz G' en forma estándar, que es una matriz generadora de un código C' equivalente a C.

Ejemplo. La matriz

$$G = \begin{bmatrix} 011000010 \\ 000100110 \\ 000010010 \\ 000001100 \\ 000000001 \end{bmatrix}$$

es una matriz generadora en FER con las columnas 2, 4, 5, 6 y 9 como columnas líderes. Reordenando las columnas en el orden 2, 4, 5, 6, 9, 1, 3, 7, 8 produce

$$G' = \begin{bmatrix} 10000 & 0101 \\ 01000 & 0010 \\ 00100 & 0001 \\ 00010 & 0010 \\ 00001 & 0000 \end{bmatrix} = [I_5, X],$$

que es una matriz generadora en forma estándar para un código equivalente al código generado por G.

Ejercicios.

48. Halle el código sistemático C' equivalente al código C dado. Verifique que C y C' tengan el mismo largo, dimensión y distancia:

    a) $C = \{00000, 10110, 10101, 00011\}$.

    b) $C = \{00000, 11100, 00111, 11011\}$.

49. Halle la matriz generadora G en forma estándar equivalente al código con matriz generadora G dada por:

    (a) $G = \begin{bmatrix} 101010 \\ 011000 \\ 110100 \\ 101011 \end{bmatrix}$ (b) $G = \begin{bmatrix} 111000000 \\ 000111000 \\ 000111111 \end{bmatrix}$

50. Halle la matriz generadora G' en forma estándar para el código C con matriz de control de paridad H dada por:

(a) $H = \begin{bmatrix} 110 \\ 100 \\ 011 \\ 010 \\ 001 \end{bmatrix}$. (b) $H = \begin{bmatrix} 100 \\ 111 \\ 010 \\ 110 \\ 101 \\ 001 \\ 011 \end{bmatrix}$.

51. Pruebe qué códigos lineales equivalentes siempre tienen el mismo largo, dimensión y distancia.

52. Determine si cada uno de los siguientes pares de matrices G_1 y G_2 generan códigos equivalentes:

a)

$$G_1 = \begin{bmatrix} 1100 \\ 0110 \\ 0011 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1001 \\ 0101 \\ 0011 \end{bmatrix}$$

b)

$$G_1 = \begin{bmatrix} 110000 \\ 001100 \\ 000011 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 111111 \\ 011011 \\ 001001 \end{bmatrix}$$

c)

$$G_1 = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1011000 \\ 0101100 \\ 0010110 \\ 0001011 \end{bmatrix}.$$

2.8. Distancia de un código lineal

Vimos que la distancia de un código lineal es el menor peso de cualquier palabra-código. La distancia de un código lineal puede también ser determinada a partir de la matriz de control de paridad del código.

Teorema 16. *Sea H una matriz de control de paridad de un código lineal C. Entonces C tiene distancia d si y solo si cualquier conjunto de $d-1$ filas de H es linealmente independiente y al menos un conjunto de d filas de H es linealmente dependiente.*

La idea de la prueba del teorema es que si v es una palabra de largo n, entonces vH es una combinación lineal de exactamente $\mathrm{w}(v)$ filas de H. Así, si $v \in C$ y $\mathrm{w}(v) = d$, como $vH = 0$, tenemos que ciertas d filas de H son linealmente dependientes. Si $vH = 0$, entonces v es una palabra-código, de modo que $\mathrm{w}(v) \geq d$.

Ejemplo. Sea C el código lineal con matriz de control de paridad

$$H = \begin{bmatrix} 110 \\ 011 \\ 100 \\ 010 \\ 111 \end{bmatrix}.$$

Se ve, por inspección directa, que no hay dos filas de H que sumen 000, de modo que cualquier par de filas de H son linealmente independientes, pero las filas 1, 3 y 4, por ejemplo, suman 000, y de aquí, son linealmente dependientes. Por lo tanto, $d-1=2$. Luego, la distancia de C es $d=3$.

Ejercicios.

53. Halle el código C en el ejemplo anterior. Compute el peso de cada palabra-código y verifique que C tiene distancia 3.

54. Halle la distancia del código lineal C con cada una de las matrices de control de paridad dadas. Use el Teorema 16 y luego chequee sus respuestas para hallar $\mathrm{w}(v)$, para cada $v \in C$:

(a) $H = \begin{bmatrix} 0111 \\ 1110 \\ 1000 \\ 0100 \\ 0001 \\ 0001 \end{bmatrix}$ (b) $H = \begin{bmatrix} 1110 \\ 1101 \\ 1011 \\ 0111 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}$ (c) $H = \begin{bmatrix} 1101 \\ 1011 \\ 1110 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}$

55. Halle, usando el Teorema 16, la distancia del código lineal con la siguiente matriz generadora:

(a) $G = \begin{bmatrix} 111000000 \\ 000111000 \\ 111111111 \end{bmatrix}$ (b) $G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}$

2.9. Clases módulo de un código lineal

En esta sección, consideraremos un tópico que será útil al decodificar un código lineal, y al que volveremos en la próxima sección.

Si C es un código lineal de largo n, y si u es una palabra de largo n, definimos la *clase módulo C determinada por u* como el conjunto de todas las palabras de la forma $v+u$, a medida que v recorre las palabras de C. Denotamos esta clase como $C+u$. Luego,

$$C+u = \{v+u | v \in C\}.$$

Ejemplo. Sea $C=\{000, 111\}$, y sea $u=101$. Luego,

$$C+101 = \{000+101, 111+101\} = \{101, 010\}.$$

Nótese también que

$$C+111 = \{000+111, 111+111\} = \{111, 000\} = C$$

y que

$$C+010 = \{000+010, 111+010\} = \{010, 101\} = C+101.$$

Ejercicios.

56. Liste el resto de las clases de $C = \{000, 111\}$. Note que hay ocho posibilidades para las clases de C, una para cada palabra de K^3, pero solo cuatro clases son distintas.

Si C es un código lineal de largo n, entonces podríamos hallar que existen hasta 2^n clases diferentes $C + u$ de C, una por cada una de las diferentes 2^n palabras u de largo n. Como el ejemplo anterior evidencia, esto casi nunca acontece. Es muy posible que $C + u_1$ sea idéntica a $C + u_2$, con $u_1 \neq u_2$.

El siguiente teorema contiene algunos hechos importantes y útiles respecto de las clases módulo C. Un estudio cuidadoso de los ejemplos que siguen al teorema debe ayudar a comprender estos hechos. Las pruebas son técnicas de teoría de conjuntos y, por lo tanto, han sido relegadas a ejercicios.

Teorema 17. *Sea C un código lineal de largo n. Sean u y v palabras de largo n.*

1. *Si u está en la clase $C+v$, entonces $C+u = C+v$; (cada palabra en una clase determina esa clase).*
2. *La palabra u está en la clase $C + u$.*
3. *Si $u + v \in C$, entonces u y v están en la misma clase.*
4. *Si $u + v \notin C$, entonces u y v están en clase diferentes.*
5. *Cada palabra en K^n está contenida en una y solo una clase módulo C; (o bien $C + u = C + v$ o bien $C + u$ y $C + v$ no tienen palabras en común).*
6. *$|C+u| = |C|$; (el número de palabras en una clase módulo C iguala el número de palabras en el código C).*
7. *Si C tiene dimensión k, entonces hay exactamente 2^{n-k} clases módulo C diferentes, y cada clase contiene exactamente 2^k palabras.*
8. *El mismo código C es una de las clases módulo C.*

Ejemplo. Listamos las clases del código

$$C = \{0000, 1011, 0101, 1110\}.$$

Primero que todo, C es una clase en si mismo por la propiedad (8) del teorema. Cada palabra en C determina la clase C, por (1) y (5), de modo que tomamos una palabra $u \in K^4 \setminus C$. Para uso posterior en decodificación, ayudará que tomemos u con el menor peso posible. Así, tomamos $u = 1000$. Luego obtenemos la clase

$$C + 1000 = \{1000, 0011, 1101, 0110\}$$

al sumar 1000 a cada palabra en C. Nótese que $u = 1000$ está en la clase $C + u = C + 1000$. Ahora tomemos otra palabra de peso mínimo en K^4 pero no en C, ni en $C + 1000$, digamos 0100. Formamos otra clase

$$C + 0100 = \{0100, 1111, 0001, 1010\}.$$

Repitiendo el proceso con 0010, obtenemos la clase

$$C + 0010 = \{0010, 1001, 0111, 1100\}.$$

El código C tiene dimensión $k = 2$. Hemos listado $2^{n-k} = 2^{4-2} = 2^2 = 4$ clases, cada una con $2^k = 4$ palabras, y toda palabra en $\mathbb{K}^4$ se contabiliza al aparecer exactamente en una clase. También observamos que $0001 + 1010 = 1011$ está en C; luego 0001 y 1010 están en la misma clase, de hecho $C + 0100$ (ver propiedad (3) en el teorema). Por otra parte, $0100 + 0010 = 0110$ no está en C, y 0100 y 0010 están en clases diferentes (ver propiedad (4) del teorema).

Ejemplo. Listamos las clases módulo el código lineal C con matriz generadora $G = \begin{bmatrix} 100110 \\ 010011 \\ 001111 \end{bmatrix}$, evitando ahora paréntesis y comas, por simplicidad:

000000	100000	010000	001000	000100	000010	000001	000101
100110	000110	110110	101110	100010	100100	110111	100011
010011	110011	000011	011011	010111	010001	010011	010110
001111	101111	011111	000111	001011	001101	001110	001010
110101	010101	100101	111101	110001	110111	110100	110000
101001	001001	111001	100001	101101	101011	101000	101101
011100	111100	001100	010100	011000	011110	011101	011001
111010	011010	101010	110010	111110	111000	101110	110110

Ejercicios.

57. Liste todas las clases de cada uno de los siguientes códigos lineales:

    a) $C = \{0000, 1001, 0101, 1100\}$

    b) $C = \{0000, 1010, 1101, 0111\}$

    c) $C = \{00000, 10100, 01011, 11111\}$

    d) $C = \{0000\}$

58. Liste todas las clases de cada uno de los códigos lineales generados por las siguientes matrices:

    (a) $G = \begin{bmatrix} 111000 \\ 001110 \\ 100011 \end{bmatrix}$ (b) $G = \begin{bmatrix} 101010 \\ 010101 \end{bmatrix}$

    (c) $G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}$ (d) $G = \begin{bmatrix} 10001 \\ 01001 \\ 00101 \\ 00011 \end{bmatrix}$

    (e) $G = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}$ (f) $G = [1111]$

59. Liste todas las clases del código que tiene la matriz de control de paridad dada:

(a) $H = \begin{bmatrix} 10 \\ 11 \\ 10 \\ 01 \end{bmatrix}$ (b) $H = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix}$ (c) $H = \begin{bmatrix} 100 \\ 010 \\ 001 \\ 001 \\ 001 \\ 001 \end{bmatrix}$

60. Pruebe el Teorema 17.

2.10. DMV para códigos lineales

Una de nuestras metas es diseñar códigos que permitan una decodificación fácil y rápida. De hecho, los códigos lineales ofrecen un método más eficiente de implementar DMV que simplemente usar una tabla de DMVI. Describiremos un procedimiento de DMVC o DMVI para un código lineal. La matriz de control de paridad y las clases de tal un código juegan roles fundamentales en este proceso de decodificación.

Sea C un código lineal. Supongamos que una palabra de C es transmitida y que se recibe la palabra w, resultando en un patrón de error $u = v + w$. Luego, $w + u = v$ está en C, de modo que el patrón de error u y la palabra recibida w están en la misma clase módulo C, por la propiedad (3) del Teorema 17.

Como los patrones de error de pequeño peso son los más verosímiles, veamos como la DMV opera para un código lineal C. Al recibir la palabra w seleccionamos una palabra u de menor peso en la clase $C+w$ (que debe contener w) y concluimos que $v = w+u$ fue la palabra enviada.

Ejemplo. Sea $C = \{0000, 1011, 0101, 1110\}$. Las clases módulo C del penúltimo ejemplo fueron

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

Supongamos que $w = 1101$ sea recibida. La clase $C + w = C + 1101$, que contiene a w, es la segunda en la lista (en la segunda fila). La palabra de menor peso en esta clase es $u = 1000$, que seleccionamos como patrón de error. Concluimos que $v = w + u = 1101 + 1000$ fue la palabra enviada más verosímil. Ahora supongamos que $w = 1111$ sea recibida. En la clase $C + w$, que contiene 1111, hay dos palabras de menor peso: 0100 y 0001. Con DMVC podemos elegir arbitrariamente una de las dos como patrón de error, digamos $u = 0100$, y ahí concluimos que $v = w + u = 1111 + 0100 = 1011$ fué la palabra enviada más verosímil.

Ejercicios.

61. Sea C el código del último ejemplo de la Sección 20. Use el procedimiento para DMVC justamente ejemplificado para decodificar cada una de las siguientes palabras recibidas:

    (a) 000011 (b) 001001 (c) 001101

    (d) 010110 (e) 110101 (f) 001010

Las tareas más difíciles en el procedimiento comentado son buscar la clase que contiene la palabra recibida w y luego hallar la palabra de menor peso en tal clase. Podemos usar una matriz de control de paridad para desarrollar un procedimiento que facilite estas tareas.

Sea C un código lineal de largo n y dimensión k. Sea H una matriz de control de paridad para C. Para cualquier palabra en $\mathbb{K}^n$, el *síndrome* de w es la palabra wH en $\mathbb{K}^{n-k}$.

Ejemplo. La matriz H de tipo 4×2 abajo presente es una matriz de control de paridad para el código C del ejemplo anterior. Si $w = 1101$, entonces el síndrome de w es

$$wH = 1101 \begin{bmatrix} 11 \\ 01 \\ 10 \\ 01 \end{bmatrix} = 11.$$

Nótese que la palabra de menor peso en la clase $C + w$ es $u = 1000$, (ver último ejemplo), y que el síndrome de u es

$$uH = 1000 \begin{bmatrix} 11 \\ 01 \\ 10 \\ 01 \end{bmatrix} = 11 = wH.$$

Más aún, si $w = 1101$ es recibida, DMVC concluye que $v = w + u = 1101 + 1000 = 0101$ fue enviada, de modo que hubo un error en el primer dígito. Nótese también que para el patrón de error u, el síndrome uH toma la primera fila de H, correspondiente a la localización del error más verosímil.

El siguiente teorema contiene algunos hechos básicos y útiles respecto del síndrome. Las pruebas pueden se hechas usando las definiciones de los conceptos envueltos y las propiedades de clases en el Teorema 17.

Teorema 18. *Sea C un código lineal de largo n. Sea H una matriz de control de paridad de C. Sean w y u palabras de K^n.*

1. *$wH = 0$ si y solo si w es una palabra de C.*
2. *$wH = uH$ si y solo si w y u yacen en la misma clase módulo C.*
3. *Si u es un patrón de error de una palabra recibida w, entonces uH es la suma de las filas de H que corresponden a las posiciones en las cuales ocurrieron los errores durante la transmisión.*

Nótese que si ningún error ocurre en la transmisión y w es recibida, entonces $wH = 0$. Pero $wH = 0$ no implica que ningún error ocurra, pues la palabra-código w no precisa ser la palabra-código que fue enviada.

Como las palabras en una misma clase tienen el mismo síndrome, mientras que palabras en diferentes clases tiene diferentes síndromes, podemos identificar una clase con su síndrome. El síndrome de una clase es el síndrome de cualquier palabra de esa clase. Luego, si el código tiene largo n y dimensión k, entonces cada una de las 2^{n-k} palabras de largo $n-k$ ocurre como síndrome de exactamente una de las 2^{n-k} clases.

Ejemplo. El código C del primer ejemplo de la sección tiene largo $n = 4$ y dimensión $k = 2$.

Las clases módulo C (listadas en ese ejemplo) contienen todas las $2^n = 2^4 = 16$ palabras de largo $n = 4$. Hay $2^{n-k} = 2^{4-2} - 2^2 = 4$ palabras de largo $n - k = 2$. Cada una de ellas es el síndrome de exactamente una de las $2^{n-k} = 4$ clases módulo C.

Para computar el síndrome de una clase particular, podemos elegir una palabra w en esa clase. Entonces wH será el síndrome de la clase. Para DMV, queremos una palabra de menor peso en la clase para usarla como su patrón de error. En los ejemplos de la sección pasada habíamos ordenado cuidadosamente las clases de modo que una palabra de menor peso quedaba arriba, o sea en primer lugar. Cualquier palabra de menor peso en una clase es llamada *líder de clase*. Si hubiese más de un candidato para ese líder de clase, elegimos uno arbitrariamente cuando se hace DMVC.

Ejemplo. Nuevamente, sea C el código citado en el ejemplo anterior. Para cada clase computamos el síndrome usando el líder de clase y desplegamos los resultados como en la siguiente tabla:

Líder u de clase	Síndrome uH
0000	00
1000	11
0100	01
0010	10

Nótese nuevamente que cada palabra de largo 2 ocurre una vez y solo una como un síndrome.

La tabla de este ejemplo, que empareja cada síndrome con su líder de clase, es llamada *arreglo estándar de decodificación*, o AED. Para construir un AED, primero listamos todas las clases del código, y seleccionamos de cada clase una palabra de peso mínimo como líder de clase u. Luego hallamos la matriz de control de paridad para el código y, para cada líder de clase u, computamos el síndrome uH. Una forma más rápida de construir un AED, dada la matriz de control de paridad H y la distancia d para el código C, sería generar todos los patrones de error e con $\mathrm{w}(e) \leq \lfloor (d-1)/2 \rfloor$ y computar el síndrome $s = eH$ para cada uno de ellos.

Ejemplo. Construyamos un AED para el código C del último ejemplo de la sección 20 (donde las clases módulo C habían sido listadas). Para cada una de las primeras siete clases no tuvimos candidatos para líderes de clase. De hecho, la palabra superior es la única palabra de peso mínimo en cada clase. Pero en la última clase, el menor peso de una palabra es 2, y esa clase contiene tres palabras de peso 2: 000101, 001010 y 110000. Usando DMVC, podríamos seleccionar arbitrariamente 000101 como nuestro patrón de error. Usando DMVI, pediríamos retransmisión y colocaríamos un asterisco * en cada entrada del AED para así indicarlo. Podemos construir la siguiente matriz de control de paridad para C:

$$H = \begin{bmatrix} 110 \\ 011 \\ 111 \\ 100 \\ 010 \\ 001 \end{bmatrix}.$$

Luego, podemos obtener el siguiente AED para C suponiendo que se usa DMVC:

Patrón de error	Síndrome de H
000000	000
100000	110
010000	011
001000	111
000100	100
000010	010
000001	001
000101	101

Notar que los síndromes son precisamente todas las palabras de K^3. La clase C siempre tiene la palabra nula como líder de clase y siempre tiene síndrome nulo. El líder de clase elegido para la última clase, $u = 000101$, nos da un síndrome $uH = 101$, que es la suma de las filas 4 y 6 de H, las posiciones con 1 en el patrón de error u. Usando DMVI, esta entrada sería un asterisco *.

Ejercicios.

62. Construya un AED suponiendo DMVI para cada uno de los códigos del Ejercicio 57.

63. Construya un AED suponiendo DMVI para cada uno de los códigos del Ejercicio 58.

64. Construya un AED suponiendo DMVI para cada uno de los códigos del Ejercicio 59.

65. Pruebe el Teorema 18.

Finalmente, podemos hacer alguna decodificación. Una vez producido el AED, es fácil usar la DMV. Cuando recibimos una palabra w, primero computamos el síndrome wH. Luego hallamos el líder de clase u con $wH = uH$ en el AED. Y concluimos que la palabra $v = w + u$ fue la palabra enviada más verosímil.

Ejemplo. Sea C el código del primer ejemplo de la sección. Un AED para el mismo aparece en el penúltimo ejemplo. La matriz de control de paridad H aparece en el tercer ejemplo de la sección. Supongamos que $w = 1101$ sea recibida. Entonces el síndrome es $wH = 11$, llevándonos a la segunda fila del AED, donde el líder de clase es $u = 1000$. Concluimos que $v = w + u = 0101$ fue enviada. Si $w = 1111$ es recibida, entonces $wH = 01 = uH$ para $u = 0100$ del AED. Descodificamos w como $v = w + u = 1011$. Estos resultados son los mismos que en el primer ejemplo de la sección.

Si $w = 1101$ es recibida, decodificamos $v = 0101$ como la palabra enviada. Los cálculos:

$$d(0000, 1101) = 3 \quad d(0101, 1101) = 1$$
$$d(1011, 1101) = 2 \quad d(1110, 1101) = 2$$

dan las distancias entre w y cada una de las palabras en C, y muestran que de hecho $v = 0101$ es la palabra en C más próxima a w.

Sin embargo, si $w = 1111$ es recibida, los mismos cálculos:

$$d(0000, 1111) = 4 \quad d(0101, 1111) = 2$$
$$d(1011, 1111) = 1 \quad d(1110, 1111) = 1$$

revelan un empate para la palabra más próxima a w en C. Esto no nos sorprende, pues hubo una elección de un líder en la clase que contenía a w. Estamos haciendo DMVC, de modo que seleccionamos arbitrariamente un líder de clase, que en efecto determina arbitrariamente una palabra más próxima a w.

Ejemplo. Sea C el último código de la Sección 20. Un AED para el mismo fue construido en el ejemplo anterior al Ejercicio 62. Haremos algo de decodificación usando este AED. Supongamos que recibimos $w = 110111$. Luego $wH = 010$, que nos lleva a la sexta fila del AED. El líder de clase en esa fila es $u = 000010$. Luego la DMVC concluye que $v = w + u = 110111 + 000010 = 110101$ fue la palabra enviada. Ahora supongamos que $w = 110000$ sea recibida. El síndrome $wH = 101$ nos lleva a la última fila del AED, donde el líder de clase es $u = 000101$. Decodificamos w como $v = w+u = 110000+000101 = 110101$. Si hubiéramos elegido la palabra $u' = 001010$ como líder para la última clase, entonces habríamos decodificado w como $w + u' = 110000 + 001010 = 111010$.

Ejercicios.

66. Continuando con el último ejemplo, si $w = 110000$ es recibida, decodifique suponiendo que $u'' = 110000$ fue elegida como líder de clase para la última clase.

67. En el último ejemplo, de nuevo, chequear si en efecto $v = 110101$ es la palabra más próxima en C a w.

68. En el mismo ejemplo con $w = 110000$ recibida, halle todas las palabra más próximas a w.

69. Repita la decodificación del penúltimo ejemplo usando el AED del ejemplo anterior al Ejercicio 62.

70. Para el código del último ejemplo, decodifique las siguientes palabras recibidas w:
(a) 011101 (b) 110101 (c) 111111 (d) 000000

71. Para cada uno de los siguientes códigos, use el AED para decodificar las palabras recibidas. (Los AED para estos códigos fueron construidos en los Ejercicios 62 y 63):

    a) $C = \{0000, 1001, 0101, 1100\}$:
    (i) $w = 1110$, (ii) $w = 1001$, (iii) $w = 0101$.

    b) $C = \{00000, 10100, 01011, 11111\}$:
    (i) $w = 10101$, (ii) $w = 01110$, (iii) $w = 10001$.

72. Sea C el código con matriz de control de paridad

$$H = \begin{bmatrix} 011 \\ 101 \\ 110 \\ 100 \\ 010 \\ 001 \end{bmatrix}.$$

Decodifique (a) 110100, (b) 111111, (c) 101010 y (d) 000110.

73. Sea C el código de largo 7 que tiene como matriz de control de paridad la matriz H cuyas filas son las palabras no nulas de largo 3. Decodifique

    (a) 110100, (b) 111111, (c) 101010, (d) 000110.

2.11. Confiabilidad de DMVI para códigos lineales

Sea C un código lineal de largo n y dimensión k. Recordemos que $\theta_p(C, v)$ era la probabilidad de que si v era enviada en un CSB de probabilidad p, entonces la DMVI correctamente concluía que v era enviada.

Para cada líder de clase único u y para cada palabra v en C, $v + u$ está más próxima de v que cualquier otra palabra-código. Además, si $w \neq v + u$, donde $v \in C$ y u es un líder de clase, entonces w está al menos más próxima de alguna otra palabra en C de lo que lo está de v. De modo que el conjunto $L(v)$ de palabras más próximas de v que de otra palabra-código es

$$L(v) = \{w | w = u + v, \text{ donde } u \text{ es un líder de clase único}\}.$$

Si $w = v + u$, entonces $\theta_p(v, w)$ sólo depende de $\mathrm{w}(u)$. Por lo tanto, para un código lineal C, tenemos que $\theta_p(C, v)$ no depende de v. Denotamos este valor común de $\theta_p(C, v)$ por medio de $\theta_p(C)$, y entonces tenemos:

$$\theta_p(C) = \sum_{u \in L(0)} p^{n-\mathrm{W}(u)}(1-p)^{\mathrm{W}(u)}.$$

Luego, para hallar la confiabilidad de un código lineal, precisamos preocuparnos solo de los líderes de clase únicos: simplemente, computamos la probabilidad de que cada líder de clase que ocurre como patrón de error, sea único en su clase módulo C, y luego, sumamos las probabilidades resultantes para obtener $\theta_p(C)$.

Ejemplo. Sea C el código del último ejemplo de la Sección 20. Usando DMVI, existe un líder de clase de peso 0 y seis de peso 1. Luego,

$$\theta_p(C) = p^6 + 6p^5(1-p).$$

Ejercicios.

74. Calcule $\theta_p(C)$, para cada uno de los códigos de los Ejercicios 59, 60 y 61.

CAPÍTULO 3
De cómo se acotan y perfeccionan los códigos

Vamos ahora al problema de determinar cuántas palabras puede tener un código lineal de largo n y distancia d. Este problema está lejos de su solución, aunque ha sido establecido para ciertos valores de n y d. Sin embargo, podemos hallar ciertas cotas sobre el tamaño de un código con estos parámetros n y d.

3.1. Algunas cotas para códigos

Recordemos que si t y n son enteros con $0 \leq t \leq n$, entonces el símbolo

$$\binom{n}{t} = \frac{n!}{t!(n-t)!} = \frac{1 \cdot 2 \cdots (n-1) \cdot n}{(1 \cdot 2 \cdots (t-1) \cdot t)(1 \cdot 2 \cdot \cdots (n-t))},$$

leído "n toma t", representa el número de maneras en que una colección no ordenada de t objetos puede retirarse de un conjunto de n objetos. Luego, $\binom{n}{t}$ *representa el número de palabras de largo* n *y peso* t.

Teorema 19. *Si* $0 \leq t \leq n$ *y si* v *es una palabra de largo* n*, entonces el número de palabras de largo* n *y distancia a* v *a lo sumo* t *es precisamente*

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}.$$

Como existen 2^n palabras de largo n, poniendo $t = n$ en el Teorema 1 nos da

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} = 2^n.$$

Ejercicios.

1. Ilustre el Teorema 19 para $v = 10110$ y $t = 3$ listando todas las palabras de $\mathbb{K}^5$ que están a distancia a lo sumo 3 de v, y luego chequee que el Teorema 1 da la respuesta correcta para tales palabras.

Para hallar todas las palabras a una cierta distancia t de una palabra fija v, simplemente sumamos a v todas las palabras de peso t. Existen $\binom{n}{t}$ tales palabras. Si C es un código de largo n y distancia $d = 2t+1$, entonces no existe ninguna palabra w a distancia a lo sumo t de dos palabras v_1 y v_2 de C. De hecho, si $d(w, v_1) \leq t$ y $d(w, v_2) \leq t$ con $v_1 \neq v_2$, entonces

$$d(v_1, v_2) \leq d(v_1, w) + d(w, v_2) \leq 2t < d = 2t + 1,$$

lo cual es imposible, pues C tiene distancia mínima d. Luego, si C tiene largo n y distancia $2t+1$, entonces la lista de palabras de $\mathbb{K}^n$ a distancia a lo sumo t de una palabra v_1 de C no posee palabras-código en común con la lista de palabras-código a distancia a lo sumo t de otra palabra v_2, $(v_2 \neq v_1)$. Esto establece el siguiente resultado.

Teorema 20. (Cota de Hamming) *Si C es un código de largo n y distancia $d = 2t+1$ ó $2t+2$, entonces*

$$|C| \left(\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t} \right) \leq 2^n,$$

o sea

$$|C| \leq \frac{2^n}{\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}}.$$

La cota de Hamming es una cota superior para el número de palabras en un código (lineal o no) de largo n y distanca $d = 2t+1$. Nótese que $t = \lfloor (d-1)/2 \rfloor$ de modo que por el Teorema 3, tal código corregirá todos los patrones de error de peso $\leq t$.

Ejemplo. Calculemos una cota superior para el tamaño $|C|$ ó la dimensión k de un código lineal C de largo $n = 6$ y distancia $d = 3$. Como $d = 3 = 2t + 1$, obtenemos $t = 1$. La cota de Hamming nos da:

$$|C| \leq \frac{2^6}{\binom{6}{0} + \binom{6}{1}} = \frac{64}{7}.$$

Pero C debe ser una potencia de 2, de modo que $|C| \leq 8$, y entonces $k \leq 3$.

Ejercicios.

2. Halle una cota superior para el tamaño $|C|$ ó dimensión k de un código lineal con los valores dados de n y d:

   (a) $n = 8, d = 3$ (b) $n = 7, d = 3$ (c) $n = 10, d = 5$

   (d) $n = 15, d = 3$ (e) $n = 15, d = 5$ (f) $n = 23, d = 7$.

3. Verifique las cotas de Hamming para el código lineal C con la matriz generadora G dada:

$$\text{(a) } G = \begin{bmatrix} 1111110000000000 \\ 0000011111000000 \\ 0000011111111111 \end{bmatrix} \quad \text{(b) } G = \begin{bmatrix} 100111 \\ 010101 \\ 001011 \end{bmatrix} \quad \text{(c) } G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}.$$

De la Sección 17 y Teorema 16, sabemos que una matriz de control de paridad H de un código (n, k, d)-lineal es una matriz $n \times (n-k)$ tal que cada $d-1$ filas de H son independientes. Como esas filas tienen largo $n-k$, no podemos tener más de $n-k$ filas que sean linealmente independientes como vectores. Por lo tanto, $d-1 \leq n-k$, o equivalentemente, $k \leq n-d+1$. Esto establece el siguiente resultado, conocido como *cota de Singleton*.

Teorema 21. (Cota de Singleton) *Para cualquier código (n, k, d)-lineal, vale que $d-1 \leq n-k$.*

En cierto sentido la cota de Singleton es más débil que la cota de Hamming. Por ejemplo, si $n = 15$ y $d = 5$, entonces el Teorema 21 implica que $k \leq 11$, mientras que el Teorema 20 implica que $k \leq 8$. Sin embargo, algunos códigos alcanzan igualdad en la cota de Singleton, de modo ésta es usada para definir una clase importante y útil de códigos llamados códigos máxima distancia separables.

Un código (n, k, d)-lineal es llamado *máxima distancia separable* (o MDS) si $d = n - k + 1$ (ó $k = n - d + 1$). Hay varias caracterizaciones equivalentes para códigos MDS.

Teorema 22. *Para un código (n, k, d)-lineal C, las siguientes afirmaciones son equivalentes:*

1. $d = n - k + 1$;
2. *cada $n - k$ filas de la matriz de control de paridad son linealmente independientes*;
3. *cada k columnas de la matriz generadora son linealmente independientes*;
4. *C es MDS.*

Demostración. El Teorema 21 dice que $d \leq n - k + 1$. Pero $d \geq n - k + 1$ si y solo si cada $n - k$ filas de la matriz de control de paridad son linealmente independientes. Luego (1) y (2) son equivalentes. Para probar (3), notar que si $d = n - k + 1$, entonces ninguna palabra-código puede tener más de $k - 1$ ceros. Sin embargo, k columnas de la matriz $k \times n$ generadora forman un conjunto linealmente dependiente si y solo si ciertas palabras-código no nulas tienen 0 en las posiciones coordenadas indicadas por esas k columnas. La equivalencia de (4) con (1)-(3) es relativamente fácil de establecer y se deja como ejercicio. □

Corolario 23. *El dual de un código MDS $(n, k, n - k + 1)$-lineal es un código MDS $(n, n - k, k + 1)$-lineal.*

Ejercicios.

4. Las columnas 2, 3 y 5 de la matriz generadora

$$G = \begin{bmatrix} 11001 \\ 01110 \\ 00101 \end{bmatrix}$$

   forman un conjunto linealmente dependiente. Hallar una palabra-código que tenga ceros en las posiciones 2, 3 y 5.

5. Muestre que si una matriz generadora $k \times n$ tiene un conjunto linealmente dependiente de k columnas, entonces existe una palabra-código con ceros en esas k posiciones.

Aún nos gustaría construir códigos con parámetros dados n, k y d. Las cotas superiores eliminan algunos valores paramétricos. Por ejemplo, la cota de Hamming dice que un código de largo $n = 15$ y distancia $d = 5$ no puede tener dimensión $k = 10$. Sin embargo, esta cota no elimina la posibilidad para la existencia de un código $(15, 8, 5)$-lineal.

¿Cómo podríamos hacer para hallar un código (15, 8, 5)-lineal? En general, este es un problema difícil. Un enfoque es intentar hallar una matriz de control de paridad H para un tal código. O sea, que poniendo $r = n - k$, deberíamos hallar n vectores de largo r para formar las filas de una matriz H tal que cada conjunto de $d - 1$ de esos vectores sea linealmente independiente.

Ejemplo. Sea $n = 15$, $k = 6$ y $d = 5$. Luego $r = 15 - 6 = 9$. De modo que desearíamos hallar 15 vectores no nulos de largo 9 con la propiedad de que cualesquiera cuatro de estos sean linealmente independientes. Poner las primeras 9 filas es fácil: tomamos la matriz 9×9 identidad I_9. Supongamos hallados tres vectores más hasta un total de 12 filas y, así, pondríamos:

$$H = \begin{bmatrix} I_9 \\ 111100000 \\ 100011100 \\ 101000011 \\ ? \end{bmatrix}.$$

Antes de procurar el próximo vector, observemos que el siguiente argumento de conteo nos dice que existe uno más. Entre los 2^9 vectores no podemos elegir ni el vector nulo ni ninguno de los 12 ya elegidos hasta ahora. Esto elimina 1+12 vectores. También eliminamos cualquier vector que se pueda escribir como la suma de 2 ó 3 de esos vectores, pues crearíamos conjuntos linealmente dependientes de 3 ó 4 vectores respectivamente. Esto elimina a lo sumo $\binom{12}{2} + \binom{12}{3}$ vectores adicionales. Sin embargo, cualquier vector remanente podría ser elegido. Como

$$1 + \binom{12}{1} + \binom{12}{2} + \binom{12}{3} \leq 2^9$$

sabemos que podemos hallar aún otro vector. Por ejemplo, uno puede elegir el vector 010101010101010 para que sea la siguiente fila de H. La tarea de hallar los vectores restantes de H se relega a los ejercicios.

Este ejemplo (y los ejercicios relacionados) muestran que un código $(15, 6, 5)$-lineal existe. El argumento cubierto establece también una *cota inferior* al tamaño máximo (o dimensión) de un código lineal con $n = 15$ y $d = 5$, donde $6 \leq k \leq 8$.

El siguiente resultado formaliza el enfoque del ejemplo dado para construir códigos lineales (y así establecer cotas inferiores). Las pruebas son dejadas también como ejercicios.

Teorema 24. (Cota de Gilbert-Varshamov) *Existe un código lineal de largo n, dimensión k y distancia d si*

$$\binom{n-1}{0} + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} < 2^{n-k}.$$

Corolario 25. *Si $n \neq 1$ y $d \neq 1$, entonces existe un código lineal C de largo n y distancia al menos d con*

$$|C| \geq \frac{2^{n-1}}{\binom{n-1}{0} + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2}}.$$

Ejemplo. ¿Existe un código lineal de largo $n = 9$, dimensión $k = 2$ y distancia $d = 5$?

Para determinar si tal código existe, usemos el Teorema 24:

$$\binom{n-1}{0} + \cdots + \binom{n-1}{d-2} = \binom{8}{0} + \binom{8}{1} + \binom{8}{2} + \binom{8}{3} = 93$$

y $2^{n-k} = 2^{9-2} = 2^7 = 128$. Como $93 < 128$, tal código existe.

Ejemplo. Establezcamos una cota inferior y otra superior para el tamaño $|C|$ ó dimensión k de un código lineal con $n = 9$ y $d = 5$. Para hallar una cota inferior para el máximo de palabras que un tal código C tenga, usamos el Corolario 25:

$$|C| \geq \frac{2^{n-1}}{\binom{n-1}{0} + \cdots + \binom{n-1}{d-2}} = \frac{2^{9-1}}{\binom{8}{0} + \binom{8}{1} + \binom{8}{2} + \binom{8}{3}} = \frac{2^8}{93} = 2{,}75.$$

Como C es linear, $|C|$ es una potencia de 2. Luego, $|C| \geq 4$.

Para hallar una cota superior de $|C|$, usamos la cota de Hamming:

$$|C| \leq \frac{2^9}{\binom{9}{0} + \binom{9}{1} + \binom{9}{2}} = \frac{512}{1+9+36} = \frac{512}{46} = 11{,}13.$$

Como $|C|$ es linear, es una potencia de 2. Luego, $|C| \leq 8$.

Combinando las cotas, un código linear con parámetros $(9, k, 5)$ y con 4 palabras existe, pero no existe ningún $(9, k, 5)$ con más de 8 palabras.

Ejemplo. ¿Existe algún código $(15, 7, 5)$-lineal? Nuevamente podemos intentar usar el Teorema 24 para responder la pregunta:

$$\binom{n-1}{0} + \cdots + \binom{n-1}{d-2} = \binom{14}{0} + \binom{14}{1} + \binom{14}{2} + \binom{14}{3}$$

$$= 1 + 14 + 91 + 364 = 470,$$

y $2^{n-k} = 2^{15-7} = 256$. En este caso no se satisface la desigualdad, de modo que el Teorema 24 no nos dice si tal código existe. De hecho, estos son los parámetros de un código BCH, a ser cubiertos más tarde.

Ejercicios.

6. Para cada parte del Ejercicio 2, con $k = 2d$ y cuando sea posible, decida si un código lineal con los parámetros dados existe. Halle cotas inferior y superior para el máximo número de palabras que tal código pueda tener, suponiendo que k no está restringido.

7. Halle cotas inferior y superior para el máximo número de palabras en un código lineal de largo n y distancia d cuando

   (a) $n = 15, d = 5$ (b) $n = 15, d = 3$ (c) $n = 11, d = 3$

   (d) $n = 12, d = 3$ (e) $n = 12, d = 4$ (f) $n = 12, d = 5$.

8. ¿Es posible tener un código lineal con parámetros $(8, 3, 5)$?

9. Hallar un código $(15, 6, 5)$-lineal construyendo la matriz de control de paridad. (Ver ejemplo previo al Teorema 24, donde cada uno de los 3 vectores que faltan deben tener peso al menos 4. ¿Por cual razón?).

10. Sea H_1 cualquier matriz $i \times (n-k)$ sin ningún conjunto linealmente dependiente formado por $d-1$ filas.

    a) Pruebe que hay a lo sumo

    $$N_i = \binom{i}{0} + \binom{i}{1} + \cdots + \binom{i}{d-2}$$

    palabras en $\mathbb{K}^{n-k}$ que son combinaciones lineales de a lo sumo $d-2$ filas de H_1.

    b) Pruebe que si $N_i < 2^{n-k}$, entonces se puede agregar una fila más a H_1 de tal forma que ningún conjunto de $d-1$ filas de la matriz resultante sea linealmente dependiente.

    c) Pruebe el Teorema 24.

    d) Pruebe el Corolario 25. (Pista: El máximo que $|C|$ puede asumir para largo n y distancia al menos d es tal que en $\mathbb{K}^n$ no hay ninguna palabra, aparte de las que ya están en C, a distancia $\leq d$ de cada palabra en C. En otras palabras, las esferas de radio $d-1$ centradas en las palabras $c \in C$ recubren todo $\mathbb{K}^n$).

3.2. Códigos perfectos

Un código C de largo n y distancia impar $d = 2t+1$ es llamado *código perfecto* si C alcanza la cota de Hamming del Teorema 20, o sea, si

$$|C| = \frac{2^n}{\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}}.$$

Desgraciadamente, no hay muchos códigos lineales perfectos, pero los que sí existen son muy útiles. El problema principal de hallar códigos lineales perfectos es que el número $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$ debe ser una potencia de 2, (pues C es una potencia de 2).

Ejemplo. Sea $t = 0$. Entonces, $\binom{n}{0} = 1 = 2^0$, de modo que $|C| = 2^n/\binom{n}{0} = 2^n$. El único código con 2^n palabras de largo n es $C = \mathbb{K}^n$. En este sentido, $\mathbb{K}^n$ es un código lineal perfecto.

Ejemplo. Sea $n = 2t + 1$. Como $\binom{n}{n-i} = \binom{n}{i}$, entonces

$$\binom{n}{0} + \cdots + \binom{n}{t} = \frac{1}{2}\left(\binom{n}{0} + \cdots + \binom{n}{n}\right) = \frac{1}{2} \cdot 2^n = 2^{n-1}.$$

Por lo tanto,

$$|C| = \frac{2^n}{\binom{n}{0} + \cdots + \binom{n}{t}} = \frac{2^n}{2^{n-1}} = 2.$$

Luego, todo código perfecto de largo y distancia $2t + 1$ tiene exactamente 2 palabras. Entre los códigos lineales existe exactamente un tal código, que es el código de repetición que consiste de la palabra nula y de la palabra cuyos dígitos valen todos 1. Y de hecho, ese código es perfecto.

Los códigos en estos dos últimos ejemplos no son muy interesantes. Por lo tanto, ellos son llamados códigos perfectos *triviales*.

Ejemplo. Sea $n = 7$ y $d = 3$. Entonces $t = 1$ y

$$|C| = \frac{2^7}{\binom{7}{0} + \binom{7}{1}} = \frac{128}{8} = 16 = 2^4.$$

Luego, puede existir un código lineal perfecto con $n = 7$ y $d = 3$, lo que se verá en la próxima sección bajo el nombre de código de Hamming.

Ejemplo. Sea $n = 23$ y $d = 7$. Entonces $t = 3$ y

$$|C| = \frac{2^{23}}{\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}} = \frac{2^{23}}{1 + 23 + 253 + 1771} = \frac{2^{23}}{2048}$$

$= \frac{2^{23}}{2^{11}} = 4096$. Esto muestra que un código lineal perfecto con $n = 23$ y $d = 7$ puede existir. Y de hecho, este existe y es conocido como código binario de Golay.

Ejercicios.

11. Muestre que para $n = 2^r - 1$, vale que $\binom{n}{0} + \binom{n}{1} = 2^r$.

12. ¿Pueden existir códigos perfectos para los siguientes valores de n y d?
    (a) $n = 15, d = 3$ (b) $n = 31, d = 3$ (c) $n = 15, d = 5$.

Los largos posibles para códigos perfectos fueron determinados por el finlandés Tietävainen, el holandés Van Lint y los rusos Zinoviev y Leontiev, en tres trabajos publicados en 1973. El teorema que probaron es el siguiente, cuya demostración queda fuera del alcance de estas notas.

Teorema 26. *Si C es un código perfecto de largo n y distancia $d = 2t + 1$, entonces: o bien $n = 23$ y $d = 7$, o bien $n = 2^r - 1$, para algún $r \geq 2$ y $d = 3$.*

Si un código lineal tiene largo n y distancia $d = 2t + 1$, entonces por el Teorema 3 vale que C corregirá todos los patrones de error de peso $\leq t = (d-1)/2$. Luego, cada palabra de largo n y peso $\leq t$ es un líder de clase módulo C. Hay exactamente $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$ tales palabras. Pero este es precisamente el número de clases módulo C, si C es perfecto. Acabamos de probar el siguiente teorema.

Teorema 27. *Si C es un código perfecto de largo n y distancia $d = 2t+1$, entonces C corregirá todos los patrones de error de peso $\leq t$, y ningún otro patrón de error.*

El Teorema 27 afirma que cada una de las 2^n palabras de $\mathbb{K}^n$ yace a una distancia $\leq t$ de exactamente una palabra-código. Esta propiedad nos permite, por ejemplo, contar el número de palabras-código de peso mínimo no nulo en un código perfecto.

Un código perfecto que corrige todos los patrones de error de peso $\leq t$ es llamado un *código perfecto t-corrector* (o sea, corrector de t errores). Del Teorema 26, los únicos valores posibles para t son $t = 1$ y $t = 3$. Examinamos el caso $t = 1$ en la próxima sección.

3.3. Códigos de Hamming

Presentamos una importante familia de códigos para los que será fácil codificar y decodificar, y que corrigen un error por vez.

Un código de largo $n = 2^r - 1$, $(r \geq 2)$, y que tenga una matriz de control de paridad cuyas filas consisten de vectores no nulos de largo r es llamado *código de Hamming* de largo $2^r - 1$.
Ejemplo. Una posibilidad para matriz de control de paridad de un código de Hamming de largo 7, $(r = 3)$, es

$$H = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix}.$$

Por el Algoritmo C de la página 46, tenemos que una matriz generadora para este código es

$$G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}.$$

Luego, este código tiene dimensión 4 y contiene 2^4 palabras. El Teorema 16 de la página 61 puede ser usado para deteminar la distancia del código, la cual es 3. La razón de información es $4/7$. En los ejercicios, codificamos algunos mensajes usando este código. Hay otras posibilidades para la matriz de control de paridad para un código de Hamming de largo 7, pero todos producen códigos equivalentes.

Como la matriz de control de paridad H de un código de Hamming C contiene todas las r filas de peso 1, entonces las r columnas de H son linealmente independientes. Luego, *un código de Hamming tiene dimensión* $2^r - 1 - r$ *y contiene* 2^{2^r-1-r} *palabras.*

La matriz H no posee ninguna fila que sea nula, de modo que H no posee un conjunto linealmente dependiente formado por una sola fila. Luego, C tiene distancia al menos 2. Tampoco hay dos filas distintas en H que sean iguales, de modo que H no posee un conjunto linealmente dependiente formado por dos filas distintas. Luego, C tiene distancia al menos 3. Pero H contiene las filas $v_1 = 100\ldots0, v_2 = 010\ldots0$ y $v_3 = v_1 + v_2 = 110\ldots0$, que forman un conjunto linealmente dependiente. Luego, por el Teorema 16, *un código de Hamming C tiene distancia 3.*

Ahora bien, para $n = 2^r - 1$ y $d = 2t + 1 = 3$ (o sea $t = 1$):

$$\frac{2^n}{\binom{n}{0} + \cdots + \binom{n}{t}} = \frac{2^n}{\binom{n}{0} + \binom{n}{1}} = \frac{2^{2^r-1}}{1+n} = \frac{2^{2^r-1}}{1+2^r-1} = 2^{2^r-r-1},$$

de modo que los códigos de Hamming son códigos perfectos. Más aún, por el Teorema 27, *códigos de Hamming son códigos perfectos 1-correctores.*

Es fácil construir un AED para un código de Hamming. Todas las palabras de largo $2^r - 1$ y peso 1 son patrones de error que se corrigen, y por lo tanto deben ser líderes de clase. Si e es un patrón de error, entonces eH suma todas las filas de la matriz de control de paridad H correspondientes a las posiciones en las cuales los errores acontecen. Como H tiene $2^r - 1$ filas, esto nos provee el siguiente arreglo como un AED para código de Hamming:

Líder de clase	Síndrome
$000\ldots0$	$00\ldots0$
I_{2^r-1}	H

Ejemplo. Decodifiquemos $w = 1101001$ para el código de Hamming del ejemplo anterior. El síndrome es $wH = 011$, que se presenta como la cuarta fila de H. Entonces, el líder de clase u es la cuarta fila de I_7: $u = 0001000$. Luego, decodificamos w como $w + u = 1100001$.

Ejercicios.

13. Halle una matriz generadora en forma estándar para un código de Hamming de largo 15. Luego, codifique el mensaje

    11111100000.

14. Construya un AED para un código de Hamming de largo 7 y úselo para decodificar las siguientes palabras:

    (a) 1101011 (b) 0011010 (c) 0100011

    (d) 1111111 (e) 0101011 (f) 0001011

15. Construya un AED para un código de Hamming de largo 15 y úselo para decodificar las siguientes palabras:

    (a) 01010 01010 01000 (b) 11110 00101 10110

    (c) 11100 01110 00111 (d) 11100 10110 00000

    (e) 00011 10100 00110 (f) 11001 11001 11000

16. Muestre que cada una de las siguientes matrices es una matriz de control de paridad para un código de Hamming de largo 7, y que los códigos son ambos equivalentes al dado en el último ejemplo.

$$H' = \begin{bmatrix} 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{bmatrix} \quad H'' = \begin{bmatrix} 100 \\ 110 \\ 111 \\ 011 \\ 101 \\ 010 \\ 001 \end{bmatrix}.$$

17. Pruebe que todos los códigos de Hamming que tienen el mismo largo son equivalentes.

18. ¿Es la siguiente matriz la transpuesta de la matriz de control de paridad para un código de Hamming de largo 15?

$$H^T = \begin{bmatrix} 100011011101000 \\ 111001000111110 \\ 010110010111101 \\ 100010101100111 \end{bmatrix}.$$

19. Muestre que el código de Hamming de largo $2^r - 1$ para $r = 2$ es un código trivial.

20. Use el código de Hamming de largo 7 del último ejemplo y la asignación de mensajes del Ejercicio 37 de las páginas 52 y 53 para decodificar el siguiente mensaje recibido: 1010111, 0110111, 1000010, 0010101, 1001011, 0010000, 1111100.

3.4. Códigos extendidos

A veces, acrecentar el largo de un código con un dígito adicional, o tal vez varios dígitos adicionales, resulta en un nuevo código con detección o corrección de error mejorada, que compensa por una razón menor de información de error. Consideremos una posibilidad sencilla en esta sección.

Sea C un código lineal de largo n. El código C^* de largo $n+1$ obtenido de C al agregar un dígito extra a cada palabra de C para hacer que cada palabra en el nuevo código tenga peso par, es llamado *código extendido* de C.

En el ejemplo de la página 9 se construyó un código extendido de C_2 y el lector hizo lo mismo para C_3 en el Ejercicio 8 de esa página.

Si el código original C tiene una $k \times n$ matriz generadora G, entonces el código extendido C^* tiene matriz generadora $k \times (n+1)$

$$G^* = [G, b],$$

donde la última columna b, que se coloca luego de G en G^*, hace que cada fila tenga peso par.

Una matriz de control de paridad H^* para C^* puede ser construida a partir de G^* usando el Algoritmo C. Pero existe una forma más fácil de construir tal H^* a partir de una matriz de control de paridad H de C. Con este procedimiento, el código extendido C^* tendrá matriz de control de paridad

$$H^* = \begin{bmatrix} H & j \\ 0 & 1 \end{bmatrix},$$

donde j es vector columna, o matriz $n \times 1$, con todas sus entradas valiendo 1. Nótese que H^* es una matriz $(n+1) \times (n+1-k)$. Como H tiene rango $n-k$, la última fila de H^* asegura que H^* tiene rango $n-k+1$. Más aún,

$$G^*H^* = [G, b] \begin{bmatrix} H & j \\ 0 & 1 \end{bmatrix} = [GH, Gj + b].$$

Ahora bien, $GH = 0$, y Gj suma los unos en cada fila de G. De la definición de b se sigue que $Gj + b = 0$. Por lo tanto, $G^*H^* = 0$. Por el Teorema 12 de la página 54, G^* y H^* son, de hecho, las matrices generadora y de control de paridad, respectivamente, para el código lineal C^*.

Ejemplo. Sea C el código lineal con matriz generadora

$$G = \begin{bmatrix} 10010 \\ 01001 \\ 00111 \end{bmatrix}.$$

Entonces,

$$H = \begin{bmatrix} 10 \\ 01 \\ 11 \\ 10 \\ 01 \end{bmatrix}$$

es la matriz de control de paridad para C, por el Algoritmo C. De modo que las matrices

generadora y de control de paridad para el código extendido C^* son

$$G^* = \begin{bmatrix} 10010|0 \\ 01001|0 \\ 00111|1 \end{bmatrix} \text{ y } H^* = \begin{bmatrix} 10|1 \\ 01|1 \\ 11|1 \\ 10|1 \\ 01|1 \\ 00|1 \end{bmatrix}.$$

Si v es una palabra en el código original C y si v^* es la palabra correspondiente en el código extendido C^*, entonces

$$\mathrm{w}(v^*) = \begin{cases} \mathrm{w}(v), & \text{si } \mathrm{w}(v) \text{ es par} \\ \mathrm{w}(v)+1, & \text{si } \mathrm{w}(v) \text{ es impar} \end{cases}$$

Por lo tanto, si la distancia d de C es impar, entonces la distancia de C^* es $d+1$, pero si d es par, entonces la distancia de C^* es todavía d. Así, un código extendido tiene algún interés sólo cuando d es impar, en cuyo caso no corrige más errores que C, pero detecta un error más. Nótese que no vale la pena en extender un código un par de veces.

Ejemplo. Supongamos que C tiene distancia $d = 5$. Entonces C^* tiene distancia $d^* = 6$. Por el Teorema 2 de la página 27, C detecta todos los patrones de error no nulos de peso $\leq d-1 = 4$, y C^* detecta todos los patrones de error no nulos de peso $\leq d^* - 1 = 5$. Por el Teorema 3 de la página 30, C corrige todos los patrones de error de peso $\leq \lfloor (d-1)/2 \rfloor = \lfloor 4/2 \rfloor = 2$, y C^* corrige todos los patrones de error de pesos $\leq \lfloor (d^*-1)/2 \rfloor = \lfloor 5/2 \rfloor = 2$.

Ejercicios.

21. Halle las matrices generadora y de control de paridad para un código extendido de Hamming de largo 8.

22. Construya un AED para un código extendido de Hamming de largo 8 y úselo para decodificar las siguientes palabras:

    (a)10101010 (b) 11010110 (c) 11111111

23. Muestre que un código extendido de largo 8 es *auto-dual* (o sea, muestre que $C = C^{\perp}$).

24. Halle una fórmula para la distancia d^* de un código extendido C^* en términos de la distancia del código original C.

25. Sea C un código de Hamming de largo 15. Halle el número de patrones de error que C^* corregirá por el Teorema 2. ¿Cuántos patrones de error corrige C^*?

CAPÍTULO 4
De cómo se acotan los polinomios binarios

Hallamos conveniente representar códigos en términos de polinomios. Para ello, revistamos conceptos necesarios sobre polinomios (en una indeterminada x que se conoce también como incógnita o variable).

4.1. Polinomios y palabras

Un *polinomio de grado n sobre* $\mathbb{K}$ es una expresión $a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$, donde $a_0, a_1, a_2, \ldots, a_n$ son elementos del cuerpo $\mathbb{K}$. El conjunto de polinomios sobre $\mathbb{K}$ es denotado $\mathbb{K}[x]$. Elementos de $\mathbb{K}[x]$ son denotados sucintamente, por ejemplo, por medio de $f(x), g(x), h(x), p(x)$, etc. Además, el grado de un polinomio $f(x)$ es indicado $\mathrm{gr}(f(x))$.

Sumamos y multiplicamos los polinomios sobre $\mathbb{K}$ en forma usual, excepto que como $1+1 = 0$, tenemos que $x^k + x^k = 0$. Por lo tanto, el grado de $f(x) + g(x)$ no es necesariamente $\max\{\mathrm{gr}(f(x)), \mathrm{gr}(g(x))\}$.

Ejemplo. Sean $f(x) = 1 + x + x^3 + x^4$, $g(x) = x + x^2 + x^3$ y $h(x) = 1 + x^2 + x^4$. Entonces,

1. $f(x) + g(x) = 1 + x^2 + x^4$;
2. $f(x) + h(x) = x + x^2 + x^3$;
3. $f(x)g(x) = (x + x^2 + x^3) + x(x + x^2 + x^3) + x^3(x + x^2 + x^3) + x^4(x + x^2 + x^3) = x + x^7$.

Ejercicios.

1. Halle la suma y el producto de cada uno de los siguientes pares de polinomios sobre $\mathbb{K}$:
   a) $f(x) = x^5 + x^6 + x^7, h(x) = 1 + x^2 + x^3 + x^4$;
   b) $f(x) = 1 + x^2 + x^3 + x^8 + x^{13}, h(x) = 1 + x^3 + x^9$;
   c) $f(x) = 1 + x, h(x) = 1 + x + x^2 + x^3 + x^4$.
2. Sea $f(x) = 1 + x$. Halle
   (a) $(f(x))^2$ (b) $(f(x))^3$ (c) $(f(x))^4$.
3. Repita el Ejercicio 2 para $f(x) = 1 + x + x^2$.

4. Liste todos los polinomios sobre $\mathbb{K}$ de grado n, para $n = 0$, $n = 2$, $n = 3$ y $n = 4$.

5. Halle el número de polinomios sobre $\mathbb{K}$ de grado a lo sumo 10.

6. Para cualesquiera polinomios $f(x)$ y $g(x)$ sobre $\mathbb{K}[x]$,

$$(f(x) + g(x))^2 = (f(x))^2 + (g(x))^2,$$

pues $x^k + x^k = 0$. ¿Existe alguna regla especial sobre $\mathbb{K}[x]$ para

a) $(f(x) + g(x))^4$?

b) $(f(x) + g(x))^3$?

c) $(f(x) + g(x))^n$, para cualquier entero positivo n?

El proceso usual de división larga funciona para polinomios sobre $\mathbb{K}$ en forma similar a como funciona para números racionales.

Algoritmo de División: Sean $f(x), h(x) \in \mathbb{K}[x]$ con $h(x) \neq 0$. Entonces, existen polinomios unívocamente determinados $q(x)$ y $r(x)$ tales que

$$f(x) = q(x)h(x) + r(x),$$

con $r(x) = 0$ ó $\mathrm{gr}(r(x)) < \mathrm{gr}(h(x))$.

El polinomio $q(x)$ es llamado *cociente* de la división, y $r(x)$ es llamado *resto* de la división. Un procedimiento para hallar cociente y resto cuando $f(x)$ es dividido por $h(x)$ es el de división larga, pero con la aritmética entre coeficientes considerada módulo $\mathbb{K}$.

Ejemplo. Sea $f(x) = x + x^2 + x^6 + x^7 + x^8$ y $h(x) = 1 + x + x^2 + x^4$. Entonces,

$$\begin{array}{lllllll|l}
x^8 & +x^7 & +x^6 & +x^2 & +x & & & x^4+x^2+x+1 \\
\cline{8-8}
x^8 & +x^6 & +x^5 & +x^4 & & & & x^4+x^3 \\
\cline{1-4}
 & x^7 & +x^5 & +x^4 & +x^2 & +x & & \\
 & x^7 & +x^5 & +x^4 & +x^3 & & & \\
\cline{1-5}
 & & & & x^3 & +x^2 & +x &
\end{array}$$

Luego, el cociente de esta división es $q(x) = x^3 + x^4$ y el resto correspondiente es $r(x) = x + x^2 + x^3$. Podemos escribir $f(x) = h(x)(x^3 + x^4) + (x + x^2 + x^3)$. Nótese que $3 = \mathrm{gr}(r(x)) < \mathrm{gr}(h(x)) = 4$.

Ejercicios.

6. Halle el cociente y el resto cuando $h(x)$ es dividido por $f(x)$, para cada uno de los pares de polinomios del Ejercicio 1.

7. Halle el cociente y el resto en cada ítem abajo, cuando $h(x)$ es dividido por $f(x)$.

   a) $f(x) = x^2 + x^3 + x^4 + x^8, h(x) = 1 + x^5$

   b) $f(x) = 1 + x^{10}, h(x) = 1 + x^5$

   c) $f(x) = 1 + x^7, h(x) = 1 + x + x^3$

   d) $f(x) = 1 + x^{15}, h(x) = 1 + x^4 + x^6 + x^7 + x^8$.

El polinomio $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1}$ de grado a lo sumo $n-1$ sobre $\mathbb{K}$ puede ser considerado como la palabra $v = a_0a_1 \ldots a_{n-1}$ de largo n sobre $\mathbb{K}^n$. Por ejemplo si $n = 7$,

Polinomio	Palabra
$1 + x + x^2 + x^4$	1110100
$1 + x^4 + x^5 + x^6$	1000111
$1 + x + x^3$	1101000

Luego, *un código C de largo n puede ser representado como un conjunto de polinomios sobre $\mathbb{K}$ de grado a lo sumo $n-1$.*

Es conveniente, para propósitos de representación de palabras por medio de polinomios, el numerar los dígitos de una palabra de largo n desde 0 hasta $n-1$, más bien que desde 1 hasta n. Por ejemplo, la palabra $a_0a_1a_2a_3$ de largo 4 es representada por el polinomio $a_0 + a_1x + a_2x^2 + a_3x^3$, de grado 3.

Ejemplo. El código C presentado en la columna izquierda del arreglo abajo está representado por los polinomios en el lado derecho:

Palabra-código c	Polinomio $c(x)$
0000	0
1010	$1 + x^2$
0101	$x + x^3$
1111	$1 + x + x^2 + x^3$

Ejercicios.

8. Represente cada palabra de C en los siguientes códigos por medio de un polinomio:

   a) $C = \{000, 001, 010, 011\}$

   b) $C = \{00000, 11111\}$

   c) $C = 0000, 0001, 1110\}$

   d) $C = \{0000, 1001, 0110, 1111\}$

   e) $C = \{00000, 11100, 00111, 11011\}$.

9. Escriba el código de Hamming de largo 7 generado por la matriz G, y luego represente este código por medio de polinomios.

$$G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}$$

En el Ejercicio 7, el lector computó el resto $r(x)$ cuando $f(x) = x^2 + x^3 + x^4 + x^8$ era dividido por $h(x) = 1 + x^5$. El resultado fue $r(x) = x^2 + x^4$. Por el Algoritmo de División, $r(x)$ tiene grado menor que el grado del divisor $h(x)$.

Decimos que $f(x)$ *módulo* $h(x)$ es igual a $r(x)$ si $r(x)$ es el resto de dividir $f(x)$ por $h(x)$. Más aún, decimos que dos polinomios $f(x)$ y $p(x)$ son *equivalentes módulo* $h(x)$ si ellos tienen el mismo resto cuando son divididos por $h(x)$, o sea, si

$$f(x) \mod h(x) = r(x) = p(x) \mod h(x).$$

Indicamos esto por medio de

$$f(x) \equiv p(x) \pmod{h(x)}.$$

Ejemplo. Sea $h(x) = 1 + x^5$ y $f(x) = 1 + x^4 + x^9 + x^{11}$. Entonces, dividiendo $f(x)$ por $h(x)$ nos da $r(x) = 1 + x$. Decimos que $r(x) = f(x) \mod h(x)$.

En forma similar, si $p(x) = 1 + x^6$, entonces $1 + x = 1 + x^6 \mod (1 + x^5)$, y escribimos $p(x) \equiv f(x) \pmod{h(x)}$.

Ejemplo. Sea $h(x) = 1 + x^2 + x^5$. Computando $f(x) \mod h(x)$, con $f(x) = 1 + x^2 + x^6 + x^9 + x^{11}$, hallamos el resto $r(x) = x + x^4$, y de aquí $x + x^4 = f(x) \mod h(x)$. Nótese que si $p(x) = x^2 + x^8$, entonces $p(x) \mod h(x) = 1 + x^3$. Luego, $p(x)$ y $f(x)$ no son equivalentes módulo $h(x)$.

La suma y la multiplicación de polinomios son compatibles con la equivalencia de polinomios definida arriba. O sea:

Lema 28. *Si* $f(x) \equiv g(x) \pmod{h(x)}$, *entonces*

$$f(x) + p(x) \equiv g(x) + p(x) \pmod{h(x)}$$

y

$$f(x)p(x) \equiv g(x)p(x) \pmod{h(x)}.$$

Demostración. Supongamos que $r(x) = f(x) \mod h(x)$, que $r(x) = g(x) \mod h(x)$ y que $s(x) = p(x) \mod h(x)$. Entonces,

$$\begin{aligned} f(x) + p(x) &= q_1(x)h(x) + r(x) + q_2(x)h(x) + s(x) \\ &= (q_1(x) + q_2(x))h(x) + r(x) + s(x) \end{aligned}$$

Luego, $r(x) + s(x) = f(x) + p(x) \mod h(x)$, pues $\mathrm{gr}(r(x) + s(x)) < \mathrm{gr}(h(x))$. (¿Por qué?). Argumentos similares muestran que $r(x)s(x) = g(x)p(x) \mod h(x)$. Dejamos los argumentos restantes como ejercicios. □

Ejemplo. Sea $h(x) = 1 + x^5, f(x) = 1 + x + x^7, g(x) = 1 + x + x^2$ y $p(x) = 1 + x^6$, de modo que $f(x) \equiv g(x) \pmod{h(x)}$. Entonces,

$$f(x) + p(x) = x + x^6 + x^7$$

y

$$g(x) + p(x) = x + x^2 + x^6$$

pero

$$(x + x^6 + x^7) \mod h(x) = x^2 = (x + x^2 + x^6) \mod h(x).$$

Del mismo modo,

$$(1 + x + x^7)(1 + x^6) \mod h(x) = 1 + x^3 = (1 + x + x^2)(1 + x^6) \mod h(x).$$

Nótese que $1 + x = (1 + x^6) \mod h(x)$. Entonces tenemos

$$\begin{aligned}(1 + x + x^7)(1 + x^6) &\equiv (1 + x + x^2)(1 + x^6) \\ &\equiv (1 + x + x^2)(1 + x) \equiv 1 + x^3 \pmod{h(x)}.\end{aligned}$$

Ejercicios.

10. Sea $h(x) = 1 + x^3 + x^5$. Compute $f(x) \mod h(x)$ y su correspondiente palabra, cuando:
    a) $f(x) = 1 + x + x^6$;
    b) $f(x) = x + x^4 + x^7 + x^8$;
    c) $f(x) = 1 + x^{10}$.
11. Sea $h(x) = 1 + x^7$. Compute $f(x) \mod h(x)$ y $p(x) \mod h(x)$, y decida si $f(x) \equiv p(x) \pmod{h(x)}$, cuando:
    a) $f(x) = 1 + x^3 + x^8, p(x) = x + x^3 + x^7$;
    b) $f(x) = x + x^5 + x^9, p(x) = x + x^5 + x^6 + x^{13}$;
    c) $f(x) = 1 + x, p(x) = x + x^7$.
12. Sea $h(x) = 1 + x^7$. Compute $(f(x) + g(x)) \mod h(x)$ y $(f(x)g(x)) \mod h(x)$, donde
    a) $f(x) = 1 + x^6 + x^8, g(x) = 1 + x$;
    b) $f(x) = 1 + x^5 + x^9, g(x) = x + x^2 + x^7$;
    c) $f(x) = 1 + x^4 + x^4, g(x) = 1 + x + x^2$.
13. Pruebe que si $f(x) \equiv g(x) \pmod{h(x)}$, entonces $f(x)p(x) \equiv g(x)p(x) \pmod{h(x)}$.

4.2. Introducción a los códigos cíclicos

Comenzamos el estudio de una clase de códigos llamados *códigos cíclicos*. En un momento dado estaremos preparados para usar nuestro conocimiento sobre estos códigos cíclicos para construir una matriz generadora que corrija dos o más errores. Veremos también que los códigos de Hamming son códigos cíclicos.

Sea v una palabra de largo n. Decimos que el *traslado cíclico* $\pi(v)$ de v es la palabra de largo n obtenida a partir de v al tomar su último dígito, moviéndolo al comienzo de v, y siendo todos los demás dígitos movidos una posición hacia su derecha. Por ejemplo,

v	10110	111000	0000	1011
$\pi(v)$	01011	011100	0000	1101

Decimos que un código C es *cíclico* si el traslado cíclico de cada palabra de C es una palabra de C.

Ejemplo. El código lineal $C = \{000, 110, 101, 011\}$ es un código cíclico. Computamos $\pi(v)$, para todo $V \in C$:

$$\pi(000) = 000,\ \pi(110) = 011,\ \pi(101) = 110,\ \pi(011) = 101.$$

Como $\pi(v)$ está también en C para cada $v \in C$, vale que C es cíclico.

Ejemplo. Veamos que el código $C = \{000, 100, 011, 111\}$ no es cíclico: el traslado cíclico de $v = 100$ es $\pi(100) = 010$, que no está en C.

Lema 29. *$\pi(v + w) = \pi(v) + \pi(w)$ y $\pi(av) = a\pi(v)$, donde $v, w \in \mathbb{K}^n$ y $a \in \mathbb{K}$. Luego, para mostrar que un código lineal C es cíclico, basta mostrar que $\pi(v) \in C$ para cada palabra v en una base de C.*

Demostración. Si $v = (v_0, v_1, \ldots, v_{n-1})$ y $w = (w_0, w_1, \ldots, w_{n-1})$ entonces $v+w = (v_0+w_0, v_1+w_1, \ldots, v_{n-1}+w_{n-1})$ y $\pi(v + w) = (v_{n-1} + w_{n-1}, v_0 + w_0, \ldots, v_{n-2} + w_{n-2}) = \pi(v) + \pi(w)$. □

Ejemplo. En el primer ejemplo de esta sección, $\{110, 101\}$ es una base de C. Como $\pi(110) = 011$ y $\pi(101) = 110$ están en C, entonces C es un código cíclico lineal.

Si deseamos construir un código cíclico lineal C, entonces tomamos una palabra v, formamos el conjunto S que consiste de v y de sus traslados cíclicos sucesivos, o sea $S = \{v, \pi(v), \pi^2(v), \ldots, \pi^{n-1}(v)\}$, y definimos C como la expansión lineal de S; o sea $C = \langle S\rangle$. (Usamos la notación $\pi^2(v) = \pi(\pi(v)), \pi^3(v) = \pi(\pi(\pi(v)))$, etc.) Como S contiene una base de C, luego C debe ser cíclico, por el Lema 29.

Ejemplo. Sea $n = 3$ y $v = 100$. Entonces $S = \{v, \pi(v), \pi^2(v)\} = \{100, 010, 001\}$ y $\langle S\rangle{=}\mathbb{K}^3$. Nótese que si $w = a_0v + a_1\pi(v) + a_2\pi^2(v)$, entonces $\pi(w) = a_0\pi(v) + a_1\pi^2(v) + a_2\pi^3(v) = a_2v + a_0\pi(v) + a_1\pi^2(v)$.

Ejemplo. Sea $n = 4$ y $v = 0101$. Entonces $\pi(v) = 1010$ y $\pi^2(v) = 0101 = v$. Luego, $S = \{0101, 1010\}$ y $C = \langle S\rangle$ es el código cíclico $C = \{0000, 0101, 1010, 1111\}$.

Si una palabra v y sus traslados cíclicos sucesivos forman un conjunto $S = \{v, \pi(v), \ldots, \pi^{n-1}(v)\}$ que expande el código C (de modo que $C = \langle S\rangle$), entonces decimos que v es un *generador* del código cíclico lineal C. Como todo código cíclico lineal que contiene a v debe contener a todo S, decimos que C es el menor código cíclico lineal que contiene a v. Vale la pena notar que un código cíclico lineal puede tener muchos generadores.

Ejercicios.

14. Halle una base para el menor código cíclico lineal de largo n que contenga a v:

    a) $v = 1101000, n = 7$;

    b) $v = 010101, n = 6$;

    c) $v = 11011000, n = 8$.

15. Halle todas las palabras v de largo n tal que $\pi(v) = v$.

16. Halle todas las palabras v de largo 6 tales que

    a) $\pi^2(v) = v$;

    b) $\pi^3(v) = v$.

Los códigos cíclicos tienen una representación conveniente en términos de polinomios. Esta representación está basada en la simple observación de que si la palabra v corresponde al polinomio $v(x)$, entonces el traslado cíclico $\pi(v)$ de v corresponde al polinomio $xv(x) \mod 1+x^n$. Nótese que en general $1 \equiv x^n \pmod{1+x^n}$.

Ejemplo. Si $v = 100$, entonces $v(x) = x^0 = 1$, y $\pi(v) = 010$ corresponde a $xv(x) = x$. Del mismo modo, si $v = 1101$ entonces $v(x) = 1 + x + x^3$, y $\pi(v) = 1110$ corresponde a $xv(x) \mod 1 + x^4 = 1 + x + x^2$.

Nos referimos a los elementos del un código cíclico no sólo como palabras sino también como polinomios. Luego, podemos poner la discusión hecha arriba en términos de polinomios. Dada una palabra v de largo n, indicamos con $v(x)$ el polinomio correspondiente a v. Entonces, los traslados cíclicos sucesivos de v corresponden a los polinomios $x^i v(x) \mod 1 + x^n$, para $i = 0, 1, \ldots, n-1$.

Ejemplo. Sea $v = 1101000$ y $n = 7$. Entonces $v(x) = 1 + x + x^3$. Computamos $x^i v(x)$ para $1 \leq i \leq 6$ en la siguiente tabla de representaciones polinómicas de traslados cíclicos sucesivos:

Palabra	Polinomio mod $(1 + x^7)$
0110100	$xv(x) = x + x^2 + x^4$
0011010	$x^2v(x) = x^2 + x^3 + x^5$
0001101	$x^3v(x) = x^3 + x^4 + x^6$
1000110	$x^4v(x) = x^4 + x^5 + x^7 \equiv 1 + x^4 + x^5 \pmod{1+x^7}$
0100011	$x^5v(x) = x^5 + x^6 + x^8 \equiv x + x^5 + x^6 \pmod{1+x^7}$
1010001	$x^6v(x) = x^6 + x^7 + x^9 \equiv 1 + x^2 + x^6 \pmod{1+x^7}$

Claramente, si $c(x) \in \langle\{v(x), xv(x), \ldots, x^{n-1}v(x)\}\rangle$ (reduciendo cada producto mod $1+x^n$), entonces

$$\begin{aligned} c(x) &= (a_0 v(x) + a_1 xv(x) + \cdots + a_{n-1}x^{n-1}v(x)) \mod 1 + x^n \\ &= (a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1}x^{n-1})v(x) \mod 1 + x^n \\ &= a(x)v(x) \mod 1 + x^n. \end{aligned}$$

Por lo tanto, obtenemos el siguiente resultado.

Lema 30. *Sea C un código cíclico lineal y sea $v \in C$. Entonces para cada polinomio $a(x)$, $c(x) = a(x)v(x) \mod (1 + x^n)$ es una palabra de C.*

Entre todas las palabras no nulas en un código cíclico lineal C, existe una única palabra $g \in C$ tal que $g(x)$ tiene grado mínimo, como el siguiente argumento indica. Es claro que existe en C al menos una palabra o polinomio de menor grado. Si dos palabras no nulas g y g' corresponden a polinomios $g(x)$ y $g'(x)$ de grado mínimo k, entonces $g(x) + g'(x) = c(x) \in C$, (pues C es lineal), y $\mathrm{gr}(c(x)) < k$ (pues $x^k + x^k = 0$). Como g es una palabra no nula de grado mínimo, $\mathrm{gr}(c(x)) < k$ quiere decir $c(x) = 0$, de modo que $g(x) = g'(x)$, y así $g(x)$ es única.

Definimos el *polinomio generador* del código cíclico lineal C como el único polinomio no nulo de peso mínimo en C. De la discusión pasada, sabemos que es único. Pero, ¿es realmente un generador?

Para ver esto, debemos mostrar que para cualquier palabra $c(x)$ en C existe $a(x)$ tal que $c(x) = a(x)g(x) \mod 1 + x^n$. De hecho, mostraremos que $c(x) = a(x)g(x)$. Como $\mathrm{gr}(c(x)) \geq \mathrm{gr}(g(x))$, tenemos por el Algoritmo de División que:

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

o sea

$$r(x) = q(x)g(x) + c(x).$$

Sin embargo, ambos polinomios, $c(x)$ y $q(x)g(x)$, son palabras de C por el Lema 30 y entonces también $r(x)$ es palabra de C. Pero, por el Algoritmo de División, $r(x) = 0$ ó $\mathrm{gr}(r(x)) < \mathrm{gr}(g(x))$. Como la última opción es imposible salvo si $r = 0$, concluimos que $r(x) = 0$, y luego, $g(x)$ es divisor de toda palabra $c(x) \in C$.

Teorema 31. *Sea C un código cíclico de largo n y sea $g(x)$ su polinomio generador. Si $k = n - \mathrm{gr}(g(x))$, entonces*

1. *C tiene dimensión k,*

2. *las palabras código correspondientes a $g(x)$, $xg(x), \ldots,$*

   $x^{k-1}g(x)$ forman una base de C, y

3. *$c(x) \in C$ si y solo si $c(x) = a(x)g(x)$, para algún polinomio $a(x)$ con $\mathrm{gr}(a(x)) < k$, (o sea, $g(x)$ es un divisor de cada polinomio $c(x)$).*

Demostración. La discusión previa al enunciado del teorema demuestra (3). Si $g(x)$ tiene grado $n - k$, entonces $g(x)$, $xg(x), \ldots,$ $x^{k-1}g(x)$ deben ser linealmente independientes. (¿Por qué?). Como $g(x)$ divide cada palabra-código, existe un único polinomio $a(x) = a_0 + a_1x + \cdots + a_{k-1}x^{k-1}$ tal que $c(x) = a(x)g(x) = a_0g(x) + a_1xg(x) + \cdots + a_{k-1}x^{k-1}g(x)$. Por lo tanto, $c(x)$ está en $\langle\{g(x), xg(x), \ldots, x^{k-1}g(x)\}\rangle$, y luego, $\{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ es una base de C. □

Ejemplo. Sea $n = 7$ y $g(x) = 1 + x + x^3$ un polinomio generador para el código cíclico C. Una base de C es

$$\begin{aligned} g(x) &= 1 + x + x^3 &&\leftrightarrow 1101000 \\ xg(x) &= x + x^2 + x^4 &&\leftrightarrow 0110100 \\ x^2g(x) &= x^2 + x^3 + x^5 &&\leftrightarrow 0011010 \\ x^3g(x) &= x^3 + x^4 + x^6 &&\leftrightarrow 0001101 \end{aligned}$$

Nótese que $x^4g(x) \mod 1 + x^7 = 1 + x^4 + x^5$ es una palabra-código, pues $1 + x^4 + x^5 = (1 + x + x^2)(1 + x + x^3) = (1 + x + x^2)g(x)$.

Ejemplo. Sea C el código cíclico $\{0000, 1010, 0101, 1111\}$. El conjunto de polinomios correspondiente es $\{0, 1 + x^2, x + x^3, 1 + x + x^2 + x^3\}$. Nótese que $1 + x^2 \leftrightarrow 1010$ es el polinomio generador de C, pues C contiene sólo un polinomio de grado 2 y ninguno de grado 1. Además, cada palabra (polinomio) en C es un múltiplo del polinomio generador:

$$\begin{aligned} 0 &= 0(1 + x^2), & x + x^3 &= x(1 + x^2), \\ 1 + x^2 &= 1(1 + x^2), & 1 + x + x^2 + x^3 &= (1 + x)(1 + x^2). \end{aligned}$$

Ejemplo. El código cíclico lineal más pequeño C de largo 6 que contiene $g(x) = 1 + x^3 \leftrightarrow 100100$ es

$$\{000000, 100100, 010010, 001001, 110110, 101101, 011011, 111111\}$$

Esto puede ser verificado con las técnicas descritas arriba. El polinomio de menor grado que representa una palabra en C es encontrado por inspección: $g(x) = 1 + x^3$, y C no contiene otro polinomio de grado 3. Luego, $g(x) = 1 + x^3$ es el polinomio generador de C. Representamos cada palabra no nula en C como un múltiplo de $g(x)$ en la siguiente tabla de palabras código

que son múltiplos del polinomio generador:

Palabra	Polinomio $f(x)$	Factorización $h(x)g(x)$ de $f(x)$
100100	$1+x^3$	$1(1+x^3)$
010010	$x+x^4$	$x(1+x^3)$
001001	x^2+x^5	$x^2(1+x^3)$
110110	$1+x+x^3+x^4$	$(1+x)(1+x^3)$
101101	$1+x^2+x^3+x^5$	$(1+x^2)(1+x^3)$
011011	$x+x^2+x^4+x^5$	$(x+x^2)(1+x^3)$
111111	$1+x+x^2+x^3+x^4+x^5$	$(1+x+x^2)(1+x^3)$

Podemos generar códigos cíclicos de forma bastante fácil tomando una palabra v y poniendo $C = \langle\{v(x), xv(x), \ldots, x^{k-1}v(x)\}\rangle \pmod{1+x^n}$. Sin embargo, precisamos hallar el polinomio generador para tal código, por lo que listar todas las palabras código no es un abordaje razonable. El polinomio generador para un código cíclico tiene una propiedad importante:

Teorema 32. *$g(x)$ es el polinomio generador de un código cíclico lineal de largo n si y solo si $g(x)$ divide $1+x^n$, (de modo que $1+x^n = h(x)g(x)$).*

Demostración. Por el Lema 30, $c(x) \equiv h(x)g(x) \pmod{1+x^n}$ implica que $c(x) = h(x)g(x) + q(x)(1+x^n)$ es una palabra-código, para cada polinomio $h(x)$. Por el Algoritmo de División, $g(x)$ dividirá cada palabra-código $c(x)$ si y solo si divide $1+x^n$. De aquí, por el Teorema 31, $g(x)$ es el polinomio generador de un código cíclico de largo n si y solo si $g(x)$ divide $1+x^n$. □

Corolario 33. *El polinomio generador $g(x)$ del código cíclico lineal más pequeño de largo n que contenga la palabra v (polinomio $v(x)$) es el máximo común divisor de $v(x)$ y $1+x^n$, (o sea, $g(x) = \mathrm{mcd}(v(x), 1+x^n)$).*

Demostración. Si $g(x)$ es el polinomio generador, entonces $g(x)$ divide ambos $v(x)$ y $1+x^n$. Pero $g(x)$ se halla en

$$\langle\{v(x), xv(x), \ldots, x^{n-1}v(x)\}\rangle,$$

por lo que tenemos:

$$g(x) = a(x)v(x) \mod 1+x^n$$

o equivalentemente por el Algoritmo de División:

$$g(x) = a(x)v(x) + b(x)(1+x^n).$$

Luego, cada divisor común de $v(x)$ y $1+x^n$ debe dividir $g(x)$. Luego, $g(x)$ es el máximo común divisor citado. □

Ejemplo. Sea $n = 8$ y $v = 11011000$. Luego, $v(x) = 1+x+x^3+x^4$. El máximo común divisor de $v(x)$ y $1+x^n = 1+x^8$ es $1+x^2$. Entonces, $g(x) = 1+x^2$, y el menor código cíclico lineal que contiene a $v(x)$ tiene dimensión 6 y $g(x)$ como polinomio generador.

El algoritmo de Euclides para computar el máximo común divisor de dos polinomios, (verApéndice B), puede usarse en estas circunstancias. Otro abordaje alternativo para hallar el polinomio generador de un código cíclico de largo n y dimensión k usa simplemente reducción de filas de matrices. Si tomamos una base (o matriz generadora) y la ponemos en forma escalonada reducida (o en FER) con las primeras k columnas como columnas líderes, entonces la fila (palabra-código) de menor grado será el polinomio generador.

Ejercicios.

17. Halle el polinomio generador del código cíclico lineal más pequeño que contenga la palabra:

    a) 010101;
    b) 010010;
    c) 01100110;
    d) 0101100;
    e) 001000101110000;
    f) 000010010000000;
    g) 010111010000000.

18. Halle el polinomio generador del menor código cíclico lineal que contenga la palabra

    (a) 101010 (b) 1100
    (c) 10001000 (d) 011011
    (e) 10101 (f) 111111

19. Para cada código $C = \langle S \rangle$, halle el polinomio generador $g(x)$, y luego represente cada palabra en C como múltiplo de $g(x)$:

    a) $S = \{010, 011, 111\}$;
    b) $S = \{1010, 0101, 1111\}$;
    c) $S = 0101, 1010, 1100\}$;
    d) $S = \{1000, 0100, 0010, 0001\}$;
    e) $S = \{11000, 01111, 11110, 01010\}$.

4.3. Las matrices G y H para códigos cíclicos

Podemos hallar varias matrices generadoras para códigos cíclicos lineales. La más simple es la matriz en la cual las filas son las palabras código correspondientes al polinomio generador y a sus primeros $k-1$ traslados cíclicos sucesivos (ver Teorema 31).

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}$$

Ejemplo. Sea $C = \{0000, 1010, 0101, 1111\}$ el código cíclico lineal. El polinomio generador para C es $g(x) = 1 + x^2$. Aquí, $n = 4$ y $k = 2$, de modo que una base de C consiste de

$$g(x) = 1 + x^2 \leftrightarrow 1010, xg(x) = x + x^3 \leftrightarrow 0101,$$

como puede verificarse fácilmente. Una matriz generadora para C es

$$G = \begin{bmatrix} g(x) \\ xg(x) \end{bmatrix} = \begin{bmatrix} 1010 \\ 0101 \end{bmatrix}.$$

Ejemplo. Sea C un código cíclico lineal de largo $n = 7$ con polinomio generador $g(x) = 1 + x + x^3$ de grado $n - k = 3$. Entonces, $k = 4$, de modo que una base de C es

$$\begin{aligned} g(x) &= 1 + x + x^3 \\ xg(x) &= x + x^2 + x^4 \\ x^2g(x) &= x^2 + x^3 + x^5 \\ x^3g(x) &= x^3 + x^4 + x^6 \end{aligned}$$

y una matriz generadora para C es

$$G = \begin{bmatrix} 1101000 \\ 0110100 \\ 0011010 \\ 0001101 \end{bmatrix}.$$

Sea C un código cíclico lineal de largo n y dimensión k (de modo que el polinomio generador $g(x)$ tiene grado $n - k$). La k-upla de dígitos de información $(a_0, a_1, \ldots, a_{k-1})$ a ser codificada puede ser representada por el polinomio $a(x) = a_0 + a_1x + \cdots + a_{k-1}x^{k-1}$, llamado *polinomio mensaje o de información*. La codificación consiste simplemente en multiplicación polinomial, o sea: $a(x)g(x) = c(x)$, de modo que, en lugar de almacenar toda la matriz $k \times n$ generadora, solo tenemos que almacenar el polinomio generador, lo cual constituye un mejoramiento significativo en términos de complejidad de la codificación.

La operación inversa a la multiplicación polinomial es la división polinomial. Por lo tanto, hallar el mensaje correspondiente a la palabra-código $c(x)$ más próxima a la palabra recibida consiste en dividir $c(x)$ por $g(x)$, así recobrando el polinomio mensaje $a(x)$.

Ejemplo. Sea $g(x) = 1 + x + x^3$ y $n = 7$. Entonces, $n - k = 7 - 4 = 3$. Sea $a(x) = 1 + x^2$ el polinomio mensaje correspondiente a la palabra $a = 1010$. El mensaje $a(x)$ es codificado como $c(x) = a(x)g(x)$, de modo que

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

con $c = 1110010$ como palabra-código correspondiente.

Si $c(x) = 1 + x + x^4 + x^6$, entonces el polinomio mensaje correspondiente es $c(x)/g(x) = a(x) = 1 + x^3$, que corresponde con el mensaje $a = 1001$.

Ejercicios.

20. Sea $g(x) = 1 + x^2 + x^3$ el polinomio generador de un código cíclico lineal de largo 7.

    a) Codifique los siguientes mensajes polinomiales: $1 + x^3$, x, $x + x^2 + x^3$.

    b) Halle el polinomio mensaje correspondiente a las palabras código $c(x)$: $x^2 + x^4 + x^5, 1 + x + x^2 + x^4, x^2 + x^3 + x^4 + x^5$.

21. Halle base y matriz generadora para el código cíclico lineal de largo n con polinomio generador $g(x)$:

    a) $n = 7, g(x) = 1 + x^2 + x^3$

    b) $n = 9, g(x) = 1 + x^3 + x^6$

    c) $n = 15, g(x) = 1 + x + x^4$

    d) $n = 15, g(x) = 1 + x^4 + x^6 + x^7 + x^8$

Un polinomio $f(x) \in \mathbb{K}[x]$ de grado al menos 1 es *irreducible* si no es el producto de dos polinomios sobre $\mathbb{K}[x]$, ambos de grado al menos 1. No es fácil hallar los factores irreducibles de $1 + x^n$ (que además proveerían todos los factores de $1 + x^n$).

El factor 1 de $1 + x^n$ tiene grado 0, y por lo tanto genera un código cíclico de dimensión n. Este código cíclico es $\mathbb{K}^n$. También como caso especial definimos el código $\{0\}$, consistente sólo de la palabra nula de largo n, como código cíclico con generador $g(x) = 0 = 1+x^n \mod 1+x^n$.

Denotaremos los códigos cíclicos lineales $\mathbb{K}^n$ y $\{0\}$ como códigos cíclicos *impropios*.

Ejemplo. Para $n = 3$, $1 + x^3 = (1 + x)(1 + x + x^2)$ es la factorización de $1 + x^3$ en factores irreducibles. Luego, existen dos códigos cíclicos propios de largo 3. Uno tiene generador $g(x) = 1 + x$ y matriz generadora

$$G = \begin{bmatrix} 110 \\ 011 \end{bmatrix}.$$

Este código es $C = \{000, 110, 011, 101\}$. El otro código tiene generador $g(x) = 1 + x + x^2$ y matriz generadora $G = [111]$, de modo que éste es $C = \{000, 111\}$.

Ejemplo. Para $n = 6$, factorizamos $1 + x^6$ en factores irreducibles:

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

Para hallar los generadores de los códigos cíclicos lineales propios de largo 6, formamos todos los productos posibles de estos factores, excepto 1 y $1 + x^6$. Cada tal producto es el generador de un código cíclico lineal de largo 6. Estos productos y las dimensiones de los códigos cíclicos lineales de largo 6 que ellos generan, son como sigue:

Generador	Dimensión
$1 + x$	5
$(1 + x)^2 = 1 + x^2$	4
$1 + x + x^2$	4
$(1 + x + x^2)^2 = 1 + x^2 + x^4$	2
$(1 + x)(1 + x + x^2) = 1 + x^3$	3
$(1 + x)^2(1 + x + x^2) = 1 + x + x^3 + x^4$	2
$(1 + x)(1 + x + x^2)^2 = 1 + x + x^2 + x^3 + x^4 + x^5$	1

Teorema 34. *Si $n = 2^r s$, entonces $1 + x^n = (1 + x^s)^{2^r}$.*

Demostración. Si $n = 2s$, entonces $(1 + x^s)^2 = 1 + x^s + x^s + x^{2s} = 1 + x^{2s}$. Luego, se procede por inducción. □

Corolario 35. *Sea $n = 2^r s$, donde s es impar, y sea $1+x^s$ el producto de z factores irreducibles. Entonces, existen $(2^r + 1)^z$ códigos cíclicos lineales de largo n y $(2^r + 1)^z - 2$ códigos cíclicos lineales propios de largo n.*

Ejemplo. En el penúltimo ejemplo, se mostró que $1 + x^3$ es el producto de dos polinomios irreducibles: $1 + x$ y $1 + x + x^2$. Aplicando el Corolario 35 con $r = 0$ y $z = 2$, hallamos que existen $(2^0 + 1)^2 = 4$ códigos cíclicos lineales de largo 3, dos de los cuales son propios, como fue visto en aquel ejemplo. Ahora bien, para $1 + x^6$, tenemos $n = 6 = 2^1 3$, de modo que $r = 1 \cdot z$ es todavía 2. Luego, existen $(2 + 1)^2 = 9$ códigos cíclicos lineales de largo 6, siete de los cuales son propios, como se ve en el ejemplo previo a este.

Ejercicios.

25. Halle el número de códigos cíclicos lineales de largo $n = 4, 5, 7, 14, 15, 56, 120, 1024$.

26. Halle los polinomios generadores para todos los códigos cíclicos lineales de largo n, donde $n = 4, 5$.

27. Halle dos generadores de grado 4 para un código cíclico lineal de largo 7.

28. Halle un polinomio generador y una matriz generadora para el código cíclico lineal de orden n y dimensión k, donde

    (a) $n = 12, k = 5$ (b) $n = 12, k = 7$

    (c) $n = 14, k = 5$ (d) $n = 14, k = 6$ (e) $n = 14, k = 8$.

Podemos hallar todos los códigos cíclicos, o equivalentemente factorizar $1 + x^n$ por medio de un procedimiento sencillo. Durante esta discusión, supondremos que n es impar.

El primer paso es generar todos los polinomios $I(x) \pmod{1+x^n}$ tales que $I(x) = I^2(x) \pmod{1+ x^n}$. Estos polinomios son llamados polinomios *idempotentes*. Es fácil ver que si $u(x)$ y $v(x)$ son idempotentes, también lo son su suma $u(x)+v(x)$ y su producto $u(x)v(x) \pmod{1+x^n}$. Luego, necesitamos construir solo el conjunto *básico* de polinomios idempotentes. Para construirlos, precisamos particionar $\mathbf{Z}_n = \{0, 1, \ldots, n-1\}$ en *clases*.

Sea $C_i = \{s = 2^j \dot{i} \pmod{n} | j = 0, 1, \ldots, r-1\}$, donde $1 = 2^r \mod n$.

Ejemplo. Para $n = 7$, tenemos

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

Para $n = 9$, tenemos

$$C_0 = \{0\}, C_1 = \{1, 2, 4, 8, 7, 5\} \text{ y } C_3 = \{3, 6\}.$$

Continuando, para cada clase C_i diferente formamos el polinomio

$$c_i(x) = \sum_{j \in C_i} x^j.$$

Mostraremos que $c_i(x)$ es un idempotente y notemos, al pasar, que cualquier idempotente $I(x) \pmod{1 + x^n}$ se expresa como

$$I(x) = \sum_{i=0}^{k} a_i c_i(x), \text{ donde } a_i \in \{0, 1\}.$$

Para ver que $c_i(x)$ es idempotente, ponemos:

$$c_i(x)^2 = c_i(x^2) = \sum_{j \in C_i} x^{2j} = \sum_{k \in C_i} x^k \pmod{1 + x^n},$$

y esto vale, pues si j está en C_i entonces también lo está $2j \pmod n$.

Ejemplo. Para $n = 7$, tenemos

$$\begin{array}{lll} C_0 = \{0\}, & \text{de modo que} & c_0(x) = x^0 = 1, \\ C_1 = \{1, 2, 4\}, & \text{de modo que} & c_1(x) = x^1 + x^2 + x^4, \\ C_3 = \{3, 5, 6\}, & \text{de modo que} & c_3(x) = x^3 + x^5 + x^6. \end{array}$$

Luego, cualquier polinomio idempotente módulo $1 + x^7$ puede expresarse como

$$I(x) = a_0c_0(x) + a_1c_1(x) + a_3c_3(x), \text{ donde } a_i \in \{0, 1\}.$$

Por lo tanto, tenemos 2^3 idempotentes diferentes módulo $1+x^7$, (pero $I(x) = 0$ es trivialmente idempotente).

La conexión entre idempotentes y códigos cíclicos es la siguiente:

Teorema 36. *Todo código cíclico contiene un único polinomio idempotente que lo genera.*

Demostración. Sea $g(x)$ el generador de un código cíclico C de largo n y sea $g(x)h(x) = 1+x^n$, (donde n es impar). Entonces, $\text{mcd}(h(x), g(x)) = 1$, y por el Algoritmo de Euclides, (ver Apéndice B), existen polinomios $t(x)$ y $s(x)$ tales que

$$1 = t(x)g(x) + s(x)h(x). \tag{4.1}$$

Multiplicando ambos lados por $t(x)g(x)$ da

$$t(x)g(x) = (t(x)g(x))^2 + t(x)s(x)(1 + x^n)$$

o sea

$$t(x)g(x) = (t(x)g(x))^2 \quad \text{mód } 1 + x^n.$$

Luego, $t(x)g(x)$ es un idempotente y

$$g(x) = \text{mcd}(t(x)g(x), 1 + x^n).$$

Multiplicando ambos lados de la relación (1) por $g(x)$ se obtiene que $g(x)$ es a su vez un múltiplo módulo $1+x^n$ de $t(x)g(x)$. Por lo tanto, el idempotente $t(x)g(x)$ también es un generador de C. □

Ejemplo. Para hallar todos los códigos cíclicos de largo 9, hallamos sencillamente los polinomios idempotentes y luego los polinomios generadores correspondientes. Como

$$C_0 = \{0\}, C_1 = \{1, 2, 4, 8, 7, 5\}, C_3 = \{3, 6\},$$

tenemos

$$c_0(x) = 1, c_1(x) = x + x^2 + x^4 + x^5 + x^7 + x^8, c_3(x) = x^3 + x^6,$$

y

$$I(x) = a_0c_0(x) + a_1c_1(x) + a_3c_3(x).$$

Idempotente $I(x)$	Generador $g(x) = \text{mcd}(I(x), 1 + x^9)$
1	1
$x + x^2 + x^4 + x^5 + x^7 + x^8$	$1 + x + x^3 + x^4 + x^6 + x^7$
$x^3 + x^6$	$1 + x^3$
$1 + x + x^2 + x^4 + x^5 + x^7 + x^8$	$1 + x + x^2$
$1 + x^3 + x^6$	$1 + x^3 + x^6$
$x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$	$1 + x$
$1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$	Es el mismo polinomio

Ejercicios.

29. Halle todos los idempotentes módulo $1+x^n$ y los polinomios generadores correspondientes para $n = 5$, 7, 11, 15, 31.

4.5. Códigos cíclicos duales

Otro hecho útil acerca de los códigos cíclicos es que sus códigos duales también son cíclicos. De hecho, daremos un procedimiento para construir el polinomio generador de un código dual.

Que el dual de un código cíclico es cíclico sigue directamente del hecho de que si $a \cdot b = 0$, entonces $\pi(a)\cdot\pi(b) = 0$, (donde π es el traslado cíclico), como lo muestra el siguiente argumento. (Nótese que $a \cdot b = a_0b_0 + a_1b_1 + \cdots + a_{n-1}b_{n-1}$ y $\pi(a) \cdot \pi(b) = a_1b_1 + a_2b_2 + \cdots a_{n-1}b_{n-1} + a_0b_0 = a \cdot b = 0$).

Consideremos el código cíclico C generado por la palabra v. Luego $C = \langle\{v, \pi(v), \pi^2(v), \ldots, \pi^{n-1}(v)\}\rangle$. Si $u \in C^\perp$, entonces $\pi^i(v) \cdot u = 0$, para $i = 0, 1, \ldots, n-1$. Sin embargo, esto implica que

$$\pi^{i+1}(v) \cdot \pi(u) = 0,$$

y luego $\pi(u)$ es ortogonal a

$$\langle\{\pi(v), \pi^2(v), \ldots, \pi^n(v)\}\rangle = C,$$

pues $\pi^n(v) = v$. Como $u \in C^\perp$ implica que $\pi(u) \in C^\perp$, concluimos que $C^\perp$ es cíclico.

Para hallar el generador del código dual, precisamos relacionar el producto de polinomios con el producto de vectores.

Lema 37. *Si $a \leftrightarrow a(x)$, $b \leftrightarrow b(x)$ y $b' \leftrightarrow b'(x) = x^n b(x^{-1}) \mod 1 + x^n$, entonces $a(x)b(x) \mod 1 + x^n = 0$ si y sólo si $\pi^k(a)b' = 0$, para $k = 0, 1, \ldots, n-1$.*

Demostración. Sea $c(x) = a(x)b'(x) \mod 1 + x^n$. Entonces, el coeficiente de x^k en $c(x)$ es

$$c_k = a_kb_0 + a_{k+1}b_{n-1} + \cdots + a_{n-1}b_{k+1} + a_0b_k + \cdots + a_{k-1}b_1,$$

pues $x^k \equiv x^{n+k} \pmod{1 + x^n}$. Nótese que si $a = (a_0, a_1, \ldots, a_{n-1})$ y $b = (b_0, b_1, \ldots, b_{n-1})$. entonces $b' = (b_{n-1}, \ldots, b_1, b_0)$ y así, $c_k = \pi^k(a) \cdot b'$. Luego, $c_k = 0$, para $k = 0, 1, \ldots, n-1$, si y sólo si $c(x) = 0 = a(x)b'(x) \mod 1 + x^n$. □

Nuevamente, sea C un código cíclico lineal de largo n y sea $g(x)$ su polinomio generador. Sabemos que $g(x)$ divide $1 + x^n$, y que luego existe un único polinomio $h(x)$ tal que $1 + x^n = g(x)h(x)$. Por el Lema 37, sabemos que $x^kh(x^{-1})$ está en $C^\perp$, pero queremos hallar el generador para $C^\perp$.

Teorema 38. *Si C es un código cíclico lineal de largo n y dimensión k con generador $g(x)$, y si $1 + x^n = g(x)h(x)$, entonces $C^\perp$ es un código cíclico de dimensión $n - k$ con generador $x^k h(x^{-1})$.*

Demostración. Como C tiene dimensión k, entonces $g(x)$ tiene grado $n - k$ y $h(x)$ tiene grado k. Como $g(x)h(x) = 1 + x^n$, tenemos que $g(x^{-1})h(x^{-1}) = 1 + (x^{-1})^n$ y

$$\begin{aligned} x^n g(x^{-1})h(x^{-1}) &= x^n(1 + x^{-n}) \\ x^{n-k} g(x^{-1}) x^k h(x^{-1}) &= 1 + x^n. \end{aligned}$$

Luego, $x^k h(x^{-1})$ es un factor de $1 + x^n$ de grado k, y por lo tanto es el polinomio generador del código cíclico lineal $C^\perp$ de dimensión $n - k$ que contiene a $x^k h(x^{-1})$. □

Ejemplo. El polinomio $g(x) = 1 + x + x^3$ es el generador de un código cíclico de largo 7 y dimensión $7 - 3 = 4$. Como $g(x)$ es un factor de $1 + x^7$, podemos hallar $h(x)$ por división larga de $1 + x^7 = g(x)h(x)$. En este caso, $h(x) = 1 + x + x^2 + x^4$. El generador para $C^\perp$ es $g^\perp(x) = x^4(1+x^{-1}+x^{-2}+x^{-4}) = 1+x^2+x^3+x^4$, que corresponde a $1011100 = w$. Claramente, $g \cdot w = 1101000 \cdot 1011100 = 0$, y también $\pi^k(g) \cdot w = 0$. Nótese que $g^\perp(x) \neq h(x)$.

Ejemplo. Sea $g(x) = 1 + x + x^2$ el generador de un código cíclico lineal de largo 6. Hallamos que $h(x) = 1+x+x^3+x^4$ satisface $g(x)h(x) = 1+x^6$. Por lo tanto, $g^\perp(x) = x^4 h(x^{-1}) = x^4(1+x^{-1}+x^{-3}+x^{-4}) = x^4+x^3+x+1$ es el generador del código dual. Nótese en este ejemplo que $g^\perp(x) = h(x)$.

Ejercicios.

30. Halle el polinomio generador del dual del código cíclico de largo n con polinomio generador $g(x)$, donde:

    a) $n = 6, g(x) = 1 + x^2$

    b) $n = 6, g(x) = 1 + x^3$;

    c) $n = 8, g(x) = 1 + x^2$

    d) $n = 9, g(x) = 1 + x^3 + x^6$

    e) $n = 15, g(x) = 1 + x + x^4$

    f) $n = 15, g(x) = 1 + x^4 + x^6 + x^7 + x^8$

    g) $n = 23, g(x) = 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11}$

    h) $n = 7, g(x) = 1 + x + x^2 + x^4$.

CAPÍTULO 5
De cómo aparecen los cuerpos finitos

En este capítulo, consideramos una clase especial de códigos cíclicos, tomando un enfoque diferente que utiliza cuerpos finitos $\mathbb{F}(2^r)$ para decodificarlos.

5.1. Cuerpos finitos

Recordemos que un polinomio no nulo $d(x)$ es un divisor o factor del polinomio $f(x)$ si $f(x) = g(x)d(x)$, para algún $g(x) \in K[x]$. Por supuesto, 1 y $f(x)$ son siempre divisores de $f(x)$, pero, como tales, son triviales. Decimos que cualquier otro divisor es divisor no trivial o *propio* de $f(x)$, y que un polinomio $f(x) \in K[x]$ es *irreducible* sobre K si no tiene divisores propios en $K[x]$. Caso contrario, decimos que es *reducible* (o factorizable) sobre K.

Ejemplo. Los polinomios x y $1 + x$ son irreducibles por definición; $1 + x + x^2$ no tiene ni a x ni a $1 + x$ como factores; luego, también es irreducible. Sin embargo, x^2 y $x + x^2$ no son irreducibles, pues ambos tienen a x como factor; Además, $1 + x^2$ tiene a $1 + x$ como factor.

En general, $1 + x$ es un factor de $f(x)$ si y solo si 1 es una raíz de $f(x)$, o sea si y solo si $f(1) = 0$. Nótese que $1 + x$ es un factor de $f(x) = 1 + x^2$ y que $f(1) = 1 + 1 = 0$. Del mismo modo, x es un factor de $g(x)$ si y solo si $g(0) = 0$. Sin embargo, hallar factores irreducibles de un polinomio no es fácil y, por ahora, una cuestión de ensayo y error.

Ejemplo. Si $f(x) = 1 + x + x^2 + x^3$, entonces $f(1) = 1 + 1 + 1 + 1 = 0$, y así $1 + x$ es un factor de $f(x)$. Por división larga, $f(x) = (1 + x)(1 + x^2)$. Por otra parte, si $g(x) = 1 + x + x^3$, entonces $g(0) = 1 \neq 0$ y $g(1) = 1 \neq 0$, de modo que $g(x)$ no tiene factores lineales. Por lo tanto, $g(x)$ es irreducible sobre K, pues si un polinomio cúbico es reducible entonces debe tener un factor lineal.

Ejemplo. Sea $f(x) = 1 + x + x^4$. Como $f(0) \neq 0$ y $f(1) \neq 0$, $f(x)$ no tiene factores lineales. Así, si $f(x)$ fuese reducible, entonces $f(x)$ debería tener un factor cuadrático. El único polinomio cuadrático irreducible sobre K es $g(x) = 1 + x + x^2$. Luego de dividir $g(x)$ por $f(x)$, no hallamos ningún resto no nulo. De modo que $1 + x + x^2$ no es factor de $f(x)$. Por lo tanto, $f(x)$ es irreducible sobre K.

Ejercicios.

1. Determine si cada uno de los siguientes polinomios es irreducible sobre K:

$$\begin{array}{ll} \text{(a) } f(x) = 1 + x^2 + x^4 & \text{(b) } f(x) = 1 + x^8 \\ \text{(c) } f(x) = 1 + x^2 + x^3 + x^5 & \text{(d) } f(x) = 1 + x^2 + x^6 \\ \text{(e) } f(x) = 1 + x^4 + x^5 & \text{(f) } f(x) = 1 + x + x^3 + x^7 \end{array}$$

2. Halle todos los polinomios irreducibles de grados 3 y 4 sobre K.

3. Halle todos los polinomios irreducibles de grado 5 sobre K.

Decimos que un polinomio irreducible sobre K de grado $n > 1$ es *primitivo* si no es divisor de $1 + x^m$ para ningún $m < 2^n - 1$. Veremos que un polinomio irreducible de grado n siempre divide $1 + x^m$ cuando $m = 2^n - 1$.

Ejemplo. Como $1 + x + x^2$ no es factor de $1 + x^m$ para ningún $m < 3 = 2^2 - 1$, entonces es primitivo. De modo similar, $1+x+x^3$ no es factor de $1+x^m$ para ningún $m < 7 = 2^3 - 1$, y luego, también es primitivo.

Sin embargo, $1 + x^5 = (1 + x)(1 + x + x^2 + x^3 + x^4)$, y $1 + x + x^2 + x^3 + x^4$ es irreducible (ver Ejercicio 2), pero $5 < 15 = 2^4 - 1$, y luego $1 + x + x^2 + x^3 + x^4$ no es primitivo.

Recordemos que podemos definir adición y multiplicación de polinomios módulo un polinomio $h(x)$ de grado n. Sea $K^n[x]$ el conjunto de todos los polinomios en $K[x]$ que tienen grado menor que n. Por supuesto que cada palabra de K^n corresponde a un polinomio de $K^n[x]$, de modo que podemos, en efecto, definir adición y multiplicación de palabras en K^n.

En este capítulo, introducimos una estructura adicional, la de cuerpo finito, para asistir en la construcción y decodificación de códigos. Ya habíamos definido la adición y la multiplicación de palabras en K^n, pero para que K^n sea un cuerpo, precisamos ser cuidadosos en nuestra elección de $h(x)$. Por ejemplo, en un cuerpo debe acontecer que si $ab = 0$, entonces $a = 0$ ó $b = 0$.

Ejemplo. Tratemos de usar multiplicación de polinomios módulo $1 + x^n$ para definir multiplicación de palabras en K^4:

$$\begin{aligned} 0101 \cdot 0101 &\leftrightarrow (x + x^3)(x + x^3) \\ &= x^2 + x^6 \\ &= (x^2 + x^2) \pmod{1 + x^4} \\ &= 0 \\ &\leftrightarrow 0000, \end{aligned}$$

de modo que $0101 \cdot 0101 = 0000$, pero $0101 \neq 0000$ en K^4. Luego, K^4 no puede ser un cuerpo bajo esta definición de multiplicación.

La dificultad en este último ejemplo está en que $1 + x^4$ no es irreducible sobre K. La forma de definir multiplicación para hacer de K^n un cuerpo es *definir multiplicación en K^n módulo un polinomio irreducible de grado n*. La prueba de este hecho puede hallarse en un curso de álgebra moderna.

Ejemplo. Definamos multiplicación en K^4 usando el polinomio irreducible $h(x) = 1+x+x^4$. Para hallar el producto $1101 \cdot 0101$, notemos que

$$1101 \cdot 0101 \leftrightarrow (1 + x + x^3)(x + x^3).$$

Pero $(1+x+x^3)(x+x^3) = x+x^2+x^3+x^6$, y

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

Luego, $1101 \cdot 0101 = 0100 \leftrightarrow x$.

Ejercicios.

4. Defina multiplicación en K^4 módulo $h(x) = 1+x+x^4$ y calcule los siguientes productos:

   (a) $0011 \cdot 1011$ (b) $0100 \cdot 0010$
   (c) $1110 \cdot 1001$ (d) $1100 \cdot 0111$
   (e) $1010 \cdot 0110$ (f) $1111 \cdot 0011$.

5. Halle todos los productos de elementos de $\mathbb{F}(2^k)$ usando $1+x+x^2$ para definir multiplicación, (o sea, haga una tabla de multiplicación).

Ejemplo. Consideremos la construcción de $\mathbb{F}(2^3)$ usando el polinomio primitivo $h(x) = 1+x+x^3$ para definir multiplicación. Hacemos esto computando $x^i \mod h(x)$:

Palabra	$\leftrightarrow$	$x^i \mod h(x)$
100	$\leftrightarrow$	1
010	$\leftrightarrow$	x
001	$\leftrightarrow$	x^2
110	$\leftrightarrow$	$x^3 \equiv 1+x$
011	$\leftrightarrow$	$x^4 \equiv x+x^2$
111	$\leftrightarrow$	$x^5 \equiv 1+x+x^2$
101	$\leftrightarrow$	$x^6 \equiv 1+x^2$.

Para computar $110 \cdot 001 \leftrightarrow (1+x)x^2$ notar por la tabla que tenemos $1+x = x^3 \mod h(x)$, y así:

$$\begin{aligned} x^2(1+x) &\equiv x^2 \cdot x^3 \\ &\equiv x^5 \\ &\equiv 1+x+x^2 \pmod{h(x)}. \end{aligned}$$

Luego, $110 \cdot 001 = 111$.

El uso de un polinomio primitivo para construir $\mathbb{F}(2^r)$ hace la computación en el cuerpo mucho más fácil que con el uso de un polinomio irreducible no primitivo. Para verlo, sea $\beta \in K^n$ la palabra correspondiente a $x \mod h(x)$, donde $h(x)$ es un polinomio primitivo de grado n. Entonces $\beta^i \leftrightarrow x^i \mod h(x)$. Nótese que $1 = x^m \mod h(x)$ significa que $0 = 1+x^m \mod h(x)$, y así $h(x)$ dividiría $1+x^m$ para $m < 2^n-1$. Como $h(x)$ es primitivo, sabemos que $h(x)$ no divide $1+x^m$ para $m < 2^n-1$, y así $\beta^m \neq 1$ para $m < 2^n-1$. Como $\beta^j = \beta^i$ para $j \neq i$ si y sólo si $\beta^i = \beta^{j-i}\beta^i$, o sea $\beta^{j-i} = 1$, concluimos que

$$K^n \setminus \{0\} = \{\beta^i \mid i = 0, 1, \ldots, 2^n-2\},$$

lo cual quiere decir que cada palabra no nula en K^n puede ser representada por alguna potencia de β. Esta es la propiedad que hace más fácil la multiplicación en el cuerpo.

Un elemento $\alpha \in \mathbb{F}(2^r)$ es *primitivo* si $\alpha^m \neq 1$ para $1 \leq m < 2^r-1$. Equivalentemente, α es *primitivo* si cada palabra no nula en $\mathbb{F}(2^r)$ puede ser expresada como una potencia de α. Vemos

Palabra	Polinomio en x mód $h(x)$	Potencia de β
0000	0	–
1000	1	β^0
0100	x	β^1
0010	x^2	β^2
0001	x^3	β^3
1100	$1+x \equiv x^4$	β^4
0110	$x+x^2 \equiv x^5$	β^5
0011	$x^2+x^3 \equiv x^6$	β^6
1101	$1+x+x^3 \equiv x^7$	β^7
1010	$1+x^2 \equiv x^8$	β^8
0101	$x+x^3 \equiv x^9$	β^9
1110	$1+x+x^2 \equiv x^{10}$	β^{10}
0111	$x+x^2+x^3 \equiv x^{11}$	β^{11}
1111	$1+x+x^2+x^3 \equiv x^{12}$	β^{12}
1011	$1+x^2+x^3 \equiv x^{13}$	β^{13}
1001	$1+x^3 \equiv x^{14}$	β^{14}

Cuadro 5.1.: **Tabla III**

que por el párrafo anterior, si usamos un polinomio primitivo para construir $\mathbb{F}(2^r)$, entonces β es un elemento primitivo.

Ejemplo. Construyamos $\mathbb{F}(2^4)$ usando el polinomio primitivo

$$h(x) = 1 + x + x^4.$$

Escribamos cada vector como una potencia de $\beta \leftrightarrow x \mod h(x)$ de acuerdo con la siguiente tabla, y notando que $\beta^{15} = 1$:

Por ejemplo, $0110 \cdot 1101 \leftrightarrow \beta^5 \cdot \beta^7 = \beta^{12} \leftrightarrow 1111$, pues

$$(x + x^2)(1 + x + x^2) \equiv x^5 \cdot x^7 \equiv x^{12} \pmod{h(x)}.$$

Ejercicios.

6. Use $\mathbb{F}(2^4)$, construido en la Tabla III, para computar los productos en K^4 del Ejercicio 4.

7. Construya los siguientes cuerpos como en el último ejemplo (Tabla III):

   a) Construya $\mathbb{F}(2^2)$ usando $h(x) = 1 + x + x^2$.

   b) Construya $\mathbb{F}(2^3)$ usando $h(x) = 1 + x^2 + x^3$.

   c) Construya $\mathbb{F}(2^4)$ usando $h(x) = 1 + x^3 + x^4$.

   d) Construya $\mathbb{F}(2^5)$ usando $h(x) = 1 + x^2 + x^5$.

8. Muestre que si $h(x) \in K[x]$ es un polinomio irreducible de grado n, entonces $h(x)$ divide $1 + x^m$ para algún $m \leq 2^n - 1$.

9. Halle todos los elementos primitivos en $\mathbb{F}(2^4)$, (ver Tabla III).

10. Muestre que β^i es primitivo si y solo si $\text{mcd}(i, 2^r - 1) = 1$.

5.2. Polinomios primitivos

Recordemos que un elemento α en un cuerpo $F = \mathbb{F}(2^r)$ es *raíz* de un polinomio $p(x) \in F[x]$ si y solo si $p(\alpha) = 0$. En ese caso, si $p(x) = a_0 + a_1 x + \cdots + a_k x^k$, entonces

$$p(\alpha) = a_0 + a_1\alpha + \cdots + a_k\alpha^k = 0.$$

Ejemplo. Sea $p(x) = 1 + x^3 + x^4$, y sea β un el elemento primitivo en $\mathbb{F}(2^4)$ construido usando $h(x) = 1 + x + x^4$, (ver Tabla III). Tenemos

$$\begin{aligned} p(\beta) &= 1 + \beta^3 + \beta^4 = 1000 + 0001 + 1100 \\ &= 0101 \\ &= \beta^9, \end{aligned}$$

de modo que β no es raíz de $p(x)$. Sin embargo

$$\begin{aligned} p(\beta^7) &= 1 + (\beta^7)^3 + (\beta^7)^4 \\ &= 1 + \beta^{21} + \beta^{28} \\ &= 1 + \beta^6 + \beta^{13} \quad (\text{pues } \beta^{15} = 1) \\ &\leftrightarrow 1000 + 0011 + 1011 = 0000 \\ &\leftrightarrow 0. \end{aligned}$$

Como $p(\beta^7) = 0$, tenemos que β^7 es raíz de $p(x)$. Nótese que usamos la convención de que $1 \leftrightarrow 1000$ tanto como el hecho de que $\beta^{15} = 1$. Luego, $\beta^{21} = \beta^{15}\beta^6 = 1 \cdot \beta^6 = \beta^6$ y $\beta^{28} = \beta^{15}\beta^{13} = 1 \cdot \beta^{13} = \beta^{13}$.

En general, el *orden* de un elemento no nulo α en $\mathbb{F}(2^r)$ es el menor entero positivo m tal que $\alpha^m = 1$. Sabemos que todo α no nulo en $\mathbb{F}(2^r)$ tiene orden $m \leq 2^r - 1$. En particular, $\alpha \in \mathbb{F}(2^r)$ es un elemento primitivo si tiene orden $2^r - 1$.

Para cualquier elemento $\alpha \in \mathbb{F}(2^r)$ definimos el *polinomio minimal* de α como el polinomio en $K[x]$ de menor grado que tenga a α como raíz. Denotamos tal polinomio con $m_\alpha(x)$. Nótese que si α tiene orden m, (o sea, $\alpha^m = 1$), entonces α es una raíz de $1 + x^m$, de modo que todo elemento de $\mathbb{F}(2^r)$ es raíz de algún polinomio en $K[x]$.

Para hallar el polinomio minimal de un elemento de $\mathbb{F}(2^r)$, nos servirá saber algunos hechos relacionados con polinomios minimales.

Teorema 39. *Sea $0 \neq \alpha \in \mathbb{F}(2^r)$. Sea $m_\alpha(x)$ el polinomio minimal de α. Entonces,*

(a) $m_\alpha(x)$ es irreducible sobre K,

(b) si $f(x)$ es un polinomio sobre K tal que $f(\alpha) = 0$, entonces $m_\alpha(x)$ es un factor de $f(x)$,

(c) el polinomio minimal $m_\alpha(x)$ es único, y

(d) el polinomio minimal $m_\alpha(x)$ es un factor de $1 + x^{2^r-1}$.

Demostración. (a) Si $m_\alpha(x) = g(x)h(x)$, entonces $m_\alpha(\alpha) = 0$ implica que $g(\alpha)h(\alpha) = 0$. Luego, o bien $g(\alpha) = 0$ o bien $h(\alpha) = 0$. Como $m_\alpha(x)$ es el polinomio de menor grado tal que $m_\alpha(\alpha) = 0$, entonces o bien $g(x) = 1$ o bien $h(x) = 1$. Por lo tanto, $m_\alpha(x)$ es irreducible sobre K.

(b) Por el Algoritmo de División,

$$f(x) = m_\alpha(x)q(x) + r(x),$$

donde $r(x) = 0$ ó $\mathrm{gr}(r(x)) < \mathrm{gr}(m_\alpha(x))$. Como $f(\alpha) = 0$ y

$$f(\alpha) = m_\alpha(\alpha)q(\alpha) + r(\alpha) = 0 \cdot q(\alpha) + r(\alpha),$$

tenemos que $r(\alpha) = 0$. Por la minimalidad del grado de $m_\alpha(x)$, se tiene que $r(x) = 0$. Por lo tanto, $f(x) = m_\alpha(x)q(x)$, y $m_\alpha(x)$ es un factor de $f(x)$.

(c) Si $m'(x)$ es también un polinomio de menor grado tal que $m'(\alpha) = 0$, entonces por el ítem (b), $m_\alpha(x)$ es un factor de $m'(x)$ y $m'(x)$ es un factor de $m_\alpha(x)$. Por lo tanto, $m_\alpha(x) = m'_\alpha(x)$, de modo que el polinomio minimal es único.

(d) Sea β un elemento primitivo de $\mathbb{F}(2^r)$ y $\alpha = \beta^i$. Entonces, $\alpha^{2^r-1} = (\beta^i)^{2^r-1} = (\beta^{2^r-1})^i = 1^i = 1$. Luego, α es raíz de $1 + x^{2^r-1}$, y por (b), $m_\alpha(x)$ es un factor de $1 + x^{2^r-1}$. □

El hallazgo del polinomio minimal de un elemento $\alpha \in \mathbb{F}(2^r)$ se reduce a hallar una combinación lineal de los vectores $1, \alpha, \alpha^2, \ldots, \alpha^r$ que sume 0. Como cualquier conjunto de $r + 1$ vectores en K^r es dependiente, sabemos que tal combinación existe.

Una vez que tenemos construido $\mathbb{F}(2^r)$ usando un polinomio primitivo, es conveniente denotar $m_\alpha(x)$ como $m_i(x)$, donde $\alpha = \beta^i$. Introducimos esta notación en el siguiente ejemplo.

Ejemplo. Hallemos el polinomio minimal de $\alpha = \beta^3$ en $\mathbb{F}(2^4)$, construido usando $h(x) = 1+x+x^4$, (ver Tabla III). Si $m_\alpha(x) = m_3(x) = a_0+a_1x+a_2x^2+a_3x^3+a_4x^4$, entonces debemos hallar los valores de a_0, a_1, a_2, a_3, a_4. Nótese que

$$\begin{aligned} 0 = m_\alpha(\alpha) &= a_0 1 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3 + a_4\alpha^4 \\ &= a_0\beta^0 + a_1\beta^3 + a_2\beta^6 + a_3\beta^9 + a_4\beta^{12}, \\ \text{y así, } 0000 &= a_0 1000 + a_1 0001 + a_2 0011 + a_3 0101 + a_4 1111. \end{aligned}$$

Resolviendo esta ecuación para a_0, a_1, a_2, a_3, a_4, hallamos que

$$\begin{aligned} &a_0 = a_1 = a_2 = a_3 = a_4 = 1 \quad \text{y} \\ &m_\alpha(x) = 1 + x + x^2 + x^3 + x^4. \end{aligned}$$

El conjunto de raíces sucesivas de $m_\alpha(x)$ es

$$\{\alpha, \alpha^2, \alpha^3, \alpha^4\} = \{\beta^3, \beta^6, \beta^9, \beta^{12}\}$$

y entonces $m_3(x) = m_6(x) = m_9(x) = m_{12}(x)$, (donde $m_i(x)$ denota el polinomio minimal de β^i).

Elemento de $\mathbb{F}(2^4)$	Polinomio minimal
0	x
1	$1+x$
$\beta, \beta^2, \beta^4, \beta^8$	$1+x+x^4$
$\beta^3, \beta^6, \beta^9, \beta^{12}$	$1+x+x^2+x^3+x^4$
β^5, β^{10}	$1+x+x^2$
$\beta^7, \beta^{11}, \beta^{13}, \beta^{14}$	$1+x^3+x^4$

Cuadro 5.2.: **Tabla IV**

Tenemos otros hechos útiles para la búsqueda de todos los polinomios minimales de elementos de $\mathbb{F}(2^r)$. Recordemos que $f(x)^2 = f(x^2)$, de modo que

$$\left(\sum_{i=0}^{n} a_i x^i\right)^2 = \sum_{i=0}^{n} a_i^2 (x^i)^2 = \sum_{i=0}^{n} a_i (x^2)^i.$$

Esto proviene de $(a+b)^2 = a^2 + b^2$ y de $a_i^2 = a_i$, pues $a_i \in \{0, 1\}$.

Luego, si $f(\alpha) = 0$, tenemos que $f(\alpha^2) = (f(\alpha))^2$, y así, α^2 también es una raíz de $f(x)$. En forma similar, tenemos que $f(\alpha^4) = (f(\alpha^2))^2 = 0$, etc., y de ahí que si α es una raíz de $f(x)$, también lo son $\alpha^2, \alpha^4, \ldots, \alpha^{2^i}$, etc. Con un poco más de esfuerzo se puede probar lo siguiente.

Teorema 40. *Si α es un elemento en $\mathbb{F}(2^r)$ con polinomio minimal $m_\alpha(x)$, entonces $\{\alpha, \alpha^2, \alpha^4, \ldots, \alpha^{2^{r-1}}\}$ es el conjunto de todas las raíces de $m_\alpha(x)$. En particular, el grado de $m_\alpha(x)$ es $|\{\alpha, \alpha^2, \ldots, \alpha^{2^{r-1}}\}|$.*

Ejemplo. Sea $m_5(x)$ el polinomio minimal de $\alpha = \beta^5 \in \mathbb{F}(2^4)$, (ver Tabla III). Como $\{\alpha, \alpha^2, \alpha^4, \alpha^8\} = \{\beta^5, \beta^{10}\}$ por el Teorema 40, las raíces de $m_5(x)$ son β^5 y β^{10}, lo que quiere decir que $\mathrm{gr}(m_5(x)) = 2$. Luego, $m_5(x) = a_0 + a_1 x + a_2 x^2$, de donde

$$\begin{aligned} 0 &= a_0 + a_1\beta^5 + a_2\beta^{10} \\ &\leftrightarrow a_0 1000 + a_1 0110 + a_2 1110. \end{aligned}$$

Luego, $a_0 = a_1 = a_2 = 1$ y $m_5(x) = 1 + x + x^2$.

Del mismo modo, podemos hallar los polinomios minimales del resto de los elementos del cuerpo $\mathbb{F}(2^4)$ construido usando $1 + x + x^4$. El resultado se halla en la tabla IV.

Ejercicios.

11. Verifique las entradas en la Tabla IV.
12. Halle el polinomio minimal de cada elemento de $\mathbb{F}(2^3)$ construido usando $p(x) = 1+x+x^3$.
13. Halle el polinomio minimal de cada elemento de $\mathbb{F}(2^4)$ construido usando $p(x) = 1+x^3+x^4$.
14. Halle el polinomio minimal de cada elemento de $\mathbb{F}(2^5)$ construido usando $p(x) = 1+x^2+x^5$.
15. Muestre que $1 + x + x^2 = (\beta^5 + x)(\beta^{10} + x)$, usando la Tabla III.
16. Muestre que $m_\alpha(x)$ es un polinomio primitivo si y solo si α es un elemento primitivo.

5.3. Códigos cíclicos de Hamming

Ya sabemos que los códigos de Hamming tienen la importante ventaja de ser 1-perfectos, o sea, que corrigen todos los patrones de error de peso 1, y que por lo tanto, tienen un esquema de decodificación bien simple. En esta sección mostraremos que existe un código cíclico de Hamming de largo $n = 2^r - 1$ para cada $r \geq 2$. Este código tiene la ventaja adicional, común a todos los códigos cíclicos, de ser de fácil decodificación.

La matriz de control de paridad de un código de Hamming de largo $n = 2^r - 1$ tiene como filas las $2^r - 1$ palabras no nulas de largo r. Si β es un elemento primitivo de $\mathbb{F}(2^r)$ entonces por definición las potencias de β son todas distintas. Por lo tanto, podemos construir un código de Hamming de largo $n = 2^r - 1$ que tiene

$$H = \begin{bmatrix} 1 \\ \beta \\ \beta^2 \\ \vdots \\ \beta^{2^r-2} \end{bmatrix}$$

como matriz de control de paridad. Nótese que H es una matriz $(2^r - 1) \times r$. Además, para cualquier palabra recibida $w = w_0 w_1 \cdots w_{n-1}$ tenemos que $wH = w_0\beta^0 + w_1\beta + \cdots + w_{n-1}\beta^{n-1} \leftrightarrow w(\beta)$, de modo que w es una palabra-código si y sólo si β es una raíz de $w(x)$. Por lo tanto, por el Teorema 39(b), $m_\beta(x)$ divide cada palabra-código, y claramente es una palabra-código ella misma, de modo que este código es cíclico y $m_\beta(x)$ es su polinomio generador. Obtuvimos el siguiente resultado.

Teorema 41. *Un polinomio primitivo de grado r es polinomio generador de un código cíclico de Hamming de largo $2^r - 1$.*

Ejemplo. Sea $r = 3$, de modo que $n = 2^3 - 1 = 7$. Usemos $p(x) = 1 + x + x^3$ para construir $\mathbb{F}(2^3)$, y $\beta \leftrightarrow 010$ como elemento primitivo. Recordemos que $\beta^i \leftrightarrow x^i \mod p(x)$. Por lo tanto, una matriz de control de paridad para un código de Hamming es

$$\begin{bmatrix} 1 \\ \beta \\ \beta^2 \\ \beta^3 \\ \beta^4 \\ \beta^5 \\ \beta^6 \end{bmatrix} \leftrightarrow \begin{bmatrix} 100 \\ 010 \\ 001 \\ 110 \\ 011 \\ 111 \\ 101 \end{bmatrix} = H$$

que es igual a la matriz de control de paridad del código cíclico con polinomio generador $p(x) = m_\beta(x)$.

Decodificar el código cíclico de Hamming es fácil. Si el generador es el polinomio primitivo $m_\alpha(x)$ y si $w(x)$ es recibido, entonces $w(x) = c(x) + e(x)$, donde $c(x)$ es una palabra-código y $w(\alpha) = e(\alpha)$. Pero como e tiene peso 1, sabemos que $e(\alpha) = \alpha^j$ (donde j es la posición del 1 en e, al rotular las posiciones de e con $0, 1, \ldots, n-1$). Por lo tanto, el más factible polinomio de error es $e(x) = x^j$, y así, $c(x) = w(x) + x^j$.

Ejemplo. Supongamos que $\mathbb{F}(2^3)$ fue construido usando $1+x+x^3$. Luego, $m_1(x) = 1+x+x^3$

es el generador para un código cíclico de Hamming de largo 7. Supongamos que $w(x) = 1 + x^2 + x^3 + x^6$ es recibido. Luego,

$$\begin{aligned} w(\beta) &= 1 + \beta^2 + \beta^3 + \beta^6 \\ &\leftrightarrow 100 + 001 + 110 + 101 \\ &= 110 \\ &\leftrightarrow \beta^3. \end{aligned}$$

y así, $e(x) = x^3$ y $c(x) = w(x) + x^3 = 1 + x^2 + x^6$.

Ejercicios.

17. Halle una matriz de control de paridad para un código cíclico de Hamming de largo 7 usando $\mathbb{F}(2^3)$ construido con $p(x) = 1 + x + x^3$, donde el polinomio generador es $m_3(x)$. Si $w(x) = x + x^2 + x^4$ es recibido, halle la más factible palabra-código $c(x)$.

18. Repita el Ejercicio 17 usando $\mathbb{F}(2^3)$ construido con $p(x) = 1+x^2+x^3$ y polinomio generador $m_1(x)$.

19. Repita el Ejercicio 17 usando $\mathbb{F}(2^3)$ construido con $p(x) = 1+x^2+x^3$ y polinomio generador $m_3(x)$.

20. Construya una matriz de control de paridad para un código cíclico de Hamming de largo 15.

21. Halle un polinomio generador para un código cíclico de Hamming C de largo 15 que tenga raíces $1, \beta^7, \beta^5 \in \mathbb{F}(2^4)$ (construido usando $1 + x + x^4$). Construya una matriz de control de paridad para C. Muestre que $c(x) \in C$ si y sólo si $w(c)$ es par.

22. Muestre que cada palabra de un código cíclico tiene peso par si y sólo si $1 + x$ es un factor del polinomio generador.

Vale la pena notar que resultados más generales se siguen de las observaciones discutidas en esta sección. Sea C un código cíclico de largo n con polinomio generador $g(x)$. Supongamos que $\alpha \in \mathbb{F}(2^r)$ es una raíz de $g(x)$. Entonces para todo $c(x) \in C$ vale que $c(\alpha) = 0$, y por el Teorema 39(b) vale que m_α es divisor de $c(x)$. Siempre podemos expresar $g(x)$ como el producto de polinomios minimales de elementos de $\mathbb{F}(2^r)$. Podemos usar esto para construir una matriz de control de paridad y hallar un algoritmo de decodificación para C. El caso en el que $g(x) = m_\beta(x)m_{\beta^3}(x)$ es discutido en la próxima sección.

5.4. Códigos BCH

Una clase importante de códigos correctores de múltiples errores son los códigos de Bose-Chaudhuri-Hocquenghem, o códigos BCH. La construcción y los procedimientos de decodificación para códigos BCH generales serán desarrollados más tarde. En primer lugar, construiremos y decodificaremos un ejemplo importante de la clase; de hecho, la familia de códigos 2-correctores.

Los códigos BCH son importantes por dos razones. Primero, ellos admiten un esquema de decodificación relativamente sencillo, y segundo, la clase de códigos BCH es bastante extensa. De hecho, para cualquier dos enteros positivos r y t con $t \leq 2^{r-1} - 1$, existe un código BCH de largo $n = 2^r - 1$ que es t-corrector y que tiene dimensión $k \geq n - rt$.

El código BCH 2-corrector de largo $2^r - 1$ es el código cíclico lineal generado por $g(x) = m_\beta(x)m_{\beta^3}(x)$, donde β es un elemento primitivo en $\mathbb{F}(2^r)$ y $r \geq 4$. Como $n = 2^r - 1$ y $g(x)$ divide $1 + x^n$ (por el Teorema 39(d)), $g(x)$ es el polinomio generador de un código cíclico.

Ejemplo. Sea β un elemento primitivo en $\mathbb{F}(2^4)$ construido con $p(x) = 1 + x + x^4$, (ver Tabla III). Tenemos que $m_1(x) = 1 + x + x^4$ y $m_3(x) = 1 + x + x^2 + x^3 + x^4$. Por lo tanto,

$$g(x) = m_1(x)m_3(x) = 1 + x^4 + x^6 + x^7 + x^8$$

es el generador para un código BCH 2-corrector de largo 15.

Ejercicios.

23. Los códigos BCH 2-correctores son definidos para $r \geq 4$. ¿Qué código genera $g(x) = m_1(x)m_3(x)$ cuando $r = 3$?

24. Sea β un elemento primitivo de $\mathbb{F}(2^4)$ construido usando el polinomio irreducible $p(x) = 1 + x^3 + x^4$. Halle el polinomio generador $g(x)$ del código BCH 2-corrector de largo 15, usando esta representación de $\mathbb{F}(2^4)$, o sea, halle $g(x) = m_1(x)m_3(x)$, (ver el Ejercicio 7).

25. Halle un polinomio generador para un código BCH 2-corrector de largo 31 construyendo $\mathbb{F}(2^5)$ con el polinomio irreducible $1 + x^2 + x^5$, (ver el Ejercicio 7).

Lema 42. *La siguiente matriz H es una matriz de control de paridad para el código BCH 2-corrector de largo $2^r - 1$, donde β es un elemento primitivo en $\mathbb{F}(2^r)$, y el polinomio generador es $g(x) = m_1(x)m_3(x)$:*

$$H = \begin{bmatrix} \beta^0 & \beta^0 \\ \beta^1 & \beta^3 \\ \beta^2 & \beta^6 \\ \vdots & \vdots \\ \beta^i & \beta^{3i} \\ \vdots & \vdots \\ \beta^{2^r-2} & \beta^{3(2^r-2)} \end{bmatrix}.$$

Como β^i y β^{3i} son elementos de $\mathbb{F}(2^r)$, representan palabras de largo r, de modo que H es una matriz $(2^r - 1) \times (2r)$. Además, como $\mathrm{gr}(m_1(x)) = r = \mathrm{gr}(m_3(x))$, entonces el grado de $g(x) = m_1(x)m_3(x)$ es $2r$ y luego el código tiene dimensión $n - 2r = 2^r - 1 - 2r$. (Dejamos la demostración de que $m_3(x)$ tiene grado r como ejercicio).

Por ejemplo, usamos $\mathbb{F}(2^4)$, construido en la Tabla III con el polinomio primitivo $p(x) = 1 + x + x^4$, para construir un código BCH 2-corrector C_{15}. Definimos C_{15} como el código lineal con la matriz 15×8 de control de paridad H y polinomio generador $m_1(x)m_3(x)$, de acuerdo con la siguiente tabla, que muestra la matriz de control de paridad de C_{15}:

Teorema 43. *Para cualquier entero $r \geq 4$, existe un código BCH 2-corrector de largo $n = 2^r - 1$, dimensión $k = 2^r - 2r - 1$ y distancia $d = 5$ que tiene polinomio generador $m_1(x)m_3(x)$.*

$$\begin{bmatrix} 1 & 1 \\ \beta^1 & \beta^3 \\ \beta^2 & \beta^6 \\ \beta^3 & \beta^9 \\ \beta^4 & \beta^{12} \\ \beta^5 & 1 \\ \beta^6 & \beta^3 \\ \beta^7 & \beta^6 \\ \beta^8 & \beta^9 \\ \beta^9 & \beta^{12} \\ \beta^{10} & 1 \\ \beta^{11} & \beta^3 \\ \beta^{12} & \beta^6 \\ \beta^{13} & \beta^9 \\ \beta^{14} & \beta^{12} \end{bmatrix} \leftrightarrow \begin{bmatrix} 1000 & 1000 \\ 0100 & 0001 \\ 0010 & 0011 \\ 0001 & 0101 \\ 1100 & 1111 \\ 0110 & 1000 \\ 0011 & 0001 \\ 1101 & 0011 \\ 1010 & 0101 \\ 0101 & 1111 \\ 1110 & 1000 \\ 0111 & 0001 \\ 1111 & 0011 \\ 1011 & 0101 \\ 1001 & 1111 \end{bmatrix} = H$$

Cuadro 5.3.: **Tabla V**

Para probar que la distancia es 5, mostraremos que el código puede corregir dos errores. Luego, tiene distancia al menos 5. Por definición de matriz de control de paridad, se ve que $n = 2^r - 1$, y como $m_1(x)$ y $m_3(x)$ tienen grado r cada uno, el grado de $g(x)$ es $\text{gr}(g(x)) = n - k = 2r$, y así, $k = 2^r - 2r - 1$.

Ejercicios.

26. Muestre que las columnas de la matriz de control de paridad de C_{15} en la Tabla V forman un conjunto linealmente independiente, y de ello, que C_{15} tiene dimensión $k = 7$.

27. Muestre que $d = 5$ para C_{15} usando la matriz de control de paridad.

28. Muestre que si β es un elemento primitivo de $\mathbb{F}(2^r)$, con $r > 2$, entonces $|\{\beta^{2^i} | 0 \leq i \leq r - 1\}| = r$ y $|\{(\beta^3)^{2^i} | 0 \leq i \leq r - 1\}| = r$. Y que por lo tanto, ambos $m_1(x)$ y $m_3(x)$ tienen grado r.

29. Determine si cada una de las siguientes palabras de largo 15 es un palabra de C_{15}, donde $g(x) = 1 + x^4 + x^6 + x^7 + x^8$:

    (a) 011001011000010 (b) 000111010000110
    (c) 011100000010110 (d) 111111111111111

5.5. Decodificando un código BCH 2-corrector

Describimos un esquema de decodificación para el código BCH 2-corrector construido en la Sección 35. A lo largo de la presente sección, identificaremos una palabra binaria de largo r con su correspondiente potencia de β.

Una matriz de control de paridad para el código BCH $(2^r - 1, 2^r - 2r - 1, 5)$-lineal 2-corrector con generador $g(x) = m_1(x)m_3(x)$ está dada por la matriz H definida en el Lema 42.

Supongamos que la palabra w sea recibida y que $w \leftrightarrow w(x)$. Entonces el síndrome de w es

$$wH = [w(\beta), w(\beta^3)] = [s_1, s_3],$$

donde s_1 y s_3 son palabras de largo r.
Si no ocurren errores en la transmisión, entonces el síndrome es $wH = 0$, de modo que $s_1 = s_3 = 0$.
Si solo ocurre un error en la transmisión, entonces el polinomio de error es $e(x) = x^i$, por lo que $wH = eH = [e(\beta), e(\beta^3)] = [\beta^i, \beta^{3i}] = [s_1, s_3]$. Por lo tanto, $s_1^3 = s_3$.

Si ocurren dos errores en la transmisión, digamos en las posiciones i y j, con $i \neq j$, entonces $e(x) = x^i + x^j$ y $wH = eH = [e(\beta), e(\beta^3)] = [s_1, s_3]$. Luego, el síndrome wH está dado por

$$wH = [s_1, s_3] = [\beta^i + \beta^j, \beta^{3i} + \beta^{3j}].$$

Comparando las entradas de esta última igualdad, resulta el sistema de ecuaciones:

$$\begin{aligned} \beta^i + \beta^j &= s_1 \\ \beta^{3i} + \beta^{3j} &= s_3. \end{aligned}$$

Ahora bien, tenemos la factorización

$$(\beta^i + \beta^j)(\beta^{2i} + \beta^{i+j} + \beta^{2j}) = \beta^{3i} + \beta^{3j}.$$

y además

$$s_1^2 = (\beta^i + \beta^j)^2 = \beta^{2i} + \beta^{2j}.$$

Por lo tanto,

$$\begin{aligned} s_3 &= \beta^{3i} + \beta^{3j} \\ &= (\beta^i + \beta^j)(\beta^{2i} + \beta^{i+j} + \beta^{2j}) \\ &= s_1(s_1^2 + \beta^{i+j}). \end{aligned}$$

Luego,

$$\frac{s_3}{s_1} + s_1^2 = \beta^{i+j}.$$

Ahora bien, β^i y β^j son raíces de la ecuación cuadrática

$$x^2 + (\beta^i + \beta^j)x + \beta^{i+j} = 0,$$

o sea, que son raíces de

$$x^2 + s_1 x + \left(\frac{s_3}{s_1} + s_1^2\right) = 0.$$

Por lo tanto, podemos hallar las posiciones de los errores hallando las soluciones de esta ecuación. El polinomio de la izquierda en esta ecuación es llamado *polinomio localizador de errores*.

Ejemplo. Sea $w \leftrightarrow w(x)$ una palabra recibida con síndromes $s_1 = 0111 = w(\beta)$ y $s_3 = 1010 = w(\beta^3)$, donde w fué codificada usando C_{15}. De la Tabla III de la página 113 tenemos que $s_1 \leftrightarrow \beta^{11}$ y que $s_3 \leftrightarrow \beta^8$. Luego,

$$\begin{aligned} \tfrac{s_3}{s_1} + s_1^2 &= \beta^8\beta^{-11} + \beta^{22} \\ &= \beta^{12} + \beta^7 \\ &= \beta^2. \end{aligned}$$

Formamos el polinomio $x^2+\beta^{11}x+\beta^2$ y hallamos que tiene raíces β^4 y β^{13}. Por lo tanto, podemos decidir que los errores más factibles ocurrieron en las posiciones 4 y 13 (o sea, $e(x) = x^4 + x^{13}$), de modo que el patrón de error más factible es

$$000010000000010.$$

Ejercicios.

30. Verifique por sustitución que β^4 y β^{13} son de hecho las soluciones de la ecuación cuadrática $x^2 + \beta^{11}x + \beta^2 = 0$. Además, verifique que la suma de las filas cuarta y décimotercera de H en la Tabla III es $[s_1, s_3]$.

31. Halle las raíces en $\mathbb{F}(2^4)$ de los siguientes polinomios, si fuese posible, usando la Tabla III:

(a) $x^2 + \beta^4 x + \beta^{13}$	(b) $x^2 + \beta^7 x + \beta^2$
(c) $x^2 + \beta^2 x + \beta^5$	(d) $x^2 + \beta^6$
(e) $x^2 + \beta^2 x$	(f) $x^2 + x + \beta^8$.

Hemos llegado a un esquema para DMVI de un código BCH 2-corrector. Sea w una palabra recibida. Claramente, una vez que se determina un patrón de error, el algoritmo termina.

Algoritmo D. (Decodificación de códigos BCH 2-correctores) Supongamos que el polinomio generador es $m_1(x)m_3(x)$.

1. Calcular el síndrome $wH = [s_1, s_3] = [w(\beta), w(\beta^3)]$.
2. Si $s_1 = s_3 = 0$, concluir que no ocurrieron errores. Decodificar $c = w$ como la palabra-código enviada.
3. Si $s_1 = 0$ y $s_3 \neq 0$, entonces pedir retransmisión.
4. Si $s_1^3 = s_3$, entonces corregir un error singular en la posición i, donde $s_1 = \beta^i$.
5. Formar la ecuación cuadrática
$$x^2 + s_1 x + \frac{s_3}{s_1} + s_1^2 = 0.$$
6. Si esta ecuación tiene dos raíces distintas β^i y β^j, corregir errores en las posiciones i y j.
7. Si la ecuación no tiene dos raíces distintas en $\mathbb{F}(2^r)$, concluir que al menos tres errores han ocurrido en la transmisión y solicitar una retransmisión.

Ejemplo. Supongamos que w es recibida y que el síndrome es $wH = 01111010 \leftrightarrow [\beta^{11}, \beta^8]$. Ahora bien,

$$s_1^3 = (\beta^{11})^3 = \beta^{33} = \beta^3 \neq \beta^8 = s_3.$$

En este caso, la ecuación del ítem (5) del Algoritmo D es $x^2 + \beta^{11}x + \beta^2 = 0$, como fue mostrado en el ejemplo anterior. Esta ecuación tiene raíces β^4 y β^{13}, de modo que corregimos errores en las posiciones $i = 4$ y $j = 13$; en otras palabras, el más factible patrón de error es
$u = 000010000000010$, y $e(x) = x^4 + x^{13}$ es el polinomio de error.

Ejemplo. Supongamos que el síndrome es $wH = [w(\beta), w(\beta^3)] = [\beta^3, \beta^9]$. Luego, $s_1^3 = (\beta^3)^3 = \beta^9 = s_3$. Por lo tanto, es más factible que un error singular haya ocurrido en la posición $i = 3$. El más factible patrón de error es $u = 000100000000000$, y $e(x) = x^3$ es el polinomio de error.

Ejemplo. Supongamos que $w = 110111101011000$ sea recibida. El síndrome es

$$wH = 01110110 \leftrightarrow [\beta^{11}, \beta^5] = [s_1, s_3].$$

Ahora bien, $s_1^3 = (\beta^{11})^3 = \beta^{33} = \beta^3 \neq s_3 = \beta^5$. Para formar la ecuación cuadrática del ítem (5) del Algoritmo D, primero computamos

$$\begin{aligned} \tfrac{s_3}{s_1} + s_1^2 &= \beta^5\beta^{-11} + (\beta^{11})^2 \\ &= \beta^9 + \beta^7 \\ &\leftrightarrow 0101 + 1101 \\ &= 1000 \\ &\leftrightarrow \beta^0. \end{aligned}$$

Así, en este caso la ecuación toma la forma

$$x^2 + \beta^{11}x + \beta^0 = 0.$$

Sustituyendo los elementos sucesivos de $\mathbb{F}(2^4)$ en la ecuación, llegamos a probar que $x = \beta^7$ es una raíz de la misma, y hallamos:

$$\begin{aligned} (\beta^7)^2 + \beta^{11}\beta^7 + \beta^0 &= \beta^{14} + \beta^3 + \beta^0 \\ &\leftrightarrow 1001 + 0001 + 1000 \\ &= 0000. \end{aligned}$$

Ahora bien, $\beta^7\beta^j = 1 = \beta^{15}$, de modo que $\beta^j = \beta^8$ es la otra raíz. Por lo tanto, corregimos errores en las posiciones $i = 7$ y $j = 8$; o sea $u = 0000000110000000$ es el más factible patrón de error. Decodificamos $v = w + u = 110111110011000$ como la palabra enviada.

Ejemplo. Supongamos que una palabra de C_{15} es enviada y que ocurren errores en las posiciones 2, 6 y 12. Entonces, el síndrome wH es la suma de las filas 2, 6 y 12 de H, donde w es la palabra enviada. Luego,

$$\begin{aligned} wH &= 00100011 + 00110001 + 11110011 \\ &= 11100001 \leftrightarrow [\beta^{10}, \beta^3] = [s_1, s_3]. \end{aligned}$$

Ahora bien, $s_1^3 = (\beta^{10})^3 = \beta^{30} = 1 \neq \beta^3 = s^3$. Calculamos:

$$\tfrac{s_3}{s_1} + s_1^2 = \beta^3\beta^{-10} + \beta^{20} = \beta^8 + \beta^5 \leftrightarrow 1010 + 0110 = 1100 \leftrightarrow \beta^4$$

y luego formamos la ecuación cuadrática

$$x^2 + \beta^{10}x + \beta^4 = 0.$$

Sustituyendo los elementos de $\mathbb{F}(2^4)$ en la ecuación, vemos que esta no tiene raíces en $\mathbb{F}(2^4)$. Por lo tanto, la DMVI para C_{15} concluye correctamente que al menos tres errores han ocurrido y requerimos una retransmisión.

Ejercicios.

32. Supongamos que codificamos mensajes usando C_{15}. Si es posible, determine la localización de los errores, si w es recibida y el síndrome wH es dado como en los casos siguientes:

(a) 0100 0101 (b) 0000 0100
(c) 1110 1000 (d) 1010 0100
(e) 1100 1101 (f) 0011 1101
(g) 0100 0000 (h) 0000 0000

33. Usemos el código C_{15}. Si es posible, decodifique cada una de las siguientes palabra recibidas:

(a) 11000 00000 00000 (b) 10101 00101 10001
(c) 00001 00001 00001 (d) 01000 01000 00001
(e) 01000 10101 00000 (f) 01010 10010 11000
(g) 11001 11001 11000 (h) 11011 10100 01100
(i) 11001 11001 00000 (j) 10111 00000 01000
(k) 11100 00000 00001 (l) 11100 10110 00000
(m) 10111 00000 00000 (n) 00011 10100 00110

CAPÍTULO 6
Usando códigos sobre cuerpos finitos

Entre los códigos más prácticos que se conocen, se hallan los códigos de Reed-Solomon, o códigos RS, que trataremos en este capítulo. Estos códigos son usados actualmente por la NASA, la Agencia Espacial Europea, etc. Además, los códigos usados en los discos compactos provienen de esta familia de códigos.

6.1. Códigos sobre $\mathbb{F}(2^r)$

En el capítulo V habíamos estudiado extensivamente los códigos BCH 2-correctores binarios. De hecho, los códigos de Reed-Solomon son también códigos BCH, pero los dígitos en cada palabra-código ya no son dígitos binarios. Esto puede parecer extraño, pues hemos terminado alabando los usos prácticos de aquellos códigos, y las transmisiones son siempre en canales binarios. Como mostraremos, estos nuevos códigos tienen una representación binaria, pero no es así como los estaremos visualizando en un comienzo.

Antes de continuar, desarrollamos algo de notación. Sea $\mathbb{F}(2^r)[x]$ el conjunto de todos los polinomios con coeficientes sobre $\mathbb{F}(2^r)$. Este conjunto contiene a $K[x]$, el conjunto de todos los polinomios con coeficientes binarios, donde $K = \mathbb{F}(2) = \{0, 1\}$. Como anteriormente, podemos identificar palabras c de un código lineal C de largo n sobre $\mathbb{F}(2^r)$ con polinomios $c(x) \in \mathbb{F}(2^r)[x]$ que tengan grado $\text{gr}(c(x)) < n$.

Recordemos que habíamos definido códigos cíclicos de largo n en términos de las raíces de los polinomios correspondientes. Por ejemplo, palabras del código BCH 2-corrector binario de largo $n = 2^r - 1$ pueden ser descritas por medio de $c(x) \in C_K$ si y sólo si $\beta^1, \beta^2, \beta^3, \beta^4$ son raíces de $c(x)$, donde $c(x) \in K[x]$, $\text{gr}(c(x)) < n$ y β es un elemento primitivo en $\mathbb{F}(2^r)$. En este caso, $g_{_K}(x) = m_1(x)m_3(x)$ es el polinomio generador de tal código cíclico, y $c(x) \in C_K$ si y sólo si $c(x) = a(x)g_{_K}(x)$.

Podemos generalizar esta observación a códigos sobre $\mathbb{F}(2^r)$ eligiendo el polinomio $c(x)$ en $\mathbb{F}(2^r)[x]$. Nuevamente, $c(x) \in C$ si y sólo si $\beta^1, \beta^2, \beta^3, \beta^4$ son raíces de $c(x)$. Ahora sin embargo, los polinomios $x + \beta, x + \beta^2, x + \beta^3$ y $x + \beta^4$ están en$\mathbb{F}(2^r)[x]$, y por lo tanto $c(x) \in C$ si y sólo si $g(x) = (\beta + x)(\beta^2 + x)(\beta^3 + x)(\beta^4 + x)$ divide $c(x)$.

El código binario C_K definido arriba es un código BCH. El código C sobre $\mathbb{F}(2^r)$ recién definido contiene C_K como subcódigo y es un ejemplo de código de Reed-Solomon. En general, el código C_K es llamado *subcódigo subcuerpo* de C pues $C_K \subseteq C$ y cada palabra en C_K tiene todos sus dígitos en el subcuerpo K de $\mathbb{F}(2^r)$; o sea, $C_K = C \cap K^n$.

Estos dos códigos, C_Ky C, son cíclicos, pues $c(x) \in C$ implica que $c(x) = xc(x)$ mód $(1+x^n)$ está en C. Esta implicación se sigue del Algoritmo de División y del hecho de que β^i es una

raíz tanto de $1+x^n$ como de $xc(x)$. De hecho, no es difícil mostrar que si $g(x)$ genera un código cíclico lineal de largo 2^r-1 sobre $\mathbb{F}(2^r)$, entonces el generador del subcuerpo subcódigo es el polinomio $g_{\kappa}(x)$ que tiene sus raíces formando el menor conjunto R que satisfaga:

(a) si α es raíz de $g(x)$, entonces $\alpha \in R$, y

(b) si $\alpha \in R$, entonces $\alpha^2 \in R$.

Al juntar estas observaciones, nos da el siguiente resultado:

Teorema 44. *Sean $\alpha_1, \alpha_2, \ldots, \alpha_k$ elementos no nulos y distintos en $\mathbb{F}(2^r)$. Entonces $g(x) = (\alpha_1 + x)(\alpha_2 + x)\cdots(\alpha_k + x)$ genera un código cíclico lineal de largo $2^r - 1$ sobre $\mathbb{F}(2^r)$.*

Ejemplo. Supongamos que $F = \mathbb{F}(2^4)$ fué construido usando $1+x+x^4$; (ver Tabla III, página 113). El polinomio $g(x) = (\beta + x)(\beta^2 + x) = \beta^3 + \beta^5 x + x^2$ genera un código cíclico lineal sobre F de largo 15. Luego, la palabra-código correspondiente a $g(x)$ es $\beta^3\beta^5 1000000000000$.

Además, $g_{\kappa}(x) = 1 + x + x^4 \leftrightarrow 110010000000000$ está en este código, y de hecho genera un subcódigo subcuerpo cíclico binario. Para ver esto usando la notación arriba, hallamos R: $\beta, \beta^2 \in R$ por (a); $(\beta^2)^2 = \beta^4 \in R$ y $(\beta^4)^2 = \beta^8 \in R$ por (b); de modo que $R = \{\beta, \beta^2, \beta^4, \beta^8\}$. Luego $g_{\kappa}(x) = (x + \beta^4)(x + \beta^8)g(x)$.

Resumimos algunos resultados básicos respecto de códigos cíclicos sobre $\mathbb{F}(2^r)$:

Teorema 45. *Sea C un código cíclico lineal de largo n sobre $\mathbb{F}(2^r)$. Entonces cada palabra $c(x) \in C$ puede ser escrita unívocamente como $a(x)g(x)$, donde $g(x)|1 + x^n$, para algún $a(x) \in \mathbb{F}(2^r)[x]$ de grado menor que $n - \mathrm{gr}(g(x))$. Además, $g(x)$ divide $f(x)$ si y sólo si $f(x) \in C$.*

Corolario 46. *Sea $g(x)$ de grado $n-k$. Si $g(x)$ genera un código cíclico lineal C sobre $\mathbb{F}(2^r)$ de largo $n = 2^r - 1$ y dimensión k, entonces*

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}$$

es una matriz generadora para C, y el número de palabras de C es $(2^r)^k$.

El hecho de que $|C| = 2^{rk}$ sigue del Teorema 45, pues todos los polinomios $a(x)$ en $\mathbb{F}(2^r)[x]$ de grado menor que k dan palabras diferentes $a(x)g(x)$; pero existen 2^{rk} tales polinomios $a(x)$, pues cada uno de los k coeficientes en $a(x)$ puede ser cualquier un de los 2^r elementos del cuerpo.

Ejemplo. Consideremos $\mathbb{F}(2^3)$ construido usando $1 + x + x^3$ con β como elemento primitivo. Sea $g(x) = (\beta + x)(\beta^2 + x) = \beta^3 + \beta^4 x + x^2$. Entonces, $g(x)$ genera un código cíclico lineal C sobre $\mathbb{F}(2^3)$ de largo 7. Una matriz generadora para C es

$$G = \begin{bmatrix} \beta^3 & \beta^4 & 1 & 0 & 0 & 0 & 0 \\ 0 & \beta^3 & \beta^4 & 1 & 0 & 0 & 0 \\ 0 & 0 & \beta^3 & \beta^4 & 1 & 0 & 0 \\ 0 & 0 & 0 & \beta^3 & \beta^4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \beta^3 & \beta^4 & 1 \end{bmatrix}.$$

El código C tiene 8^5 palabras. La palabra de C correspondiente a $a(x) = 1 + \beta x + \beta^3 x^4 \leftrightarrow 1\beta 00\beta^3 = m$, por ejemplo, es $a(x)g(x) \leftrightarrow mG = \beta^3 0 \beta^4 \beta \beta^6 1 \beta^3$.

Ejercicios.

1. Construya $\mathbb{F}(2^3)$ usando $1+x+x^3$. Sea $g(x)=(1+x)(\beta+x)$ generador de un código C de largo 7 sobre $\mathbb{F}(2^3)$.
   a) ¿Cuántas palabras tiene C?
   b) Construya una matriz generadora G para C usando el Corolario 46.
   c) Codifique los siguientes mensajes usando G:
      1) $a(x)=1+\beta^6 x$
      2) $a(x)=\beta^4 x^4$
      3) $a(x)=1+x+x^2$
   d) Halle el polinomio generador del subcódigo subcuerpo binario.

2. Construya $\mathbb{F}(2^4)$ usando $1+x+x^4$. Sea $g(x)=(\beta+x)(\beta^2+x)(\beta^3+x)(\beta^4+1)$ generador de un código cíclico lineal C sobre $\mathbb{F}(2^4)$.
   a) ¿Cuántas palabras tiene C?
   b) Construya una matriz generadora G para C usando el Corolario 46.
   c) Codifique los siguientes mensajes usando G:
      1) $a(x)=1+\beta^7 x^{10}$
      2) $a(x)=\beta^2 x+x^2$
      3) $a(x)=1+x+x^2$
   d) Halle el polinomio generador $g_{_K}(x)$ del subcódigo subcuerpo binario. Halle $a(x)$ tal que $g_{_K}(x)=a(x)g(x)$.

6.2. Códigos de Reed-Solomon

En la última sección fueron introducidos generadores para códigos cíclicos lineales sobre $\mathbb{F}(2^r)$, pero no fueron dadas pistas sobre cuáles son las capacidades correctivas de estos códigos. Nos dedicaremos aquí a este problema y luego definiremos los códigos RS, o códigos de Reed-Solomon. Consideramos solo RS códigos, pero la mayor parte de los resultados respecto de ellos se aplican directamente a códigos BCH, que son justamente códigos subcuerpos. Comenzamos con un lema técnico.

Lema 47. *(Matriz de Vandermonde) Sean $\alpha_1,\alpha_2,\ldots,\alpha_t$ elementos no nulos de $\mathbb{F}(2^r)$. Entonces,*

$$\det\begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{t-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{t-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha_t & \alpha_t^2 & \cdots & \alpha_t^{t-1} \end{bmatrix} = \prod_{1\le j<i\le t}(\alpha_i+\alpha_j). \tag{6.1}$$

Demostración. Si $\alpha_i=\alpha_j$ para algún $i\neq j$, entonces dos filas de la matriz son idénticas, y así el determinante es nulo. Por lo tanto, para $1\le j<i\le t$ vale que $(\alpha_i+\alpha_j)$ es un factor del determinante, de modo que $\Pi_{1\le j<i\le t}(\alpha_i+\alpha_j)$ divide al determinante. Usando el hecho de que ambos lados de (2) se despliegan en polinomios en $\alpha_1,\alpha_2,\ldots,\alpha_t$ de igual grado, vemos que

estos polinomios difieren en, a lo sumo, un factor común. Este factor común es 1, como se puede ver comparando los coeficientes de $\Pi_{i=1}^{t}\alpha_i^{i-1}$ en ambos lados de (2). □

Ejemplo. Usando el Lema 47 y $\mathbb{F}(2^4)$ construido por medio de $1+x+x^4$ y la Tabla III de la página 113, hallamos que

$$\begin{aligned}\det\begin{bmatrix}1 & \beta^2 & \beta^4\\ 1 & \beta^7 & \beta^{14}\\ 1 & \beta^{10} & \beta^5\end{bmatrix} &= (\beta^7+\beta^2)(\beta^{10}+\beta^2)(\beta^{10}+\beta^7)\\ &= \beta^{12}\cdot\beta^4\cdot\beta^6\\ &= \beta^7.\end{aligned}$$

Ejercicios.

3. Halle los siguientes determinantes usando el Lema 47. Suponga que β es el elemento primitivo en $\mathbb{F}(2^4)$ construido usando $1+x+x^4$, (Tabla III):

$$\text{(a) } \det\begin{bmatrix}1 & \beta & \beta^2\\ 1 & \beta^4 & \beta^8\\ 1 & \beta^7 & \beta^{14}\end{bmatrix} \quad \text{(b) } \det\begin{bmatrix}1 & \beta^2 & \beta^4 & \beta^6\\ 1 & \beta^3 & \beta^6 & \beta^9\\ 1 & \beta^5 & \beta^{10} & 1\\ 1 & \beta^8 & \beta & \beta^9\end{bmatrix} \quad \text{(c) } \det\begin{bmatrix}1 & \beta^3\\ 1 & \beta^7\end{bmatrix}.$$

Ya estamos preparados para el principal teorema referente a códigos BCH generales. Este resultado no es presentado aquí en su forma más general, pero será suficiente para tratar códigos RS.

Teorema 48. *Sea $g(x)=(\beta^{m+1}+x)(\beta^{m+2}+x)\cdots(\beta^{m+\delta-1}+x)$ polinomio generador de un código cíclico lineal C sobre $\mathbb{F}(2^r)$ de largo $n=2^r-1$, donde β es un elemento primitivo en $\mathbb{F}(2^r)$ y m es entero. Entonces, $d(C)\geq\delta$.*

Demostración. Para $1\leq i\leq\delta-1$ tenemos que β^{m+i} es raíz de $g(x)$. Luego, las columnas de

$$H=\begin{bmatrix}1 & 1 & \cdots & 1\\ \beta^{m+1} & \beta^{m+2} & \cdots & \beta^{m+\delta-1}\\ (\beta^{m+1})^2 & (\beta^{m+2})^2 & \cdots & (\beta^{m+\delta-1})^2\\ \vdots & \vdots & & \vdots\\ (\beta^{m+1})^{n-1} & (\beta^{m+2})^{n-1} & \cdots & (\beta^{m+\delta-1})^{n-1}\end{bmatrix}.$$

expanden $C^{\perp}$. Ninguna combinación lineal de $\delta-1$ filas de esta matriz se anula, como puede verse evaluando el determinante de cualquier conjunto de $\delta-1$ filas, digamos

$$\begin{aligned}&\det\begin{bmatrix}(\beta^{m+1})^{j_1} & \cdots & (\beta^{m+\delta-1})^{j_1}\\ (\beta^{m+1})^{j_2} & \cdots & (\beta^{m+\delta-1})^{j_2}\\ \vdots & & \vdots\\ (\beta^{m+1})^{j_{\delta-1}} & \cdots & (\beta^{m+\delta-1})^{j_{\delta-1}}\end{bmatrix}\\ &= \beta^{(m+1)(j_1+j_2+\cdots+j_{\delta-1})}\det\begin{bmatrix}1 & \beta^{j_1} & \cdots & (\beta^{j_1})^{\delta-2}\\ 1 & \beta^{j_2} & \cdots & (\beta^{j_2})^{\delta-2}\\ \vdots & \vdots & & \vdots\\ 1 & \beta^{j_{\delta-1}} & \cdots & (\beta^{j_{\delta-1}})^{\delta-2}\end{bmatrix}\end{aligned}$$

$$= \beta^{(m+1)(j_1+j_2+\cdots+j_{\delta-1})} \prod_{1\le j_y<j_x\le\delta-1} (\beta^{j_x} + \beta^{j_y})$$

que no se anula pues β tiene orden $n = 2^r - 1$ y $1 \le j_1 < j_2 < \ldots < j_{\delta-1} \le n-1$. Por lo tanto, ninguna combinación lineal de $\delta - 1$ o menos filas de H se anula y así, por el Teorema 16 de la página 61, tenemos que $d(C) \ge \delta$. Nótese que las columnas de H forman un conjunto linealmente independiente. Luego, H es una matriz de control de paridad. □

La prueba del Teorema 48 se aplica a cualquier código cíclico binario de largo $2^r - 1$ con un generador que contenga $\beta^{m+1}, \ldots, \beta^{m+\delta-1}$ entre sus raíces. Estos códigos binarios son llamados códigos BCH *primitivos* y δ es llamada la *distancia diseñada* de un tal código. Como estos códigos son subcódigos subcuerpos, $C_K \subseteq C$, de códigos de Reed-Solomon C, a verse a continuación, debemos tener $d(C_K) \ge \delta$ para ellos también.

Un *código de Reed-Solomon* de tipo $RS(2^r, \delta)$ es un código cíclico lineal sobre $\mathbb{F}(2^r)$ con generador $g(x) = (\beta^{m+1} + x)(\beta^{m+2} + x)\cdots(\beta^{m+\delta-1} + x)$, donde m es entero y β es elemento primitivo de $\mathbb{F}(2^r)$.

Teorema 49. *Si C es un código de tipo $RS(2^r, \delta)$, entonces:*

(a) $n = 2^r - 1$,

(b) $k = 2^r - \delta$,

(c) $d = \delta$, *y*

(d) $|C| = 2^{rk}$.

Demostración. El ítem (a) se sigue del Teorema 44. Los ítems (b) y (d) se siguen del Corolario 46. (Nótese que un código lineal sobre $\mathbb{F}(2^r)$ de dimensión k tiene 2^{rk} palabras, lo cual es consistente con el resultado de que un código linear binario, o sea un código sobre $K = \mathbb{F}(2)$ de dimensión k, tiene 2^k palabras). El hecho de que $d \ge \delta$ se sigue del Teorema 48. Por otra parte, la cota de Singleton asegura que $d \le \delta$, (Teorema 21 de la página 75). □

Nótese que como $d = n - k + 1$, entonces los códigos RS son máxima distancia separables, (MDS, ver Teorema 22, página 73.)

Antes de dar otro ejemplo, nótese que cada código C de tipo $RS(2^r, \delta)$ puede ser representado como un código binario simplemente al reemplazar cada dígito en cada palabra-código por la palabra binaria de largo r dada por una tabla de índices de $\mathbb{F}(2^r)$. Este código tiene largo $r(2^r - 1)$, mientras que el subcódigo subcuerpo binario tiene largo $2^r - 1$.

Sea $\hat{c}$ la representación binaria de $c \in C$ formada de esta forma . Sea $\hat{C}$ el código binario resultante. Una de las razones de la utilidad de $\hat{C}$ está en que se comporta bien como código corrector de ráfagas.

Ejemplo. Sea C de tipo $RS(4, 2)$ con $g(x) = \beta + x$, donde $\mathbb{F}(2^2)$ es construido usando $1 + x + x^2$. Por el Teorema 48, C tiene $n = 3, k = 2, d = 2$ y $|C| = 16$. Por el Corolario 46, una matriz generadora para C es

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

De $\mathbb{F}(2^2)$ sabemos que $0, 1, \beta$ y β^2 corresponden a los vectores 00, 10, 01 y 11, respectivamente. Los 16 mensajes u y sus representaciones binarias $\hat{u}$, junto a las palabras correspondientes

$c = uG$ de C y sus representaciones binarias $\hat{c}$, son:

$\hat{u}$	u	$c = uG$	$\hat{c}$	$\hat{u}$	u	$c = uG$	$\hat{c}$
0000	$0\,0$	$0\,0\,0$	000000	0001	$0\,\beta$	$0\,\beta^2\beta$	001101
1000	$1\,0$	$\beta\,1\,0$	011000	1001	$1\,\beta$	$\beta\beta\beta$	010101
0100	$\beta\,0$	$\beta^2\beta\,0$	110100	0101	$\beta\beta$	$\beta^2\,1\,\beta$	111001
1100	$\beta^2\,0$	$1\beta^2\,0$	101100	1101	$\beta^2\beta$	$1\,0\,\beta$	100001
0010	$0\,1$	$0\beta\,1$	000110	0011	$0\,\beta^2$	$0\,1\,\beta^2$	001011
1010	$1\,1$	$\beta\beta^2\,1$	011110	1011	$1\,\beta^2$	$\beta 0\beta^2$	010011
0110	$\beta\,1$	$\beta^2\,0\,1$	110010	0111	$\beta\beta^2$	$\beta^2\beta^2\beta^2$	111111
1110	$\beta^2\,1$	$1\,1\,1$	101010	1111	$\beta^2\beta^2$	$1\,\beta\beta^2$	100111

Ejercicios.

4. Sea C un código de tipo $RS(4,3)$ con generador $g(x) = (1+x)(\beta+x)$.

   a) Halle n, k, d y $|C|$ para este código.

   b) Construya una matriz generadora G para C usando el Corolario 46.

   c) Halle todas las palabras en C, las palabras binarias correspondientes en $\hat{C}$ y los mensajes correspondientes, (por supuesto, codificando los mensajes usando la matriz G pedida en (b)).

5. Sea C un código de tipo $RS(8,5)$ con generador $g(x) = (1+x)(\beta+x)(\beta^2+x)(\beta^3+x)$ usando $\mathbb{F}(2^3)$ construido con $1 + x + x^3$.

   a) Halle n, k, d y $|C|$.

   b) Halle la matriz generadora G de C usando el Corolario 46.

   c) Codifique cada uno de los siguientes mensajes, usando G, a una palabra en C, y luego a una palabra en $\hat{C}$:

   1) $10\beta^2$,
   2) 111,
   3) $\beta^2\beta^4\beta^6$.

6. Usando los cuerpos construidos en el Ejercicio 7 de la página 115, halle polinomios generadores para los códigos de tipo $RS(2^r, \delta)$ con los los siguientes valores de r, δ y m:

   a) $r = 2, \delta = 3, m = 2$,

   b) $r = 3, \delta = 3, m = 2$,

   c) $r = 3, \delta = 5, m = 0$,

   d) $r = 4, \delta = 5, m = 0$,

   e) $r = 5, \delta = 5, m = 0$,

7. Para cada uno de los códigos del ejercicio anterior, halle los valores de n, k, d y $|C|$.

Por el Teorema 48, tenemos que si C es un código de tipo $RS(2^r, \delta)$, entonces $n = 2^r - 1$. A menudo precisamos usar códigos de largos $\neq 2^r - 1$, pero tales códigos pueden ser formados como sigue, a partir de un código de tipo $RS(2^r, \delta)$. Para cualquier entero s tal que $1 \leq s \leq 2^r - \delta$, y para cualquier código C de tipo $RS(2^r, \delta)$, formamos el código $C(s)$ de tipo $RS(2^r, \delta)$ *recortado*, a partir de C, tomando todas las palabras-código en C que tengan ceros en las últimas s posiciones, y luego eliminando, o recortando, sus últimas s posiciones.

Ejemplo. Sea C el código de tipo $RS(4, 2)$ del ejemplo anterior. El código $C(1)$ de tipo $RS(4, 2)$ recortado (con $s = 1$) es formado tomando todas las palabras-código que tienen los últimos $s = 1$ dígitos nulos, o
sea

$$000, \quad \beta 10, \quad \beta^2\beta 0 \quad \text{y} \quad 1\beta^2 0,$$

y luego eliminando las $s = 1$ últimas posiciones. Así:

$$C(1) = \{00, \beta 1, \beta^2\beta, 1\beta^2\}.$$

Alternativamente, usando la representación polinomial para un código de tipo $RS(2^r, \delta)$, el código recortado $C(s)$ queda formado por el conjunto de polinomios en C de grado menor que $n - s = 2^r - 1 - s$. Así, si $g(x)$ es el polinomio generador de C, entonces $C(s)$ es el conjunto de polinomios $c(x) = a(x)g(x)$, donde $\text{gr}(a(x)) < k - s = 2^r - \delta - s$, (pues $\text{gr}(g(x)) = \delta$). Por lo tanto, una matriz generadora $G(s)$ para el código $C(s)$ está dada por

$$G(s) = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-s-1}g(x) \end{bmatrix}.$$

Comparando esta matriz con la matriz generadora G de C dada en el Corolario 46, resulta que $G(s)$ está formada por el truncamiento de las primeras $k - s$ filas de G, a las cuales las últimas columnas les han sido eliminadas, recortadas, o truncadas.

Así, si C es un código de tipo $RS(2^r, \delta)$ con parámetros n, k y d, entonces tenemos que $C(s)$ tiene largo $n(s) = n - s = 2^r - 1 - s$ y dimensión $k(s) = k - s = 2^r - \delta - s$.

Para hallar la distancia $d(s)$ de $C(s)$, notemos que si c_1 y c_2 son palabras de $C(s)$, entonces la distancia entre c_1 y c_2 es igual a la distancia entre las palabras correspondientes $c_1 00 \ldots 0$ y $c_2 00 \ldots 0$ en C. Por lo tanto, $d(C(s)) \geq d(C) = \delta$. Además, la cota de Singleton del Teorema 21 de la página 75 implica que

$$\begin{aligned} d(s) &\leq n(s) - k(s) + 1 \\ &= 2^r - 1 - s - (2^r - \delta - s) + 1 \\ &= \delta. \end{aligned}$$

Así, tenemos que $d(s) = \delta$, y por el Teorema 22 de la página 75, vale que $C(s)$ es también un código MDS. Por lo tanto, obtenemos el siguiente resultado.

Teorema 50. *Sea C un código de tipo $RS(2^r, \delta)$ y sea $C(s)$ el código de tipo $RS(2^r, \delta)$ recortado asociado, con parámetros $n(s)$, $k(s)$ y $d(s)$. Entonces,*

$$\begin{aligned} n(s) &= 2^r - 1 - s, \\ k(s) &= 2^r - \delta - s, \\ d(s) &= \delta, \end{aligned}$$

y $C(s)$ es un código MDS.

Otros códigos de tipo $RS(2^r, \delta)$ pueden ser formados eliminando cualquier conjunto de s coordenadas, en lugar de las últimas s entradas, como hicimos más arriba. Como los códigos de tipo $RS(2^r, \delta)$ son códigos MDS, cualquier código de tipo $RS(2^r, \delta)$ recortado también tendrá las propiedades descritas en el Teorema 50.

Ejemplo. En el último ejemplo de la sección pasada, construimos un código C de tipo $RS(2^3, 3)$ con polinomio generador $g(x) = \beta^3 + \beta^4 x + x^2$. El código $C(2)$ de tipo $RS(2^3, 3)$ recortado correspondiente tiene matriz generadora

$$G(2) \leftrightarrow \begin{bmatrix} \beta^3 \beta^4 1\ 0\ 0 \\ 0\ \beta^3 \beta^4 1\ 0 \\ 0\ 0\ \beta^3 \beta^4 1 \end{bmatrix}$$

y tiene parámetros $n(2) = 5, k(2) = 3$ y $d(2) = 3$. Nótese que $G(2)$ está formada al eliminar, o recortar, las últimas $s = 2$ columnas de la matriz generadora G en aquel ejemplo.

6.3. Decodificación de los códigos RS

Como los dígitos en códigos de tipo $RS(2^r, \delta)$ son elementos de $\mathbb{F}(2^r)$, corregir una palabra recibida tiene que ver no sólo con hallar las ubicaciones de los errores, pero también con las "magnitudes" de los mismos, pues los dígitos del más factible patrón de error pertenecen a $\mathbb{F}(2^r)$. Con esto en mente, damos las siguientes definiciones. Las *ubicaciones de los errores* de una palabra recibida son las coordenadas no nulas del (más factible) patrón de error. Cada ubicación de error es indicada, o apuntada, por un *número de ubicación de error*: si la j-ésima coordenada de la palabra recibida es una ubicación de error, entonces su número de ubicacion de error es β^i; (donde las coordenadas son indicadas con $0, 1, \ldots, n-1$, como lo fueron en los códigos BCH 2-correctores). Por ejemplo, los ítems (4) y (6) del Algoritmo D de la página 124 hallan los números de ubicación de error del más factible patrón de error, cuando se usa un código BCH 2-corrector. La *magnitud de error* de una ubicación de error i es el elemento de $\mathbb{F}(2^r)$ que se halla en la coordenada i del (más factible) patrón de error. Como el código BCH 2-corrector definido en el Capítulo V es código sobre $\mathbb{F}(2)$, todas las magnitudes de error deben valer 1, (el único elemento no nulo de $\mathbb{F}(2)$), y así son completamente determinadas por las ubicaciones de errores. Éste no es el caso de los códigos sobre un cuerpo $\mathbb{F}(2^r)$ con $r > 1$, y así para decodificar los códigos RS precisamos hallar tanto las ubicaciones como las magnitudes de los errores.

Ejemplo. Usando el código de tipo $RS(8, 3)$ construido en el último ejemplo de la Sección 37 (página 129), si $c = \beta^3\beta^4\beta^0 0000$ es transmitida y si $w = \beta^3\beta^4\beta^5 0000$ es recibida, entonces el más factible patrón de error es $c + w = e = 00\beta^4 0000$. Así, el número de ubicación de error es β^2 y la magnitud de error correspondiente es β^4.

Desarrollaremos ahora un algoritmo para decodificar un código de tipo $RS(2^r, \delta)$ (y el correspondiente subcódigo subcuerpo BCH) con generador $g(x) = (\beta^{m+1} + x)(\beta^{m+2} + x) \cdots (\beta^{m+\delta-1} + x)$, donde β es un elemento primitivo de $\mathbb{F}(2^r)$. Sea $t = \lfloor (\delta - 1)/2 \rfloor$, como es usual, y sean $a_1, \ldots, a_e$ y $b_1, \ldots, b_e$ los números de ubicación de error y sus correspondientes magnitudes, respectivamente, donde $e \leq t$, (de modo que en el ejemplo anterior, $t = 1$, y como un error ocurrió en la segunda posición, entonces $a_1 = \beta^2$ y $b_1 = \beta^4$). Si $e < t$, entonces será conveniente

definir $a_i = 0$ para $e+1 \leq i \leq t$, aun cuando no existan tales ubicaciones de error. Luego, podemos computar $\delta - 1$ síndromes $s_{m+1}, \ldots, s_{m+\delta-1}$, los cuales son definidos por:

$$s_j = w(\beta^j), \text{ para } m+1 \leq j \leq m+\delta-1.$$

(Nótese que estas generalizan las definiciones de s_1 y s_3 usadas para código BCH 2-corrector). Para $m+1 \leq j \leq m+\delta-1$, tenemos que β^j es una raíz de $g(x)$, y por lo tanto, es una raíz de las palabras-código, de modo que

$$s_j = w(\beta^j) = c(\beta^j) + e(\beta^j) = e(\beta^j) = \sum_{i=1}^{t} b_i a_i^j. \tag{6.2}$$

De modo que el problema de decodificar aquí es hallar una forma efectiva de resolver $2e$ de las $\delta - 1$ ecuaciones (3) para las $2e$ incógnitas $a_1, \ldots, a_e$ y $b_1, \ldots, b_e$. (Nótese que $2e \leq 2t \leq \delta - 1$). La dificultad en hacer esto yace en la no linealidad de las ecuaciones resultantes al elevar los a_i a la j-ésima potencia. Sin embargo, mostraremos a continuación cómo se halla fácilmente un polinomio cuyas raíces son $a_1, \ldots, a_e$, justamente como hicimos en el ítem 6 del Algoritmo D de la página 124 cuando decodificamos un código BCH 2-corrector.

Sea $A = \{a_1, \ldots, a_e\}$. Definamos el *polinomio ubicador de errores* $\sigma_A(x)$ como el polinomio cuyas raíces son precisamente $a_1, \ldots, a_e$. Así:

$$\sigma_A(x) = (a_1 + x)(a_2 + x) \cdots (a_e + x). \tag{6.3}$$

Ahora definamos σ_j como el coeficiente de x^j en $\sigma_A(x)$. Luego de expandir el producto en $\sigma_A(x)$, obtenemos:

$$\sigma_A(x) = \sigma_0 + \sigma_1 x + \cdots + \sigma_{e-1} x^{e-1} + x^e. \tag{6.4}$$

Para cualquier i con $1 \leq i \leq e$, podemos multiplicar ambos lados de la ecuación (5) por $b_i a_i^j$, luego sustituir $x = a_i$ y finalmente sumar ambos lados sobre i desde 1 hasta t, pero la ecuación (4) garantiza que $\sigma_A(a_i) = 0$, que nos lleva a

$$\begin{aligned} 0 &= (\sum_{i=1}^{t} b_i a_i^j)\sigma_0 + (\sum_{i=1}^{t} b_i a_i^{j+1})\sigma_1 + \cdots + \sum_{i=1}^{t} b_i a_i^{j+e} \\ &= s_j\sigma_0 + s_{j+1}\sigma_1 + \cdots + s_{j+e} \end{aligned}$$

que puede ser reescrita como:

$$s_{j+e} = s_j\sigma_0 + s_{j+1}\sigma_1 + \cdots + s_{j+e-1}\sigma_{e-1}. \tag{6.5}$$

Pero afortunadamente conocemos los valores de $s_{m+1}, s_{m+2} \ldots, s_{m+2e}$, de modo que podemos sustituir $j = m+1, \ldots, m+e$ alternativamente para obtener e ecuaciones lineales en las incógnitas $\sigma_0, \ldots, \sigma_{e-1}$. Estas ecuaciones pueden ser escritas más sucintamente en forma matricial como sigue, (donde la i-ésima fila brega con la ecuación (5), para $j = m + i$):

$$\begin{bmatrix} s_{m+1} & s_{m+2} & \cdots & s_{m+e} \\ s_{m+2} & s_{m+3} & \cdots & s_{m+e+1} \\ \vdots & \vdots & & \vdots \\ s_{m+e} & s_{m+e+1} & \cdots & s_{m+2e-1} \end{bmatrix} \begin{bmatrix} \sigma_0 \\ \sigma_1 \\ \vdots \\ \sigma_{e-1} \end{bmatrix} = \begin{bmatrix} s_{m+e+1} \\ s_{m+e+2} \\ \vdots \\ s_{m+2e} \end{bmatrix}. \tag{6.6}$$

Es importante ver si este sistema lineal tiene una solución no trivial. Para verlo, denotemos con M la matriz $e \times e$ en la ecuación matricial (7). Entonces, M tiene rango completo, (o sea, que tiene determinante no nulo). Esto puede verse escribiendo $M =$

$$\begin{bmatrix} 1 & \dots & 1 \\ a_1 & \dots & a_e \\ \vdots & & \vdots \\ a_1^{e-1} & \dots & a_e^{e-1} \end{bmatrix} \begin{bmatrix} b_1 a_1^{m+1} & & 0 \\ & \ddots & \\ 0 & & b_e a_e^{m+1} \end{bmatrix} \begin{bmatrix} 1 & a_1 & \dots & a_1^{e-1} \\ \vdots & \vdots & & \vdots \\ 1 & a_e & \dots & a_e^{e-1} \end{bmatrix}.$$

Cada una de las tres matrices aquí tiene rango completo por el Lema 47, pues $a_1, \dots, a_e$ son distintos y $a_1, \dots, a_e, b_1, \dots, b_e$ son todos no nulos. Por lo tanto, la ecuación (7) puede ser resuelta para $\sigma_0, \dots, \sigma_{e-1}$. Nótese además que si el decodificador comienza suponiendo que $e = t$, (por supuesto, el valor de e nos es desconocido al principio), entonces M es una matriz $t \times t$, pero tendrá rango e. Esto se concluye también al partir M en las tres matrices arriba, usando el hecho de que definimos $a_i = 0$ para $e+1 \leq i \leq t$. Por lo tanto, el decodificador ahora conoce el valor de e.

Podemos hallar $a_1, \dots, a_e$ sustituyendo los elementos del cuerpo en la ecuación $\sigma_A(x) = \sigma_0 + \sigma_1 x + \dots + x^e$ (que es conocida ahora), pues las raíces de $\sigma_A(x)$ son precisamente $a_1, \dots, a_e$.

Ahora que $a_1, \dots, a_e$ son conocidas, la ecuación (3) forma un sistema lineal de ecuaciones en las variables $b_1, \dots, b_e$, que ya puede ser resuelto. Nuevamente, este sistema puede ser representado en forma matricial, como sigue:

$$\begin{bmatrix} a_1^{m+1} & a_2^{m+1} & \dots & a_e^{m+1} \\ a_1^{m+2} & a_2^{m+2} & \dots & a_e^{m+2} \\ \vdots & \vdots & & \vdots \\ a_1^{m+e} & a_2^{m+e} & \dots & a_e^{m+e} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_e \end{bmatrix} = \begin{bmatrix} s_{m+1} \\ s_{m+2} \\ \vdots \\ s_{m+e} \end{bmatrix}. \tag{6.7}$$

(Nuevamente por el Lema 47, y como $a_1, \dots, a_e$ son distintos y no nulos, esta matriz tiene rango completo, de modo que el sistema lineal siempre puede ser resuelto en $b_1, \dots, b_e$).

Por lo tanto tenemos el siguiente algoritmo de decodificación para códigos RS. En este algoritmo, definimos M' como la matriz extendida formada por M al agregársele una $e+1$-ésima coincidente con el lado derecho de la ecuación (7), o sea:

$$M' = \begin{bmatrix} s_{m+1} & s_{m+2} & \dots & s_{m+e+1} \\ s_{m+2} & s_{m+3} & \dots & s_{m+e+2} \\ \vdots & \vdots & & \vdots \\ s_{m+e} & s_{m+e+1} & \dots & s_{m+2e} \end{bmatrix}.$$

Algoritmo E. (Decodificación de códigos RS) Supongamos que una palabra en un código C de tipo $RS(2^r, \delta)$ con generador $g(x) = (\beta^{m+1} + x) \cdots (\beta^{m+\delta-1} + x)$ es transmitida y que w es recibida. Si $t = \lfloor (\delta - 1)/2 \rfloor$, hallamos la palabra-código más próxima a w en C como sigue:

1. Calcular $s_j = w(\beta^j)$, para $m + 1 \leq j \leq m + 2t$.

2. Colocando $e = t$, hallar el rango de la matriz extendida M'.

3. Si e es el rango de M', resolver el sistema lineal (7) para σ_0, ..., σ_{e-1}.

4. Hallar las raíces de $\sigma_A(x) = \sigma_0 + \sigma_1 x + \cdots + x^e$; estas raíces son las ubicaciones de error $a_1, \ldots, a_e$.

5. Resolver el sistema lineal (8) para $b_1 \ldots, b_e$; estas son las magnitudes correspondientes a $a_1, \ldots, a_e$, de modo que el más factible patrón de error queda completamente determinado.

Nótese que no se requieren más reducciones de filas de matrices en el ítem (5) del Algoritmo E, pues la matriz aquí es una submatriz de la que se puso en forma escalonada (o FE) en el ítem 2. El siguiente ejemplo deja esto claro.

Ejemplo. Sea

$$g(x) = (1+x)(\beta+x)(\beta^2+x)(\beta^3+x) = \beta^6 + \beta^5 x + \beta^5 x^2 + \beta^2 x^3 + x^4$$

el generador de un código de tipo $RS(2^3, 5)$, (de modo que $m = -1$ y $t = 2$), donde $\mathbb{F}(2^3)$ es construido usando $1 + x + x^3$. Supongamos que se recibe

$$w = \beta^6 \beta \beta^5 \beta^2 10 \beta^2.$$

Decodificamos w usando el Algoritmo E:

1. Como $m = -1$ y $\delta = 5$, computamos los cuatro síndromes s_0, s_1, s_2 y s_3 (en otras palabras, computamos s_i si β^i es una raíz de $g(x)$):

$$\begin{array}{llllllll}
s_0 = w(\beta^0) = \beta^6 & +\beta & +\beta^5 & +\beta^2 & +1 & +0+\beta^2 & = 1; \\
s_1 = w(\beta^1) = \beta^6 & +\beta^2 & +\beta^7 & +\beta^5 & +\beta^4 & +0+\beta^8 & = \beta^3; \\
s_2 = w(\beta^2) = \beta^6 & +\beta^3 & +\beta^9 & +\beta^8 & +\beta^8 & +0+\beta^{14} & = \beta^3; \\
s_3 = w(\beta^3) = \beta^6 & +\beta^4 & +\beta^{11} & +\beta^{11} & +\beta^{12} & +0+\beta^{20} & = 1.
\end{array}$$

2. Colocando $e = t = 2$, la matriz extendida M' es:

$$\begin{bmatrix} 1 & \beta^3 & \beta^3 \\ \beta^3 & \beta^3 & 1 \end{bmatrix}.$$

   Reducción de filas de M' produce la matriz

$$\begin{bmatrix} 1 & \beta^3 & \beta^3 \\ 0 & \beta^4 & \beta^2 \end{bmatrix}$$

   que tiene rango 2.

3. Como M' tiene rango completo, entonces que $e = 2$, y así resolvemos el sistema lineal

$$M \begin{bmatrix} \sigma_0 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} s_2 \\ s_3 \end{bmatrix}.$$

   Sin embargo, como observamos arriba, ya habíamos reducido M por filas en el ítem 2, de modo que debemos resolver

$$\begin{bmatrix} 1 & \beta^3 \\ 0 & \beta^4 \end{bmatrix} \begin{bmatrix} \sigma_0 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \beta^3 \\ \beta^2 \end{bmatrix}.$$

   Entonces, $\beta^4 \sigma_1 = \beta^2$, de modo que $\sigma_1 = \beta^5$. Por ende, $\sigma_0 + \beta^3 \beta^5 = \beta^3$, y luego $\sigma_0 = 1$.

4. Concluimos que el polinomio ubicador de errores es $\sigma_A(x) = \sigma_0 + \sigma_1 x + x^2 = 1 + \beta^5 x + x^2$. Substituyendo los elementos del cuerpo en $\sigma_A(x)$, hallamos que $\sigma_A(\beta) = 0$ y $\sigma_A(\beta^6) = 0$. Por lo tanto
$$\sigma_A(x) = 1 + \beta^5 x + x^2 = (\beta + x)(\beta^6 + x).$$
Así, los números de ubicación de error son $a_1 = \beta$ y $a_2 = \beta^6$.

5. Hay que resolver el siguiente sistema lineal:
$$\begin{bmatrix} 1 & 1 \\ \beta & \beta^6 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \beta^3 \end{bmatrix}$$
o bien
$$\begin{bmatrix} 1 & 1 \\ 0 & \beta^5 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$
Entonces, $\beta^5 b_2 = 1$ y así $b_2 = \beta^2$ y $b_1 + b_2 = 1$, de modo que $b_1 = \beta^6$. Por lo tanto, el más factible patrón de error es
$$e = 0\beta^6 0000\beta^2$$
y la más factible palabra-código es
$$c = w + e = \beta^6\beta^5\beta^5\beta^2 100.$$

Ejemplo. Sea
$$\begin{aligned} g(x) &= (1+x)(\beta+x)(\beta^2+x)(\beta^3+x)(\beta^4+x)(\beta^5+x) = \\ &1 + \beta^4 x + \beta^2 x^2 + \beta x^3 + \beta^{12} x^4 + \beta^9 x^5 + x^6 \end{aligned}$$
el generador de un código de tipo $RS(2^4, 7)$, (de modo que $m = -1$ y $t = 3$), donde $\mathbb{F}(2^4)$ es construido usando $1 + x + x^4$, (Tabla III de la página 113). Supongamos que se recibe
$$w(x) = 1 + \beta^4 x + \beta x^3 + \beta^9 x^5 + x^6.$$
Apliquemos el Algoritmo E a esta situación:

1.
$$\begin{aligned} s_0 &= w(\beta^0) = 1 + \beta^4 + \beta^1 + \beta^9 + \beta^0 = \beta^7 \\ s_1 &= w(\beta^1) = 1 + \beta^5 + \beta^4 + \beta^{14} + \beta^6 = \beta^0 \\ s_2 &= w(\beta^2) = 1 + \beta^6 + \beta^7 + \beta^{19} + \beta^{12} = \beta^9 \\ s_3 &= w(\beta^3) = 1 + \beta^7 + \beta^{10} + \beta^{24} + \beta^{18} = \beta^{12} \\ s_4 &= w(\beta^4) = 1 + \beta^8 + \beta^{13} + \beta^{29} + \beta^{24} = \beta^9 \\ s_5 &= w(\beta^5) = 1 + \beta^9 + \beta^{16} + \beta^{34} + \beta^{30} = \beta^7 \end{aligned}$$

2.
$$M' = \begin{bmatrix} \beta^7 & 1 & \beta^9 & \beta^{12} \\ 1 & \beta^9 & \beta^{12} & \beta^9 \\ \beta^9 & \beta^{12} & \beta^9 & \beta^7 \end{bmatrix} \leftrightarrow \begin{bmatrix} \beta^7 & 1 & \beta^9 & \beta^{12} \\ 0 & \beta^{12} & \beta^7 & \beta^6 \\ 0 & \beta^7 & \beta^2 & \beta \end{bmatrix}$$
$$\leftrightarrow \begin{bmatrix} \beta^7 & 1 & \beta^9 & \beta^{12} \\ 0 & \beta^{12} & \beta^7 & \beta^6 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$
de modo que M tiene rango 2 y por lo tanto el más factible patrón de error tiene peso $e = 2$.

3. Con $e = 2$, el sistema lineal (7) toma la forma

$$\begin{bmatrix} \beta^7 & 1 \\ 1 & \beta^9 \end{bmatrix} \begin{bmatrix} \sigma_0 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \beta^9 \\ \beta^{12} \end{bmatrix},$$

pero en el ítem 2 hubo reducción de filas de la matriz M', que nos permite poner:

$$\begin{bmatrix} \beta^7 & 1 \\ 0 & \beta^{12} \end{bmatrix} \begin{bmatrix} \sigma_0 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \beta^9 \\ \beta^7 \end{bmatrix}.$$

Entonces, $\beta^{12}\sigma_1 + \beta^7 = 0$, y así $\sigma_1 = \beta^{10}$, y $\beta^7\sigma_0 + \sigma_1 + \beta^9 = 0$, de modo que $\sigma_0 = \beta^6$.

4. $\sigma_A(x) = \beta^6 + \beta^{10}x + x^2 = (\beta^2 + x)(\beta^4 + x)$. Por lo tanto, $a_1 = \beta^2$ y $a_2 = \beta^4$.

5.

$$\begin{bmatrix} 1 & 1 \\ \beta^2 & \beta^4 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \beta^7 \\ 1 \end{bmatrix},$$

de modo que

$$\begin{bmatrix} 1 & 1 \\ 0 & \beta^{10} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \beta^7 \\ \beta^7 \end{bmatrix}.$$

Por lo tanto, $b_2 = \beta^{12}$ y $b_1 = \beta^2$. Así el más factible patrón de error es $e = 00\beta^2 0\beta^{12}0\ldots0$ y la más factible palabra-código enviada es

$$c = w + e = 1\beta^4\beta^2\beta\beta^{12}\beta^9100\ldots0.$$

Nótese que este esquema de decodificación es independiente de la naturaleza cíclica del código, y entonces, funcionará también para códigos de tipo $RS(2^r, \delta)$ recortados de largo n.

Ejercicios.

8. Sea C un código de tipo $RS(2^4, 7)$ con generador $g(x) = (1 + x)(\beta + x)(\beta^2 + x)(\beta^3 + x)(\beta^4 + x)(\beta^5 + x)$, donde $\mathbb{F}(2^4)$ está construido usando $1 + x + x^4$, (Tabla III, página 113). Decodifique las siguientes palabras recibidas, que se codificaron usando C:

   a) $0\beta^3\beta\beta^5\beta^3\beta^2\beta^6\beta^{10}\beta000000$;

   b) $1\beta^4\beta^2\beta0010\beta\beta^5\beta^3\beta^2 0\beta^{10}\beta$;

   c) $\beta0\beta^7 0\beta^{12}\beta^3\beta^3 10000000$.

9. Sea C un código de tipo $RS(2^4, 5)$ con generador $g(x) = (\beta + x)(\beta^2 + x)(\beta^3 + x)(\beta^4 + x)$, donde $\mathbb{F}(2^4)$ está construido usando $1 + x + x^4$, (Tabla III, notando que aquí $m = 0$). Decodifique las siguientes palabras recibidas usando C:

   a) $001\beta^8 00\beta^5 00000000$;

   b) $0\beta^{10}0\beta^6\beta^{13}\beta^8\beta^{11}\beta^3\beta^5 00000$;

   c) $\beta^4 0100\beta^2\beta^5\beta^{12}\beta^{14}000000$.

10. Tome el código C de tipo $RS(2^4, 5)$ del Ejercicio 9 y forme el código $C(4)$ de largo $n = 11$ (y dimensión $k = 7$). Decodifique las siguientes palabras recibidas, que fueron codificadas usando C:

a) $001\beta^8 00\beta^5 0000$;

b) $0\beta^{10}0\beta^6\beta^{13}0\beta^8\beta^{11}\beta^3\beta^5 0$;

c) $\beta^4 0100\beta^2\beta^5\beta^{12}\beta^{14}00$.

11. Sea C un código de tipo $RS(2^4, 9)$ con generador $g(x) = (1+x)(\beta+x)\cdots(\beta^7+x)$ y con $\mathbb{F}(2^4)$ construido usando $1+x+x^4$, (Tabla III). Halle el más factible patrón de error para palabras recibidas que fueron codificadas usando C y que tienen los siguiente síndromes:

a) $s_0 = \beta^2, s_1 = \beta^3, s_2 = \beta^4, s_3 = \beta^5, s_4 = \beta^6, s_5 = \beta^7, s_6 = \beta^8$ y $s_7 = \beta^9$;

b) $s_0 = \beta^9, s_1 = \beta^{13}3, s_2 = \beta^7, s_3 = \beta^4, s_4 = \beta^{12}, s_5 = \beta^4, s_6 = \beta^8$ y $s_7 = \beta^2$;

c) $s_0 = 1, s_1 = 1, s_2 = 1, s_3 = 1, s_4 = 1, s_5 = 1, s_6 = 1$ y $s_7 = 1$;

d) $s_0 = \beta^{10}, s_1 = \beta^3, s_2 = \beta^{13}, s_3 = \beta^3, s_4 = \beta^{12}, s_5 = \beta^5, s_6 = \beta^{13}$ y $s_7 = \beta^3$;

e) $s_0 = \beta^{12}, s_1 = \beta^8, s_2 = 0, s_3 = \beta^7, s_4 = \beta^{13}, s_5 = \beta^4, s_6 = \beta^{13}$ y $s_7 = 1$;

f) $s_0 = \beta^{12}, s_1 = 0, s_2 = 0, s_3 = \beta^2, s_4 = 0, s_5 = 0, s_6 = \beta^2$ y $s_7 = 0$.

CAPÍTULO 7
Codificando sobre álgebras booleanas

Un álgebra booleana es una estructura algebraica (una colección de elementos y operaciones sobre esos elementos obedeciendo axiomas definidos) que captura las propiedades esenciales de las operaciones de conjuntos (alternativamente, de las operaciones lógicas). Especificamente, brega con las operaciones conjuntistas de intersección $\cap$, unión $\cup$ y complemento de conjuntos (alternativamente, las operaciones lógicas de conjunción, disjunción y negación). Nosotros trabajaremos exclusivamente con el álgebra booleana sobre el conjunto universal $\{0, 1, \ldots, m-1\}$, donde $0 < m \in \mathbb{Z}$. Dados dos subconjuntos del mismo, digamos A y B, las operaciones conjuntistas citadas nos dan:

$$\begin{array}{lcl} A \cap B & = & \{x \text{ such that } x \in A \text{ y } x \in B\} \\ A \cup B & = & \{x \text{ such that } x \in A \text{ ó } x \in B\} \\ A^c & = & \{x \text{ such that } x \notin A\} \end{array}$$

donde las tablas de valores de verdad de la conjunción 'y', la disjunción ó y la negación '$\notin$' de la relación de pertenencia '$\in$' lo son (con C por cierto y F por falso):

$x \in A$	$x \in B$	$x \in A$ y $x \in B$	$x \in A$ ó $x \in B$	$x \notin A$
C	C	C	C	F
C	F	F	C	F
F	C	F	C	C
F	F	F	F	C

En este capítulo, trataremos otros códigos correctores de errores de gran importancia, llamados códigos de Reed-Muller, cuya decodificación lo es por el método conocido como lógica mayoritaria, (ver Secciones 42–43), en contraste con el método de decodificacón de máxima verosimilitud, usado hasta ahora en las presentes notas.

Los códigos de Reed-Muller son códigos formados por palabras cuyas posiciones coordenadas están indicadas por conjuntos de un álgebra booleana, como se explica antes del ejemplo de las páginas 152 y 153. Primero, presentamos una construccón recursiva de estos códigos.

7.1. Construcción recursiva de códigos de Reed-Muller

Una clase de códigos que incluyen los códigos extendidos de Hamming tratados en la Sección 26 del Capítulo 2 son los códigos de Reed-Muller, que definiremos inicialmente de la siguiente forma.

El código de Reed-Muller de largo 2^m y orden r, denotado $RM(r,m)$ donde $0 \leq r \leq m$, se define en forma recursiva así::

1. $RM(0,m) = \{00\ldots0, 11\ldots1\}; RM(m,m) = K^{2^m}$;
2. $RM(r,m) = \{(x, x+y) | x \in RM(r,m-1), y \in RM(r-1,m-1)\}$, para $0 < r < m$.

Ejemplo.

$$\begin{array}{ll} RM(0,0) = \{0,1\}; & \\ RM(0,1) = \{00,11\}; & RM(1,1) = K^2 = \{00,01,10,11\}; \\ RM(0,2) = \{0000,1111\}; & RM(2,2) = K^4; \\ RM(1,2) = \{(x,x+y)| & x \in \{00,01,10,11\}, y \in \{00,11\}\}. \end{array}$$

Una construcción recursiva de la matriz generadora de $RM(r,m)$, que denotaremos con $G(r,m)$, se inicializa con $G(0,1) = [1\,1]$ y continúa con

$$G(m,m) = \begin{bmatrix} G(m_1,m) \\ 00\cdots01 \end{bmatrix}, G(r,m) = \begin{bmatrix} G(r,m_1) & G(r,m_1) \\ 0 & G(r-1,m_1) \end{bmatrix}$$

para $0 < r \leq m_1 = m-1$.

Ejemplo. Las matrices generadoras para $RM(0,1)$ y $RM(1,1)$ son

$$G(0,1) = [1\,1] \text{ y } G(1,1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Ejemplo. Para $m = 2$, el largo es $n = 2^2 = 4$, y para $r = 1, 2$, tenemos:

$$G(1,2) = \begin{bmatrix} G(1,1) & G(1,1) \\ 0 & G(0,1) \end{bmatrix}, \quad G(2,2) = \begin{bmatrix} G(1,2) \\ 0001 \end{bmatrix}.$$

Usando el penúltimo ejemplo, tenemos:

$$G(1,2) = \begin{bmatrix} 11 & 11 \\ 01 & 01 \\ 00 & 11 \end{bmatrix}; \quad G(2,2) = \begin{bmatrix} 1111 \\ 0101 \\ 0011 \\ 0001 \end{bmatrix}.$$

Ejemplo. Para $m = 3$, tenemos $n = 2^3 = 8$, y así:

$$G(0,3) = (11111111); \quad G(3,3) = \begin{bmatrix} G(2,3) \\ 00000001 \end{bmatrix};$$

$$G(1,3) = \begin{bmatrix} G(1,2) & G(1,2) \\ 0 & G(0,2) \end{bmatrix}; \quad G(2,3) = \begin{bmatrix} G(2,2) & G(2,2) \\ 0 & G(1,2) \end{bmatrix}.$$

Luego, usando el ejemplo anterior:

$$G(1,3) = \begin{bmatrix} 1111 & 1111 \\ 0101 & 0101 \\ 0011 & 0011 \\ 0000 & 1111 \end{bmatrix}.$$

Ejercicios.

1. Halle la matriz generadora $G(2,3)$.
2. Halle las matrices generadoras $G(r,4)$ de los códigos $RM(r,4)$, donde $r=0,1,2$.

Con esta definición recursiva, se pueden probar fácilmente por inducción las propiedades básicas de los códigos de Reed-Muller.

Teorema 51. *El código $RM(r,m)$ tiene las siguientes propiedades:*

1. *su largo es $n=2^m$;*
2. *su distancia es $d=2^{m-r}$;*
3. *su dimensión es $\sum_{i=0}^{r}\binom{m}{i}$;*
4. *contiene los códigos $RM(r-1,m)$, para $r>0$;*
5. *su código dual es $RM(m-1-r,m)$, donde $r<m$.*

Demostración. Las pruebas de estas afirmaciones usan todas inducción. Dejamos como ejercicio probar que este teorema vale para todo código $RM(r,m)$, para $m=1,2,3,4$. Además, notamos que estas afirmaciones son obviamente ciertas para $r=0$ y $r=m$.

En primer lugar, queremos mostrar que $RM(r-1,m)\subseteq RM(r,m)$. Comenzamos con

$$G(1,m)=\begin{bmatrix} G(1,m-1) & G(1,m-1) \\ 0 & G(0,m-1)\end{bmatrix}.$$

Como $\mathbf{1}$ es la fila tope de $G(1,m-1)$, entonces la fila de todos unos $(\mathbf{1},\ \mathbf{1})$ está contenida en $RM(1,m)$.

En general, como $G(r-1,m-1)$ es una submatriz de $G(r,m-1)$ y $G(r-2,m-1)$ es una submatriz de $G(r-1,m-1)$, resulta que

$$G(r-1,m)=\begin{bmatrix} G(r-1,m-1) & G(r-1,m-1) \\ 0 & G(r-2,m-1)\end{bmatrix}$$

es una submatriz de $G(r,m)$ y luego $RM(r-1,m)$ es un subcódigo de $RM(r,m)$.

Vamos ahora a establecer la distancia $d=2^{m-r}$ para $RM(r,m)$ usando inducción en r.

Como $RM(r,m)=\{(x,x+y)\,|\,x\in RM(r,m-1), y\in RM(r-1,m-1)\}$ y $RM(r-1,m-1)\subseteq RM(r,m-1)$, entonces $x+y\in RM(r,m-1)$, y así si $x\neq y$ entonces, por la hipótesis inductiva, $\mathrm{w}(x+y)\leq 2^{m-1-r}$. Además, $\mathrm{w}(x)\leq 2^{m-1-r}$. Por lo tanto $\mathrm{w}(x,x+y)=\mathrm{w}(x+y)+\mathrm{w}(x)\leq 2\cdot 2^{m-1-r}=2^{m-r}$. Si $x=y$, entonces $\mathrm{w}(x,x+y)=\mathrm{w}(y,0)$, pero $y\in RM(r-1,m-1)$ y luego $\mathrm{w}(y,0)=\mathrm{w}(y)\leq 2^{m-1-(r-1)}=2^{m-r}$.

Por la definición de $G(r,m)$ tenemos:

$$\begin{aligned}\dim RM(r,m) &= \dim RM(r,m-1)+\dim RM(r-1,m-1)\\ &=\sum_{i=0}^{r}\binom{m-1}{i}+\sum_{i=0}^{r-1}\binom{m-1}{i}\end{aligned}$$

$$= \sum_{i=1}^{r} \left(\binom{m-1}{i} + \binom{m-1}{i-1} \right) + \binom{m-1}{0}.$$

Como $\binom{m}{i} = \binom{m-1}{i} + \binom{m-1}{i-1}$ y $\binom{m-1}{0} = 1 = \binom{m}{0}$, tenemos

$$\dim RM(r,m) = \sum_{i=0}^{r} \binom{m}{i}.$$

Finalmente, sea

$$RM(r,m) = \{(x, x+y) \,|\, x \in RM((r,m-1), y \in RM(r-1,m-1)\}$$

y sea

$$RM(m-r-1,m) = \{(x', x'+y') \,|\, x' \in RM(m-r-1,m-1),$$
$$y' \in RM(m-r-2,m-1)\}.$$

Por inducción, el dual de $RM(r,m-1)$ es $RM(m-r-2,m-1)$ y el dual de $RM(r-1,m-1)$ es $RM(m-r-1,\, m-1)$. Luego $x \cdot y' = 0$ y $x' \cdot y = 0$. Además, como $RM(r-1,\, m-1) \subseteq RM(r,m-1)$, tenemos que $y \cdot y' = 0$. Por lo tanto,

$$\begin{aligned}(x, x+y) \cdot (x', x'+y') &= (x+y) \cdot (x'+y') + x \cdot x' \\ &= 2(x \cdot x') + x \cdot y' + y \cdot x' + y \cdot y' \\ &= 0.\end{aligned}$$

Vemos que cada vector en $RM(r,m)$ es ortogonal a cada vector en $RM(m-r-1,m)$. Como

$$\dim RM(r,m) + \dim RM(m-r-1,m) = \sum_{i=0}^{r} \binom{m}{i} + \sum_{i=0}^{m-r-1} \binom{m}{i}$$

$$= \sum_{i=0}^{r} \binom{m}{m-i} + \sum_{j=0}^{m-r-1} \binom{m}{j} = \sum_{j=0}^{m} \binom{m}{j} = 2^m,$$

entonces el código $RM(r,m)$ es el dual de $RM(r,m)$. □

Ejercicios.

3. Mostrar que el Teorema 51 vale para los códigos $RM(r,m)$ con $1 \leq m \leq 4$, construidos en los ejemplos iniciales del capítulo y los ejercicios 1 y 2.

Consideremos los códigos the Reed Muller de primer orden, $RM(1,m)$. Notemos que $RM(m-2,m)$ tiene dimensión $2^m - m - 1$, distancia 4, largo 2^m y que por lo tanto es un código extendido de Hamming. Por el Teorema 51 sabemos que $RM(1,m)$ es el dual de este código extendido. Nótese que $RM(1,m)$ es un código pequeño con una distancia mínima grande, de modo que un buen algoritmo de decodificación es de hecho el más elemental: para cada palabra recibida w, hallar la palabra-código en $RM((1,m)$ más cercana a w, lo cual puede ser hecho muy eficientemente.

Ejemplo. Sea $m = 3$ y consideremos el código $RM(1,3)$, que tiene largo $8 = 2^3$ y cuyo número de palabras es $16 = 2^{3+1}$. Su distancia mínima es 4. Sea

$$G(1,3) = \begin{bmatrix} 1111 & 1111 \\ 0101 & 0101 \\ 0011 & 0011 \\ 0000 & 1111 \end{bmatrix}.$$

Nótese que si w es recibida y $d(w,c) < 2$, entonces decodificamos w como c, pero si $d(w,c) > 6$, entonces $d(w, \mathbf{1}+c) < 2$ y decodificamos w como $\mathbf{1}+c$. Por ejemplo, si $w = 10001111$ es recibida, entonces $c = 00001111$ es la palabra-código más cercana. Si $w = 10101011$ es recibida y hallamos $c = 01010101$ con $d(w,c) > 6$, entonces $\mathbf{1}+c = 10101010$ es la palabra-código más cercana. Luego, debemos examinar a lo sumo la mitad de las palabras-código en $RM(1,m)$.

Ejercicios.

4. Sea $G(1,3)$ la matriz generadora para el código $RM(1,3)$. Decodifique las siguientes palabras recibidas:

   a) 0101 1110
   b) 0110 0111
   c) 0001 0100
   d) 1100 1110

5. Sea $G(1,4)$ la matriz generadora del código $RM(1,4)$. Decodifique las siguientes palabras recibidas:

   a) 1011 0110 0110 1001
   b) 1111 0000 0101 1111

7.2. Construcción por funciones booleanas

Damos una construcción de los códigos de Reed-Muller más adecuada con la tarea de decodificación.

Como hicimos con los códigos de Reed-Solomon, etiquetaremos las posiciones coordenadas de las palabras de largo $n = 2^m$ con vectores de K^m, o sea, etiquetaremos las posiciones coordenadas con vectores $u_i \in K^m$, donde u_i es la representación binaria del entero i, con dígitos en reversa (dígitos de menor orden al inicio, no al final). Llamemos a este el *ordenamiento estándar* de K^m.

Ejemplo. El ordenamiento estándar para K^2 es $(00, 10, 01, 11)$, y para K^3 es $(000, 100, 010, 110, 001, 101, 011,$

Cualquier función f de K^k en $\{0,1\}$ tiene una única representación o *forma vectorial* $v = (f(u_0), f(u_1), \ldots, f(u_{2^m-1})) \in K^n$, donde $u_i \in K^m$, $n = 2^m$ y $u_0, u_1, \ldots u_{2^m-1}$ es el ordenamiento estándar de vectores de K^m, como quedó descrito arriba.

Estamos interesados en una clase de funciones básicas. Dado un subconjunto $I \subseteq \{0, 1, \ldots, m-1\}$, definimos una función

$$f_I(x_0, x_1, \ldots, x_{m-1}) = \begin{cases} \Pi_{i \in I}(x_i + 1), & \text{si } I \neq \emptyset, \\ 1, & \text{si } I = \emptyset. \end{cases}$$

f_I es una aplicación que asigna a los puntos de K^m valores en $\{0,1\}$. Por lo tanto f_I es una función booleana, pues asigna un valor 0 ó 1 en K a cada vector $v \in K^m$, que es el *vector característico* del conjunto $S(v) \subseteq \{0,\dots,m-1\}$ formado por aquellos valores $i \in \{0,\dots,m-1\}$ con coordenada $v_i = 1$. Tales conjuntos $S(v)$ forman lo que se conoce como el álgebra booleana de orden m, donde cada $S(v)$ es el *soporte* del vector v correspondiente.

Definimos v_I como la forma vectorial correspondiente a f_I.

Ejemplo. Sea $m = 3$, de modo que $n = 2^3$.

1. Si $I = \{1,2\}$, entonces $f_I(x_0,x_1,x_2) = (x_1+1)(x_2+1)$. La forma vectorial de $f_{\{1,2\}}(x_0,x_1,x_2)$ es formada al tomar cada uno de los elementos $x_0,x_1,x_2 \in K_3$ (usando el ordenamiento estándar) y al evaluar $f_{\{1,2\}}(x_0,x_1,x_2)$. Así,

$$f_{\{1,2\}}(0,0,0) = 1, f_{\{1,2\}}(1,0,0) = 1, f_{\{1,2\}}(0,1,0) = 0,$$

$$f_{\{1,2\}}(1,1,0) = 0, f_{\{1,2\}}(0,0,1) = 0, f_{\{1,2\}}(1,0,1) = 0,$$

   $f_{\{1,2\}}(0,1,1) = 0$ y $f_{\{1,2\}}(1,1,1) = 0$. Por lo tanto
   $v_I = 11000000$.

2. Si $I = \{0\}$, entonces $f_I(x_0,x_1,x_2) = (x_0+1)$ y $v_I = 10101010$,

3. Si $I = \emptyset$, entonces $f_\emptyset(x_0,x_1,x_2) = 1$ y $v_I = 11111111$.

Existen dos hechos respecto de la función f_I que serán usados posteriormente. Primero, $f(x_0x_1,\ \dots\ ,x_{m-1}) = 1$ si y solo si $x_i = 0$ para todo $i \in I$. Luego, en el ítem (1) del ejemplo anterior, $I = \{1,2\}$ nos da $f_I(x_0,x_1,x_2) = (x_1+1)(x_2+1)$ y $f_I(x_0,0,0) = (0+1)(0+1) = 1$ para $x_0 \in \{0,1\}$. Segundo, para cada $u_i \in K^m$ vale que $f_I(u_i)f_J(u_i) = f_{I\cup J}(u_i)$ y entonces

$$v_I \cdot v_J = \sum_{i=0}^{2^m-1} f_I(u_i)f_J(u_i) = \sum_{i=0}^{2^m-1} f_{I\cup J}(u_i) = \mathrm{w}(v_{I\cup J}) \pmod 2.$$

Usaremos $\mathbb{Z}_m$ para denotar el conjunto de enteros $\{0,1,2,\dots,m-1\}$.

Ejercicios.

6. Sea $m = 4$ de modo que $n = 2^4$. Para una de las siguientes selecciones de subconjuntos I de $\mathbb{Z}_4$, hallar f_I y v_I:

   a) $I = \{0,3\}$
   b) $I = \{0,1,3\}$
   c) $I = \{1\}$
   d) $I = \{1,3\}$
   e) $I = \emptyset$
   f) $I = \mathbb{Z}_4$

7. Sea $m = 5$ de modo que $n = 2^5$. Para cada una de las siguientes selecciones de subconjuntos I de $\mathbb{Z}_5$, hallar f_I y v_I:

a) $I = \{0, 2, 4\}$

b) $I = \{0, 1, 3, 4\}$

c) $I = \{1\}$

d) $I = \{1, 2, 4\}$

e) $I = \emptyset$

f) $I = \mathbb{Z}_5$

8. Sea I un subconjunto de $\mathbb{Z}_m$. Use el primer hecho citado arriba para mostrar que $\mathrm{w}(v_I) = 2^{m-|I|}$.

9. Si v es una combinación lineal de vectores de la forma v_I, ¿Cuándo tendrá v peso par?

10. Sea $m = 4$ de modo que $n = 2^4$. Para $I = \{0, 1, 3\}$ y $J + \{2, 3\}$, compute $v_I \cdot v_J$.

El código de Reed-Muller $RM(r, m)$ puede ser definido como el código lineal $\langle \{v_I \,|\, I \subseteq \mathbb{Z}_m, |I| \leq r\} \rangle$. Afirmamos que $S = \{v_I \,|\, I \subseteq Z_m, |I| \leq r\}$ es un conjunto linealmente independiente, (ver Ejercicio 12 abajo), y luego es una base para $RM(r, m)$. Contando el número de palabras v_I con $I \subseteq \mathbb{Z}_m$ y $|I| \leq r$, tenemos que para $RM(r, m)$ vale que $k = \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r}$, y claramente $n = 2^m$. Por supuesto, las palabras v_I pueden ser arregladas en cualquier orden para formar una matriz generadora para $RM(r, m)$. Definimos la matriz generadora $G_{r,m}$ de $RM(r, m)$ como en *forma canónica* si las filas son ordenadas de modo que v_I quede antes que v_J si $|I| < |J|$, o si $|I| = |J|$ y existe $j \in \mathbb{Z}_m$ tal que $f_I(u_j) < f_J(u_j)$ pero $f_I(u_i) = f_J(u_i)$ para $i > j$.

Ejemplo. La matriz generadora $G_{4,4}$ para $RM(4, 4)$ en forma canónica está dada como sigue, donde por conveniencia hemos escrito $v_{\{3\}}$ como v_3 y así siguiendo:

$$G_{4,4} = \begin{bmatrix} 1111111111111111 \\ \\ 1111111100000000 \\ 1111000011110000 \\ 1100110011001100 \\ 1010101010101010 \\ \\ 1111000000000000 \\ 1100110000000000 \\ 1010101000000000 \\ 1100000011000000 \\ 1010000010100000 \\ 1000100010001000 \\ \\ 1100000000000000 \\ 1010000000000000 \\ 1000100000000000 \\ 1000000010000000 \\ \\ 1000000000000000 \end{bmatrix} \begin{matrix} v_\emptyset \\ \\ v_3 \\ v_2 \\ v_1 \\ v_0 \\ \\ v_{2,3} \\ v_{1,3} \\ v_{0,3} \\ v_{1,2} \\ v_{0,2} \\ v_{0,1} \\ \\ v_{1,2,3} \\ v_{0,2,3} \\ v_{0,1,3} \\ v_{0,1,2} \\ \\ v_{0,1,2,3} \end{matrix}$$

Este ordenamiento se sigue de la definición, como se ejemplifica a continuación. Si $I = \{3\}$ y $J = \{0, 2\}$, como $|I| < |J|$, entonces $v_3 = v_I$ precede a $v_{0,2} = v_J$. Si $I = \{2, 3\}$ y $J = \{0, 2\}$, entonces $f_I(u_i) = f_J(u_i)$ para $i > 10$, pero $f_I(u_{10}) = 0 < 1 = f_J(u_{10})$. (Por supuesto, u_{10} en el ordenamiento estándar de K^4 es 0101). Por lo tanto $v_{2,3} = v_I$ precede a $v_{0,2} = v_J$.

Ahora es fácil ver que $G_{0,4}$, $G_{1,4}$, $G_{2,4}$ y $G_{3,4}$ son sencillamente las submatrices de $G_{4,4}$ formadas por las primeras $\binom{4}{0} = 1$, $\binom{4}{0} + \binom{4}{1} = 5$, $\binom{4}{0} + \binom{4}{1} + \binom{4}{2} = 11$ y $\binom{4}{3} = 15$ filas, respectivamente.

Ejercicios.

11. Halle (a) $G_{2,3}$ (b) $G_{2,4}$ (c) $G_{3,5}$ (d) $G_{0,10}$.

12. Muestre que para todo $r \geq m$ vale que $v_I \mid |I| \leq r$, $I \subseteq \mathbb{Z}_m\}$ es un conjunto linealmente independiente. (Pista: Arreglar las palabras en este conjunto de modo que v_I venga antes que v_j si, para algún j vale que $f_I(u_i) = f_J(u_i)$ para $j + 1 \leq i \leq m$ y $f_I(u_i) > f_J(u_i)$. O más formalmente, usar inducción en m y r).

La codificación se realiza, como para cualquier código lineal, al multiplicar un mensaje por $G_{r,m}$. Entonces, cualquier palabra-código c puede ser escrita como

$$c = \sum_{I \subseteq \mathbb{Z}_m, |I| \leq r} m_I v_I,$$

donde los dígitos del mensaje son etiquetados m_I para corresponder a las filas v_I de $G_{r,m}$.

Ejemplo. Codificar cada mensaje m como sigue, usando $G_{2,4}$, resulta en una palabra código c correspondiente:

1. Si $m = 1\ 0000\ 001000$ (de modo que $m_\emptyset = 1$ y $m_{0,3} = 1$) entonces

$$c = v_\emptyset + v_{0,3} = 0101010111111111.$$

2. Si $m = 0\ 0101\ 001001$ (de modo que $m_2 = m_0 = m_{0,3} = m_{0,1} = 1$) entonces $c = v_2 + v_0 + v_{0,3} + v_{0,1} = 0111100011010010$.

Ejercicios. [13.] Codifique los siguientes mensajes usando $G_{2,4}$:

1. $0\ 0101\ 000000$
2. $0\ 0000\ 000001$
3. $0\ 0100\ 001000$

7.3. Decodificación de códigos de Reed-Muller

Decodificamos los códigos de Reed-Muller decodificación *por lógica mayoritaria*. Precisamos algunos resultados preliminares. Para cualquier $I \subseteq \mathbb{Z}_m$, definimos $I^c = \mathbb{Z}_m \setminus I$, el *complemento* de I en $\mathbb{Z}_m$.

Sea $H_I = \{u \in K^m \mid f_I(u) = 1\}$, el *soporte* de I. Recordemos que

$$f_I(x_0, x_1, \ldots, x_{m-1}) = \Pi_{i \in I}(x_i + 1) = 1$$

si y sólo si $x_i = 0$ para todo $i \in I$. Claramente, si $x, y \in H_I$, entonces $x_i = y_i = 0 = x_i + y_i$ para todo $i \in I$; luego $x + y \in H_I$.

Para cualquier $u = (x_0, x_1, \ldots, x_{m-1}) \in K^m$ y para cualquier $t = (t_0, t_1, \ldots, t_{m-1}) \in K^m$ definimos otra función $f_{I,t}(x_0, x_1, \ldots, x_{m-1}) = f_I(x_0 + t_0, x_1 + t_1, \ldots, x_{m-1} + t_{m-1}) = f_I(x + t)$ y definimos $v_{I,t}$ como la forma vectorial de $f_{I,t}$.

Nos va a interesar el cálculo de $v_{I,s} \cdot v_{J^c,t}$. Y así precisaremos contar el número de palabras $u \in K^m$ para las cuales $f_{I,s}(u) f_{J^c,t}(u) = 1$. Por la definición de H_I tenemos que $f_{I,t}(u) = f_I(u+t) = 1$ si y sólo si $u + t = u' \in H_I$, o equivalentemente $u = u' + t \in H_I + t$, donde $H_I + t$ es la clase módulo H determinada por t. Y el valor de $f_{I,s}(u) f_{J^c,t}(u) = \Pi_{i \in I}(x_i + s_i + 1) \Pi_{j \in J^c}(x_j + t_j + 1)$ permanece igual para todas las selecciones de $x_k \in \{0, 1\}$, $k \in \mathbb{Z}_m \setminus (I \cup J^c)$. Como existen $2^{m - |I \cup J^c|}$ tales selecciones para $u \in K^m$, entonces el número de veces que $f_{I,s}(u) f_{J^c,t}(u)$ vale 1 es un múltiplo de $2^{m-|I \cup J^c|}$, aun cuando $|I \cup J^c| = m$; esto es, aun cuando $I \cup J^c = \mathbb{Z}_m$. Sin embargo, si suponemos que $|I| \leq |J|$, entonces $|J^c| \leq |I^c|$, y así $|I \cup J^c| = |I| + |J^c| - |I \cap J^c| < m$, a menos que $I = J$. Si $I = J$, entonces existe solo una palabra $u \in K^m$ para la cual $f_{I,s}(u) f_{J^c,t}(u) = 1$, que es la palabra u para la cual $x_i = s_i$ para todo $i \in I$ y $x_i = t_i$ para todo $i \in I^c$.

Por supuesto que hallar el número de posiciones en las cuales vale

$$f_{I,s}(u) f_{J^c,t}(u) = 1$$

nos provee inmediatamente $v_{I,s} \cdot v_{J^c,t}$, de modo que obtenemos el siguiente resultado.

Lema 52. *Sean I y J subconjuntos de $\mathbb{Z}_m$ con $|I| < |J|$. Luego para cada $s \in H_{I^c}$ y cada $t \in H_J$ vale que $v_{I,s} \cdot v_{J^c,t} = 1$ si y sólo si $I = J$.*

Ahora podemos fácilmente obtener el siguiente resultado, que es la base del esquema de decodificación a usar.

Corolario 53. *Si c es una palabra en $RM(r, m)$ y si $|J| = r$, entonces $m_J = c \cdot v_{J^c,t}$, para cada $t \in H_J$.*

Demostración. Si $|J| = r$, entonces para cada $t \in H_J$ vale que

$$c \cdot v_{J^c,t} = \sum_{I \subseteq \mathbb{Z}_m, |I| \leq r} m_I v_I \cdot v_{J^c,t} = m_J v_J \cdot v_{J^c,t} = m_J,$$

pues por el Lema 52 el único producto interno que no se anula en la sumatoria es aquel que tiene $I = J$. □

Lema 54. *Sea $J \subseteq \mathbb{Z}_m$. Para cada palabra e (de largo 2^m), $e \cdot v_{J^c,t} = 1$ a lo sumo para* $\mathrm{w}(e)$ *valores de $t \in H_J$.*

Demostración. Recordemos que para cada subespacio S de K^m, dos palabras están en la misma clase módulo S precisamente cuando su suma es una palabra en S. Además, H_I es un subespacio de K^m y la única palabra en ambos H_J y H_{J^c} es la palabra nula. Se sigue de estas observaciones que no existen dos palabras de H_J que ocurren juntas en una clase módulo H_{J^c}. Por lo tanto, cuando t recorre los elementos de H_J tenemos que $H_{J^c} + t$ forma la descomposición en clases módulo H_{J^c}.

El resultado se sigue ahora pues si $t_1 \neq t_2$ en H_J, justo habíamos probado que $(H_{J^c} + t_1) \cap (H_{J^c} + t_2) = \emptyset$, de modo que v_{J^c,t_1} y v_{J^c,t_2} no tienen posiciones en común donde ambos dígitos valgan 1. Por lo tanto cada uno de los $\mathrm{w}(e)$ dígitos no nulos en e afecta a lo sumo uno de los valores de $e \cdot v_{J^c}$ cuando t recorre todos los elementos de H_J. □

Podemos ahora obtener un algoritmo de decodificación como sigue. Sea $w = c+e$ una palabra recibida, donde c es palabra de $RM(r,m)$, de modo que $c = \sum_{I\subseteq\mathbb{Z}_m} m_I v_I$, donde $|I| \leq r$. Sea $J \subseteq \mathbb{Z}_m$ un conjunto de tamaño r. Luego por el Lema 53 tenemos que $e \cdot v_{J^c,t} = 0$ para al menos $|H_J| - \mathrm{w}(e)$ valores de t en H_J. Para tales valores de t tenemos:

$$\begin{aligned} w \cdot v_{J^c,t} &= c \cdot v_{J^c,t} + e \cdot v_{J^c,t} \\ &= c \cdot v_{J^c,t} \\ &= m_J \text{ por el Corolario 53.} \end{aligned}$$

De modo que si $2\mathrm{w}(e) < |H_J|$, cuando t recorre los elementos de H_J, más de la mitad de los $w \cdot v_{J^c,t}$ valdrán m_J.

Una vez que m_J haya sido calculado de esta manera para todos los $J \subseteq \mathbb{Z}_m$ con $|J| = r$, sea $w(r-1) = w + \sum_{|J|=r} m_J v_J$. Ahora $w((r-1)$ puede ser decodificada al tratarla como palabra recibida que fuera codificada usando $RM(r-1,m)$. Este proceso puede ser continuado hasta que m_J haya sido hallada para todos los $J \subseteq \mathbb{Z}_m$ con $|J| \leq r$.

Antes de recopilar el algoritmo notemos que el mismo corrige todos los patrones de error de peso menor que $|H_J|/2$ cuando $|J| \leq r$. Sin embargo, por el Ejercicio 8 tenemos que $|H_J| = \mathrm{w}(v_J) = 2^{m-|J|}$. De modo que todos los patrones de error de peso menor que 2^{m-r-1} son corregidos y por lo tanto $RM(r,m)$ tiene distancia mínima al menos 2^{m-r}. No obstante, si $I \subseteq \mathbb{Z}_m$ y $|I| = r$ entonces v_I es una palabra de $RM(r,m)$ de peso 2^{m-r}, de modo que tenemos otra prueba del siguiente resultado.

Lema 55. *La distancia mínima de* $RM(r,m)$ *es* 2^{m-r}.

Algoritmo de Lógica Mayoritaria para $RM(r,m)$**:** Para decodificar una palabra recibida, procedemos como sigue:

1. Sea $i = r$ y sea $w(r) = w$.

2. Para cada $J \subseteq \mathbb{Z}_m$ con $|J| = i$, calcular $w(i) \cdot v_{J^c,t}$ para cada $t \in H_J$ hasta que 0 ó 1 acontece más de 2^{m-r-1} veces, y sea m_J igual a 0 ó 1 respectivamente. Si ambos 0 y 1 acontecen más de $e = 2^{m-r-1}-$ veces entonces pedir retransimisión.

3. Si $i > 0$ entonces sea $w(i-1) = w(i) + \sum_{J\subseteq\mathbb{Z}_m,|J|=i} m_J v_J$ donde $|J| = i$. Si $w(i-1)$ tiene peso a lo sumo $e = 2^{m-r-1} - 1$ entonces colocar $m_J = 0$ para todo $J \subseteq \mathbb{Z}_m$ con $|J| \leq r$ y parar. Caso contrario, sustituir i con $i-1$ y volver al paso 2. (Si $i = 0$ entonces m_J ha sido calculado para todo $J \subseteq \mathbb{Z}_m$ con $|J| \leq 3$, de modo que el más factible mensaje ha sido hallado).

Ejemplo. Usemos el Algoritmo de Lógica Mayoritaria para decodificar la palabra recibida $w = 0101011110100000$ que fué originalmente codificada usando $G_{2,4}$.

Comenzamos con $i = r = 2$ y $w(2) = w$. Consideremos las siguientes computaciones:

J	t	$v_{J^c,t}$	$w \cdot v_{J^c,t}$	m_J
$[0,1]$	0000	1111 0000 0000 0000	0	
	0010	0000 1111 0000 0000	1	0
	0001	0000 0000 1111 0000	0	
	0011	0000 0000 0000 1111	0	
$[0,2]$	0000	1100 1100 0000 0000	0	
	0100	0011 0011 0000 0000	1	1
	0001	0000 0000 1100 1100	1	
	0101	0000 0000 0011 0011	1	
$[1,2]$	0000	1010 1010 0000 0000	1	
	1000	0101 0101 0000 0000	0	0
	0001	0000 0000 1010 1010	0	
	1001	0000 0000 0101 0101	0	
$[0,3]$	0000	1100 0000 1100 0000	0	
	0100	0011 0000 0011 0000	0	0
	0010	0000 1100 0000 1100	1	
	0110	0000 0011 0000 0011	0	
$[1,3]$	0000	1010 0000 1010 0000	0	
	1000	0101 0000 0101 0000	0	0
	0010	0000 1010 0000 1010	1	
	1010	0000 0101 0000 0101	0	
$[2,3]$	0000	1000 1000 1000 1000	1	
	1000	0100 0100 0100 0100	0	0
	0100	0010 0010 0010 0010	0	
	1100	0001 0001 0001 0001	0	

J	t	$v_{J^c,t}$	$w(1)\cdot v_{J^c,t}$	m_J
[0]	0000	1100 0000 0000 0000	0	
	0100	0011 0000 0000 0000	0	
	0010	0000 1100 0000 0000	1	0
	0110	0000 0011 0000 0000	0	
	0001	0000 0000 1100 0000	0	
	0101	0000 0000 0011 0000	0	
	0011	0000 0000 0000 1100		
	0111	0000 0000 0000 0011		
[1]	0000	1010 0000 0000 0000	0	
	1000	0101 0000 0000 0000	0	
	0010	0000 1010 0000 0000	1	0
	1010	0000 0101 0000 0000	0	
	0001	0000 0000 1010 0000	0	
	1001	0000 0000 0101 0000	0	
	0011	0000 0000 0000 1010		
	1011	0000 0000 0000 0101		
[2]	0000	1000 1000 0000 0000	1	
	1000	0100 0100 0000 0000	0	
	0100	0010 0010 0000 0000	0	0
	1100	0001 0001 0000 0000	0	
	0001	0000 0000 1000 1000	0	
	1001	0000 0000 0100 0100	0	
	0101	0000 0000 0010 0010		
	1101	0000 0000 0001 0001		
[3]	0000	1000 0000 1000 0000	1	
	1000	0100 0000 0100 0000	1	
	0100	0010 0000 0010 0000	1	1
	1100	0001 0000 0001 0000	1	
	0010	0000 1000 1000 1000	0	
	1010	0000 0100 0000 0100	1	
	0100	0000 0010 0000 0010		
	1110	0000 0001 0000 0001		

De aquí, vemos que $m_{2,3} = m_{1,3} = m_{0,3} = m_{1,2} = m_{0,1} = 0$ y $m_{0,2} = 1$. Luego,

$$w(1) = w(2) + v_{0,2} = 1111\ 0111\ 0000\ 0000$$

y $i = 1$. De las computaciones hechas, de nuevo concluimos que $m_3 = 1$ y $m_2 = m_1 = m_0 = 0$. Sea $w(0) = w(1) - v_3 = 0000\ 1000\ 0000\ 0000$ y sea $i = 1$. Como $w(0)$ tiene peso a lo sumo $e = 1$, colocamos $m_\emptyset = 0$ y nos paramos.

De modo que el más factible mensaje transmitido es 0 1000 000010 (pues los mensajes fueron codificados usando $G_{2,4}$).

Ejercicios.

14. Aquí los mensajes son codificados usando la matriz generadora $G_{2,4}$. Si fuese posible, decodificar los siguientes mensajes recibidos:

a) $w = 0111\ 0101\ 1000\ 1000$

b) $w = 0110\ 0110\ 0001\ 0000$

c) $w = 0101\ 1010\ 0100\ 0101$

d) $w = 1110\ 1000\ 1001\ 0001$

e) $w = 0011\ 0000\ 0011\ 0100$

f) $w = 1001\ 0110\ 0101\ 1010$

g) $w = 1010\ 1000\ 1010\ 0000$

h) $w = 0011\ 1100\ 0001\ 1100$

i) $w = 1001\ 1101\ 0001\ 1101$

15. Aquí los mensajes son codificados usando la matriz generadora $G_{2,5}$. Si fuese posible, decodificar los siguientes mensajes recibidos:

a) $w = 1100\ 1000\ 1110\ 0000\ 1100\ 0000\ 1100\ 0100$

b) $w = 0101\ 0111\ 0101\ 1000\ 1000\ 1000\ 0111\ 1010$

c) $w = 0011\ 0011\ 1111\ 0011\ 0011\ 0011\ 1111\ 1111$

d) $w = 0100\ 0000\ 1111\ 1111\ 0000\ 1100\ 0000\ 1111$

e) $w = 1001\ 0101\ 0110\ 1001\ 1001\ 0111\ 0110\ 1010$

f) $w = 0011\ 1111\ 0011\ 0011\ 1100\ 1100\ 1100\ 0100$

g) $w = 0100\ 0100\ 1111\ 1111\ 0000\ 1100\ 0000\ 1111$

7.4. Decisión en lógica mayoritaria

En otras palabras, el código $RM(2,\ 4)$ es un código [16,11,4] de segundo orden. Su matriz generadora $G_{2,4}$, formada por las once primeras filas de la matriz $G_{4,4}$ dada en la página 155, nos dice que los símbolos de un mensaje

$$a = a_\emptyset a_3 a_2 a_1 a_0 a_{23} a_{13} a_{12} a_{03} a_{02} a_{01}$$

son codificados en la palabra

$$c = aG = a_\emptyset \mathbf{1} + a_3 v_3 + \cdots + a_0 v_0 + \cdots + a_{01} v_{01}$$

$$x_0 x_1 \cdots x_{15}, \quad \text{digamos}$$

Este es un código 1-corrector, y veremos como corregir un error por lógica mayoritaria. La primera etapa es recuperar los seis símbolos $a_{01}, \ldots, a_{23}$. Si no hay errores, $G_{2,4}$ nos dice que

$$\begin{aligned}
a_{01} &= x_0 + x_1 + x_2 + x_3 \\
&= x_4 + x_5 + x_6 + x_7 \\
&= x_8 + x_9 + x_{10} + x_{11} \\
&= x_{12} + x_{13} + x_{14} + x_{15}, \\
a_{02} &= x_0 + x_1 + x_4 + x_5 \\
&= x_2 + x_3 + x_6 + x_7 \\
&= x_8 + x_9 + x_{12} + x_{13} \\
&= x_{10} + x_{11} + x_{14} + x_{15}, \\
&\cdots\cdots\cdots\cdots \\
a_{23} &= x_0 + x_4 + x_8 + x_{12} \\
&= x_1 + x_5 + x_9 + x_{13} \\
&= x_2 + x_6 + x_{10} + x_{14} \\
&= x_3 + x_7 + x_{11} + x_{15}.
\end{aligned}$$

Las primeras cuatro ecuaciones aquí nos proveen 4 votos para el valor de a_{01}. Las siguientes cuatro ecuaciones nos proveen 4 votos para el valor de a_{02}, y así siguiendo. De modo que si un error acontece, la mayoría de votos es aún correcta, y así cada a_{ij} es obtenido correctamente.

Para hallar los símbolos $a_0, \ldots, a_3$, restamos

$$a_{23}v_3v_4 + \cdots + a_{02}v_0v_2$$

de x, obteniéndose digamos $x' = x'_0x'_1 \cdots x'_{15}$. Nuevamente de la matriz $G_{2,4}$ observamos que

$$\begin{aligned}
a_0 &= x'_0 + x'_1 \\
&= x'_2 + x'_3 \\
&\cdots \\
&= x'_{14} + x'_{15}, \\
a_1 &= x'_0 + x'_2 \\
&= \cdots
\end{aligned}$$

Ahora está más fácil: hay 8 votos para cada a_i, y así si hay un error, el voto mayoritario da cada a_i correctamente. Queda por determinar $a_\emptyset$. Tenemos,

$$\begin{aligned}
x'' &= x' - a_3v_3 - \cdots - a_0v_0 \\
&= a_\emptyset \mathbf{1} + \text{ error},
\end{aligned}$$

y $a_\emptyset = 0$ ó 1 de acuerdo con el número de unos en x''.

Este esquema es llamado *algoritmo de decodificación de Reed*.

¿Cómo se hallan las componentes de las palabras-código que son usadas en los controles de paridad expresados en los grupos de ecuaciones arriba que producen los votos para a_{01}, a_{02}, etc.? Daremos para ver esto una descripción geómetrica del algoritmo de decodificación de $RM(r, m)$. Primero, hallamos a_σ, donde $\sigma = \sigma_1 \cdots \sigma_r$, digamos. La fila correspondiente de la matriz generadora $v_{\sigma_1 \cdots \sigma_r} = v_{\sigma_1} \cdots v_{\sigma_r}$ (cada coordenada de la cual es obtenida multiplicando módulo 2 las coordenadas correspondientes de los v_{σ_i}) es el vector de incidencia de un subespacio S de dimensión $(m - r)$ en el espacio euclideano $\mathbf{F}(2^m)$. Por ejemplo, la línea doblemente trazada en la Figure 3 indica el plano S correspondiente a a_{01}. Sea T el subespacio "complementario" correspondiente a S con vector de incidencia $v_{\tau_1}, \cdots, v_{\tau_{m-r}}$, donde $\{\tau_1, \ldots, \tau_{m-r}\}$ es el complemento de $\{\sigma_1, \ldots, \sigma_r\}$ en $\mathbb{Z}_m$. Claramente, T interseca a S en un solo punto.
Sean $U_1, \ldots, U_{2^{m-r}}$ todas las traslaciones de T en $\mathbf{F}(2^m)$, incluyendo T mismo. (Estas traslaciones están sombreadas en la Figura 3). Cada U_i interseca a S en exactamente un punto.

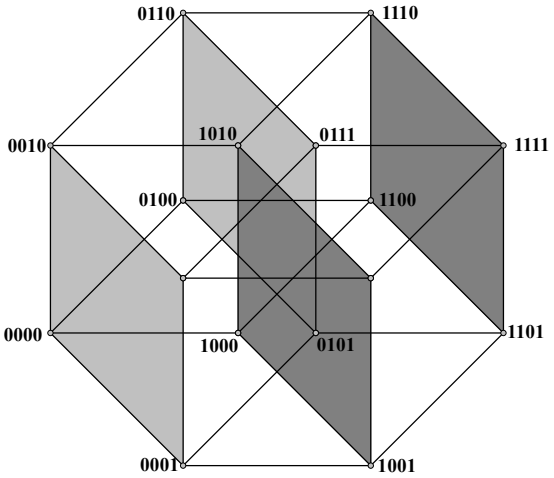

Figura 7.1: Espacio euclideano $\mathbf{F}(2^4)$

Teorema 56. *Si no hay errores, entonces a_σ esta dado por*

$$a_\sigma = \sum_{P \in U_i} x_P, \quad i = 1, \ldots, 2^{m-r}.$$

Estas ecuaciones son una generalización de las ecuaciones ejemplificadas arriba y producen 2^{m-r} votos para a_σ.

Demostración. Por la forma que tiene la matriz generadora, la palabra-código c es

$$c = \sum_{\rho = \rho_1 \cdots \rho_s} a_\rho v_{\rho_1} \cdots v_{\rho_s},$$

donde la suma acontece sobre todos los subconjuntos $\{\rho_1, \ldots, \rho_s\}$ de $\mathbb{Z}_m$. Por lo tanto

$$\begin{aligned} \textstyle\sum_{P \in U_i} x_P &= \textstyle\sum_\rho a_\rho \sum_{P \in U_i} (v_{\rho_1} \cdots v_{\rho_s})_P \\ &= \textstyle\sum_P a_P N(U_i, \rho), \end{aligned}$$

donde $N(U_i, \rho)$ es el número de puntos en la intersección de U_i con el subespacio W con vector de incidencia $v_{\rho_1} \cdots v_{\rho_s}$.

Usamos el hecho de que la intersección de dos subespacios es un subespacio, y todos los subespacios (excepto los puntos) contienen un número par de puntos. Ahora bien, T y W se intersecan el el subespacio

$$v_{r_1} \cdots v_{r-m} v_{\rho_1} \cdots v_{\rho_s}.$$

Si $s < r$, este subespacio tiene dimensión al menos 1, y $N(U_i, \rho)$ es par. Por otra parte, si $s = r$ pero $W \neq S$, entonces uno de los ρ_i debe ser igual a uno de los τ_j, digamos $\rho_1 = \tau_1$. Entonces T y W se intersecan en

$$v_{\tau_1} \cdots v_{\tau_{m-r}} v_{\rho_2} \cdots v_{\rho_s},$$

que de nuevo tiene dimensión al menos 1, y $N(U_i, \rho)$ es par. Finalmente, si $W = S$, entonces $N(U_i, \rho) = 1$. □

El Teorema 56 implica que, si no ocurren más de $\lfloor (2^{m-r}-1)/2 \rfloor$ errores, entonces la decodificación por lógica mayoritaria recuperará cada uno de los símbolos a_σ correctamente, donde σ es cualquier cadena de r símbolos. El resto de los $a$es puede ser recuperado en la misma forma, como se mostró en el ejemplo anterior. Luego, el algoritmo de decodificación de Reed puede corregir $\lfloor (d-1)/2 \rfloor = \lfloor (2^{m-r}-1)/2 \rfloor$ errores.

APÉNDICE A
Soluciones sobre ejercicios seleccionados

I. Introducción a la Teoría de Códigos

1. (a) 000, 001, 010, 011, 100, 101, 110, 111 (b) 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111.

2. 2^n.

4. Tal canal puede ser convertido en un canal perfecto reemplazando cada 1 por un 0 y cada 0 por un 1.

5. Reemplazar cada 0 por un 1 y cada uno por un 0.

6. 001.

7. $C = \{0000, 0011, 0101, 0110, 1001, 1100, 1111\}$; (a) Sí; (b) 0101, 1001, 1100, 1111; (c) No; cada palabra de largo 4 que no está en C tiene 4 palabras-código diferentes más cercanas.

9. 8.

10. 16, 32, 2^{n-1}.

11. $1, 3/4, 1/3$.

12. (a) $p^3(1-p)5 = 2{,}2 \times 10^{-8}$; (b) $p^7 = {,}81$; (c) $(1-p)^5 = 2{,}4 \times 10^{-8}$; (d) $p^5 = {,}86$; (e) $p^4(1-p)^3 = 2{,}4 \times 10^{-5}$; (f) $(1-p)^5 = 2{,}4 \times 10^{-8}$; (g) $(1-p)^5 = 7{,}3 \times 10^{-10}$.

13. 0001110.

14. 101101101.

15. 00011.

16. 100110.

17. 110101 ó 101000.

18. (a) $\phi_p(v_1, w) \leq \phi_p(v_2, w)$ si y sólo si $d_1 \leq d_2$; (b) $\phi_p(v, w) = (1/2)^n$, para algún w y cualquier v.

26. Si 000, 001,010 ó 011 es recibida, entonces la DMVI decide que 001 fue enviada. En todos los otros casos, la DMVI decide incorrectamente que 101 fue enviada.

27. 000 es decodificada como 000; 001, 011 y 101 son decodificadas como 110; 010 y 100 requieren retransmisión.

28. * en las tablas siguientes indica que se requiere retransmisión.

(a)

Palabra recibida	Decodifica a
001	*
001	*
010	011
011	011
100	101
101	101
110	111
111	111

(b)

Palabra recibida	Decodifica a
001	000
001	*
010	010
011	011
100	000
101	001
110	010
111	011

29. (a) $L(001) = \{000, 001, 010, 011\}$; por lo tanto, $\theta_p(C, 001) = p^3 + 2p^2(1-p) + p((1-p)^2$; (b) $L(101) = \{100, 101, 110, 111\}$; por lo tanto, $\theta_p(C, 101) = p^3 + 2p^2(1-p) + p(1-p)^2$.

30. (a) $\theta_p(C, 110) = p^3 + p^2(1-p)$; (b) Para decodificar 000, sólo 000 puede ser recibida, de modo que $\Theta_p(110, 000) = p(1-p)^2$.

31. (a) $\theta_p(C, 101) = p^3 + p^2(1-p)$, para todo $v \in C$; (b) $\theta_p(C, v) = p^3 + p^2(1-p)$, para todo $v \in C$; (c) $\theta_p(C, 0000) = p^4 + 3p^3(1-p), \theta_p(C, 0001) = p^4 + 3p^3(1-p)$ y $\theta_p(C, 1110) = p^4 + 4p^3(1-p)$; (e) $\theta_p(C, 00000) = \theta_p(C, 11111) = p^5 + 5p^4(1-p) + 10p^3 + (1-p)^2$; (g) $\theta_p(C, v) = p^5 + 3p^4(1-p)$, para todo $v \in C$; (h) $\theta_p(C, v) = p^6 + 6p^5(1-p) + 9p^4(1-p)^2$, para todo $v \in C$.

32. (a) No; (b) Sí; (c) No.

33. (a)(i) No; (ii) Sí; (iii) No; (b)(i) Sí; (ii) Sí; (iii) No.

34. Ninguno;

36. (a) 001, 011, 101, 111; (c) $K^4 \setminus \{0000, 0001, 1110, 1111\}$; (e) $K^5 \setminus \{00000, 11111\}$; (b) $K^6 \setminus \{000000, 101010, 010101, 111111\}$.

38. (a) 1; (b) 1; (c) 1; (d) 2; (e) 5; (f) 3; (g) 2; (h) 3.

39. 2.

41. $K^3 \setminus \{000, 011, 101, 110\}$.

42. (a) Ninguno; (d) 1000, 0100,0010 y 0001; (e) todos los patrones de error de pesos 1,2,3 ó 4; (h) todos los patrones de error de pesos 1 ó 2.

48. (a)(i) 000,001; (ii) 000; (c)(i) 0000, 0010,0100, 1000; (ii) 0000,; (f)(i) 00000, 10000, 01000, 00100, 00010, 00001; (ii) 00000, 10000, 01000, 00100, 00010, 00001; (g)(i) 00000, 01000, 00100, 00010; (ii) 00000.

II. Códigos Lineales

1. (a) y (c) no son códigos lineales; los restantes son códigos lineales.

5. (a) $\langle S \rangle = \{000, 010, 011, 111, 001, 101, 100, 110\}$;
(b) $\langle S \rangle = \{0000, 1010, 0101, 1111\}$; (d) $\langle S \rangle = K^4$.

8. (a) $C^\perp = \{000\}$; (b) $C^\perp = \{0000, 1010, 0101, 1111\}$; (c) $C^\perp = \{0000, 1111\}$.

12. (a) linealmente independiente; (b) $\{101, 011, 010\}$; (e) linealmente independiente; (f) $\{1100, 1010, 1001\}$; (i) $\{10101010, 01010101\}$.

13. (a) $B = \{100, 010, 001\}$, $B^\perp = \emptyset$; (b) $B = \{1010, 0101\}$, $B^\perp = B$; (c) $B = \{0101, 1010, 1100\}$, $B^\perp = \{1111\}$;
(e) $B = \{11000, 01111, 11110, 01010\}, B^\perp = \{11111\}$.

14. (a) $\dim C = 3$, $\dim C^\perp = 0$; (b) $\dim C = 2$, $\dim C^\perp = 2$; (c) $\dim C = 3$, $\dim C^\perp = 1$; (e) $\dim C = 4$, $\dim C^\perp = 1$; (f) $\dim C = 3$, $\dim C^\perp = 2$.

18. (a) $\dim C = 4$; (b) $|C| = 16$.

21. $BC = \begin{bmatrix}110000\\011101\\101101\end{bmatrix}$, $BD = \begin{bmatrix}1000\\0010\\1010\end{bmatrix}$, $DC = \begin{bmatrix}101011\\110000\\011011\\000110\end{bmatrix}$.

25. $A \leftrightarrow \begin{bmatrix}11011\\00101\\00000\end{bmatrix}$, $B \leftrightarrow \begin{bmatrix}1001\\0101\\0000\end{bmatrix}$, $C \leftrightarrow \begin{bmatrix}101011\\011011\\000110\\000000\end{bmatrix}$, $D = \begin{bmatrix}1000\\0101\\0010\\0000\end{bmatrix}$.

26. (a) $\{100, 010, 001\}$; (c) $\{1001, 0101, 0011\}$;

(e) $\{10111, 01111, 00101, 00011\}$ es bueno, pero ya que estamos, confesemos nos gusta más $\{10001, 01001, 00100, 00011\}$, pues está en FRF.

27. (a) $\{010, 011, 111\}$; (c) $\{0101, 1010, 1100\}$;

(e) $\{11000, 01111, 11110, 01010\}$; (g) $\{0110, 1010, 0011\}$.

28. (a) $\emptyset$; (b) $\{1010,0101\}$; $(e)\{11111\}$; (h) $\{101000, 110110, 000101\}$.

30. (a) $B = \{111000, 000111\}$;

(b) $B = \{1000110, 0100011, 0010111, 0001101\}$;

(c) $B = \{1000001, 010001, 0010001, 0001001, 0000101,$
$0000011\}$;

31. (i) Sí; (ii) No.

32. (a) $\begin{bmatrix}010\\001\end{bmatrix}$; (b) $\begin{bmatrix}1001\\0110\end{bmatrix}$; (c) $\begin{bmatrix}11011\\00111\end{bmatrix}$.

33. (a) $\begin{bmatrix}100110\\010101\\001011\end{bmatrix}$, $\dim C = 3$.

34. (a) $\begin{bmatrix}10010110\\01010101\\00110011\\00001111\end{bmatrix}$, $(8, 4, 4)$; (c) $\begin{bmatrix}100100100\\010010010\\001001001\end{bmatrix}$, $(9, 3, 3)$.

(f) $\begin{bmatrix}101010\\011010\\000111\end{bmatrix}$, $(6, 3, 2)$; (g) $\begin{bmatrix}1001011\\0101010\\0011001\\0000111\end{bmatrix}$, $(7, 4, 3)$.

35. (a) (i) 10011; (ii) 01010; (iii) 11100.

38. 33.(a) $|C| = 8, R = 1/2$; (b) $|C| = 8, R = 1/3$; (c) $|C| = 4, R = 1/5$; 34.(a) $|C| = 16, R = 1/2$; (b) $R = 16, R1/2$; (c) $|C| = 8, R = 1/3$; (d) $|C| = 8, R = 3/5$; (f) $|C| = 8, R = 1/3$; (g) $|C| = 16, R = 4/7$.

39. (a) $\begin{bmatrix}1\\0\\0\end{bmatrix}$; (b) $\begin{bmatrix}01\\10\\10\\01\end{bmatrix}$; (e) $\begin{bmatrix}001\\111\\100\\010\\001\end{bmatrix}$.

40. 33.(a) $\begin{bmatrix}110\\101\\011\\100\\010\\001\end{bmatrix}$; (b) $\begin{bmatrix}10010\\01010\\00101\\10000\\01000\\00100\\00010\\00001\end{bmatrix}$; 34.(b) $\begin{bmatrix}1000\\1000\\0010\\0010\\0100\\0100\\0001\\0001\end{bmatrix}$; (d) $\begin{bmatrix}11\\01\\10\\01\\01\end{bmatrix}$; (g) $\begin{bmatrix}111\\110\\101\\100\\011\\010\\001\end{bmatrix}$.

41. (a) $G(C^{\perp}) = G(C) = \begin{bmatrix}110000\\001010\\000101\end{bmatrix}$.

42. $C^{\perp}$ consiste de las 16 palabras de peso par en K^5.

43. (a) $\dim C = t, \dim C^{\perp} = 2^t - t - 1, |C| = 2^t, |C^{\perp} = 2^{2^t-t-1}, R = t/(2^t - 1)$; (b) $\dim C = 11, \dim C^{\perp} = 12, |C| = 2^{11} = 2048, |C^{\perp} = 2^{12} = 4096, R = 11/23$; (c) $\dim C = 8, \dim C^{\perp} = 7, |C| = 2^8 = 256, |C^{\perp}| = 2^7 = 128, R = 8/15$.

45. (a) 1111100; (b) 1011000.

48. (a) $C' = 00000, 11100, 10101, 01001\}$.

49. (a) $G' = \begin{bmatrix}100011\\010010\\001001\\000100\end{bmatrix}$.

50. (a) $G' = \begin{bmatrix}10110\\01011\end{bmatrix}$.

52. (a) Sí; (b) Sí; (c) No.

54. (a) 4; (b) 4; (c) 4.

57. $C, C + 1000, C + 0010, C + 0011$; (b0 $C, C + 1000, C + 0100, C + 0001$.

58. (a) $C, C+100000, C+010000, C+001000, C+000100, C+000010, C+000001, C+001001$; (d) $C, C + 100000$; (f) $C, C + 1000, C + 0100, C + 0010, C + 0001, C + 1100, C + 1010, C + 1001$.

59. (a) $C, C + 1000, C0100, C + 0001$; (b) $C, C + 1000000, C + 0100000, C + 001000, C + 0001000, C+0000100, C+0000010, C+0000001$; (c) $C, C+000100, C+010000, C+001100, C+100000, C + 100100, C + 110000, C + 110100$.

61 (a) 010011; (b) 101001; (c) 001111; (d) 010011; (e) 110101; (f) 001111.

62. (a)

Patrón de error	Síndrome
*	11
0000	00
*	01
0010	10

$H = \begin{bmatrix}01\\01\\10\\01\end{bmatrix}$

63. (a)

64. (a)

Patrón de error	Síndrome
000000	000
000001	001
000010	010
100000	011
000100	100
010000	101
001000	110
$*$	111

$$H = \begin{bmatrix} 011 \\ 101 \\ 110 \\ 100 \\ 010 \\ 001 \end{bmatrix}$$

Patrón de error	Síndrome
0000000	000
0000001	001
0000010	010
0001000	011
0000100	100
0010000	101
0100000	110
1000000	111

71. (a)(i) 1100; (ii) 1001; (iii) 0101; (c)(i) 001110; (ii) 001110; (iii) 011011.

73. (a)

Patrón de error	Síndrome
0000000	000
0000001	001
0000010	010
0001000	011
0000100	100
0010000	101
0100000	110
1000000	111

74. 57.(a) $\theta_p(C) = p^4+p^3(1-p)$; (c) $\theta_(c) = p^5+3p^4(1-p)$; 57.(a) $\theta_p(C) = p^6+6p^5(1-p)$; (b) $\theta_p(C) = p^6+6p^5(1-p)+9p^4(1-p)^2$; 59.(a) $\theta_p(C) = p^4+2p^3(1-p)$; (b) $\theta_p(C) = p^7+7p^6(1-p)$.

III. Códigos Perfectos

2. (a) 2^4; (b) 2^4; (c) 2^8; (e) 2^8; (f) $2^{12} = 4096$.

6. (a) (8,6,3), no, $16 \leq |C| \leq 16$; (d) (15,6,3), si, , $2048 \leq |C| \leq 2048$.

7. (a) $64 \leq |C| \leq 256$; (b) $2048 \leq |C| \leq 2048$; (c) $128 \leq |C| \leq 128$; (d) $256 \leq |C| \leq 256$; (e) $32 \leq |C| \leq 256$; (f) $16 \leq |C| \leq 32$.

8. No;

14.

Patrón de error	Síndrome
0000000	000
1000000	111
0100000	110
0010000	101
0001000	011
0000100	100
0000010	010
0000001	001

$$; \quad H = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix};$$

(a) 0101011; (c) 0011110.

25. Los tres números solicitados son 17, 17 y 696, respectivamente.

IV. Códigos Cíclicos

6. (a) $q(x) = x^3$, $r(x) = x^3$.

8. (a) $\{0, x^2, x, x+x^2\}$; (c) $\{0, x^3, 1+x+x^2\}$.

19 (a) $g(x) = 1$; (e) $g(x) = 1 + x$.

21. (a) $\begin{bmatrix} 1011000 \\ 0101100 \\ 0010110 \\ 0001011 \end{bmatrix}$

22. (b) $g(x) = 1 + x^2 + x^4$, $\begin{bmatrix} 101010 \\ 010101 \end{bmatrix}$

V. Códigos de Bose-Chaudhuri-Hocquenghem

7. (a) $\begin{array}{l} 00 \leftrightarrow 0 \\ 10 \leftrightarrow \beta^0 \\ 01 \leftrightarrow \beta \\ 11 \leftrightarrow \beta^2 \end{array}$; (b) $\begin{array}{l} 000 \leftrightarrow 0 \\ 100 \leftrightarrow \beta^0 \\ 010 \leftrightarrow \beta \\ 001 \leftrightarrow \beta^2 \\ 101 \leftrightarrow \beta^3 \\ 111 \leftrightarrow \beta^4 \\ 110 \leftrightarrow \beta^5 \\ 011 \leftrightarrow \beta^6 \end{array}$.

9. $\beta, \beta^2, \beta^4, \beta^7, \beta^8, \beta^{11}, \beta^{13}, \beta^{14}$.

12.

Elemento	Polinomio minimal
0	x
1	x
β, β^2, β^4	$a + x + x^2$
$\beta^3, \beta^6, \beta^5$	$1 + x^2 + x^4$

13.

Elemento	Polinomio minimal
0	x
1	$1 + x$
β^5, β^{10}	$1 + x + x^2$
$\beta^7, \beta^{14}, \beta^{13}, \beta^{11}$	$1 + x + x^4$
$\beta, \beta^2, \beta^4, \beta^8$	$1 + x^3 + x^4$
$\beta^3, \beta^6, \beta^9, \beta^{12}$	$1 + x + x^2 + x^3 + x^4$

32. Pedir retransmisión; (b) 10; (c) 5 y 8; (d) 6 y 11; (e) y (f) pedir retransmisión; (g) 0 y 13; (h) es palabra-código.

VI. Códigos de Reed-Solomon

1. (a) 2^{15}; (b) $g(x) = \beta + \beta^3 x + x^2$; (c)(i) $\beta\beta\beta^6\beta^6000$; (d) $g_K(x) = (1+x)(\beta+x)(\beta^2+x)(\beta^4+x)$.

2. (a) 2^{44}; (b) $g(x) = \beta^{10} + \beta^3 x + \beta^6 x^2 + \beta^{13} x^3 + x^4$;

(c)(i) $\beta^{10}\beta^3\beta^6\beta^{13}100000\beta^2\beta^{10}\beta^{13}\beta^5\beta^7$;

(d) $g_K(x) = (\beta^8 + x)(\beta^6 + x)(\beta^{12} + x)(\beta^9 + x)g(x)$.

3. (a) β^2; (b) β^5; (c) β^4.

4. (a) $n = 3, k = 1, d = 3$ y $|C| = 4$; (b) $G = [\beta\beta^2 1]$; (c):

Mensaje	Palabra código	$f(c)$
0	0 0 0	000000
1	$\beta\beta^2 1$	011110
β	$\beta^2 1\beta$	111001
β^2	$1\beta\beta^2$	100111

5. (a) $m = 7, k = 3, d = 5$ y $|C| = 8^3 = 512$; (b) $g(x) = \beta^6 + beta^5 x + \beta^5 x^2 + \beta^2 x^3 + x^4$.
6. (a) $\beta + \beta^2 x + x^2 = (\beta^3 + x)(\beta^4 + x) = (1 + x)(\beta + x)$;
(b) $1 + \beta^6 + x^2 = (\beta^3 + x)(\beta^4 + x)$;
(c) $\beta^3 + \beta x + x^2 + \beta^3 x^3 + x^4 = (\beta + x)(\beta^2 + x)(\beta^3 + x)(\beta^4 + x)$;
(d) $\beta^{10} + \beta^3 x + \beta^6 x^2 + \beta^3 x^3 + x^4 = (\beta + x)(\beta^2 + x)(\beta^3 + x)(\beta^4 + x)$;
(e) $\beta^{21} + \beta^{24} x + \beta^{16} x^2 + \beta^{24} x^3 + \beta^9 x^4 + \beta^{10} x^5 + x^6 = (\beta + x)(\beta^2 + x) \cdots (\beta^6 + x)$.
8. (a) $00\beta\beta^5\beta^3\beta^2\beta^{13}\beta^{10}\beta 000000$;
(b) $1\beta^4\beta^2\beta\beta^{12}\beta^9 10\beta\beta^5\beta^3\beta^2\beta^{13}\beta^{10}\beta$;
(c) $\beta\beta^{10}\beta^7 0\beta^{12}\beta^3\beta^3 10000000$;
9. (a) $001\beta^8\beta^{11}\beta^3\beta^5 00000000$;
(b) $0\beta^{10}\beta^3\beta^6\beta^{13}0\beta^8\beta^{11}\beta^3\beta^5 00000$;
(c) $\beta^4\beta^{12}\beta^7 0\beta^2\beta^5\beta^{12}\beta^{14}000000$.
11. (a) $0\beta^2 0000000000000$;
(b) $00\beta 00\beta^3 000000000$;
(c) 100000000000000;
(d) $\beta^5 11100000000000$;
(e) $\beta^{10}\beta^3 0001000010000$;
(f) $\beta^2 0000\beta^2 0000\beta^2 0000$.

APÉNDICE B
El algoritmo de Euclides para polinomios

Suponemos al lector conocedor de la noción de máximo común divisor de dos enteros, no ambos nulos, así como del algoritmo de Euclides asociado. Veremos en este apéndice cómo estos conceptos se generalizan a polinomios.

El máximo común divisor (mcd) de dos polinomios $f(x), g(x) \in K[x]$, no ambos nulos, es el polinomio $d(x) \in K[x]$ de mayor grado tal que $f(x) = q_1(x)d(x)$ y $g(x) = q_2(x)d(x)$.

Denotaremos $d(x) = \text{mcd}(f(x), g(x))$.

Ejemplo. Hallamos el máximo común divisor de $f(x)$ y $g(x)$ suponiendo que conocemos las factorizaciones de $f(x)$ y de $g(x)$ en factores irreducibles, donde

$$f(x) = 1 + x^2 + x^3 + x^6 + x^7 + x^8 = (1 + x)(1 + x + x^3)(1 + x^4)$$

y

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

El polinomio de mayor grado que es factor común de ambos, $f(x)$ y $g(x)$, es $(1+x)(1+x+x^3) = 1 + x^2 + x^3 + x^4$. Por lo tanto,

$$\text{mcd}(f(x), g(x) = 1 + x^2 + x^3 + x^4.$$

Este no es el mejor método para hallar el máximo común divisor de dos polinomios. Presentamos a continuación un método más eficiente de llevar a cabo tal tarea cuando ambos polinomios son no nulos.

Algoritmo de Euclides. Dados $f(x), g(x) \in K[x]$ no nulos y con $\text{gr}(f(x)) \geq \text{gr}(g(x))$:

1. Inicializar colocando $r_0(x) = f(x)$, $r_1(x) = g(x)$ y $i = 1$.

2. Mientras $r_i(x) > 0$, dividir $r_i(x)$ por $r_{i-1}(x)$ y denotar el resto de esta división con $r_{i+1}(x)$, o sea que $r_{i+1}(x) = r_{i-1}(x) \mod r_i(x)$.

3. Si $r_i(x) = 0$, entonces colocar $\text{mcd}(f(x), g(x)) = r_{i-1}(x)$; caso contrario, incrementar i en una unidad y repetir el ítem 2.

Notar que este algoritmo debe parar luego de un número finito de pasos, pues para cada $i > 1$, el grado del resto $r_{i+1}(x)$ es menor que el grado de $r_i(x)$.

Podemos modificar este algoritmo para producir polinomios $t_i(x)$ y $s_i(x)$ en $K[x]$ tales que

$$t_i(x)f(x) + s_i(x)g(x) = r_i(x), \text{ para } i = 0, 1, \ldots$$

Definimos

$$\begin{aligned} t_0(x) &= 1, \quad t_1(x) = 0, \\ s_0(x) &= 0, \quad s_1(x) = 1. \end{aligned}$$

Suponiendo que $r_{i-1}(x) = q_i(x)r_i(x) + r_{r+1}(x)$ (usando el Algoritmo de División), definimos

$$\begin{aligned} t_i(x) &= q_{i-1}(x)t_{i-1}(x) + r_{i-2}(x) \\ s_i(x) &= q_{i-1}(x)s_{i-1}(x) + s_{i-2}(x), \text{ para } i = 2, \ldots \end{aligned}$$

Entonces,

$$\begin{aligned} r_j(x) &= (-1)^j[-t_j(x)r_0(x) + s_j(x)r_j(x)] \\ &= t_j(x)r_0(x) + s_j(x)r_1(x). \end{aligned}$$

Como estamos trabajando sobre el cuerpo binario, podemos ignorar el signo negativo.

Ejemplo. Usamos el Algoritmo de Euclides para hallar el máximo común divisor de los polinomios

$$\begin{aligned} f(x) &= x^2 + x^3 + x^6 + x^7, \\ g(x) &= 1 + x^3 + x^4 + x^5. \end{aligned}$$

Las computaciones son como sigue. Colocar $i = 0$, $r_0(x) = f(x)$ y $r_1(x) = g(x)$. Dividiendo $r_1(x)$ por $r_0(x)$ nos da

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

Luego, $r_2(x) = 1 + x^4$ y $q_2(x) = 1 + x^2$. Dividiendo $r_2(x)$ por $r_1(x)$ nos da

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

Luego $r_3(x) = x + x^3$ y $q_3(x) = 1 + x$. Siguiendo así:

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

Continuando:

$$x + x^3 = (1 + x^2)(x) + 0,$$

de modo que $r_5(x) = 0$. Como el último resto no nulo es $r_4(x) = 1 + x^2$, tenemos que $r_4(x)$ es el común divisor requerido de $f(x)$ y $g(x)$:

$$1 + x^2 = \text{mcd}(1 + x^3 + x^4 + x^5, x^2 + x^6 + x^7).$$

En este ejemplo, pudimos computar también $t_i(x)$ y $s_i(x)$ usando los cocientes $q_i(x)$ producidos en cada iteración, (ver tabla abajo). Afirmamos que para cada $i = 0, 1, 2, 3, 4$:

$$r_i(x) = t_i(x)f(x) + s_i(x)g(x).$$

Esto es claramente cierto para $i = 0, 1$ y 2. Para $i = 3$

$$\begin{aligned} r_3(x) = x + x^3 &= (1 + x)f(x) + (x + x^2 + x^3)g(x) \\ &= (1 + x)((x^2 + x^3 + x^6 + x^7) \\ &\quad +(x + x^2 + x^3)(1 + x^3 + x^4 + x^5), \\ \text{y } r_4(x) = 1 + x^2 &= (1 + x + x^2)f(x) + (1 + x^3 + x^4)g(x). \end{aligned}$$

Resumiendo:

i	$t_i(x)$	$s_i(x)$	$r_i(x)$
0	1	0	$f(x)$
1	0	1	$g(x)$
2	1	$1+x^2$	$1+x^4$
3	$1+x^2$	$x+x^2+x^3$	$x+x^3$
4	$1+x+x^2$	$1+x^3+x^4$	$1+x^2$
	–	–	0

Usando inducción, se puede probar lo siguiente:

Teorema 57. *Si $d(x) = \text{mcd}(f(x), g(x))$, entonces existen polinomios $t(x)$ y $s(x)$ en $K[x]$ tales que*

$$t(x)f(x) + s(x)g(x) = d(x).$$

Ejercicios.

B.1. Halle el máximo común divisor de cada uno de los siguientes pares de polinomios:

1. $f(x) = 1 + x + x^5 + x^6 + x^7, g(x) = 1 + x + x^3 + x^5$;
2. $f(x) = 1 + x^2 + x^3 + x^7, g(x) = 1 + x + x^3$;
3. $f(x) = 1 + x + x^4 + x^5 + x^8 + x^9, g(x) = 1 + x^2 + x^3 + x^7$;
4. $f(x) = 1 + x + x^2 + x^3 + x^4, g(x) = x + x^3 + x^4$.

B.2. Halle $\text{mcd}(f(x)), g(x))$, para $f(x) = 1 + x^9$ y $g(x)$ así dado:

1. $g(x) = x + x^2 + x^4 + x^5 + x^8$;
2. $g(x) = x^3 + x^6$;
3. $g(x) = 1 + x + x^2 + x^4 + x^5 + x^7 + x^8$;
4. $g(x) = 1 + x^3 + x^6$;
5. $g(x) = x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$.

B.3. Halle $\text{mcd}(f(x), g(x))$, para $f(x) = 1 + x^{15}$ y $g(x) = x + x^2 + x^4 + x^8$.

B.4. Halle $\text{mcd}(f(x), g(x))$, para $f(x) = 1 + x^{23}$ y

$$g(x) = x + x^2 + x^3 + x^4 + x^6 + x^8 + x^9 + x^{12} + x^{13} + x^{16} + x^{18}.$$

Referencias bibliográficas

[1] Herstein, I. N. (1990). *Abstract Algebra*, Nueva York: Macmillan.

[2] Hoffman, K. Kunze, R. (1971). *Linear Algebra*, Englewood Cliffs: Prentice-Hall Inc. Second Edition.

[3] Huffman, W. C., Pless, V. (2010). *Fundamentals of Error-Correcting Codes*, Cambridge University Press.

[4] MacWilliam, F. J., Sloane, N. J. A. (1977) *The Theory of Error-Correcting Codes*, Nueva York: Elsevier/North Holland.

[5] Friedberg, S., Insel, A., Spence, L. (2017). *Linear Algebra*, Englewood Cliffs: Prentice-Hall Inc. Second Edition.

[6] Malik, D. S., Mordeson, J. M., Sen, M. K. (1997). *Fundamentals of Abstract Algebra*, International series in Pure and Applied Mathematics, Nueva York: McGraw-Hill.

Sobre los autores

CARLOS ARAUJO MARTÍNEZ Licenciado en Matemáticas y Física de la Universidad del Atlántico. Magíster en Matemáticas y Doctor en Matemáticas de la Universidad de Puerto Rico. Profesor durante varios años en las áreas del álgebra y combinatoria en la Facultad de Ciencias Básicas de la Universidad del Atlántico, en las que ha desarrollado su producción científica. Además, ha sido docente de matemáticas por varios años en la escuela secundaria.

MIGUEL ANTONIO CARO CANDEZANO Ph. D. en Ciencias de la Computación y Matemáticas Computacionales de la Universidad de Sao Paulo-Brasil. Terminó el pregrado y la Maestría en Matemáticas Aplicadas e Informática en la Universidad Rusa de la Amistad de los Pueblos. Durante más de 20 años se desempeñó como profesor de computación del Instituto Experimental del Atlántico. Desde 1998 es profesor del Departamento de Matemáticas de la Universidad del Atlántico. Sus áreas principales de estudio son la mecánica de fluidos computacional, así como el análisis numérico. Ha llevado su experiencia en la enseñanza de la computación, con énfasis en la matemática en la escuela secundaria, a seminarios dirigidos a los docentes de educación media en esta área.

ITALO J. DEJTER Licenciado en Matemáticas de la Universidad de Buenos Aires (1967). Doctor en Matemáticas (Ph.D.) de Rutgers University (1975). Profesor en el Departamento de Matemáticas de la Universidad Federal de Santa Catarina, Brasil (1977-1984). Profesor en el Departamento de Matemáticas de la Universidad de Puerto Rico, Río Piedras (1984-2018). Retirado de la Universidad de Puerto Rico desde el 1 de marzo de 2018. Especialista en topología algebraica, teoría combinatoria, teoría de gráfos, teoría de códigos y diseños combinatorios. Investigador con más de 77 trabajos publicados en revistas científicas internacionales.