Universidad de El Salvador

Facultad de Ciencias Naturales y Matemática

Escuela de Matemática

"Códigos cíclicos y algunas aplicaciones"

Presentado por:
Br. Patricia Esmeralda Rodríguez Borja, RB12001

Asesor interno:
M.Sc. Ingrid Carolina Martínez Barahona
Universidad de El Salvador
Asesor externo:
Dr. Sergio Roberto López-Permouth
Ohio University

Ciudad Universitaria, 20 de septiembre del 2019

Autoridades

Universidad de El Salvador

Rector:

MSc. Roger Armando Arias Alvarado

Vice-Rector Administrativo:

Ing. Nelson Bernabé Granados

Secretario General:

Lic. Cristobal Hernán Ríos Benítez

Fiscal General:

Lic. Rafael Humberto Peña Marín

Facultad de Ciencias Naturales y Matemática

Decano:

Lic. Mauricio Hernan Lovo Córdova

Vice-Decano:

Lic. Carlos Antonio Quintanilla Aparicio

Secretaria:

Licda. Damaris Melany Herrera Turcios

Escuela de Matemática

Director:

Dr. José Nerys Funes Torres

Secretaria:

MSc. Alba Idalia Córdova Cuéllar

Agradecimientos

Antes que nada, agradezco a Dios por haberme dado fortaleza en todo este largo y arduo trayecto de desarrollarme como profesional y por haberme dado la oportunidad de finalizar una de las etapas más importante de mi vida.

A mis padres por su apoyo incondicional ante todas las adversidades y por darme la oportunidad de estudiar una carrera universitaria, por haber hecho de mí una persona de bien y ser así alguien útil a la sociedad.

A mis hermanos por confiar en mí y a mi novio Luisantos Bonilla por sus sabios consejos y recomendaciones, las cuales han permitido mejorar mi vida académica y personal.

A mis asesores: MSc. Ingrid Martínez y Dr. Sergio López-Permouth por tomarse el tiempo en asesorarme en todo el avance de mi trabajo con el fin de lograr un mejor entendimiento y desarrollo de los contenidos.

Al jurado calificador: Lic. Mario Carpio y M.Sc. René Palacios, por su dedicación en la revisión y corrección de mi trabajo.

Índice general

Resumen **5**

Introducción **6**

Metodología **7**

1. Códigos lineales y polinomios **8**
1.1. Nociones básicas sobre códigos 8
1.2. Códigos lineales 17
1.3. Generalidades sobre polinomios en K 23

2. Códigos cíclicos **35**
2.1. Polinomios y palabras de un código cíclico 35
2.2. El anillo de ideales principales K_n 48
2.3. Codificación y decodificación polinomial 51
2.4. Encontrando códigos cíclicos 55
2.5. Dual de un código cíclico 58

3. Código de Hamming y código BCH corrector de errores dobles **62**
3.1. Campos finitos 62
3.2. Polinomios mínimos 69
3.3. Código cíclico de Hamming 74
3.4. Códigos BCH 77
3.5. Decodificación del código BCH corrector de errores dobles 84

A. Implementación en Octave de C_{15} **88**

Conclusiones **96**

Bibliografía **97**

Resumen

La transmisión de datos se ha convertido en algo muy esencial y necesario en la actualidad. El problema de dicha transmisión radica en que los canales de comunicación pueden conducirnos a errores que generan daño en nuestros códigos y es por ello que la teoría de códigos busca formas eficientes de codificar información para que los errores antes mencionados puedan ser detectados e incluso corregidos.

Una clase muy importante de códigos autocorrectores, son los llamados *códigos cíclicos*, estos son un tipo de códigos lineales muy útiles para codificar y decodificar de manera eficiente. El álgebra lineal, álgebra moderna y la teoría de cuerpos finitos son piezas muy importantes y muy esenciales en los procesos de codificación y decodificación de los códigos, que desde el punto de vista práctico, permiten una implementación fácil.

Introducción

En el presente trabajo de investigación se estudia una clase muy importante de códigos lineales, los cuales son llamados *códigos cíclicos*. Para ello, tratamos los códigos de manera general en un campo K, su definición, ejemplos, conceptos básicos y propiedades que cumplen. Llegando así, a los códigos lineales, los cuales aparecen cuando un código satisface las condiciones de un subespacio vectorial y se caracterizan por tener una matriz de control de paridad y una matriz generadora. De aquí nacen los códigos cíclicos, cuyo propósito radica en la simplificación de los procesos de codificación y decodificación en los sistemas de comunicación.

Para el estudio de los códigos cíclicos, se realizó un breve repaso sobre generalidades de polinomios, de tal manera que pudiéramos representar los códigos como conjuntos de polinomios y así tener una mejor estructura matemática. Todo esto con la idea de definir adecuadamente los códigos cíclicos, los cuales satisfacen que la permutación de las posiciones de los dígitos, nos devuelve otra palabra del código (cerrado bajo cierta permutación). Esto nos llevó a encontrar el polinomio generador de un código cíclico y su dual.

De forma breve, tratamos a los códigos de Hamming, los cuales también son cíclicos y para los que se definió su matriz de control de paridad de tal manera que simplificaron muchos cálculos en la decodificación. Además tratamos un tipo especial de códigos cíclicos, los BCH, sin embargo, tratamos únicamente a los que corrigen a lo sumo 2 errores, a estos también se les definió su matriz de control de paridad en la que encontramos mecanismos para poder corregir eficazmente errores en la transmisión mediante el uso de los síndromes.

Para finalizar el estudio, se hizo una aplicación de los códigos BCH, mediante la implementación en un software matemático del algoritmo desarrollado en el último capítulo.

Metodología

1. La investigación realizada es del tipo bibliográfica de la cual se hizo uso de diferentes bibliografías, basándonos en un libro en especifico: **Coding Theory and Cryptography** de D.R. Hankerson, D.G. Hoffman, del cual se efectuó el desarrollo de los capítulos 4 y 5.

2. Se comprendieron y expusieron de manera detallada los resultados más importantes del tema, el cual comprende el desarrollo de los 3 capítulos siguientes:

   Capítulo 1. Códigos lineales y polinomios.

   Capítulo 2. Códigos cíclicos.

   Capítulo 3. Código de Hamming y BCH corrector de errores dobles.

3. Se implementaron los resultados obtenidos a través de algoritmos.

Capítulo 1

Códigos lineales y polinomios

1.1. Nociones básicas sobre códigos

Antes de comenzar el estudio de una clase de códigos muy importante, se introducirá una serie de conceptos básicos que son necesarios para una clara comprensión del mismo.

Definición 1.1.1. *Llamaremos **alfabeto** a un conjunto finito K, que está compuesto por símbolos.*

Ejemplo 1.1.2. $K = \{x_1, x_2, ..., x_n\}$ *es un alfabeto donde los* x_i *son símbolos.*

Definición 1.1.3. *Una **palabra** es una secuencia de símbolos.*

Ejemplo 1.1.4. $S = \{021, 1021, 200100\}$ *es un conjunto de palabras en el alfabeto* $K = \{0, 1, 2\}$.

Definición 1.1.5. *Un **código** es un conjunto* **C** *de palabras.*

Ejemplo 1.1.6. *El conjunto S del ejemplo anterior es un código.*

Nuestro interés será limitado a usar **códigos binarios**, cuyo alfabeto se compone únicamente de dos elementos $K = \{0, 1\}$. Note que este conjunto cumple ser un campo y sus inversos aditivos son ellos mismos.

Además, los códigos en los cuales todas sus palabras tienen la misma longitud se les conoce como **códigos de bloque** y para uso práctico, llamaremos únicamente **código** a los códigos de bloque binarios.

Definición 1.1.7. *El **tamaño** de una palabra x es el número de dígitos que tiene y se denotará por* $\ell(x)$.

Ejemplo 1.1.8. $\ell(10010) = 5 \quad y \quad \ell(101) = 3$ *(es decir, 10010 y 101 son palabras de longitud 5 y 3 respectivamente).*

Definición 1.1.9. *Llamaremos **canal binario** o solo **canal** al medio por el cual es enviada o transmitida la palabra.*

Definición 1.1.10. *Un **canal binario** se dice **simétrico**, si* 0 *y* 1 *son transmitidos con igual precisión. Es decir, la probabilidad de recibir el dígito correcto es independiente de cuál dígito,* 0 *o* 1 *se está transmitiendo.*

Los códigos *BCH* tienen la propiedad que con probabilidad p se recibirá un dígito que es transmitido correctamente (sea el enviado) y con probabilidad $q = 1 - p$ cuando el dígito recibido sea transmitido incorrectamente (no sea el enviado).

Por otro lado, se define de manera natural una operación suma y producto (módulo 2), para el conjunto de todas las palabras de longitud n. De modo que dicho conjunto, al cual describiremos como $K^n = \{(a_1, a_2, ..., a_n) : a_i \in K\}$, cumple con las propiedades fundamentales de espacio vectorial sobre el campo $K = \{0, 1\}$.

En K^n, la suma y el producto por escalar se define como sigue:

1. La suma se define componente a componente.

   Ejemplo 1.1.11. *Para el caso de* $n = 4$,

   $$(1, 0, 1, 0) + (1, 1, 1, 0) = (1 + 1, 0 + 1, 1 + 1, 0 + 0) = (0, 1, 0, 0).$$

2. El producto por escalar sobre el campo K, se define componente a componente.

   Ejemplo 1.1.12. *Para el caso de* $n = 3$,

   $$1 * (1, 0, 1) = (1 * 1, 1 * 0, 1 * 1) = (1, 0, 1).$$

El espacio vectorial K^n sobre el campo K, es un conjunto no vacío, dotado de una operación interna (suma) y una operación externa (producto por escalar), ambas cerradas:

Suma $+$**:** $K^n \times K^n \to K^n$; $\quad (u, v) \to w = u + v$

Operación interna que cumple:

- Propiedad conmutativa, es decir,
  $u + v = v + u, \quad \forall u, v \in K^n$.
- Propiedad asociativa, es decir,
  $u + (v + w) = (u + v) + w, \quad \forall u, v, w \in K^n$.
- Posee elemento neutro O (palabra cero en K^n) es decir,
  $\exists 0 \in K^n : u + 0 = u, \quad \forall u \in K^n$.

- Posee elemento inverso aditivo, es decir,
  $\forall u \in K^n,\ \exists w : u + w = 0.$
  Debido a que en K los inversos aditivos son ellos mismos, entonces esta propiedad es heredada a K^n.
  Por lo tanto $w = u$, lo que significa que cada palabra es su propio inverso.

Producto $*$: $K \times K^n \to K^n;\ (k, u) \to w = k * u$.
Operación externa que cumple:

- Propiedad asociativa:
  $a * (b * u) = (a.b) * u,\ \forall a, b \in K, \forall u \in K^n.$
- Existencia del elemento neutro multiplicativo 1 del cuerpo K.
  $\exists 1 \in K : 1 * u = u,\ \forall u \in K^n.$
- Propiedad distributiva del producto sobre la suma de vectores:
  $a * (u + v) = a * u + a * v,\ \forall a \in K, \forall u, v \in K^n.$
- Propiedad distributiva del producto sobre la suma de escalares:
  $(a + b) * u = a * u + b * u,\ \forall a, b \in K, \forall u \in K^n$

Definición 1.1.13. *El **peso de Hamming** o **peso** de una palabra v de longitud n, denotado por $wt(v)$, es el número de veces que el dígito 1 aparece en v.*

Proposición 1.1.14. *$\forall a \in K$ y $w, v \in K^n$, el peso de Hamming, cumple las siguientes propiedades:*

1. $0 \leq wt(v) \leq n \quad y \quad wt(v) = 0$ *si y sólo si* $v = \mathbf{0}$.
2. $wt(v + w) \leq wt(v) + wt(w)$.
3. $wt(a \cdot v) = a \cdot wt(v)$.

Demostración. Para la demostración de cada una de las propiedades anteriores, haremos uso de la *Definición 1.1.11*.

1. Sabemos que:
   Dado que $v \in K^n$, sabemos que $v = (a_1, a_2, ..., a_n)$, con $a_i \in K = \{0, 1\}$, $\forall i \in \{1, ..., n\}$.

   Entonces, como $a_i \geq 0$, $\forall a_i$:

$$\sum_{i=1}^{n} a_i \geq \sum_{i=1}^{n} 0 = 0. \tag{1.1}$$

Y por definición,

$$wt(v) = \sum_{i=1}^{n} a_i \geq 0. \tag{1.2}$$

Por otro lado, $a_i \leq 1$, $\forall a_i \in \{0, 1\}$, así:

$$\sum_{i=1}^{n} a_i \leq \sum_{i=1}^{n} 1 = n \cdot 1 = n. \tag{1.3}$$

Y por definición,

$$wt(v) = \sum_{i=1}^{n} a_i \leq n. \tag{1.4}$$

Por lo tanto, por (1.2) y (1.4) se concluye que: $0 \leq wt(v) \leq n$.

Ahora probaremos que dado $wt(v) = 0$, entonces $v = \mathbf{0}$.

($\Rightarrow$) Suponemos por contradicción que $v \neq 0$, es decir, que $\exists a_i \in K$ tal que $a_i \neq 0$.

Entonces,

$$\begin{aligned} wt(v) = \sum_{i=1}^{n} a_i &= a_1 + ... + a_i + ... + a_n. \\ &= 0 + ... + 0 + 1 + 0 + ... + 0. \\ &= 1 \neq 0. \quad (\rightarrow\leftarrow) \end{aligned}$$

Contradicción, dado que por hipótesis $wt(v) = 0$. La contradicción provino de suponer que $\exists a_i \in K$ tal que $a_i \neq 0$. Por tanto, si $wt(v) = 0$, entonces, $\forall a_i \in K$, $a_i = 0$, es decir, $v = \mathbf{0}$.

($\Leftarrow$) Ahora, asumiendo que $v = \mathbf{0}$, se probará que $wt(v) = 0$.

Sabemos que si $v = (a_1, ..., a_n) = \mathbf{0}$, entonces $a_i = 0$, $\forall a_i \in K$, $i = \{1, ..., n\}$.
Por lo tanto,

$$wt(v) = \sum_{i=1}^{n} a_i = \sum_{i=1}^{n} 0 = n \cdot 0 = 0. \tag{1.5}$$

Por lo tanto, $wt(v) = 0$ si y sólo si $v = \mathbf{0}$.

2. Por la simplicidad del campo $K = \{0, 1\}$, podemos notar que en la suma dentro de K, sucede únicamente lo siguiente:

$$\begin{cases} 0+0=0. \\ 0+1=1. \\ 1+0=1. \\ 1+1=0. \end{cases} \tag{1.6}$$

Es notorio ver que los posibles resultados únicamente son 0 y 1. Es por ello que podemos hacer lo que sigue a continuación:

Dados $v, w \in K^n$, i.e. $v = (a_1, ..., a_n)$ y $w = (b_1, ..., b_n)$. Tenemos:

$$wt(v+w) = wt(a_1+b_1, ..., a_n+b_n) = \sum_{i=1}^{n} a_i + b_i = \sum_{a_i+b_i=0} a_i + b_i + \sum_{a_i+b_i=1} a_i + b_i.$$

El sumatorio: $\sum_{a_i+b_i=0} a_i + b_i$, tiene a todos los ceros que son resultado de sumar un elemento de v y uno de w en la misma posición. Análogamente la suma: $\sum_{a_i+b_i=1} a_i + b_i$, tiene a todos los unos que son resultado de sumar un elemento de v y uno de w en la misma posición.

En el sistema de (1.6), se observa que la suma nos devuelve cero sólo si los dígitos que se están sumando son iguales, en el caso de 1+1=0 ($a_i = b_i = 1, \forall i$), tenemos que se cumple:

$$\sum_{a_i+b_i=0} a_i + b_i < \sum_{a_i+b_i=0} a_i + \sum_{a_i+b_i=1} b_i. \tag{1.7}$$

Lo anterior se puede visualizar mejor con un caso en particular. Si $v = (111)$ y $w = (111)$, los pesos de estas palabras de manera individual, corresponden a $wt(v) = 3$ y $wt(w) = 3$, de lo cual, al sumarlos nos da 6. Sin embargo, si encontramos el peso de la suma de estas palabras: $v + w = (000)$, claramente sabemos que $wt(v+w) = 0$.

Y fácilmente en el caso de $0 + 0 = 0$ ($a_i = b_i = 0, \forall i$), se cumple la igualdad:

$$\sum_{a_i+b_i=0} a_i + b_i = \sum_{a_i+b_i=0} a_i + \sum_{a_i+b_i=0} b_i. \tag{1.8}$$

Por lo tanto, de (1.7) y (1.8) se tiene que:

$$\sum_{a_i+b_i=0} a_i + b_i \leq \sum_{a_i+b_i=0} a_i + \sum_{a_i+b_i=0} b_i. \tag{1.9}$$

Por otro lado, para los casos en los que $1 + 0 = 1$ y $0 + 1 = 1$ ($a_i + b_i = 1, \forall i$), se cumple la igualdad:

Si $a_i = 1$ y $b_i = 0$, $\forall i$:

$$\sum_{a_i+b_i=1} a_i + b_i = \sum_{a_i+b_i=1} a_i + \sum_{a_i+b_i=1} b_i. \tag{1.10}$$

Analogamente, si $a_i = 0$ y $b_i = 1$, $\forall i$:

$$\sum_{a_i+b_i=1} a_i + b_i = \sum_{a_i+b_i=1} a_i + \sum_{a_i+b_i=1} b_i. \tag{1.11}$$

Ahora, teniendo en cuenta lo anterior, procedemos a demostrar lo que se pide.
Dados $v, w \in K^n$, donde $v = (a_1, a_2, ..., a_n)$ y $w = (b_1, b_2, ..., b_n)$ se tiene:

$$\begin{aligned}
wt(v+w) &= \sum_{a_i+b_i=0} (a_i + b_i) + \sum_{a_i+b_i=1} (a_i + b_i). \\
&= \sum_{a_i+b_i=0} (a_i + b_i) + \sum_{a_i+b_i=1} (a_i) + \sum_{a_i+b_i=1} (b_i);\ \text{por (1.11)} \\
&\leq \sum_{a_i+b_i=0} (a_i) + \sum_{a_i+b_i=0} (b_i) + \sum_{a_i+b_i=1} (a_i) + \sum_{a_i+b_i=1} (b_i);\ \text{por (1.10)} \\
&= \sum_{a_i+b_i=0} (a_i) + \sum_{a_i+b_i=1} (a_i) + \sum_{a_i+b_i=0} (b_i) + \sum_{a_i+b_i=1} (b_i). \\
&= \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i. \\
&= wt(v) + wt(w).
\end{aligned}$$

Por tanto, $wt(v+w) \leq wt(v) + wt(w)$.

3. Sea $v \in K^n$, es decir, $v = (a_1, ..., a_n)$ y $a \in K$.
Entonces,

$$a \cdot v = (aa_1, ..., aa_n). \tag{1.12}$$

$$wt(a \cdot v) = \sum_{i=1}^{n} a \cdot a_i = a \sum_{i=1}^{n} a_i = a \cdot wt(v). \tag{1.13}$$

Por lo tanto, $wt(a \cdot v) = a \cdot wt(v)$.

□

Definición 1.1.15. *El espacio vectorial K^n, con el peso de Hamming $wt(\cdot)$, es un **espacio vectorial normado**. En otras palabras el par $(K^n, wt(\cdot))$, es un espacio vectorial normado.*

Definición 1.1.16. *La **distancia de Hamming** o **distancia** entre dos palabras v y w, se define como $d(v,w) = wt(v+w)$; lo que significa que es el número de dígitos que no comparten en la misma posición.*

Proposición 1.1.17. *Sea w,v y u, palabras de longitud n, entonces la distancia de Hamming, cumple las siguientes propiedades:*

1. $0 \leq d(v,w) \leq n$.
2. $d(v,w) = 0$ *si y sólo si* $v = w$.
3. $d(v,w) = d(w,v)$.
4. $d(v,w) \leq d(v,u) + d(u,w)$.
5. $d(av,aw) = a.d(v,w)$.

Demostración.

1. Sean $v, w \in K^n$. Por la definición anterior sabemos que $d(v,w) = wt(v+w)$.

   Entonces, por la propiedad 1 del *peso de Hamming*, se tiene:

   $$0 \leq wt(v) \leq n, \forall v \in K^n. \tag{1.14}$$

   Y como $v + w \in K^n$,

   $$\Rightarrow 0 \leq wt(v+w) \leq n. \Rightarrow 0 \leq d(v,w) \leq n.$$

2. $\Rightarrow$ Para la primera implicación, se supone $d(v,w) = 0$:

   Por definición $d(v,w) = wt(v+w) = 0$ y por la propiedad 1 del *peso de Hamming* se tiene que:

   $$wt(v+w) = 0 \Leftrightarrow v + w = 0. \tag{1.15}$$

   Si en $v + w = 0$, sumamos w a ambos lados, entonces nos queda $v + w + w = 0 + w$, y como cada palabra es su propio inverso, entonces $w + w = 0$ y así: $v = w$.

   $\Leftarrow$ Ahora, si $v = w$ se tiene que $v + w = 0$, por (1.15).

   Así, $d(v,w) = wt(v+w) = wt(0)$, pero nuevamente por (1.15) se tiene que $wt(0) = 0$.

   Por tanto, $d(v,w) = 0$ si y sólo si $v = w$.

3. Usando la definición de distancia de Hamming:

$$d(v,w) = wt(v+w) = wt(w+v) = d(w,v).$$

4. Usando la propiedad 3 del peso de Hamming se tiene:

$$d(av,aw) = wt(av+aw) = wt(a(v+w)) = a \cdot wt(v+w) = a \cdot d(v,w).$$

Por lo tanto, se concluye que la distancia de Hamming es una métrica, i.e. (K^n, d) es un espacio métrico.

□

Definición 1.1.18. *La **distancia del código** C es la distancia mínima entre todas las palabras de código, tomadas dos a dos y sin que estas sean iguales; lo que significa:*

$$d = \text{mín}\{d(v,w) : v,w \in C, v \neq w\}.$$

Ejemplo 1.1.19. *Calcular el peso de cada una de las siguientes palabras y la distancia entre cada par de ellas: $v_1 = (1,0,0,1,0,1,0)$, $v_2 = (0,1,1,0,1,0,1)$ y $v_3 = (0,0,1,1,1,1,0)$.*

Solución.
Encontramos los pesos de cada una de las palabras.

$$wt(v_1) = wt((1,0,0,1,0,1,0)) = 3.$$
$$wt(v_2) = wt((0,1,1,0,1,0,1)) = 4.$$
$$wt(v_3) = wt((0,0,1,1,1,1,0)) = 4.$$

Ahora, las distancias dos a dos son:

$$\begin{aligned} d(v_1,v_2) &= wt(v_1+v_2) \\ &= wt((1,0,0,1,0,1,0)+(0,1,1,0,1,0,1)) \\ &= wt((1,1,1,1,1,1,1)) \\ &= 7. \end{aligned}$$

$$\begin{aligned} d(v_2,v_3) &= wt(v_2+v_3) \\ &= wt((0,1,1,0,1,0,1)+(0,0,1,1,1,1,0)) \\ &= wt((0,1,0,1,0,1,1)) \\ &= 4. \end{aligned}$$

$$\begin{aligned} d(v_1,v_3) &= wt(v_1+v_3) \\ &= wt((1,0,0,1,0,1,0)+(0,0,1,1,1,1,0)) \\ &= wt((1,0,1,0,1,0,0)) \\ &= 3. \end{aligned}$$

Definición 1.1.20. ***Codificación*** *es el método que permite convertir un carácter de un lenguaje natural (como el de un alfabeto o silabario) en un símbolo de otro sistema de representación, como un número o una secuencia de pulsos eléctricos en un sistema electrónico.*

Definición 1.1.21. ***Decodificación*** *es el proceso en el cual el receptor transforma el código utilizado por el emisor al lenguaje natural. De esta forma los signos son asociados a las ideas que el emisor trató de comunicar.*

En la decodificación, cuando recibimos una palabra $w \in K^n$, se verifica si esta pertenece o no al código; cuando la palabra w pertenece a C, entonces se asume que dicha palabra es la que se envió. Sin embargo, cuando esto no ocurre, se verifica cuál es la palabra del código que difiere una mínima cantidad de dígitos con la palabra w, es decir, se pretende buscar la palabra en C, que cumple que la distancia mínima entre dicha palabra y w sea mínima.

Pero por otro lado, en dicho proceso, puede ocurrir que se encuentre más de una palabra que cumpla tener una misma distancia mínima, entonces es ahí donde se habla acerca de los tipos de decodificación: IMLD o CMLD, los cuales consisten en pedir retransmisión o elegir una de esas posibles palabras que están a una misma distancia con w, respectivamente.

Por lo tanto, es necesario aclarar el tipo de decodificación a usar en el resto de la investigación. La cual será del tipo IMLD. Teniendo en cuenta que este método no siempre es factible en caso que ocurran demasiados errores en un canal BSC, ya que se tendrían muchas retransmisiones en caso de que hayan muchas palabras que tengan más de una palabra de código con la misma distancia mínima con w.

1.2. Códigos lineales

Definición 1.2.1. *Un código C es llamado **código lineal**, si $\forall u, v \in C$ la palabra $u + v$ es una palabra del código de C.*

Ejemplo 1.2.2. *El código $C = \{(0,0,0,0), (0,0,1,0), (0,1,0,0), (0,1,1,0)\}$ es un código lineal.*

Definición 1.2.3. *Sea $v = (a_1, \ldots, a_n)$ y $w = (b_1, \ldots, b_n)$ palabras de K^n. Definimos el **producto escalar** o **producto punto** $v \cdot w$ de v y w como:*

$$v \cdot w = a_1 b_1 + \ldots + a_n b_n.$$

Definición 1.2.4. *Dos palabras v, u en K^n se dice que son **ortogonales**, si $v \cdot u = 0$.*

Ejemplo 1.2.5. *Las palabras $v = (1,0,1)$ y $w = (1,1,1)$ son ortogonales.*

$$v \cdot w = 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 1 = 0.$$

Definición 1.2.6. *Sea S subconjunto de K^n, al conjunto de todas las palabras ortogonales a S se denota como $S^\perp$ y se le llama **complemento ortogonal** a S.*

Teorema 1.2.7. *Un código lineal C de dimensión k contiene 2^k palabras de código.*

Demostración.
Supongamos que B es una base para el código C. Entonces $B = \{b_1, b_2, ..., b_k\}$, dado que C tiene dimensión k.

Por definición de base, tenemos que cada palabra $v = (a_1, ..., a_n)$ en C, puede ser escrita como combinación lineal de elementos de la base, es decir:

$$v = a_1 b_1 + a_2 b_2 + \cdots + a_k b_k, \qquad a_i \in K, \forall i \in \{1, 2, \ldots, k\}.$$

Debido a que los a_i son elementos de $K = \{0, 1\}$, tendremos que para cada uno de los k sumandos en la expresión anterior, hay dos opciones de ser, 0 ó 1, y así, por el principio de la multiplicación tendremos 2^k palabras de código en C. □

Teorema 1.2.8. *Sea $C =< S >$ un código lineal generado por S de K^n. Entonces:*

$$dim(C) + dim(C^\perp) = n.$$

Demostración.
Supongamos que el código C tiene dimensión k.

1. Si $k = 0$, significa que en nuestro código C solo está la palabra cero de longitud n, $C = \{\mathbf{0}\}$ y por consiguiente $C^\perp = K^n$, donde K^n es todo el espacio vectorial.

2. Si $0 < k < n$, se forma una matriz M de tal manera que cada fila de M sea una palabra del conjunto S. Como las palabras de S tienen longitud n, entonces se tendrán n columnas para dicha matriz, luego se procede a resolver el sistema homogéneo $Mx^T = \mathbf{0}$, donde x^T es la transpuesta de la palabra solución. De aquí, como se sabe que la $dim(C) = k$, entonces en el sistema habrán k filas linealmente independientes y $n-k$ filas linealmente dependientes, por tanto al resolverlo, se encuentra la palabra solución x^T en términos de las variables independientes, y se separa dicho vector en $n-k$ sumandos con respecto a cada variable independiente, de lo que se sabe que por cada sumando, se encuentran los vectores que forman una base para nuestro código dual $C^\perp$, y por tanto, $dim(C^\perp) = n-k$. Y entonces se cumple que $k + (n-k) = n$.

3. Si $k = n$, el código C es todo K^n y por consiguiente, el dual $C^\perp$ tiene únicamente a la palabra cero, $C^\perp = \{\mathbf{0}\}$ y eso implica que $\dim(C^\perp) = 0$. Y se cumple la igualdad.

□

Definición 1.2.9. *Una matriz H es llamada **matriz de control de paridad** para un código lineal C, si las columnas de H forman una base para el código dual $C^\perp$.*

Teorema 1.2.10. *Si H es la matriz de control de paridad de C un código lineal (n,k,d), entonces C consiste de todas las palabras $v \in K^n$ tal que $vH = \mathbf{0}$.*

Demostración. Sea $v \in K^n$ con $v \notin C$ tal que $vH = \mathbf{0}$, y sean $h_1, h_2, \ldots, h_{n-k}$ las columnas de H. Por definición de matriz de control de paridad, $\{h_1, h_2, \ldots, h_{n-k}\}$ forman una base para el dual de C. Entonces:

$$vH = v\begin{bmatrix} h_1 & h_2 & \ldots & h_{n-k} \end{bmatrix} = (v \cdot h_1, v \cdot h_2, \ldots, v \cdot h_{n-k}) = (0, 0, \ldots, 0).$$

Entonces $v \cdot h_i = 0$, $\forall i \in \{1, ..., n-k\}$, por lo tanto, v es ortogonal a cada uno de los h_i, es decir, es ortogonal a la base de $C^\perp$, pero si es ortogonal a la base del dual, entonces también lo será para $C^\perp$, así $v \in (C^\perp)^\perp$, pero el dual del dual de C es el mismo C, entonces $v \in C$, lo cual es una contradicción, ya que se asumió que $v \notin C$. Por lo tanto, C está conformado por todas las palabras $v \in K^n$ tales que $vH = \mathbf{0}$. □

Definición 1.2.11. *Sea C un código lineal de K^n y sea $u \in K^n$, definimos la **clase lateral de C determinado por u** como el conjunto de todas las palabras de la forma $v + u$, para cada $v \in C$. Esta clase lateral se denota por $C + u$, así:*

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

Teorema 1.2.12. *Sea C un código lineal (n,k,d) y sean u y v palabras de tamaño n. Entonces las clases laterales cumplen:*

1. *Si $u \in C + v$, entonces $C + u = C + v$; es decir, cada palabra en la clase lateral determina esa clase lateral.*

2. *La palabra u está en la clase lateral $C + u$.*

3. *Si $u + v \in C$, entonces u y v están en la misma clase lateral.*
4. *Si $u + v$ no está en C, entonces u y v están en diferentes clases laterales.*
5. *Cada palabra en K^n está contenida en uno y solo una clase lateral de C; es decir, ya sea $C + u = C + v$, o $C + u$ y $C + v$ no tienen palabras en común.*
6. *$|C + u| = |C|$; es decir, el número de palabras en una clase lateral de C es igual al número de palabras en el código C.*
7. *Si C tiene la dimensión k, entonces existen exactamente 2^{n-k} diferentes clases laterales, y cada clase lateral contiene exactamente 2^k palabras.*
8. *El código C es también una clase lateral.*

Demostración.

1. Dado $u \in C + v$, se reescribe como $u = w + v$ para algún $w \in C$, así:

$$\begin{aligned} C + u &= \{h + u : h \in C\} \\ &= \{h + w + v : h \in C\} \\ &= \{(h + w) + v : h \in C\} \\ &= \{r + v : r \in C\}, si\ r = h+w \\ &= C + v. \end{aligned}$$

   Por lo tanto $C + u = C + v$.

2. Como el código C es lineal, se tiene que $\mathbf{0} \in C$, entonces $u = \mathbf{0} + u \in C + u$ y por lo tanto $u \in C + u$.
3. Si $u + v \in C$, entonces existe $w \in C$ tal que $u + v = w$, lo que es igual a escribir $u = w + v$, esto significa que $u \in C + v$ y usando la propiedad (1) se tiene: Si $u \in C + v \Rightarrow C + u = C + v$.
4. Por contradicción se debe suponer que u y v pertenecen a la misma clase lateral, así $u, v \in C + w$, entonces existen $h, r \in C$, tales que $u = h + w$ y $v = r + w$ y haciendo la suma de u y v, se tiene:

$$\begin{aligned} u + v &= (h + w) + (r + w) \\ &= h + r + (w + w) \\ &= h + r. \end{aligned}$$

   De aquí, $h + r \in C \Rightarrow u + v \in C$.

   Contradicción.. Porque por hipótesis tenemos que la suma no está en C. Por lo tanto, u y v están en distintas clases laterales.

5. Se sabe que dos clases laterales que están determinadas por u y v tales que $u \neq v$, satisfacen lo siguiente:

$$(C+u) \cap (C+v) = \varnothing \qquad \text{o} \qquad (C+u) \cap (C+v) \neq \varnothing.$$

Caso 1. Si $(C+u) \cap (C+v) = \varnothing$. $\Rightarrow$ Las clases laterales $C+u$ y $C+v$ no tienen palabras en común.

Caso 2. Si $(C+u) \cap (C+v) \neq \varnothing$. $\Rightarrow$ Existe w tal que $w \in C+u$ y $w \in C+v$ y aplicando el resultado del literal (a) tendremos:

$$C+u = C+w = C+v \Rightarrow C+u = C+v.$$

6. Para demostrar que la cardinalidad de $C+u$ es la misma que la de C, se probará que existe una biyección entre ambos conjuntos:

Sea

$$\begin{array}{rccc} f: & C & \longrightarrow & C+u \\ & r & \mapsto & f(r) = r+u. \end{array}$$

Se verá a continuación que está bien definida:

Dados r_1 y r_2 en C:

$$\text{Si } r_1 = r_2 \Rightarrow f(r_1) = r_1 + u = r_2 + u = f(r_2).$$

Por lo tanto, f está bien definida.

Se probará que la función f así definida, es biyectiva.

Inyectividad: Sean $f(r_1)$ y $f(r_2)$ en C tales que $f(r_1) = f(r_2)$, entonces $r_1 + u = r_2 + u$, sumamos la palabra u y por tanto $r_1 = r_2$.

Así, f es inyectiva.

Sobreyectividad: Sea $w \in C+u$, entonces $w = r_1 + u$, $r_1 \in C$.

Si tomamos a $r_1 = w + u$, se tiene que $f(r_1) = r_1 + u = w$. Entonces, $\exists r_1 \in C$ tal que $f(r_1) = w$.

Así, f es sobreyectiva.

$\therefore f$ es biyectiva.

Además, como C y $C+u$ son subconjuntos del conjunto finito discreto K^n, ambos subconjuntos son finitos y por lo tanto teniendo en cuenta que existe una biyección entre conjuntos finitos, entonces $|C+u| = |C|$.

7. Se sabe que K^n contiene 2^n palabras, además la cantidad de palabras que conforman una clase lateral, está determinada por la cardinalidad de C (por los ítems anteriores), de donde nosotros podemos deducir que si C tiene un máximo de 2^k palabras de código, entonces para averiguar cuántas clases laterales diferentes existen, basta hacer la división siguiente:

$$\frac{2^n}{2^k} = 2^{n-k}.$$

8. Si tomamos la clase lateral que está determinada por $\mathbf{0} \in K^n$ se tiene:

$$\begin{aligned} C + \mathbf{0} &= \{v + \mathbf{0} : v \in C\} \\ &= \{v : v \in C\} \\ &= C. \end{aligned}$$

   Por lo tanto C es una clase lateral.

□

Definición 1.2.13. *Sea C un código lineal (n, k, d) y H la matriz de control de paridad de C, entonces para la palabra $w \in K^n$ llamaremos **síndrome** de w a la palabra wH en K^{n-k}.*

Definición 1.2.14. *La **distancia** de un código lineal C es igual al peso mínimo de alguna palabra de código distinta de la palabra cero.*

Teorema 1.2.15. *Sea C un código lineal (n, k, d) y H la matriz de control de paridad de C. Sean w y u palabras en K^n. Entonces se verifica:*

1. *$wH = \mathbf{0}$ si y sólo si w es una palabra de código en C.*
2. *$wH = uH$ si y sólo si w y u se encuentran en la misma clase lateral de C.*
3. *Si u es el patrón de error para una palabra recibida w, entonces uH es la suma de las filas de H que corresponden a las posiciones en las que se produjeron errores en la transmisión.*

Demostración.

1. ($\Rightarrow$) Por el *Teorema 1.2.10*, se sabe que C contiene a todas las palabras de $v \in K^n$ tales que $vH = \mathbf{0}$, entonces como w cumple $wH = \mathbf{0}$, entonces $w \in C$.

   ($\Leftarrow$) Si $w \in C$ entonces, dado que H es la matriz de control de paridad de C, se cumple que $wH = \mathbf{0}$.

2. ($\Rightarrow$)

$$wH = uH \Rightarrow wH + uH = \mathbf{0} \Rightarrow (w+u)H = \mathbf{0}.$$

Y por el ítem (1) se sabe que si $(w+u)H = \mathbf{0}$, entonces $w+u \in C$, y por el ítem (3) del *Teorema 1.2.12*, w y u están en la misma clase lateral de C.

($\Leftarrow$) Si w y u están en la misma clase lateral de C, se tiene que $w \in C+u$, entonces existe $h \in C$ tal que $w = h+u$, y al despejar h se tiene:

$$w + u = h \in C \Rightarrow w + u \in C.$$

Y por el ítem (1):

$$(w+u)H = \mathbf{0} \Rightarrow wH + uh = \mathbf{0} \Rightarrow wH = uH.$$

3. Como u es el patrón de error para la palabra recibida w, este nos dice las posiciones en las cuales los dígitos de nuestra palabra w han cambiado, entonces cuando se efectúa uH, estamos multiplicando u por las filas de H, lo que nos dice que en las posiciones donde u tenga ceros, el multiplicar hará que las filas de H correspondientes a las posiciones de u donde hay ceros, se anulen y por tanto, únicamente quedaría la suma de las filas de H que corresponden a las posiciones de u en donde se produjo error en la transmisión.

□

1.3. Generalidades sobre polinomios en K

En este apartado usaremos notación polinomial para representar palabras, debido a que esto nos ayudará a manipular mejor nuestros códigos.

Definición 1.3.1. *Un polinomio de grado n sobre K, en la variable x, es un polinomio de la forma* $a_o + a_1x + ... + a_nx^n$, *donde los coeficientes* $a_0, ..., a_n$ *son elementos de K y* $a_n \neq 0$.

Denotaremos por $K[x]$ al conjunto de todos los polinomios sobre K.

Además, la suma y el producto de estos se definen de manera usual, como se ve en los cursos de álgebra. Sin embargo, en nuestro estudio, el campo a utilizar esta conformado únicamente por dos elementos, de los cuales hay que tener en cuenta que 1+1=0.

Ejemplo 1.3.2. *Sea* $f(x) = x^3 + x^2 + 1$ *y* $g(x) = x^3 + 1$

Solución

1. $$\begin{aligned} f(x) + g(x) &= (x^3 + x^2 + 1) + (x^3 + 1) \\ &= (x^3 + x^3) + x^2 + (1 + 1) \\ &= x^2. \end{aligned}$$

2. $$\begin{aligned} f(x)g(x) &= (x^3 + x^2 + 1)(x^3 + 1) \\ &= x^3(x^3 + 1) + x^2(x^3 + 1) + 1(x^3 + 1) \\ &= x^6 + x^3 + x^5 + x^2 + x^3 + 1 \\ &= x^6 + x^5 + x^2 + 1. \end{aligned}$$

Como podemos notar, ambos polinomios son de grado 3, sin embargo, la suma (1) es de grado 2, lo que nos dice que no necesariamente el grado de la suma coincide con ser el $max\{deg\ f(x), deg\ g(x)\}$. Por tanto, se tendrá en cuenta que $x^s + x^s = 0$, $\forall s$.

Ejemplo 1.3.3. *Sea* $f(x) = 1 + x$. *Encuentre:*

a) $(f(x))^2$.

b) $(f(x))^3$.

c) $(f(x))^4$.

Solución

a) $(f(x))^2 = (1 + x)^2 = 1 + 2x + x^2 = 1 + x^2$.

b) $(f(x))^3 = (1 + x)^3 = 1 + 3x + 3x^2 + x^3 = 1 + x + x^2 + x^3$.

c) $(f(x))^4 = (1 + x)^4 = 1 + x^4$.

De lo anterior, nos preguntamos si existe una regla especial en $K[x]$ para $(f(x)+g(x))^n$, con n un entero positivo.

Para analizar esto, se desarrollarán los primeros 8 casos para poder deducir si existe alguna regla, dado un $n \in Z^+$; exceptuando el caso trivial cuando $n = 1$.

- Para $n = 2$.
$$\begin{aligned}(f(x)+g(x))^2 &= (f(x))^2 + 2f(x)g(x) + (g(x))^2 \\ &= (f(x))^2 + (g(x))^2.\end{aligned}$$

- Para $n = 3$.
$$\begin{aligned}(f(x)+g(x))^3 &= (f(x)+g(x))(f(x)+g(x))^2 \\ &= (f(x)+g(x))(f(x))^2 + (g(x))^2 \\ &= ((f(x))^3 + f(x)(g(x))^2 + (f(x))^2 g(x) + (g(x))^3.\end{aligned}$$

- Para $n = 4$.
$$\begin{aligned}(f(x)+g(x))^4 &= (f(x)+g(x))^2(f(x)+g(x))^2 \\ &= ((f(x))^2 + (g(x))^2)((f(x))^2 + (g(x))^2) \\ &= ((f(x))^2 + (g(x))^2)^2 \\ &= (f(x))^4 + (g(x))^4.\end{aligned}$$

- Para $n = 5$.
$$\begin{aligned}(f(x)+g(x))^5 &= (f(x)+g(x))(f(x)+g(x))^4 \\ &= (f(x)+g(x))((f(x))^4 + (g(x))^4) \\ &= (f(x))^5 + f(x)(g(x))^4 + g(x)(f(x))^4 + (g(x))^5.\end{aligned}$$

- Para $n = 6$.
$$\begin{aligned}(f(x)+g(x))^6 &= (f(x)+g(x))^2(f(x)+g(x))^2(f(x)+g(x))^2 \\ &= ((f(x))^2 + (g(x))^2)((f(x))^2 + (g(x))^2)((f(x))^2 + (g(x))^2) \\ &= ((f(x))^2 + (g(x))^2)^2((f(x))^2 + (g(x))^2) \\ &= ((f(x))^4 + (g(x))^4)((f(x))^2 + (g(x))^2) \\ &= (f(x))^6 + (f(x))^4(g(x))^2 + (g(x))^4(f(x))^2 + (g(x))^6.\end{aligned}$$

- Para $n = 7$.
$$\begin{aligned}(f(x)+g(x))^7 &= (f(x)+g(x))(f(x)+g(x))^2((f(x)) + (g(x)))^4 \\ &= (f(x)+g(x))((f(x))^2 + (g(x))^2)((f(x))^4 + (g(x))^4) \\ &= (f(x)+g(x))(f(x))^6 + (f(x))^4(g(x))^2 + (g(x))^4(f(x))^2 + (g(x))^6 \\ &= f(x))^7 + (f(x))^5(g(x))^2 + (g(x))^4(f(x))^3 + f(x)(g(x))^6 \\ &+ f(x))^6 g(x) + (f(x))^4(g(x))^3 + (g(x))^5(f(x))^2 + (g(x))^7.\end{aligned}$$

- Para $n = 8$.

$$\begin{aligned}(f(x)+g(x))^8 &= (f(x)+g(x))^4(f(x)+g(x))^4\\ &= ((f(x))^4+(g(x))^4)((f(x))^4+(g(x))^4)\\ &= ((f(x))^4+(g(x))^4)^2\\ &= (f(x))^8+(g(x))^8.\end{aligned}$$

En conclusión, podemos deducir que existe una regla especial para $(f(x)+g(x))^n$, cuando n es tal que $n = 2^k, \forall k \in \mathbb{Z}^+$.

Se procederá a demostrar lo anterior por inducción.

Ya se analizaron varios casos base, para $n = 2$, $n = 2^2 = 4$ y $n = 2^3 = 8$.

Suponemos que se cumple para $\mathbf{n = 2^k}$. Es decir,

$$(f(x)+g(x))^{2^k} = (f(x))^{2^k}+(g(x))^{2^k}.$$

Ahora, se probará para $\mathbf{n = 2^{k+1}}$.

$$(f(x)+g(x))^{2^{k+1}} = (f(x))^{2^{k+1}}+(g(x))^{2^{k+1}}.$$

Pero, por hipótesis inductiva:

$$\begin{aligned}(f(x)+g(x))^{2^{k+1}} &= ((f(x)+g(x))^{2^k})^2\\ &= ((f(x))^{2^k}+(g(x))^{2^k})^2\\ &= ((f(x))^{2^k})^2+((g(x))^{2^k})^2\\ &= (f(x))^{2^k\cdot 2}+(g(x))^{2^k\cdot 2}\\ &= (f(x))^{2^{k+1}}+(g(x))^{2^{k+1}}.\end{aligned}$$

Por lo tanto, queda demostrado el siguiente Teorema:

Teorema 1.3.4. *Sea n tal que $n = 2^k$, con $k \in \mathbb{Z}^+$ y para cualesquiera polinomios $f(x)$ y $g(x)$ en $K[x]$, se cumple:*

$$(f(x)+g(x))^n = (f(x))^n+(g(x))^n.$$

Ejemplo 1.3.5. *Encuentre el número de polinomios sobre K de grado a lo sumo 10.*

Solución.

Para encontrar la cantidad de polinomios de grado a lo sumo 10, sobre K, lo podemos hacer para cada n. Es decir:

- Para $n = 0$, nos referimos a los polinomios de grado cero que en nuestro caso serían únicamente 2, $0x^0 = 0 \quad$ y $\quad 1x^0 = 1$.
- Para $n = 1$, los polinomios serian de la forma $a_o + a_1x$, donde $a_o, a_1 \in K = \{0, 1\}$, entonces necesitamos todas las posibles permutaciones de 0 y 1 en las posiciones de a_o y a_1, esto lo hacemos mediante el principio de la multiplicación.

  $$\underline{\ 2\ } \times \underline{\ 2\ } = 4 = 2^2.$$

  Por tanto, existen 4 polinomios sobre K, de grado a lo sumo 1.
- Para $n = 2$, hacemos lo mismo que en el caso anterior. Dado que la forma de los polinomios es $a_o + a_1x + a_3x^2$, encontramos todas las posibles permutaciones de 0 y 1 en las posiciones de a_o, a_1 y a_3.

  $$\underline{\ 2\ } \times \underline{\ 2\ } \times \underline{\ 2\ } = 8 = 2^3.$$

  Y por tanto, existen 8 polinomios de grado a lo sumo 2 sobre K.

De lo anterior, deducimos que un polinomio de grado a lo sumo n tiene $n + 1$ posiciones para sus coeficientes y por cada espacio podemos usar 2 elementos (0 y 1) y por el principio de la multiplicación encontramos que la expresión que nos devuelve la cantidad de polinomios de grado a lo sumo n es 2^{n+1}.

Así, $f(n) = 2^{n+1}$ nos devuelve el número de polinomios posibles de grado a lo sumo n en $K = \{0, 1\}$.

Entonces, para $n = 10$, $f(10) = 2^{10+1} = 2^{11} = 2048$.

Por tanto, la cantidad total de polinomios de grado a lo sumo 10 es de: 2048.

Ejemplo 1.3.6. *Liste todos los polinomios sobre K de grado n, para $n = 2$ y $n = 3$.*

Solución
*Para $n = 2$, tenemos que $f(2) = 2^3 = 8$, i.e. 8 polinomios de grado a lo sumo 2, luego quitamos los polinomios de grado a lo sumo 1: $f(1) = 2^2 = 4$, así: $8 - 4 = 4$.

palabra	*polinomio*
001	$0 + 0x + 1x^2 = \boldsymbol{x^2}$
011	$0 + 1x + 1x^2 = \boldsymbol{x + x^2}$
101	$1 + 0x + 1x^2 = \boldsymbol{1 + x^2}$
111	$1 + 1x + 1x^2 = \boldsymbol{1 + x + x^2}$

*Para $n = 3$, tenemos que $f(3) = 16$ polinomios, quitamos $f(2) = 8$: $16 - 8 = 8$.

palabra	*polinomio*
0001	$0 + 0x + 0x^2 + 1x^3 = \boldsymbol{x^3}$
0011	$0 + 0x + 1x^2 + 1x^3 = \boldsymbol{x^2 + x^3}$
0101	$0 + 1x + 0x^2 + 1x^3 = \boldsymbol{x + x^3}$
0111	$0 + 1x + 1x^2 + 1x^3 = \boldsymbol{x + x^2 + x^3}$
1001	$1 + 0x + 0x^2 + 1x^3 = \boldsymbol{1 + x^3}$
1011	$1 + 0x + 1x^2 + 1x^3 = \boldsymbol{1 + x^2 + x^3}$
1101	$1 + 1x + 0x^2 + 1x^3 = \boldsymbol{1 + x + x^3}$
1111	$1 + 1x + 1x^2 + 1x^3 = \boldsymbol{1 + x + x^2 + x^3}$

La división de polinomios en álgebra es un algoritmo que nos permite dividir un polinomio por otro que no sea nulo y este funciona para cualquier polinomio sobre los racionales y por tanto en K.

Algoritmo 1.3.7. *(Algoritmo de la División) Sean $f(x)$ y $h(x)$ polinomios en $K[x]$, con $h(x) \neq 0$. Entonces existen los polinomios únicos $q(x)$ y $r(x)$ en $K[x]$ tal que:*

$$f(x) = h(x)q(x) + r(x)$$

*donde $r(x) = 0$ ó $deg(r(x)) < deg(h(x))$. El polinomio $q(x)$ es llamado **cociente**, y $r(x)$ es llamado el **residuo**.*

El algoritmo de la división funciona debido a que en nuestro campo, los inversos aditivos son ellos mismos, es decir $x^s + x^s = 0$, $\forall s \in \mathbb{N}$. Además el proceso para encontrar el cociente y el residuo, es el que nosotros conocemos de división larga. Veamos el siguiente ejemplo:

Ejemplo 1.3.8. *Sea $f(x) = x + x^2 + x^6 + x^7 + x^8$ y $h(x) = 1 + x + x^2 + x^4$. Entonces:*

$$\begin{array}{l|l}
x^8 + x^7 + x^6 \qquad\qquad + x^2 + x & x^4 + x^2 + x + 1 \\
\cline{2-2}
x^8 \quad + x^6 + x^5 + x^4 & x^4 + x^3 \\
\cline{1-1}
\quad x^7 \quad + x^5 + x^4 \quad + x^2 + x & \\
\quad x^7 \quad + x^5 + x^4 + x^3 & \\
\cline{1-1}
\qquad\qquad\qquad\qquad x^3 + x^2 + x. &
\end{array}$$

De lo anterior, el polinomio $f(x)$ lo podemos escribir de la siguiente manera:

$$f(x) = h(x)(x^3 + x^4) + (x + x^2 + x^3).$$

$$x + x^2 + x^6 + x^7 + x^8 = (1 + x + x^2 + x^4)(x^3 + x^4)+(x + x^2 + x^3).$$

Y notemos que $grad(r(x)) < grad(h(x))$.

Recordemos que la razón por la cual se repasan estos principios, es para poder manejar de manera más fácil nuestros códigos. Es por ello que dado un polinomio de la forma $f(x) = a_o + a_1x + a_2x^2... + a_{n-1}x^{n-1}$, de grado $n-1$, este puede tratarse como la palabra $v = (a_0, a_1, ..., a_{n-1})$ de longitud n en nuestro espacio K^n.

Entonces, es necesario hacer la siguiente aclaración. En general, teniendo en cuenta el algoritmo de la división, el polinomio $r(x)$ siempre tendrá grado a lo sumo $n-1$, si $h(x)$ es de grado n; lo que significa que la palabra que corresponde a nuestro polinomio $r(x)$ se encuentra en nuestro espacio vectorial K^n.

Revisando el *ejemplo 1.3.8*, el polinomio $h(x) = 1 + x + x^2 + x^4$ es de grado 4, entonces el polinomio $r(x) = x + x^2 + x^3$ que es de grado a lo sumo 3, corresponde a la palabra (0,1,1,1) que pertenece al espacio vectorial K^4.

Observemos la siguiente tabla. Los polinomios de la izquierda se traducen a las palabras de la derecha si suponemos que el espacio de trabajo es K^7.

polinomio	*palabra*
$1 + x + x^2 + x^4$	(1,1,1,0,1,0,0)
$1 + x^4 + x^5 + x^6$	(1,0,0,0,1,1,1)
$1 + x + x^3$	(1,1,0,1,0,0,0)

Resumen.
Para representar un polinomio a su respectiva palabra, es necesario saber el espacio vectorial en el que se está trabajando, ya que puede suceder que en un polinomio, por ejemplo $x^2 + 1$ de grado 2, se crea que su respectiva palabra es 101, que está en K^3 pero que originalmente la palabra a la cual representa este polinomio, esté en el espacio vectorial K^5, y sea la palabra 00101.

Esto no sucede al revés. Al conocer las palabras y su longitud, automáticamente sabremos el espacio vectorial que se está tratando. Por ejemplo, la palabra $(a_0, a_1, a_2, a_3, a_4)$ de longitud 5 (en K^5), está representado por el polinomio $a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$ de grado a lo sumo 4.

Veamos un ejemplo más concreto:

Ejercicio 1.3.9. *Represente cada palabra del código C por polinomios.*

a) $C = \{(0,0,0), (0,0,1), (0,1,0), (0,1,1)\}$.

b) $C = \{(0,0,0,0), (1,0,0,1), (0,1,1,0), (1,1,1,1)\}$.

Solución
a)

codigo palabra c	*polinomio c*(x)
(0,0,0)	0
(0,0,1)	x^2
(0,1,0)	x
(0,1,1)	$x + x^2$

d)

codigo palabra c	*polinomio c*(x)
(0,0,0,0)	0
(1,0,0,1)	$1 + x^3$
(0,1,1,0)	$x + x^2$
(1,1,1,1)	$1 + x + x^2 + x^3$

Con el ejemplo anterior, notamos que un código C de longitud n, puede ser representado como conjunto de polinomios de grado a lo sumo $n - 1$ en el campo K.

Ejercicio 1.3.10. *Escriba el código de Hamming de longitud 7 generado por la matriz G y luego represente este código por polinomios.*

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

Solución

Por definición, para conocer el código de Hamming de la matriz G, basta multiplicar (por la derecha) a G por todas las palabras de longitud 4, ya que G es de orden 4×7. Por ejemplo si multiplicamos la palabra $(0, 0, 1, 0)$ con G tendremos:

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

Por lo tanto, la palabra $(0, 0, 1, 0, 1, 0, 1)$ es la palabra perteneciente al código de Hamming de longitud 7 y cuyo polinomio asociado es $x^6 + x^4 + x^2$.

Así, se presentan los resultados a continuación.

En la última columna de la tabla siguiente, se coloca el polinomio asociado a cada palabra del código de Hamming (hay que recordar que las palabras del código son de longitud 7, entonces los polinomios que se tendrán por cada palabra de código serán de grado a lo sumo 6).

N	Palabras de K^4	uG donde $u \in K^4$	Polinomios asociados a las palabras de código
1	(0,0,0,0)	(0,0,0,0,0,0,0)	0
2	(0,0,0,1)	(0,0,0,1,0,1,1)	$x^6 + x^5 + x^3$
3	(0,0,1,0)	(0,0,1,0,1,0,1)	$x^6 + x^4 + x^2$
4	(0,0,1,1)	(0,0,1,1,1,1,0)	$x^5 + x^4 + x^3 + x^2$
5	(0,1,0,0)	(0,1,0,0,1,1,0)	$x^5 + x^4 + x$
6	(0,1,0,1)	(0,1,0,1,1,0,1)	$x^6 + x^4 + x^3 + x$
7	(0,1,1,0)	(0,1,1,0,0,1,1)	$x^6 + x^5 + x^2 + x$
8	(0,1,1,1)	(0,1,1,1,0,0,0)	$x^3 + x^2 + x$
9	(1,0,0,0)	(1,0,0,0,1,1,1)	$x^6 + x^5 + x^4 + 1$
10	(1,0,0,1)	(1,0,0,1,1,0,0)	$x^4 + x^3 + 1$
11	(1,0,1,0)	(1,0,1,0,0,1,0)	$x^5 + x^2 + 1$
12	(1,0,1,1)	(1,0,1,1,0,0,1)	$x^6 + x^3 + x^2 + 1$
13	(1,1,0,0)	(1,1,0,0,0,0,1)	$x^6 + x + 1$
14	(1,1,0,1)	(1,1,0,1,0,1,0)	$x^5 + x^3 + x + 1$
15	(1,1,1,0)	(1,1,1,0,1,0,0)	$x^4 + x^2 + x + 1$
16	(1,1,1,1)	(1,1,1,1,1,1,1)	$x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$

Finalmente podemos decir de la tabla anterior que el código de Hamming de longitud 7 es:

$$C = \{(0,0,0,0,0,0,0), (0,0,0,1,0,1,1), (0,0,1,0,1,0,1), (0,0,1,1,1,1,0), (0,1,0,0,1,1,0), (0,1,0,1,1,0,1), (0,1,1,0,0,1,1), (0,1,1,1,0,0,0), (1,0,0,0,1,1,1), (1,0,0,1,1,0,0), (1,0,1,0,0,1,0), (1,0,1,1,0,0,1), (1,1,0,0,0,0,1), (1,1,0,1,0,1,0), (1,1,1,0,1,0,0), (1,1,1,1,1,1,1)\}.$$

Y los polinomios asociados a cada una de estas palabras son:

$$\begin{array}{ll} 0 & x^6 + x^5 + x^4 + 1 \\ x^6 + x^5 + x^3 & x^4 + x^3 + 1 \\ x^6 + x^4 + x^2 & x^5 + x^2 + 1 \\ x^5 + x^4 + x^3 + x^2 & x^6 + x^3 + x^2 + 1 \\ x^5 + x^4 + x & x^6 + x + 1 \\ x^6 + x^4 + x^3 + x & x^5 + x^3 + x + 1 \\ x^6 + x^5 + x^2 + x & x^4 + x^2 + x + 1 \\ x^3 + x^2 + x & x^6 + x^5 + x^4 + x^3 + x^2 + x + 1. \end{array}$$

A continuación, extenderemos la idea de módulo que se estudia en cualquier curso de álgebra abstracta. En principio se dice que $a = b$ mód n, si el resto que deja b al ser dividido por n es a.

Ahora bien, se dice que dos elementos c y d son equivalentes o congruentes módulo n, si ambos c y d, dejan el mismo resto al dividir por n, es decir: $c \equiv d$ mód n. Entonces, con esta idea en mente, se tiene la siguiente definición:

Definición 1.3.11. *Decimos que $f(x)$ módulo $h(x)$ es $r(x)$ si $r(x)$, es el resto cuando $f(x)$ está dividido por $h(x)$. Se denotar como $r(x) = f(x)$ mód $h(x)$.*

Definición 1.3.12. *Dos funciones $f(x)$ y $p(x)$ son módulo equivalentes $h(x)$ si y sólo si tienen el mismo resto cuando se divide por $h(x)$; eso es si:*

$$f(x) \text{ mód } h(x) = r(x) = p(x) \text{ mód } h(x)$$

Denotamos esto por $f(x) \equiv p(x)(\text{mód} h(x))$.

Ejemplo 1.3.13. *Sea $h(x) = 1 + x^5$ y $f(x) = 1 + x^4 + x^9 + x^{11}$. Entonces dividiendo $f(x)$ por $h(x)$ da un resto de $r(x) = 1 + x$. Decimos que $r(x) = f(x)$ mód $h(x)$.*

De forma similar, si $p(x) = 1 + x^6$, entonces $1 + x = 1 + x^6$ mód $(1 + x^5)$ y por lo tanto decimos $p(x) \equiv f(x)$ mód $h(x)$.

Ejemplo 1.3.14. *Sea $h(x) = 1 + x^2 + x^5$. Calculando $f(x)$* mód *$h(x)$, con $f(x) = 1 + x^2 + x^6 + x^9 + x^{11}$, encontramos que el resto es $r(x) = x + x^4$ y entonces $x + x^4 = f(x)$* mód *$h(x)$.*

Tenga en cuenta que si $p(x) = x^2 + x^8$, entonces $p(x)$ mód *$h(x) = 1 + x^3$ y $p(x)$ y $f(x)$ no son equivalentes* mód *$h(x)$.*

La adición y multiplicación de polinomios respeta la equivalencia de los polinomios definidos anteriormente. Es decir:

Lema 1.3.15. *Si $f(x) \equiv g(x)$* mód *$h(x)$, entonces, para algún $p(x) \in K[x]$, se cumple:*

$$f(x) + p(x) \equiv g(x) + p(x)(\text{mód} h(x))$$

y

$$f(x)p(x) \equiv g(x)p(x)(\text{mód} h(x)).$$

Demostración.
Suponga que $r(x) = f(x)$ mód $h(x)$, $r(x) = g(x)$ mód $h(x)$ y $s(x) = p(x)$ mód $h(x)$ entonces tenemos:

$$\begin{aligned} f(x) + p(x) &= q_1(x)h(x) + r(x) + q_2(x)h(x) + s(x) \\ &= (q_1(x) + q_2(x))h(x) + r(x) + s(x). \end{aligned}$$

Lo que implica que $r(x) + s(x) = (f(x) + p(x))$ mód $h(x)$.

Además, también se cumple que $r(x) + s(x) = (g(x) + p(x))$ mód $h(x)$.

$$\begin{aligned} g(x)+p(x) &= q_3(x)h(x)+r(x)+q_2(x)h(x)+s(x) \\ &= (q_3(x)+q_2(x))h(x)+r(x)+s(x). \end{aligned}$$

Se observa que, $deg(r(x)+s(x)) \leq \text{máx}\{deg(r(x)), deg(s(x))\}$, por definición.

Como $deg(r(x)) < deg(h(x))$ y $deg(s(x)) < deg(h(x))$ se cumple: $deg(r(x)+s(x)) < deg(h(x))$.

Por lo tanto, $f(x)+p(x) \equiv g(x)+p(x)(\text{mód}\, h(x))$.

Por otro lado, se tiene:

$$\begin{aligned} f(x)p(x) &= (q_1(x)h(x)+r(x))(q_2(x)h(x)+s(x)) \\ &= (q_1(x)h(x)+r(x))(q_2(x)h(x))+(q_1(x)h(x)+r(x))(s(x)) \\ &= (q_1(x)h(x))(q_2(x)h(x))+(r(x))(q_2(x)h(x))+(q_1(x)h(x))(s(x))+(r(x))(s(x)) \\ &= (q_1(x)q_2(x)h(x)+r(x)q_2(x)+q_1(x)s(x))h(x)+r(x)s(x). \end{aligned}$$

Lo anterior, implica que $r(x)s(x) = f(x)p(x)$ mód $h(x)$.

$$\begin{aligned} g(x)p(x) &= (q_3(x)h(x)+r(x))(q_2(x)h(x)+s(x)) \\ &= (q_3(x)h(x)+r(x))(q_2(x)h(x))+(q_3(x)h(x)+r(x))(s(x)) \\ &= (q_3(x)h(x))(q_2(x)h(x))+(r(x))(q_2(x)h(x))+(q_3(x)h(x))(s(x))+(r(x))(s(x)) \\ &= (q_3(x)q_2(x)h(x)+r(x)q_2(x)+q_3(x)s(x))h(x)+r(x)s(x). \end{aligned}$$

Note que también se cumple que $r(x)s(x) = g(x)p(x)$ mód $h(x)$.

Por lo tanto, $f(x)p(x) \equiv g(x)p(x)(\text{mód}\, h(x))$.

□

Ejemplo 1.3.16. *Sea* $h(x) = 1+x^5$, $f(x) = 1+x+x^7$, $g(x) = 1+x+x^2$ *y* $p(x) = 1+x^6$; *entonces* $f(x)+p(x) \equiv g(x)+p(x)(\text{mód}\, h(x))$ *y* $f(x)p(x) \equiv g(x)p(x)(\text{mód}\, h(x))$.

Se tiene

$$f(x)+p(x) = x+x^6+x^7$$

y

$$g(x)+p(x) = x+x^2+x^6,$$

pero

$$(x+x^6+x^7) \text{ mód } h(x) = x^2 = (x+x^2+x^6) \text{ mód } h(x).$$

Similarmente $(1+x+x^7)(1+x^6)$ mód $h(x) = 1+x^3 = (1+x+x^2)(1+x^6)$ mód $h(x)$. *De aquí que* $f(x)p(x) \equiv g(x)p(x)(\text{mód}\, h(x))$.

Note que $1+x = (1+x^6)$ mód $h(x)$. *Así tenemos:*

$$\begin{aligned} (1+x+x^7)(1+x^6) &\equiv (1+x+x^2)(1+x^6) \\ &\equiv (1+x+x^2)(1+x) \\ &\equiv 1+x^3 \text{ mód } h(x). \end{aligned}$$

Esto nos dice que si el polinomio $p(x)$ es de grado mayor al de $h(x)$, nosotros podemos trabajar con el resto que deja $p(x)$ al ser dividido por $h(x)$.

Ejemplo 1.3.17. *Sea $h(x) = 1 + x^7$. Calcular $f(x)$* mód *$h(x)$ y $p(x)$* mód *$h(x)$ y compruebe si $f(x) \equiv p(x)$ (*mód *$h(x)$).*

a) $f(x) = 1 + x^6 + x^8$, $p(x) = x + x^3 + x^7$.

b) $f(x) = x + x^5 + x^9$, $p(x) = x + x^5 + x^6 + x^{13}$.

c) $f(x) = 1 + x$, $p(x) = x + x^7$.

Solución
a) $r_1(x) = f(x)$ mód $h(x)$

$$\begin{array}{rrrl|l} x^8 & & +x^3 & +1 & x^7+1 \\ \hline x^8 & & & +x & x \\ \hline & & x^3 & +x+1. & \end{array}$$

Entonces $r_1(x) = x^3 + x + 1$.

$r_2(x) = p(x)$ mód $h(x)$

$$\begin{array}{rrrl|l} x^7 & & +x^3 & +x & x^7+1 \\ \hline x^7 & & & +1 & 1 \\ \hline & & x^3 & +x+1. & \end{array}$$

Entonces $r_2(x) = x^3 + x + 1$.
Como se cumple que $r_1(x) = r_2(x)$, entonces $f(x) \equiv p(x)$ (mód $h(x)$).

b) $r_1(x) = f(x)$ mód $h(x)$

$$\begin{array}{rrrrl|l} x^9 & & +x^5 & & +x & x^7+1 \\ \hline x^9 & & & +x^2 & & x^2 \\ \hline & & x^5 & +x^2+x. & & \end{array}$$

Entonces $r_1(x) = x^5 + x^2 + x$.

$r_2(x) = p(x)$ mód $h(x)$

$$\begin{array}{rrrrl|l} x^{13} & & +x^6+x^5 & & +x & x^7+1 \\ \hline x^{13} & +x^7 & & & & x^6+1 \\ \hline & x^7+x^6+x^5 & & +x & & \\ & x^7 & & & +1 & \\ \hline & & x^6+x^5 & +x+1. & & \end{array}$$

Entonces $r_2(x) = x^6 + x^5 + x + 1$.

Como se cumple que $r_1(x) \neq r_2(x)$, entonces $f(x) \not\equiv p(x) \pmod{h(x)}$.

c) $r_1(x) = f(x)$ mód $h(x)$

Dado que el dividendo ($f(x) = 1 + x$) es de menor grado que el divisor ($h(x) = 1 + x^7$). Entonces $r_1(x) = f(x) = 1 + x$.

$r_2(x) = p(x)$ mód $h(x)$

$$\begin{array}{rrl} x^7 & +x & \big| \; x^7 + 1 \\ \cline{3-3} x^7 & +1 & \; 1 \\ \cline{1-2} & +x+1. & \end{array}$$

Entonces $r_2(x) = x + 1$. Por tanto $r_1 = r_2$ y claramente $f(x) \equiv p(x) \pmod{h(x)}$.

Hasta este punto, se han desarrollado los resultados más importantes de códigos lineales y polinomios sobre K, estos serán de ayuda en el desarrollo de la parte final de la investigación, la cual comprende el estudio de los códigos cíclicos en K^n.

Capítulo 2

Códigos cíclicos

2.1. Polinomios y palabras de un código cíclico

En esta sección trataremos los códigos cíclicos y cómo estos se definen a partir de un código lineal. Además, se muestra la relación que existe entre una palabra y un polinomio. De modo que una vez definidos los códigos cíclicos, podemos hablar de sus cambios cíclicos, de cómo estos corresponden a un polinomio en $K[x]$ y finalmente se define el polinomio generador de un código cíclico y las propiedades que cumple.

Definición 2.1.1. *Una permutación o cambio cíclico es una función de K^n a K^n, que envía $v = (a_0, \ldots, a_{n-1}) \in K^n$ a*

$$\pi(v) = \pi(a_0, \ldots, a_{n-1}) = (a_{n-1}, a_0, \ldots, a_{n-2}).$$

Llamaremos a $\pi(v)$ el cambio cíclico de v.

En general, el cambio cíclico de $v \in K^n$ de longitud n, es la palabra que resulta de trasladar el último dígito de v al inicio. Todos los demás dígitos se mueven una posición a la derecha.

Ejemplo 2.1.2.

v	$(1,0,1,1,0)$	$(1,1,1,0,0,0)$	$(0,0,0,0)$	$(1,0,1,1)$
$\pi(v)$	$(0,1,0,1,1)$	$(0,1,1,1,0,0)$	$(0,0,0,0)$	$(1,1,0,1)$

Definición 2.1.3. *Un código C es **cíclico** si es lineal y el cambio cíclico de cada palabra en C, es también una palabra en C.*

Ejemplo 2.1.4. *El código C = {(0,0,0),(1,1,0),(1,0,1),(0,1,1)} es un código cíclico.*

Por la Definición 1.2.1, del capítulo 1, se sabe que dicho código es lineal. A continuación, se analizarán los cambios cíclicos $\pi(v)$ para todo v en C:

$$\begin{aligned} \pi(0,0,0) &= (0,0,0) \\ \pi(1,1,0) &= (0,1,1) \\ \pi(1,0,1) &= (1,1,0) \\ \pi(0,1,1) &= (1,0,1). \end{aligned}$$

*De lo anterior, se concluye que para cada v en C, $\pi(v)$ también está en C, entonces **C es cíclico.***

Notar que π bajo la *Definición 2.1.1* es una transformación lineal. Con esto, podemos demostrar el siguiente resultado:

Lema 2.1.5. *Si π es un cambio cíclico, entonces π es una transformación lineal. Es decir,*

$$\pi(v+w) = \pi(v) + \pi(w) \quad y \quad \pi(av) = a\pi(v), \;\; a \in K = \{0,1\}.$$

Demostración. Sea $v = (v_0, v_1, ..., v_{n-1})$ y $w = (w_0, w_1, ..., w_{n-1})$, entonces:

$$v + w = (v_0 + w_0, v_1 + w_1, ..., v_{n-1} + w_{n-1})$$

$$\begin{aligned} \pi(v+w) &= (v_{n-1} + w_{n-1}, v_0 + w_0, ..., v_{n-2} + w_{n-2}) \\ &= (v_{n-1}v_0...v_{n-2}) + (w_{n-1}w_0...w_{n-2}) \\ &= \pi(v) + \pi(w) \end{aligned}$$

Por otro lado, dado $a \in K$:

$$\begin{aligned} av &= a(v_0, v_1, ..., v_{n-1}) \\ &= (av_0, av_1, ..., av_{n-1}) \end{aligned}$$

$$\begin{aligned} \pi(av) &= (av_{n-1}, av_0, ..., av_{n-2}) \\ &= a(v_{n-1}, v_0, ..., v_{n-2}) \\ &= a\pi(v) \end{aligned}$$

□

Dado que π es una transformación lineal, note que si tenemos una base $\beta = \{v_1, v_2, ..., v_n\}$ para un código lineal C y queremos saber si éste es cíclico, basta probar que los cambios cíclicos de cada uno de los elementos de la base pertenezcan al código lineal C.

Es decir, si C es un código lineal cíclico y $w \in C$, entonces como β es una base para C, se tiene:

$$w = \alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_n v_n.$$

Y el cambio cíclico de w es:

$$\begin{aligned} \pi(w) &= \pi(\alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_n v_n). \\ &= \pi(\alpha_1 v_1) + \pi(\alpha_2 v_2) + ... + \pi(\alpha_n v_n). \\ &= \alpha_1 \pi(v_1) + \alpha_2 \pi(v_2) + ... + \alpha_n \pi(v_n). \end{aligned}$$

De lo anterior, como cada $v_i \in C$, $\forall i = \{1, ..., n\}$ y C es cíclico, entonces $\pi(v_i) \in C$ y por ser C lineal, la suma en la derecha está en C, lo que significa que $\pi(w)$ también debe estar en C.

Eso quiere decir que es suficiente probar que $\pi(v) \in C$ para cada v en una base de C.

Ejemplo 2.1.6. *Sea* $C = \{(0,0,0),(1,1,0),(1,0,1),(0,1,1)\}$ *un código lineal y* $B = \{(1,1,0),(1,0,1)\}$ *una base para el código C. Además, como* $\pi(1,1,0) = (0,1,1) \in C$ *y* $\pi(1,0,1) = (1,1,0) \in C$ *entonces C es un código lineal cíclico.*

Nota: Se utilizará la notación $\pi^2(v) = \pi(\pi(v))$, $\pi^3(v) = \pi(\pi(\pi(v)))$, etc. En general, para referirse al $k-ésimo$ cambio cíclico de alguna palabra v, se escribirá $\pi^k(v)$.

Para construir un código lineal cíclico, podemos elegir una palabra v de K^n y encontrar todos sus desplazamientos cíclicos para formar un conjunto S que consistirá de v y de todos sus desplazamientos, es decir, $S = \{v, \pi(v), ..., \pi^{n-1}(v)\}$, para luego definir a C como el generado lineal de S, lo que significa que $C =< S >$.

Además, como $S \subset C$ y S contiene una base para C, los cambios cíclicos de dicha base estarán siempre en S y por tanto en C, entonces C debe ser cíclico.

Ejemplo 2.1.7. *Si* $v = (1,0,0)$ *y* $n = 3$. *Entonces* $S = \{v, \pi(v), \pi^2(v)\}$=$\{(1,0,0),(0,1,0),(0,0,1)\}$ *genera a* K^3. *Es decir,* $K^3 =< S >$.

Si $w \in K^3$, $\quad w = a_0v + a_1\pi(v) + a_2\pi^2(v)$.

Entonces, $\pi(w) = a_0\pi(v) + a_1\pi^2(v) + a_2\pi^3(v)$. *Pero* $\pi^3(v) = v$.

Así, $\pi(w) = a_2v + a_0\pi(v) + a_1\pi^2(v)$.

Esto significa que el cambio cíclico de w, también puede ser escrito como combinación lineal del conjunto generador de K^3 *para todo* $w \in K^3$.

Ejemplo 2.1.8. *Si* $v = (0,1,0,1)$ *y* $n = 4$. *Entonces,* $\pi(v) = (1,0,1,0)$ *y* $\pi^2(v) = (0,1,0,1) = v$.

Entonces, $S = \{(0,1,0,1),(1,0,1,0)\}$ $\quad y \quad C =< S >$.

Así,

$$C = \{(0,0,0,0),(0,1,0,1),(1,0,1,0),(1,1,1,1)\}.$$

En general, dado que una palabra v y sus desplazamientos cíclicos forman un conjunto $S = \{v, \pi(v), ..., \pi^{n-1}(v)\}$ del cual se define C como el generado de S, diremos que v es un generador del código C.

Además por conocimientos previos, se sabe que C siendo el conjunto generado por un conjunto S, es el conjunto más pequeño que contiene a S.

Ejercicio 2.1.9. *Encuentre una base para el código cíclico lineal más pequeño de longitud 7 que contiene a* $v = (1,0,0,1,0,0,0)$.

Solución

a) Formamos el conjunto S que consistirá en v y en todos sus desplazamientos cíclicos.

$$\begin{aligned}
\pi(1,0,0,1,0,0,0) &= (0,1,0,0,1,0,0)\\
\pi^2(1,0,0,1,0,0,0) &= (0,0,1,0,0,1,0)\\
\pi^3(1,0,0,1,0,0,0) &= (0,0,0,1,0,0,1)\\
\pi^4(1,0,0,1,0,0,0) &= (1,0,0,0,1,0,0)\\
\pi^5(1,0,0,1,0,0,0) &= (0,1,0,0,0,1,0)\\
\pi^6(1,0,0,1,0,0,0) &= (0,0,1,0,0,0,1)
\end{aligned}$$

Así, $S = \{(1,0,0,1,0,0,0), (0,1,0,0,1,0,0), (0,0,1,0,0,1,0), (0,0,0,1,0,0,1), (1,0,0,0,1,0,0), (0,1,0,0,0,1,0), (0,0,1,0,0,0,1)\}$ y como ya sabemos que $C =< S >$, entonces falta ver si son l.i.

$$\begin{aligned}
(0,0,0,0,0,0,0) &= a_0(1,0,0,1,0,0,0) + a_1(0,1,0,0,1,0,0) + a_2(0,0,1,0,0,1,0) + a_3(0,0,0,1,0,0,1)\\
&+ a_4(1,0,0,0,1,0,0) + a_5(0,1,0,0,0,1,0) + a_6(0,0,1,0,0,0,1).\\
&= (a_0,0,0,a_0,0,0,0) + (0,a_1,0,0,a_1,0,0) + (0,0,a_2,0,0,a_2,0)\\
&+ (0,0,0,a_3,0,0,a_3) + (a_4,0,0,0,a_4,0,0) + (0,a_5,0,0,0,a_5,0) + (0,0,a_6,0,0,0,a_6).\\
&= (a_0+a_4, a_1+a_5, a_2+a_6, a_0+a_3, a_1+a_4, a_2+a_5, a_3+a_6)
\end{aligned}$$

Y por tanto:

$$\begin{aligned}
a_0 + a_4 &= 0\\
a_1 + a_5 &= 0\\
a_2 + a_6 &= 0\\
a_0 + a_3 &= 0\\
a_1 + a_4 &= 0\\
a_2 + a_5 &= 0\\
a_3 + a_6 &= 0
\end{aligned}$$

De lo anterior se verifica que $a_0 = a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = 0$ ó bien $a_0 = a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = 1$. Por tanto, como los escalares pueden ser todos nulos o iguales a 1, el conjunto es linealmente dependiente, es decir, al menos uno de los vectores se puede escribir como combinación lineal de los demás.

Entonces, al quitar uno de esos vectores, verificamos si ahora son linealmente independientes.

Quitamos $(0,0,1,0,0,0,1)$:

$$\begin{aligned}(0,0,0,0,0,0,0) &= a_0(1,0,0,1,0,0,0)+a_1(0,1,0,0,1,0,0)+a_2(0,0,1,0,0,1,0)+a_3(0,0,0,1,0,0,1)\\ &+ a_4(1,0,0,0,1,0,0)+a_5(0,1,0,0,0,1,0).\\ &= (a_0,0,0,a_0,0,0,0)+(0,a_1,0,0,a_1,0,0)+(0,0,a_2,0,0,a_2,0)\\ &+ (0,0,0,a_3,0,0,a_3)+(a_4,0,0,0,a_4,0,0)+(0,a_5,0,0,0,a_5,0).\\ &= (a_0+a_4,a_1+a_5,a_2,a_0+a_3,a_1+a_4,a_2+a_5,a_3)\end{aligned}$$

Y por tanto:

$$\begin{aligned}a_0+a_4 &= 0\\ a_1+a_5 &= 0\\ a_2 &= 0\\ a_0+a_3 &= 0\\ a_1+a_4 &= 0\\ a_2+a_5 &= 0\\ a_3 &= 0\end{aligned}$$

Luego, la única solución es $a_0 = a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = 0$, y el nuevo conjunto $S' = \{(1,0,0,1,0,0,0),(0,1,0,0,1,0,0),(0,0,1,0,0,1,0),(0,0,0,1,0,0,1),(1,0,0,0,1,0,0),$ $(0,1,0,0,0,1,0)\}$ es generador y linealmente independiente. Por tanto S' es una base para C.

Ejercicio 2.1.10. *Encuentre todas las palabras v de longitud n tal que* $\pi(v) = v$.

Solución

Sea $v = (a_0, a_1, ..., a_{n-1})$, entonces $\pi(v) = (a_{n-1}, a_0, ..., a_{n-2})$. Luego, para que $v = \pi(v)$:

$$(a_0, a_1, ..., a_{n-1}) = (a_{n-1}, a_0, ..., a_{n-2})$$

Así,

$$\begin{aligned}a_0 &= a_{n-1}\\ a_1 &= a_0\\ a_2 &= a_1\\ &\vdots\\ a_{n-1} &= a_{n-2}\end{aligned}$$

Es decir, $a_0 = a_1 = ... = a_{n-1}$, pero como $K = \{0,1\}$ entonces los $a_i = 0$ ó $a_i = 1$, $\forall i = 1, ..., n$.

Así, las únicas palabras que cumplen son:

$$(0,0,...,0) \quad \text{y} \quad (1,1,...,1).$$

Se sabe que a la palabra $v = (a_0, a_1, ..., a_{n-1})$ en C, con $l(v) = n$ le corresponde el polinomio $v(x) = a_0 + a_1x + ... + a_{n-1}x^{n-1}$.

Así,

$$\begin{aligned} xv(x) &= a_0x + a_1x^2 + ... + a_{n-1}x^n. \\ &= a_0x + a_1x^2 + ... + a_{n-1}x^n + (a_{n-1} + a_{n-1}). \\ &= a_{n-1} + a_0x + a_1x^2 + ... + a_{n-1}x^n + a_{n-1}. \\ &= (a_{n-1} + a_0x + a_1x^2 + ... + a_{n-2}x^{n-1}) + a_{n-1}(x^n + 1). \end{aligned}$$

Y como $\pi(v) = (a_{n-1}, a_0, a_1, ..., a_{n-2})$ y $\pi(v)(x) = a_{n-1} + a_0x + ... + a_{n-2}x^{n-1}$. Entonces:

$$xv(x) = a_{n-1}(x^n + 1) + \pi(v)(x).$$

De aquí se sabe que:

$$\pi(v)(x) = xv(x) \text{ mód } (x^n + 1).$$

Lo que significa que a la palabra $\pi(v)$ le corresponde el polinomio $xv(x)$ mód $(1 + x^n)$.

De esta manera, a $\pi^i(v)$ le corresponde el polinomio $x^iv(x)$ mód $(1 + x^n)$ para $i = 0, ..., n - 1$ y $l(v) = n$.

Ejemplo 2.1.11. *Sea $v = (1, 1, 0, 1, 0, 0, 0)$ y $n = 7$. Así, $v(x) = 1 + x + x^3$ y calculamos $x^iv(x)$ para $1 \leq i \leq 6$ en la siguiente tabla.*

Palabra	*Polinomio* mód $(1 + x^7)$
(0,1,1,0,1,0,0)	$xv(x) = x + x^2 + x^4$
(0,0,1,1,0,1,0)	$x^2v(x) = x^2 + x^3 + x^5$
(0,0,0,1,1,0,1)	$x^3v(x) = x^3 + x^4 + x^6$
(1,0,0,0,1,1,0)	$x^4v(x) = x^4 + x^5 + x^7 \equiv 1 + x^4 + x^5$ mód $(1 + x^7)$
(0,1,0,0,0,1,1)	$x^5v(x) = x^5 + x^6 + x^8 \equiv x + x^5 + x^6$ mód $(1 + x^7)$
(1,0,1,0,0,0,1)	$x^6v(x) = x^6 + x^7 + x^9 \equiv 1 + x^2 + x^6$ mód $(1 + x^7)$

Lema 2.1.12. *Dado $C \subseteq K^n$ un código cíclico y $v \in C$. Entonces para cualquier polinomio $a(x)$, $c(x) = a(x)v(x)$ mód $(1 + x^n)$ es una palabra de C.*

Demostración. Se sabe que si $v \in C$, su polinomio correspondiente es $v(x)$.
Sea $a(x) = a_0 + a_1x + ... + a_{n-1}x^{n-1}$ un polinomio de grado a lo sumo $n - 1$ en $K[x]$.

Entonces,

$$\begin{aligned} a(x)v(x) \text{ mód } (1 + x^n) &= (a_0 + a_1x + ... + a_{n-1}x^{n-1})v(x) \text{ mód } (1 + x^n) \\ &= (a_0v(x) + a_1xv(x) + ... + a_{n-1}x^{n-1}v(x)) \text{ mód } (1 + x^n) \\ &= a_0v(x) \text{ mód } (1 + x^n) + a_1xv(x) \text{ mód } (1 + x^n) + ... \\ &+ a_{n-1}x^{n-1}v(x) \text{ mód } (1 + x^n). \end{aligned}$$

Dado que a $\pi^i(v)$ le corresponde el polinomio $x^i v(x)$ mód $(1+x^n)$ y al multiplicar dichos polinomios por a_i con $a_i \in K$, este nos devuelve ya sea el polinomio nulo o el mismo polinomio, entonces toda la suma anterior también corresponde a una palabra de nuestro código cíclico C.

Por tanto, $c(x) = a(x)v(x)$ mód $(1+x^n) \in C$. □

Proposición 2.1.13. *Sea C un código lineal, entonces existe un único polinomio de grado mínimo distinto de cero.*

Demostración. El espacio vectorial K^n es finito por definición, entonces dado $C \subseteq K^n$, C es finito y por tanto, existe $g \in C$ ($g \neq \mathbf{0}$) tal que $g(x)$ tiene grado mínimo.

Supongamos que existen $h(x)$ y $g(x)$ polinomios no nulos de grado mínimo $m \neq 0$ en C. Entonces como C es lineal, $h(x) + g(x) = i(x)$ y dado que $h(x)$ y $g(x)$ poseen el término x^m, entonces $x^m + x^m = 0$ y por tanto, $deg(i(x)) < m$.

Pero esto significa que $deg(i(x)) = 0$ debido a que $h(x)$ y $g(x)$ son los únicos polinomios de grado mínimo $m \neq 0$ y así $i(x) = \mathbf{0}$ (polinomio cero).

De lo anterior, $h(x) + g(x) = \mathbf{0}$ y ésto se cumple sí y sólo sí $h(x) = g(x)$. □

Notación: Denotaremos al polinomio de menor grado en C como $g(x)$.

Observación: Note que por la *Proposición 2.1.13*, g es único.

Proposición 2.1.14. *Sea C un código lineal y $g \in C$, entonces $g(x)$ genera al código C.*
A este polinomio se le llamará ***polinomio generador de C****.*

Demostración. Para ver que $g(x)$ genera al código C, hay que probar que para toda palabra $c \in C$, existe $a(x)$ tal que $c(x) = a(x)g(x)$.

Se sabe que $deg(g(x)) \leq deg(c(x))$, por ser $g(x)$ el polinomio de grado mínimo. Así, se tiene por el algoritmo de la división:

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

para algún $q(x)$ y $r(x)$, con $deg(r(x)) < deg(g(x))$.

Además, $deg(q(x)g(x)) \leq deg(c(x))$, entonces $deg(q(x)g(x)) \leq n-1$.

Por otro lado, dado que $g(x) \in C$, entonces por el *Lema 2.1.12*, $q(x)g(x)$ mód $(1+x^n) \in C$, pero $deg(q(x)g(x)) \leq n-1 < n$, entonces $q(x)g(x)$ mód $(1+x^n) = q(x)g(x)$ y así, $q(x)g(x) \in C$.

La igualdad $c(x) = q(x)g(x) + r(x)$, la podemos reescribir como $r(x) = q(x)g(x) + c(x)$ y como se sabe que $q(x)g(x)$ y $c(x)$ están en C y además que C es lineal, $q(x)g(x) + c(x) \in C$ y entonces $r(x) \in C$.

Sin embargo, no puede suceder que exista otro polinomio $r(x) \in C$ de grado menor que el de $g(x)$. Entonces debe ocurrir que $r(x) = \mathbf{0}$, por lo tanto

$$c(x) = q(x)g(x).$$

□

Observación: Por la proposición anterior, en el *Lema 2.1.12*, en particular si $v = g(x)$ entonces se cumple que

$$c(x) = a(x)g(x) \in C, \quad \text{donde} \quad deg(a(x)g(x)) < n.$$

Teorema 2.1.15. *Sea C un código cíclico de longitud n y sea $g(x)$ el polinomio generador. Si $n - k = deg(g(x))$, entonces*

1. *El polinomio $c(x) \in C$ si y sólo si $c(x) = a(x)g(x)$ para algún polinomio $a(x)$ con $deg(a(x)) < k$ (es decir, $g(x)$ es un divisor de cada palabra de código $c(x)$.)*
2. *C tiene dimensión k.*
3. *Las palabras de código correspondientes a $g(x)$, $xg(x), \ldots, x^{k-1}g(x)$ forman una base para C.*

Demostración.
1. ($\Rightarrow$) Por la observación anterior, dado que $g(x)$ es generador, existe $a(x)$ tal que $c(x) = a(x)g(x)$.

Además,

$$\begin{aligned} deg(c(x)) &\leq n-1 \\ deg(a(x)g(x)) &\leq n-1 \\ deg(a(x)) + deg(g(x)) &\leq n-1 \\ deg(a(x)) + n - k &\leq n-1 \\ deg(a(x)) &\leq n-1-n+k \\ deg(a(x)) &\leq k-1. \end{aligned}$$

Por lo tanto, $deg(a(x)) < k$.

($\Leftarrow$) Dado que $g(x) \in C$, entonces $c'(x) = a(x)g(x)$ mód $(1 + x^n)$ es una palabra en C, pero por hipótesis $c(x) = a(x)g(x)$, con $deg(a(x)) < k$. Entonces, $deg(c(x)) \leq n - 1$.

Además

$$\begin{aligned} deg(c(x)) &= deg(a(x)) + deg(g(x)) \\ &= deg(a(x)) + n - k \\ &\leq k - 1 + n - k = n - 1, \end{aligned}$$

lo que significa que $deg(a(x)g(x)) \leq n - 1$ y entonces $a(x)g(x)$ mód $(1 + x^n) = a(x)g(x)$.

Por lo tanto, $c'(x) = c(x)$ y $c(x) \in C$.

2. Que C tenga dimensión k implica que deben haber k elementos en la base de C, pero como C es cíclico y $g(x)$ su polinomio generador, entonces se reduce a probar que el conjunto $\{g(x), xg(x), ..., x^{k-1}g(x)\}$ es un conjunto linealmente independiente, ya que por 1, tenemos que para toda palabra c en C la podemos escribir como:

$$\begin{aligned} c(x) &= a(x)g(x) \\ &= (a_0 + a_1x + a_2x^2 + \ldots + a_{k-1}x^{k-1})g(x) \\ &= a_0g(x) + a_1xg(x) + a_2x^2g(x) + \ldots + a_{k-1}x^{k-1}g(x). \end{aligned}$$

Por lo tanto, es un conjunto generador de C.

Veamos si existen escalares $a_0, a_1, \ldots, a_{k-1}$ todos nulos, tal que:

$$\begin{aligned} \mathbf{0} &= a_0g(x) + a_1xg(x) + \ldots + a_{k-1}x^{k-1}g(x) \\ \mathbf{0} &= (a_0 + a_1x + \ldots + a_{k-1}x^{k-1})g(x). \end{aligned}$$

Y dado que $K[x]$ es un dominio de integridad, dicho producto dará el polinomio cero, siempre que

$$a_0 + a_1x + \ldots + a_{k-1}x^{k-1} = \mathbf{0} \qquad \text{ó} \qquad g(x) = \mathbf{0}.$$

Sin embargo, $g(x)$ no puede ser el polinomio cero, dado que es el polinomio generador.

Así, $a_0 + a_1x + \ldots + a_{k-1}x^{k-1} = \mathbf{0}$, y esto sucede si $a_0 = a_1 = \ldots = a_{k-1} = 0$.

Entonces, el conjunto $\{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ es linealmente independiente y por lo tanto, es una base para C.

3. Por el numeral 2 se sabe que C tiene dimensión k, entonces solo basta probar que el conjunto $\{g, \pi(g), \ldots, \pi^{k-1}(g)\}$ de k elementos, es linealmente independiente.

Sean $g, \pi(g), \pi^2(g), \ldots, \pi^{k-1}(g)$ las palabras correspondientes a los siguientes polinomios $g(x)$, $xg(x), \ldots, x^{k-1}g(x)$ respectivamente. Además, $\alpha_0, \alpha_1, \ldots, \alpha_{k-1}$ escalares en K, tal que:

$$\alpha_0g + \alpha_1\pi(g) + \ldots + \alpha_{k-1}\pi^{k-1}g = 0.$$

Supongamos por contradicción que existe algún $\alpha_i = 1$, con $i \in \{0, ..., k-1\}$.

Entonces,

$$\begin{aligned} \alpha_i\pi^i(g) &= \alpha_0g + \alpha_1\pi(g) + \ldots + \alpha_{i-1}\pi^{i-1}(g) + \alpha_{i+1}\pi^{i+1}(g) + \ldots + \alpha_{k-1}\pi^{k-1}(g) \\ \pi^i(g) &= \alpha_0g + \alpha_1\pi(g) + \ldots + \alpha_{i-1}\pi^{i-1}(g) + \alpha_{i+1}\pi^{i+1}(g) + \ldots + \alpha_{k-1}\pi^{k-1}(g). \end{aligned}$$

Lo anterior, lo podemos visualizar a través de sus correspondientes polinomios:

$$x^ig(x) = \alpha_0g(x) + \alpha_1xg(x) + \ldots + \alpha_{i-1}x^{i-1}g(x) + \alpha_{i+1}x^{i+1}g(x) + \ldots + \alpha_{k-1}x^{k-1}g(x).$$

Claramente hay una contradicción, ya que en 2 se probó que el conjunto $\{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ es linealmente independiente. Es decir, ninguno de sus elementos puede escribirse en combinación lineal de los demás.

Por lo tanto, la contradicción surge debido a que se supuso la existencia de algún $\alpha_i \neq 0$, entonces debe cumplirse que $\forall i, \alpha_i = 0$ y así, el conjunto $\{g, \pi(g), \pi^2(g), \ldots, \pi^{k-1}(g)\}$ debe ser linealmente independiente.

Ahora, probaremos que para todo $w \in C$, éste se puede escribir como combinación lineal del conjunto $\{g, \pi(g), \pi^2(g), \ldots, \pi^{k-1}(g)\}$.

Se sabe que para todo $w(x) \in C$, existe $a(x)$ tal que:

$$\begin{aligned} w(x) &= a(x)g(x) \\ w(x) &= (a_0 + a_1x + \ldots + a_{k-1}x^{k-1})g(x) \\ w(x) &= a_0g(x) + a_1xg(x) + \ldots + a_{k-1}x^{k-1}g(x). \end{aligned}$$

Pero lo anterior, lo podemos visualizar como a continuación:

$$w = a_0g + a_1\pi(g) + \ldots + a_{k-1}\pi^{k-1}(g).$$

Lo que significa que existen escalares $a_0, a_1, \ldots, a_{k-1} \in K$, tal que $w \in C$ puede escribirse como combinación lineal de $g, \pi(g), \pi^2(g), \ldots, \pi^{k-1}(g)$. Es decir, el conjunto $\{g, \pi(g), \pi^2(g), \ldots, \pi^{k-1}(g)\}$ es generador.

Por lo tanto, $\{g, \pi(g), \pi^2(g), \ldots, \pi^{k-1}(g)\}$ es una base para C. □

Ejemplo 2.1.16. *Sea $n = 7$ y $g(x) = 1 + x + x^3$ el polinomio generador para el código cíclico C. Entonces la dimensión de C debe ser 4 (por el Teorema 2.1.15) y así, una base para C es*

$$\begin{aligned} g(x) &= 1 + x + x^3 \leftrightarrow (1,1,0,1,0,0,0) \\ xg(x) &= x + x^2 + x^4 \leftrightarrow (0,1,1,0,1,0,0) \\ x^2g(x) &= x^2 + x^3 + x^5 \leftrightarrow (0,0,1,1,0,1,0) \\ x^3g(x) &= x^3 + x^4 + x^6 \leftrightarrow (0,0,0,1,1,0,1) \end{aligned}$$

Note que $x^4g(x)$ no es una palabra del código C, dado que el $deg(x^4g(x)) = 7$ y el conjunto de polinomios de C son de grado a lo sumo 6. Sin embargo, $x^4g(x)$ mód $(1 + x^7) = 1 + x^4 + x^5$ sí es una palabra del código C, ya que $1 + x^4 + x^5 = (1 + x + x^2)(1 + x + x^3) = (1 + x + x^2)g(x)$.

Ejemplo 2.1.17. *Sea $C = \{(0,0,0,0), (1,0,1,0), (0,1,0,1), (1,1,1,1)\}$ código cíclico. Los polinomios correspondientes son $\{0, 1 + x^2, x + x^3, 1 + x + x^2 + x^3\}$.*

Sabemos que el polinomio $1 + x^2 \leftrightarrow (1,0,1,0)$ es el polinomio generador para C, ya que es el polinomio de menor grado distinto del polinomio cero en C.

Además, cada polinomio en C es un múltiplo del polinomio generador:

$$\begin{array}{cc} \boldsymbol{0} = \boldsymbol{0}(1+x^2) & x+x^3 = x(1+x^2) \\ 1+x^2 = 1(1+x^2) & 1+x+x^2+x^3 = (1+x)(1+x^2) \end{array}$$

Ejemplo 2.1.18. *El código lineal cíclico más pequeño C de longitud 6 que contiene* $g(x) = 1+x^3 \leftrightarrow (1,0,0,1,0,0)$ *es*

C={(0,0,0,0,0,0),(1,0,0,1,0,0),(0,1,0,0,1,0),(0,0,1,0,0,1), (1,1,0,1,1,0), (1,0,1,1,0,1),(0,1,1,0,1,1),(1,1,1,1,1,1)}

Se sabe que el código cíclico lineal más pequeño que contiene a $g = (1,0,0,1,0,0)$ es aquel formado por g, sus cambios cíclicos, los que resultan de la suma dos a dos de dichos cambios más la palabra cero.
Además, se puede verificar que el polinomio de grado más pequeño que representa una palabra en C es $g(x) = 1+x^3$, y C no contiene ningún otro polinomio de grado 3. Así, $g(x) = 1+x^3$ es el polinomio generador de C.

A continuación, representamos cada palabra de C, como un múltiplo del polinomio generador $g(x)$.

Palabra	*Polinomio* $f(x)$	*Factorización* $h(x)g(x)$ *de* $f(x)$
(0,0,0,0,0,0)	$\mathbf{0}$	$\mathbf{0}(1+x^3)$
(1,0,0,1,0,0)	$1+x^3$	$1(1+x^3)$
(0,1,0,0,1,0)	$x+x^4$	$x(1+x^3)$
(0,0,1,0,0,1)	x^2+x^5	$x^2(1+x^3)$
(1,1,0,1,1,0)	$1+x+x^3+x^4$	$(1+x)(1+x^3)$
(1,0,1,1,0,1)	$1+x^2+x^3+x^5$	$(1+x^2)(1+x^3)$
(0,1,1,0,1,1)	$x+x^2+x^4+x^5$	$(x+x^2)(1+x^3)$
(1,1,1,1,1,1)	$1+x+x^2+x^3+x^4+x^5$	$(1+x+x^2)(1+x^3)$

Teorema 2.1.19. *El polinomio* $g(x)$ *es el polinomio generador para un código lineal cíclico de longitud n sí y sólo si* $g(x)$ *divide a* $1+x^n$ *(es decir* $1+x^n = h(x)g(x)$*).*

Demostración.
($\Rightarrow$) Dado que $deg(g(x)) < deg(1+x^n)$, entonces por el algoritmo de la división se tiene

$$1+x^n = h(x)g(x) + r(x)$$

con $r(x) = \mathbf{0}$ ó $deg(r(x)) < deg(g(x))$.

De manera equivalentemente se tiene

$$r(x) = (1+x^n) + h(x)g(x).$$

Que en términos de módulo, se traduce a

$$r(x) = h(x)g(x) \text{ mód } (1+x^n).$$

Donde $h(x)g(x)$ mód $(1+x^n) \in C$ (por el *Lema 2.1.12*) y de esta manera $r(x) \in C$. Por lo tanto, como no puede haber otro polinomio $r(x)$ distinto de cero en C tal que $deg(r(x)) < deg(g(x))$, se concluye que $r(x) = \mathbf{0}$ y así,

$$1 + x^n = h(x)g(x).$$

($\Leftarrow$) Sabemos que $g(x)$ satisface que $1 + x^n = g(x)h(x)$, con $g(x) \neq \mathbf{0}$. Además consideremos el código cíclico que genera g, donde g es la palabra asociada al polinomio $g(x)$. Así sea $C =< \{g, \pi(g), \pi^2(g), \ldots, \pi^{n-1}(g)\} >$.

Por contradicción asumamos que $g(x)$ no es el polinomio generador de C, pero por la *Proposición 2.1.14*, todo código cíclico tiene un polinomio generador. De esta forma, consideremos $v(x)$ el polinomio generador de C con $v(x) \neq g(x)$ y además $deg(v(x)) < deg(g(x))$, dado que $g(x)$ no es el polinomio generador.

Además, ya que los cambios cíclicos de g generan a C, entonces a v lo podemos escribir como combinación lineal del conjunto generador. Es decir,

$$v = \alpha_0 g + \alpha_1 \pi(g) + ... + \alpha_{n-1}\pi^{n-1}(g),$$

donde su correspondiente polinomio es:

$$\begin{aligned} v(x) &= \alpha_0 g(x) + \alpha_1 x g(x) + ... + \alpha_{n-1}x^{n-1}g(x) \text{ mód } (1+x^n) \\ v(x) &= (\alpha_0 + \alpha_1 x + ... + \alpha_{n-1}x^{n-1})g(x) \text{ mód } (1+x^n) \\ v(x) &= a(x)g(x) \text{ mód } (1+x^n) \\ \Leftrightarrow\ v(x) &= a(x)g(x) + q(x)(1+x^n). \end{aligned}$$

Lo anterior significa que $g(x)$ divide a $v(x)$, ya que $g(x)$ divide a $1 + x^n$ por hipótesis y también divide a $a(x)g(x)$, lo cual implica que $deg(g(x)) < deg(v(x))$. Pero esto no puede ser, dado que $v(x)$ es el polinomio generador y por tanto, el polinomio de menor grado en C.

La contradicción provino de suponer que $g(x)$ no es el polinomio generador de C, por lo tanto, $g(x)$ es el polinomio generador de C. □

Corolario 2.1.20. *El polinomio generador $g(x)$ para el código cíclico más pequeño de longitud n que contiene la palabra v, es el máximo común divisor de $v(x)$ y $1 + x^n$ (es decir, $g(x) = mcd(v(x), 1 + x^n)$).*

Demostración. Sea $v \in C$ y $g(x)$ el polinomio generador, entonces $g(x)$ divide tanto a $v(x)$ como $1 + x^n$, por el *Teorema 2.1.15* y *Teorema 2.1.19* respectivamente.

Pero $g(x)$ está en $C =< \{v(x), xv(x), ..., x^{n-1}v(x)\} >$, por tanto existe un polinomio $a(x)$ tal que

$$g(x) = a(x)v(x) \ mod \ (1 + x^n)$$

o equivalentemente por el algoritmo de división:

$$g(x) = a(x)v(x) + b(x)(1 + x^n).$$

Se sabe que $g(x)$ divide a $a(x)v(x)$ y divide a $b(x)(1+x^n)$, entonces cualquier otro divisor común de $v(x)$ y $1+x^n$ deberá dividir a $g(x)$ y por lo tanto, $g(x)$ es el máximo común divisor de $v(x)$ y $1+x^n$. □

Ejemplo 2.1.21. *Sea $n = 8$ y $v = (1,1,0,1,1,0,0,0)$, es decir, $v(x) = 1 + x + x^3 + x^4$.*

Utilizando el algoritmo de la división, el mcd de $v(x)$ y $1+x^8$ es $1+x^2$. Así $g(x) = 1+x^2$ y el código cíclico lineal más pequeño que contiene a $v(x)$ tiene una dimensión de 6 y $g(x)$ como el polinomio generador.

2.2. El anillo de ideales principales K_n

En ésta sección estudiaremos la conexión que existe entre los códigos cíclicos y algunas estructuras algebraicas básicas. Primero hagamos un breve repaso de algunas definiciones básicas que deben tenerse en cuenta:

Definición 2.2.1. *Un **anillo conmutativo** R **con unidad** es un conjunto con dos operaciones $+$ (suma) y $*$ (producto), con las siguientes propiedades:*

1. *R es cerrado bajo la suma y el producto.*
2. *R es un grupo abeliano sobre $+$.*
3. *La ley asociativa en el producto $a * (b * c) = (a * b) * c$, para todo a, b y c en R.*
4. *La multiplicación es conmutativa $a * b = b * a$, para todo a y b en R.*
5. *R tiene la identidad multiplicativa 1. Esto significa que $a * 1 = 1 * a = a$, para todo a en R.*
6. *La ley distributiva de la multiplicación sobre la suma. En otras palabras $a * (b + c) = a * b + a * c$, para todo a, b y c en R.*

***Observación:** Si todos los elementos de R distintos del cero tienen inverso multiplicativo y no tiene divisores de cero, entonces R es llamado **campo**.*

Definición 2.2.2. *Un **ideal** I en un anillo conmutativo R es un conjunto de elementos en R que satisface las siguientes condiciones:*

1. *Si $a \in I$, entonces $a * b \in I$, para todo b en R.*
2. *Si $a, b \in I$, entonces $a + b$ y $a - b$ están en I.*

***Observación:** Es importante notar que la segunda condición nos dice que I es subgrupo aditivo del grupo R.*

Definición 2.2.3. *Un ideal I de un anillo conmutativo R es llamado **ideal principal**, si I es generado por un solo elemento. Esto es $I =< a >= aR$, para algún $a \in R$.*

Ahora estamos preparados para comenzar a estudiar la conexión que existe con las estructuras definidas anteriormente y los códigos cíclicos. En primer lugar es importante notar que el lugar donde sacamos los escalares, K es un campo, esto permite que al definir el conjunto de todos los polinomios de variable x sobre K, sea un **DI**, es decir tenemos que $K[x]$ es un **dominio de integridad**.

La propiedad que hemos usado de $K[x]$ en secciones previas, es la siguiente, ya que $K[x]$ es un dominio de integridad, no tiene divisores de cero, es decir:

Para cualesquiera $p(x), q(x) \in K[x]$, *tal que* $p(x)q(x) = \mathbf{0}$, *entonces* $p(x) = \mathbf{0}$ *o* $q(x) = \mathbf{0}$.

La propiedad enunciada anteriormente fue fundamental para poder encontrar una base a cualquier código cíclico dado su polinomio generador $g(x)$. Además de $K[x]$ se ha trabajado de forma indirecta sobre otra estructura definida a partir de la anterior, la cual definimos a continuación.

Definición 2.2.4. *Dados* $K[x]$ *y el polinomio* $1 + x^n$, *para algún* $n \in \mathbb{N}$, *definimos el anillo cociente* $K_n = K[x]/ < 1 + x^n >$, *donde* $< 1 + x^n >$ *es el ideal generado por* $1 + x^n$.

Así tanto $K[x]$ como K_n son ejemplos de **anillos conmutativos con unidad** (tal como se estudia de manera general en cursos de álgebra abstracta). Lo anterior es fácil de verificar ya que todas las operaciones de suma y producto (en el caso de K_n su producto se hace módulo $1 + x^n$ y esto permite además que las clases en K_n puedan ser tratadas como polinomios de grado a lo sumo $n - 1$) se han definido de manera natural y se comportan como esperamos. De esta manera cumple todos los axiomas de anillo.

Ahora estamos listos para probar un teorema que será de mucha importancia en este capítulo.

Teorema 2.2.5. *Un conjunto* S *de* K_n *corresponde a un código cíclico* C *de* K^n *si y sólo si* S *es un ideal de* K_n.

Observación: *No debe olvidar que la multiplicación en* K_n *es hecha módulo* $1 + x^n$.

Demostración. Para la primera implicación supongamos que S corresponde a un código cíclico C, así probaremos que S es un ideal de K_n.

La condición 2 de la *Definición 2.2.2* la cumple S, ya que al estar en correspondencia con C es cerrado bajo la suma y por lo tanto es un subgrupo aditivo de K_n. Basta demostrar la primera condición para demostrar que S es ideal de K_n. Así, sean $a(x)$ y $b(x) = b_0 + b_1x + b_2x^2 + \ldots + b_{n-1}x^{n-1}$ polinomios cualesquiera de los conjuntos S y K_n, respectivamente. Con estos polinomios efectuamos el producto de ellos.

$$\begin{aligned} a(x) * b(x) &= a(x)\left(b_0 + b_1x + b_2x^2 + \ldots + b_{n-1}x^{n-1}\right) \quad \text{mód } (1 + x^n) \\ &= b_0a(x) + b_1xa(x) + b_2x^2a(x) + \ldots + b_{n-1}x^{n-1}a(x) \text{ mód } (1 + x^n). \end{aligned}$$

A dicho polinomio le corresponde la palabra:

$$\begin{aligned} w &= b_0a + b_1\pi(a) + b_2\pi^2(a) + \ldots + b_{n-1}\pi^{n-1}(a) \\ &= w \in C, \quad \text{por ser } C \text{ un código cíclico.} \end{aligned}$$

Donde $w(x) \in S$, por la correspondencia de S con C.

Por lo tanto $a(x) * b(x) \in S$, para todo $a(x)$ y $b(x)$ que están en S y K_n, respectivamente. Esto permite concluir que S es ideal de K_n.

Ahora para la segunda implicación asumimos que S es ideal de K_n. Además definimos el conjunto C como el conjunto de palabras de longitud n que se forman con los coeficientes de los polinomios de S, por la relación tenemos que C es lineal, ya que S es un subgrupo aditivo (en otras palabras es cerrado bajo la suma).

Bastaría probar que el cambio cíclico de cualquier palabra de C también está en C. Así, sea $v \in C$ y por la correspondencia tenemos, $v(x) \in C$

$$\begin{aligned} &\Rightarrow x * v(x) \in S, \quad \text{ya que } S \text{ es ideal de } K_n. \\ &= xv(x) \text{ mód } (1 + x^n). \end{aligned}$$

Así tenemos que $\pi(v)$ le corresponde el polinomio $x * v(x) \in S \Rightarrow \pi(v) \in C$, por la correspondencia de S con C. Por lo tanto C es cíclico, así el conjunto al que corresponde S es un código cíclico. □

Una consecuencia del teorema anterior y el *Teorema 2.1.19*, es que debido a que los códigos cíclicos son generados por el polinomio $g(x)$ ($C =< g(x) >$; son generados por un solo elemento), tenemos que todos los ideales de K_n son ideales principales, ya que cumplen la *Definición 2.2.3*. Además tenemos que K_n está en correspondencia directa con K^n, por como se definió K_n.

Por lo tanto tenemos K_n es un **anillo de ideales principales** y las consecuencias que esto tendrá en el desarrollo del estudio de los códigos cíclicos lo estudiaremos en las siguientes secciones.

2.3. Codificación y decodificación polinomial

En esta sección trataremos el proceso de codificación de las palabras de un código cíclico C, mediante la matriz generadora. Además veremos la relación que esta tiene con el polinomio generador del código y la utilidad de dicho polinomio al momento de descodificar los mensajes.

La matriz generadora más simple para un código lineal cíclico, es aquella en la que las filas de dicha matriz son las palabras de código correspondientes al polinomio generador y sus primeros $k-1$ cambios cíclicos, es decir, la matriz está formada por una base de C.

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}.$$

Ejemplo 2.3.1. *Sea $C = \{(0,0,0,0),(1,0,1,0),(0,1,0,1),(1,1,1,1)\}$ un código lineal cíclico. El polinomio generador para C es $g(x) = 1 + x^2$.*

Aquí $n = 4$ y $deg(g(x)) = 2$, entonces $k = 2$ y una base para C la conforman

$$g(x) = 1 + x^2 \leftrightarrow (1,0,1,0) \quad y$$
$$xg(x) = x + x^3 \leftrightarrow (0,1,0,1).$$

Entonces, una matriz generadora para C es:

$$G = \begin{bmatrix} g(x) \\ xg(x) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Ejemplo 2.3.2. *Sea C el código lineal cíclico de longitud $n = 7$ con polinomio generador $g(x) = 1 + x + x^3$ de grado 3.*

Entonces $k = 4$ y una base para C la conforman los polinomios:

$$g(x) = 1 + x + x^3,$$
$$xg(x) = x + x^2 + x^4,$$
$$x^2g(x) = x^2 + x^3 + x^s,$$
$$x^3g(x) = x^3 + x^4 + x^6.$$

Y una matriz generadora para C es:

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

Si tenemos un código cíclico de dimensión k y polinomio generador $g(x)$ tal que $deg(g(x)) = n - k$, entonces su matriz generadora de $k \times n$, puede ser multiplicada por palabras de longitud k de modo que el producto esté bien definido.

Es así que los dígitos de información de la palabra $(a_0, a_1, \ldots, a_{k-1})$ se pueden codificar, multiplicando dicha palabra con la matriz generadora, de modo que esto devuelva la información ya codificada.

Es necesario mencionar que los k dígitos de la palabra $(a_0, a_1, \ldots, a_{k-1})$ a codificar, la podemos considerar como el polinomio $a(x) = a_0 + a_1x + \ldots + a_{k-1}x^{k-1}$, al cual llamaremos como *la información o mensaje polinomial* y la codificación consiste en multiplicar el polinomio $a(x)$ con $g(x)$, es decir, $a(x)g(x) = c(x)$ con el producto usual de polinomios, donde $c(x)$ es el mensaje codificado.

Todo este cambio es con el fin de obtener una mejora significativa a la hora de codificar, ya que en vez de almacenar la matriz generadora, sólo debemos almacenar el polinomio generador $g(x)$.

Dado que la operación utilizada en la codificación es una multiplicación usual de polinomios, el proceso inverso (división) nos ayudará a poder encontrar el mensaje polinomial más cercano, de tal manera que si dividimos el mensaje codificado en el receptor $c'(x)$ por $g(x)$, recuperaremos el mensaje polinomial $a'(x)$ que es cercano a $a(x)$.

Ejemplo 2.3.3. *Sea $g(x) = 1 + x + x^3$ y $n = 7$.*

Entonces $k = 4$. Sea $a(x) = 1 + x^2$ el polinomio del mensaje a codificar correspondiente a la palabra $a = (1, 0, 1, 0)$.

El mensaje $a(x)$ se codifica como $c(x) = a(x)g(x)$:

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

con $c = (1, 1, 1, 0, 0, 1, 0)$ como la palabra de código correspondiente.

Si el ruido afecta a $c(x)$ y lo que recibimos es $c'(x) = 1 + x + x^4 + x^6$, entonces el polinomio del mensaje correspondiente es $c'(x)/g(x) = a'(x) = 1 + x^3$, es decir, $a' = (1, 0, 0, 1)$.

Hasta este punto, es necesario considerar una matriz de control de paridad para dichos códigos, ya que nos ayuda en la detección de errores.

Sabemos que en un código C de longitud n y dimensión k, si se envía $c(x)$ y se recibe $w(x)$, con $w(x) = c(x) + e(x)$, a uno le gustaría calcular el síndrome y el polinomio de error más probable $e(x)$ (la palabra e que define el polinomio $e(x)$, es la palabra de error como se vio en el capítulo 1).

Definición 2.3.4. *Sea $s(x)$ un polinomio tal que $s(x) = w(x)$ mód $g(x)$, donde $deg(s(x)) < n - k$. Al polinomio $s(x)$ se le llamará **polinomio del síndrome**.*

Si $w(x) = c(x) + e(x)$ y además cada palabra de C la podemos escribir como producto de algún polinomio $a(x)$ por el polinomio generador, entonces $c(x) = a(x)g(x)$ y así:

$$\begin{aligned} w(x) &= c(x) + e(x) \\ w(x) &= a(x)g(x) + e(x). \end{aligned}$$

Entonces usando la *Definición 2.3.4*,

$$\begin{aligned} s(x) &= (a(x)g(x) + e(x)) \text{ mód } g(x) \\ s(x) &= \cancel{a(x)g(x) \text{ mód } g(x)} + e(x) \text{ mód } g(x) \\ s(x) &= e(x) \text{ mód } g(x). \end{aligned}$$

De lo que se deduce que el polinomio síndrome depende únicamente del error.

Definición 2.3.5. *Llamaremos* ***matriz de control de paridad*** *a la matriz H de dimensión $(n \times n-k)$ cuya i-ésima fila se corresponde con el polinomio $r_i(x)$, que satisface que $r_i(x) = x^i \text{ mód } g(x)$.*

Observación: El $deg(r_i) < n-k$. Esto es debido a que la matriz de control de paridad es formada en sus columnas por una base de $C^\perp$ y si C tiene dimensión k, entonces la dimensión de $C^\perp$ es $n-k$.

Veamos que al definir la matriz de control de paridad H de esta forma, cumple todas las propiedades vistas en el capítulo 1. Primero veamos que al multiplicar la palabra recibida, por H, nos devolverá el síndrome.

Si w es una palabra recibida, entonces:

$$w(x) = c(x) + e(x),$$

que se corresponde con:

$$\begin{aligned} w &= c + e \\ wH &= (c+e)H \\ &= \sum_{i=0}^{n-1} (c_i + e_i) r_i, \end{aligned}$$

que a su vez, se corresponde con:

$$\begin{aligned} \sum_{i=0}^{n-1} (c_i + e_i) r_i(x) &= \sum_{i=0}^{n-1} c_i r_i(x) + \sum_{i=0}^{n-1} e_i r_i(x) \\ &= \sum_{i=0}^{n-1} c_i x^i \text{ mód } g(x) + \sum_{i=0}^{n-1} e_i x^i \text{ mód } g(x) \\ &= \left(\sum_{i=0}^{n-1} c_i x^i \right) \text{ mód } g(x) + \left(\sum_{i=0}^{n-1} e_i x^i \right) \text{ mód } g(x) \\ &= c(x) \text{ mód } g(x) + e(x) \text{ mód } g(x) \\ &= \mathbf{0} + e(x) \text{ mód } g(x) \\ &= s(x). \end{aligned}$$

$\therefore wH = s.$

Ahora veamos que H anula a cualquier palabra de C. Sea $v \in C$,

$$vH = \sum_{i=0}^{n-1} v_i r_i,$$

que se corresponde con:

$$\begin{aligned}
\sum_{i=0}^{n-1} v_i r_i(x) &= \sum_{i=0}^{n-1} v_i x^i \text{ mód } g(x) \\
&= v_0 + v_1 x + \ldots + v_{n-1} x^{n-1} \text{ mód } g(x) \\
&= v(x) \text{ mód } g(x) \\
&= \mathbf{0}(x)
\end{aligned}$$

$\therefore vH = 0$.

Por lo tanto, H definida de esta forma, anula a todo C, entonces se tiene que $s(x) = \mathbf{0}$ si y sólo si $w(x)$ es una palabra de código y así, H es una matriz de control de paridad.

Ejemplo 2.3.6. *Sea $n = 7$, y $g(x) = 1 + x + x^3$. Entonces $n - k = 3$. Es decir la longitud de cada r_i es 3. Entonces H se forma de la siguiente manera:*

$$\begin{aligned}
r_0(x) &= 1 \text{ mód } g(x) = 1 \leftrightarrow (1,0,0) \\
r_1(x) &= x \text{ mód } g(x) = x \leftrightarrow (0,1,0) \\
r_2(x) &= x^2 \text{ mód } g(x) = x^2 \leftrightarrow (0,0,1) \\
r_3(x) &= x^3 \text{ mód } g(x) = 1 + x \leftrightarrow (1,1,0) \\
r_4(x) &= x^4 \text{ mód } g(x) = x + x^2 \leftrightarrow (0,1,1) \\
r_5(x) &= x^5 \text{ mód } g(x) = 1 + x + x^2 \leftrightarrow (1,1,1) \\
r_6(x) &= x^6 \text{ mód } g(x) = 1 + x^2 \leftrightarrow (1,0,1).
\end{aligned}$$

Así:

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

Si $w(x) = 1 + x^5 + x^6$ se recibe, con $w = (1,0,0,0,0,1,1)$, entonces $wH = s = (1,1,0)$ y $s(x) = 1 + x = 1 + x^5 + x^6 \mod (1 + x + x^3)$.

2.4. Encontrando códigos cíclicos

En la *Sección 2.1* vimos algunas propiedades importantes que satisface el polinomio generador de un código. En esta sección, retomaremos una de esas propiedades, que nos resultará muy útil en la construcción de códigos cíclicos.

Por el *Teorema 2.1.19* se sabe que para que $g(x)$ sea el polinomio generador de un código lineal cíclico de longitud n, debe suceder que $g(x)$ divida a $1 + x^n$ o que es lo mismo a decir que $g(x)$ sea un divisor de $1 + x^n$.

De modo que para construir un código lineal cíclico de longitud n y dimensión k, se debe encontrar un factor de $1 + x^n$ que tenga grado $n - k$, como lo anuncia el *Teorema 2.1.15*. Dicho proceso, se reduce a encontrar todos los factores irreducibles del polinomio $1 + x^n$.

Podemos hablar también del código cíclico $\{0\}$ que consiste únicamente de la palabra cero de longitud n, con generador $g(x) = 1 + x^n$ ya que $g(x) = \mathbf{0} = 1 + x^n$ mód $(1 + x^n)$ y K^n con generador $g(x) = 1$.

Definición 2.4.1. *Los códigos cíclicos K^n y $\{0\}$ son llamados **códigos cíclicos incorrectos**. A todos los demás códigos cíclicos exceptuando los incorrectos, se les llamará **códigos cíclicos adecuados**.*

Observación: Los códigos incorrectos son llamados como tal, debido a que no tienen sentido en la comunicación.

Ejemplo 2.4.2. *Para $n = 3$, $1 + x^3 = (1 + x)(1 + x + x^2)$ es la factorización de $1 + x^3$ en factores irreducibles.*

Es fácil averiguar cuáles son debido al campo K, las únicas raíces posibles en nuestros polinomios son 0 y 1. Si evaluamos ambos valores en $1 + x^3$, resulta que 1 es una raíz, dado que $1 + 1^3 = 1 + 1 = 0$, por tanto hay un factor $(x + 1)$ y de esta manera podemos encontrar todos los demás factores por el algoritmo de la división.

Entonces, hay dos códigos cíclicos adecuados de longitud 3. Uno de ellos tiene polinomio generador $g(x) = 1 + x$, dimensión $k = n - deg(g(x)) = 3 - 1 = 2$ y matriz generadora:

$$G = \begin{bmatrix} g \\ \pi(g) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

El código es $\mathbf{C} = \{(\mathbf{0},\mathbf{0},\mathbf{0}), (\mathbf{1},\mathbf{1},\mathbf{0}), (\mathbf{0},\mathbf{1},\mathbf{1}), (\mathbf{1},\mathbf{0},\mathbf{1})\}$.

El otro código tiene como polinomio generador $g(x) = 1 + x + x^2$ y matriz generadora

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

por lo que es el código $\mathbf{C} = \{(\mathbf{0},\mathbf{0},\mathbf{0}), (\mathbf{1},\mathbf{1},\mathbf{1})\}$.

Ejemplo 2.4.3. *Para $n = 6$, la factorización de $1 + x^6$ en factores irreducibles es:*

$$\begin{aligned} 1 + x^6 &= (1 + x^3)^2 \\ &= [(1 + x)(1 + x + x^2)]^2 \\ &= (1 + x)^2(1 + x + x^2)^2. \end{aligned}$$

Para encontrar los generadores de códigos cíclicos lineales adecuados de longitud 6, tenemos que formar todos los productos posibles de estos factores, excepto 1 y $1 + x^6$ ya que son incorrectos. Cada uno de dichos productos es el generador de un código lineal cíclico de longitud 6.

Estos productos que generan los códigos cíclicos lineales de longitud 6 y la dimensión de cada uno de ellos se pueden visualizar en la siguiente tabla:

Generador del código C	Dimensión del código C
$1 + x$	5
$1 + x + x^2$	4
$(1 + x)^2 = 1 + x^2$	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

Observación: Por el principio de la multiplicación se puede encontrar la cantidad de códigos cíclicos adecuados, ya que una vez obtenidos los factores irreducibles como en el ejemplo anterior, sabemos que por cada uno tenemos 3 posibles potencias: 0, 1 y 2 y dado que sólo son 2 factores, tenemos $3 \times 3 = 9$ códigos cíclicos, de los cuales 7 son adecuados y 2 incorrectos. Esto es porque cada código cíclico tiene un polinomio generador que es divisor de $1 + x^6$.

Teorema 2.4.4. *Si $n = 2^r s$ entonces $1 + x^n = (1 + x^s)^{2^r}$.*

Demostración. Si $n = 2^r s$, entonces por el *Teorema 1.2.4*, se tiene lo siguiente:

$$\begin{aligned} 1 + x^{2^r s} &= 1 + (x^s)^{2^r} \\ &= 1^{2^r} + (x^s)^{2^r} \\ &= (1 + x^s)^{2^r}. \end{aligned}$$

□

Corolario 2.4.5. *Sea $n = 2^r s$, donde s es impar y sea $1 + x^s$ el producto de z polinomios irreducibles distintos. Entonces hay $(2^r + 1)^z$ códigos cíclicos lineales de longitud n y $(2^r + 1)^z - 2$ códigos cíclicos lineales adecuados de longitud n.*

Demostración. Por el *Teorema 2.4.4* se sabe que si $n = 2^r s$, entonces

$$1 + x^n = (1 + x^s)^{2^r}$$

pero por hipótesis

$$1 + x^s = p_1 \cdot p_2 \cdot ... \cdot p_z.$$

donde todos los p_i son polinomios irreducibles distintos.

Así,

$$\begin{aligned} 1 + x^n &= (1 + x^s)^{2^r} \\ &= [p_1 \cdot p_2 \cdot \ldots \cdot p_z]^{2^r} \\ &= p_1^{2^r} \cdot p_2^{2^r} \cdot \ldots \cdot p_z^{2^r} \end{aligned}$$

donde por la observación anterior, cada factor tiene $2^r + 1$ posibles potencias. Es decir que hay $(2^r + 1)^z - 2$ códigos cíclicos adecuados de longitud n (usando el principio de la multiplicación y quitando los 2 códigos que son incorrectos). □

Ejemplo 2.4.6. *En el ejemplo 2.4.2 se muestra que* $1 + x^3$ *es el producto de dos polinomios irreducibles:* $1 + x$ *y* $1 + x + x^2$*. Al aplicar el Corolario 2.4.5 con* $n = 2^0 \cdot 3$*, entonces* $r = 0, s = 3$ *y* $z = 2$*, encontramos que hay* $(2^0 + 1)^2 = (2)^2 = 4$ *códigos cíclicos lineales de longitud* 3*, de los cuales 2 son adecuados.*

Por otro lado, en el ejemplo 2.4.3, para $1 + x^6$*, tenemos* $n = 6 = 2^1 \cdot 3$*, por lo que* $r = 1$*,* $s = 3$ *y* $z = 2$*, por lo tanto hay* $(2 + 1)^2 = 9$ *códigos cíclicos lineales de longitud* 6*, de los cuales 7 son adecuados.*

Como se puede observar en los ejemplos, los cálculos concuerdan con lo encontrado anteriormente.

2.5. Dual de un código cíclico

A continuación, estudiaremos el dual de un código cíclico, el cual también cumple ser cíclico y por ende, tener polinomio generador.

Teorema 2.5.1. *Si C es un código cíclico, entonces su dual $C^{\perp}$ también es un código cíclico.*

Demostración. Sea C un código cíclico de K^n con $dim(C) = k$, entonces se sabe que

$$\forall a \in C \text{ y } \forall b \in C^{\perp},\ a \cdot b = 0.$$

Entonces, si $a = (a_0, a_1, ..., a_{n-1})$ y $b = (b_0, b_1, ..., b_{n-1})$:

$$\begin{aligned} 0 &= (a_0, a_1, ..., a_{n-1}) \cdot (b_0, b_1, ..., b_{n-1}) \\ &= a_0b_0 + a_1b_1 + ... + a_{n-2}b_{n-2} + a_{n-1}b_{n-1} \\ &= a_{n-1}b_{n-1} + a_0b_0 + a_1b_1 + ... + a_{n-2}b_{n-2}, \quad \text{ya que la suma es conmutativa.} \\ &= (a_{n-1}, a_0, a_1, ..., a_{n-2}) \cdot (b_{n-1}, b_0, b_1, ..., b_{n-2}) \\ &= \pi(a) \cdot \pi(b). \end{aligned}$$

Por lo tanto, $\pi(a) \cdot \pi(b) = 0$, $\forall a \in C$ y $\forall b \in C^{\perp}$. (*)

Se sabe que por la *sección 2.1*, si $v \in C$ con $v \neq 0$ entonces el código C puede ser generado por v, de la siguiente manera:

$$C =< \{v, \pi(v), ..., \pi^{n-1}(v)\} > .$$

Así, $\forall a \in C$,

$$a = \alpha_0 v + \alpha_1 \pi(v) + ... + \alpha_{n-1}\pi^{n-1}(v).$$

Entonces $\forall b \in C^{\perp}$, se tiene:

$$\begin{aligned} a \cdot \pi(b) &= (\alpha_0 v + \alpha_1 \pi(v) + ... + \alpha_{n-1}\pi^{n-1}(v)) \cdot \pi(b) \\ &= \alpha_0 v \cdot \pi(b) + \alpha_1 \pi(v) \cdot \pi(b) + ... + \alpha_{n-1}\pi^{n-1}(v) \cdot \pi(b) \\ &= \alpha_0(v \cdot \pi(b)) + \alpha_1(\pi(v) \cdot \pi(b)) + ... + \alpha_{n-1}(\pi^{n-1}(v) \cdot \pi(b)) \\ &= \alpha_0(\pi^n(v) \cdot \pi(b)) + \alpha_1(\pi(v) \cdot \pi(b)) + ... + \alpha_{n-1}(\pi^{n-1}(v) \cdot \pi(b)) \\ &= \alpha_0(\cancel{\pi(\pi^{n-1}(v)) \cdot \pi(b)}) + \alpha_1(\cancel{\pi(v) \cdot \pi(b)}) + ... + \alpha_{n-1}(\cancel{\pi^{n-1}(v) \cdot \pi(b)}), \quad \text{por (*).} \\ &= \alpha_0(0) + \alpha_1(0) + ... + \alpha_{n-1}(0) \\ &= 0. \end{aligned}$$

Esto significa que $\forall b \in C^{\perp}$, $\pi(b) \in C^{\perp}$, ya que $\pi(b)$ anula a todo C y además $C^{\perp}$ es lineal dado que C lo es por definición. Por lo tanto, $C^{\perp}$ es un código lineal cíclico.

□

Para encontrar el polinomio generador del dual necesitamos relacionar el producto de polinomios y el producto punto de vectores.

Lema 2.5.2. *Sea $a \leftrightarrow a(x), b \leftrightarrow b(x)$ y $b' \leftrightarrow b'(x) = x^n b(x^{-1})$ mód $(1 + x^n)$, entonces $a(x)b(x)$ mód $(1 + x^n) = 0$ si y sólo si $\pi^i(a) \cdot b' = 0$ para $i = 0, 1, \ldots, n-1$.*

Demostración. Sea $c(x) = a(x)b(x)$ mód $1 + x^n$, donde $a = (a_0, a_1, \ldots, a_{n-1})$ y $b = (b_0, b_1, \ldots, b_{n-1})$. Dado que $x^k = x^{n+k}$ mód $(1 + x^n)$, el coeficiente del término x^k en $c(x)$, es:

$$c_k = a_k b_0 + a_{k+1} b_{n-1} + \ldots + a_{n-1} b_{k+1} + a_0 b_k + \ldots + a_{k-1} b_1.$$

Además como $a(x) = a_0 + a_1 x, + \ldots + a_{n-1} x^{n-1}$ y $b(x) = b_0 + b_1 x + \ldots + b_{n-1} x^{n-1}$, entonces,

$$\begin{aligned}
x^n b(x^{-1}) \text{ mód } (1 + x^n) &= x^n (b_0 + b_1 x^{-1} + b_2 x^{n-2} + \ldots + b_{n-1} x^{-n+1}) \text{ mód } (1 + x^n) \\
&= b_0 x^n + b_1 x^{n-1} + \ldots + b_{n-1} x \text{ mód } (1 + x^n) \\
&= b_{n-1} x + \ldots + b_2 x^{n-2} + b_1 x^{n-1} + b_0 x^n \text{ mód } (1 + x^n) \\
&= b_{n-1} x + \ldots + b_2 x^{n-2} + b_1 x^{n-1} + b_0 x^n + b_0 + b_0 \text{ mód } (1 + x^n) \\
&= (b_0 + b_{n-1} x + \ldots + b_2 x^{n-2} + b_1 x^{n-1}) + b_0 x^n + b_0 \text{ mód } (1 + x^n) \\
&= (b_0 + b_{n-1} x + \ldots + b_2 x^{n-2} + b_1 x^{n-1}) + b_0 (1 + x^n) \text{ mód } (1 + x^n) \\
&= (b_0 + b_{n-1} x + \ldots + b_2 x^{n-2} + b_1 x^{n-1}) \\
&\leftrightarrow (b_0, b_{n-1}, \ldots, b_2, b_1) = b'.
\end{aligned}$$

Note que si $\pi^{n-k}(a) = (a_k, a_{k+1}, \ldots, a_{n-1}, a_0, \ldots, a_{k-1})$ y $b' = (b_0, b_{n-1}, b_{n-2}, \ldots, b_1)$, se tiene:

$$\begin{aligned}
\pi^{n-k}(a) \cdot b' &= (a_k, a_{k+1}, \ldots, a_{n-1}, a_0, \ldots, a_{k-1}) \cdot (b_0, b_{n-1}, \ldots, b_2, b_1) \\
&= a_k b_0 + a_{k+1} b_{n-1} + \ldots + a_{n-1} b_{k+1}, a_0 b_k + \ldots + a_{k-1} b_1 \\
&= c_k.
\end{aligned}$$

Por lo tanto, $c_k = \pi^{n-k}(a) \cdot b'$, $\forall k = 0, 1, \ldots, n-1$.

($\Rightarrow$) Si $c(x) = a(x)b(x)$ mód $(1 + x^n) = \mathbf{0}$, significa que todos los coeficientes de $c(x)$ valen cero ($c_k = 0$), entonces $\pi^{n-k}(a) \cdot b' = 0$, $\forall k = 0, 1, \ldots, n-1$.

($\Leftarrow$) Si $c_k = \pi^{n-k}(a) \cdot b' = 0$, $\forall k = 0, 1, \ldots, n-1$. Entonces todos los coeficientes de $c(x)$ valen cero y entonces $c(x) = \mathbf{0}$, es decir, $a(x)b(x)$ mód $(1 + x^n) = \mathbf{0}$.

Finalmente, el conjunto de todos los $\pi^{n-k}(a)$, $\forall k = 0, 1, \ldots, n-1$ es: $\{\pi^n(a), \pi^{n-1}(a), \ldots, \pi(a)\} = \{a, \pi(a), \ldots, \pi^{n-1}(a)\}$ y es el conjunto de a con sus $n-1$ cambios cíclicos. Por lo tanto, hemos probado que $a(x)b(x) = \mathbf{0}$ mód $(1 + x^n)$ si y sólo si $\pi^i(a) \cdot b' = 0$, $\forall i = 0, ..., n-1$ donde $b' \leftrightarrow b'(x) = x^n b(x^{-1})$ mód $(1 + x^n)$.

□

Sea C un código lineal cíclico de longitud n y $g(x)$ el polinomio generador de C. Sabemos que $g(x)$ divide a $1 + x^n$ y, por lo tanto, hay un polinomio único $h(x)$, tal que $1 + x^n = g(x)h(x)$.

Por el *Lema 2.5.2* sabemos que $x^n h(x^{-1})$ está en $C^\perp$, ya que $1 + x^n = g(x)h(x)$, entonces $g(x)h(x) = \mathbf{0}$ mód $(1 + x^n)$. Y ahora, el siguiente teorema nos ayuda a encontrar el polinomio generador para el dual de un código cíclico.

Teorema 2.5.3. *Si C es un código cíclico lineal de longitud 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 $x^k h(x^{-1})$. Denotaremos al polinomio generador de $C^\perp$ como $g^\perp(x)$, es decir $g^\perp(x) = x^k h(x^{-1})$.*

Demostración. Como C tiene una dimensión k, $g(x)$ tiene un grado $n-k$ y, por lo tanto, $h(x)$ tiene un grado k. Ya que:

$$g(x)h(x) = 1 + x^n.$$

Además, tenemos que

$$g(x^{-1})h(x^{-1}) = 1 + (x^{-1})^n.$$

Y

$$\begin{aligned} x^n g(x^{-1})h(x^{-1}) &= x^n(1 + (x^{-1})^n). \\ x^{n-k} g(x^{-1}) x^k h(x^{-1}) &= 1 + x^n. \end{aligned}$$

Entonces

$$x^k h(x^{-1}) | 1 + x^n.$$

Así, por el *Teorema 2.1.19*, genera a un código cíclico.

Ahora probemos que $C^\perp =< x^k h(x^{-1}) >$, por construcción $g(x)h(x) = 1 + x^n = \mathbf{0}$ mód $(1+x^n)$.

Entonces $g(x)h(x) = \mathbf{0}$ mód $(1+x^n)$ y tomando $a(x) = g(x)$ y $b(x) = h(x)$, tenemos por el lema anterior:

$$\pi^i(g) \cdot b' = 0, \, \forall i \in \{0, 1, ..., n-1\},$$

donde

$$b' \leftrightarrow b'(x) = x^n h(x^{-1}) \text{ mód } (1+x^n).$$

Pero

$$x^n h(x^{-1}) = x^{n-k}(x^k h(x^{-1})).$$

Así, podemos reescribir la congruencia anterior, como:

$$\begin{aligned} b' \leftrightarrow b'(x) &= x^n h(x^{-1}) \text{ mód } (1+x^n) \\ &= x^{n-k}(x^k h(x^{-1})) \text{ mód } (1+x^n) \\ &= x^{n-k}(g^\perp(x)) \text{ mód } (1+x^n), \quad \text{donde} \quad g^\perp(x) = x^k h(x^{-1}) \\ &\leftrightarrow \pi^{n-k}(g^\perp). \end{aligned}$$

Entonces, b' la palabra asociada al polinomio $b'(x)$ es igual a $\pi^{n-k}(g^\perp)$.

$$\pi^i(g) \cdot \pi^{n-k}(g^\perp) = 0, \, \forall i = \{0, 1, ..., n-1\}.$$

Pero $C =< g, \pi(g), \pi^2(g), ..., \pi^{n-1}(g) >$, por ser $g(x)$ el polinomio generador de C.

Entonces $\pi^{n-k}(g^\perp)$ anula a cada elemento del generador de C, así $\pi^{n-1}(g^\perp)$ anula a todo C.

$$\forall a \in C,\ a \cdot \pi^{n-k}(g^\perp) = 0.$$

Entonces $\pi^{n-k}(g^\perp) \in C$, por definición de $C^\perp$.

Pero vimos que $C^\perp$ es cíclico, por lo tanto están todos sus cambios cíclicos de $\pi^{n-k}(g^\perp)$, en particular,

$$\begin{aligned} \pi^k(\pi^{n-k}(g^\perp)) \in C &\Leftrightarrow \pi^{n-k+k}(g^\perp) \in C \\ &\Leftrightarrow \pi^n(g^\perp) \in C \\ &\Leftrightarrow g^\perp \in C \\ &\Leftrightarrow x^k h(x^{-1}) \in C^\perp. \end{aligned}$$

Por lo tanto, $x^k h(x^{-1})$ es un factor de $1 + x^n$, que tiene grado k y así, el polinomio generador para el código lineal cíclico $C^\perp$ de dimensión $n - k$ es $g^\perp(x) = x^k h(x^{-1})$.

□

Ejemplo 2.5.4. *Sea $g(x) = 1 + x + x^3$ es el generador de un código cíclico de longitud 7 y dimensión $k = 7 - 3 = 4$.*

Dado que $g(x)$ es un factor de $1 + x^7$, ya que $1 + x^7 = g(x)h(x)$, entonces haciendo la división de $1 + x^7$ entre $g(x)$, podemos encontrar $h(x)$.

En este caso $h(x) = 1 + x + x^2 + x^4$. El generador para $C^\perp$ es $g^\perp(x) = x^4 h(x^{-1}) = x^4(1 + x^{-1} + x^{-2} + x^{-4}) = 1 + x^2 + x^3 + x^4$ que corresponde a $g^\perp = (1, 0, 1, 1, 1, 0, 0)$.

Claramente $g \cdot g^\perp = (1, 1, 0, 1, 0, 0, 0, 0) \cdot (1, 0, 1, 1, 1, 0, 0) = 0$ y $\pi^k(g) \cdot g^\perp = 0$. Además, tenga en cuenta que $g^\perp(x) \neq h(x)$.

Ejemplo 2.5.5. *Sea $g(x) = 1 + x + x^2$ el polinomio generador de un código cíclico lineal de longitud 6.*

Encontramos 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 polinomio generador para el código dual. Y note en este ejemplo que $g^\perp(x) = h(x)$.

Observación: En general, no siempre sucede que $g^\perp(x) \neq h(x)$.

Capítulo 3

Código de Hamming y código BCH corrector de errores dobles

3.1. Campos finitos

En esta sección, construiremos un campo finito, que nos ayudará junto con el anillo K_n a trabajar de manera más práctica con los códigos cíclicos.

Definición 3.1.1. *Denotaremos por $GF(n)$ a un campo finito de n elementos.*

Observación: Las siglas ***GF*** hacen alusión a "Galois Field", que se traduce a *Campo de Galois*.

Definición 3.1.2. *El polinomio $d(x) \in K[x]$ es un divisor o factor de $f(x)$ si $f(x) = g(x)d(x)$, para algún $g(x) \in K[x]$.*

Observación: Los polinomios **1** y $f(x)$ son claramente divisores de $f(x)$. A estos polinomios les llamaremos *divisores impropios*. Cualquier otro divisor, se dice que es un *divisor propio*.

Definición 3.1.3. *Un polinomio $f(x) \in K[x]$ se dice que es **irreducible**, si no tiene divisores propios en K[x]. De lo contrario se dice que es **reducible** (o factorizable) en $K[x]$.*

Observación: Los polinomios x y $1 + x$ son irreducibles en $K[x]$ por definición.

Teorema 3.1.4. *Dado $f(x) \in K[x]$ con $deg(f(x)) \geq deg(1 + x)$. El polinomio 1+x es un divisor de $f(x)$ si y sólo si 1 es raíz de $f(x)$ (es decir, $f(1) = 0$).*

Demostración. ($\Rightarrow$) Dado que $1 + x$ divide a $f(x)$, se tiene que:

$$f(x) = g(x)(1 + x).$$

Si $x = 1$,

$$\begin{aligned} f(1) &= g(1)(1 + 1) \\ &= g(1)(0) \\ &= 0 \end{aligned}$$

Por lo tanto, ya que $f(1) = 0$, entonces 1 es raíz de $f(x)$.

($\Leftarrow$) Por hipótesis se sabe que 1 es raíz de $f(x)$, es decir, $f(1) = 0$ y que $deg(f(x)) \geq deg(1+x)$.

Así, por el algoritmo de la división existen $g(x)$ y $r(x)$ tal que

$$f(x) = g(x)(1+x) + r(x),$$

con

$$deg(r(x)) < deg(1+x) = 1 \text{ ó } r(x) = 0,$$

en donde la única forma que se cumpla lo anterior, es si $r(x) = 0$ ó $r(x) = 1$.

Si $x = 1$,

$$\begin{aligned} f(1) &= g(1)(1+1) + r(1) \\ &= g(1)(0) + r(1) \\ &= r(1). \end{aligned}$$

Pero como $f(1) = 0$, entonces $r(1) = 0$ y esto es posible sólo si $r(x) = 0$.

Por lo tanto, $f(x) = g(x)(1+x)$ y así, $1+x$ divide a $f(x)$. □

Teorema 3.1.5. *Dado $g(x) \in K[x]$ con $deg(g(x)) \geq deg(x)$. El polinomio x es un divisor de $g(x)$ sí y sólo si 0 es raíz de $g(x)$ ($g(0) = 0$).*

Demostración. ($\Rightarrow$) Dado que x divide a $g(x)$, se tiene que:

$$g(x) = h(x)(x).$$

Si $x = 0$,

$$\begin{aligned} g(0) &= h(0)(0) \\ &= 0 \end{aligned}$$

Por lo tanto, 0 es raíz de $g(x)$.

($\Leftarrow$) Por hipótesis se sabe que 0 es raíz de $g(x)$, es decir, $g(0) = 0$ y que $deg(g(x)) \geq deg(x)$.

Así, por el algoritmo de la división existen $q(x)$ y $r(x)$ tal que

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

con

$$deg(r(x)) < deg(x) = 1 \text{ ó } r(x) = 0,$$

en donde la única forma que se cumpla lo anterior, es si $r(x) = 0$ ó $r(x) = 1$.

Si $x = 0$,

$$\begin{aligned} g(0) &= h(0)(0) + r(0) \\ &= r(0). \end{aligned}$$

Pero como $g(0) = 0$, entonces $r(0) = 0$ y esto es posible sólo si $r(x) = 0$.

Por lo tanto, $g(x) = h(x)(x)$ y así, x divide a $g(x)$. □

Ejemplo 3.1.6. *El polinomio $1 + x + x^2$ no tiene raíces en K, entonces los polinomios x y $1 + x$, no son divisores. Por lo tanto, se concluye que es irreducible.*

Por otra parte, los polinomios x^2, $1 + x^2$, y $x + x^2$ no son irreducibles, ya que 0 es raíz de x^2 y $x + x^2$, lo que significa que tienen a x como divisor. Además, los polinomios $1 + x^2$ y $x + x^2$ tienen como raíz al 1, es decir, $1 + x$ es divisor.

Ejemplo 3.1.7. *Sea $f(x) = 1 + x + x^2 + x^3$. Si $x = 1$, $f(1) = 1 + 1 + 1 + 1 = 0$, entonces, $1 + x$ es un factor de $f(x)$.*

Por el algoritmo de la división se tiene que $f(x) = (1 + x)(1 + x^2) = (1 + x)(1 + x)^2 = (1 + x)^3$.

Por otro lado, sea $g(x) = 1 + x + x^3$, entonces si $x = 0$, $g(0) = 1 \neq 0$ y si $x = 1$, $g(1) = 1 \neq 0$, entonces $g(x)$ no tiene factores lineales. Por lo tanto, $g(x)$ es irreducible en $K[x]$, ya que si un polinomio cúbico es reducible, entonces debe tener al menos un factor lineal.

Ejemplo 3.1.8. *Sea $f(x) = 1 + x + x^4$. Dado que $f(0) \neq 0$ y $f(1) \neq 0$, $f(x)$ no tiene factores lineales.*

Entonces, si $f(x)$ es reducible, $f(x)$ debe tener un factor cuadrático reducible. La única cuadrática irreducible en $K[x]$ es $g(x) = 1 + x + x^2$. Después de dividir $g(x)$ en $f(x)$, encontramos un resto distinto de cero. Entonces $1 + x + x^2$ no es un factor de $f(x)$. Por lo tanto, $f(x)$ es irreducible en $K[x]$.

Definición 3.1.9. *Un polinomio irreducible en $K[x]$ de grado n, $n > 1$ se dice que es **primitivo** si no es un divisor de $1 + x^m$ para cualquier $m < 2^n - 1$.*

Ejemplo 3.1.10. *Dado que $1 + x + x^2$ no es un factor de $1 + x^m$ para $m < 3 = 2^2 - 1$, entonces es primitivo. De manera similar, $1 + x + x^3$ no es un factor de $1 + x^m$ para cualquier $m < 7 = 2^3 - 1$ y, por lo tanto, 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, donde $5 < 15 = 2^4 - 1$ y por lo tanto, $1 + x + x^2 + x^3 + x^4$ no es primitivo.

En la *Sección 2.1*, definimos K_n para poder hablar de la multiplicación de palabras mediante sus polinomios. Veamos con el siguiente ejemplo que K_n definido de esa manera, no es un campo.

Ejemplo 3.1.11. *Usaremos la multiplicación de polinomios módulo $1 + x^4$ para definir la multiplicación de palabras en K^4. Entonces,*

Al producto $(0, 1, 0, 1)(0, 1, 0, 1)$ le corresponde:

$$\begin{aligned}(x + x^3)(x + x^3) &= x^2 + x^6 \\ &= (x^2 + x^6) \text{ mód } (1 + x^4) \\ &= \mathbf{0}.\end{aligned}$$

A quien le corresponde la palabra $(0,0,0,0)$.

Entonces $(0,1,0,1)(0,1,0,1) = (0,0,0,0)$, *pero* $(0,1,0,1) \neq (0,0,0,0)$ *en* K^4. *Por lo tanto, dado que* K^4 *tiene divisores de cero, entonces no puede ser un campo bajo esta definición de multiplicación.*

Por definición, $K_n = K[x]/ < 1 + x^n >$ y sabemos que por los cursos elementales de álgebra, que si $K[x]$ lo efectuamos módulo $h(x)$, donde $h(x)$ es un polinomio irreducible sobre K, tendremos que $K_n = K[x]/ < h(x) >$ es un campo. Así, para que haya una correspondencia entre las palabras de K^n y $K_n = K[x]/ < h(x) >$, necesitamos que $h(x)$ sea de grado n y así, ambos conjuntos tendrían 2^n elementos debido al campo K y habría una correspondencia entre K^n y un campo.

En resumen, se tiene la definición siguiente:

Definición 3.1.12. *El campo* $GF(2^n)$ *generado por el polinomio irreducible* $h(x)$ *de grado* n, *es el anillo cociente*

$$GF(2^n) = K[x]/ < h(x) > .$$

Observación: Para la construcción de $GF(2^n)$ usaremos a menudo la correspondencia que existe entre $GF(2^n)$ y K^n.

Notación: Denotaremos como $\mathbb{1}$ a la palabra $(1,0,\ldots,0)$ y a su respectivo polinomio como **1**. De la misma manera, $\mathbb{0}$ denotará la palabra $(0,0,\ldots,0)$ y su respectivo polinomio **0**.

Ejemplo 3.1.13. *Dado el polinomio irreducible* $h(x) = 1 + x + x^4$ *(visto en el Ejemplo 3.1.8). Consideremos el campo* $GF(2^4) = K[x]/ < h(x) >$.

Para encontrar el producto $(0,1,0,1)(0,1,0,1)$ *note que:*

$$(0,1,0,1)(0,1,0,1) \text{ que se corresponde con } (x + x^3)(x + x^3).$$

Pero

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

Y

$$x^2 + x^6 \text{ mód } (1 + x + x^4) = x^3.$$

Por lo tanto $(0,1,0,1)(0,1,0,1) = (0,0,0,1)$ *y le corresponde el polinomio* x^3.

Ejemplo 3.1.14. *Consideremos la construcción de* $GF(2^3)$ *utilizando el polinomio primitivo* $h(x) = 1 + x + x^3$ *para definir la multiplicación. Hacemos esto calculando* x^i mód $h(x)$:

Palabra	$\leftrightarrow$	$x^i \mod h(x)$
$(1,0,0)$		$\mathbf{1}$
$(0,1,0)$		x
$(0,0,1)$		x^2
$(1,1,0)$		$x^3 = 1 + x$
$(0,1,1)$		$x^4 = x + x^2$
$(1,1,1)$		$x^5 = 1 + x + x^2$
$(1,0,1)$		$x^6 = 1 + x^2$

Para calcular $(1,1,0)(0,0,1)$ que se corresponde con $(1+x)(x^2)$, observe que de la tabla anterior $1 + x = x^3$ mód $h(x)$ así que:

$$\begin{aligned}(x^2)(1+x) &= x^2 \cdot x^3 \\ &= x^5 \\ &= 1 + x + x^2 \text{ mód } (h(x)).\end{aligned}$$

Así,

$$(1,1,0)(0,0,1) = (1,1,1).$$

Teorema 3.1.15. *Sea $GF(2^n)$ el campo formado por el polinomio primitivo $h(x)$ de grado n. Si $\beta \in K^n$ representa a la palabra x* mód $h(x)$*, entonces*

$$GF(2^n)\backslash\{0\} = \{\beta^i / i = 0, 1, 2, \ldots, 2^n - 2\}.$$

Demostración. Ya que a β le corresponde el polinomio x mód $h(x)$, entonces a la palabra β^i le corresponde el polinomio x^i mód $h(x)$. Además, notemos que si $\mathbf{1} = x^m$ mód $h(x)$, significa que $\mathbf{0} = 1 + x^m$ mód $h(x)$ y por tanto, $h(x)$ divide a $1 + x^m$.

Como $h(x)$ es primitivo, sabemos que $h(x)$ no divide a $1 + x^m$ para $m < 2^n - 1$ y por lo tanto $\beta^m \neq \mathbb{1}$ para $m < 2^n - 1$.

Además si $\beta^j = \beta^i$, para $j > i$, entonces $\beta^j + \beta^i = \mathbb{0} \Leftrightarrow \beta^i(\beta^{j-i} + \mathbb{1}) = \mathbb{0}$. Pero ya que estamos en un campo, tenemos que $\beta^i = \mathbb{0}$ ó $\beta^{j-i} + \mathbb{1} = \mathbb{0}$.

Ya que $\beta \neq \mathbb{0}$ entonces $\beta^i \neq \mathbb{0}$, así quien debe ser cero es $\beta^{j-i} + \mathbb{1}$, lo cual en términos de polinomios es equivalente a:

$$\begin{aligned}\beta^{j-i} &= x^{j-i} \text{ mód } h(x) \\ \Leftrightarrow \beta^{j-i} + \mathbb{1} &= (x^{j-i} + 1) \text{ mód } h(x) \\ \mathbb{0} &= (x^{j-i} + 1) \quad \text{mód } h(x).\end{aligned}$$

Pero por todo lo anterior eso significa que $j - i \geq 2^n - 1$. En resumen tenemos que si $\beta^j = \beta^i$, entonces $j - i \geq 2^n - 1$, con $j > i \geq 0$, que es equivalente a decir que si $j - i < 2^n - 1$, entonces

$\beta^j \neq \beta^i$, con $j > i \geq 0$. Por lo tanto tenemos que en el conjunto $\{\beta, \beta^2, \beta^3, \ldots, \beta^{2^n-1}\}$ ninguna de sus potencias se repite y dicho conjunto tiene $2^n - 1$ elementos distintos del cero, por lo tanto podemos hacer una correspondencia uno a uno con $GF(2^n)\backslash\{0\}$ (que también tiene $2^n - 1$).

Finalmente notemos que por la correspondencia entre $\{\beta, \beta^2, \beta^3, \ldots, \beta^{2^n-1}\}$ y $GF(2^n)\backslash\{0\}$, existe una potencia $k \leq 2^n - 1$ de β tal que sea la identidad. Así,

$\forall i$ tal que $1 \leq i \leq 2^n - 1$ y $i \neq k$, tenemos:

$$\begin{aligned} \beta^k \beta^i &= \beta^i \\ \beta^{k+i} &= \beta^i \\ \Rightarrow k + i - i &\geq 2^n - 1 \\ k &\geq 2^n - 1, \text{ pero } k \leq 2^n - 1. \\ \Rightarrow k &= 2^n - 1. \end{aligned}$$

Por lo tanto $\beta^k = \beta^{2^n-1} = \mathbb{1} = \beta^0$. Así tenemos:

$$GF(2^n)\backslash\{0\} = \{\beta^i / i = 0, 1, 2, \ldots, 2^n - 2\}.$$

□

Observaciones:

- De la demostración anterior vimos que siendo $h(x)$ polinomio primitivo de grado n y $\beta^{2^n-1} = \mathbb{1}$, en términos de polinomio se tiene que

$$\begin{aligned} \mathbf{1} &= x^{2^n-1} \text{ mód } h(x). \\ &\Leftrightarrow x^{2^n-1} + 1 = \mathbf{0} \text{ mód } h(x) \\ &\Leftrightarrow h(x) | x^{2^n-1} + 1 \end{aligned}$$

  Es decir, un polinomio primitivo de grado n, siempre divide a $1 + x^m$ cuando $m = 2^n - 1$.

- Cada palabra que no sea cero en K^n puede ser representada por alguna potencia de β y nos facilita la multiplicación en el campo.

Definición 3.1.16. *Un elemento $\alpha \in GF(2^n)$ se dice **primitivo** si $\alpha^m \neq \mathbb{1}$ para $1 \leq m < 2^n - 1$. Es decir, α es **primitivo** si cada palabra que no sea cero en $GF(2^n)$ se puede expresar como una potencia de α.*

Observación: Si se usa un polinomio primitivo para construir $GF(2^n)$, donde β es la palabra definida anteriormente, entonces β es un elemento primitivo.

Ejemplo 3.1.17. *Construya $GF(2^4)$ usando el polinomio primitivo $h(x) = 1 + x + x^4$ y escriba cada palabra como una potencia de β, donde a β le corresponde el polinomio x mód $h(x)$. Tenga en cuenta que $\beta^{15} = \mathbb{1}$.*

Palabra	Polinomio en x mód $h(x)$	Potencia de β
$(0,0,0,0)$	$\mathbf{0}$	$---$
$(1,0,0,0)$	$\mathbf{1}$	$\beta^0 = \mathbb{1}$
$(0,1,0,0)$	x	β
$(0,0,1,0)$	x^2	β^2
$(0,0,0,1)$	x^3	β^3
$(1,1,0,0)$	$x^4 = 1 + x$	β^4
$(0,1,1,0)$	$x^5 = x + x^2$	β^5
$(0,0,1,1)$	$x^6 = x^2 + x^3$	β^6
$(1,1,0,1)$	$x^7 = 1 + x + x^3$	β^7
$(1,0,1,0)$	$x^8 = 1 + x^2$	β^8
$(0,1,0,1)$	$x^9 = x + x^3$	β^9
$(1,1,1,0)$	$x^{10} = 1 + x + x^2$	β^{10}
$(0,1,1,1)$	$x^{11} = x + x^2 + x^3$	β^{11}
$(1,1,1,1)$	$x^{12} = 1 + x + x^2 + x^3$	β^{12}
$(1,0,1,1)$	$x^{13} = 1 + x^2 + x^3$	β^{13}
$(1,0,0,1)$	$x^{14} = 1 + x^3$	β^{14}

Para calcular $(0,1,1,0)(1,1,0,1) = \beta^5 \cdot \beta^7 = \beta^{12} = (1,1,1,1)$ ya que $(x + x^2)(1 + x + x^3) = x^5 \cdot x^7 = x^{12}$ mód $h(x)$.

3.2. Polinomios mínimos

En esta sección, definiremos la noción de polinomio mínimo para el campo $GF(2^n)$.

Definición 3.2.1. *Un elemento α en un campo $F = GF(2^n)$, se dice que es raíz de un polinomio $p(x) \in F[x]$ sí y sólo si $p(\alpha) = \mathbb{0}$. Es decir, si $p(x) = a_0 + a_1x + \ldots + a_kx^k$. Entonces*

$$p(\alpha) = a_0 + a_1\alpha + \ldots + a_k\alpha^k = \mathbb{0}.$$

Ejemplo 3.2.2. *Sea $p(x) = \mathbb{1} + x^3 + x^4$, y sea β el elemento primitivo en $GF(2^4)$ construido usando $h(x) = 1 + x + x^4$ (ver tabla del Ejercicio 3.1.17).*

$$\begin{aligned} p(\beta) = \mathbb{1} + \beta^3 + \beta^4 &= (1,0,0,0) + (0,0,0,1) + (1,1,0,0) \\ &= (0,1,0,1) \\ &= \beta^9. \end{aligned}$$

Así que β no es una raíz de $p(x)$. Sin embargo,

$$\begin{aligned} p(\beta^7) &= \mathbb{1} + \left(\beta^7\right)^3 + \left(\beta^7\right)^4 \\ &= \mathbb{1} + \beta^{21} + \beta^{28} \\ &= \mathbb{1} + \beta^6 + \beta^{13} \quad \textit{(ya que } \beta^{15} = \mathbb{1}\textit{)} \\ &= (1,0,0,0) + (0,0,1,1) + (1,0,1,1) \\ &= (0,0,0,0) \\ &= \mathbb{0}. \end{aligned}$$

Por lo tanto, β^7 es raíz de $p(x)$.

Definición 3.2.3. *El **orden** de un elemento α en un campo $GF(2^n)$, es el entero positivo más pequeño m, tal que $\alpha^m = \mathbb{1}$.*

Observación: Usando la *Definición 3.1.16* y *Definición 3.2.3*, si α es un elemento primitivo, entonces su orden es $2^n - 1$.

Definición 3.2.4. *Para cualquier elemento α en $GF(2^n)$, definimos el polinomio mínimo de α como el polinomio en $K[x]$ distinto del **0** de grado más pequeño que tiene a α como raíz.*

Notación: Sea $m_\alpha(x)$ el polinomio mínimo de α.

Observaciones:

- Dado que no es posible evaluar α en un polinomio de $K[x]$, se hace una identificación de $K[x]$ con $GF(2^n)[x]$ y así, por ejemplo, el polinomio $p(x) = 1 + x + x^3$ en $K[x]$, es identificado en $GF(2^n)[x]$ como $p(\alpha) = \mathbb{1} + \mathbb{1}\alpha + \mathbb{1}\alpha^3$.

- Tenga en cuenta que si α tiene orden m, (es decir, $\alpha^m = \mathbb{1}$), entonces α es una raíz de $1 + x^m$, por lo que cada elemento en $GF(2^n)$ es una raíz de algún polinomio en $K[x]$.

Teorema 3.2.5. *Sea $\alpha \neq \mathbb{0}$ un elemento de $GF(2^n)$ y sea $m_\alpha(x)$ el polinomio mínimo de α. Entonces*

(a) el polinomio $m_\alpha(x)$ es irreducible sobre $K[x]$;

(b) si $f(x)$ es cualquier polinomio sobre K tal que $f(\alpha) = \mathbb{0}$, entonces $m_\alpha(x)$ es un factor de $f(x)$;

(c) el polinomio mínimo es único;

(d) el polinomio mínimo $m_\alpha(x)$ es un factor de $1 + x^{2^n-1}$.

Demostración.

(a) Por contradicción, supongamos que existen $g(x)$ y $h(x)$ en $K[x]$ con $deg(g(x)) < deg(m_\alpha(x))$ y $deg(h(x)) < deg(m_\alpha(x))$ respectivamente, tal que

$$m_\alpha(x) = g(x)h(x).$$

Además, como $m_\alpha(\alpha) = \mathbb{0}$. Entonces, si $x = \alpha$,

$$m_\alpha(\alpha) = g(\alpha)h(\alpha) = \mathbb{0}.$$

Y como $GF(2^n)$ es un campo, entonces $g(\alpha) = \mathbb{0}$ ó $h(\alpha) = \mathbb{0}$ lo que conduce a una contradicción, dado que $m_\alpha(x)$ es el polinomio de menor grado tal que $m_\alpha(\alpha) = \mathbb{0}$, entonces $g(x) = \mathbf{1}$ ó $h(x) = \mathbf{1}$. Por lo tanto, $m_\alpha(x)$ es irreducible sobre $K[x]$.

(b) Por el algoritmo de división, dado que $deg(m_\alpha(x)) \leq deg(f(x))$,

$$f(x) = m_\alpha(x)g(x) + r(x),$$

donde $deg(r(x)) < deg(m_\alpha(x))$. Ahora, como $f(\alpha) = \mathbb{0}$, entonces

$$\begin{aligned} \mathbb{0} = f(\alpha) &= m_\alpha(\alpha)g(\alpha) + r(\alpha) \\ &= \mathbb{0} \cdot g(\alpha) + r(\alpha) \\ &= \mathbb{0} + r(\alpha) \\ &= r(\alpha). \end{aligned}$$

Tenemos que $r(\alpha) = \mathbb{0}$. Pero dado que $m_\alpha(x)$ es el polinomio de menor grado, tal que α es su raíz, no puede existir otro de grado menor que el de $m_\alpha(x)$ que satisfaga dicha propiedad. Es así que $r(x) = \mathbf{0}$. Por lo tanto, $f(x) = m_\alpha(x)g(x)$, y $m_\alpha(x)$ es un factor de $f(x)$.

(c) Supongamos que existen dos polinomios mínimos: $m'_\alpha(x)$ y $m_\alpha(x)$, tal que $m'_\alpha(\alpha) = \mathbb{0}$ y $m_\alpha(\alpha) = \mathbb{0}$. Pero por el literal b), dado $m_\alpha(x)$ polinomio mínimo y $m'_\alpha(x)$ otro polinomio en $K[x]$ con $m'_\alpha(\alpha) = 0$, entonces $m_\alpha(x)|m'_\alpha(x)$.

Análogamente, siendo $m'_\alpha(x)$ polinomio mínimo y $m_\alpha(x)$ otro polinomio en $K[x]$ con $m_\alpha(\alpha) = 0$, entonces $m'_\alpha(x)|m_\alpha(x)$.

Por lo tanto, dado que $m_\alpha(x)|m'_\alpha(x)$ y $m'_\alpha(x)|m_\alpha(x)$, entonces $m_\alpha(x) = m'_\alpha(x)$. Y así, $m_\alpha(x)$ es único.

(d) Sea β un elemento primitivo en $GF(2^n)$, entonces un elemento cualquiera en $GF(2^n)$ se escribe como $\alpha = \beta^i$. Entonces $\alpha^{2^n-1} = (\beta^i)^{2^n-1} = (\beta^{2^n-1})^i = \mathbb{1}^i = \mathbb{1}$ y así $\mathbb{1} + \alpha^{2^n-1} = \mathbb{1} + \mathbb{1} = \mathbb{0}$ y por lo tanto, α es una raíz de $1 + x^{2^n-1}$ y por el literal (b), $m_\alpha(x)$ es un factor de $1 + x^{2^n-1}$.

□

Si se desea encontrar el polinomio mínimo de $\alpha \in GF(2^n)$, sabemos que dicho polinomio debe tener como raíz a α y además ser el polinomio de menor grado que cumpla dicha propiedad. Es por ello que si en general tenemos un polinomio $p(x) = a_0 + a_1x + a_2x^2 + \ldots + a_mx^m$ del cual α es raíz, debe cumplir que $p(\alpha) = a_0 + a_1\alpha + a_2\alpha^2 + \ldots + a_m\alpha^m = \mathbb{0}$, lo que se deduce a encontrar una combinación lineal del conjunto $\{\mathbb{1}, \alpha, \alpha^2, \ldots, \alpha^m\}$, cuya suma sea $\mathbb{0}$. Sin embargo, se sabe que la dimensión de $GF(2^n)$ es n.

Si $m < n$, entonces el conjunto $\{a_0, a_1, \ldots, a_m\}$ puede ser linealmente independiente y así todos los a_i deben ser iguales a cero, pero entonces $p(x) = \mathbf{0}$, lo cual no puede ser, dado que buscamos el polinomio mínimo.

Si $m \geq n$, tendríamos el conjunto $\{a_0, a_1, \ldots, a_m\}$ de $m+1$ elementos, linealmente dependiente, cuya combinación lineal genera al polinomio cero, sin embargo como se desea el polinomio más pequeño en grado, se elige el de grado $m = n$, es así que todo nuestro trabajo se reduce a encontrar una combinación lineal del conjunto $\{\mathbb{1}, \alpha, \alpha^2, \ldots, \alpha^n\}$.

Notación: Se sabe que si $GF(2^n)$ es generado por un polinomio primitivo, cada elemento α de $GF(2^n)$ lo podemos representar como una potencia del elemento primitivo β, entonces al polinomio mínimo de α ($m_\alpha(x)$) lo representaremos por $m_i(x)$, donde $\alpha = \beta^i$.

Ejemplo 3.2.6. *Encuentre el polinomio mínimo de $\alpha = \beta^3$, con $\alpha \in GF(2^4)$ generado por el polinomio primitivo $h(x) = 1 + x + x^4$ (Ver tabla del Ejemplo 3.1.17).*

Sea $m_\alpha(x) = m_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$. Si $x = \alpha$, entonces

$$m_\alpha(\alpha) = m_3(\alpha) = \mathbb{0}.$$

Así,

$$\begin{aligned} \mathbb{0} &= a_0\mathbb{1} + a_1\alpha + a_2\alpha^2 + a_3\alpha^3 + a_4\alpha^4 \\ &= a_0\mathbb{1} + a_2\beta^3 + a_2\beta^6 + a_3\beta^9 + a_4\beta^{12} \\ &= a_0(1,0,0,0) + a_1(0,0,0,1) + a_2(0,0,1,1) + a_3(0,1,0,1) + a_4(1,1,1,1). \end{aligned}$$

Resolviendo para a_0, a_1, a_2, a_3, a_4, encontramos:

$$a_0 = a_1 = a_2 = a_3 = a_4 = 1 \;\; y \;\; m_\alpha(x) = 1 + x + x^2 + x^3 + x^4.$$

Teorema 3.2.7. *Sea α un elemento en $GF(2^n)$ con polinomio mínimo $m_\alpha(x)$, entonces $\{\alpha, \alpha^2, \alpha^4, \ldots, \alpha^{2^{n-1}}\}$ es el conjunto de todas las raíces de $m_\alpha(x)$. En particular, el grado de $m_\alpha(x)$ es $\left|\{\alpha, \alpha^2, \ldots, \alpha^{2^n-1}\}\right|$.*

Demostración. Por definición de $m_\alpha(x)$, se sabe que $m_\alpha(\alpha) = 0$.

Entonces,

$$\begin{aligned} m_\alpha(\alpha^{2^i}) &= (m_\alpha(\alpha))^{2^i} \\ &= (\mathbb{0})^{2^i} \\ &= \mathbb{0}, \forall i \in \mathbb{N}. \end{aligned}$$

Además, si $\alpha = \beta^j$, donde $\beta \in GF(2^n)$ es un elemento primitivo. Entonces,

$$\begin{aligned} \alpha^{2^n} &= (\beta^j)^{2^n} \\ &= (\beta^{2^n-1} \cdot \beta)^j \\ &= (\mathbb{1} \cdot \beta)^j \\ &= \beta^j \\ &= \alpha. \end{aligned}$$

Lo que significa que el ciclo se repite, lo cual nos hace pensar que el conjunto $\{\alpha, \alpha^2, \alpha^{2^2}, \ldots, \alpha^{2^{n-1}}\}$ tiene a todas las raíces de $m_\alpha(x)$.

Para comprobar lo anterior, supongamos que existe otra raíz, digamos γ, tal que $m_\alpha(\gamma) = \mathbb{0}$ y además que cumpla que $\gamma \notin \{\alpha, \alpha^2, \alpha^{2^2}, \ldots, \alpha^{2^{n-1}}\}$.

Por otro lado, $m_\gamma(x)$ es el polinomio mínimo de γ y dado que existe otro polinomio $m_\alpha(x)$ que tiene a γ como raíz, debe suceder que $m_\gamma(x)|m_\alpha(x)$, por el *Teorema 3.2.5, b)* y además $deg(m_\gamma(x)) < deg(m_\alpha(x))$.

Entonces, por el algoritmo de la división, se tiene que, existe $f(x)$ tal que $deg(f(x)) < deg(m_\alpha(x))$.

$$m_\alpha(x) = m_\gamma(x) f(x).$$

Si $x = \alpha$, entonces,

$$\begin{aligned} m_\alpha(\alpha) &= m_\gamma(\alpha) f(\alpha) \\ \mathbb{0} &= m_\gamma(\alpha) f(\alpha). \end{aligned}$$

Dado que $GF(2^n)$ es un campo, entonces $m_\gamma(\alpha) = \mathbb{0}$ ó $f(\alpha) = \mathbb{0}$. Pero en ambos casos, no puede suceder que exista otro polinomio de menor grado que el de $m_\alpha(x)$ tal que tenga a α como raíz, dado que el polinomio mínimo es único. Así el conjunto $\{\alpha, \alpha^2, \alpha^{2^2}, \ldots, \alpha^{2^{n-1}}\}$ tiene a todas las raíces de $m_\alpha(x)$.

Por lo tanto, el grado de $m_\alpha(x)$ es $\left|\{\alpha, \alpha^2, \ldots, \alpha^{2^n-1}\}\right|$. □

Observación: Tenga en cuenta que la cardinalidad del conjunto $\left|\{\alpha, \alpha^2, \ldots, \alpha^{2^n-1}\}\right|$ es por definición, sin repetición.

Ejemplo 3.2.8. *Sea $m_5(x)$ el polinomio mínimo de $\alpha = \beta^5$, donde $\beta^5 \in GF(2^4)$ (generado por $h(x)$ como en el Ejemplo 3.1.17).*

Por el Teorema 3.2.7, como $\{\alpha, \alpha^2, \alpha^4, \alpha^8\} = \{\beta^5, \beta^{10}\}$, las raíces de $m_5(\alpha)$ son β^5 y β^{10}, lo que significa que el grado $(m_5(x)) = 2$.

Así, $m_5(x) = a_0 + a_1 x + a_2 x^2$ y por lo tanto:

$$\mathbb{0} = a_0 \mathbb{1} + a_1 \beta^5 + a_2 \beta^{10}$$
$$\mathbb{0} = a_0(1,0,0,0) + a_1(0,1,1,0) + a_2(1,1,1,0).$$

Resolviendo para a_0, a_1 y a_2 se obtiene que $a_0 = a_1 = a_2 = \mathbb{1}$ y $m_5(x) = 1 + x + x^2$.

La siguiente tabla muestra los polinomios mínimos del resto de los elementos del campo en $GF(2^4)$ construido usando el polinomio $h(x) = 1 + x + x^4$.

Elemento de $GF(2^4)$	Polinomio mínimo
$\mathbb{0}$	x
$\mathbb{1}$	$1 + x$
$\beta, \beta^2, \beta^4, \beta^8$	$1 + x + x^4$
$\beta^3, \beta^6, \beta^9, \beta^{12}$	$1 + x + x^2 + x^3 + x^4$
β^5, β^{10}	$1 + x + x^2$
$\beta^7, \beta^{11}, \beta^{13}, \beta^{14}$	$1 + x^3 + x^4$

3.3. Código cíclico de Hamming

En esta sección se probará que los códigos de Hamming también son cíclicos y además se mejorará la decodificación, todo esto gracias a las herramientas desarrolladas en las secciones anteriores.

Definición 3.3.1. *El **código de Hamming** 1-corrector de errores de longitud $2^r - 1$, con $r \geq 2$, tiene una matriz H de control de paridad tal que sus filas son todas las palabras de tamaño r distintas de la palabra cero.*

Por la definición anterior, si β es un elemento primitivo de $GF(2^r)$, entonces las potencias de β que forman a todas las demás palabras de $GF(2^r)$, las podemos ubicar en H de la siguiente manera:

$$H = \begin{bmatrix} \mathbb{1} \\ \beta \\ \beta^2 \\ \vdots \\ \beta^{2^r-2} \end{bmatrix}.$$

Observación: Note que H es una matriz de tamaño $(2^r - 1) \times r$.

Teorema 3.3.2. *Un polinomio primitivo de grado r es el polinomio generador de un código cíclico de Hamming de longitud $2^r - 1$.*

Demostración. Sea β un elemento primitivo de $GF(2^r)$ y C el código de Hamming de longitud $2^r - 1$.

Así,

$$H = \begin{bmatrix} \mathbb{1} \\ \beta \\ \beta^2 \\ \vdots \\ \beta^{2^r-2} \end{bmatrix}_{(2^r-1)\times r}.$$

Si se recibe w tal que $w = (a_0, a_1, \ldots, a_{2^r-2})$, entonces $w(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{2^r-2}x^{2^r-2}$.

Por otro lado,

$$wH = (a_0, a_1, \ldots, a_{2^r-2}) \begin{bmatrix} \mathbb{1} \\ \beta \\ \beta^2 \\ \vdots \\ \beta^{2^r-2} \end{bmatrix}$$

$$\begin{aligned} wH &= a_0\mathbb{1} + a_1\beta + a_2\beta^2 + \ldots + a_{2^r-2}\beta^{2^r-2} \\ &= w(\beta) \end{aligned}$$

Ahora, no es necesario multiplicar la palabra recibida con la matriz H. Sino que, una vez recibida la palabra w, cuyo polinomio correspondiente es $w(x)$, se debe evaluar el elemento primitivo β en $w(x)$ para comprobar si dicha palabra ha sufrido error.

Por lo tanto,

$$wH = w(\beta). \tag{3.1}$$

Si $w \in C$, donde C es un código de Hamming de longitud $n = 2^r - 1$, con $r \geq 2$. Entonces,

$$wH = \mathbb{0}.$$

Así, $w(\beta) = \mathbb{0}$ y por el *Teorema 3.2.5 b)*, se tiene que $m_\beta(x)|w(x)$.

Por lo tanto,

$$\forall v \in C, \;\; m_\beta(x)|v(x). \tag{3.2}$$

Además, por el *Teorema 3.2.5 d)*, el polinomio mínimo $m_\beta(x)$ es un factor de $1 + x^{2^r-1}$, pero como $2^r - 1 = n$. Entonces $m_\beta(x)$ es el polinomio generador de un código cíclico de longitud $2^r - 1$.

Y así, $C =< m_\beta(x) >$. Además, $m_\beta(x)$ es un polinomio primitivo, para ver esto notemos que por el *Teorema 3.2.7*, el conjunto $\{\beta, \beta^2, \beta^4, \ldots, \beta^{2^{r-1}}\}$ tiene todas las raíces del polinomio mínimo $m_\beta(x)$, además dicho conjunto tiene r elementos y β es un elemento primitivo, es así, que por el *Teorema 3.1.15*, ninguna de esas raíces se repite, por lo tanto $m_\beta(x)$ tiene r raíces distintas y por tanto es de grado r, y como estamos trabajando con un código de Hamming, $r > 1$.

Por último asumamos que existe m entero positivo tal que $m < 2^r - 1$ y $m_\beta(x)|1 + x^m$. Así,

$$\begin{aligned} & 1 + x^m = a(x)m_\beta(x), \;\; \text{para algún } a(x). \\ \Rightarrow\; & \mathbb{1} + \beta^m = a(\beta)m_\beta(\beta) = \mathbb{0} \\ & \mathbb{1} + \beta^m = \mathbb{0} \\ & \beta^m = \mathbb{1}. \end{aligned}$$

Pero, $\beta^m = \mathbb{1}$ es una contradicción ya que m es una potencia menor a $2^r - 1$ y el orden de β por ser un elemento primitivo de $GF(2^r)$ es de hecho $2^r - 1$. Por lo tanto el código de Hamming C es generado por un polinomio primitivo de grado r como se quería verificar. □

Ejemplo 3.3.3. *Construimos $GF(2^3)$ con $p(x) = 1 + x + x^3$ polinomio primitivo, donde $r = 3$ y $n = 2^3 - 1 = 7$. Además $\beta = (0, 1, 0)$ es el elemento primitivo y β^i se corresponde con el polinomio $x^i \mod p(x)$.*

De esta manera, una matriz de control de paridad para un código de Hamming de longitud 7, es la siguiente:

$$H = \begin{bmatrix} \mathbb{1} \\ \beta \\ \beta^2 \\ \beta^3 \\ \beta^4 \\ \beta^5 \\ \beta^6 \end{bmatrix} \textit{ se corresponde con } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

Donde fácilmente se puede verificar que es la misma matriz de control de paridad del código cíclico que tiene como polinomio generador a $p(x) = m_\beta(x)$.

De la demostración del *Teorema 3.3.2.* se sabe que el polinomio generador de un código cíclico de Hamming es el polinomio primitivo $m_\beta(x)$ y si $w(x)$ es el polinomio correspondiente a la palabra recibida, entonces $w(x) = c(x) + e(x)$, donde $c(x)$ es una palabra del código.

Si $x = \beta$, entonces

$$\begin{aligned} w(x) &= c(x) + e(x) \\ w(x) &= m_\beta(x)a(x) + e(x) \\ w(\beta) &= \mathbb{0} + e(\beta) \\ w(\beta) &= e(\beta) \end{aligned}$$

Sin embargo, el código de Hamming corrige 1 error, es decir que el polinomio del error tiene sólo un $\mathbb{1}$ como coeficiente de alguna potencia de x. Lo que significa que $e(x) = x^j$, donde j es la posición donde se encuentra el 1 en e (teniendo en cuenta que j varía de 0 a $n-1$). Por lo tanto, $c(x) = w(x) + x^j$.

Ejemplo 3.3.4. *Suponga que $GF(2^3)$ se ha construido usando el polinomio primitivo $1 + x + x^3$. Entonces $m_1(x) = 1 + x + x^3$ es el polinomio generador para un código cíclico de Hamming de longitud 7. Supongamos que se recibe $w(x) = 1 + x^2 + x^3 + x^6$. Entonces*

$$\begin{aligned} w(\beta) &= \mathbb{1} + \beta^2 + \beta^3 + \beta^6 \\ &= (1,0,0) + (0,0,1) + (1,1,0) + (1,0,1) \\ &= (1,1,0) \\ &= \beta^3. \end{aligned}$$

Así, $e(x) = x^3$ y $c(x) = w(x) + x^3 = 1 + x + x^6$.

3.4. Códigos BCH

En esta sección estudiaremos un tipo de código, que corrige a lo sumo 2 errores, que es uno de los muchos códigos BCH existentes.

Definición 3.4.1. *El código BCH corrector de errores dobles de longitud* $2^r - 1$*, es el código generado por* $g(x) = m_\beta(x)m_{\beta^3}(x)$*, donde* β *es un elemento primitivo en* $GF(2^r)$ *con* $r \geq 4$*.*

Observación: Dado que $n = 2^r - 1$ y por el *Teorema 3.2.5 c) y d)*, $g(x)$ divide a $1 + x^n$ y así $g(x)$ es el polinomio generador para un código cíclico.

Ejemplo 3.4.2. *Sea* β *un elemento primitivo en* $GF(2^4)$ *construido con* $p(x) = 1 + x + x^4$ *(Ver tabla del Ejemplo 3.1.17)). Tenemos que* $m_1(x) = 1 + x + x^4$ *y* $m_3(x) = 1 + x + x^2 + x^3 + x^4$*. Por lo tanto*

$$g(x) = m_1(x)m_3(x) = 1 + x^4 + x^6 + x^7 + x^8.$$

Y $g(x)$ *es el polinomio generador para un código BCH corrector de errores dobles de longitud* 15*. De aquí en adelante, a este código se le llamará* C_{15}*.*

Lema 3.4.3. *La siguiente matriz* H *es una matriz de comprobación de paridad para el código BCH corrector de errores dobles de longitud* $2^r - 1$ *y* $r \geq 4$*, donde* β *es un elemento primitivo en* $GF(2^r)$*, y el polinomio generador es* $g(x) = m_1(x)m_3(x)$*.*

$$H = \begin{bmatrix} \beta^0 & \beta^0 \\ \beta & \beta^3 \\ \beta^2 & \beta^6 \\ \vdots & \vdots \\ \beta^i & \beta^{3i} \\ \vdots & \vdots \\ \beta^{2^r-2} & \beta^{3(2^r-2)} \end{bmatrix}.$$

Demostración. Antes de comenzar la prueba del lema, mostraremos por inducción el siguiente resultado:

$$\forall n \geq 2, \quad 2^{n-1} < 2^n - 1.$$

Si $n = 2$,

$$2^{2-1} = 2^1 = 2$$

y

$$2^2 - 1 = 4 - 1 = 3.$$

En efecto $2 < 3$.

Supongamos que se cumple para $n = k-1$, es decir, $2^{(k-1)-1} < 2^{k-1} - 1 \quad$ ó $\quad 2^{(k-2)} < 2^{k-1} - 1$.

Probaremos que se cumple para $n = k$.

$$\begin{aligned} 2^{k-1} &= 2^{k-2} \cdot 2 \\ &< (2^{k-1} - 1) \cdot 2 \quad \text{por hipótesis inductiva} \\ &= 2^k - 2 \\ &< 2^k - 1 \end{aligned}$$

Por lo tanto, $\forall n \geq 2, \quad 2^{n-1} < 2^n - 1$.

Ahora, para la prueba del lema, notemos que H es una matriz de dimensión $(2^r - 1) \times 2r$.

Si el código C es generado por $g(x) = m_\beta(x) m_{\beta^3}(x)$, analicemos qué dimensión tiene su matriz de control de paridad.

Por ser β un elemento primitivo, el grado de $m_\beta(x)$ es r, por el *Teorema 3.2.7* y *Teorema 3.1.15* . Además, dado que β es el elemento primitivo, en el conjunto $\{\beta, \beta^2, \beta^4, \ldots, \beta^{2^{r-1}}\}$ ninguno de sus elementos se repite.

Por el *Teorema 3.2.7*, todas las raíces de m_{β^3} están en el conjunto $\{\beta^3, (\beta^3)^2, (\beta^3)^{2^2}, \ldots, (\beta^3)^{2^{r-1}}\}$.

Asumamos que existe una de ellas que se repite, es decir, para $j > i$, con $i, j = \{0, 1, \ldots, r-1\}$, $(\beta^3)^{2^j} = (\beta^3)^{2^i}$.

Y dado que los exponentes de β se repiten por ciclos, lo anterior se cumple siempre que sus índices cumplan lo siguiente:

$$\begin{aligned} 3(2^j) &= 3(2^i) \text{ mód } (2^r - 1) \\ \Leftrightarrow 3(2^j) - 3(2^i) &= 0 \text{ mód } (2^r - 1) \end{aligned}$$

Lo anterior significa que $(2^r - 1) \mid 3(2^j) - 3(2^i) \Leftrightarrow (2^r - 1) \mid 3(2^j - 2^i)$, pero si $r \geq 4$, se tiene que $2^r - 1 > 3$ y así, $(2^r - 1)$ no puede dividir a 3, sino que $(2^r - 1) | \ (2^j - 2^i)$, lo que significa que

$$2^j = 2^i \text{ mód } (2^r - 1).$$

Y nos hemos encontrado con una contradicción, dado que usando el resultado que se probó al inicio de la prueba del lema, tenemos la siguiente cadena:

$$2^i < 2^j \leq 2^{r-1} < 2^r - 1.$$

Así, $deg(m_3(x)) = r$. Entonces, el grado de $g(x)$ es $2r$.

Por el *Teorema 2.1.15*, el $deg(g(x)) = n - k$, entonces $2r = 2^r - 1 - k$. Así, la dimensión del dual es $2r$.

Por lo tanto, la dimensión de la matriz de control de paridad para C es $(2^r - 1) \times 2r$. Y así H tiene las medidas exactas.

Notemos que las columnas son linealmente independientes, ya que por construcción, en la segunda fila, aparecen (β, β^3) y eso hace que hayan ceros en lugares donde las demás filas tienen unos.

Veámos que H es la matriz de control de paridad de C.

$\forall v \in C$,

$$\begin{aligned} vH &= (v(\beta), v(\beta^3)) \\ &= (\cancel{a(\beta)g(\beta)}, \cancel{a(\beta^3)g(\beta^3)}) \text{ ya que } g(\beta) = \mathbb{0} \text{ y } g(\beta^3) = \mathbb{0} \\ &= \mathbb{0}. \end{aligned}$$

Así, H anula a todo C. Por lo tanto, H es la matriz de control de paridad de un *código BCH corrector de errores dobles*. □

Ejemplo 3.4.4. *Por ejemplo, usamos $GF(2^4)$ construido en el Ejemplo 3.1.17, con el polinomio primitivo $p(x) = 1 + x + x^4$ para construir un código BCH corrector de errores dobles C_{15}. Definimos C_{15} como el código lineal con la matriz H de comprobación de paridad de 15×8 y el polinomio generador $m_1(x)m_3(x)$.*

$$H = \begin{bmatrix} \mathbb{1} & \mathbb{1} \\ \beta & \beta^3 \\ \beta^2 & \beta^6 \\ \beta^3 & \beta^9 \\ \beta^4 & \beta^{12} \\ \beta^5 & \mathbb{1} \\ \beta^6 & \beta^3 \\ \beta^7 & \beta^6 \\ \beta^8 & \beta^9 \\ \beta^9 & \beta^{12} \\ \beta^{10} & \mathbb{1} \\ \beta^{11} & \beta^3 \\ \beta^{12} & \beta^6 \\ \beta^{13} & \beta^9 \\ \beta^{14} & \beta^{12} \end{bmatrix} \textit{ se corresponde con } \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Lema 3.4.5. *Un código BCH corrector de errores dobles de longitud $n = 2^n - 1$ generado por $g(x) = m_1(x)m_3(x)$, corrige a lo sumo 2 errores.*

Demostración. Sea H la matriz de control de paridad para el código *BCH* corrector de errores dobles con polinomio generador $g(x) = m_1(x)m_3(x)$.

Supongamos que se recibe la palabra w a quien le corresponde el polinomio $w(x)$, entonces el síndrome de w es:

$$wH = (w(\beta), w(\beta^3)) = (s_1, s_3).$$

Donde las palabras s_1 y s_3 son de longitud r.

Además, si $wH = \mathbb{0}$, entonces no se produjo error en la transmisión. Si se produjo un error en la transmisión, recordemos que el polinomio $e(x) = x^i$ y se tiene que:

$$\begin{aligned} wH &= eH \\ &= (e(\beta), e(\beta^3)) \\ &= (\beta^i, \beta^{3i}) \\ &= (s_1, s_3) \end{aligned}$$

Así, $s_1^3 = s_3$.

Ahora consideremos que se cumple $s_1^3 = s_3$, además como $s_1 \in GF(2^r)$, entonces existe i tal que $s_1 = \beta^i$. Así,

$$\begin{aligned} s_1^3 &= s_3 \\ \beta^{3i} &= s_3 \end{aligned}$$

Entonces tenemos la palabra $(s_1, s_3) = (\beta^i, \beta^{3i})$. Por lo tanto se puede corregir un error y todos los pasos son reversibles.

Si se han producido dos errores en la transmisión, digamos que en las posiciones i y j, $i \neq j$, entonces $e(x) = x^i + x^j$ y $wH = eH = (e(\beta), e(\beta^3)) = (s_1, s_3)$. Por lo tanto, el síndrome wH viene dado por:

$$wH = (s_1, s_3) = (\beta^i + \beta^j, \beta^{3i} + \beta^{3j}).$$

Consideramos el sistema de ecuaciones resultante.

$$\begin{aligned} \beta^i + \beta^j &= s_1 \\ \beta^{3i} + \beta^{3j} &= s_3. \end{aligned}$$

Ahora tenemos la factorización:

$$(\beta^i + \beta^j)(\beta^{2i} + \beta^{i+j} + \beta^{2j}) = \beta^{3i} + \beta^{3j}.$$

Y

$$s_1^2 = (\beta^i + \beta^j)^2 = \beta^{2i} + \beta^{2j}.$$

Por lo tanto:

$$\begin{aligned} s_3 &= \beta^{3i} + \beta^{3j} \\ &= (\beta^i + \beta^j)(\beta^{2i} + \beta^{i+j} + \beta^{2j}) \\ &= s_1(s_1^2 + \beta^{i+j}). \end{aligned}$$

Así

$$\frac{s_3}{s_1} + s_1^2 = \beta^{i+j}.$$

Ahora β^i y β^j son raíces de la ecuación cuadrática:

$$x^2 + (\beta^i + \beta^j)x + \beta^{i+j} = \mathbb{0}.$$

Es decir, raíces de la ecuación:

$$x^2 + s_1 x + \left(\frac{s_3}{s_1} + s_1^2\right) = \mathbb{0}.$$

Ahora veamos que si la ecuación anterior tiene dos soluciones, podemos llegar a establecer los mismos síndromes. En términos generales, la ecuación anterior puede tener dos soluciones, una solución o ninguna solución. Analicemos por casos esta situación.

Primero consideremos que tiene dos soluciones distintas, sean β^i y β^j, dos raíces distintas tales que $i \neq j$. Por ser raíces tenemos el siguiente sistema de ecuaciones,

$$\begin{aligned}\beta^{2i} + s_1\beta^i + \frac{s_3}{s_1} + s_1^2 &= \mathbb{0}\\ \beta^{2j} + s_1\beta^j + \frac{s_3}{s_1} + s_1^2 &= \mathbb{0}.\end{aligned}$$

Ahora sumando ambas ecuaciones tenemos:

$$\begin{aligned}\beta^{2i} + \beta^{2j} + s_1(\beta^i + \beta^j) &= \mathbb{0}\\ s_1(\beta^i + \beta^j) &= \beta^{2i} + \beta^{2j}\\ s_1(\beta^i + \beta^j) &= (\beta^i + \beta^j)^2\\ s_1 &= \beta^i + \beta^j.\end{aligned}$$

Ahora sustituyendo en la segunda ecuación, tenemos:

$$\begin{aligned}\beta^{2j} + (\beta^i + \beta^j)\beta^j + \frac{s_3}{\beta^i + \beta^j} + (\beta^i + \beta^j)^2 &= \mathbb{0}\\ \beta^{2j} + \beta^{i+j} + \beta^{2j} + \frac{s_3}{\beta^i + \beta^j} + \beta^{2i} + \beta^{2j} &= \mathbb{0}\\ \beta^{i+j} + \frac{s_3}{\beta^i + \beta^j} + \beta^{2i} + \beta^{2j} &= \mathbb{0}\\ \frac{s_3}{\beta^i + \beta^j} &= \beta^{2i} + \beta^{i+j} + \beta^{2j}\\ s_3 &= (\beta^i + \beta^j)(\beta^{2i} + \beta^{i+j} + \beta^{2j})\\ s_3 &= \beta^{3i} + \beta^{3j}.\end{aligned}$$

Entonces tenemos la siguiente palabra formada por los síndromes s_1 y s_3.

$$(s_1, s_3) = (\beta^i + \beta^j, \beta^{3i} + \beta^{3j}).$$

En este caso es posible corregir dos errores. Por otro lado, si la ecuación $x^2 + s_1 x + \left(\frac{s_3}{s_1} + s_1^2\right) = \mathbb{0}$, tiene una solución (o sea raíz duplicada), tenemos la siguiente cadena de igualdades.

$$\begin{aligned}(x+\beta^i)^2 &= x^2 + s_1 x + \left(\frac{s_3}{s_1} + s_1^2\right)\\ x^2 + \beta^{2i} &= x^2 + s_1 x + \left(\frac{s_3}{s_1} + s_1^2\right).\end{aligned}$$

Y por igualdad de polinomios tenemos $s_1 = \mathbb{0}$. Pero esto es contradictorio ya que no podríamos formar la expresión:

$$\frac{s_3}{s_1} + s_1^2.$$

Esto, debido a que no podemos dividir por cero.

Finalmente, es claro en este punto que si la ecuación no tiene solución, entonces jamás podríamos saber con qué potencias formar cada síndrome. En conclusión, si la ecuación solo tiene una solución o ninguna solución, entonces no podemos saber dónde ocurren los errores.

Por lo tanto, podemos encontrar las posiciones de los errores al encontrar las soluciones de la ecuación $x^2 + s_1 x + \left(\frac{s_3}{s_1} + s_1^2\right) = \mathbb{0}$.

El código BCH no puede corregir 3 errores en general, es decir, no puede corregir todos los posibles errores de peso 3, ya que por ejemplo, si tomamos β^i, β^j y β^k, con $i \neq j \neq k$, es decir, todas las palabras distintas tales que:

$$\begin{aligned}\beta^k &= \beta^i + \beta^j\\ \mathbb{0} &= \beta^k + \beta^i + \beta^j,\end{aligned}$$

entonces tendremos que los síndromes serían:

$$\begin{aligned}(s_1, s_3) &= (\beta^k + \beta^i + \beta^j, \beta^{3k} + \beta^{3i} + \beta^{3j})\\ &= (\mathbb{0}, \beta^{3k} + \beta^{3i} + \beta^{3j}).\end{aligned}$$

Entonces, el polinomio que generaría este síndrome tiene como raíz a β y también los múltiplos de dicho polinomio. Por lo tanto, es imposible saber en qué posiciones ocurre el error, ya que no sabemos nada acerca de la naturaleza de β^{3k}, β^{3i} y β^{3j}. Así, el código BCH no puede corregir 3 errores en general.

Por las construcciones hechas, es posible corregir a lo sumo 2 errores, resolviendo las ecuaciones anteriores. Por lo tanto, el código BCH es corrector de errores dobles.

□

Teorema 3.4.6. *Para cualquier entero $r \geq 4$, existe un código BCH corrector de errores dobles de longitud $n = 2^r - 1$ y de dimensión $k = 2^r - 2r - 1$, con polinomio generador $m_1(x)m_3(x)$.*

Demostración. Por el *Lema 3.4.3*, existe dicho código de longitud $2^r - 1$, con polinomio generador $g(x) = m_1(x)m_3(x)$ y matriz de control de paridad

$$H = \begin{bmatrix} \beta^0 & \beta^0 \\ \beta & \beta^3 \\ \beta^2 & \beta^6 \\ \vdots & \vdots \\ \beta^i & \beta^{3i} \\ \vdots & \vdots \\ \beta^{2^r-2} & \beta^{3(2^r-2)} \end{bmatrix}.$$

y por el *Lema 3.4.5*, corrige 2 errores. Finalmente,

$$\begin{aligned} deg(g(x)) &= n - k \\ 2r &= 2^r - 1 - k \\ k &= 2^r - 2r - 1. \end{aligned}$$

□

3.5. Decodificación del código BCH corrector de errores dobles

Se describió un esquema de decodificación en el *Lema 3.4.4* para los código BCH corrector de errores dobles que fueron construidos en la última sección. A lo largo de esta, identificaremos una palabra binaria de longitud r con la potencia correspondiente de β, donde β es un elemento primitivo de $GF(2^r)$.

Además, recordemos que una matriz de control de paridad para el código BCH corrector de errores dobles, con polinomio generador $g(x) = m_1(x)m_3(x)$ es H, como se define en el *Lema 3.4.3*. Por último, definamos adecuadamente un polinomio que fue descrito en la demostración del *Lema 3.4.4*.

Definición 3.5.1. *Sean v y u elementos del campo $GF(2^r)$. Al polinomio:*

$$x^2 + vx + \left(\frac{u}{v} + v^2\right)$$

se le llamará ***polinomio localizador de errores****.*

Ahora veamos un ejemplo donde apliquemos las ideas descritas en la sección anterior.

Ejemplo 3.5.2. *Sea w y su polinomio respectivo $w(x)$ una palabra recibida con los síndromes:*

$$s_1 = (0,1,1,1) = w(\beta) \text{ y } s_3 = (1,0,1,0) = w(\beta^3).$$

Donde w se codificó utilizando C_{15}. Del Ejemplo 3.1.17, tenemos que s_1 corresponde a β^{11} y s_3 a β^8. Así,

$$\begin{aligned}\frac{s_3}{s_1} + s_1^2 &= \beta^8\beta^{-11} + \beta^{22} \\ &= \beta^{12} + \beta^7 \\ &= \beta^2.\end{aligned}$$

Con lo anterior, formamos el polinomio localizador de errores $x^2 + \beta^{11}x + \beta^2$ y encontramos que tiene raíces β^4 y β^{13}. Por lo tanto, podemos decidir que los errores más probables se produjeron en las posiciones 4 y 13 (es decir, $e(x) = x^4 + x^{13}$), por lo que el patrón de error más probable es:

$$(0,0,0,0,1,0,0,0,0,0,0,0,0,1,0).$$

Hemos llegado a un esquema para la decodificación IMLD para los códigos BCH 2-corrector de errores. Por el *Lema 3.4.4*, se determina el algoritmo siguiente:

Algoritmo 3.5.3. *(IMLD para códigos BCH 2-corrector de errores con polinomio generador $m_1(x)m_3(x)$) Sea w la palabra recibida de longitud $2^r - 1$ y H la matriz de control de paridad del código BCH corrector de errores dobles.*

1. *Calcule el síndrome $wH = (s_1, s_3) = (w(\beta), w(\beta^3))$.*
2. *Si $s_1 = s_3 = \mathbb{0}$, concluya que no se produjeron errores. Decodifique $c = w$ como la palabra más probable de haber sido enviada.*
3. *Si $s_1 = \mathbb{0}$ y $s_3 \neq \mathbb{0}$ entonces solicite la retransmisión.*
4. *Si $s_1^3 = s_3$, corrija un solo error en la posición i, donde $s_1 = \beta^i$.*
5. *Formar la ecuación cuadrática:*

$$x^2 + s_1 x + \frac{s_3}{s_1} + s_1^2 = \mathbb{0}. \tag{3.3}$$

6. *Si la ecuación (3.3) tiene dos raíces **distintas** β^i y β^j, corregir errores en las posiciones i y j.*
7. *Si la ecuación (3.3) no tiene dos raíces distintas en $GF(2^r)$, concluya que al menos tres errores se produjeron en la transmisión y solicite una retransmisión.*

Todos los ejemplos y ejercicios que siguen a continuación utilizan C_{15}, cuya matriz de control de paridad es mostrada en el *Ejemplo 3.4.4* y el polinomio generador $g(x)$ es mostrado en el *Ejemplo 3.4.2*.

Ejemplo 3.5.4. *Suponga que se recibe w y el síndrome es $wH = (0,1,1,1,1,0,1,0) = (\beta^{11}, \beta^8)$. Ahora.*

$$s_1^3 = (\beta^{11})^3 = \beta^{33} = \beta^3 \neq \beta^8 = s_3.$$

En este caso, la ecuación (3.3) es $x^2 + \beta^{11}x + \beta^2 = \mathbb{0}$. Como se muestra en el Ejemplo 3.5.2, esta ecuación tiene raíces β^4 y β^{13}. Entonces, corregimos los errores en las posiciones $i = 4$ y $j = 13$; en otras palabras, el patrón de error más probable es $u = (0,0,0,0,1,0,0,0,0,0,0,0,0,1,0)$ y $e(x) = x^4 + x^{13}$ es el polinomio de error presunto.

Ejemplo 3.5.5. *Suponga que el síndrome es $wH = (w(\beta), w(\beta^3)) = (\beta^3, \beta^9)$. Entonces $s_1^3 = (\beta^3)^3 = \beta^9 = s_3$. Por lo tanto, es más probable que haya ocurrido un solo error en la posición $i = 3$. El patrón de error más probable es $u = (0,0,0,1,0,0,0,0,0,0,0,0,0,0,0)$, y $e(x) = x^3$ es el polinomio de error.*

Ejemplo 3.5.6. *Supongamos que $w = (1,1,0,1,1,1,1,0,1,0,1,1,0,0,0)$ se recibe. El síndrome es:*

$$wH = (0,1,1,1,0,1,1,0) = (\beta^{11}, \beta^5) = (s_1, s_3).$$

Ahora, verificamos si ha ocurrido un error:

$$s_1^3 = (\beta^{11})^3 = \beta^{33} = \beta^3 \neq \beta^5 = s_3.$$

Dado que no cumple la igualdad, no ha ocurrido 1 errores. Entonces, verificamos si han ocurrido 2.

Para formar la ecuación cuadrática (3.3), primero calculamos lo siguiente:

$$\begin{aligned}\frac{s_3}{s_1} + s_1^2 &= \beta^5\beta^{-11} + (\beta^{11})^2 \\ &= \beta^9 + \beta^7 \\ &= (0,1,0,1) + (1,1,0,1) \\ &= (1,0,0,0) \\ &= \beta^0.\end{aligned}$$

En este caso, la ecuación (3.3) se convierte en:

$$x^2 + \beta^{11}x + \beta^0 = \mathbb{0}.$$

Luego, probamos los elementos de $GF(2^4)$ para encontrar sus raíces y encontramos que $x = \beta^7$ es raíz, como se muestra a continuación:

$$\begin{aligned}(\beta^7)^2 + \beta^{11}\beta^7 + \beta^0 &= \beta^{14} + \beta^3 + \beta^0 \\ &= (1,0,0,1) + (0,0,0,1) + (1,0,0,0) \\ &= (0,0,0,0).\end{aligned}$$

De la misma forma tenemos que β^8 es la otra raíz. Por lo tanto, corregimos errores en las posiciones $i = 7$ y $j = 8$; esto es $u = (0,0,0,0,0,0,0,1,1,0,0,0,0,0,0)$ es el patrón de error más probable. Decodificamos $v = w + u = (1,1,0,1,1,1,1,1,0,0,1,1,0,0,0)$ como la palabra enviada.

Ejemplo 3.5.7. *Suponga que se envía una palabra de código de C_{15} y se producen errores en las posiciones 2, 6 y 12. Luego, el síndrome wH es la suma de las filas 2, 6 y 12 de H, donde w es la palabra recibida. Así:*

$$\begin{aligned}wH &= (0,0,1,0,0,0,1,1) + (0,0,1,1,0,0,0,1) + (1,1,1,1,0,0,1,1) \\ &= (1,1,1,0,0,0,0,1) = (\beta^{10}, \beta^3) = (s_1, s_3).\end{aligned}$$

Ahora $s_1^3 = (\beta^{10})^3 = \beta^{30} = 1 \neq \beta^3 = s_3$. Calculamos:

$$\begin{aligned}\frac{s_3}{s_1} + s_1^2 = \beta^3\beta^{-10} + \beta^{20} &= \beta^8 + \beta^5 \\ &= (1,0,1,0) + (0,1,1,0) = (1,1,0,0) = \beta^4.\end{aligned}$$

Y luego formar la ecuación cuadrática es:

$$x^2 + \beta^{10}x + \beta^4 = \mathbb{0}.$$

Al probar cada uno de los elementos de $GF(2^4)$, vemos que esta ecuación no tiene raíces en $GF(2^4)$. Por lo tanto, IMLD para C_{15} concluye correctamente que ocurrieron al menos tres errores y solicitamos una retransmisión.

Apéndice A

Implementación en Octave de C_{15}

Durante el desarrollo del *Capítulo 3*, se construyó el código *BCH* corrector de errores dobles C_{15}. A continuación, se presenta la implementación del *Algoritmo 3.5.3* para el código C_{15} en el software Octave.

Primero se creó una función que permita saber qué palabra del campo $GF(2^4)$ le corresponde la potencia j. Los comandos para esta función se muestran a continuación.

```
function v=GF24(j)
GF24=[[1,0,0,0];

[0,1,0,0];

[0,0,1,0];

[0,0,0,1];

[1,1,0,0];

[0,1,1,0];

[0,0,1,1];

[1,1,0,1];

[1,0,1,0];

[0,1,0,1];

[1,1,1,0];
```

```
[0,1,1,1];

[1,1,1,1];

[1,0,1,1];

[1,0,0,1]];

i=rem(j,15);

v=GF24(i+1,:);
```

La segunda función creada fue la que nos permitió decidir si dos palabras v y u son iguales. Esta función devuelve 1 si dichas palabras son iguales y 0 si no lo son. En el cuadro siguiente se muestra como está definida.

```
function r=compara(v,u)

r=1;

for i=1:4

  if v(i)!=u(i)

    r=0;

  endif

endfor
```

La tercera función nos permite saber qué potencia le corresponde dentro del campo $GF(2^4)$ respecto al elemento primitivo $(0,1,0,0)$. A continuación vemos como está definida esta función.

```
function i=expoGF24(v)

GF24=[[1,0,0,0];

[0,1,0,0];

[0,0,1,0];
```

```
[0,0,0,1];
[1,1,0,0];
[0,1,1,0];
[0,0,1,1];
[1,1,0,1];
[1,0,1,0];
[0,1,0,1];
[1,1,1,0];
[0,1,1,1];
[1,1,1,1];
[1,0,1,1];
[1,0,0,1]];
for j=0:14
c=compara(v,GF24(j+1,:));
  if c==1
    i=j;
    break
  endif
endfor
```

Con las tres funciones definidas anteriormente se implementa el *Algoritmo 3.5.3* para el código C_{15}. Dicha implementación estaría definida como se muestra en el cuadro siguiente.

```
function v=DC15(w)

v=[];

s1=[0,0,0,0];

for i=0:14

  s1=s1+(w(i+1)*GF24(i));

endfor

s1=rem(s1,2);

s3=[0,0,0,0];

for i=0:14

  s3=s3+(w(i+1)*GF24(3*i));

endfor

s3=rem(s3,2);

r1=compara(s1,[0,0,0,0]);

r2=compara(s3,[0,0,0,0]);

if r1==1

  r3=0;

else

  r3=compara(GF24(3*expoGF24(s1)),s3);

endif

if r1==1&&r2==1

  fprintf('No se produjeron errores en la transmisión.\n')
```

```
  v=w;
elseif r1==1&&r2==0
  fprintf('Se necesita retransmisión.\n')
elseif r3==1
  i=expoGF24(s1);
  e=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
  e(i+1)=1;
  fprintf('El dígito cambiado está en la posicion:\n')
  i
  fprintf('La palabra que se envio es:\n')
  v=rem(w+e,2);
else
  exps1 = expoGF24(s1);
  exps3 = expoGF24(s3);
  for ri=0:14
   raiz=rem(GF24(2*ri)+GF24(ri+exps1)+GF24(exps3+15-exps1)+GF24(2*exps1),2);
   compararaiz = compara(raiz,[0,0,0,0]);
   if compararaiz==1;
    i=ri;
    break
   endif
  endfor
```

```
if i==14
  j=i;
else
  for rj=i+1:14
    raiz=rem(GF24(2*rj)+GF24(rj+exps1)+GF24(exps3+15-exps1)+GF24(2*exps1),2);
    compararaiz = compara(raiz,[0,0,0,0]);
    if compararaiz==1;
      j=rj;
      break
    endif
  endfor
endif
if i!=j
  e=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
  e(i+1)=1;
  e(j+1)=1;
  fprintf('Los dígitos cambiados están en las posiciones:\n')
  i
  j
  fprintf('La palabra que se envio es:\n')
  v=rem(w+e,2);
else
```

```
    fprintf('Se necesita retransmisión.\n')
  endif
endif
```

También se creó una función especial, la cual permite crear cualquier palabra del código C_{15}. El cuadro que aparece a continuación, muestra cómo está definida.

```
function v=C15(u)
g=[1,0,0,0,1,0,1,1,1];
v=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
for n=0:14
  for i=0:8
    for j=0:6
      if i+j==n
        v(n+1)= (g(i+1)*u(j+1))+v(n+1);
      endif
    endfor
  endfor
endfor
v=rem(v,2);
```

Para finalizar, supongamos que deseamos enviar la palabra $v = [0,1,0,1,0,0,1,0,1,1,0,0,0,0,1]$ del código C_{15} y en la transmisión, v sufre dos errores en las posiciones 1 y 9 (recordemos que las posiciones fueron enumeradas comenzando desde cero).

Entonces se recibe $w = [0,1,1,1,0,0,1,0,1,0,0,0,0,0,1]$. Así, aplicando la función definida para el *Algoritmo 3.5.3*, tenemos los siguientes resultados en pantalla.

```
DC15(w)

Los dígitos cambiados están en las posiciones:

i = 2

j = 9

La palabra que se envió es:

ans =

0 1 0 1 0 0 1 0 1 1 0 0 0 0 1
```

Como vemos el *Algoritmo 3.5.3*, es capaz de corregir dos errores.

Conclusiones

1. *Se realizó un estudio detallado de los códigos cíclicos y de algunas aplicaciones: código BCH corrector de errores dobles y código cíclico de Hamming.*
2. *Gracias al polinomio generador de un código cíclico, hemos encontrado una forma más práctica de codificar las palabras.*
3. *Dado el anillo K_n, podemos saber todos los códigos cíclicos posibles, encontrando todos los divisores propios del polinomio $1 + x^n$.*
4. *Para mejorar la corrección de errores, fue necesario construir un campo para los síndromes: $GF(2^r)$ con r la longitud de los síndromes.*
5. *El polinomio mínimo de un elemento primitivo, es el polinomio generador de un código cíclico.*
6. *Encontramos una forma eficiente de decodificar la información para que los errores encontrados en la transmisión, puedan ser detectados y corregidos.*

Bibliografía

[1] D. R. Hankerson, Gary Hoffman, Douglas A. Leonard, Charles C. Lindner, Kevin T. Phelps, Chris A. Rodger, and James R. Wall. *Coding Theory and Cryptography: The Essentials*. CRC Press, 2000.

[2] Steven Roman. *Introduction to Coding and Information Theory*. Springer Verlag, 1996.

[3] Steven Roman. *Coding and Information Theory*, volume 134. Springer Science & Business Media, 1992.

[4] Vera Pless. *Introduction to the Theory of Error-Correcting Codes*, volume 48. John Wiley & Sons, 2011.

[5] Luisantos Bonilla Mejía. *Estudio de los códigos perfectos y nociones similares*. Tesis de licenciatura en matemática, Universidad de El Salvador, 2018.