Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Códigos BCH y de Reed-Solomon

Tesis presentada al

Colegio de Matemáticas

como requisito parcial para la obtención del grado de

Licenciado en Matemáticas Aplicadas

por

Luis Enrique Pineda Ramírez

Asesorado por

Dr. Carlos Alberto López Andrade

Puebla Pue.
Febrero de 2020

"En la vida, no hay nada que temer, sólo hay que comprender"

Marie Curie

Agradecimientos

Cualquier meta cumplida es, generalmente, imposible que sea de carácter individual. Esta no es la excepción, pues el culminar mi licenciatura no hubiese sido posible sin la ayuda e influencia de grandes personas y profesionales que encontré durante su transcurso. Por ello, considero necesario agradecer a las principales personas que durante este tiempo he considerado como vitales para llegar hasta el día de hoy.

Agradezco profundamente a mi familia, mi principal soporte en mi día a día durante toda mi vida. Mi madre merece una mención especial, es la persona más importante que he tenido, brindándome su apoyo de manera incondicional como nadie, nada de lo que vaya a lograr o haya logrado se entendería sin ella. A mi tía Esther, le debo agradecer infinitamente confiar en mi y el haberme dado todo el soporte que necesitaba para lograr exitosamente mi licenciatura. Para ambas, no me queda más que decir gracias, gracias por creer en mí.

Agradezco con toda el alma a mi asesor de tesis, el Dr. Carlos Alberto López Andrade, por permitirme realizar este trabajo bajo su tutela. Y hay que mencionarlo, es uno de los mejores profesionales y personas que algún alumno se pueda encontrar en su vida. De verdad, es un placer trabajar con él.

Agradezco a mis sinodales, Dr. Mauricio Gabriel Medina Bárcenas, Dr. Iván Fernando y Dr. César Cejudo Castilla por su tiempo y dedicación en la revisión de este trabajo, siendo importantes para una mejora positiva del mismo.

Ahora, de manera individual, agradezco al Dr. César Cejudo Castilla, por ser una de las personas más importantes que he tenido y pude haber encontrado en la facultad. Considero que él ha sido una de las piedras angulares del desarrollo de mi licenciatura, siendo también de los que más me ha motivado por incursionar en la bellísima álgebra.

Más allá de los profesores, es imposible imaginar cosas buenas sin haber encontrado compañeros, conocidos y amigos que hicieron de estos años los mejores de mi vida. Por todos ellos, le agradezco inmensamente a la vida por haberme dado el privilegio de conocerlos y haber convivido muchos momentos que difícilmente se olvidaran.

Finalmente, a cada profesor, profesional y trabajador que de alguna manera ha influido positivamente en mi, por más mínimo que sea, les agradezco su labor, pues contribuye significativamente a que muchas personas logren cumplir sus objetivos.

Índice general

Introducción

La teoría de códigos es un área activa dentro de la matemática actual por las virtudes de su objeto de estudio (códigos) y su relación con teorías matemáticas inmersas en aspectos fundamentales de la vida moderna, por ejemplo, la teoría de la información. Dada la importancia del manejo de datos, su representación y transmisión en la tecnología, los códigos suelen ser una herramienta primordial. Los enfoques de la teoría de códigos son variados, según sea la herramienta matemática que se utilice para su estudio. Los códigos algebraicos fundamentan su teoría en la sólida rama matemática conocida como álgebra y su característico enfoque. Este tipo de códigos serán un pilar fundamental de este trabajo. De gran interés es la detección y corrección de errores que suceden en la transmisión de datos digitales, pues los errores conllevan a recibir datos distintos a los originalmente transmitidos. Los códigos detectores-correctores realizan tal tarea, por lo que son imprescindibles hoy en día. Dentro de los códigos detectores-correctores de errores se encuentra una clase muy importante, la clase de los códigos lineales detectores-correctores de errores. Además, la estructura teórica de tales códigos es muy rica, lo que hace muy interesante su estudio. A su vez, dentro de los códigos lineales se encuentran otros de gran relevancia, como lo son los código cíclicos. Los códigos cíclicos tienen propiedades aún más interesantes desde el punto de vista algebraico. Finalmente, como una subclase destacada de los códigos cíclicos tenemos a los códigos BCH y a los Códigos Reed-Solomon, los cuales son bastante apreciados en las aplicaciones.

Esta tesis tiene como principal objetivo de estudio a los códigos BCH y a los códigos Reed-Solomon. La base de este trabajo se encuentra en el libro "Coding Theory A First Course" (c.f [6]) el cual aborda propiedades interesantes y relevantes de los códigos antes mencionados. Más allá de las propiedades teóricas, será de interés visualizar ideas y resultados en cuestiones de programación, lo que permita aterrizar y comprender la importancia del material analizado. Se presentarán 5 capítulos en este trabajo. Cada uno consta de lo siguiente:
Capítulo 1: En este capítulo se darán los preliminares para el resto de la tesis, principalmente abordando conceptos y resultados de las estructuras algebraicas que sobre las cuales se desarrollan los siguientes capítulos.
Capítulo 2: Respecto a este capítulo, se analizan los conceptos básicos de la teoría de códigos sobre campos finitos. Capítulo 3: Este es el capítulo principal de esta tesis, en el cual se analizan los códigos BCH, sus características, propiedades, además del desarrollo de un importante algoritmo de decodificación que permita llevar a la programación los resultados clave del capítulo.
Capítulo 4: Con el precedente del capítulo 3, se estudian los códigos Reed-Solomon. De interés son sus propiedades respecto a los códigos BCH en general, sus características, propiedades, capacidad de corrección de errores. También, un algoritmo de decodificación es analizado con el propósito de implementarse mediante programación.
Capítulo 5. Este capítulo aborda la programación de los algoritmos de decodificación presentados en los dos capítulos previos. El sustento de los programas se basa en resultados vistos en todo el trabajo. La idea es aterrizar los conceptos a un ámbito palpable.

Una razón importante de este trabajo es que tras el análisis de los códigos BCH y los códigos Reed-Solomon sea posible incursionar en el estudio de los códigos de Goppa. Los códigos de Goppa son una interesante clase de códigos lineales detectores-correctores de errores que permiten entender

a una clase más grande de códigos llamados alternantes, entre los que encontramos a los códigos BCH y los códigos de Goppa (cf. [6]).

Capítulo 1

PRELIMINARES

1.1. ESTRUCTURAS ALGEBRAICAS BÁSICAS

El contenido de este capítulo es sobre estructuras algebraicas que serán de utilidad en los capítulos siguientes. Definiciones, resultados y demostraciones son principalmente extraídos de los libros Galois Theory de J. Rotman (cf. [13]) y Finite Fields de H. Niederreiter y R. Lidl (cf. [9]). La mayoría de teoremas, lemas y proposiciones que se enuncian son demostrados, en caso contrario se citará una referencia que permita consultar una prueba del resultado pertinente.

En lo posterior, se considerará a $\omega = \{0, 1, 2, ...\}$ como el conjunto de los números naturales, el cual se denotará también por $\mathbb{N}_0$. Por $\mathbb{N}$ nos referimos al conjunto de los números naturales positivos.

Definición 1.1 i) $(S, \cdot)$ es un semigrupo si S es un conjunto no vacío y $\cdot : S \times S \longrightarrow S$ es una operación binaria que cumple lo siguiente:

$$\forall x, y, z \in S\text{: } (x \cdot y) \cdot z = x \cdot (y \cdot z).$$

ii) $(M, \cdot, e)$ es un monoide si $(M, \cdot)$ es un semigrupo y $e \in M$ es tal que para cada $x \in M$:

$$x \cdot e = e \cdot x = x.$$

iii) $(G, \cdot, e)$ es un grupo si $(G, \cdot, e)$ es un monoide y para cada $x \in G$ existe $y \in G$ tal que $x \cdot y = y \cdot x = e$.
iv) Al grupo $(G, \cdot, e)$ se le llama abeliano si para cada $x, y \in G$ se cumple lo siguiente:

$$x \cdot y = y \cdot x$$

Ejemplo 1.2 i) $(\mathbb{N}, +)$ es un semigrupo.

ii) $(\mathbb{N}, \cdot, 1)$ es un monoide.

iii) $(\mathbb{Z}, +, 0)$ es un grupo.

Definición 1.3 i) Un anillo (asociativo) R es un conjunto no vacío con dos operaciones binarias

$$+ : R \times R \longrightarrow R \text{ y } \cdot : R \times R \longrightarrow R,$$

llamadas suma y producto, respectivamente, tales que cumplen lo siguiente:
1) $(R, +, 0)$ es un grupo abeliano.
2) $(R, \cdot)$ es un semigrupo.
3) $\forall x, y, z \in R : (x \cdot (y + z) = x \cdot y + x \cdot z) \wedge ((y + z) \cdot x = y \cdot x + z \cdot x)$.

ii) Un anillo R es un anillo con identidad si existe $1 \in R$ tal que $(R, \cdot, 1)$ es un monoide.

iii) R es un anillo conmutativo si para cada $x, y \in R$ se cumple que $x \cdot y = y \cdot x$.

Ejemplo 1.4 i) El conjunto de los enteros pares $2\mathbb{Z}$ con la suma y producto usuales es un anillo asociativo sin identidad.

ii) El conjunto de los números enteros $\mathbb{Z}$ con la suma y producto usuales es un anillo asociativo con identidad, además es conmutativo.

iii) Las matrices de tamaño $n \times n$ con coeficientes en los números reales, junto con la suma y producto usuales, forman un anillo asociativo con identidad, en este caso el anillo no es conmutativo.

En el resto de este trabajo, cualquier anillo al que se refiera será un anillo con identidad.

Definición 1.5 i) Un elemento x de un anillo R se llama unidad si existe $y \in R$ tal que $xy = yx = 1$.

ii) Un anillo R tal que $0 \neq 1$, se dice que es un anillo con división si cada elemento de R distinto de cero es una unidad. De manera equivalente, R es anillo con división si $(R - \{0\}, \cdot, 1)$ es un grupo.

iii) Un anillo $\mathbb{F}$ es un campo si $\mathbb{F}$ es anillo con división conmutativo.

Ejemplo 1.6 1) Las unidades de $\mathbb{Z}$ son 1 y -1.

2) $\mathbb{Q}$, $\mathbb{R}$ y $\mathbb{C}$ con sus operaciones usuales de suma y producto son campos.

3) $\mathbb{Z}_p$ con p un número primo, es un campo con la suma y producto usuales entre clases.

4) El conjunto de los cuaterniones reales forman un anillo con división que no es un campo. Cabe resaltar, que todo anillo finito con división es un campo (Teorema de Wedderburn).

Definición 1.7 Sean R un anillo e $I \subseteq R$. Se dice que I es un ideal de R si se cumple lo siguiente:
1) I es un subgrupo de $(R, +, 0)$.
2) $\forall x \in R$, $\forall j \in I$: $xj, jx \in I$.

Definición 1.8 Sean R y R' anillos.

1) Un morfismo de anillos es una función $\varphi : R \longrightarrow R$ que cumple lo siguiente:
i) $\forall a, b \in R : \varphi(a + b) = \varphi(a) + \varphi(b)$
ii) $\forall a, b \in R : \varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$.

2) El núcleo de un morfismo de anillos $\varphi : R \longrightarrow R'$ se define y se denota como $ker(\varphi) := \{r \in R : \varphi(r) = 0\}$

3) Si un morfismo de anillos $\varphi : R \longrightarrow R'$ es biyectivo, entonces se dice que φ es isomorfismo y que R y R' son isomorfos, lo cual se denota $R \cong R'$.

Observación i) El núcleo de un morfismo de anillos $\varphi : R \longrightarrow R'$ es un ideal de R.
ii) No es difícil ver que un morfismo de anillos $\varphi : R \longrightarrow R'$ es inyectivo si y sólo si $ker(\varphi) = \{0\}$.

Definición 1.9 Sea R un anillo conmutativo.
i) Un elemento $x \in R - \{0\}$ se dice que es un divisor de cero si existe $b \in R - \{0\}$ tal que $xb = 0$.
ii) A R se le llama dominio entero si no tiene divisores de cero.

Sean R un anillo e I un ideal de R. Entonces se define la siguiente relación en R:

$$r_1 \sim_I r_2 \text{ si y sólo si } r_1 - r_2 \in I.$$

Nótese que para cada $r \in R$: $r - r = 0 \in I$. Si $r - s \in I$ entonces $s - r = -(r - s) \in I$. Finalmente, si $x - y, y - z \in I$ entonces $x - z = (x - y) + (y - z) \in I$. Todo lo anterior se cumple del hecho de que I es un subgrupo de $(R, +, 0)$. Por lo tanto, $\sim_I$ es una relación de equivalencia. La clases de equivalencia bajo esta relación son de la forma $r + I = \{r + j : j \in I\}$. Al conjunto $\frac{R}{I} = \{r + I : r \in R\}$ le podemos dar una estructura de anillo, tal como se analiza a continuación.

Proposición 1.10 Sea R un anillo e I un ideal de R. Entonces el conjunto $\frac{R}{I} = \{r + I : r \in R\}$ es un anillo con las siguientes operaciones binarias:

$$+ : \tfrac{R}{I} \longrightarrow \tfrac{R}{I}$$
$$(r + I, s + I) \longmapsto (r + s) + I,$$
$$\cdot : \tfrac{R}{I} \longrightarrow \tfrac{R}{I}$$
$$(r + I, s + I) \longmapsto (rs) + I.$$

Demostración. Supongamos que $r_1 + I = s_1 + I$ y $r_2 + I = s_2 + I$. Entonces $r_1 - s_1, s_2 - r_2 \in I$, lo cual implica que $(r_1 - s_1) - (s_2 - r_2) = (r_1 + r_2) - (s_1 + s_2) \in I$, de ahí que $(r_1 + r_2) + I = (s_1 + s_2) + I$. Por lo tanto la suma está bien definida. Como $r_1 - s_1, r_2 - s_2 \in I$, entonces existe $t_1, t_2 \in I$ tales que $r_1 - s_1 = t_1$ y $r_2 - s_2 = t_2$. Se sigue que $s_1 = r_1 - t_1$, $s_2 = r_2 - t_2$ y $s_1 s_2 = r_1 r_2 - r_1 t_2 - t_1 r_2 + t_1 t_2$, donde $k = -r_1 t_2 - t_1 r_2 + t_1 t_2 \in I$. De ahí que $(s_1 s_2) + I = (r_1 r_2) + k + I = (r_1 r_2) + I$. Por lo tanto, el producto está bien definido. Teniendo la buena definición de las operaciones, no es difícil comprobar que $(\frac{R}{I}, +, I)$ es un grupo abeliano, $(\frac{R}{I}, \cdot, 1 + I)$ es un monoide y se cumplen las leyes distributivas. ■

1.2. ANILLO DE POLINOMIOS

Definición 1.11 1) Sea R un anillo. Definimos una sucesión en R como una función $f : \mathbb{N}_0 \longrightarrow R$. Podemos pensar en $f(n) = a_n$ y $f = (a_0, a_1, ..., a_n, ...)$.

2) Dada una sucesión f en R definimos su soporte como $sop(f) = \{n \in \mathbb{N}_0 : f(n) \neq 0\}$.

3) Un polinomio con coeficientes en R es una sucesión en R con soporte finito. Al conjunto de polinomios con coeficientes en R lo denotamos por $R^{(\mathbb{N}_0)}$, es decir, $R^{(\mathbb{N}_0)} = \{f \in R^{\mathbb{N}_0} : sop(f) \text{ es finitio}\}$.

Ejemplo 1.12 $f = (1, 2, 0, 5, 1, \bar{0}, ...) \in \mathbb{Z}^{(\mathbb{N}_0)}$, donde $\bar{0}$ indica que su coordenada y las subsecuentes son iguales a 0. Entonces $sop(f) = \{0, 1, 3, 4\}$.

Definición 1.13 Sea R un anillo. Se define la operación binaria

$$+ : R^{(\mathbb{N}_0)} \times R^{(\mathbb{N}_0)} \longrightarrow R^{(\mathbb{N}_0)}$$
$$(f, g) \longmapsto f + g,$$

donde para cada $n \in \mathbb{N}_0$: $(f + g)(n) = f(n) + g(n)$ (la suma del lado derecho es en R).

Proposición 1.14 La operación binaria en $R^{(\mathbb{N}_0)}$ de la Definición 1.13 es, en efecto, una función.

Demostración. Basta comprobar que $sop(f+g)$ es finito. Sea $n \in sop(f+g)$, entonces $f(n)+g(n) = (f + g)(n) \neq 0$. Se sigue que $f(n) \neq 0$ ó $g(n) \neq 0$; en consecuencia $n \in sop(f) \cup sop(g)$, o bien, $sop(f+g) \subseteq sop(f) \cup sop(g)$. Ya que $sop(f)$ y $sop(g)$ son finitos, entonces $sop(f) \cup sop(g)$ es finito. Además, como subconjuntos de conjuntos finitos son finitos, se concluye que $sop(f + g)$ es finito, es decir, $f + g \in R^{(\mathbb{N}_0)}$. ■

Definición 1.15 Sea R un anillo. Definimos la operación binaria

$$\cdot : R^{(\mathbb{N}_0)} \times R^{(\mathbb{N}_0)} \longrightarrow R^{(\mathbb{N}_0)}$$
$$(f, g) \longmapsto f \cdot g,$$

donde para cada $n \in \mathbb{N}_0$: $(f \cdot g)(n) = \sum_{i+j=n} f(i)g(j)$.

Proposición 1.16. La operación binaria en $R^{(\mathbb{N}_0)}$ de la Definición 1.15 es, en efecto, una función.

Demostración. Sea $n \in sop(f \cdot g)$. Entonces, $0 \neq (f \cdot g)(n) = \sum_{i+j=n} f(i)g(j)$. Se sigue que existen $i, j \in \mathbb{N}$ tales que $f(i)g(j) \neq 0$, en consecuencia, $i \in sop(f)$ y $j \in sop(g)$. Es decir, para cada $n \in sop(f \cdot g)$ existe $(i, j) \in sop(f) \times sop(g)$ tal que $i + j = n$. Así, se tiene que $|sop(f \cdot g)| \leq |sop(f) \times sop(g)|$. Como $sop(f)$, $sop(g)$ son finitos, entonces $sop(f) \times sop(g)$ es finito. Por lo tanto, $sop(f \cdot g)$ es finito, es decir, $f \cdot g \in R^{(\mathbb{N}_0)}$. ■

Proposición 1.17 Sea R un anillo. Entonces $R^{(\mathbb{N}_0)}$ es un anillo con las operaciones definidas en 1.13 y 1.15, donde

$$\widehat{0} : \mathbb{N}_0 \longrightarrow R$$
$$n \longmapsto 0,$$
$$\widehat{1} : \mathbb{N}_0 \longrightarrow R$$
$$n \longmapsto \begin{cases} 1 & , n = 0 \\ 0 & , n \neq 0 \end{cases}$$

son los neutros de las operaciones $+$ y $\cdot$, respectivamente.

Definición 1.18 Sea R un anillo. Se definen los siguientes polinomios con coeficientes en R.

i) $x = (0, 1, \bar{0}, ...)$;

ii) Para $r \in R$, se define el polinomio constante $\widehat{r} = (r, \bar{0}, ...)$.

Observación. Si $f \in R^{(\mathbb{N}_0)}$, donde $f = (a_0, a_1, ..., a_n, \bar{0}, ...)$, entonces $f = a_0 + a_1x + ... + a_nx^n$, por lo que $R^{(\mathbb{N}_0)}$ puede ser considerado, de manera usual, como $R[x]$.

Definición 1.19 Sea $f \in R[x]$.

1) El grado de f, el cual denotamos por $grad(f)$, se define como

$$grad(f) = \text{máx}\{k \in \mathbb{N}_0 : f(k) \neq 0\}.$$

Nótese que la anterior definición no incluye al polinomio $\widehat{0}$.

2) Si $g = grad(f)$, entonces $f(g)$ es llamado el coeficiente principal de f.

3) Si el coeficiente principal de f es 1, entonces se dice que f es un polinomio mónico.

Lema 1.20 Si $\mathbb{F}$ es un campo, entonces $\mathbb{F}[x]$ es un dominio entero.

Demostración. Sean $f, g \in \mathbb{F}[x]$ distintos de cero. Basta demostrar que $grad(fg) = grad(f) + grad(g)$. Supongamos que $grad(f) = n$ y $grad(g) = m$. Entonces $f(i) = 0$ para cada $i > n$ y $g(j) = 0$ para cada $j > n$. Entonces $(fg)(n+m) = \sum_{i+j=n+m} f(i)g(j) = f(n)g(m) \neq 0$, de ahí que $grad(fg) = grad(f) + grad(g) = n + m$. ■

Teorema 1.21 Si $\mathbb{F}$ es un campo y $f(x), g(x) \in \mathbb{F}[x]$, con $g(x) \neq \widehat{0}$, entonces existen únicos $q(x), r(x) \in \mathbb{F}[x]$ tales que $f(x) = g(x)q(x) + r(x)$, donde $r(x) = \widehat{0}$ ó $grad(r) < grad(g)$.

Demostración. Sean $f(x), g(x) \in \mathbb{F}[x]$. Supongamos que $f(x) \neq \widehat{0}$ (en otro caso $f(x) = \widehat{0}g(x) + \widehat{0}$). La demostración se hará por inducción sobre $grad(f)$ teniendo los siguientes tres casos:
i) Si $grad(f) = 0 = grad(g)$, entonces f y g son polinomios constantes. Como $\mathbb{F}$ es un campo, entonces $f(x) = \frac{f(x)}{g(x)}g(x) + \widehat{0}$.
ii) Si $grad(f) < grad(g)$, entonces $f(x) = \widehat{0}g(x) + f(x)$.
iii) Si $n = grad(f) > grad(g) = m$, tenemos que $f(x) = a_0 + a_1x + ... + a_nx^n$ y $g(x) = b_0 + b_1x + ... + b_mx^m$. Si se multiplica a g por $\frac{a_n}{b_m}x^{n-m}$ y se resta a $f(x)$ se tiene que $f(x) - \frac{a_n}{b_m}x^{n-m}g(x) = \widehat{0}$ ó $grad(f(x) - \frac{a_n}{b_m}x^{n-m}) < grad(f)$. De lo anterior se derivan dos subcasos:
a) Si $f(x) - \frac{a_n}{b_m}x^{n-m}g(x) = \widehat{0}$, entonces $f(x) = \frac{a_n}{b_m}g(x) + \widehat{0}$.
b) Si $grad(f - \frac{a_n}{b_m}x^{n-m}) < grad(f)$, entonces por hipótesis de inducción, existen únicos $q(x), r(x) \in \mathbb{F}[x]$ tales que $f(x) - \frac{a_n}{b_m}x^{n-m}g(x) = g(x)q(x) + r(x)$, donde $r(x) = \widehat{0}$ ó $grad(r) < grad(g)$. Entonces, $f(x) = (\frac{a_n}{b_m}x^{n-m} + q(x))g(x) + r(x)$ con $r(x) = \widehat{0}$ ó $grad(r) < grad(g)$. ■

Teorema 1.22 Si $\mathbb{F}$ es un campo, entonces $\mathbb{F}[x]$ es un anillo de ideales principales, es decir, cada ideal de $\mathbb{F}[x]$ es de la forma $\langle g(x)\rangle := \{f(x)g(x) : f(x) \in \mathbb{F}[x]\}$ para algún $g(x) \in \mathbb{F}[x]$.

Demostración. Sea I un ideal de $\mathbb{F}[x]$.
i) Si $I = \{\widehat{0}\}$, entonces $I = \langle\widehat{0}\rangle$.
ii) Supongamos que $I \neq \{\widehat{0}\}$. Sea $A = \{n \in \mathbb{N} : grad(f) = n \text{ para algún } f \in I\}$. De la hipótesis de este caso, se sigue que $A \neq \emptyset$. Por el principio del buen orden, el conjunto A tiene elemento menor m. Entonces existe $h(x) \in I$ tal que $grad(h(x)) = m$, donde $h(x) = a_0 + a_1x + ... + a_mx^m$. Sea $g(x) = \frac{1}{a_m}h(x) \in I$. Veamos que $I = \langle g(x)\rangle$.
[$\subseteq$] Sea $f(x) \in I$. Entonces, por el algoritmo de la división, existen únicos $q(x), r(x) \in \mathbb{F}[x]$

tales que $f(x) = g(x)q(x) + r(x)$, donde $r(x) = \widehat{0}$ ó $grad(r) < grad(g)$. Si $r(x) \neq \widehat{0}$, entonces $r(x) = f(x) - g(x)q(x) \in I$ y $grad(r) < grad(g)$, lo cual es una contradicción. Por lo tanto, $r(x) = \widehat{0}$ y así tenemos que $f(x) = g(x)q(x) \in \langle g(x) \rangle$.
[$\supseteq$] Si $f(x) \in \langle g(x) \rangle$, entonces existe $q(x) \in \mathbb{F}[x]$ tal que $f(x) = q(x)g(x)$, de ahí que $f(x) \in I$, pues I es un ideal y $g(x) \in I$. ■

Definición 1.23 Sean $f(x), g(x) \in \mathbb{F}[x]$.

a) Un polinomio $d(x) \in \mathbb{F}[x]$ es llamado un máximo común divisor de $f(x)$ y $g(x)$ ($d(x) = mcd\{f(x), g(x)\}$) si cumple:
i) $d(x)|f(x)$ y $d(x)|g(x)$;
ii) Si $k(x) \in \mathbb{F}[x]$ es tal que $k(x)|f(x)$ y $k(x)|g(x)$, entonces $k(x)|d(x)$;
iii) $d(x)$ es mónico.

b) Un polinomio $m(x) \in \mathbb{F}[x]$ es llamado un mínimo común múltiplo de $f(x)$ y $g(x)$ ($m(x) = mcm\{f(x), g(x)\}$) si cumple:
i) $f(x)|m(x)$ y $g(x)|m(x)$
ii) Si $k(x) \in \mathbb{F}[x]$ es tal que $f(x)|k(x)$ y $g(x)|k(x)$, entonces $m(x)|k(x)$;
iii) $m(x)$ es mónico.

Se enuncia sin demostración (cf. [13]) la siguiente proposición.

Proposición 1.24 Sean $f(x), g(x) \in \mathbb{F}[x]$. Entonces el máximo común divisor y el mínimo común múltiplo de $f(x)$ y $g(x)$ existen y son únicos.

1.3. DOMINIOS DE FACTORIZACIÓN ÚNICA

Definición 1.25 Sea D un dominio entero.

1) Sea $p \in D - \{0\}$ un elemento que no es unidad. Entonces p es llamado irreducible si cualquier factorización $p = ab$ con $a, b \in D$, implica que a es unidad o b es unidad.

2) $a, b \in D$ son asociados si existe una unidad $u \in D$ tal que $b = au$.

Definición 1.26 Sea D un dominio entero. Se dice que D es un dominio de factorización única (DFU) si cumple las siguientes propiedades:

1) Cada $d \in D - \{0\}$ que no es unidad se puede factorizar como un producto finito de irreducibles.
2) Si $p_1 p_2 \cdot \ldots \cdot p_r$ y $t_1 t_2 \cdot \ldots \cdot t_s$ son dos factorizaciones en irreducibles de un mismo elemento, entonces $r = s$ y los $t'_j s$ se pueden renumerar de tal manera que p_i y t_i son asociados.

Lema 1.27 Sea D un dominio de ideales principales. Si $I_1 \subseteq I_2 \subseteq I_3 \subseteq \cdots$ es una cadena ascendente de ideales de D, entonces existe un entero positivo r tal que para cada $s \geq r$ se cumple $I_s = I_r$.

Demostración. Sea $I = \bigcup_{i \in \mathbb{N}} I_i$. Entonces I es un ideal de D. Para comprobar esto, obsérvese que $0 \in I_i$ para cada $i \in \mathbb{N}$, lo cual implica que $0 \in I$. Además, si $x, y \in I$ entonces existen $i, j \in \mathbb{N}$ tales que $x \in I_i$ y $y \in I_j$. Sin pérdida de generalidad, supongamos que $i \leq j$, entonces $x, y \in I_j$, lo

cual implica que $x+y \in I_j \subseteq I$. Si $r \in D$, entonces $rx \in I_j \subseteq I$, ya que I_j es un ideal de D. Por lo tanto, I es un ideal de D. Ya que los ideales de D son principales, entonces existe $g \in D$ tal que $I = \langle g \rangle$. Como $g \in \langle g \rangle = I$, entonces existe $r \in \mathbb{N}$ tal que $g \in I_r$. Se sigue que $I = \langle g \rangle \subseteq I_r \subseteq I$. Por lo tanto, $I = I_r = \langle g \rangle$.
Finalmente, obsérvese que para cada $s \geq r$ se tiene que $I = \langle g \rangle = I_r \subseteq I_s \subseteq I$, es decir, $I_s = I_r$. ■

Teorema 1.28 Sea D un dominio de ideales principales. Entonces un elemento $d \in D - \{0\}$ que no es unidad se puede factorizar como producto de irreducibles.

Demostración. Sea $d \in D - \{0\}$ que no es unidad.
i) Si d es irreducible la afirmación del teorema se sigue.
ii) Supongamos que d no es irreducible, entonces existen $a_1, b_1 \in D$ no unidades tales que $d = a_1 b_1$. Se sigue que $\langle d \rangle \subsetneq \langle a_1 \rangle$, pues si se tiene la igualdad entonces d y a_1 serían asociados (pues existirían $s_1, s_2 \in D$ tales que $a = s_1 d$ y $d = s_2 a$, lo cual implica que $a = s_1 s_2 a$ y en consecuencia $1 = s_1 s_2$, es decir, s_1 y s_2 son unidades y por lo tanto a_1 y d son asociados), lo cual implica que b_1 es unidad y se llega a una contradicción.
Si a_1 no es irreducible, entonces existen $a_2, b_2 \in D$ no unidades tales que $a_1 = a_2 b_2$, lo cual implica que $\langle d \rangle \subsetneq \langle a_1 \rangle \subsetneq \langle a_2 \rangle$. Continuando de esta manera, podemos formar una cadena ascendente $\langle d \rangle \subsetneq \langle a_1 \rangle \subsetneq ... \subsetneq \langle a_n \rangle \subsetneq ...$. Por el Lema 1.27, existe $r \in \mathbb{N}$ tal que $\langle a_r \rangle = \langle a_s \rangle$ para cada $s \geq r$. Por como se formo la anterior cadena, se sigue que a_r tiene que ser irreducible. Entonces $d = p_1 c_1$, donde $p_1 = a_r$ es irreducible y c_1 no es unidad (en caso contrario d y a_r serían asociados y en consecuencia d sería irreducible). Como c_1 no es unidad, se puede repetir el argumento anterior y encontrar p_2 irreducible y c_2 no unidad tal que $c_1 = p_2 c_2$. Continuando este proceso obtenemos una cadena ascendente $\langle d \rangle \subseteq \langle p_1 \rangle \subseteq \langle p_2 \rangle \subseteq ...$; por el Lema 1.27, existe $t \in \mathbb{N}$ tal que $\langle p_t \rangle = \langle p_s \rangle$ para cada $s \geq t$. Se sigue que $d = p_1 p_2 \cdot ... \cdot p_t c_t$ con cada factor irreducible (incluso c_t es irreducible pues la cadena $\langle d \rangle \subseteq \langle p_1 \rangle \subseteq \langle p_2 \rangle \subseteq ...$ tiene longitud t). ■

Teorema 1.29 Sea D un dominio de ideales principales.

1) $p \in D$ es irreducible si y sólo si $\langle p \rangle$ es un ideal máximo (no existe un ideal propio de D que contenga propiamente a $\langle p \rangle$).

2) Si $p \in D$ es irreducible y $p|ab$ (existe $c \in D$ tal que $ab = cp$) entonces $p|a$ o $p|b$.

3) Si $p \in D$ es irreducible y $p|a_1 a_2 \cdot ... \cdot a_k$, entonces $p|a_i$ para algún $i \in \{1, ..., k\}$.

Demostración. 1) [$\Longrightarrow$] Supongamos que $p \in D$ es irreducible. Sea J un ideal de D tal que $\langle p \rangle \subseteq J$. Como D es un dominio de ideales principales, existe $j \in D$ tal que $J = \langle j \rangle$. Dado que $p \in \langle p \rangle \subseteq J = \langle j \rangle$, existe $i \in D$ tal que $p = ij$. Se sigue que i es unidad ó j es unidad. De aquí se desprenden los siguientes dos casos:
Caso 1) Si j es unidad, entonces $1 = j^{-1} j \in J$, lo cual implica que $J = D$.
Caso 2) Si i es unidad, entonces $j = i^{-1} p \in \langle p \rangle$, lo cual implica que $J \subseteq \langle p \rangle$, es decir, $J = \langle p \rangle$.

[$\Longleftarrow$] Ahora, supongamos que $\langle p \rangle$ es un ideal máximo de D. Sean $a, b \in D$ tales que $p = ab$. Entonces $\langle p \rangle \subseteq \langle b \rangle$. De la hipótesis se tienen los siguientes dos casos:
Caso a) Si $\langle b \rangle = D$, entonces $1 \in \langle b \rangle$, de ahí que $1 = cb$ para algún $c \in D$, es decir, b es unidad.
Caso b) Si $\langle b \rangle = \langle p \rangle$, entonces $b = dp$ para algún $d \in D$. Se sigue que $p = ab = apd$ y en consecuencias $p(1 - ad) = 0$. Como D es un dominio entero, se tiene que $1 - ad = 0$ (p no es cero pues genera a un ideal máximo), es decir, a es unidad.
De los casos a) y b) se sigue que p es un elemento irreducible de D.

2) Sean $a, b, p \in D$ tales que $p|ab$ y p es un elemento irreducible. Supongamos que p no divide a a y demostremos que $p|b$. Entonces, $\langle p\rangle \subseteq \langle p\rangle + \langle a\rangle := \{r \in D : r = x + y, x \in \langle p\rangle, y \in \langle a\rangle\}$. Como $\subseteq \langle p\rangle + \langle a\rangle$ es un ideal, $\langle p\rangle$ es un ideal máximo (por el inciso 1)) y p no divide a a, se sigue que $\subseteq \langle p\rangle + \langle a\rangle = D$. Entonces $1 = sp + ta$ para algunos $s, t \in D$. Multiplicando la última igualdad por b se tiene que $b = bsp + bta$, donde $ab = mp$ para algún $m \in D$ ($p|ab$ por hipótesis), es decir, $b = bsp + tmp = (bs + tm)p$. Por lo tanto, $p|b$.

3) La demostración de este inciso se hará por inducción sobre $k \geq 2$.
i) Si $k = 2$ la afirmación es cierta por el inciso 2).
ii) Supongamos que la afirmación es valida para $k > 2$ y supongamos que $p|a_1a_2 \cdot ... \cdot a_ka_{k+1}$. Entonces $p|(a_1a_2 \cdot ... \cdot a_k)a_{k+1}$, de $i)$ se sigue que $p|a_{k+1}$ ó $p|a_1a_2 \cdot ... \cdot a_k$. Si $p|a_{k+1}$ hemos terminado. En otro caso, si $p|a_1a_2 \cdot ... \cdot a_k$, por hipótesis de inducción se tiene que existe $j \in \{1, ..., k\}$ tal que $p|a_j$. En cualquier caso, existe $i \in \{1, ..., k + 1\}$ tal que $p|a_i$.

■

Teorema 1.30 Si D es un dominio de ideales principales, entonces D es un dominio de factorización única.

Demostración. Sea D un dominio de ideales principales y $a \in D - \{0\}$ no unidad. Por el Teorema 1.28, existen $p_1, ..., p_r \in D$ irreducibles tales que $a = p_1 \cdot ... \cdot p_r$. Sea $a = q_1 \cdot ... \cdot q_s$ otra factorización de a en irreducibles. Entonces, por 3) del Teorema 1.29 se tiene que $p_1|q_j$ para algún $j \in \{1, ..., s\}$. Renumerando se tiene que $p_1|q_1$. Entonces existe $u_1 \in D$ tal que $q_1 = p_1u_1$ (ya que q_1 es irreducible), entonces p_1 y q_1 son asociados. Se sigue que $p_1p_2 \cdot ... \cdot p_r = (p_1u_1)q_2 \cdot ... \cdot q_s$, en consecuencia $p_2 \cdot ... \cdot p_r = u_1q_2 \cdot ... \cdot q_s$ (las leyes de cancelación son válidas en dominios enteros). Repitiendo el proceso para p_2 tenemos que $p_2 = u_2q_2$ con u_2 unidad, de ahí que $p_3 \cdot ... \cdot p_r = u_1u_2q_3 \cdot ... \cdot q_s$. Continuando (suponiendo sin pérdida de generalidad que $r \leq s$) tenemos que $1 = u_1 \cdot u_1u_2 \cdot ... \cdot u_rq_{r+1}q_{r+2} \cdot ... \cdot q_s$, lo cual implica que $q_{r+1}, q_{r+2}, ..., q_s$ son unidades, lo cual no es posible pues son irreducibles, entonces $r = s$. Por lo tanto, D es un dominio de factorización única. ■

Obsérvese que a la luz del Lema 1.20 y el Teorema 1.22, los anteriores resultados son aplicables a $\mathbb{F}[x]$ para algún campo $\mathbb{F}$.

Definición 1.31 Sea $f(x) \in \mathbb{F}[x]$. Diremos que $f(x)$ es irreducible sobre $\mathbb{F}$ si se cumple lo siguiente:
1) $grad(f(x)) > 0$
2) Si $f(x) = h(x)g(x)$ para algunos $h(x), g(x) \in \mathbb{F}[x]$, entonces $grad(h(x)) = 0$ ó $grad(g(x)) = 0$.

Observación. Sea $\mathbb{F}$ un campo. Las unidades de $\mathbb{F}[x]$ son todos los polinomios constantes distintos de cero, de ahí que en $\mathbb{F}[x]$ la definición 1.31 es equivalente a 1) de la definición 1.25.

1.4. EXTENSIONES DE CAMPO

Definición 1.32 Si $\mathbb{F}$ es un subcampo de un campo $\mathbb{E}$, entonces decimos que $\mathbb{E}$ es un campo de extensión de $\mathbb{F}$.

Teorema 1.33 Sean $\mathbb{F}$ un campo y $f(x) \in \mathbb{F}[x]$. Entonces existe un campo de extensión $\mathbb{E}$ de $\mathbb{F}$ y $\alpha \in \mathbb{E}$ tal que $f(\alpha) = 0$.

Demostración. Como $\mathbb{F}$ es un campo, entonces $\mathbb{F}[x]$ es un dominio de factorización única (Teorema 1.30). Sea $p(x) \in \mathbb{F}[x]$ un polinomio irreducible de la factorización en irreducibles de $f(x)$. Por 1) del Teorema 1.29, $\langle p(x)\rangle$ es un ideal máximo en $\mathbb{F}[x]$, por lo que $\frac{F[x]}{\langle p(x)\rangle}$ es un campo.
Sea $\psi : \mathbb{F} \longrightarrow \frac{\mathbb{F}[x]}{\langle p(x)\rangle}$ definida por $\psi(a) = a + \langle p(x)\rangle$. Ya que $Ker(\psi)$ es un ideal de $\mathbb{F}$ y $\mathbb{F}$ no tiene ideales no triviales, entonces $Ker(\psi) = \{0\}$ (pues ψ no es el morfismo cero). Se sigue que ψ es inyectiva y la función $\psi|^{\psi(\mathbb{F})} : \mathbb{F} \longrightarrow \psi(\mathbb{F})$ definida por $\psi|^{\psi(\mathbb{F})}(a) = \psi(a)$ es un isomorfismo.
Sean $\mathbb{E} = \frac{F[x]}{\langle p(x)\rangle}$ y $\alpha \in \mathbb{E}$ donde $\alpha = x + \langle p(x)\rangle$. Consideremos $\phi_\alpha : \mathbb{F}[x] \longrightarrow \mathbb{E}$ definida por $\phi_\alpha(\sum_{i=0}^{n} a_i x^i) = \sum_{i=0}^{n} \psi(a_i)\alpha^i$. Entonces, si $p(x) = a_0 + a_1 x + ... + a_n x^n$, entonces $\phi_\alpha(p(x)) = (a_0 + \langle p(x)\rangle)(1 + \langle p(x)\rangle) + (a_1 + \langle p(x)\rangle)(x + \langle p(x)\rangle) + ... + (a_n + \langle p(x)\rangle)(x^n + \langle p(x)\rangle) = (a_0 + a_1 x + ... + a_n x^n) + \langle p(x)\rangle = p(x) + \langle p(x)\rangle = 0$. Por lo tanto, la afirmación del teorema es cierta (con $\mathbb{F}$ identificado como $\psi(F)$ en $\mathbb{E}$). ■

Notación: Si $\mathbb{E}$ es un campo de extensión de $\mathbb{F}$, escribiremos $\mathbb{E}/\mathbb{F}$.

Lema 1.34 Sean $\mathbb{E}/\mathbb{F}$ una extensión de campo, $\alpha \in \mathbb{E}$ y $p(x) \in \mathbb{F}[x]$ un polinomio mónico irreducible que tiene a α como raíz. Entonces

i) Si $f(x) \in \mathbb{F}[x]$ es tal que $f(\alpha) = 0$, entonces $grad(f(x)) \geq grad(p(x))$.

ii) Si $q(x)$ es un polinomio mónico en $\mathbb{F}[x]$ tal que $grad(q(x)) = grad(p(x))$ y tiene a α como raíz, entonces $q(x) = p(x)$.

Demostración. i) Sea $I = \{q(x) \in \mathbb{F}[x] : q(\alpha) = 0\}$. Entonces I es un ideal de $\mathbb{F}[x]$. Dado que $\mathbb{F}[x]$ es un dominio de ideales principales, entonces existe $h(x) \in \mathbb{F}[x]$ tal que $I = \langle h(x)\rangle$. Ya que $p(x) \in I$, existe $q(x) \in \mathbb{F}[x]$ tal que $p(x) = q(x)h(x)$. Como $p(x)$ es irreducible entonces $q(x)$ es unidad ($h(x)$ no puede ser unidad pues tiene a α como raíz). Más precisamente, $q(x) = \widehat{1}$, pues $p(x)$ es mónico. Por lo tanto $p(x) = h(x)$. Si $f(x) \in \mathbb{F}[x]$ es tal que $f(\alpha) = 0$, entonces $f(x) \in I = \langle p(x)\rangle$, lo cual implica que $p(x)$ divide a $f(x)$, de ahí que $grad(f(x)) \geq grad(p(x))$.

ii) Supongamos que existe $q(x)$ un polinomio mónico en I tal que $grad(q(x)) = grad(p(x))$. Si $p(x) \neq q(x)$ entonces $q(x) - p(x) \neq 0$ y $grad(p(x) - q(x)) < grad(p(x))$, pero $p(\alpha) - q(\alpha) = 0$, lo cual contradice el inciso anterior. ■

Definición 1.35 Sea $\mathbb{E}/\mathbb{F}$ una extensión de campo. La dimensión de $\mathbb{E}$ como $\mathbb{F}$-espacio vectorial es llamada el grado de $\mathbb{E}$ sobre $\mathbb{F}$ y es denotado por $[\mathbb{E} : \mathbb{F}]$.

Teorema 1.36 Sea $p(x) \in \mathbb{F}[x]$ un polinomio irreducible de grado d. Entonces $\mathbb{E} = \frac{\mathbb{F}[x]}{\langle p(x)\rangle}$ es una extensión de $\mathbb{F}$ de grado d. Más aún, $\mathbb{E}$ contiene una raíz α de $p(x)$ y $\{1, \alpha, ..., \alpha^{d-1}\}$ es una base de $\mathbb{E}$ como $\mathbb{F}$-espacio vectorial.

Demostración. Sean $I = \langle p(x)\rangle$ y $\alpha = x + I$. Por el Teorema 1.33, resta demostrar que $\{1, \alpha, ..., \alpha^{d-1}\}$ es una base de $\mathbb{E}$.
Supongamos que $\{1, \alpha, ..., \alpha^{d-1}\}$ es linealmente dependiente. Entonces existen $c_0, c_1, ..., c_{d-1} \in \mathbb{F}$ no todos cero tales que $c_0 + c_1\alpha + ... + c_{d-1}\alpha^{d-1} = 0$. Se sigue que α es raíz del polinomio $f(x) = c_0 + c_1 x + ... + c_{d-1}x^{d-1} \in \mathbb{F}[x]$ con $grad(f(x)) < grad(p(x))$, lo cual es una contradicción a 1) del Lema 1.34. Por lo tanto, $\{1, \alpha, ..., \alpha^{d-1}\}$ es linealmente independiente.

Ahora, veamos que $\{1, \alpha, ..., \alpha^{d-1}\}$ genera a $\mathbb{E}$ como $\mathbb{F}$-espacio vectorial. Sea $t(x) \in \mathbb{F}[x]$. Por el algoritmo de la división, existen $q(x), r(x) \in \mathbb{F}[x]$ tales que $t(x) = q(x)p(x) + r(x)$, donde $r(x) = \widehat{0}$ ó $grad(r(x)) < d$. Entonces $t(x) - r(x) = q(x)p(x) \in I$, en consecuencia $t(x) + I = r(x) + I$. Si $r(x) = \widehat{0}$, se sigue que $t(x) + I = 0 \in \langle\{1, \alpha, ..., \alpha^{d-1}\}\rangle$ (subespacio generado por el conjunto $\{1, \alpha, ..., \alpha^{d-1}\}$). Supongamos que $r(x) \neq \widehat{0}$ tal que $r(x) = b_0 + b_1x + ... + b_mx^m$, donde $m < d$. Entonces $f(x)+I = r(x)+I = (b_0+b_1x+...+b_mx^m)+I = b_0(1+I)+b_1(x+I)+...+b_m(x+I)^m = b_0\alpha^0 + b_1\alpha + ... + b_m\alpha^m \in \langle\{1, \alpha, ..., \alpha^{d-1}\}\rangle$. Por lo tanto, $\{1, \alpha, ..., \alpha^m\}$ genera a $\mathbb{E}$ como $\mathbb{F}$-espacio vectorial. ■

Definición 1.37 i) Sea $\mathbb{E}/\mathbb{F}$ una extensión de campo y sean $\alpha_1, ..., \alpha_n \in \mathbb{E}$. Entonces $\mathbb{F}(\alpha_1, ..., \alpha_n)$ es llamado campo de adjunción de $\alpha_1, ..., \alpha_n$ a $\mathbb{F}$ y es la intersección de todos los subcampos de $\mathbb{E}$ que contienen a $\mathbb{F}$ y $\{\alpha_1, ..., \alpha_n\}$.

ii) Una extensión de campo $\mathbb{E}/\mathbb{F}$ es llamada simple si $\mathbb{E} = \mathbb{F}(\alpha)$ para algún $\alpha \in \mathbb{E}$.

Definición 1.38 i) Sean $\mathbb{E}/\mathbb{F}$ una extensión de campo y $\alpha \in \mathbb{E}$. Entonces α es llamado algebraico sobre $\mathbb{F}$ si α es raíz de algún polinomio mónico en $\mathbb{F}[x]$; en otro caso, α es trascendental sobre $\mathbb{F}$.

ii) Una extensión $\mathbb{E}/\mathbb{F}$ es llamada algebraica si todo elemento de $\mathbb{E}$ es algebraico sobre $\mathbb{F}$

Teorema 1.39 Si $\mathbb{E}/\mathbb{F}$ es una extensión finita, entonces $\mathbb{E}/\mathbb{F}$ es algebraica.

Demostración. Sean $[\mathbb{E} : \mathbb{F}] = n$ con $n \in \mathbb{N}$ y $\alpha \in \mathbb{E}$. Entonces el conjunto $\{1, \alpha, ..., \alpha^n\}$ es linealmente dependiente, ya que tiene $n + 1$ elementos. Entonces existen $c_0, c_1, ..., c_n \in \mathbb{F}$ no todos cero tales que $c_0 + c_1\alpha + ... + c_n\alpha^n = 0$. Se sigue que α es raíz de $f(x) = c_0 + c_1x + ... + c_nx^n \in \mathbb{F}[x]$. Por lo tanto, α es algebraico sobre $\mathbb{F}$. ■

Ahora, enunciamos un teorema bastante importante para resultados posteriores.

Teorema 1.40 Sea $\mathbb{E}/\mathbb{F}$ una extensión de campo y $\alpha \in \mathbb{E}$ un elemento algebraico sobre $\mathbb{F}$. Entonces se cumple lo siguiente:

1) Existe un polinomio mónico irreducible $p(x) \in \mathbb{F}[x]$ tal que $p(\alpha) = 0$;

2) $\frac{\mathbb{F}[x]}{\langle p(x)\rangle} \cong \mathbb{F}(\alpha)$;

3) $p(x)$ es el único polinomio de menor grado en $\mathbb{F}[x]$ que tiene a α como raíz;

4) $[\mathbb{F}(\alpha) : \mathbb{F}] = grad(p(x))$

Demostración. Sólo probaremos 1), 3) y 4). La demostración de 2) se puede consultar en el Teorema 47 de [13]. 1) Sea $I = \{f(x) \in \mathbb{F}[x] : f(\alpha) = 0\}$. Como α es algebraico sobre $\mathbb{F}$, entonces I es un ideal distinto de cero. Ya que $\mathbb{F}[x]$ es un dominio de ideales principales, existe $h(x) \in \mathbb{F}[x]$ mónico tal que $I = \langle h(x)\rangle$. Sea $p(x)$ el polinomio mónico de menor grado que genera a I (existe por el principio del buen orden). Veamos que $p(x)$ es irreducible. Sean $k(x), g(x) \in \mathbb{F}[x]$ tales que $p(x) = k(x)g(x)$. Entonces $grad(p(x)) = grad(k(x)) + grad(g(x))$ y $0 = p(\alpha) = k(\alpha)g(\alpha)$, lo cual implica que $k(\alpha) = 0$ ó $g(\alpha) = 0$. Si $k(\alpha) = 0$, entonces $k(x) \in I = \langle p(x)\rangle$, por consiguiente $p(x)|k(x)$ y $grad(p(x)) \leq grad(k(x)) \leq grad(k(x)) + grad(g(x)) = grad(p(x))$. Se sigue que $grad(g(x)) = 0$. Similarmente, si ocurre que $g(\alpha) = 0$ entonces $grad(k(x)) = 0$. Por lo tanto, $p(x) \in \mathbb{F}[x]$ es un polinomio mónico irreducible que tiene a α como raíz.

3) Se sigue de 2) del Lema 1.34.

4) Se sigue del Teorema 1.36. ■

Definición 1.41 Sean $\mathbb{E}/\mathbb{F}$ una extensión de campo y $\alpha \in \mathbb{E}$ un elemento algebraico. Al polinomio mónico irreducible del Teorema 1.40 se le llama polinomio mínimo de α sobre $\mathbb{F}$.

Definición 1.42 Sean $\mathbb{E}/\mathbb{F}$ una extensión de campo y $f(x) \in \mathbb{F}[x]$. $\mathbb{E}$ es un campo de descomposición para $f(x)$ si se cumple que $f(x)$ es un producto de polinomios lineales sobre $\mathbb{E}$ y no lo es sobre ningún subcampo propio de $\mathbb{E}$.

Teorema 1.43 Sea $\mathbb{F}$ un campo. Entonces cada $f(x) \in \mathbb{F}[x]$ tiene un campo de descomposición.

Demostración. Primero, veamos que existe un campo $\mathbb{L}$ que contiene a $\mathbb{F}$ como subcampo y es tal que $f(x)$ se escribe como producto de polinomios lineales en $\mathbb{E}[x]$. Para ello hagamos inducción sobre $grad(f(x))$.
i) Si $grad(f(x)) = 1$, entonces $f(x)$ es lineal, así que basta tomar $\mathbb{E} = \mathbb{F}$.
ii) Supongamos que $grad(f(x)) > 1$. Sea $p(x) \in \mathbb{F}[x]$ irreducible tal que $f(x) = p(x)h(x)$ para algún $h(x) \in \mathbb{F}[x]$ (esto es posible por ser $\mathbb{F}[x]$ un dominio de factorización única). Por el Teorema 1.36, existe un campo $\mathbb{K}$ tal que $\mathbb{F} \subseteq \mathbb{K}$ y tiene una raíz a de $p(x)$. Entonces $p(x) = (x-a)g(x)$ para algún $g(x) \in \mathbb{K}[x]$, por consiguiente $f(x) = (x-a)g(x)h(x)$. Por hipótesis de inducción, existe un campo $\mathbb{L}$ que contiene a $\mathbb{K}$ y es tal que $g(x)h(x)$ se escribe como producto de polinomios lineales en $\mathbb{L}[x]$. Por lo tanto $\mathbb{L}$ contiene a $\mathbb{F}$ y $f(x)$ se escribe como producto de polinomios lineales en $\mathbb{L}[x]$. Ahora, sean $\alpha_1, .., \alpha_n$ las raíces de $f(x)$ en $\mathbb{L}$ y definamos $\mathbb{E} = \mathbb{L}(\alpha_1, ..., \alpha_n)$. Entonces $f(x)$ se escribe como producto de polinomios lineales sobre $\mathbb{E}$ y no lo hace sobre ningún subcampo propio, es decir, $\mathbb{E}$ es el campo de descomposición de $f(x) \in \mathbb{F}[x]$. ■

El Teorema anterior, garantiza la existencia de los campos de descomposición. En seguida se enuncia (sin demostración) un resultado de unicidad (cf. [13]).

Teorema 1.44 Cualesquiera dos campos de descomposición de $f(x) \in \mathbb{F}[x]$ son isomorfos.

1.5. CARACTERÍSTICA DE UN CAMPO

Definición 1.45 Sea $\mathbb{F}$ un campo. Al menor entero positivo n que cumple $\underbrace{1 + ... + 1}_{n-sumandos} = 0$ se le llama la característica de $\mathbb{F}$ y se denota por $Car(\mathbb{F})$. Si no existe tal entero positivo, se dice que $\mathbb{F}$ es de característica 0.

Lema 1.46 Sea $\mathbb{F}$ un campo tal que $Car(\mathbb{F}) \neq 0$. Entonces $Car(\mathbb{F})$ es un número primo.

Demostración. Para $n \in \mathbb{N}$ y $r \in \mathbb{F}$, nr denota la suma $\sum_{i=1}^{n} r \in \mathbb{F}$. Supongamos que $Car(\mathbb{F})$ no es un número primo. Entonces $Car(\mathbb{F}) = ts$ donde $1 < t, s < Car(\mathbb{F})$. Como $Car(\mathbb{F})a = 0$ para cada $a \in \mathbb{F}$, en particular para $a \neq 0$ se tiene que $0 = Car(\mathbb{F})a = Car(\mathbb{F})(1_{\mathbb{F}}a) = (Car(\mathbb{F})1_{\mathbb{F}})a =$

$((ts)1_{\mathbb{F}})a = (t1_{\mathbb{F}})(s1_{\mathbb{F}})a$, con $t1_{\mathbb{F}}, s1_{\mathbb{F}} \in \mathbb{F}$ y $a \neq 0$. Como $\mathbb{F}$ es un dominio entero, se sigue que $0 = (t1_{\mathbb{F}})(s1_{\mathbb{F}})$, en consecuencia $t1_{\mathbb{F}} = 0$ ó $s1_{\mathbb{F}} = 0$. En cualquier caso, se contradice que $Car(\mathbb{F})$ es el menor entero positivo tal que $Car(\mathbb{F})1_{\mathbb{F}} = 0$. Por lo tanto, $Car(\mathbb{F})$ es un número primo. ■

A continuación se enuncia un resultado bastante conocido sobre potencias de binomios en campos de característica distinta de 0, tal demostración se puede consultar en [9].

Proposición 1.47 Sea $\mathbb{F}$ un campo de característica p, donde p es un número primo. Entonces

$$\forall n \in \mathbb{N},\ \forall a, b \in \mathbb{F} : (a+b)^{p^n} = a^{p^n} + b^{p^n}.$$

Definición 1.48 Dado un campo $\mathbb{F}$, definimos su campo primo como la intersección de todos sus subcampos.

El siguiente resultado, cuya demostración puede consultarse en [13], caracteriza al subcampo primo de un campo $\mathbb{F}$.

Teorema 1.49 Si $\mathbb{F}$ es un campo, entonces su campo primo es isomorfo a $\mathbb{Q}$ ó es isomorfo a $\mathbb{Z}_p$ para algún número primo p.

Observación. La característica de un campo puede ser definida de manera equivalente a la luz del Teorema 1.49. Si el campo primo es isomorfo a $\mathbb{Q}$ entonces el campo es de característica 0, en otro caso, si el campo primo es isomorfo a $\mathbb{Z}_p$, entonces el campo tiene característica p.

1.6. CAMPOS FINITOS

Lema 1.50 Sea $\mathbb{F}$ un campo finito ($|\mathbb{F}| < \infty$) que contiene a un subcampo $\mathbb{K}$ tal que $|\mathbb{K}| = q$. Entonces $\mathbb{F}$ tiene q^m elementos, donde $m = [\mathbb{F} : \mathbb{K}]$.

Demostración. $\mathbb{F}$ es un espacio vectorial sobre $\mathbb{K}$, como $\mathbb{F}$ es finito, entonces es finitamente dimensional sobre $\mathbb{K}$. Si $[\mathbb{F} : \mathbb{K}] = m$, entonces $\mathbb{F}$ tiene una base sobre $\mathbb{K}$ que consiste de m-elementos, digamos $b_1, ..., b_m$. Se sigue que todo elemento de $\mathbb{F}$ puede ser representado de manera única en la forma $a_1b_1 + a_2b_2 + ... + a_mb_m$, con $a_1, ..., a_m \in \mathbb{K}$. Ya que cada a_i tiene q posibles valores, se sigue que $|\mathbb{F}| = q^m$. ■

Observación. 1) Un campo finito tiene característica p con p un número primo, pues por ser finito no podría ser de característica 0. Por la observación posterior al Teorema 1.49, su campo primo es isomorfo a $\mathbb{Z}_p$.
2) Si $\mathbb{F}$ es un campo finito con p elementos con p un número primo, entonces $\mathbb{F}$ coincide con su campo primo y es isomorfo a $\mathbb{Z}_p$. Se sigue que hay un único campo (salvo isomorfismo) con p elementos, el cual denotamos por $\mathbb{F}_p$.

Teorema 1.51 Sea $\mathbb{F}$ un campo finito. Entonces $\mathbb{F}$ tiene p^n elementos, donde el número p es la característica de $\mathbb{F}$ y n es el grado de $\mathbb{F}$ sobre su campo primo.

Demostración. Por la observación inmediata anterior a este teorema, $\mathbb{F}$ tiene característica un número primo p y su campo primo es isomorfo a $\mathbb{Z}_p$. Se sigue que el campo primo de $\mathbb{F}$ tiene p elementos. El resto se sigue del Lema 1.50. ■

Lema 1.52 Si $\mathbb{F}$ es un campo finito con q elementos, entonces para cada $a \in \mathbb{F}$ se satisface que $a^q = a$.

Demostración. La relación $a^q = a$ es inmediata para $a = 0$. Por otro lado, los elementos distintos de cero de $\mathbb{F}$ forman un grupo de orden $q-1$ bajo el producto. Así, $a^{q-1} = 1$ para cada $a \in \mathbb{F}$ con $a \neq 0$; multiplicando por a obtenemos el resultado deseado. ■

Lema 1.53 Si $\mathbb{F}$ es un campo finito con q elementos y $\mathbb{K}$ es un subcampo de $\mathbb{F}$, entonces el polinomio $x^q - x \in \mathbb{K}[x]$ se factoriza en $\mathbb{F}[x]$ como

$$x^q - x = \prod_{a \in \mathbb{F}} (x - a)$$

y $\mathbb{F}$ es un campo de descomposición de $x^q - x$ sobre $\mathbb{K}$.

Demostración. El polinomio $x^q - x$ de grado q tiene a lo más q raíces en $\mathbb{F}$. Por el Lema 1.52 conocemos q de tales raíces, a saber todos los elementos $\mathbb{F}$. Así, el polinomio dado se escribe como producto de polinomios lineales en $\mathbb{F}[x]$ tal como lo indica el enunciado del lema, y es claro que no lo hace respecto a cualquier subcampo propio de $\mathbb{F}$ (pues tal campo propio no tiene las q raíces del polinomio). ■

Antes de proseguir, se enuncia la definición de derivada para un polinomio en $\mathbb{F}[x]$ además de un resultado importante que involucra este concepto.

Definición 1.54 Sea $f(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1} + a_n x^n \in \mathbb{F}[x]$, entonces la derivada formal de $f(x)$, denotada como $f'(x)$, es el polinomio $f'(x) = a_1 + 2x + ...ia_i x^{i-1} + ... + (n-1)a_{n-1}x^{n-2} + na_n x^n \in \mathbb{F}[x]$.

El siguiente lema exhibe la importancia de la derivada formal de un polinomio $f(x) \in \mathbb{F}[x]$ respecto a raíces múltiples. La demostración respectiva se puede consultar en [13].

Lema 1.55 El polinomio $f(x) \in \mathbb{F}[x]$ no tiene raíces múltiples si y sólo si $mcd(f(x), f'(x)) = 1$.

Teorema 1.56 (Existencia y unicidad de campos finitos) Para cada número primo p y todo entero positivo n existe un campo finito con p^n elementos. Todo campo finito con $q = p^n$ elementos es isomorfo al campo de descomposición de $x^q - x$ sobre $\mathbb{F}_p$.

Demostración. (Existencia) Para $q = p^n$, considere $x^q - x \in \mathbb{F}_p$ y sea $\mathbb{F}$ su campo de descomposición sobre $\mathbb{F}_p$. Este polinomio tiene q distintas raíces en $\mathbb{F}$ ya que su derivada es $qx^{q-1} - 1 = -1$ en $\mathbb{F}_p$ por lo que no puede tener raíces en común con $x^q - x$ (Lema 1.55). Sea $S = \{a \in \mathbb{F} : a^q - a = 0\}$. Entonces S es un subcampo de $\mathbb{F}$ ya que : i) $0, 1 \in S$; ii) si $a, b \in S$, entonces de la Proposición 1.47, se sigue que $(a-b)^q = a^q - b^q = a - b$, de ahí que $a - b \in S$; iii) para $a, b \in S$ y $b \neq 0$ tenemos

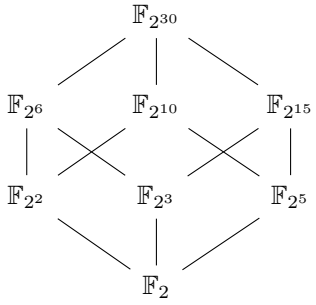
que $(ab^{-1})^q = a^q(b^{-1})^q = ab^{-1}$, y en consecuencia $ab^{-1} \in S$. Entonces $\mathbb{F} = S$, ya que S tiene q elementos. (Unicidad) Sea $\mathbb{F}$ un campo finito con $q = p^n$ elementos. Entonces por el Teorema 1.51, $\mathbb{F}$ tiene característica p, de ahí que contiene a $\mathbb{F}_p$ como un subcampo. Se sigue del Lema 1.53 que $\mathbb{F}$ es un campo de descomposición de $x^q - x$ sobre $\mathbb{F}_p$. Entonces el resultado deseado es una consecuencia de la unicidad (salvo isomorfismo) de los campos de descomposición. ■

Notación. Sea p un número primo y n un entero positivo. Al único campo (salvo isomorfismo) con $q = p^n$ elementos se denotará por $\mathbb{F}_q$.

Teorema 1.57 (Criterio del subcampo). Sea $\mathbb{F}_q$ el campo finito con $q = p^n$ elementos. Entonces cada subcampo de $\mathbb{F}_q$ tiene orden p^m, donde m es un divisor positivo de n. A la inversa, si m es un divisor positivo de n, entonces existe un único subcampo de $\mathbb{F}_q$ con p^m elementos.

Demostración. Sea $\mathbb{K}$ un subcampo de $\mathbb{F}_q$. Entonces $\mathbb{K}$ contiene al campo primo de $\mathbb{F}_q$ el cual tiene p elementos. Se sigue del Lema 1.50 que $|\mathbb{K}| = p^m$ con $m \leq n$. También del Lema 1.50, se sigue que $q = p^n$ debe ser una potencia de p^m, es decir, $p^n = (p^m)^s = p^{ms}$ para algún entero positivo s, de ahí que $n = ms$, lo cual implica que $m|n$. A la inversa, si m es un divisor positivo de n, entonces existe un entero positivo r tal que $n = rm$. Entonces $p^n - 1 = ((p^m)^r - 1) = (p^m - 1)(1 + p^m + (p^m)^2 + ... + (p^m)^{r-1})$, es decir, $p^m - 1$ divide $p^n - 1$ y entonces existe un entero positivo t tal que $p^n - 1 = t(p^m - 1)$, así $x^{p^n-1} - 1 = x^{t(p^m-1)} - 1 = (x^{p^m-1} - 1)(1 + x^{p^m-1} + ... + (x^{p^m-1})^{t-1})$, es decir, $x^{p^m-1} - 1$ divide a $x^{p^n-1} - 1$ en $\mathbb{F}_p[x]$. En consecuencia, $x^{p^m} - x | x^{p^n} - x$, es decir, $x^{p^m} - x | x^q - x$ en $\mathbb{F}_p[x]$. Entonces, cada raíz de $x^{p^m} - x$ es una raíz de $x^q - x$, por lo que tal raíz pertenece a $\mathbb{F}_q$. Se sigue que $\mathbb{F}_q$ contiene un subcampo que es campo de descomposición de $x^{p^m} - x$ sobre $\mathbb{F}_p$, y como se vio en la prueba del Teorema 1.56, tal campo de descomposición tiene orden p^m. Si hay dos subcampos distintos de orden p^m en $\mathbb{F}_q$, su unión contendría más de p^m raíces de $x^{p^m} - x$ en $\mathbb{F}_q$, lo cual es una contradicción. ■

Ejemplo 1.58 Los subcampos del campo finito $\mathbb{F}_{2^{30}}$ pueden ser determinados listando a los divisores de 30. Las relaciones de contención entre los diversos subcampos se muestran en el siguiente diagrama:



Por el Teorema 1.57, las relaciones de contención son las relaciones de divisibilidad entre los divisores positivos de 30.

Teorema 1.59 El grupo multiplicativo $\mathbb{F}_q^*$ de los elementos distintos de cero de un campo finito $\mathbb{F}_q$ es cíclico (cf. [9]).

Definición 1.60 Un generador del grupo cíclico $\mathbb{F}_q^*$ es llamado elemento primitivo de $\mathbb{F}_q$.

Teorema 1.61 Sea $\mathbb{F}_q$ un campo finito y $\mathbb{F}_r$ una extensión finita. Entonces $\mathbb{F}_r$ es una extensión algebraica simple de $\mathbb{F}_q$ y todo elemento primitivo de $\mathbb{F}_r$ puede servir como un elemento que define a $\mathbb{F}_r$ sobre $\mathbb{F}_q$.

Demostración. Sea ξ un elemento primitivo de $\mathbb{F}_r$. Claramente tenemos $\mathbb{F}_q(\xi) \subseteq \mathbb{F}_r$. Por otro lado, $\mathbb{F}_q(\xi)$ contiene a 0 y a todas las potencias de ξ, y así a todos los elementos de $\mathbb{F}_r$. Por lo tanto, $\mathbb{F}_r = \mathbb{F}_q(\xi)$. ■

Corolario 1.62 Sean $\mathbb{F}_q$ un campo finito y n un entero positivo. Entonces existe un polinomio irreducible en $\mathbb{F}_q[x]$ de grado n.

Demostración. Sea $\mathbb{F}_r$ una extensión de campo de $\mathbb{F}_q$ de grado n. Por el Teorema 1.61 tenemos que $\mathbb{F}_r = \mathbb{F}_q(\xi)$ para algún $\xi \in \mathbb{F}_r$. Se sigue que ξ es algebraico sobre $\mathbb{F}_q$ pues es raíz del polinomio $x^{\circ(\xi)} - 1 \in \mathbb{F}_q[x]$, donde $\circ(\xi)$ es el orden del elemento ξ en el el grupo multiplicativo de los elementos distintos de cero de $\mathbb{F}_r$. Entonces el polinomio mínimo de ξ sobre $\mathbb{F}_q$ es un polinomio irreducible en $\mathbb{F}_q[x]$, de acuerdo al Teorema 1.40. ■

El siguiente teorema acerca del grado de extensiones de campo es necesario para resultados posteriores.

Teorema 1.63 Si $\mathbb{F} \subseteq \mathbb{B} \subseteq \mathbb{E}$ son campos tales que $\mathbb{E}/\mathbb{B}$ y $\mathbb{B}/\mathbb{F}$ son extensiones finitas, entonces $\mathbb{E}/\mathbb{F}$ es finita y

$$[\mathbb{E} : \mathbb{F}] = [\mathbb{E} : \mathbb{B}][\mathbb{B} : \mathbb{F}].$$

Demostración. Sea $\{\alpha_1, ..., \alpha_m\}$ una base de $\mathbb{E}$ como $\mathbb{B}$-espacio vectorial y $\{\beta_1, ..., \beta_n\}$ una base de $\mathbb{B}$ como $\mathbb{F}$-espacio vectorial. Es suficiente probar que $S = \{\beta_j \alpha_i : 1 \leq j \leq m, 1 \leq i \leq n\}$ es una base de $\mathbb{E}$ como $\mathbb{F}$-espacio vectorial.
Si $x \in \mathbb{E}$, entonces existen $b_1, ..., b_m \in \mathbb{B}$ tales que $x = \sum_{i=1}^{m} b_i \alpha_i$. Pero $b_i = \sum_{j=1}^{n} c_{i_j} \beta_j$ para $c_{i_j} \in \mathbb{F}$, de ahí que $x = \sum_{i=1}^{m} (\sum_{j=1}^{n} c_{i_j} \beta_j) \alpha_i = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i_j} \beta_j \alpha_i \in \langle S \rangle$. Por lo tanto, S genera a $\mathbb{E}$ como $\mathbb{F}$-espacio vectorial.
Supongamos que $\sum_{i=1}^{m} \sum_{j=1}^{n} r_{i_j} \beta_j \alpha_i = 0$, con $r_{i_j} \in \mathbb{F}$. Entonces $\sum_{i=1}^{m} (\sum_{j=1}^{n} r_{i_j} \beta_j) \alpha_i$, y de la independencia lineal de los α_i sobre $\mathbb{B}$ se sigue que $\sum_{j=1}^{n} r_{i_j} \beta_j = 0$. Como los β_j son linealmente independientes sobre $\mathbb{F}$, concluimos que los c_{i_j} son 0. Por lo tanto S es linealmente independiente sobre $\mathbb{F}$; por consiguiente, S es una base de $\mathbb{E}$ como espacio vectorial sobre $\mathbb{F}$. ■

Teorema 1.64 Sea $f(x) \in \mathbb{F}_q[x]$ un polinomio irreducible sobre $\mathbb{F}_q$ de grado m. Entonces $f(x)$ divide a $x^{q^n} - x$ si y sólo si m divide a n.

Demostración. Supongamos que $f(x)$ divide $x^{q^n} - x$ en $\mathbb{F}_q[x]$. Sea α una raíz de $f(x)$ en su campo de descomposición sobre $\mathbb{F}_q$. Entonces $\alpha^{q^n} = \alpha$, de ahí que $\alpha \in \mathbb{F}_{q^n}$ (ya que $\mathbb{F}_{q^n}$ es el campo de descomposición de $x^{q^n} - x$ sobre $\mathbb{F}_q$). Se sigue que $\mathbb{F}_q(\alpha)$ es un subcampo de $\mathbb{F}_{q^n}$. Entonces, por el Teorema 1.40, $[\mathbb{F}_q(\alpha) : \mathbb{F}_q] = m$. Del Teorema 1.63, tenemos que m divide a n, ya que $n = [\mathbb{F}_{q^n} : \mathbb{F}_q] = [\mathbb{F}_{q^n} : \mathbb{F}_q(\alpha)][\mathbb{F}_q(\alpha) : \mathbb{F}_q] = [\mathbb{F}_{q^n} : \mathbb{F}_q(\alpha)]m$. A la inversa, si m divide a n, entonces por el Teorema 1.57, se tiene que $\mathbb{F}_{q^n}$ contiene a $\mathbb{F}_{q^m}$ como subcampo. Si α es una raíz de $f(x)$ en su campo de descomposición sobre $\mathbb{F}_q$, entonces $[\mathbb{F}_q(\alpha) : \mathbb{F}_q] = m$ (por el Teorema 1.40), es decir, $\mathbb{F}_q(\alpha) = \mathbb{F}_{q^m}$. Por consiguiente, $\alpha \in \mathbb{F}_{q^n}$ y se cumple que $\alpha^{q^n} = \alpha$, por lo que α es una raíz de $x^{q^n} - x \in \mathbb{F}_q[x]$. Si a_m es el coeficiente principal de $f(x)$, entonces α tiene a $a_m^{-1}f(x)$ como polinomio mínimo sobre $\mathbb{F}_q$. Ya que α es raíz de $x^{q^n} - x \in \mathbb{F}_q[x]$, se sigue que $a_m^{-1}f(x)$ divide a $x^{q^n} - x \in \mathbb{F}_q[x]$, lo cual implica que $f(x)$ divide a $x^{q^n} - x$. ■

Teorema 1.65 Si $f(x)$ es un polinomio irreducible en $\mathbb{F}_q[x]$ de grado m, entonces $f(x)$ tiene una raíz α en $\mathbb{F}_{q^m}$. Además, todas las raíces de $f(x)$ son simples y están dadas por los m distintos elementos $\alpha, \alpha^q, \alpha^{q^2}, ..., \alpha^{q^{m-1}}$ (llamados los conjugados de α) de $\mathbb{F}_{q^m}$. Además, m es el menor entero positivo para el cual $\alpha^{q^m} = \alpha$.

Demostración. Sea α una raíz de $f(x)$ en su campo de descomposición sobre $\mathbb{F}_q$. Entonces, $[\mathbb{F}_q(\alpha) : \mathbb{F}_q] = m$ lo cual implica que $\mathbb{F}_q(\alpha) = \mathbb{F}_{q^m}$, en particular $\alpha \in \mathbb{F}_{q^m}$. Ahora se demostrará que si $\beta \in \mathbb{F}_{q^m}$ es una raíz de $f(x)$, entonces β^q también es una raíz de $f(x)$. Sea $f(x) = \sum_{i=0}^{m} a_i x^i$, con $a_i \in \mathbb{F}_q$ para $0 \leq i \leq m$. Entonces, usando el Lema 1.52 y la Proposición 1.47, tenemos que $f(\beta^q) = \sum_{i=0}^{m} a_i \beta^{qi} = \sum_{i=0}^{m} a_i^q \beta^{qi} = \sum_{i=0}^{m} (a_i \beta^i)^q = (\sum_{i=0}^{m} a_i \beta^i)^q = f(\beta)^q = 0$. Por lo tanto, los elementos $\alpha, \alpha^q, \alpha^{q^2}, ..., \alpha^{q^{m-1}}$ son raíces de $f(x)$. Veamos que que estos elementos son distintos por pares. Supongamos lo contrario, es decir, supongamos que $\alpha^{q^j} = \alpha^{q^k}$ para algunos enteros j y k, con $0 \leq j < k \leq m-1$. Elevando esta identidad a la potencia q^{m-k}, obtenemos $\alpha^{q^{m-k+j}} = \alpha^{q^m} = \alpha$. Entonces α es raíz de $x^{q^{m-k+j}} - x$, por lo que $f(x)$ divide a $x^{q^{m-k+j}} - x$. Por el Teorema 1.64, esto es posible si y sólo si m divide a $m - k + j$. Pero tenemos que $0 < m - k + j < m$, entonces se ha llegado a una contradicción. Finalmente, si $m_1 < m$ y $\alpha^{q^{m_1}} = \alpha$, entonces $\alpha^{q^{m_1}} - \alpha = 0$. Se sigue que α es una raíz del polinomio $x^{q^{m_1}} - x$, lo cual implica que $f(x) | (x^{q^{m_1}} - x)$ y esto es equivalente (por el Teorema 1.64) a que $m | m_1$, lo que contradice que $m_1 < m$.

■

Corolario 1.66 Sea $f(x)$ un polinomio irreducible en $\mathbb{F}_q[x]$ de grado m. Entonces el campo de descomposición de $f(x)$ sobre $\mathbb{F}_q$ está dado por $\mathbb{F}_{q^m}$.

Demostración. El Teorema 1.65, muestra que $f(x)$ se escribe como un producto de polinomios lineales en $\mathbb{F}_{q^m}[x]$. Además, $\mathbb{F}_q(\alpha, \alpha^q, \alpha^{q^2}, ..., \alpha^{q^{m-1}}) = \mathbb{F}_q(\alpha) = \mathbb{F}_{q^m}$ para una raíz α de $f(x)$ en $\mathbb{F}_{q^m}$, donde la segunda identidad es tomada de la demostración del Teorema 1.65. ■

Nota. Sea $\mathbb{F}_q$ un campo finito. Del Teorema 1.65 y el Corolario 1.66, cabe destacar que un elemento primitivo del campo $\mathbb{F}_{q^m}$ se puede definir como la raíz de un polinomio mónico irreducible sobre $\mathbb{F}_q$ de grado m.

1.7. RAÍCES N-ÉSIMAS DE LA UNIDAD

Definición 1.67 Sea n un entero positivo. El campo de descomposición de $x^n - 1$ sobre un campo $\mathbb{K}$ es llamado el n-ésimo campo ciclotómico sobre $\mathbb{K}$ y es denotado por $\mathbb{K}^{(n)}$. Las raíces de $x^n - 1$ en $\mathbb{K}^{(n)}$ son llamadas las raíces n-ésimas de la unidad sobre $\mathbb{K}$ y el conjunto de todas esas raíces es denotada por $\mathbb{E}^{(n)}$.

Teorema 1.68 Sea n un entero positivo y $\mathbb{K} = \mathbb{F}_{p^m}$ un campo finito. Si p no divide a n, entonces $\mathbb{E}^{(n)}$ es un grupo cíclico de orden n con respecto a la multiplicación en $\mathbb{K}^{(n)}$.

Demostración. El caso $n = 1$ es inmediato. Para $n \geq 2$, $x^n - 1$ y su derivada nx^{n-1} no tienen raíces en común, ya que $nx^{n-1} \neq 0$ (pues p, la característica de $\mathbb{K}$, no divide a n) y sólo tiene a 0 como raíz en $\mathbb{K}^{(n)}$. Por el Lema 1.55, $x^n - 1$ no puede tener raíces múltiples, de ahí que $\mathbb{E}^{(n)}$ tiene n elementos. Sean $\xi, \eta \in \mathbb{E}^{(n)}$, entonces $(\xi\eta^{-1})^n = \xi^n(\eta^n)^{-1} = 1$, entonces $\xi\eta^{-1} \in \mathbb{E}^{(n)}$. Por lo tanto, $\mathbb{E}^{(n)}$ es un grupo multiplicativo. Sea $p_1(x)p_2(x) \cdot ... \cdot p_t(x)$ la factorización en irreducible de $x^n - 1$ en $\mathbb{K}[x]$ con $m_i = grad(p_i(x))$. Entonces $\mathbb{F}_{q^{m_i}}$ es el campo de descomposición de $p_i(x)$ sobre $\mathbb{F}_q$ (por el Corolario 1.66). Del criterio del subcampo, si $c = mcm\{m_1, ..., m_t\}$ se sigue que $\mathbb{F}_{q^c}$ contiene a cada $\mathbb{F}_{q^{m_i}}$, por lo que $\mathbb{K}^{(n)}$ está contenido en $\mathbb{F}_{q^c}$. Por lo tanto, $\mathbb{K}^{(n)}$ es un campo finito. Por el Teorema 1.59, el grupo formado por los elementos distintos de cero de $\mathbb{K}^{(n)}$ es un grupo cíclico, el cual contiene como subgrupo a $\mathbb{E}^{(n)}$. Se concluye que $\mathbb{E}^{(n)}$ es cíclico (subgrupos de grupos cíclicos son cíclicos). ■

Definición 1.69 Sean $\mathbb{K} = \mathbb{F}_{p^m}$ un campo finito y n un entero positivo que no es divisible por p. Entonces un generador del grupo cíclico $\mathbb{E}^{(n)}$ es llamado una raíz n-ésima primitiva de la unidad sobre $\mathbb{K}$.

1.8. CLASES CICLOTÓMICAS

Definición 1.70 Sean q, $n \in \mathbb{N}$ tales que $mcd(n, q) = 1$ e i$\in \{0, 1, ..., n-1\}$. Se define la i-ésima clase ciclotómica de q módulo n como el conjunto

$$C_q(i) := \{iq^j \pmod{n} | j \in \mathbb{N}_0\}.$$

Observaciones. Para cada $i \in \{0, 1, ..., n-1\}$ tenemos lo siguiente:

1) $C_q(i) \neq \emptyset$, ya que $i \in C_q(i)$.

2) Para $j \in \mathbb{N}_0$, $iq^j \pmod{n}$ se refiere al residuo de dividir al entero iq^j por n.

3) $C_q(i) \subseteq \{0, 1, ..., n-1\}$.

Proposición 1.71 Sea $h \in \{0, 1, ..., n-1\}$, entonces: $h \in C_q(i) \Leftrightarrow iq^{j_0} \equiv h \pmod{n}$ para algún $j_0 \in \mathbb{N}_0$.

Demostración. $\Rightarrow$] Supongamos que $h \in C_q(i)$, entonces h es el residuo de dividir para algún $j_0 \in \mathbb{N}_0$ a iq^{j_0} por n, es decir, existe $a \in \mathbb{Z}$ tal que $iq^{j_0} = an + h$ lo cual implica que $n|(iq^{j_0} - h)$, o bien, $iq^{j_0} \equiv h \pmod{n}$.

$\Leftarrow]$ Sea $j_0 \in \mathbb{N}_0$ tal que $iq^{j_0} \equiv h \pmod{n}$, entonces $n|(iq^{j_0} - h)$, por lo que existe $a \in \mathbb{Z}$ tal que $iq^{j_0} - h = an$, o de manera equivalente, $iq^{j_0} = an + h$. Como $h \in \{0, 1, ..., n-1\}$ entonces es el residuo de dividir a iq^{j_0} por n, lo que nos dice que $h \in C_q(i)$. ■

Lema 1.72. Sean $q, n \in \mathbb{N}$ primos relativos. Entonces el conjunto de las clases ciclotómicas de q módulo n es una partición de $\{0, 1, ..., n-1\}$.

Demostración. Veamos que se cumplen las siguientes tres condiciones:
i) Para cada $i \in \{0, 1, ..., n-1\}$, $C_q(i) \neq \emptyset$.

ii) $\bigcup_{i=0}^{n-1} C_q(i) = \{0, 1, ..., n-1\}$.

iii) Para $i, j \in \{0, 1..., n-1\}$ se tiene que: $C_q(i) \cap C_q(j) \neq \emptyset$ implica que $C_q(i) = C_q(j)$.

La condición $i)$ se cumple de acuerdo a la observación 1). También de la observación 1) se deduce que $\{0, 1, ..., n-1\} \subseteq \bigcup_{i=1}^{n-1} C_q(i)$, y de la observación 3) se sigue que $\bigcup_{i=1}^{n-1} C_q(i) \subseteq \{0, 1, ..., n-1\}$. Por lo tanto, $\bigcup_{i=1}^{n-1} C_q(i) = \{0, 1, ..., n-1\}$ y (ii) se satisface. Ahora, sean $i_1, i_2 \in \{0, 1, ..., n-1\}$ tales que $C_q(i_1) \cap C_q(i_2) \neq \emptyset$. Entonces, por la Proposición 1.71, existe $h \in \{0, 1, ..., n-1\}$ tal que $i_1 q^{j_1} \equiv h \pmod{n}$ e $i_2 q^{j_2} \equiv h \pmod{n}$ para algunos $j_1, j_2 \in \mathbb{N}_0$. Además, supongamos sin pérdida de generalidad que $j_1 \leq j_2$, por lo que existe $r \in \mathbb{N}_0$ tal que $j_1 + r = j_2$. Ya que la relación de congruencia módulo n es de equivalencia en $\mathbb{Z}$, entonces $i_2 q^{j_2} \equiv i_1 q^{j_1} \pmod{n}$, o equivalentemente, $n|(i_2 q^{j_2} - i_1 q^{j_1})$. Como $i_2 q^{j_2} - i_1 q^{j_1} = i_2 q^{j_1 + r} - i_1 q^{j_1} = q^{j_1}(i_2 q^r - i_1)$ y $n|(i_2 q^{j_2} - i_1 q^{j_1})$ con $(n, q^{j_1}) = 1$ (lo cual se sigue de la hipótesis $mcd(n, q) = 1$), entonces podemos implicar que $n|i_2 q^r - i_1$, o bien, $i_2 q^r \equiv i_1 \pmod{n}$. Demostremos que $C_q(i_1) = C_q(i_2)$.
$\subseteq]$ Sea $h_1 \in C_q(i_1)$, entonces existe $m_1 \in \mathbb{N}_0$ tal que $i_1 q^{m_1} \equiv h_1 \pmod{n}$. Ya que $i_2 q^r \equiv i_1 \pmod{n}$, entonces $i_2 q^{r+m_1} \equiv i_1 q^{m_1} \pmod{n}$; por consiguiente $i_2 q^{r+m_1} \equiv h_1 \pmod{n}$, y de acuerdo a la Proposición 1.71, tenemos que $h_1 \in C_q(i_2)$.

$\supseteq]$ Sea $h_2 \in C_q(i_2)$, entonces existe $m_2 \in \mathbb{N}_0$ tal que $i_2 q^{m_2} \equiv h_2 \pmod{n}$. Ya que $(n, q^r) = 1$ (se sigue de $mcd(q, n) = 1$), entonces de acuerdo al Teorema de Euler, se tiene que: $(q^r)^{\varphi(n)} \equiv 1 \pmod{n}$. De $i_2 q^{m_2} \equiv h_2 \pmod{n}$ y $q^{(\varphi(n)-1)r+r} \equiv 1 \pmod{n}$ se sigue que $i_2 q^{m_2} q^{(\varphi(n)-1)r+r} \equiv h_2 \pmod{n}$, o equivalentemente, $i_2 q^r q^{(\varphi(n)-1)r+m_2} \equiv h_2 \pmod{n}$. Ahora, de $i_2 q^r q^{(\varphi(n)-1)r+m_2} \equiv h_2 \pmod{n}$ e $i_2 q^r \equiv i_1 \pmod{n}$ podemos implicar que

$$i_1 q^{(\varphi(n)-1)r+m_2} \equiv h_2 \pmod{n},$$

así $h_2 \in C_q(i_1)$. Por lo tanto, $C_q(i_1) = C_q(i_2)$ y $iii)$ se satisface. Ya que se cumplen $i)$, $ii)$ y $iii)$, se concluye que el conjunto de las clases ciclotómicas de q módulo n es una partición de $\{0, 1, ..., n-1\}$. ■

Sean $q, n \in \mathbb{N}$ primos relativos e $i \in \{0, 1, .., n-1\}$. Ya que $\{0, 1, ..., n-1\}$ es un conjunto finito, entonces $C_q(i)$ también lo es, por lo que existen $h \in C_q(i)$ y $j_1, j_2 \in \mathbb{N}_0$, con $j_1 < j_2$, tales que

$$iq^{j_1} \equiv h \pmod{n} \text{ e } iq^{j_2} \equiv h \pmod{n}.$$

Entonces $iq^{j_2} \equiv iq^{j_1} \pmod{n}$, o equivalentemente, $n|(iq^{j_2} - iq^{j_1})$. Como $j_1 < j_2$, entonces existe $r \in \mathbb{N}$ tal que $j_1 + r = j_2$, por lo que $iq^{j_2} - iq^{j_1} = iq^{j_1+r} - iq^{j_1} = q^{j_1}(iq^r - i)$, donde $(n, q^{j_1}) = 1$. De $n|q^{j_1}(iq^r - i)$ podemos implicar que $n|(iq^r - i)$, lo cual equivale a $iq^r \equiv i \pmod{n}$. Sea $I_i = \{t \in \mathbb{N}|iq^t \equiv i \pmod{n}\} \subseteq \mathbb{N}$. De lo anterior, se sigue que $r \in I_i$, y por lo tanto, $I_i \neq \emptyset$. Por el principio del buen orden, I_i tiene elemento menor d.

Lema 1.73. Sean $q, n \in \mathbb{N}$ primos relativos e $i \in \{0, 1, ..., n-1\}$. Si d es el elemento menor de

$$I_i = \{t \in \mathbb{N}|iq^t \equiv i \pmod{n}\},$$

entonces para cada $m \in \mathbb{N}$: $iq^{md} \equiv i \pmod{n}$.

Demostración. (Por inducción sobre m).
Si $m = 1$, la proposición es cierta por hipótesis. Supongamos que para $m \in \mathbb{N}$ se cumple que $iq^{md} \equiv i \pmod{n}$. Nótese que $iq^{(m+1)d} = (iq^d)q^{md}$, donde $iq^d \equiv i \pmod{n}$. Entonces $(iq^d)q^{md} \equiv iq^{md} \pmod{n}$, y de la hipótesis de inducción podemos implicar que $(iq^d)q^{md} \equiv i \pmod{n}$, o bien, $iq^{(m+1)d} \equiv i \pmod{n}$. Por lo tanto, para cada $m \in \mathbb{N} : iq^{md} \equiv i \pmod{n}$. ■

Proposición 1.74. Sean $q, n \in \mathbb{N}$ primos relativos e $i \in \{0, 1, ..., n-1\}$. Entonces

$$C_q(i) = \{iq^j \pmod{n}|j \in \{0, 1, .., d-1\}\} \text{ y } |C_q(i)| = d,$$

donde d es el menor entero positivo tal que $iq^d \equiv i \pmod{n}$.

Demostración. Es claro que $\{iq^j \pmod{n}|j \in \{0, 1, ..., n-1\}\} \subseteq C_q(i)$. Sea $h \in C_q(i)$, entonces existe $j \in \mathbb{N}_0$ tal que $iq^j \equiv h \pmod{n}$. Por el algoritmo de la división, existen $a, b \in \mathbb{Z}$ tales que $j = ad + b$, con $0 \leq b < |d| = d$. Note que $a \in \mathbb{N}_0$, ya que si $a \leq -1$ entonces $j = ad + b \leq -d + b < 0$, lo cual es una contradicción. Entonces, por el Lema 1.73, $iq^{ad} \equiv i \pmod{n}$, por lo que $iq^j = iq^{ad+b} = iq^{ad}q^b \equiv iq^b \pmod{n}$, en consecuencia, $iq^b \equiv h \pmod{n}$, es decir, $h \in \{iq^j \pmod{n}|j \in \{0, 1, ..., d-1\}\}$. Por lo tanto $C_q(i) = \{iq^j \pmod{n}|j \in \{0, 1, ..., d-1\}\}$. Además, si $h \in C_q(i)$ es tal que para $j_1, j_2 \in \{0, 1, ..., d-1\}$, con $j_1 < j_2$, cumple que

$$iq^{j_1} \equiv h \pmod{n} \text{ e } iq^{j_2} \equiv h \pmod{n},$$

entonces podemos implicar que $iq^{j_2} \equiv iq^{j_1} \pmod{n}$, por lo que $n|q^{j_1}(iq^{j_2-j_1} - i)$ donde $(n, q^{j_1}) = 1$. De ahí que $n|iq^{j_2-j_1} - i$, lo cual equivale a $iq^{j_2-j_1} \equiv i \pmod{n}$ con $0 \leq j_2 - j_1 < d$, lo cual es un contradicción. Por lo tanto $|C_q(i)| = d$. ■

Ejemplo 1.75 Sean $n = 7$ y $q = 2$. Consideremos d como en la Proposición 1.74, entonces las distintas clases ciclotómicas de q módulo n son:

$C_q(0) = \{0\}$, donde $d = 1 = |C_q(0)|$.

$C_q(1) = \{2^j \pmod{7}|j \in \{0, 1, 2\}\} = \{1, 2, 4\}$, donde $d = 3 = |C_q(1)|$.

$C_q(3) = \{3(2^j) \pmod{7}|j \in \{0, 1, 2\}\} = \{3, 5, 6\}$, donde $d = 3 = |C_q(3)|$.

Además, es fácil ver que $C_q(1) = C_q(2) = C_q(4)$ y $C_q(3) = C_q(5) = C_q(6)$.

Observamos que $\{C_q(0), C_q(1), C_q(3)\}$ es una partición de $\{0, 1, ..., 6\}$, como se afirmaba en el Lema 1.72.

1.8.1. FACTORIZACIÓN DE $x^n - 1$ EN TÉRMINOS DE CLASES CICLOTÓMICAS.

Sean q una potencia de un primo, n un entero positivo primo relativo a q y β un elemento primitivo de $\mathbb{F}_{q^s}$, donde s es el menor entero positivo para el cual $q^s \equiv 1 \pmod{n}$. Entonces $\frac{q^s-1}{n} \in \mathbb{Z}$ y

$$\omega = \beta^{\frac{q^s-1}{n}}.$$

es una raíz n-ésima primitiva de la unidad. Para ver esto, obsérvese que $(\beta^{\frac{q^s-1}{n}})^n = \beta^{q^s-1} = 1$, es decir, ω es raíz n-ésima de la unidad. Si $0 < r < n$ es tal que $\omega^r = 1$ entonces $\beta^{\frac{(q^s-1)r}{n}} = 1$ lo cual implica que $(q^s-1)|\frac{(q^s-1)r}{n}$ (ya que β es de orden q^s-1), entonces existe $l \in \mathbb{Z}$ tal que $\frac{(q^s-1)r}{n} = (q^s-1)l$. Como q es una potencia de un primo y s es un entero positivo, se sigue que $q^s - 1 \neq 0$. Entonces de $\frac{(q^s-1)r}{n} = (q^s-1)l$ se tiene que $\frac{r}{n} = l \in \mathbb{Z}$ lo cual contradice que $0 < r < n$. Por lo tanto ω tiene orden n, es decir, es una raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$.

Las raíces de $x^n - 1$ son $1, \omega, \omega^2, ..., \omega^{n-1}$. Para $i = 0, ..., n-1$ los conjugados de w^i son

$$\omega^i, \omega^{iq}, \omega^{iq^2}, ..., \omega^{iq^{d-1}} \tag{1.1}$$

donde d es el menor entero positivo para el cual $\omega^{iq^d} = \omega^i$. Por lo tanto, el polinomio mínimo para las raíces en (1.1) es

$$m_i(x) = (x - \omega^i)(x - \omega^{iq}) \cdot ... \cdot (x - \omega^{iq^{d-1}}).$$

El conjunto de exponentes en (1.1) es $\{i, iq, ..., iq^{d-1}\}$ donde d es el menor entero positivo tal que $iq^d \equiv 1 \pmod{n}$. A los elementos del anterior conjunto los podemos considerar reducción módulo n, pues por el algoritmo de la división, existen $k, n \in \mathbb{Z}$ tales que $iq^j = kn + r$ con $0 \leq r < n$. Así, $\omega^{iq^j} = \omega^{kn+r} = \omega^{kn}\omega^r = \omega^r$. Por lo que de manera más precisa, al conjunto de exponentes en (1.1) lo podemos tomar como la clase ciclotómica $C_q(i)$ que contiene a i.

Dicho lo anterior, considerando a los exponentes en (1.1) reducción módulo n, se tiene que

$$m_i(x) = \prod_{i \in C_q(i)} (x - \omega^i).$$

Obsérvese que de la misma manera, a partir de cada clase ciclotómica podemos contruir los polinomios mínimos de las raíces de $x^n - 1$.

Finalmente, como $x^n - 1$ es el producto de los distintos polinomios mínimos de sus raíces, entonces si Ω es un conjunto mínimo de representantes de las distintas clases ciclotómicas de q módulo n, se tiene que:

$$x^n - 1 = \prod_{i \in \Omega} m_i(x) = \prod_{j \in \bigcup\limits_{i \in \Omega} C_q(i)} (x - \omega^j).$$

Capítulo 2

CÓDIGOS

La teoría de códigos detectores-correctores de errores es una área activa dentro de la matemática desde hace varias décadas, cuyo principal fin es buscar resolver el problema de detectar y corregir errores en la transmisión de información a partir de datos recibidos.

Si suponemos que un mensaje $\mathbf{u}$ se codifica en una palabra-código $\mathbf{x}$ y se transmite por un canal y recibimos al vector $\mathbf{y}$, entonces $\mathbf{y}$ puede ser diferente de $\mathbf{x}$ (debido al ruido del canal), es decir, $\mathbf{y} = \mathbf{x} + \mathbf{e}$, donde $\mathbf{e}$ es un vector error. El problema consiste en determinar $\mathbf{e}$ y así obtener un estimado de $\mathbf{x}$ y en consecuencia del mensaje transmitido.

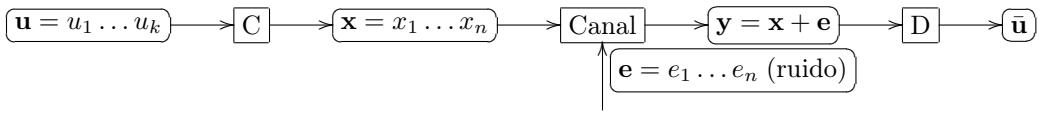


Figura 2.1: Sistema general de transmisión de información

2.1. CÓDIGOS LINEALES

Definición 2.1 $\mathcal{C}$ es un código de longitud n sobre $\mathbb{F}_q$ si $\mathcal{C} \subseteq \mathbb{F}_q^n$. Un $(n, M)-$código $\mathcal{C}$ sobre $\mathbb{F}_q$ es un código de longitud n tal que $|\mathcal{C}| = M$. A un elemento (arbitrario) de un código $\mathcal{C}$ se le llama palabra-código.

Definición 2.2 Sean $\mathbf{x} = (x_1, ..., x_n), \mathbf{y} = (y_1, ..., y_n) \in \mathbb{F}_q^n$. La distancia de Hamming de $\mathbf{x}$ a $\mathbf{y}$, lo cual denotamos $d(\mathbf{x}, \mathbf{y})$, se define como

$$d(\mathbf{x}, \mathbf{y}) = |\{i : 1 \leq i \leq n, x_i \neq y_i\}|.$$

Definición 2.3 El peso de $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{F}_q^n$, lo cual denotamos por $wt(\mathbf{x})$, se define como

$$wt(\mathbf{x}) = |\{i : 1 \leq i \leq n, x_i \neq 0\}|.$$

Observación 2.4 (a) De las Definiciones 2.1 y 2.2 se tiene en la Definición 2.3 que $wt(\mathbf{x}) = d(\mathbf{x}, \mathbf{0})$, donde $\mathbf{0}$ es el vector de $\mathbb{F}_q^n$ con todas sus componentes iguales a cero.
(b) Sean $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$. Por la Definición 2.1 y la Definición 2.2 se sigue que $d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x} - \mathbf{y})$, ya que si $s = d(\mathbf{x}, \mathbf{y})$, entonces hay s componentes en las que $\mathbf{x}$ e $\mathbf{y}$ difieren. En consecuencia hay $n - s$ componentes en las que $\mathbf{x}$ e $\mathbf{y}$ coinciden, de ahí que en $\mathbf{x} - \mathbf{y}$ hay $n - s$ componentes igaules a cero y s componentes distintas de cero, es decir, $wt(\mathbf{x} - \mathbf{y}) = s$.

Proposición 2.5 Para $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$ se tiene que $wt(x) + wt(y) \geq wt(x + y)$.

Demostración. Sean $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, $Sop(\mathbf{x}) = \{k : 1 \leq k \leq n \ \wedge \ x_k \neq 0\}$, $Sop(\mathbf{y}) = \{k : 1 \leq k \leq n \ \wedge \ x_k \neq 0\}$ y $Sop(\mathbf{x}+\mathbf{y}) = \{k : 1 \leq k \leq n \ \wedge \ x_k + y_k \neq 0\}$. Entonces, si $j \in Sop(\mathbf{x}+\mathbf{y})$ se sigue que $x_j + y_j \neq 0$. Lo anterior implica que $x_j \neq 0 \vee y_j \neq 0$, es decir, $j \in Sop(\mathbf{x}) \cup Sop(\mathbf{y})$. Por lo tanto

$$wt(\mathbf{x} + \mathbf{y}) = |Sop(\mathbf{x}+\mathbf{y})| \leq |Sop(\mathbf{x}) \cup Sop(\mathbf{y})| \leq |Sop(\mathbf{x})| + |Sop(\mathbf{y})| = wt(\mathbf{x}) + wt(\mathbf{y}).$$

■

Proposición 2.6 Sean $\mathbf{x} = (x_1, ..., x_n), \mathbf{y} = (y_1, ..., y_n), \mathbf{z} = (z_1, ..., z_n) \in \mathbb{F}_q^n$. Entonces se tiene lo siguiente:
(i) $0 \leq d(\mathbf{x}, \mathbf{y}) \leq n$;
(ii) $d(\mathbf{x}, \mathbf{y}) = 0$ si y sólo si $x = y$;
(iii) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$;
(iv) $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ (desigualdad del triángulo).

Demostración. (i), (ii) y (iii) se siguen inmediatamente de la Definición 2.2. Para demostrar (iv) consideremos $j \in \{i : 1 \leq i \leq n, x_i \neq z_i\}$; entonces, $x_j \neq z_j$lo cual implica que $x_j \neq y_j$ ó $y_j \neq z_j$, pues en caso contrario, si $x_j = y_j$ e $y_j = z_j$ implica que $x_j = z_j$ lo cual contradice la elección de j. Por lo tanto $j \in \{i : 1 \leq i \leq n, x_i \neq y_i\} \cup \{i : 1 \leq i \leq n, y_i \neq z_i\}$, y en consecuencia $\{i : 1 \leq i \leq n, x_i \neq z_i\} \subseteq \{i : 1 \leq i \leq n, x_i \neq y_i\} \cup \{i : 1 \leq i \leq n, y_i \neq z_i\}$. De la anterior contención y usando resultados conocidos sobre la cardinalidad de conjuntos se tiene lo siguiente:

$$\begin{aligned} d(\mathbf{x}, \mathbf{z}) &= |\{i : 1 \leq i \leq n, x_i \neq z_i\}| \\ &\leq |\{i : 1 \leq i \leq n, x_i \neq y_i\} \cup \{i : 1 \leq i \leq n, y_i \neq z_i\}| \\ &\leq \{i : 1 \leq i \leq n, x_i \neq y_i\}| + |\{i : 1 \leq i \leq n, y_i \neq z_i\}| \\ &= d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}). \end{aligned}$$

■

Ahora se dará paso a la definición de la distancia mínima, que es un parámetro importante de un código $\mathcal{C}$.

Definición 2.7 La distancia mínima de Hamming (ó distancia mínima) de un código $\mathcal{C}$, lo cual se denota por d, se define como

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

Definición 2.8 Un código lineal de longitud n sobre $\mathbb{F}_q$ es un subespacio vectorial de $\mathbb{F}_q^n$.

Dado un código lineal $\mathcal{C} \subseteq \mathbb{F}_q^n$, podemos definir una relación de equivalencia en $\mathbb{F}_q^n$ de la siguiente manera: $\mathbf{x} \sim \mathbf{y}$ si y sólo si $\mathbf{x} - \mathbf{y} \in \mathcal{C}$. Tal relación induce a una partición de $\mathbb{F}_q^n$ en las clases de equivalencia. Si $\mathbf{x} + \mathcal{C}$ es una clase de equivalencia de esta relación, entonces se dice que $\mathbf{c} \in \mathbf{x} + \mathcal{C}$ es líder de tal clase si $\mathbf{c}$ es una palabra de peso menor en $\mathbf{x} + \mathcal{C}$. El siguiente lema será de utilidad en el Capítulo 4, sin embargo, su enunciado y demostración son posibles en esta parte del capítulo actual.

Lema 2.9 Sea $\mathcal{C} \subseteq \mathbb{F}_q^n$ un código lineal con distancia mínima d. Entonces, $\mathbf{x} \in \mathbb{F}_q^n$ es el único líder de la clase $\mathbf{x} + \mathcal{C}$ (inducida por la relación de equivalencia $\mathbf{x} \sim \mathbf{y}$ si y sólo si $\mathbf{x} - \mathbf{y} \in \mathcal{C}$) siempre que $wt(x) \leq \lfloor \frac{d-1}{2} \rfloor$

Demostración. Sea $\mathbf{y} \in \mathbf{x} + \mathcal{C}$ con $\mathbf{y} \neq \mathbf{x}$. Entonces, existe $\mathbf{c} \in \mathcal{C} - \{\mathbf{0}\}$ tal que $\mathbf{y} = \mathbf{x} + \mathbf{c}$. Así, por la cualidad de d y la Proposición 2.5, se tiene lo siguiente:

$$d \leq wt(\mathbf{c}) = wt(\mathbf{y} - \mathbf{x}) \leq wt(\mathbf{y}) + wt(-\mathbf{x}) = wt(\mathbf{y}) + wt(\mathbf{x}),$$

lo cual implica que

$$wt(\mathbf{x}) \leq \lfloor \tfrac{d-1}{2} \rfloor \leq \tfrac{d-1}{2} < \tfrac{d+1}{2} = d - (\tfrac{d-1}{2}) \leq d - \lfloor \tfrac{d-1}{2} \rfloor \leq d - wt(\mathbf{x}) \leq wt(\mathbf{y}).$$

Entonces, $wt(\mathbf{x}) < wt(\mathbf{y})$, y por lo tanto $\mathbf{x}$ es el único líder de la clase $\mathbf{x} + \mathcal{C}$. ■

Ejemplo 2.10 Los siguientes son códigos lineales:

(i) $\mathcal{C} = \{000, 001, 010, 011\}$ sobre $\mathbb{F}_2$.

(ii) $\mathcal{C} = \{0000, 1100, 2200, 0001, 0002, 1101, 1102, 2201, 2202\}$ sobre $\mathbb{F}_3$.

Definición 2.11 Sea $\mathcal{C}$ un código lineal de longitud n sobre $\mathbb{F}_q$.

(i) El código dual de $\mathcal{C}$ es $\mathcal{C}^\perp$, el complemento ortogonal del subespacio $\mathcal{C}$ de $\mathbb{F}_q^n$.

(ii) La dimensión del código lineal $\mathcal{C}$ es la dimensión de $\mathcal{C}$ como un subespacio vectorial de $\mathbb{F}_q^n$ y se denota por $dim(\mathcal{C})$.

Observación 2.12 De resultados conocidos de álgebra lineal, se tiene lo siguiente para un código lineal $\mathcal{C}$ de longitud n sobre $\mathbb{F}_q$:

(i) $|\mathcal{C}| = q^{dim(\mathcal{C})}$;

(ii) $\mathcal{C}^\perp$ es un código lineal y $dim(\mathcal{C}) + dim(\mathcal{C}^\perp) = n$;

(iii) $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.

En el siguiente teorema se tiene una forma de ver la distancia mínima para códigos lineales, la cual es bastante útil.

Teorema 2.13 La distancia mínima de un código lineal $\mathcal{C}$ es el mínimo de los pesos de las palabras-código distintas de cero, es decir, d cumple la siguiente igualdad:

$$d = \text{mín}\{wt(\mathbf{c}) : \mathbf{c} \in \mathcal{C}, \mathbf{c} \neq 0\}$$

Demostración. De la definición de distancia mínima y la Observación 2.4 (b), se sigue que $d = \text{mín}\{d(\mathbf{u}, \mathbf{v}) : \mathbf{u}, \mathbf{v} \in \mathcal{C}, \mathbf{u} \neq \mathbf{v}\} = \text{mín}\{wt(\mathbf{u} - \mathbf{v}) : \mathbf{u}, \mathbf{v} \in \mathcal{C}, \mathbf{u} \neq \mathbf{v}\}$. Si $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ y $\mathbf{v} \neq \mathbf{u}$, entonces $\mathbf{c} := \mathbf{u} - \mathbf{v} \in \mathcal{C}$ (ya que $\mathcal{C}$ es subespacio de $\mathbb{F}_q^n$) y $\mathbf{c} \neq \mathbf{0}$, en consecuencia, $d = \text{mín}\{wt(\mathbf{c}) : \mathbf{c} \in \mathcal{C}, \mathbf{c} \neq \mathbf{0}\}$. ■

Observación 2.14 Un código lineal $\mathcal{C}$ de longitud n y dimensión k sobre $\mathbb{F}_q$ es llamado un $[n, k]$-código lineal q-ario. Si la distancia mínima d de $\mathcal{C}$ es conocida, algunas veces nos referimos a $\mathcal{C}$ como un $[n, k, d]$-código lineal.

El siguiente resultado nos da una importante cota para la distancia mínima de un código lineal, mismo que será de utilidad en algunos resultados posteriores.

Teorema 2.15 (La cota de Singleton) Si $\mathcal{C}$ es un $[n, k, d]$-código lineal sobre $\mathbb{F}_q$, entonces $d \leq n - k + 1$.

Demostración. Consideremos la función

$$T : \mathcal{C} \longrightarrow \mathbb{F}_q^{k-1}$$
$$(c_0, c_1, ..., c_{n-1}) \longmapsto (c_0, c_1, ..., c_{k-2}).$$

Entonces, T es una $\mathbb{F}_q$-transformación lineal. Ahora, $dim(T(\mathcal{C})) \leq k - 1$ (pues $dim(\mathbb{F}_q^{k-1}) = k - 1$ y $T(\mathcal{C}) \leq \mathbb{F}_q^{k-1}$). Por hipótesis $dim(\mathcal{C}) = k$, y del Teorema de la Dimensión se tiene que $dim(Ker(T)) + dim(T(\mathcal{C})) = dim(\mathcal{C}) = k$, de ahí que $1 \leq dim(Ker(T))$. Entonces, $Ker(T) \neq \{0\}$, y en consecuencia existe una palabra-código distinta de cero en $c \in \mathcal{C}$ tal que $T(c) = 0$, es decir (al menos) las primeras $k - 1$ componentes de c son iguales a 0, por lo que el peso de c depende de las $n - (k - 1)$ componentes restantes, así que $wt(c) \leq n - (k - 1) = n - k + 1$. Al encontrar una palabra-código de $\mathcal{C}$ distinta de cero de peso menor o igual a $n-k+1$, se sigue que $d \leq n-k+1$. ■

2.2. MATRIZ GENERADORA Y DE CHEQUEO DE PARIDAD PARA UN CÓDIGO LINEAL

Definición 2.16 (i) Una matriz generadora para un código lineal $\mathcal{C}$ es una matriz G tal que sus filas forman una base para $\mathcal{C}$.

(ii) Una matriz de chequeo de paridad H para un código lineal $\mathcal{C}$ es una matriz generadora para el código dual $\mathcal{C}^{\perp}$

Observación 2.17 Sea $\mathcal{C}$ un $[n, k]$-código lineal sobre $\mathbb{F}_q$.

(i) Una matriz generadora G de $\mathcal{C}$ cumple que $G \in \mathcal{M}_{k \text{ x } n}(\mathbb{F}_q)$ y si H es una matriz de chequeo de paridad se tiene que $H \in \mathcal{M}_{(n-k) \text{ x } n}(\mathbb{F}_q)$ $(dim(\mathcal{C}^{\perp}) = n - k)$.

(ii) Las filas de una matriz generadora son linealmente independientes. Lo mismo se cumple para una matriz de chequeo de paridad. Para mostrar que una matriz $G \in \mathcal{M}_{k \text{ x } n}(\mathbb{F}_q)$ es una matriz generadora para $\mathcal{C}$, es suficiente mostrar que las filas de G son palabras-código y que son linealmente independientes. Alternativamente, esto se puede hacer mostrando que $\mathcal{C}$ está contenido en el espacio fila de G.

Lema 2.18 Sea $\mathcal{C}$ un $[n,k]$-código lineal sobre $\mathbb{F}_q$, con matriz generadora G. Sea $\mathbf{v} \in \mathbb{F}_q^n$, entonces $\mathbf{v} \in \mathcal{C}^\perp$ si y sólo si $\mathbf{v}$ es ortogonal a cada fila de G; es decir, $\mathbf{v} \in \mathcal{C}^\perp$ si y sólo si $\mathbf{v}G^t = \mathbf{0}$. En particular, dada una matriz $H \in M_{(n-k)\times n}(\mathbb{F}_q)$, esta será una matriz de chequeo de paridad para $\mathcal{C}$ si y sólo si las filas de H son linealmente independientes y $HG^t = \mathbf{0}$.

Demostración. Denotemos por $\mathbf{r}_i$, $i = 1, ..., k$, a la i-ésima fila de G. En particular, por definición, $\mathbf{r}_i \in \mathcal{C}$ y para cada $\mathbf{c} \in \mathcal{C}$ existen $\lambda_1, ..., \lambda_k \in \mathbb{F}_q$ tales que:

$$c = \sum_{i=1}^{k} \lambda_i \mathbf{r}_i.$$

Si $\mathbf{v} \in \mathcal{C}^\perp$, entonces $\mathbf{v}\cdot\mathbf{c} = 0$ para cada $\mathbf{c} \in \mathcal{C}$, en particular, $\mathbf{v}$ es ortogonal a $\mathbf{r}_i$ para cada $1 \leq i \leq k$, de ahí que $\mathbf{v}G^t = [\mathbf{v}\cdot\mathbf{r}_1, \mathbf{v}\cdot\mathbf{r}_2, ..., \mathbf{v}\cdot\mathbf{r}_k] = \mathbf{0}$. A la inversa, si $\mathbf{v}\cdot\mathbf{r}_i = 0$ para todo $1 \leq i \leq k$, entonces, para cada $\mathbf{c} = \sum_{i=1}^{k} \lambda_i \mathbf{r}_i \in \mathcal{C}$ se tiene que:

$$\mathbf{v}\cdot\mathbf{c} = \sum_{i=1}^{k} \lambda_i(\mathbf{v}\cdot\mathbf{r}_i) = \sum_{i=1}^{k} \lambda_i(0) = 0.$$

Para la última afirmación, si H es una matriz de chequeo de paridad para $\mathcal{C}$, entonces las filas de H son linealmente independientes (por definición). Ya que las filas de H son palabras-código de $\mathcal{C}^\perp$, se sigue de la primera afirmación del lema que $HG^t = \mathbf{0}$. A la inversa, si $HG^t = \mathbf{0}$, la primera afirmación del lema muestra que las filas de H, y por lo tanto el espacio fila de H, están contenidas en $\mathcal{C}^\perp$. Ya que las filas de H son linealmente independientes, el espacio fila de H tiene dimensión $n-k$, donde $dim(\mathcal{C}^\perp) = n - dim(C) = n - k$ (pues $\mathbb{F}_q^n = \mathcal{C} \oplus \mathcal{C}^\perp$), de ahí que el espacio fila de H es igual a $\mathcal{C}^\perp$, por definición, se sigue que H es una matriz de chequeo de paridad para $\mathcal{C}$. ■

Observación 2.19 Una formulación equivalente para el Lema 2.18 (intercambiando los roles de $\mathcal{C}$ y $\mathcal{C}^\perp$) es la siguiente:

Sea $\mathcal{C}$ un $[n,k]$-código lineal sobre $\mathbb{F}_q$, con matriz de chequeo de paridad H. Entonces, $\mathbf{v} \in \mathbb{F}_q^n$ pertenece a $\mathcal{C}$ si y sólo si $\mathbf{v}$ es ortogonal a cada fila de H; es decir, $\mathbf{v} \in \mathcal{C}$ si y sólo si $\mathbf{v}H^t = \mathbf{0}$. En particular, $G \in M_{n\times k}(\mathbb{F}_q)$ es una matriz generadora para $\mathcal{C}$ si y sólo si las filas de G son linealmente independientes y $GH^t = \mathbf{0}$.

El siguiente teorema y el subsecuente corolario son resultados de gran utilidad, pues relacionan directamente a la distancia mínima de un código lineal y al conjunto de filas de una matriz de chequeo de paridad, permitiendo calcular (o acotar) la distancia mínima en base a una matriz de chequeo de paridad explícita.

Teorema 2.20 Sea $\mathcal{C}$ un $[n,k]$-código lineal sobre $\mathbb{F}_q$ y sea H una matriz de chequeo de paridad para $\mathcal{C}$. Entonces se cumple lo siguiente:

(i) $\mathcal{C}$ tiene distancia mínima mayor o igual a d si y sólo si cualesquiera $d-1$ columnas de H son linealmente independientes.

(ii) $\mathcal{C}$ tiene distancia mínima menor o igual a d si y sólo si H tiene d columnas linealmente dependientes.

Demostración. (i) [$\Longrightarrow$] (Por contradicción) Supongamos que existen $d-1$ columnas de H que son linealmente dependientes, digamos $c_{i_1}, c_{i_2}, ..., c_{i_{d-1}}$ $(1 \leq i_j \leq n)$. Entonces, existen $\lambda_{i_1}, ..., \lambda_{i_{d-1}} \in \mathbb{F}_q$ no todos cero tales que $\mathbf{0} = \sum_{j=1}^{d-1} \lambda_{i_j} c_{i_j}^t$. Sea $\mathbf{v} = (v_0, v_1, ..., v_n) \in \mathbb{F}_q^n$, donde $v_k = \lambda_{i_j}$ si $k = i_j$ y $v_k = 0$ en otro caso. Entonces $\mathbf{0} = \sum_{j=1}^{d-1} \lambda_{i_j} c_{i_j}^t = \mathbf{v}H^t$, lo cual implica (por la Observación 2.19) que $\mathbf{v} \in \mathcal{C}$, que por definición de $\mathbf{v}$, $\mathbf{v} \neq 0$ y $wt(\mathbf{v}) \leq d-1$, lo cual contradice que la distancia mínima de $\mathcal{C}$ es igual a d.

[$\Longleftarrow$] Sea $\mathbf{v} = (v_0, v_1, ..., v_{n-1}) \in \mathcal{C}$ una palabra-código distinta de cero. Denotemos por $\mathbf{c}_i$ $(1 \leq i \leq n)$ a las columnas de la matriz H, entonces por la Observación 2.19, se tiene que $\mathbf{0} = \mathbf{v}H^t = \sum_{i \in Sop(\mathbf{v})} v_i c_i^t$. Obsérvese que si $s = |Sop(\mathbf{v})| \leq d-1$, entonces tendríamos s columnas de H linealmente dependientes, y por lo tanto $d-1$ columnas de H linealmente dependientes (pues tal conjunto de s columnas se podría extender a un conjunto de $d-1$ columnas de H, y el nuevo conjunto sigue siendo linealmente dependiente), lo cual sería una contradicción. Por lo tanto $d \leq |Sop(\mathbf{v})|$, es decir, $d \leq wt(\mathbf{v})$, como $\mathbf{v}$ es una palabra-código distinta de cero arbitraria, se sigue que la distancia mínima de $\mathcal{C}$ es mayor igual a d.

(ii) [$\Longrightarrow$] Si $\mathcal{C}$ tiene distancia mínima d' menor o igual a d, entonces existe $\mathbf{v} = (v_0, v_1, ..., v_{n-1}) \in \mathcal{C}$ tal que $wt(\mathbf{v}) = d'$. Luego, por la Observación 2.19, $\mathbf{0} = \mathbf{v}H^t = \sum_{i \in Sop(\mathbf{v})} v_i c_i^t$, donde c_j son las columnas de H. Como $d' = |Sop(\mathbf{v})| \leq d$, entonces tenemos d' columnas linealmente dependientes, y por lo tanto d columnas linealmente dependientes (extendiendo el conjunto de d' columnas linealmente dependientes a un conjunto de d columnas de H, el nuevo conjunto seguirá siendo linealmente dependiente).

[$\Longleftarrow$] Como en la demostración de (i)[$\Longrightarrow$], si H tiene d columnas linealmente dependientes, entonces existe $v \in \mathcal{C}$ con $\mathbf{v} \neq 0$ y $wt(\mathbf{v}) \leq d$, lo cual implica que la distancia mínima de $\mathcal{C}$ es menor o igual a d. ■

Corolario 2.21 Sean $\mathcal{C}$ un código lineal y H una matriz de chequeo de paridad para $\mathcal{C}$. Entonces las siguientes afirmaciones son equivalentes:

(i) C tiene distancia mínima igual a d.

(ii) Cualesquiera $d-1$ columnas de H son linealmente independientes y H tiene d columnas linealmente dependientes.

Ejemplo 2.22 Sea $\mathcal{C}$ un código lineal binario con matriz de chequeo de paridad

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

Inspeccionando, H no tiene columnas iguales a $\mathbf{0}$, cualesquiera 2 filas de H son linealmente independientes, y las columnas 1, 3 y 4 son linealmente dependientes. Por lo tanto, en base al Corolario 2.21, la distancia mínima de $\mathcal{C}$ es 3.

2.3. CÓDIGO EXTENDIDO Y CÓDIGO CONCATENADO

Ahora veremos como construir algunos códigos a partir de otros que sean dados.

Definición 2.23 Para un código $\mathcal{C}$ de longitud n sobre $\mathbb{F}_q$, el código extendido de $\mathcal{C}$, denotado por $\overline{\mathcal{C}}$, se define como

$$\overline{\mathcal{C}} = \{(c_1, ..., c_n, -\sum_{i=1}^{n} c_i) : (c_1, ..., c_n) \in \mathcal{C}\}.$$

Cuando $q = 2$, la coordenada extra $-\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} c_i$ es llamada la coordenada de chequeo de paridad.

Teorema 2.24 Si $\mathcal{C}$ es un (n, M, d)-código sobre $\mathbb{F}_q$, entonces $\overline{\mathcal{C}}$ es un $(n+1, M, d')$-código sobre $\mathbb{F}_q$, donde $d \leq d' \leq d+1$. Además, si $\mathcal{C}$ es un código lineal, entonces $\overline{\mathcal{C}}$ es un código lineal y

$$\overline{\mathrm{H}} = \left(\begin{array}{ccc|c} & & & 0 \\ & \mathrm{H} & & \vdots \\ & & & 0 \\ \hline 1 & \cdots & 1 & 1 \end{array}\right)$$

es una matriz de chequeo de paridad para $\overline{\mathcal{C}}$ siempre que H sea una matriz de chequeo de paridad para $\mathcal{C}$.

Demostración. Sea $\mathcal{C}$ un (n, M, d)-código sobre $\mathbb{F}_q$. Por definición, $\overline{\mathcal{C}} \subseteq \mathbb{F}_q^{n+1}$ y $|\mathcal{C}| = |\overline{\mathcal{C}}|$, pues la función $\varphi : \mathcal{C} \longrightarrow \overline{\mathcal{C}}$ definida por $\varphi(c_0, c_1, ..., c_{n-1}) = (c_0, c_1, ..., c_{n-1}, -\sum_{i=0}^{n-1} c_i)$ es una biyección. Ya que la distancia mínima de $\mathcal{C}$ es d, entonces existe una palabra código distinta de cero $\mathbf{c} = (c_0, c_1, ..., c_{n-1}) \in \mathcal{C}$ tal que $wt(\mathbf{c}) = d$. Si $\overline{\mathbf{c}} = (c_0, c_1, ..., c_{n-1}, -\sum_{i=0}^{n-1} c_i) \in \overline{\mathcal{C}}$, entonces $\overline{\mathbf{c}} \neq \mathbf{0}$ y se cumple que $wt(\overline{\mathbf{c}}) = wt(\mathbf{c}) = d$ (si $\sum_{i=0}^{n-1} c_i = 0$) ó $wt(\overline{\mathbf{c}}) = wt(\mathbf{c}) + 1 = d + 1$ (si $\sum_{i=0}^{n-1} c_i \neq 0$), y en consecuencia la distancia mínima de $\overline{\mathcal{C}}$ es menor o igual a $d+1$. Sea $\overline{\mathbf{a}} = (a_0, a_1, ..., a_{n-1}, -\sum_{i=0}^{n-1} a_i) \in \overline{\mathcal{C}}$, con $\mathbf{0} \neq \mathbf{a} = (a_0, a_1, ..., a_{n-1}) \in \mathcal{C}$, entonces $d = wt(\mathbf{c}) \leq wt(\mathbf{a}) \leq wt(\overline{\mathbf{a}})$, de ahí que la distancia mínima de $\overline{\mathcal{C}}$ es mayor o igual a d (pues cada palabra-código distinta de cero de $\overline{\mathcal{C}}$ tiene peso mayor o igual a d). Entonces, si denotamos por d' a la distancia mínima de $\overline{\mathcal{C}}$, se tiene que $d \leq d' \leq d+1$.

Hasta ahora, se ha probado que $\overline{\mathcal{C}}$ es un $[n+1, M, d']$-código sobre $\mathbb{F}_q$. Supongamos que $\mathcal{C}$ es un código lineal, es decir, $C \leq \mathbb{F}_q^n$. Si $\mathbf{a} = (a_0, a_1, ..., a_{n-1}) \in \mathcal{C}$, donde $a_i = 0$, $i = 0, 1, ..., n-1$ entonces $\underbrace{(0, 0, ..., 0)}_{\text{longitu n+1}} = (a_0, a_1, ..., a_{n-1}, -\sum_{i=0}^{n-1} a_i) \in \overline{\mathcal{C}}$. Además, si $\mathbf{x} = (x_0, x_1, ..., x_{n-1}, -\sum_{i=0}^{n-1} x_i)$, $\mathbf{y} = (y_0, y_1, ..., y_{n-1}, -\sum_{i=0}^{n-1} y_i) \in \overline{\mathcal{C}}$ y $\lambda \in \mathbb{F}_q$ entonces $\lambda\mathbf{x} + \mathbf{y} = (\lambda x_0 + y_0, \lambda x_1 + y_1, ..., \lambda x_{n-1} + y_{n-1}, -\sum_{i=0}^{n-1} (\lambda x_i + y_i) \in \overline{\mathcal{C}}$, pues $(\lambda x_0 + y_0, \lambda x_1 + y_1, ..., \lambda x_{n-1} + y_{n-1}) \in \mathcal{C}$ por la hipótesis $\mathcal{C} \leq \mathbb{F}_q^n$. Por lo tanto $\overline{\mathcal{C}} \leq \mathbb{F}_q^{n+1}$.

Finalmente, sean $\overline{\mathbf{a}} = (a_0, a_1, ..., a_{n-1}, -\sum_{i=0}^{n-1} a_i) \in \overline{\mathcal{C}}$, con $\mathbf{a} = (a_0, a_1, ..., a_{n-1}) \in \mathcal{C}$ y H una matriz de chequeo de paridad para $\mathcal{C}$, entonces

$$\overline{\mathrm{H}}\overline{\mathbf{a}}^t = \left(\begin{array}{ccc|c} & \mathrm{H} & & \begin{matrix}0\\ \vdots \\ 0\end{matrix} \\ \hline 1 & \cdots & 1 & 1 \end{array}\right) \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \\ -\sum_{i=0}^{n-1} a_i \end{pmatrix} = \begin{pmatrix} \mathrm{H}a^t \\ \\ \sum_{i=0}^{n-1} a_i - \sum_{i=0}^{n-1} a_i \end{pmatrix} = \overline{\mathbf{0}}_{(n-k+1)\times 1},$$

donde las filas de $\overline{\mathrm{H}}$ son linealmente independientes, ya que su última fila no es combinación lineal de las anteriores y las filas de H son linealmente independientes. Por lo tanto, $\overline{\mathrm{H}}$ es una matriz de chequeo de paridad para $\overline{\mathcal{C}}$. ■

Ejemplo 2.25 (i) Considere el código lineal binario $\mathcal{C}_1 = \{000, 110, 011, 101\}$ ($\mathcal{C}_1 = \langle 110, 011\rangle$). Este código tiene parámetros $[3, 2, 2]$. El código extendido

$$\overline{\mathcal{C}_1} = \{0000, 1100, 0110, 1010\}$$

es un $[4, 2, 2]$-código lineal binario.

(ii) Considere el código lineal binario $\mathcal{C}_2 = \{000, 111, 011, 100\}$. Este código tiene parámetros $[3, 2, 1]$. El código extendido

$$\overline{\mathcal{C}_2} = \{0000, 1111, 0110, 1001\}$$

es un $[4, 2, 2]$-código lineal binario. Obsérvese que en (i) $\mathrm{d}(\mathcal{C}_1) = \mathrm{d}(\overline{\mathcal{C}_1})$, y en (ii) $\mathrm{d}(\mathcal{C}_2)+1 = \mathrm{d}(\overline{\mathcal{C}_2})$.

Teorema 2.26 (Código Concatenado) Sea A un $[N, K, D]$-código lineal sobre $\mathbb{F}_{q^m}$. Entonces existe un $[nN, mK, d']$-código lineal $\mathcal{C}$ sobre $\mathbb{F}_q$ con $d' \geq dD$, provisto de un $[n, m, d]$-código lineal B sobre $\mathbb{F}_q$.

Demostración. Consideremos a $\mathbb{F}_{q^m}$ como $\mathbb{F}_q$-espacio vectorial de dimensión m. Como B tiene dimensión m como $\mathbb{F}_q$ espacio vectorial, entonces existe una $\mathbb{F}_q$ transformación lineal $\phi : \mathbb{F}_{q^m} \longrightarrow B$ que es biyectiva (un isomorfismo). Podemos extender ϕ y obtener la función

$$\phi^* : \mathbb{F}_{q^m}^N \longrightarrow \mathbb{F}_q^{nN},\ (v_1, ..., v_N) \mapsto (\phi(v_1), ..., \phi(v_N)).$$

Como ϕ es una $\mathbb{F}_q$-transformación lineal se sigue que ϕ^* también lo es. La función ϕ^* es inyectiva ya que ϕ, en particular, es inyectiva. Obsérvese que ϕ^* es suprayectiva sólo si $m = n$. El código A puede ser visto como un $\mathbb{F}_q$-subespacio de $\mathbb{F}_{q^m}^N$ (pues $\mathbb{F}_q$ es subcampo de $\mathbb{F}_{q^m}$). Sea $\mathcal{C} = \phi^*(A)$. Entonces $\mathcal{C}$ es un subespacio de $\mathbb{F}_q^{nN}$ ya que ϕ^* es transformación lineal.
Claramente la longitud de $\mathcal{C}$ es nN. Ahora, por de la Observación 2.12, se tiene

$$dim_{\mathbb{F}_q}(\mathcal{C}) = log_q|\mathcal{C}| = log_q|A| = log_q((q^m)^{dim_{\mathbb{F}_{q^m}}(A)}) = log_q(q^{mK}) = mK,$$

donde la segunda igualdad se tiene de $\mathcal{C} = \phi^*(A)$ con ϕ^* inyectiva. Ahora, encontremos la distancia mínima de $\mathcal{C}$. Sea $(u_1, ..., u_N)$ una palabra-código distinta de cero de A. Si $u_i \neq 0$ para algún $1 \leq i \leq N$ entonces $wt\phi(u_i) \neq 0$ y $wt(\phi(u_i)) \geq d$, pues $\phi(u_i) \in B$, ϕ es isomorfismo y d es la distancia mínima de B. Como $(u_1, ..., u_N)$ tiene al menos D posiciones distintas de cero, el número de posiciones distinta de cero de $\phi(u_1, ..., u_N)$ es al menos dD. Como cada palabra código de $\mathcal{C}$ es de la forma $(\phi(u_1), ..., \phi(u_N))$ con $(u_1, ..., u_N) \in A$, se concluye que la distancia mínima de $\mathcal{C}$ es mayor que dD. ■

2.4. CÓDIGOS DE MÁXIMA DISTANCIA SEPARABLE

Definición 2.27 Un código lineal con parámetros $[n, k, d]$ se dice que es un código de máxima distancia separable (MDS) si $k + d = n + 1$.

Teorema 2.28 Sea $\mathcal{C}$ un código lineal sobre $\mathbb{F}_q$ con parámetros $[n, k, d]$. Sean G una matriz generadora y H una matriz de chequeo de paridad para $\mathcal{C}$. Entonces, las siguientes propiedades son equivalentes:
(i) $\mathcal{C}$ es un código MDS;
(ii) Cada conjunto de $n - k$ columnas de H es linealmente independiente;
(iii) Cada conjunto de k columnas de G es linealmente independiente;
(iv) $\mathcal{C}^\perp$ es un código MDS.

Demostración. $[(i) \Rightarrow (ii)]$ Si $\mathcal{C}$ es un código MDS, entonces $d = n - k + 1$. Entonces por el Corolario 2.21, se tiene (ii).

$[(ii) \Rightarrow (i)]$ Si suponemos (ii), por el Teorema 2.20, se tiene que $n - k + 1 \leq d$, y por la **cota de Singleton** $d \leq n - k + 1$. Por lo tanto, $d = n - k + 1$, es decir, $\mathcal{C}$ es un código MDS.

$[(iii) \Leftrightarrow (iv)]$ Ya que G es una matriz de chequeo de paridad para $\mathcal{C}^\perp$, análogamente a la demostración de $(i) \Longleftrightarrow (ii)$ (intercambiando el rol de $\mathcal{C}$ por $\mathcal{C}^\perp$), se tiene $(iii) \Leftrightarrow (iv)$.

$[(i) \Rightarrow (iv)]$ Ya que H es una matriz de chequeo de paridad para $\mathcal{C}$, entonces H es una matriz generadora para $\mathcal{C}^\perp$. Entonces $\mathcal{C}^\perp$ tiene parámetros $[n, n - k]$. Si mostramos que la distancia mínima d' de $\mathcal{C}$ es igual a $k + 1$, entonces $\mathcal{C}$ será un código MDS.

Supongamos que $d' \leq k$. Entonces, existe $\mathbf{c} \in \mathcal{C}^\perp - \{\mathbf{0}\}$, tal que $wt(\mathbf{c}) \leq k$, así que $\mathbf{c}$ tiene, al menos, $n - k$ componentes iguales a cero. Permutar las coordenadas no cambia el peso, así que podemos asumir que las últimas $n-k$ coordenadas de c son cero. Escribimos H como $H = (A|H')$, donde $A \in M_{(n-k)\times k}(\mathbb{F}_q)$ y $H' \in M_{(n-k)\times(n-k)}(\mathbb{F}_q)$. Ya que las columnas de H' son linealmente independientes (por la equivalencia $(i) \Longleftrightarrow (ii)$), entonces H' es invertible. Por lo tanto, las filas de H' son linealmente independientes. Entonces, al expresar $\mathbf{c}$ como combinación lineal de las filas de H, la única manera obtener $0's$ en las últimas $n - k$ es que en tales combinaciones lineales los escalares sean iguales a 0 (por la independencia lineal de las filas de H'). Por lo tanto, todas las componentes de $\mathbf{c}$ son 0, es decir, $c = \mathbf{0}$, lo cual es una contradicción a la elección de c. Se sigue que $k + 1 \leq d'$, y por la **cota de Singleton** $d' \leq n - (n - k) + 1 = k + 1$. Por lo tanto, $d' = k + 1$, lo cual se quería demostrar.

$[(iv) \Rightarrow (i)]$ Ya que $(\mathcal{C}^\perp)^\perp = \mathcal{C}$, la demostración es análoga a $(i) \Rightarrow (iv)$. ■

Los códigos MDS pueden ser bastante útiles respecto a la detección y corrección de errores, ya que la distancia mínima se puede obtener de forma explícita a través de los parámetros longitud y dimensión. Como se verá más adelante, los Códigos Reed-Solomon resultarán ser (en particular) códigos MDS.

2.5. CÓDIGOS CÍCLICOS LINEALES

A la función

$$\sigma : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^n$$

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

se le llama corrimiento cíclico.

Definición 2.29 Un código $\mathcal{C} \subseteq \mathbb{F}_q^n$ es llamado cíclico si es lineal y $\sigma(\mathcal{C}) = \mathcal{C}$.

Consideremos al anillo $\mathcal{R}_n = \frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$. Obsérvese que $\mathcal{R}_n$ admite una estructura de $\mathbb{F}_q$-espacio vectorial con la suma usual en $\mathcal{R}_n$ y el producto por escalares definido de la siguiente manera:

$$\forall \lambda \in \mathbb{F}_q, \forall f(x) + \langle x^n - 1\rangle \in \mathcal{R}_n : \lambda(f(x) + \langle x^n - 1\rangle) = (\lambda f(x)) + \langle x^n - 1\rangle.$$

Para ver la buena definición de está operación, si $c_1(x) + \langle x^n - 1\rangle = c_2(x) + \langle x^n - 1\rangle$ y $\lambda \in \mathbb{F}_q$. Entonces $c_1(x) - c_2(x) \in \langle x^n - 1\rangle$, lo cual implica que existe $r(x) \in \mathbb{F}_q[x]$ tal que $c_1(x) - c_2(x) = r(x)(x^n - 1)$; se sigue que $\lambda c_1(x) - \lambda c_2(x) = (\lambda r(x))(x^n - 1)$, es decir, $\lambda c_1(x) - \lambda c_2 \in \langle x^n - 1\rangle$ lo cual equivale a $(\lambda c_1(x)) + \langle x^n - 1\rangle = (\lambda c_2(x)) + \langle x^n - 1\rangle$. Entonces la función $\psi : \mathbb{F}_q^n \longrightarrow \mathcal{R}_n$ definida por

$$\psi(c_0, ..., c_{n-1}) = (c_0 + c_1 x + ... + c_{n-1}x^{n-1}) + \langle x^n - 1\rangle$$

es un isomorfismo de $\mathbb{F}_q$-espacios vectoriales.

Teorema 2.30 (cf. [5], [11]) Sea ψ como se definió previamente y $\emptyset \neq \mathcal{C} \subseteq \mathbb{F}_q^n$. Entonces $\mathcal{C}$ es un código cíclico si y sólo si $\psi(\mathcal{C})$ es un ideal del anillo $\mathcal{R}_n$.

A la luz del Teorema 2.30, veremos a un código cíclico como un ideal de $\mathcal{R}_n$ y recíprocamente, según el contexto. Obsérvese que dicho teorema, relaciona la estructura de subespacio y la de ideal de un anillo, lo cual puede ser bastante útil pues disponemos de una gran cantidad de información y resultados de ambas estructuras.

El anillo $\mathcal{R}_n$ tiene propiedades interesantes, una de ellas es que todos los ideales son principales, es decir, cada ideal es generado por un elemento del anillo.

Teorema 2.31 (cf. [5], [11]) Sea $\mathcal{C}$ un ideal distinto de cero del anillo $\mathcal{R}_n = \frac{\mathbb{F}_q[x]}{I}$, donde $I = \langle x^n - 1\rangle \leq \mathbb{F}_q[x]$. Entonces se cumple lo siguiente:

(i) Existe un único polinomio $g(x) \in \mathbb{F}_q[x]$ de grado mínimo tal que $\mathcal{C} = \langle g(x) + I\rangle$; al polinomio $g(x)$ se le llama polinomio generador de $\mathcal{C}$.

(ii) El polinomio generador $g(x)$ divide a $x^n - 1 \in \mathbb{F}_q[x]$.

(iii) Si $grad(g(x)) = r$, entonces la dimensión de $\mathcal{C}$ es $n - r$. Además,

$$\mathcal{C} = \{(r(x) + I)(g(x) + I) : grad(r(x)) < n - r\}.$$

(iv) Si $g(x) = \sum_{i=0}^{t} g_i x^i$, entonces $g_0 \neq 0$ y $\mathcal{C}$ tiene matriz generadora

$$G = \begin{pmatrix} g_0 & g_1 & \dots & g_t & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_t & 0 \dots & 0 \\ & & \ddots & & \ddots & & \\ 0 & \dots & 0 & g_0 & g_1 & \dots & g_r \end{pmatrix},$$

donde cada fila de G resulta de aplicar un corrimiento cíclico a la fila anterior.

Teorema 2.32 (cf. [5], [11]) Un polinomio mónico $p(x) \in \mathbb{F}_q[x]$ es el polinomio generador de un código cíclico en $\mathcal{R}_n$ si y sólo si $p(x)$ divide a $x^n - 1 \in \mathbb{F}_q[x]$.

Lema 2.33 Sea $g(x)$ el polinomio generador de un código cíclico $\mathcal{C}$ en $\mathcal{R}_n = \frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$. Entonces, $c(x) + \langle x^n - 1\rangle \in \mathcal{C}$ si y sólo si $g(x)$ divide a $c(x)$ en $\mathbb{F}_q[x]$.

Demostración. [$\Longrightarrow$] Sea $I = \langle x^n - 1\rangle \leq \mathbb{F}_q[x]$. Si $c(x) + I \in \mathcal{C}$, entonces existe $k(x) + I \in \mathcal{R}_n$ tal que $c(x) + I = (k(x) + I)(g(x) + I) = (k(x)g(x)) + I$. Se sigue que $c(x) - k(x)g(x) \in I$, lo cual implica que existe $r(x) \in \mathbb{F}_q[x]$ tal que $c(x) - k(x)g(x) = r(x)(x^n - 1)$; además, por el Teorema 2.31, existe $s(x) \in \mathbb{F}_q[x]$ tal que $x^n - 1 = s(x)g(x)$, de ahí que $c(x) = k(x)g(x) + r(x)s(x)g(x) = (k(x) + r(x)s(x))g(x)$, es decir, $g(x)|c(x)$.
[$\Longleftarrow$] Si $g(x)|c(x)$, entonces existe $r(x) \in \mathbb{F}_q[x]$ tal que $c(x) = r(x)g(x)$. Sea I como en la implicación anterior, entonces $c(x) + I = (r(x)g(x)) + I = (r(x) + I)(g(x) + I) \in \mathcal{C}$ ya que $g(x) + I \in \mathcal{C}$ y $\mathcal{C}$ es un ideal de $\mathcal{R}_n$. ■

Ejemplo 2.34 Considere el código cíclico $\mathcal{C} = \langle (1 + x) + I\rangle$ en $\mathcal{R}_3 = \frac{\mathbb{F}_2[x]}{I}$, donde I es el ideal $\langle x^3 - 1\rangle \leq \mathbb{F}_2[x]$. Obsérvese que en $\mathbb{F}_2[x]$: $x^3 - 1 = (1 + x)(1 + x + x^2)$. Por (iii) del Teorema 2.31, $dim(\mathcal{C}) = 3 - 1 = 2$. Entonces, $|\mathcal{C}| = 2^2 = 4$. Del mismo resultado, se sigue que $\mathcal{C} = \{I, (1 + x) + I, x(1 + x) + I, (1 + x)(1 + x) + I\} = \{I, (1 + x) + I, (x + x^2) + I, (1 + x^2) + I\}$, visto como subconjunto de $\mathbb{F}_2^3$ tenemos que $\mathcal{C} = \{000, 110, 011, 101\}$. Una matriz de chequeo de paridad para $\mathcal{C}$ es, por (iv) del Teorema 2.31,

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

Definición 2.35 Sea $g(x) \in \mathbb{F}_q[x]$ el polinomio generador de un $[n, n - r]$-código cíclico en $\mathcal{R}_n$. Ya que $g(x)$ divide a $x^n - 1 \in \mathbb{F}_q[x]$, entonces existe $h(x) \in \mathbb{F}_q[x]$ tal que $grad(h(x)) = n - r$ y $x^n - 1 = g(x)h(x)$. A $h(x)$ se le llama el polinomio de chequeo de $\mathcal{C}$.

Teorema 2.36 (cf. [5], [11]) Sean $\mathcal{C}$ un código cíclico en $\mathcal{R}_n$ y $h(x)$ un polinomio de chequeo para $\mathcal{C}$.

(i) $\mathcal{C}$ puede ser descrito de la siguiente manera:

$$\mathcal{C} = \{p(x) + I \in \mathcal{R}_n : (p(x) + I)(h(x) + I) = I\};$$

(ii) Si $h(x) = \sum_{j=0}^{n-r} h_j x^j$, entonces una matriz de chequeo de paridad para $\mathcal{C}$ es

$$H = \begin{pmatrix} h_{n-r} & h_{n-r-1} & ... & h_0 & 0 & 0 & ... & 0 \\ 0 & h_{n-r} & h_{n-r-1} & ... & h_0 & 0 & ... & 0 \\ & & \ddots & & & \ddots & & \\ 0 & 0 & ... & 0 & h_{n-r} & h_{n-r-1} & ... & h_0 \end{pmatrix};$$

(iv) El código dual $\mathcal{C}^\perp$ es un código cíclico de dimensión r cuyo polinomio generador es

$$h^\perp(x) = h_0^{-1}(h_0 x^{n-r} + h_1 x^{n-r-1} + ... + h_{n-r}).$$

Ejemplo 2.37 Sea $\mathcal{C}$ como en el Ejemplo 2.34. Entonces $h(x) = 1 + x + x^2$ es un polinomio de chequeo para $\mathcal{C}$. Obsérvese que $(0 + I)(h(x) + I) = I$, $((1 + x) + I)(h(x) + I) = (x^3 - 1) + I = I$, $((x + x^2) + I)(h(x) + I) = (x^4 - x) + I = (x(x^3 - 1)) + I = I$, $((1 + x^2) + I)(h(x) + I) = (x^3 + 1)(1 + x) + I = I$, lo cual era de esperarse por (i) del Teorema 2.36. Ahora, por (ii) del mismo teorema se tiene que

$$\mathrm{H} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

es una matriz de chequeo de paridad para $\mathcal{C}$, de ahí que $\mathcal{C}^\perp = \{000, 111\}$. En este caso, el polinomio de chequeo de paridad y el polinomio generador de $\mathcal{C}^\perp$ coinciden.

2.6. LOS CEROS DE UN CÓDIGO CÍCLICO

Si tenemos acceso a las raíces del polinomio $x^n - 1 \in \mathbb{F}_q[x]$ (esto es, a las raíces n-ésimas de la unidad), entonces es posible caracterizar a los códigos cíclicos en $\mathcal{R}_n$ de una forma ligeramente diferente que a través de polinomios generadores. Sea

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

la factorización de $x^n - 1$ en factores irreducibles sobre $\mathbb{F}_q$. Obsérvese que si α es una raíz de $m_i(x)$ en alguna extensión de $\mathbb{F}_q$, entonces $m_i(x)$ es el polinomio mínimo de α sobre $\mathbb{F}_q$. Teniendo esto en consideración se tiene el siguiente teorema.

Teorema 2.38 Sea $g(x) = q_1(x)q_2(x)\cdot ...\cdot q_t(x)$ un producto de factores irreducibles de $x^n - 1 \in \mathbb{F}_q[x]$ y sea $\{\alpha_1, \alpha_2 ..., \alpha_u\}$ el conjunto de las raíces de $g(x)$ en el campo de descomposición de $x^n - 1$ sobre $\mathbb{F}_q$. Entonces, en $\mathcal{R}_n$ se tiene lo siguiente:

$$\begin{aligned} \langle g(x) + I \rangle &= \{f(x) + I \in \mathcal{R}_n : f(\alpha_1) = 0, f(\alpha_2) = 0, ..., f(\alpha_u) = 0\} \\ &= \{f(x) + I \in \mathcal{R}_n : f(\beta_1) = 0, f(\beta_2) = 0, ..., f(\beta_t) = 0\}, \end{aligned}$$

donde $I = \langle x^n - 1 \rangle \leq \mathbb{F}_q[x]$ y β_i es alguna raíz de $q_i(x)$, $i = 1, ..., t$.

Demostración. Sean $A_1 = \langle g(x) + I \rangle$, $A_2 = \{f(x) + I \in \mathcal{R}_n : f(\alpha_1) = 0, f(\alpha_2) = 0, ..., f(\alpha_u) = 0\}$ y $A_3 = \{f(x) + I \in \mathcal{R}_n : f(\beta_1) = 0, f(\beta_2) = 0, ..., f(\beta_t) = 0\}$. Sea $c(x) + I \in A_1$, por el Lema 2.33, se sigue $g(x)|c(x)$ lo cual implica que cada raíz de $g(x)$ en el campo descomposición de $x^n - 1$ sobre $\mathbb{F}_q$ es también raíz de $c(x)$, es decir, $c(x) + I \in A_2$. Ya que cada raíz de $q_i(x)$ es raíz de $g(x)$, se tiene que $A_2 \subseteq A_3$. Sea $r(x) + I \in A_3$, entonces $r(\beta_i) = 0$ para alguna raíz β_i (en el campo de descomposición de $x^n - 1$ sobre $\mathbb{F}_q$) de $q_i(x)$, $i = 1, ..., t$. Dado que $q_i(x)$ es el polinomio mínimo de β_i sobre $\mathbb{F}_q$ se tiene que $q_i(x)|r(x)$ para cada $i = 1, ..., t$. Entonces cada $q_i(x)$ es un factor irreducible de $r(x)$ en $\mathbb{F}_q[x]$, por la unicidad de la factorización de $r(x)$ en irreducibles sobre $\mathbb{F}_q$, se sigue que $g(x) = q_i(x)q_2(x) \cdot ... \cdot q_t(x)$ es un factor de $r(x)$, es decir, $g(x)|r(x)$ lo cual implica (Lema 2.33) que $r(x) + I \in A_1$. La afirmación del teorema se sigue de que $A_1 \subseteq A_2 \subseteq A_3 \subseteq A_1$. ■

Definición 2.39 Las raíces del polinomio generador de un código cíclico son llamados los ceros del código. El resto de raíces de la unidad son llamadas "no ceros" del código.

Observación 2.40 (i) Obsérvese que si $\{\alpha_1, ..., \alpha_u\}$ es algún conjunto de raíces en el campo de descomposición de $x^n - 1$ sobre $\mathbb{F}_q$, entonces el polinomio generador del código

$$\{f(x) + \langle x^n - 1\rangle \in \mathcal{R}_n : f(\alpha_1) = ... = f(\alpha_u) = 0\}$$

es el mínimo común múltiplo de los polinomios mínimos sobre $\mathbb{F}_q$ para las raíces $\alpha_1, ..., \alpha_u$.

(ii) La representación de un código a través de sus ceros puede ser utilizada para obtener una matriz de chequeo de paridad para el código. Sea $\{\alpha_1, ..., \alpha_u\}$ un conjunto de raíces de $x^n - 1 \in \mathbb{F}_q[x]$, y supongamos que se encuentran en la extensión de campo $\mathbb{F}_{q^d}$. Sea $f(x) + \langle x^n - 1\rangle \in \mathcal{R}_n$, con $f(x) = \sum_j f_j x^j \in \mathbb{F}_q[x]$. Entonces,

$$f(\alpha_i) = 0 \text{ si y sólo si } \sum_j f_j \alpha_i^j = 0. \qquad (*)$$

Obsérvese que $\mathbb{F}_{q^d} \cong \mathbb{F}_q^d$ como $\mathbb{F}_q$-espacios vectoriales (pues ambos tienen dimensión d), vía este isomorfismo a cada potencia α_i^j la podemos pensar como un único vector columna $[\alpha_i^j]$ de longitud d sobre $\mathbb{F}_q$. Además, como $f_j \in \mathbb{F}_q$ se cumple que $[f_j \alpha_i^j] = f_j[\alpha_i^j]$. Se sigue que (*) es equivalente a $\sum_j f_j[\alpha_i^j] = [\sum_j f_j \alpha_i^j] = \mathbf{0}$.
Entonces, si definimos la matriz $H \in \mathcal{M}_{ud \text{ x } n}(\mathbb{F}_q)$ por

$$H = \begin{pmatrix} [\alpha_1^0] & [\alpha_1^1] & \cdots & [\alpha_1^{n-1}] \\ [\alpha_2^0] & [\alpha_2^1] & \cdots & [\alpha_2^{n-1}] \\ \vdots & \vdots & & \vdots \\ [\alpha_u^0] & [\alpha_u^1] & \cdots & [\alpha_u^{n-1}] \end{pmatrix},$$

y si $f = (f_0, ..., f_{n-1}) \in \mathbb{F}_q^n$, entonces $f(\alpha_i) = 0$ para cada $i = 1, ..., u$, si y sólo si $fH^t = \mathbf{0}$, es decir, por el Teorema 2.38, $f(x) + \langle x^n - 1\rangle$ es elemento del código cíclico generado por el mínimo común múltiplo de los polinomios mínimos de $\alpha_1, ..., \alpha_u$ si y sólo si $fH^t = \mathbf{0}$. Por supuesto, las filas de H pueden no ser linealmente independientes, pero eliminando cualquier fila dependiente obtenemos una matriz de chequeo de paridad para el código cíclico con ceros $\{\alpha_1, ..., \alpha_u\}$.

Capítulo 3

CÓDIGOS BCH

Podemos definir a un código cíclico especificando su polinomio generador $g(x)$, o equivalentemente, especificando las raíces n-ésimas de la unidad que son raíces de $g(x)$.

A partir de ahora, se considerara a $\mathbb{F}_q$ un campo finito y n un entero positivo tal que $mcd(q,n)=1$. Ya que las raíces n-ésimas de la unidad sobre $\mathbb{F}_q$ forman un grupo cíclico bajo la multiplicación, entonces existe ω raíz n-ésima de la unidad tal que $\{1,\omega,\omega^2,...,\omega^{n-1}\}$ es el conjunto de todas las raíces n-ésimas de la unidad. A ω se le llama raíz n-ésima primitiva de la unidad. Por ejemplo, podemos definir un código cíclico requiriendo que su polinomio generador $g(x)$ sea el polinomio de menor grado que tenga entre sus raíces a los elementos "consecutivos"

$$\omega,\omega^2,...,\omega^{\delta-1}$$

cuyos exponentes de los anteriores elementos son los $\delta-1$ enteros consecutivos $1,2,...,\delta-1$ para algún $\delta\geq 1$. Claro, esto significa que $g(x)$ puede tener raíces adicionales a los elementos anteriores. Tales códigos, como el definido anteriormente, son llamados códigos BCH (en el sentido estricto) y forman una de las familias más importantes de códigos.

3.1. PARÁMETROS DE LOS CÓDIGOS BCH

Definición 3.1 Sea ω una raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$, y sea $g(x)\in\mathbb{F}_q[x]$ el polinomio mónico de menor grado que tiene a los $\delta-1$ elementos

$$\omega^b,\omega^{b+1},...,\omega^{b+\delta-2}$$

entre sus raíces, donde $0\leq b\leq n-1$ y $2\leq\delta$. Luego,

$$g(x)=mcm\{M_b(x),M_{b+1}(x),...,M_{b+\delta-2}(x)\},$$

donde $M_i(x)$ denota el polinomio mínimo de ω^i sobre $\mathbb{F}_q$.
El código cíclico $\mathfrak{B}_q(n,\delta,\omega,b)$ de longitud n y con polinomio generador $g(x)$, es llamado un código BCH con distancia designada δ.

De la anterior definición, destacamos dos casos especiales.

1.- Cuando $b=1$, el código $\mathfrak{B}_q(n,\delta,\omega,b)$ es llamado código BCH en el sentido estricto.
2.- Cuando ω es un elemento primitivo del campo de descomposición de $x^n-1\in\mathbb{F}_q[x]$, entonces el código $\mathfrak{B}_q(n,\delta,\omega,b)$ es llamado código BCH primitivo. En este caso, en la notación se sustituye a ω por α. También obsérvese que α al ser una raíz n-ésima entonces $\alpha\in\mathbb{F}_{q^m}$ para algún $m\in\mathbb{N}$, y si α es elemento primitivo de $\mathbb{F}_{q^m}$ entonces α tiene orden q^m-1, es decir, $n=q^m-1$. Por lo

tanto, cada vez que nos referimos a un código BCH primitivo, la longitud es de la forma $q^m - 1$ para algún $m \in \mathbb{N}$.

Teorema 3.2 (i) Sea α un elemento primitivo de $\mathbb{F}_{q^m}$. Entonces, la dimensión del código BCH q-ario $\mathfrak{B}_q(q^m - 1, \delta, \alpha, b)$ es independiente de la elección del elemento primitivo α.
(ii) Un código BCH q-ario de longitud $q^m - 1$ y distancia designada δ tiene dimensión mayor o igual a $q^m - 1 - m(\delta - 1)$.

Demostración. (i) Para $b \leq i \leq b + \delta - 2$ sea $C_q(i)$ la clase ciclotómica de q módulo $q^m - 1$ que contiene a i. Además, consideremos a $S = \bigcup_{i=b}^{b+\delta-2} C_q(i)$. Entonces, $g(x) = \prod_{i \in S}(x - \alpha^i)$, y en consecuencia $q^m - 1 - grad(g(x)) = q^m - 1 - |S|$. Ya que S es independiente de la elección de α, el resultado deseado se sigue.
(ii) Sea $i \in \{b, b+1, ..., \leq b + \delta - 2\}$. Dado que $q^m \equiv 1 (\text{mód } q^m - 1)$, entonces $q^m i \equiv i (\text{mód } q^m - 1)$ y según lo visto en la sección de clases ciclotómicas se sigue que $|C_q(i)| \leq m$ donde $C_q(i)$ es la clase ciclotómica de q módulo $q^m - 1$ que contiene a i. Por la parte (i), y el parráfo anterior la dimensión k del código satisface lo siguiente:

$$k = q^m - 1 - |S| = q^m - 1 - |\bigcup_{i=b}^{b+\delta-2} C_q(i)| \geq q^m - 1 - \sum_{i=b}^{b+\delta-2} |C_q(i)| \geq q^m - 1 - \sum_{i=b}^{b+\delta-2} m$$
$$= q^m - 1 - m(\delta - 1).$$

■

El resultado anterior muestra que, para encontrar la dimensión del código BCH q-ario de longitud $q^m - 1$ generado por

$$g(x) = mcm\{M_b(x), ..., M_{b+\delta-2}(x)\},$$

es suficiente calcular la cardinalidad de $\bigcup_{i=b}^{b+\delta-2} C_q(i)$, donde $C_q(i)$ es la clase ciclotómica de q módulo $q^m - 1$ que contiene a i.

Ejemplo 3.3 (i) Considere las siguientes clases ciclotómicas de 2 módulo 15:

$$C_2(2) = \{1, 2, 4, 8\},\ C_2(3) = \{3, 6, 9, 12\}.$$

Entonces, la dimensión del código binario BCH $\mathfrak{B}_2(15 = 2^4 - 1, 3, \alpha, 2)$ generado por $g(x) = mcm\{M_2(x), M_3(x)\}$ es

$$k = 15 - |C_2(2) \cup C_2(3)| = 15 - 8 = 7.$$

En este caso, se "alcanza" la cota inferior del Teorema 3.2, pues $k = 7 = 2^4 - 1 - 4(3 - 1)$.

(ii) Ahora, considere las siguientes clases ciclotómicas de 3 módulo 26:

$$C_3(1) = \{1, 3, 9\},\ C_3(2) = \{2, 6, 18\},\ C_3(4) = \{4, 10, 12\}.$$

Entonces, la dimensión del código ternario BCH $\mathfrak{B}_3(26 = 3^3 - 1, 5, \alpha, 1)$ generado por $g(x) = mcm\{M_1(x), ..., M_4(x)\}$ es

$$k = 26 - |C_3(1) \cup C_3(2) \cup C_3(3) \cup C_3(4)| = 26 - |C_3(1) \cup C_3(2) \cup C_3(4)| = 26 - 9 = 17.$$

En este caso, la dimensión es estrictamente mayor que la cota inferior del Teorema 3.2, pues $17 > 3^3 - 1 - 3(5-1) = 14$.

Observación 3.4 Sea $t \geq 1$. Como $t \equiv t \pmod{2^m - 1}$ y $1 \equiv 2^m \pmod{2^m - 1}$, entonces $t \equiv t2^m \pmod{2^m - 1}$, o bien, $t \equiv 2t2^{m-1} \pmod{2^m - 1}$ lo cual implica que $t \in C_q(2t)$, y así $C_q(t) = C_q(2t)$ pues las clases ciclotómicas son clases de equivalencia. De lo anterior, se sigue que respecto al campo binario $M_t(x) = M_{2t}(x)$. Por lo tanto,

$$mcm\{M_1(x), ..., M_{2t-1}(x)\} = mcm\{M_1(x), ..., M_{2t}(x)\},$$

es decir, la clase de los códigos binarios primitivos BCH en el sentido estricto es la misma que la de los códigos binarios primitivos BCH en el sentido estricto con distancia designada un entero positivo impar.

Ejemplo 3.5 Sea α una raíz de $1 + x + x^4 \in \mathbb{F}_2[x]$. Entonces α es un elemento primitivo de $\mathbb{F}_{16}$. Considere las clases ciclotómicas de 2 módulo 15:

$$C_2(0) = \{0\},\ C_2(1) = \{1, 2, 4, 8\},\ C_2(3) = \{3, 6, 9, 12\},\ C_2(5) = \{5, 10\},\ C_2(7) = \{7, 11, 13, 14\}.$$

De lo anterior podemos determinar los respectivos polinomios mínimos.

$M_0(x) = \prod_{j \in C_2(0)} (x - \alpha^j) = 1 + x$

$M_1(x) = M_2(x) = M_4(x) = M_8(x) = \prod_{j \in C_2(1)} (x - \alpha^j) = 1 + x + x^4$

$M_3(x) = M_6(x) = M_9(x) = M_{12}(x) = \prod_{j \in C_2(3)} (x - \alpha^j) = 1 + x + x^2 + x^3 + x^4$

$M_5(x) = M_{10}(x) = \prod_{j \in C_2(5)} (x - \alpha^j) = 1 + x + x^2$

$M_7(x) = M_{11}(x) = M_{13}(x) = M_{14}(x) = \prod_{j \in C_2(7)} (x - \alpha^j) = 1 + x^3 + x^4$

Por el Teorema 3.2, el código binario primitivo BCH en el sentido estricto $\mathfrak{B}_2(15 = 2^4 - 1, 3, \alpha, 1)$ tiene dimensión

$$k = 15 - |\bigcup_{i=1}^{2} C_q(i)| = 15 - |C_2(1)| = 15 - 4 = 11,$$

y tiene polinomio generador

$$g(x) = mcm\{M_1(x), M_2(x)\} = M_1(x) = 1 + x + x^4,$$

el cual verifica que $k = 15 - grad(g(x)) = 15 - 4 = 11$, lo cual era de esperarse.

Similarmente, con los datos anteriores, podemos calcular la dimensión y el polinomio generador de códigos binarios BCH en el sentido estricto de longitud 15. Para algunos de ellos, presentamos los cálculos respectivos en la siguiente tabla.

n	k	δ	$g(x)$
15	11	3	$1 + x + x^4$
15	7	5	$(1 + x + x^4)(1 + x + x^2 + x^3 + x^4)$
15	5	7	$(1 + x + x^4)(1 + x + x^2 + x^3 + x^4)(1 + x + x^2)$

Teorema 3.6 Un código q-ario primitivo BCH en el sentido estricto $\mathfrak{B}_q(q^m - 1, \delta, \alpha, 1)$ tiene dimensión $k = q^m - 1 - m(\delta - 1)$, siempre que $q \neq 2$ y $mcd(q^m - 1, e) = 1$ para cada $e \in \{1, ..., \delta - 1\}$.

Demostración. De la demostración del Teorema 3.2, sabemos que la dimensión es igual a

$$k = q^m - 1 - |\bigcup_{i=1}^{\delta-1} C_q(i)|$$

donde $C_q(i)$ es la clase ciclotómica de q módulo q^m-1 que contiene a i. Por lo tanto, es suficiente demostrar que $|C_q(i)| = m$ para cada $i \in \{1, ..., \delta-1\}$, y que $C_q(i) \cap C_q(j) = \emptyset$ para $i, j \in \{1, ..., \delta-1\}$ e $i \neq j$. Obsérvese que $1 \equiv q^m \pmod{q^m-1}$, lo cual implica que $i \equiv q^m i \pmod{q^m-1}$ para $i \in \{1, ..., \delta-1\}$. Para cada entero t, tal que $1 \leq t \leq m-1$, afirmamos que $i \not\equiv q^t i \pmod{q^m-1}$ con $1 \leq i \leq \delta-1$. En caso contrario, tendríamos $(q^t-1)i \equiv 0 \pmod{q^m-1}$, o equivalentemente, $(q^m-1)|(q^t-1)i$ con $(q^m-1, i) = 1$ por hipótesis, entonces $(q^m-1)|(q^t-1)$, lo cual es una contradicción pues $t < m$. De lo anterior, tenemos que m es el menor entero positivo talque $i \equiv q^m i \pmod{q^m-1}$ para $i \in \{1, ..., \delta-1\}$, se sigue que $\forall i \in \{1, ..., \delta-1\}$: $|C_q(i)| = m$.

Sean $i, j \in \{1, ..., \delta-1\}$ tales que $i < j$. Supongamos que $j \equiv q^s i \pmod{q^m-1}$ para algún entero $s \geq 0$. Entonces $(j-i) \equiv (q^s-1)i \pmod{q^m-1}$, lo cual es equivalente a $(q^m-1)|((q^s-1)i-(j-i))$. Ya que $(q-1)|(q^m-1)$ (pues $q^m-1 = (q-1)(q^{m-1}+q^{m-2}+...+1)$), entonces $(q-1)|((q^s-1)i-(j-i))$ por la transitividad de la relación de divisibilidad, y como $(q-1)|(q^s-1)i$ (pues $(q^s-1)i = (q-1)(q^{s-1}+q^{s-2}+...+1)i$) entonces $(q-1)|(j-i)$. De $(q-1)|(j-i)$ y $(q-1)|(q^m-1)$ se sigue que $(q-1)|mcd(q^m-1, j-i)$, y esto es una contradicción pues por hipótesis $q-1 \geq 2$ y $mcd(q^m-1, j-i) = 1$ ya que $1 \leq j-i \leq \delta-1$. Entonces $j \notin C_q(i)$, lo cual implica que $Cq(j) \neq C_q(i)$ y como estas son clases de equivalencia, se tiene que $C_q(j) \cap C_q(i) = \emptyset$.

■

Ejemplo 3.7 Considere el siguiente código BCH: $\mathfrak{B}_4(63 = 4^3-1, 3, \alpha, 1)$ con α un elemento primitivo de $\mathbb{F}_{64}$.
Tenemos que las clases ciclotómicas de 4 módulo 63 que contienen a 1 y 2 respectivamente son: $C_4(1) = \{1, 4, 16\}$, $C_4(2) = \{2, 8, 32\}$. Así, sabemos que $k = 63 - |C_4(1) \cup C_4(2)| = 63 - 6 = 57$. Sin embargo, la dimensión en este caso, puede ser calculada más directamente. Note que $4 \neq 2$ y $mcd(63, i) = 1$ para $i \in \{1, 2\}$, así que por el Teorema 3.6, $k = (64-1) - 3(3-1) = 63 - 6 = 57$. Como hicimos notar antes, dentro de los códigos binarios primitivos BCH en el sentido estricto es suficiente considerar a los de distancia un número impar positivo.

Teorema 3.8 Para un código binario primitivo BCH en el sentido estricto $\mathfrak{B}_2(2^m-1, \delta = 2t+1, \alpha, 1)$, su dimensión k está acotada inferiormente por $n - m\frac{\delta-1}{2}$.

Demostración. Como se señalo en la Observación 3.4, tenemos que $C_2(i) = C_2(2i)$ para $i \in \{1, ..., 2t\}$. Además, si $i \in \{1, .., 2t\}$ tenemos que $i \equiv 2^m \pmod{2^m-1}$, así que $|C_2(i)| = \text{mín}\{n \in \mathbb{N} | i \equiv i \cdot 2^n \pmod{2^m-1}\} \leq m$. Entonces la dimensión k satisface

$$k = 2^m - 1 - |\bigcup_{i=1}^{2t} C_q(i)| = 2^m - 1 - |\bigcup_{i=1}^{t} C_q(2i-1)| \geq 2^m - 1 - \sum_{i=1}^{t} |C_q(2i-1)| \geq 2^m - 1 - tm$$

$$= 2^m - 1 - m\frac{\delta-1}{2}$$

■

Ejemplo 3.9 Un código binario primitivo BCH $\mathfrak{B}_2(63 = 2^6-1, 5, \alpha, 1)$ tiene dimensión $k = 63 - |\bigcup_{i=1}^{4} C_q(i)| = 63 - |C_2(1) \cup C_2(3)| = 63 - |\{1, 2, 4, 8, 16, 32\} \cup \{3, 6, 12, 24, 33, 48\}| = 63 - 12 = 51$, y en este caso se alcanza la cota inferior del Teorema 3.8, pues $51 = 63 - 6\frac{5-1}{2}$.

Sin embargo, un código binario primitivo BCH $\mathfrak{B}_2(31 = 2^5 - 1, 11, \alpha, 1)$ tiene dimensión $k = 31 - |\bigcup_{i=1}^{10} C_q(i)| = 31 - |C_2(1) \cup C_2(3) \cup C_2(5) \cup C_2(7)| = 31 - |\{1,2,4,8,16\} \cup \{3,6,12,17,24\} \cup \{5,9,10,18,20\} \cup \{7,14,19,25,28\}| = 31 - 20 = 11$, donde $11 > 6 = 31 - 5\frac{11-1}{2}$, es decir, la dimensión en este caso es estrictamente mayor que la cota del Teorema 3.8.

Ahora, veamos un lema sobre códigos cíclicos que será de utilidad posteriormente.

Lema 3.10 Sea $\mathcal{C}$ un código cíclico de longitud n con polinomio generador $g(x)$. Supongamos que $\alpha_1, ..., \alpha_r$ son todas la raíces de $g(x)$ y que $g(x)$ no tiene raíces múltiples. Entonces un elemento $c(x) + \langle x^n - 1\rangle \in \frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$ es una palabra-código de $\mathcal{C}$ si y sólo si $c(\alpha_i) = 0$ para cada $i \in \{1, ..., r\}$.

Demostración. [⇒] Sea $c(x)+\langle x^n-1\rangle$ una palabra código de $\mathcal{C}$, como $g(x)$ es el polinomio generador de $\mathcal{C}$, entonces existe $f(x)+\langle x^n-1\rangle \in \frac{\mathbb{F}_q}{\langle x^n-1\rangle}$ tal que $c(x)+\langle x^n-1\rangle = g(x)f(x)+\langle x^n-1\rangle$; se sigue que $c(x) - g(x)f(x) \in \langle x^n - 1\rangle$, y en consecuencia, existe $k(x) \in \mathbb{F}_q[x]$ tal que $c(x) - g(x)f(x) = k(x)(x^n - 1)$. Dado que $g(x)$ divide a $x^n - 1$, se tiene que $c(\alpha_i) = g(\alpha_i)f(\alpha_i) + k(\alpha_i)(\alpha_i^n - 1) = 0 \cdot f(\alpha_i) + k(\alpha_i) \cdot 0 = 0$ para cada $i \in \{1, ..., r\}$, es decir, $c(\alpha_i) = 0$ para cada $i \in \{1, ..., r\}$.

[⇐] Si $c(\alpha_i) = 0$ para cada $i \in \{1, ..., r\}$, entonces $c(x)$ es divisible por $g(x)$ ya que $g(x)$ no tiene raíces múltiples. Entonces existe $k(x) \in \mathbb{F}_q[x]$ tal que $c(x) = k(x)g(x)$, lo cual implica en $\frac{\mathbb{F}_q}{\langle x^n-1\rangle}$ que $c(x) + \langle x^n - 1\rangle = k(x)g(x) + \langle x^n - 1\rangle = (k(x) + \langle x^n - 1\rangle)(g(x) + \langle x^n - 1\rangle) \in \mathcal{C}$, es decir, $c(x) + \langle x^n - 1\rangle$ es una palabra-código de $\mathcal{C}$.

■

Ejemplo 3.11 Considere el $[7, 4]$-código binario de Hamming con polinomio generador $g(x) = 1 + x + x^3 \in \mathbb{F}_2[x]$. Como todos los elementos de $\mathbb{F}_8 - \{0, 1\}$ son raíces de $c(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 = \frac{x^7-1}{x-1} \in \mathbb{F}_2[x]$, y $\mathbb{F}_8$ es el campo de descomposición de $g(x)$, entonces todas las raíces de $g(x)$ son raíces de $c(x)$. Por lo tanto, $(1 + x + x^2 + x^3 + x^4 + x^5 + x^6) + \langle x^7 - 1\rangle \in \mathcal{C}$, o visto $\mathcal{C}$ como subespacio de $\mathbb{F}_2^7$, se tiene que $(1, 1, 1, 1, 1, 1, 1)$ es una palabra-código.

El siguiente resultado es de suma importancia dentro de la clase de los códigos BCH, pues relaciona directamente la distancia designada y la distancia mínima. Nos referiremos a este resultado como la cota BCH.

Teorema 3.12 (La cota BCH) Un código q-ario BCH, $\mathcal{C} = \mathfrak{B}_q(n, \delta, \omega, b)$, tiene distancia mínima mayor o igual a δ (la distancia designada).

Demostración. Sabemos que el polinomio generador de $\mathfrak{B}_q(m, \delta, \omega, b)$ es

$$g(x) = mcm\{M_b(x), M_{b+1}(x), ..., M_{b+\delta-2}\}.$$

Además está claro que $\omega^b, ..., \omega^{b+\delta-2}$ son raíces de $g(x)$, ya que $M_j(x)|g(x)$ para cada $j \in \{b, b + 1, ..., b + \delta - 2\}$. Supongamos que la distancia mínima $\mathbf{d}$ de $\mathcal{C}$ es menor que δ. Entonces existe una palabra código distinta de cero $c(x) = c_0 + c_1x + ... + c_{n-1}x^{n-1}$ tal que $wt(c) = d < \delta$. Por el Lema 3.10, tenemos que $c(\omega^i) = 0$ para cada $i \in \{b, ..., b + \delta - 2\}$; esto

último se puede representar matricialmente de la siguiente manera:

$$\begin{pmatrix} 1 & \omega^b & (\omega^b)^2 & \cdots & (\omega^b)^{n-1} \\ 1 & \omega^{b+1} & (\omega^{b+1})^2 & \cdots & (\omega^{b+1})^{n-1} \\ 1 & \omega^{b+2} & (\omega^{b+2})^2 & \cdots & (\omega^{b+2})^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{b+\delta-2} & (\omega^{b+\delta-2})^2 & \cdots & (\omega^{b+\delta-2})^{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3.1}$$

Supongamos que el soporte de $c(x)$ es $R = \{i_1, ..., i_d\}$, es decir, $c_j \neq 0$ si y sólo si $j \in R$. Entonces, (3.1) se convierte en

$$\begin{pmatrix} (\omega^b)^{i_1} & (\omega^b)^{i_2} & (\omega^b)^{i_3} & \cdots & (\omega^b)^{i_d} \\ (\omega^{b+1})^{i_1} & (\omega^{b+1})^{i_2} & (\omega^{b+1})^{i_3} & \cdots & (\omega^{b+1})^{i_d} \\ (\omega^{b+2})^{i_1} & (\omega^{b+2})^{i_2} & (\omega^{b+2})^{i_3} & \cdots & (\omega^{b+2})^{i_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\omega^{b+\delta-2})^{i_1} & (\omega^{b+\delta-2})^{i_2} & (\omega^{b+\delta-2})^{i_3} & \cdots & (\omega^{b+\delta-2})^{i_d} \end{pmatrix} \begin{pmatrix} c_{i_1} \\ c_{i_2} \\ c_{i_3} \\ \vdots \\ c_{i_d} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3.2}$$

Como $\mathbf{d} \leq \delta - 1$, entonces podemos obtener un sistema de ecuaciones eligiendo las $\mathbf{d}$ primeras ecuaciones de (3.2), y obtenemos lo siguiente:

$$\begin{pmatrix} (\omega^b)^{i_1} & (\omega^b)^{i_2} & (\omega^b)^{i_3} & \cdots & (\omega^b)^{i_d} \\ (\omega^{b+1})^{i_1} & (\omega^{b+1})^{i_2} & (\omega^{b+1})^{i_3} & \cdots & (\omega^{b+1})^{i_d} \\ (\omega^{b+2})^{i_1} & (\omega^{b+2})^{i_2} & (\omega^{b+2})^{i_3} & \cdots & (\omega^{b+2})^{i_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\omega^{b+d-1})^{i_1} & (\omega^{b+d-1})^{i_2} & (\omega^{b+d-1})^{i_3} & \cdots & (\omega^{b+d-1})^{i_d} \end{pmatrix} \begin{pmatrix} c_{i_1} \\ c_{i_2} \\ c_{i_3} \\ \vdots \\ c_{i_d} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3.3}$$

El determinante $\mathbf{D}$ de la matriz de coeficientes del sistema de ecuaciones en (3.3) es igual a

$$\mathbf{D} = \begin{vmatrix} (\omega^b)^{i_1} & (\omega^b)^{i_2} & (\omega^b)^{i_3} & \cdots & (\omega^b)^{i_d} \\ (\omega^{b+1})^{i_1} & (\omega^{b+1})^{i_2} & (\omega^{b+1})^{i_3} & \cdots & (\omega^{b+1})^{i_d} \\ (\omega^{b+2})^{i_1} & (\omega^{b+2})^{i_2} & (\omega^{b+2})^{i_3} & \cdots & (\omega^{b+2})^{i_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\omega^{b+d-1})^{i_1} & (\omega^{b+d-1})^{i_2} & (\omega^{b+d-1})^{i_3} & \cdots & (\omega^{b+d-1})^{i_d} \end{vmatrix}$$

$$= \begin{vmatrix} (\omega^b)^{i_1} \cdot 1 & (\omega^b)^{i_2} \cdot 1 & (\omega^b)^{i_3} \cdot 1 & \cdots & (\omega^b)^{i_d} \cdot 1 \\ (\omega^b)^{i_1} \cdot \omega^{i_1} & (\omega^b)^{i_2} \cdot \omega^{i_2} & (\omega^b)^{i_3} \cdot \omega^{i_3} & \cdots & (\omega^b)^{i_d} \cdot \omega^{i_d} \\ (\omega^b)^{i_1} (\omega^2)^{i_1} & (\omega^b)^{i_2} (\omega^2)^{i_2} & (\omega^b)^{i_3} (\omega^2)^{i_3} & \cdots & (\omega^b)^{i_d} (\omega^2)^{i_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\omega^b)^{i_1} (\omega^{d-1})^{i_1} & (\omega^b)^{i_2} (\omega^{d-1})^{i_2} & (\omega^b)^{i_3} (\omega^{d-1})^{i_3} & \cdots & (\omega^b)^{i_d} (\omega^{d-1})^{i_d} \end{vmatrix}$$

$$= \prod_{j=1}^{\mathbf{d}} (\omega^b)^{i_j} \begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ \omega^{i_1} & \omega^{i_2} & \omega^{i_3} & \cdots & \omega^{i_d} \\ (\omega^2)^{i_1} & (\omega^2)^{i_2} & (\omega^2)^{i_3} & \cdots & (\omega^2)^{i_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\omega^{d-1})^{i_1} & (\omega^{d-1})^{i_2} & (\omega^{d-1})^{i_3} & \cdots & (\omega^{d-1})^{i_d} \end{vmatrix}$$

$$= \prod_{j=1}^{\mathbf{d}} (\omega^b)^{i_j} \begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ \omega^{i_1} & \omega^{i_2} & \omega^{i_3} & \cdots & \omega^{i_d} \\ (\omega^{i_1})^2 & (\omega^{i_2})^2 & (\omega^{i_3})^2 & \cdots & (\omega^{i_d})^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\omega^{i_1})^{d-1} & (\omega^{i_2})^{d-1} & (\omega^{i_3})^{d-1} & \cdots & (\omega^{i_d})^{d-1} \end{vmatrix}$$

$$= \prod_{j=1}^{\mathbf{d}} (\omega^b)^{i_j} \prod_{1 \le l < k \le \mathbf{d}} (\omega^{i_k} - \omega^{l_l}) \tag{3.4}$$

Obsérvese que en la última igualdad está involucrado un determinante de Vandermonde de orden $\mathbf{d}$, para el cual se tiene una forma explícita. Por otro lado $\prod_{j=1}^{\mathbf{d}} (\omega^b)^{i_j} \neq 0$, pues en caso contrario, existiría $j_0 \in \{1, ..., d\}$ tal que $(\omega^b)^{i_{j_0}} = 0$, es decir, $\omega^{b \cdot i_{j_0}} = 0$, lo cual implica que $\omega = 0$, y esto es una contradicción ya que ω es una raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$. Además, si $r, s \in R = \{i_1, ..., i_d\} \subset \{0, ..., n-1\}$, donde $r \neq s$, entonces $\omega^r \neq \omega^s$, ya que ω es raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$; se sigue que $\prod_{1 \le l < k \le \mathbf{d}} (\omega^{i_k} - \omega^{l_l}) \neq 0$. Por lo tanto, de (3.4) se tiene que $\mathbf{D} \neq 0$. Ahora, como $\mathbf{D} \neq 0$, se sigue que el sistema de ecuaciones en (3.3) tiene solución única, a saber, $(c_{i_1}, ..., c_{i_d}) = (0, ..., 0)$, lo cual es una contradicción ya que $\{i_1, ..., i_d\}$ es el soporte de $c(x)$. Por lo tanto, $\delta \le \mathbf{d}$, o bien, la distancia designada de $\mathcal{C}$ es menor o igual a su distancia mínima. ■

En la siguiente sección se presentarán algunos resultados básicos sobre el Polinomio de Mattson-Solomon, que serán de utilidad en la sección sobre la decodificación de códigos BCH, además, proporcionará lo necesario para dar una demostración alternativa del Teorema 3.12.

3.2. EL POLINOMIO DE MATTSON-SOLOMON

Definición 3.13 Sean $n \ge 2$ un entero tal que $mcd(n, q) = 1$ y ω una raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$. Entonces para cada $\mathbf{v} = (v_0, ..., v_{n-1}) \in \mathbb{F}_q^n$, consideramos el polinomio $v(x) = \sum_{j=0}^{n-1} v_j x^j \in \mathbb{F}_q[x]$ y definimos el polinomio de Mattson-Solomon $V_{ms}(x)$ de $\mathbf{v}$ como sigue:

$$V_{ms}(x) = \sum_{j=1}^{n} v(\omega^j) x^{n-j}.$$

Obsérvese que si $\omega \in \mathbb{F}_{q^k}$, entonces $V_{mx}(x)$ es un polinomio sobre $\mathbb{F}_{q^k}$. El polinomio de Mattson-Solomon puede depender de la elección especifica de ω, pero pensamos a ω fijo y se suprime esta dependencia en la notación, es decir, dado ω raíz n-ésima primitiva de la unidad, se considerara solo el polinomio de Mattson-Solomon respecto a ω.

Lema 3.14 Sean $n \ge 2$ un entero tal que $mcd(n, q) = 1$ y ω una raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$. Entonces para un entero i tal que $|i| \in \{0, ..., n-1\}$, se cumple que

$$\sum_{j=0}^{n-1}(\omega^i)^j = n\delta_{i,0}\ ,$$

donde $\delta_{i,0}$ es la delta de Kronecker.

Demostración. La demostración se hará en tres casos:
(1) Si $i = 0$ entonces $\sum_{j=0}^{n-1}(\omega^i)^j = \sum_{j=0}^{n-1}1^j = \sum_{j=0}^{n-1}1 = n = n\cdot 1 = n\delta_{i,0}$.

(2) Si $0 < i \leq n-1$, se tiene que $1-\omega^i \neq 0$, o bien, $1 \neq \omega^i$ ya que ω es una raíz n-ésima primitiva de la unidad. Entonces, $0 = 1-(\omega^i)^n = (1-\omega^i)(\sum_{j=0}^{n-1}(\omega^i)^j)$, y por lo mencionado anteriormente, se sigue que $\sum_{j=0}^{n-1}(\omega^i)^j = 0 = n\cdot 0 = 0\delta_{i,0}$.

(3) Si $-(n-1) \leq i < 0$, entonces existe $0 < k \leq n-1$ tal que $i = -k$. Ahora, ω^{-1} también es raíz n-ésima primitiva de la unidad, así que por el inciso (2), $\sum_{j=0}^{n-1}((\omega^{-1})^k)^j = n\delta_{k,0}$, como $(\omega^{-1})^k = \omega^{-k} = \omega^i$ y $\delta_{k,0} = \delta_{-k,0}$, se sigue que $\sum_{j=0}^{n-1}(\omega^i)^j = n\delta_{i,0}$. ■

Dado $\mathbf{v} \in \mathbb{F}_q^n$, el polinomio de Mattson-Solomon se define en base a las coordenadas de $\mathbf{v}$. A la inversa, si tenemos el polinomio de Mattson-Solomon de $\mathbf{v}$, podemos recuperar las coordenadas de $\mathbf{v}$ en la siguiente forma.

Teorema 3.15 Sean $n \geq 2$ un entero tal que $mcd(n,q) = 1$ y ω una raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$. Si $\mathbf{v} = (v_0, ..., v_{n-1}) \in \mathbb{F}_q^n$, entonces

$$v_i = n^{-1}V_{ms}(\omega^i) \text{ para } i = 0, ..., n-1,$$

donde n^{-1} es el inverso multiplicativo de n (la suma de n-veces 1) considerado como un elemento del subcampo primo de $\mathbb{F}_q$.

Demostración. Para $i = 0, ..., n-1$, tenemos que

$$V_{ms}(\omega^i) = \sum_{j=1}^{n} v(\omega^j)\omega^{i(n-j)} = \sum_{j=1}^{n} v(\omega^j)\omega^{in}\omega^{-ij} = \sum_{j=1}^{n} v(\omega^j)\omega^{-ij} \overset{*}{=} \sum_{j=0}^{n-1} v(\omega^j)\omega^{-ij} =$$
$$\sum_{j=0}^{n-1}\sum_{k=0}^{n-1} v_k\omega^{jk}\omega^{-ij} = \sum_{k=0}^{n-1} v_k \sum_{j=0}^{n-1}\omega^{j(k-i)} \overset{**}{=} \sum_{k=0}^{n-1} v_k n\delta_{k,i} = nv_i.$$

La igualdad resaltada con * se debe a que $v(\omega^0)\omega^{-i\cdot 0} = v(1)\omega^0 = v(1)\cdot 1 = v(1)\cdot 1^{-i} = v(\omega^n)(\omega^n)^{-i} = v(\omega^n)\omega^{-in}$, pues ω es raíz n-ésima de la unidad. La igualdad resaltada con ** se debe al Lema 3.14, pues $0 \leq k, i \leq n-1$ implica que $0 \leq |k-i| \leq n-1$ y por ende se puede aplicar el lema mencionado. Cabe señalar que $\delta_{k-i,0} = \delta_{k,i}$.

■

Del Teorema 3.15 se desprende inmediatamente el siguiente resultado.

Corolario 3.16 (i) Sean $n \geq 2$ un entero tal que $mcd(n, q) = 1$ y ω una raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$. Si $\mathbf{v} = (v_0, v_1, ..., v_{n-1}) \in \mathbb{F}_q^n$ y consideramos $v(x) = \sum_{j=0}^{n-1} v_j x^j \in \mathbb{F}_q[x]$, entonces

$$v(x) = n^{-1} \sum_{j=0}^{n-1} V_{ms}(\omega^j) x^j.$$

donde n^{-1} es el inverso multiplicativo de n (la suma de n-veces 1) considerado como un elemento del subcampo primo de $\mathbb{F}_q$.

(ii) El peso $wt(\mathbf{v})$ de $\mathbf{v}$ es igual a $n - s$, donde s es el número de raíces de $V_{ms}(x)$ que son raíces n-ésimas de la unidad.

(iii) El peso $wt(\mathbf{v})$ de $\mathbf{v}$ es mayor o igual a $n - grad(V_{ms}(x))$.

Demostración. (i) Por el Teorema 3.15, se tiene lo siguiente:

$$v(x) = \sum_{j=0}^{n-1} v_j x^j = \sum_{j=0}^{n-1} (n^{-1} V_{ms}(\omega^j)) x^j = n^{-1} \sum_{j=0}^{n-1} V_{ms}(\omega^j) x^j.$$

(ii) Como ω es una raíz n-ésima primitiva de la unidad, entonces $\omega^0, \omega^1, ..., \omega^{n-1}$ son todas las raíces n-ésimas de la unidad, así que el número de raíces n-ésimas de la unidad que son raíces de $V_{ms}(x)$ es igual a $s = |\{j : V_{ms}(\omega^j) = 0 \text{ y } 0 \leq j \leq n-1\}|$, y del Teorema 3.15 se sigue que $s = |\{j : v_j = 0 \text{ y } 0 \leq j \leq n-1\}|$. Por lo tanto, $wt(\mathbf{v}) = n - |\{j : v_j = 0 \text{ y } 0 \leq j \leq n-1\}| = n - s$.

(iii) Usando la notación del inciso anterior, se tiene que $s \leq grad(V_{ms}(x))$, pues $V_{ms}(x)$ puede tener raíces que no son raíces n-ésimas de la unidad. Por lo tanto, $n - grad(V_{ms}(x)) \leq n - s = wt(\mathbf{v})$. ■

Antes, en el Teorema 3.12 se demostró de para un código BCH su distancia mínima es mayor o igual a su distancia designada. A continuación, usando resultados sobre el Polinomio de Mattson-Solomon, daremos una demostración alternativa del teorema mencionado.

Teorema 3.17 (La cota BCH) Sean $n \geq 2$ un entero tal que $mcd(n, q) = 1$ y ω una raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$. Consideremos $\mathcal{C} = \mathfrak{B}_q(n, \delta, \omega, b)$ un código q-ario BCH, entonces $\delta \leq d(\mathcal{C})$, es decir, $\mathcal{C}$ tiene distancia mínima mayor o igual a su distancia designada δ.

Demostración. Sean $c(x) \in \mathcal{C}$ una palabra-código distinta de cero y $g(x)$ el polinomio generador de $\mathcal{C}$. Como $c(x) \neq \mathbf{0}$ y $\text{grad}(c(x)) \leq n - 1$, se tiene que no todos los n distintos elementos ω^i, $i = 0, 1, ..., n-1$, pueden ser raíces de $c(x)$. Se sigue que $C_{ms}(x)$ es distinto de cero. De los parámetros de $\mathcal{C}$, se cumple que $g(\omega^i) = 0$ para $i = b, b+1, ..., b+\delta-2$. Como $g(x)|c(x)$, se tiene que $c(\omega^i) = 0$ para $i = b, b+1+..., b+\delta-2$. Teniendo en consideración esto último, se analizan las posibles formas de $C_{ms}(x)$ en función del orden de ω^i, $i = b, ..., b+\delta-2$ respecto al conjunto ordenado de todas las raíces n-ésimas de la unidad $\{\omega^0, \omega^1, ..., \omega^{n-1}\}$.

Caso 1. $b + \delta - 2 \leq n - 1$.
En este caso tenemos 3 subcasos:

(I). Si $b = 0$, entonces los elementos ω^i, $i = 0, 1, ..., \delta-2$, tienen el siguiente orden dentro de las raíces n-ésimas de la unidad: $\underline{\omega^0, ..., \omega^{\delta-2}}, \omega^{\delta-1}, ..., \omega^{n-1}$.

Antes en la prueba, se mostró que $c(\omega^i) = 0$ para $i = b, ..., b+\delta-2$, entonces

$$C_{ms}(x) = c(\omega^{\delta-1})x^{n-\delta+1} + ... + c(\omega^{n-1})x.$$

Si $p(x) := x^{-1}C_{ms}(x) = c(\omega^{\delta-1})x^{n-\delta} + ... + c(\omega^{n-1})$, entonces $p(\omega^j) = 0$ si y sólo si $C_{ms}(\omega^j) = 0$ (si $p(\omega^j) = 0$ entonces $(\omega^j)^{-1}C_{mx}(\omega^j) = 0$, donde $\omega^{-j} \neq 0$ ya que $\omega \neq 0$, de ahí que $C_{ms}(x) = 0$, la otra implicación es inmediata por la definición de $p(x)$). Entonces, toda raíz n-ésima de la unidad que sea raíz de $p(x)$ es también raíz de $C_{ms}(x)$, y viceversa. De ahí que $C_{ms}(x)$ tiene a lo más $n-\delta$ raíces que son raíces n-ésimas de la unidad.

(II). Si $b = 1$, entonces dentro de las raíces n-ésimas de la unidad, los elementos ω^i, $i = 1, ..., b+\delta-2 = \delta-1$, tienen el siguiente orden: $\omega^0, \underline{\omega^1, ..., \omega^{\delta-1}}, \omega^\delta, ..., \omega^{n-1}$. Ahora, en este caso,

$$C_{ms}(x) = c(\omega^\delta)x^{n-\delta} + ... + c(\omega^{n-1})x + c(\omega^n),$$

y en consecuencia, $C_{ms}(x)$ tiene lo más $n-\delta$ raíces que son raíces n-ésimas de la unidad.

(III) Si $b > 1$, entonces el orden de los elementos ω^i, $i = b, b+1, ..., b+\delta-2$ en las n raíces n-ésimas de la unidad es el siguiente: $\omega^0, \omega^1, ..., \omega^{b-1}, \underline{\omega^b, \omega^{b+1}, ..., \omega^{b+\delta-2}}, \omega^{b+\delta-1}, ..., \omega^{n-1}$. Entonces, en este caso,

$$C_{ms}(x) = c(\omega)x^{n-1} + c(\omega^2)x^{n-2}... + c(\omega^{b-1})x^{n-b+1} + c(\omega^{b+\delta-1})x^{n-b-\delta+1} + ... + c(\omega^n).$$

Multiplicando la igualdad anterior por x^{b-1} y reordenando se tiene que:
$$\begin{aligned} x^{b-1}C_{ms} &= c(\omega)x^{n+b-2} + c(\omega^2)x^{n+b-3} + ... + c(\omega^{b-1})x^n + c(\omega^{b+\delta-1})x^{n-\delta} + ... + c(\omega^n)x^{b-1} \\ &= x^n[c(\omega)x^{b-2} + c(\omega)x^{b-3} + ... + c(\omega^{b-1})] + \{c(\omega^{b+\delta-1})x^{n-\delta} + ... + c(\omega^n)x^{b-1}\}. \end{aligned}$$
Tomando a $p(x)$ como el polinomio entre corchetes y a $q(x)$ como el polinomio entre llaves, se tiene que

$$x^{b-1}C_{ms}(x) = x^n p(x) + q(x) = (x^n - 1)p(x) + p(x) + q(x).$$

De la anterior igualdad, se sigue que si $C_{ms}(\omega^j) = 0$, entonces $0 = ((\omega^j)^n - 1)p(\omega^j) + p(\omega^j) + q(\omega^j) = (1-1)p(\omega^j) + p(\omega^j) + q(\omega^j) = p(\omega^j) + q(\omega^j)$. A la inversa, si $p(\omega^j) + q(\omega^j) = 0$ entonces $0 = p(\omega^j)+q(\omega^j) = (1-1)p(\omega^j)+p(\omega^j)+q(\omega^j) = ((\omega^j)^n-1)p(\omega^j)+p(\omega^j)+q(\omega^j) = \omega^{b-1}C_{ms}(\omega^j)$; como $\omega \neq 0$, se sigue que $\omega^{b-1} \neq 0$, y en consecuencia $C_{ms}(\omega^j) = 0$. Se ha mostrado que ω^j es una raíz de $C_{ms}(x)$ si y sólo si es una raíz del polinomio $p(x)+q(x)$, el cual tiene grado a lo más $n-\delta$. Entonces $C_{ms}(x)$ tiene a lo más $n-\delta$ raíces n-ésimas de la unidad entre sus raíces.

Caso 2. $b + \delta - 2 > n - 1$.

Ahora, los elementos ω^i, $i = b, b+1, ..., b+\delta-2$, tienen el siguiente orden entre las n raíces n-ésimas de la unidad: $\underline{\omega^0 = \omega^n, \omega^1 = \omega^{n+1}, ..., \omega^k = \omega^{b+\delta-2}}, \omega^{k+1}, ..., \omega^{b-1}, \underline{\omega^b, \omega^{b+1}, ..., \omega^{n-1}}$, donde $k = (b+\delta-2)(\text{mód} n)$ y $k < b-1$ (ya que ω^{b-1} no puede ser raíz de $g(x)$, según los parámetros de $\mathcal{C}$). Entonces, en este caso se tiene lo siguiente:

$$C_{ms}(x) = c(\omega^{k+1})x^{n-k-1} + c(\omega^{k+2})x^{n-k-2} + ... + c(\omega^{b-1})x^{n-b+1}.$$

Se sigue que $$\begin{aligned} x^{b-1}C_{ms}(x) &= c(\omega^{k+1})x^{n+b-k-2} + c(\omega^{k+2})x^{n+b-k-3} + ... + c(\omega^{b-1})^n \\ &= x^n[c(\omega^{k+1})x^{b-k-2} + c(\omega^{k+2})x^{b-k-3} + ... + c(\omega^{b-1})]. \end{aligned}$$

Sea $p(x) = c(\omega^{k+1})x^{b-k-2} + c(\omega^{k+2})x^{b-k-3} + ... + c(\omega^{b-1})$. Entonces $x^{b-1}C_{ms}(x) = x^n p(x)$, y $p(\omega^j) = 0$ si y sólo si $C_{ms}(\omega^j) = 0$ (si $p(\omega^j) = 0$ entonces $0 = (\omega^j)^n p(\omega^j) = (\omega^j)^{b-1}C_{ms}(\omega^j)$, es decir, $\omega^{j(b-1)}C_{ms}(\omega^j) = 0$, lo cual implica que $C_{ms}(\omega^j) = 0$, ya que $\omega \neq 0$. La otra implicación, se tiene por la forma de $p(x)$). Entonces, toda raíz n-ésima de la unidad que es raíz de $p(x)$ es también raíz de $C_{ms}(x)$, y viceversa. Como $n = (b-1-k)+(\delta-1)$, entonces $n-\delta = b-k-2$. Así, $p(x)$ tiene a lo más grado $n-\delta$, y en consecuencia $C_{ms}(x)$ tiene a lo más $n-\delta$ raíces n-ésimas de la unidad entre sus raíces. En cualquier caso (1 ó 2), vemos que $C_{ms}(x)$ tiene a lo más $n-\delta$ raíces que son raíces n-ésimas de la unidad, es decir, $s := |\{\omega^j : 0 \leq j \leq n-1 \text{ y } C_{ms}(\omega^j) = 0\}| \leq n-\delta$.

Por el Corolario 3.16 (ii), se tiene que $\delta = n-(n-\delta) \leq n-s = wt(c(x))$. Por lo tanto, la distancia mínima de $\mathcal{C}$ es mayor o igual a δ. ■

En la siguiente sección se plantea el problema de decodificación de códigos BCH, para posteriormente proponer una solución.

3.3. DECODIFICACIÓN DE CÓDIGOS BCH

Sean $n \geq 2$ un entero tal que $mcd(n,q) = 1$, ω una raíz n-ésima primitiva de la unidad sobre $\mathbb{F}_q$ y $\mathbf{v} = (v_0, v_1, ..., v_{n-1}) \in \mathbb{F}_q^n$. Consideremos $\widehat{\mathbf{v}} = (\widehat{v}_0, \widehat{v}_1, ..., \widehat{v}_{n-1})$ su Transformada de Fourier Discreta (TFD), cuyas componentes se definen como sigue:

$$\widehat{v}_i = \sum_{j=0}^{n-1} v_j \omega^{ij}, i = 0, 1, ..., n-1 \tag{3.5}$$

Si interpretamos las componentes de $\mathbf{v}$ y $\widehat{\mathbf{v}}$ como coeficientes de polinomios, entonces $\mathbf{v}(x) = \sum_{j=0}^{n-1} v_j x^j$ y $\widehat{\mathbf{v}}(x) = \sum_{i=1}^{n-1} \widehat{v}_i x^i$ se relacionan de la siguiente manera:

$$\widehat{v}_i = \mathbf{v}(\omega^i),\ i = 0, 1, .., n-1, \text{ y en consecuencia, } \widehat{\mathbf{v}}(x) = \sum_{i=0}^{n-1} \mathbf{v}(\omega^i) x^i. \tag{3.6}$$

Note que $\widehat{\mathbf{v}}(x) = \sum_{i=1}^{n-1} \mathbf{v}((\omega^{-1})^{-i}) x^i$, que es el polinomio de Mattson-Solomon de $\mathbf{v}$ pero respecto a la raíz n-ésima primitiva ω^{-1}. Entonces, del Teorema 3.15 se tiene lo siguiente:

$$v_i = n^{-1} \widehat{\mathbf{v}}(\omega^{-i}) \tag{3.7}$$

Si multiplicamos la i-ésima componente de $\mathbf{v}$ por $\omega^{\mu i}$, $i = 0, 1, ..., n-1$, definimos un nuevo vector

$$\mathbf{v}_\mu = (v_0, v_1 \omega^\mu, ..., v_{n-1} \omega^{\mu(n-1)}) \tag{3.8}$$

entonces su TFD es

$$\widehat{\mathbf{v}_\mu} = (\widehat{v}_\mu, \widehat{v}_{\mu+1}, ..., \widehat{v}_{\mu+n-1}) \tag{3.9}$$

donde en (3.9) los subíndices se toman módulo n. Para probar la expresión en (3.9), sean $j \in \{0, ..., n-1\}$ y $k = (\mu + j)(\text{mód } n)$, entonces de (3.6) se tiene que la entrada j-ésima de $\widehat{\mathbf{v}_\mu}$ cumple lo siguiente:

$$\widehat{v_\mu}_j = \mathbf{v}_\mu(\omega^j) = \sum_{i=0}^{n-1} (v_i \omega^{\mu i})(\omega^j)^i = \sum_{i=0}^{n-1} v_i \omega^{\mu i + ji} = \sum_{i=0}^{n-1} v_i \omega^{(\mu+j)i} = \sum_{i=0}^{n-1} v_i \omega^{ki} = \mathbf{v}(\omega^k) = \widehat{v}_k.$$

Así, la j-ésima entrada de $\widehat{\mathbf{v}_\mu}$ es $\widehat{v}_k$, tal como se deseaba, pues recordemos que $k = (\mu + j)(\text{mód } n)$.

Ahora se definen los siguientes polinomios asociados a $\mathbf{v}$. El polinomio localizador de $\mathbf{v}$ es

$$\sigma_{\mathbf{v}}(x) = \prod_{i \in Sop(\mathbf{v})} (1 - \omega^i x). \tag{3.10}$$

Para cada valor de $i \in \text{Sop}(\mathbf{v})$, se define también el i-ésimo polinomio localizador perforado $\sigma_{\mathbf{v}}^{(i)}(x)$:

$$\sigma_{\mathbf{v}}^{(i)}(x) = \sigma_{\mathbf{v}}(x)/(1 - \omega^i x) = \prod_{\substack{j \in Sop(\mathbf{v}) \\ j \neq i}} (1 - \omega^j x) \tag{3.11}$$

Finalmente, definimos el polinomio evaluador de $\mathbf{v}$ como

$$\lambda_{\mathbf{v}}(x) = \sum_{i \in Sop(\mathbf{v})} v_i \sigma_{\mathbf{v}}^{(i)}(x) \tag{3.12}$$

Con la definición de los anteriores polinomios se tiene el siguiente lema.

Lema 3.18 $mcd(\sigma_{\mathbf{v}}(x), \lambda_{\mathbf{v}}(x)) = 1$.

Demostración. Para exhibir que $\sigma_v(x)$ y $\lambda_{\mathbf{v}}(x)$ son primos relativos, es suficiente observar que ellos no tienen raíces en común. En caso contrario, existe $i \in Sop(\mathbf{v})$ tal que $\lambda_v(\omega^{-i}) = 0$ (ya que las raíces de σ_v(x) son precisamente los elementos ω^{-i}, con $i \in Sop(\mathbf{v})$). Obsérvese que si $j \in Sop(\mathbf{v})$ y $j \neq i$, entonces

$$\sigma_{\mathbf{v}}^{(j)}(\omega^{-i}) = \prod_{\substack{k \in Sop(\mathbf{v}) \\ k \neq j}} (1 - \omega^k \omega^{-i}) = (1 - \omega^i \omega^{-i}) \prod_{\substack{k \in Sop(\mathbf{v}) \\ k \neq j, k \neq i}} (1 - \omega^k \omega^{-i}) = 0$$

De lo anterior, se tiene que $0 = \lambda_{\mathbf{v}}(\omega^{-i}) = \sum_{j \in Sop(\mathbf{v})} v_j \sigma_{\mathbf{v}}^{(j)}(\omega^{-i}) = v_i \sigma_{\mathbf{v}}^{(i)}(\omega^{-i})$, es decir, $v_i \sigma_{\mathbf{v}}^{(i)}(\omega^{-i}) = 0$ donde $v_i \neq 0$, pues $i \in Sop(\mathbf{v})$. Entonces $\sigma_{\mathbf{v}}^{(i)}(\omega^{-i}) = 0$, pero si $k \in Sop(\mathbf{v})$ y $k \neq i$ se sigue que $\omega^k \neq \omega^i$, y en consecuencia $1 - \omega^k \omega^{-i} \neq 0$, lo cual implica que $0 \neq \prod_{\substack{k \in Sop(\mathbf{v}) \\ k \neq i}} (1 - \omega^k \omega^{-i}) = \sigma_{\mathbf{v}}^{(i)}(\omega^{-i})$, con lo cual hemos llegado a una contradicción.

Por lo tanto $\sigma_{\mathbf{v}}(x)$ y $\lambda_{\mathbf{v}}(x)$ no tienen raíces en común, o bien, tales polinomios son primos relativos. ■

Sea $\mathbf{v}$ como hasta ahora, entonces se presenta un importante resultado que involucra a polinomios asociados a $\mathbf{v}$, más precisamente al polinomio localizador, el polinomio evaluador y a la TFD de $\mathbf{v}$ en lo que se conoce como la "ecuación clave".

Teorema 3.19 (La ecuación clave) Los polinomios $\widehat{\mathbf{v}}(x)$, $\sigma_{\mathbf{v}}(x)$ y $\lambda_{\mathbf{v}}(x)$ satisfacen

$$\sigma_{\mathbf{v}}(x)\widehat{\mathbf{v}}(x) = \lambda_{\mathbf{v}}(x)(1 - x^n). \tag{3.13}$$

Demostración. De (3.6) tenemos que

$$\widehat{\mathbf{v}}(x) = \sum_{j=0}^{n-1} (\sum_{i=0}^{n-1} v_i \omega^{ij}) x^j = \sum_{j=0}^{n-1} (\sum_{i \in Sop(\mathbf{v})} v_i \omega^{ij}) x^j = \sum_{i \in Sop(\mathbf{v})} v_i \sum_{j=0}^{n-1} \omega^{ij} x^j \tag{3.14}$$

De acuerdo a (3.11), $\sigma_{\mathbf{v}}(x) = \sigma_{\mathbf{v}}^{(i)}(x)(1 - \omega^i x)$ para cada $i \in Sop(\mathbf{v})$, y de (3.14) se sigue que

$$\begin{aligned}
\sigma_{\mathbf{v}}(x)\widehat{\mathbf{v}}(x) &= \sum_{i \in Sop(\mathbf{v})} v_i \sigma_{\mathbf{v}}^{(i)}(x)(1 - \omega^i x) \sum_{j=0}^{n-1} \omega^{ij} x^j \\
&= \sum_{i \in Sop(\mathbf{v})} v_i \sigma_{\mathbf{v}}^{(i)}(x)(\sum_{j=0}^{n-1} \omega^{ij} x^j - \sum_{j=0}^{n-1} \omega^{i(j+1)} x^{j+1}) \\
&= \sum_{i \in Sop(\mathbf{v})} v_i \sigma_{\mathbf{v}}^{(i)}(x)(\sum_{j=0}^{n-1} \omega^{ij} x^j - \sum_{j=1}^{n} \omega^{ij} x^j)
\end{aligned}$$

$$= \sum_{i \in Sop(\mathbf{v})} v_i \sigma_{\mathbf{v}}^{(i)}(x)(\omega^{i \cdot 0} x^0 - \omega^{i \cdot n} x^n)$$

$$= \sum_{i \in Sop(\mathbf{v})} v_i \sigma_{\mathbf{v}}^{(i)}(x)(1 - x^n)$$

$$= \lambda_{\mathbf{v}}(x)(1 - x^n) \ . \qquad \blacksquare$$

Ejemplo 3.20 Ahora ilustramos estos conceptos en el campo $\mathbb{F}_{2^4}$, en el cual los elementos distintos de cero se representan por potencias de un elemento primitivo ω que es raíz de $x^4 + x + 1 \in \mathbb{F}_2[x]$. Note que ω es una raíz 15-ésima de la unidad sobre $\mathbb{F}_{16}$, pues los elementos distintos de cero de $\mathbb{F}_{2^4}$ forman un grupo multiplicativo de orden 15, por lo que estos elementos son raíces de $x^{15} - 1 \in \mathbb{F}_{2^4}[x]$, y son generados por ω. Consideremos el vector

$$\mathbf{v} = (0, 0, \omega^2, 0, 0, 0, 0, \omega^7, 0, 0, 0, 0, 0, 0, 0) \in \mathbb{F}_{2^4}^{15}[x].$$

Para los cálculos en este ejemplo, a continuación se dan las representaciones en notación aditiva y multiplicativa de los elementos de $\mathbb{F}_{2^4}$ y la tabla de Cayley correspondiente a $(\mathbb{F}_{2^4}, +)$.

ω^i	Notación Aditiva
ω^0	1
ω^1	ω
ω^2	ω^2
ω^3	ω^3
ω^4	$\omega + 1$
ω^5	$\omega^2 + \omega$
ω^6	$\omega^3 + \omega^2$
ω^7	$\omega^3 + \omega + 1$
ω^8	$\omega^2 + 1$
ω^9	$\omega^3 + \omega$
ω^{10}	$\omega^2 + \omega + 1$
ω^{11}	$\omega^3 + \omega^2 + \omega$
ω^{12}	$\omega^3 + \omega^2 + \omega + 1$
ω^{13}	$\omega^3 + \omega^2 + 1$
ω^{14}	$\omega^3 + 1$

$+$	0	1	ω	ω^2	ω^3	ω^4	ω^5	ω^6	ω^7	ω^8	ω^9	ω^{10}	ω^{11}	ω^{12}	ω^{13}	ω^{14}
0	0	1	ω	ω^2	ω^3	ω^4	ω^5	ω^6	ω^7	ω^8	ω^9	ω^{10}	ω^{11}	ω^{12}	ω^{13}	ω^{14}
1	1	0	ω^4	ω^8	ω^{14}	ω	ω^{10}	ω^{13}	ω^9	ω^2	ω^7	ω^5	ω^{12}	ω^{11}	ω^6	ω^3
ω	ω	ω^4	0	ω^5	ω^9	1	ω^2	ω^{11}	ω^{14}	ω^{10}	ω^3	ω^8	ω^6	ω^{13}	ω^{12}	ω^7
ω^2	ω^2	ω^8	ω^5	0	ω^6	ω^{10}	ω	ω^3	ω^{12}	1	ω^{11}	ω^4	ω^9	ω^7	ω^{14}	ω^{13}
ω^3	ω^3	ω^{14}	ω^9	ω^6	0	ω^7	ω^{11}	ω^2	ω^4	ω^{13}	ω	ω^{12}	ω^5	ω^{10}	ω^8	1
ω^4	ω^4	ω	1	ω^{10}	ω^7	0	ω^8	ω^{12}	ω^3	ω^5	ω^{14}	ω^2	ω^{13}	ω^6	ω^{11}	ω^9
ω^5	ω^5	ω^{10}	ω^2	ω	ω^{11}	ω^8	0	ω^9	ω^{13}	ω^4	ω^6	1	ω^3	ω^{14}	ω^7	ω^{12}
ω^6	ω^6	ω^{13}	ω^{11}	ω^3	ω^2	ω^{12}	ω^9	0	ω^{10}	ω^{14}	ω^5	ω^7	ω	ω^4	1	ω^8
ω^7	ω^7	ω^9	ω^{14}	ω^{12}	ω^4	ω^3	ω^{13}	ω^{10}	0	ω^{11}	1	ω^6	ω^8	ω^2	ω^5	ω
ω^8	ω^8	ω^2	ω^{10}	1	ω^{13}	ω^5	ω^4	ω^{14}	ω^{11}	0	ω^{12}	ω	ω^7	ω^9	ω^3	ω^6
ω^9	ω^9	ω^7	ω^3	ω^{11}	ω	ω^{14}	ω^6	ω^5	1	ω^{12}	0	ω^{13}	ω^2	ω^8	ω^{10}	ω^4
ω^{10}	ω^{10}	ω^5	ω^8	ω^4	ω^{12}	ω^2	1	ω^7	ω^6	ω	ω^{13}	0	ω^{14}	ω^3	ω^9	ω^{11}
ω^{11}	ω^{11}	ω^{12}	ω^6	ω^9	ω^5	ω^{13}	ω^3	ω	ω^8	ω^7	ω^2	ω^{14}	0	1	ω^4	ω^{10}
ω^{12}	ω^{12}	ω^{11}	ω^{13}	ω^7	ω^{10}	ω^6	ω^{14}	ω^4	ω^2	ω^9	ω^8	ω^3	1	0	ω	ω^5
ω^{13}	ω^{13}	ω^6	ω^{12}	ω^{14}	ω^8	ω^{11}	ω^7	1	ω^5	ω^3	ω^{10}	ω^9	ω^4	ω	0	ω^2
ω^{14}	ω^{14}	ω^3	ω^7	ω^{13}	1	ω^9	ω^{12}	ω^8	ω	ω^6	ω^4	ω^{11}	ω^{10}	ω^5	ω^2	0

Entonces, el polinomio $\mathbf{v}(x) = \sum_{i=0}^{n-1} v_i x^i$ es

$$\mathbf{v}(x) = \omega^2 x^2 + \omega^7 x^7$$

De (3.6) podemos calcular las componentes de la TFD de $\mathbf{v}$

$\widehat{v}_0 = \mathbf{v}(\omega^0) = \omega^2 + \omega^7 = \omega^{12}$
$\widehat{v}_1 = \mathbf{v}(\omega^1) = \omega^4 + \omega^{14} = \omega^9$
$\widehat{v}_2 = \mathbf{v}(\omega^2) = \omega^6 + \omega^{21} = \omega^6 + \omega^6 = 0$
$\widehat{v}_3 = \mathbf{v}(\omega^3) = \omega^8 + \omega^{28} = \omega^8 + \omega^{13} = \omega^3$
$\widehat{v}_4 = \mathbf{v}(\omega^4) = \omega^{10} + \omega^{35} = \omega^{10} + \omega^5 = 1$
$\widehat{v}_5 = \mathbf{v}(\omega^5) = \omega^{12} + \omega^{42} = \omega^{12} + \omega^{12} = 0$
$\widehat{v}_6 = \mathbf{v}(\omega^6) = \omega^{14} + \omega^{49} = \omega^{14} + \omega^4 = \omega^9$
$\widehat{v}_7 = \mathbf{v}(\omega^7) = \omega^{16} + \omega^{56} = \omega + \omega^{11} = \omega^6$
$\widehat{v}_8 = \mathbf{v}(\omega^8) = \omega^{18} + \omega^{63} = \omega^3 + \omega^3 = 0$
$\widehat{v}_9 = \mathbf{v}(\omega^9) = \omega^{20} + \omega^{70} = \omega^5 + \omega^{10} = 1$
$\widehat{v}_{10} = \mathbf{v}(\omega^{10}) = \omega^{22} + \omega^{77} = \omega^7 + \omega^2 = \omega^{12}$
$\widehat{v}_{11} = \mathbf{v}(\omega^{11}) = \omega^{24} + \omega^{84} = \omega^9 + \omega^9 = 0$
$\widehat{v}_{12} = \mathbf{v}(\omega^{12}) = \omega^{26} + \omega^{91} = \omega^{11} + \omega = \omega^6$
$\widehat{v}_{13} = \mathbf{v}(\omega^{13}) = \omega^{28} + \omega^{98} = \omega^{13} + \omega^8 = \omega^3$
$\widehat{v}_{14} = \mathbf{v}(\omega^{14}) = \omega^{30} + \omega^{105} = \omega^0 + \omega^0 = 0.$

De lo anterior, se tiene que $\widehat{\mathbf{v}} = (\omega^{12}, \omega^9, 0, \omega^3, 1, 0, \omega^9, \omega^6, 0, 1, \omega^{12}, 0, \omega^6, \omega^3, 0)$.

También
$$\begin{aligned}\widehat{\mathbf{v}}(x) &= \omega^{12} + \omega^9 x + \omega^3 x^3 + x^4 + \omega^9 x^6 + \omega^6 x^7 + x^9 + \omega^{12} x^{10} + \omega^6 x^{12} + \omega^3 x^{13} \\ &= (\omega^{12} + \omega^9 x)(1 + \omega^6 x^3 + \omega^{12} x^6 + \omega^3 x^9 + \omega^9 x^{12}) \\ &= (\omega^{12} + \omega^9 x)\tfrac{1+x^{15}}{1+\omega^6 x^3} \\ &= \omega^{12}(1 + \omega^{12} x)\tfrac{1+x^{15}}{1+\omega^6 x^3} \\ &= \omega^{12}(1 + \omega^{12} x)\tfrac{1+x^{15}}{(1+\omega^{12}x)(1+\omega^{12}x+\omega^9 x^2)}\end{aligned}$$

$= \omega^{12}\frac{1+x^{15}}{1+\omega^{12}x+\omega^{9}x^{2}}$.

El soporte de $\mathbf{v}$ es el conjunto $\{2,7\}$, así que el polinomio localizador de $\mathbf{v}$ es

$$\sigma_{\mathbf{v}}(x) = (1+\omega^{2}x)(1+\omega^{7}x) = 1+(\omega^{7}+\omega^{2})x+\omega^{9}x^{2} = 1+\omega^{12}x+\omega^{9}x^{2}.$$

Los polinomios $\sigma_{\mathbf{v}}^{(i)}$ definidos en (3.11) son los siguientes:

$$\sigma_{\mathbf{v}}^{(2)} = (1+\omega^{7}x),\ \sigma_{\mathbf{v}}^{(7)} = (1+\omega^{2}x).$$

El polinomio $\lambda_{\mathbf{v}}(x)$ definido en (3.12) es

$$\lambda_{\mathbf{v}}(x) = \omega^{2}(1+\omega^{7}x)+\omega^{7}(1+\omega^{2}x) = \omega^{2}+\omega^{7} = \omega^{12}.$$

Al ser $\lambda_{\mathbf{v}}(x)$ un polinomio constante diferente de cero, se tiene que $mcd(\lambda_{\mathbf{v}}(x),\sigma_{\mathbf{v}}(x)) = 1$, lo cual era de esperarse según el Lema 3.18. Ahora, $\sigma_{\mathbf{v}}(x)\widehat{\mathbf{v}}(x) = (1+\omega^{12}x+\omega^{9}x^{2})(\omega^{12}\frac{1+x^{15}}{1+\omega^{12}x+\omega^{9}x^{2}}) = \omega^{12}(1+x^{15}) = \lambda_{\mathbf{v}}(x)(1-x^{15})$ (recordar que $\mathbb{F}_{2^4}$ tiene característica 2). La anterior igualdad verifica la **ecuación clave** para los polinomios asociados al vector $\mathbf{v}$ de nuestro ejemplo.

Con los resultados previos sobre la **ecuación clave** y los polinomios involucrados, podemos iniciar la discusión sobre el problema de la decodificación de los códigos BCH. De manera más precisa, se analizara la decodificación en códigos binarios primitivos BCH en el sentido estricto.

Recordar que, de acuerdo al caso especial 2 de la Definición 3.1, cada vez que nos referimos a un código binario BCH primitivo, la longitud es de la forma 2^m-1 para algún $m \in \mathbb{N}$.

Sea $\mathcal{C} = \mathfrak{B}_2(n = 2^m-1, 2t+1, \alpha, 1)$ un código binario primitivo BCH en el sentido estricto (recordar de la Observación 3.4 que la distancia designada de un código binario primitivo BCH en el sentido estricto se puede asumir de la forma $2t+1$, para algún entero positivo t). Supongamos que $\mathbf{c} = (c_0, c_1, ..., c_{n-1}) \in \mathcal{C}$ es transmitida por un canal ruidoso, y que $\mathbf{r} = (r_0, r_1, ..., r_{n-1})$ es recibida. Ahora, definimos el patrón de error como $\mathbf{e} = (e_0, e_1, ..., e_{n-1}) = \mathbf{r}-\mathbf{c}$, donde además se supone que $wt(\mathbf{e}) \leq t$. El primer paso en la decodificación es calcular los síndromes $s_1, ..., s_{2t}$, definidos por

$$s_j = \sum_{i=0}^{n-1} r_i\alpha^{ij}, j = 1, 2, ..., 2t. \tag{3.15}$$

Como $\mathbf{r} = \mathbf{c}+\mathbf{e}$, y $\mathbf{c}$ es una palabra-código, se sigue que

$$s_j = \sum_{i=0}^{n-1} e_i\alpha^{ij}, j = 1, ..., 2t, \tag{3.16}$$

así que, como era de esperarse los síndromes dependen sólo del patrón de error y no de la palabra-código transmitida. Comparando (3.16) con (3.5), vemos que s_j es la j-ésima entrada de la TFD del patrón de error; en otras palabras, los síndromes nos permiten observar $2t$ componentes consecutivas (la primera, segunda,...,$2t$-ésima) de $\widehat{\mathbf{e}}$.

Ahora, definimos el patrón de error torcido $\mathbf{w}$ como sigue:

$$\mathbf{w} = (w_0, w_1, ..., w_{n-1}) = (e_0, e_1\alpha, e_2\alpha^2, ..., e_{n-1}\alpha^{n-1}), \tag{3.17}$$

se sigue de (3.8) que $\mathbf{w} = \mathbf{e}_\mu$ con $\mu = 1$, entonces de (3.9) tenemos que $(\widehat{w}_0, \widehat{w}_1, ..., \widehat{w}_{2t-1}) = (\widehat{e}_1, \widehat{e}_2, ..., \widehat{e}_{2t-1}, \widehat{e}_{2t})$, y del párrafo anterior se tiene que $(s_1, s_2..., s_{2t}) = (\widehat{e}_1, \widehat{e}_2, ..., \widehat{e}_{2t})$, de ahí que $(s_1, s_2..., s_{2t}) = (\widehat{w}_0, \widehat{w}_1, ..., \widehat{w}_{2t-1})$. La **ecuación clave** también aplica para el vector $\mathbf{w}$ definido en

(3.17); sin embargo, sólo conocemos los primeros coeficientes de $\widehat{\mathbf{w}}(x)$ (es decir, $\widehat{w}_0, \widehat{w}_1, ..., \widehat{w}_{2t-1}$), entonces consideramos el síndrome polinomial $\mathbf{s}(x) = \sum_{i=0}^{2t-1} \widehat{w}_i x^i$ $(= \sum_{i=1}^{2t} s_i x^{i-1})$ y nos enfocamos en la **ecuación clave** reducida mód x^{2t}:

$$\sigma(x)\mathbf{s}(x) \equiv \lambda(x) \pmod{x^{2t}}. \tag{3.18}$$

En (3.18) se ha omitido $\mathbf{w}$ en la notación los subíndices de $\sigma(x)$ y $\lambda(x)$, es decir, $\sigma(x)$ es el polinomio localizador de $\mathbf{w}$ y $\lambda(x)$ es el polinomio evaluador de $\mathbf{w}$. De (3.17) se sigue que $Sop(\mathbf{w}) = Sop(\mathbf{e})$, es decir, $Sop(\mathbf{w})$ es el conjunto de las ubicaciones del error. Por esta razón, el polinomio $\sigma(x)$ en (3.18) es llamado *polinomio localizador del error*. Similarmente $\lambda(x)$ es llamado *polinomio evaluador del error*.

Para justificar (3.18), obsérvese que

$$\mathbf{s}(x) \equiv \widehat{\mathbf{w}}(x) \pmod{x^{2t}}, \tag{3.19}$$

pues $x^{2t} | (-\sum_{i=2t}^{n-1} \widehat{w}_i x^i)$, donde $-\sum_{i=2t}^{n-1} \widehat{w}_i x^i = \mathbf{s}(x) - \widehat{\mathbf{w}}(x)$. Entonces, multiplicando $\sigma(x)$ en ambos lados de la congruencia en (3.18) se tiene lo siguiente:

$$\sigma(x)\mathbf{s}(x) \equiv \sigma(x)\widehat{\mathbf{w}}(x) \pmod{x^{2t}}. \tag{3.20}$$

Ahora, ya que $x^{2t} | x^n$ entonces $1 - x^n \equiv 1 \pmod{x^{2t}}$, de ahí que

$$\lambda(x)(1 - x^n) \equiv \lambda(x) \pmod{x^{2t}}. \tag{3.21}$$

De la ecuación en (3.13) aplicada a $\mathbf{w}$ y (3.21) se sigue:

$$\sigma(x)\widehat{\mathbf{w}}(x) \equiv \lambda(x) \pmod{x^{2t}}. \tag{3.22}$$

Finalmente, de (3.20) y (3.22) se sigue (3.18). La congruencia polinomial en (3.18) se conoce como **la ecuación clave BCH**.

Como se hizo notar, $Sop(\mathbf{w}) = Sop(\mathbf{e})$ y como estamos bajo el supuesto $wt(\mathbf{e}) \leq t$, se tiene que $|Sop(\mathbf{w})| = |Sop(\mathbf{e})| \leq t$. Se sigue que $grad(\sigma(x)) = grad(\prod_{k \in Sop(\mathbf{w})} (1 - \alpha^k x)) \leq t$, y para cada $i \in Sop(\mathbf{w})$: $grad(\sigma^{(i)}(x)) = grad(\prod_{\substack{j \in Sop(\mathbf{w} \\ j \neq i)}} (1 - \alpha^j x)) \leq t - 1$, lo cual implica que $grad(\lambda(x)) = grad(\sum_{i \in Sop(\mathbf{w})} w_i \sigma^{(i)}(x)) \leq t - 1$. Teniendo esto en consideración, se obtiene un teorema de unicidad respecto a la **ecuación clave BCH** (3.18).

Teorema 3.21 Sean $\mathbf{s}$(x) como en (3.18), y $\mathbf{u}(x)$, $\mathbf{v}(x) \in \mathbb{F}_{2^m}[x]$ distintos de cero tales que $grad(\mathbf{u}(x)) \leq t - 1$, $grad(\mathbf{v}(x)) \leq t$, $mcd(\mathbf{u}(x), \mathbf{v}(x)) = 1$ y

$$\mathbf{v}(x)\mathbf{s}(x) \equiv \mathbf{u}(x) \pmod{x^{2t}}, \tag{3.23}$$

entonces se cumple lo siguiente:

$$\sigma(x) = \beta\mathbf{v}(x),\ \lambda(x) = \beta\mathbf{u}(x) \tag{3.24}$$

para algún elemento distinto de cero $\beta \in \mathbb{F}_{2^m}$.

Demostración. Multiplicando en ambos lados de la congruencia (3.23) por $\sigma(x)$, se tiene que $\mathbf{v}(x)\sigma(x)\mathbf{s}(x) \equiv \sigma(x)\mathbf{u}(x) \pmod{x^{2t}}$, y de (3.18) se sigue que $\mathbf{v}(x)\sigma(x)\mathbf{s}(x) \equiv \mathbf{v}(x)\lambda(x) \pmod{x^{2t}}$; en consecuencia,

$$\mathbf{v}(x)\lambda(x) \equiv \sigma(x)\mathbf{u}(x) \pmod{x^{2t}}. \tag{3.25}$$

Como $grad(\mathbf{v}(x)\lambda(x)) = grad(\mathbf{v}(x)) + grad(\lambda(x)) \leq t + (t-1) = 2t-1$, y $grad(\sigma(x)\mathbf{u}(x)) = grad(\sigma(x)) + grad(\mathbf{u}(x)) \leq t + (t-1) = 2t-1$, entonces $grad(\mathbf{v}(x)\lambda(x) - \sigma(x)\mathbf{u}(x)) \leq 2t-1$ y de (3.25) se tiene que $x^{2t} | (\mathbf{v}(x)\lambda(x) - \sigma(x)\mathbf{u}(x))$, lo cual implica que

$$\mathbf{v}(x)\lambda(x) = \sigma(x)\mathbf{u}(x). \tag{3.26}$$

Ya que $\mathbf{u}(x) | \sigma(x)\mathbf{u}(x)$, es decir, $\mathbf{u}(x) | \mathbf{v}(x)\lambda(x)$, entonces por la hipótesis $(\mathbf{u}(x), \mathbf{v}(x)) = 1$ se tiene que $\mathbf{u}(x) | \lambda(x)$. También, $\lambda(x) | \mathbf{v}(x)\lambda(x)$, es decir, $\lambda(x) | \sigma(x)\mathbf{u}(x)$, y por el Lema 3.18 se tiene que $mcd(\lambda(x), \sigma(x)) = 1$, lo cual implica que $\mathbf{v(x)} | \mathbf{u}(x)$. Entonces $\mathbf{u}(x) | \lambda(x)$ y $\lambda(x) | \mathbf{u}(x)$, de ahí que $\lambda(x) = \beta_1 \mathbf{u}(x)$ para algún $0 \neq \beta_1 \in \mathbb{F}_{2^m}$. Similarmente se demuestra que $\mathbf{v}(x) | \sigma(x)$ y $\sigma(x) | \mathbf{v}(x)$, así que $\sigma(x) = \beta_2 \mathbf{v}(x)$ para algún $0 \neq \beta_2 \in \mathbb{F}_{2^m}$. Para concluir la demostración resta probar que $\beta_1 = \beta_2$. De (3.26) se obtiene la siguiente igualdad:

$$\mathbf{v}(x)(\beta_1 \mathbf{u}(x)) = (\beta_2 \mathbf{v}(x))\mathbf{u}(x),$$

lo cual implica que $(\beta_1 - \beta_2)(\mathbf{u}(x)\mathbf{v}(x)) = 0$; ya que $\mathbf{u}(x)\mathbf{v}(x) \neq 0$ (pues por hipótesis $\mathbf{u}(x) \neq 0 \neq \mathbf{v}(x)$) se sigue que $\beta_1 - \beta_2 = 0$, es decir, $\beta_1 = \beta_2$. ■

Ahora observe, que dado el síndrome polinomial $\mathbf{s}(x)$ de la palabra recibida $\mathbf{r}$, o equivalentemente, $\widehat{\mathbf{w}}(x) \pmod{x^{2t}}$, si de alguna manera se pudiera resolver la **ecuación clave BCH** (3.18) para los polinomios $\sigma(x)$ y $\lambda(x)$, podríamos recuperar el patrón de error $\mathbf{e}$, y por lo tanto la palabra-código transmitida $\mathbf{c} = \mathbf{r} - \mathbf{e}$. Esto se podría hacer calculando los n valores $\sigma(\alpha^{-i})$, $i = 0, 1, ..., n-1$, donde $\sigma(\alpha^{-i}) = 0$ indica que $i \in Sop(\mathbf{w}) = Sop(\mathbf{e})$; de esta manera podríamos identificar a $Sop(\mathbf{e})$, y en consecuencia a $\mathbf{e}$, ya que sus componentes distintas de cero son iguales a 1 por ser un vector binario.

En la siguiente sección veremos un algoritmo para obtener $\sigma(x)$ y $\lambda(x)$ de la **ecuación clave BCH** (3.18), bajo el supuesto que ha tenido hasta el momento, que es $wt(\mathbf{e}) \leq t$, es decir, se ha supuesto que $\mathcal{C}$ es un código t-corrector de errores.

3.4. EL ALGORITMO DE EUCLIDES PARA POLINOMIOS

Esta sección no trata directamente el problema de la decodificación de códigos BCH. No debe perderse de vista que nuestro objetivo es resolver la **ecuación clave BCH** (3.18) para $\sigma(x)$ y $\lambda(x)$, dado $\mathbf{s}(x)$, o equivalentemente, $\widehat{\mathbf{w}}(x)(\pmod{x^{2t}}$.

A través de esta sección $a(x)$ y $b(x)$ serán polinomios fijos sobre un campo $\mathbb{F}$, con $grad(a(x)) \geq grad(b(x))$. Después, para nuestro fin, $a(x)$ será reemplazado por x^{2t}, y $b(x)$ por el síndrome polinomial $\mathbf{s}(x)$.

El algoritmo de Euclides es un procedimiento recursivo para encontrar el máximo común divisor $d(x)$ de $a(x)$ y $b(x)$, y para encontrar un combinación de $a(x)$ y $b(x)$ igual a $d(x)$, es decir, una ecuación de la forma

$$\mathbf{u}(x)\mathbf{a}(x) + \mathbf{v}(x)\mathbf{b}(x) = \mathbf{d(x)}, \text{ para algunos } \mathbf{u}(x), \mathbf{v}(x) \in \mathbb{F}[x]. \tag{3.27}$$

El algoritmo involucra 4 sucesiones de polinomios: $\{u_i(x)\}$, $\{v_i(x)\}$,$\{r_i(x)\}$, $\{q_i(x)\}$. Las condiciones inicales son:

$$\begin{aligned} u_{-1}(x) = 1, \quad & v_{-1}(x) = 0, \quad & r_{-1}(x) = a(x), \\ u_0(x) = 0, \quad & v_0(x) = 1, \quad & r_0 = b(x), \end{aligned} \tag{3.28}$$

($q_{-1}(x)$ y $q_0(x)$ no están definidos). Para $i \geq 1$, $q_i(x)$ y $r_i(x)$ se definen como el cociente y residuo, respectivamente, cuando $r_{i-2}(x)$ es dividido por $r_{i-1}(x)$:

$$r_{i-2}(x) = q_i(x)r_{i-1}(x) + r_i(x), \qquad grad(r_i) < grad(r_{i-1}). \tag{3.28}$$

Entonces los polinomios $u_i(x)$ y $v_i(x)$ se definen de la siguiente manera:

$$u_i(x) = u_{i-2}(x) - q_i(x)u_{i-1}(x), \tag{3.30}$$

$$v_i(x) = v_{i-2}(x) - q_i(x)v_{i-1}(x). \tag{3.31}$$

Dado que los grados de los polinomios residuo r_i son estrictamente decrecientes, habrá uno último distinto de cero; llamémosle r_n. Entonces, r_n es el máximo común divisor de $a(x)$ y $b(x)$, y la ecuación deseada que expresa al máximo común divisor de los polinomios originales (ecuación (3.27)) es

$$u_n(x)a(x) + v_n(x)b(x) = r_n(x). \tag{3.32}$$

A continuación se presenta una tabla con propiedades entre los polinomios del algoritmo de Euclides.

A	$v_i r_{i-1} - v_{i-1} r_i = (-1)^i a$	$0 \leq i \leq n+1$
B	$u_i r_{i-1} - u_{i-1} r_i = (-1)^{i+1} b$	$0 \leq i \leq n+1$
C	$u_i v_{i-1} - u_{i-1} v_i = (-1)^{i+1}$	$0 \leq i \leq n+1$
D	$u_i a + v_i b = r_i$	$-1 \leq i \leq n+1$
E	$grad(u_i) + grad(r_{i-1}) = grad(b)$	$1 \leq i \leq n+1$
F	$grad(v_i) + grad(r_{i-1}) = grad(a)$	$0 \leq i \leq n+1$

Tabla 3.33

Ejemplo 3.22 Sean $a(x), b(x) \in \mathbb{F}_2[x]$, con $a(x) = x^8$ y $b(x) = x^6 + x^4 + x^2 + x + 1$. El algoritmo de Euclides aplicado a $a(x)$ y $b(x)$ se presenta en la siguiente tabla:

i	u_i	v_i	r_i	q_i
-1	1	0	x^8	...
0	0	1	$x^6 + x^4 + x^2 + x + 1$	...
1	1	$x^2 + 1$	$x^3 + x + 1$	$x^2 + 1$
2	$x^3 + 1$	$x^5 + x^3 + x^2$	x^2	$x^3 + 1$
3	$x^4 + x + 1$	$x^6 + x^4 + x^3 + x^2 + 1$	$x + 1$	x
4	$x^5 + x^4 + x^3 + x^2$	$x^7 + x^6 + x^3 + x + 1$	1	$x + 1$
5	$x^6 + x^4 + x^2 + x + 1$	x^8	0	$x + 1$

La linea $i = 4$ de la anterior tabla muestra que $mcd(a(x), b(x)) = 1$ (lo cual era de esperarse, ya que 0 es la única raíz de $a(x)$ y 0 no es raíz de $b(x)$, es decir, $a(x)$ y $b(x)$ no tienen raíces en común), y de la propiedad D de la Tabla 3.33 se tiene la igualdad:

$$(x^5 + x^4 + x^3 + x^2)a(x) + (x^7 + x^6 + x^3 + x + 1)b(x) = 1.$$

Lema 3.23 Supongamos que el algoritmo de Euclides, como se describió anteriormente, se aplica a los polinomios $a(x)$ y $b(x)$. Dados dos enteros $\mu \geq 0$ y $\nu \geq 0$ con $\mu + \nu = grad(a(x)) - 1$ y $\nu \geq grad(r_n(x))$, entonces existe un único índice j, $0 \leq j \leq n$, tal que:

$$grad(v_j(x)) \leq \mu, \tag{3.34}$$

$$grad(r_j(x)) \leq \nu. \tag{3.35}$$

Demostración. Ya que, por definición de los polinomios $r_j(x)$, $grad(r_n(x)) < grad(r_{n-1}(x)) < ... < grad(r_0(x)) \leq grad(r_{-1}(x)$ y también $grad(r_n) \leq \nu < grad(a(x)) = grad(r_{-1}(x))$ (pues $\mu \leq \mu + \nu = grad(a(x)) - 1 < grad(a(x))$), entonces ν está en el intervalo $[grad(r_n(x)), grad(r_{-1}(x)))$ y en consecuencia debe existir un único $j \in \{0, 1, ..., n\}$ tal que $grad(r_j(x)) \leq \nu < grad(r_{j-1}(x))$, lo cual se sigue de la unión disjunta $[grad(r_n(x)), grad(r_{-1}(x))) = \bigcup_{0 \leq j \leq n} [grad(r_j(x)), r_{j-1}(x)))$. Tenemos que $\nu < grad(r_{j-1}(x))$, entonces $\nu + 1 \leq grad(r_{j-1}(x))$; de la propiedad F de la Tabla 3.33 se sigue que $grad(v_j(x)) = grad(a(x)) - grad(r_{j-1}(x)) \leq grad(a(x)) - \nu - 1 = (grad(a(x)) - 1) - \nu = (\nu + \mu) - \nu = \mu$. ■

Teorema 3.24 Sean $a(x)$, $b(x)$, $h(x)$ y $k(x)$ polinomios distintos de cero que satisfacen

$$h(x)b(x) \equiv k(x) \pmod{a(x)}, \tag{3.36}$$

$$grad(h(x)) + grad(k(x)) < grad(a(x)). \tag{3.37}$$

Supongamos además que $v_j(x)$ y $r_j(x)$, $j = -1, 0, ..., n + 1$, son los elementos de las sucesiones de polinomios producidas cuando el algoritmo de Euclides se aplica al par $(a(x), b(x))$, y que $grad(k(x)) \geq grad(r_n(x))$. Entonces existe un único índice j, $0 \leq j \leq n$, y un polinomio $\beta(x)$ tal que

$$h(x) = \beta(x)v_j(x), \tag{3.38}$$

$$k(x) = \beta(x)r_j(x). \tag{3.39}$$

Demostración. Primero veamos la unicidad del índice j, suponiendo verdadero (3.38) y (3.39). Entonces, sean $\mu = grad(a(x)) - grad(k(x)) - 1$ y $\nu = grad(k(x))$. De (3.38) tenemos que $grad(v_j(x)) \leq grad(h(x))$, y por (3.37) se tiene que $grad(h(x)) < grad(a(x)) - grad(k(x))$, lo cual implica que $grad(h(x)) \leq grad(a(x)) - grad(k(x)) - 1 = \mu$. Así, $grad(v_j(x)) \leq \mu$. Ahora, de (3.39) se tiene que $grad(r_j(x)) \leq grad(k(x)) = \nu$. Entonces, se ha probado que $grad(v_j) \leq \mu$ y $grad(r_j(x)) = \nu$, donde $\mu + \nu = grad(a(x)) - 1$; así que del Lema 3.23, se sigue que tal índice j es único. Ahora, veamos la existencia del índice j, que hace verdadero (3.38) y (3.39). Sea j el índice que satisface (3.34) y (3.35), con ν y μ como se definieron antes (obsérvese que por hipótesis, $\nu = grad(k(x)) \geq grad(r_n(x))$. Ahora, se reescribe la propiedad D de la Tabla 3.33 y (3.36) como sigue:

$$u_j(x)a(x) + v_j(x)b(x) = r_j(x), \tag{3.40}$$

$$\mathbf{u}(x)a(x) + h(x)b(x) = k(x), \tag{3.41}$$

donde $\mathbf{u}(x)$ es algún polinomio sobre $\mathbb{F}$. Multiplicando (3.40) por $h(x)$ y (3.41) por $v_j(x)$:

$$h(x)u_j(x)a(x) + h(x)v_j(x)b(x) = h(x)r_j(x), \tag{3.42}$$

$$v_j(x)\mathbf{u}(x)a(x) + v_j(x)h(x)b(x) = v_j(x)k(x). \tag{3.43}$$

Si restamos (3.43) a (3.42), entonces $a(x)(h(x)u_j(x) - v_j(x)\mathbf{u}(x)) = h(x)r_j(x) - v_j(x)k(x)$, es decir, $h(x)r_j(x) \equiv v_j(x)k(x) (\text{mód } a(x))$. Ahora, de (3.37) y (3.35) se tiene que $grad(h(x)r_j(x)) = grad(h(x)) + grad(r_j(x)) \leq \mu + \nu < grad(a(x))$, y por (3.34) y la definición de ν, $grad(v_j(x)k(x)) = grad(v_j(x)) + grad(k(x)) \leq \mu + \nu < grad(a(x))$; se sigue que $h(x)r_j(x) = v_j(x)k(x)$ (ya que $a(x)|(h(x)r_j(x) - v_j(x)k(x))$). Este hecho, combinado con (3.42) y (3.43), implica que $h(x)u_j(x)a(x) = v_j(x)\mathbf{u}(x)a(x)$, y en consecuencia, $h(x)u_j(x) = v_j(x)\mathbf{u}(x)$ (pues $\mathbb{F}[x]$ es dominio entero). Entonces, $u_j(x)|v_j(x)\mathbf{u}(x)$, donde la propiedad C de la Tabla 3.33 nos dice que $mcd(u_j, v_j) = 1$, y así $u_j(x)|\mathbf{u}(x)$, o bien, $\mathbf{u}(x) = \beta_1(x)u_j(x)$ para algún polinomio $\beta_1(x) \in \mathbb{F}[x]$. Similarmente, $v_j(x)|h(x)$, o bien $h(x) = \beta_2(x)v_j(x)$ para algún polinomio $\beta_2(x) \in \mathbb{F}[x]$; en consecuencia, de $h(x)u_j(x) = v_j(x)\mathbf{u}(x)$, se sigue que $\beta_1 = \beta_2$. ■

Definición 3.25 Si $(a(x), b(x))$ es un par de polinomios distintos de cero con $grad(a(x)) \geq grad(b(x))$, y si (μ, ν) es un par de enteros no negativos tales que $\mu + \nu = grad(a(x)) - 1$ y $\nu \geq grad(r_n(x))$, donde $r_n(x) = mcd(a(x), b(x))$, entonces $Euclides(a(x), b(x), \mu, \nu)$ denota el procedimiento que regresa al único par de polinomios $(v_j(x), r_j(x))$ tales que $grad(v_j(x)) \leq \mu$ y $grad(r_j(x)) \leq \nu$, cuando el algoritmo de Euclides es aplicado al par $(a(x), b(x))$.

El siguiente teorema resume los resultados de esta sección.

Teorema 3.26 Supongamos que $h(x)$ y $k(x)$ son polinomios distintos de cero que satisfacen

$$h(x)b(x) \equiv k(x) \quad (\text{mód } a(x)), \tag{3.44}$$

$$grad(h(x)) \leq \mu \tag{3.45}$$

$$grad(k(x)) \leq \nu \tag{3.46}$$

donde μ y ν son enteros no negativos tales que $\mu + \nu = grad(a(x)) - 1$, y $grad(k(x)) \geq grad(r_n(x))$, donde $r_n(x) = mcd(a(x), b(x))$. Si $(v_j(x), r_j(x))$ es el par de polinomios retornados por $Euclides(a(x), b(x), \mu, \nu)$, entonces existe un polinomio $\beta(x) \in \mathbb{F}[x]$ tal que

$$h(x) = \beta(x)v_j(x), \tag{3.47}$$

$$k(x) = \beta(x)r_j(x). \tag{3.48}$$

Demostración. El Teorema 3.24 garantiza que existe un único índice j tal que (3.47) y (3.48) se cumplen. Además, el procedimiento $Euclides(a(x), b(x), \mu, \nu)$ retorna a este par de polinomios, pues $grad(v_j(x)) \leq grad(h(x)) \leq \mu$ y $grad(r_j(x)) \leq grad(k(x)) \leq \nu$. ■

Ejemplo 3.27 Sean $a(x) = x^8$, $b(x) = x^6 + x^4 + x^2 + x + 1 \in \mathbb{F}_2[x]$. Usando la tabla del ejemplo 3.22, podemos tabular la salida de *Euclides* para los ocho posibles pares (μ, ν):

(μ, ν)	$Euclides(a(x), b(x), \mu, \nu)$
(0,7)	$(1,\ x^6+x^4+x^2+x+1)$
(1,6)	$(1,\ x^6+x^4+x^2+x+1)$
(2,5)	$(x^2+1,\ x^3+x+1)$
(3,4)	$(x^2+1,\ x^3+x+1)$
(4,3)	$(x^2+1,\ x^3+x+1)$
(5,2)	$(x^5+x^3+x^2,\ x^2)$
(6,1)	$(x^6+x^4+x^3+x^2+1,\ x+1)$
(7,0)	$(x^7+x^6+x^3+x+1,\ 1)$

Supongamos que deseamos resolver la congruencia polinomial

$$(x^6+x^4+x^2+x+1)\sigma(x) \equiv \lambda(x) \pmod{x^8}$$

para $\sigma(x)$ y $\lambda(x)$, sujeto a la restricción $grad(\sigma(x)) \leq 3$, $grad(\lambda(x)) \leq 4$. De acuerdo al Teorema 3.26, invocamos a $Euclides(x^8, x^6+x^4+x^2+x+1, 3, 4)$, que por la anterior tabla regresa al par $(x^2+1,\ x^3+x+1)$, así que todas las soluciones del problema dado son de la forma $\sigma(x) = \beta(x^2+1)$, $\lambda = \beta(x^3+x+1)$, con $grad(\lambda(x)) \leq 1$. Si además se pide que $mcd(\sigma(x), \lambda(x)) = 1$, entonces la única solución será $\sigma(x) = x^2+1$, $\lambda(x) = x^3+x+1$.

3.5. DECODIFICACIÓN DE CÓDIGOS BCH: PARTE II

Ahora, con los resultados de la sección anterior, volvemos al problema de la decodificación de los códigos BCH. De acuerdo a lo antes definido, tratamos de recuperar la palabra código $\mathbf{c}$ a partir de la palabra recibida $\mathbf{r}$.

El primer paso en el procesos de decodificación es calcular el síndrome polinomial $\mathbf{s}(x)$ definido previo a la congruencia polinomial (3.18).

El siguiente paso es utlizar el algoritmo de Euclides, en particular el procedimiento

$$Euclides(x^{2t}, \mathbf{s}(x), t, t-1)$$

con la finalidad de resolver la congruencia polinomial (3.18) para $\sigma(x)$ y $\lambda(x)$. Esto es posible, ya que por definición de $\mathbf{s}(x)$ se tiene que $grad(\mathbf{s}(x)) < 2t = grad(x^{2t})$, y como se hizo notar, de la hipótesis $|Sop(\mathbf{e})| = wt(\mathbf{e}) \leq t$ se sigue que $grad(\sigma(x)) \leq t$ y $grad(\lambda(x)) \leq t-1$; además, $grad(\lambda(x)) \geq 0 = grad(mcd(x^{2t}, \sigma(x)))$, pues $mcd(x^{2t}, \sigma(x)) = 1$. Por lo tanto, todas las condiciones del Teorema 3.26 se satisfacen para $a(x) = x^{2t}$, $b(x) = \mathbf{s}(x)$, $h(x) = \sigma(x)$, $k(x) = \lambda(x)$, $\mu = t$, $\nu = t-1$, entonces del Teorema 3.26 y el Lema 3.18, $Euclides(x^{2t}, \mathbf{s}(x), t, t-1)$ retorna al par de polinomios $(v_j(x), r_j(x))$ tales que $\sigma(x) = \beta v_j(x)$, $\lambda(x) = \beta r_j(x)$, donde β en este caso, es un escalar diferente de cero. El escalar β puede ser determinado a partir de que ocurre la igualdad $\sigma(0) = 1$ (ver (3.10)), es decir, $\beta = v_j(0)^{-1}$, y así:

$$\sigma(x) = v_j(0)^{-1} v_j(x), \qquad \lambda(x) = v_j(0)^{-1} r_j(x).$$

El paso final en el proceso de decodificación es utilizar $\sigma(x)$ y $\lambda(x)$ par determinar el patrón de error $\mathbf{e} = (e_0, e_1, ..., e_{n-1})$, y en consecuencia, la palabra código corregida $\mathbf{c} = \mathbf{r} - \mathbf{e}$. Para ello, nos basamos en $\sigma(x) = \prod_{i \in Sop(\mathbf{e})} (1-\alpha^i)x$, donde $Sop(\mathbf{e}) = \{i : (\leq i \leq n-1) \wedge (e_i \neq 0)\}$. Entonces, para localizar las posiciones distintas de cero del patrón de error $\mathbf{e}$, se calcula $\sigma(\alpha^{-j})$, $j = 0, 1, ..., n-1$, donde $\sigma(\alpha^{-j}) = 0$ indica que $j \in Sop(\mathbf{e})$ (pues las raíces de $\sigma(x)$ son precisamente los elementos α^{-j} con $j \in Sop(\mathbf{e})$). Tras lo anterior, el conjunto $Sop(\mathbf{e})$ queda determinado, y en consecuencia el patrón de error $\mathbf{e}$; una vez teniendo $\mathbf{e}$, nuestra palabra código corregida será $\mathbf{c} = \mathbf{r} - \mathbf{e}$.

El siguiente ejemplo involucra conceptos de todo el capítulo.

Ejemplo 3.28 Considere el código binario primitivo BCH en el sentido estricto , $\mathcal{C} = \mathfrak{B}_2(2^4-1 = 15, \delta = 2t+1 = 7, \alpha, 1)$, donde α es un elemento primitivo de $\mathbb{F}_{2^4}$ que es raíz de $x^4+x+1 \in \mathbb{F}_2[x]$. Como $\delta = 7$, entonces el polinomio generador de $\mathcal{C}$ es $g(x) = mcm\{M_1(x), M_2(x), ..., M_6(x)\}$. Además, en este caso, $t = 3$. Las clases ciclotómicas de 2 módulo 15 son las siguientes:

$$C_2(0) = \{0\},\ C_2(1) = \{1,2,4,8\},\ C_2(3) = \{3,6,9,12\},\ C_2(5) \text{ y } C_2(7) = \{7,11,13,14\}.$$

Como $C_2(1) = C_2(2) = C_2(4)$ y $C_2(3) = C_2(6)$, entonces $M_1(x) = M_2(x) = M_4(x)$ y $M_3(x) = M_6(x)$; se sigue que $g(x) = M_1(x)M_3(x)M_5(x) = \prod_{j\in C_2(1)}(x-\alpha^j) \prod_{j\in C_2(3)}(x-\alpha^j) \prod_{j\in C_2(5)}(x-\alpha^j) = (x^4+x+1)(x^4+x^3+x^2+x+1)(x^2+x+1) = x^{10}+x^8+x^5+x^4+x^2+x+1$. Supongamos que el vector $\mathbf{r} = (110000110110101)$ es recibido. Los síndromes $s_1, ..., s_6$ se calculan mediante la siguiente expresión (ver (3.15)):

$$s_j = \sum_{i=0}^{15} r_i\alpha^{ij} = 1+\alpha^j+\alpha^{6j}+\alpha^{7j}+\alpha^{9j}+\alpha^{10j}+\alpha^{12j}+\alpha^{14j},$$

ya que $Sop(\mathbf{r}) = \{0,1,6,7,9,10,12,14\}$. Entonces, usando la tabla del Ejemplo 3.20 para los respectivos cálculos, se tienen lo siguiente:

$s_1 = 1+\alpha+\alpha^6+\alpha^7+\alpha^7+\alpha^{10}+\alpha^{12}+\alpha^{14} = \alpha^{12}$,
$s_3 = 1+\alpha^3+\alpha^{18}+\alpha^{21}+\alpha^{27}+\alpha^{30}+\alpha^{36}+\alpha^{42} = 0$,
$s_5 = 1+\alpha^5+\alpha^{30}+\alpha^{35}+\alpha^{45}+\alpha^{50}+\alpha^{60}+\alpha^{70} = 1$.
Como $\mathbb{F}_{2^4}$ tiene característica 2 y $r_i \in \mathbb{F}_2$, entonces $s_{2j} = \sum_{i=0}^{n-1} r_i\alpha^{2ij} = \sum_{i=0}^{n-1} r_i^2\alpha^{2ij} = (\sum_{i=0}^{n-1} r_i\alpha^{ij})^2 = (s_j)^2$. Entonces el resto de síndromes se pueden determinar, a partir de los cálculos anteriores, como sigue:
$s_2 = (s_1)^2 = (\alpha^{12})^2 = \alpha^{24} = \alpha^9$,
$s_4 = (s_2)^2 = (\alpha^9)^2 = \alpha^{18} = \alpha^3$,
$s_6 = (s_3)^2 = 0^2 = 0$. Por lo tanto, $\mathbf{s}(x) = \alpha^{12}+\alpha^9x+\alpha^3x^3+x^4$. Como $t = 3$, entonces $grad(\sigma(x)) \leq 3$ y $grad(\lambda(x)) \leq 2$, y se resuelve la congruencia (3.18) para nuestro caso, es decir, se resuelve para $\sigma(x)$ y $\lambda(x)$ la siguiente congruencia :

$$\sigma(x)(\alpha^{12}+\alpha^9x+\alpha^3x^3+x^4) \equiv \lambda(x) \pmod{x^6}.$$

Entonces, podemos invocar a $Euclides(x^6, \alpha^{12}+\alpha^9x+\alpha^3x^3+x^4, 3, 2)$, procedimiento que retorna al par de polinomios $(x^2+\alpha^3x+\alpha^6, \alpha^3)$, y cumplen

$$\sigma(x) = \beta(x^2+\alpha^3x+\alpha^6)$$

$$\lambda(x) = \beta\alpha^3,$$

donde $\beta = (0^2+\alpha^30+\alpha^6)^{-1} = (\alpha^6)^{-1} = \alpha^9$ (por el último paso de nuestro algoritmo de decodificación). Por lo tanto $\sigma(x) = 1+\alpha^{12}x+\alpha^9x^2$, $\lambda(x) = \alpha^{12}$. Ahora, buscamos $i \in \{0,1,..,14\}$ tal que $\sigma(\alpha^{-i}) = 0$. Evaluando a los 14 elementos $\sigma(\alpha^i)$, $i = 0,1,..,14$, concluimos que $\sigma(\alpha^i) = 0$ para $i \in \{2,7\}$. Entonces $\mathbf{e} = (001000010000000)$, y en consecuencia $\widehat{\mathbf{c}} = \mathbf{r}-\mathbf{e} = (110000110110101)-(001000010000000) = (111000100110101)$. Observe que $\widehat{\mathbf{c}}$ es una buena estimación de la palabra código transmitida, pues el polinomio asociado al vector $\widehat{\mathbf{c}}$ es $\widehat{\mathbf{c}}(x) = 1+x+x^2+x^6+x^9+x^{10}+x^{12}+x^{14}$, el cual es divisible por el polinomio generador $g(x)$ de $\mathcal{C}$, a saber $\widehat{\mathbf{c}}(x) = g(x)(x^4+1)$, lo cual indica que, efectivamente, $\widehat{\mathbf{c}}$ es una palabra-código de $\mathcal{C}$. Por último, para justificar al par de polinomios retornados por $Euclides(x^6, \alpha^{12}+\alpha^9x+\alpha^3x^3+x^4, 3, 2)$, presentamos a continuación el algoritmo de Euclides aplicado a x^6 y $\alpha^{12}+\alpha^9x+\alpha^3x^3+x^4$:

i	u_i	v_i	r_i	q_i
-1	1	0	x^6	...
0	0	1	$x^4+\alpha^3x^3+\alpha^9x+\alpha^{12}$	...
1	1	$x^2+\alpha^3x+\alpha^6$	α^3	$x^2+\alpha^3x+\alpha^6$
2	$\alpha^{12}x^4+x^3+\alpha^6x+\alpha^9$	$\alpha^{12}x^6$	0	$\alpha^{12}x^4+x^3+\alpha^6x+\alpha^9$

Así que por el Teorema 3.26, (v_1, r_1) es el par retornado por $Euclides(x^6, \alpha^{12}+\alpha^9x+\alpha^3x^3+x^4, 3, 2)$.

Los códigos BCH tienen una subclase de códigos bastante importantes en las aplicaciones, y con propiedades teóricas interesantes, a saber, los códigos Reed-Solomon. Con la teoría vista hasta este punto, en el siguiente capítulo se presentarán algunas propiedades y características de los códigos Reed-Solomon.

Capítulo 4

CÓDIGOS REED-SOLOMON

La subclase más importante de los códigos BCH es la clase de los códigos Reed-Solomon. Los códigos Reed-Solomon fueron introducidos por I.S. Reed y G. Solomon. Estos códigos son de gran importancia en las aplicaciones prácticas por sus propiedades respecto a la corrección de errores. En esta sección, se presentarán resultados importantes sobre esta clase de códigos, teniendo como precedente la teoría de los códigos BCH del capítulo anterior.

4.1. PROPIEDADES IMPORTANTES DE LOS CÓDIGOS REED-SOLOMON

Definición 4.1 Un código q-ario Reed-Solomon es un código q-ario BCH de la forma:

$$\mathfrak{B}_q(q-1, \delta, \alpha, b),$$

donde $2 \leq \delta \leq q-1$.

Observaciones: 1) Los códigos Reed-Solomon nunca son códigos binarios.

2)Para un código q-ario Reed-Solomon, es clara la relación entre la longitud del código y q, por lo que en la notación se omite el subíndice q de $\mathfrak{B}$, es decir, escribimos $\mathfrak{B}(q-1, \delta, \alpha, b)$ para referirnos a un código Reed-Solomon.

3) Sea $\mathcal{C} = \mathfrak{B}(q-1, \delta, \alpha, b)$ un código q-ario Reed-Solomon. Ya que α es un elemento primitivo de $\mathbb{F}_q$, entonces el polinomio mónico de menor grado sobre $\mathbb{F}_q$ que tiene a α^i como raíz es $x - \alpha^i$, es decir, el polinomio mínimo de α^i con respecto a $\mathbb{F}_q$ es $M_i = x - \alpha^i$. Entonces, se sigue que el polinomio generador para $\mathcal{C}$ es

$$\begin{aligned} g(x) &= mcm\{M_b(x), M_{b+1}(x), ..., M_{b+\delta-2}\} \\ &= mcm\{x - \alpha^b, x - \alpha^{b+1}, ..., x - \alpha^{b+\delta-2}\} \\ &= (x - \alpha^b)(x - \alpha^{b+1}) \cdot ... \cdot (x - \alpha^{b+\delta-2}), \end{aligned}$$

ya que $\delta \leq q-1$ implica que los elementos $\alpha, \alpha^{b+1}, ..., \alpha^{b+\delta-2}$ son distintos por pares. Además, $grad(g(x)) = \delta - 1$.

Ejemplo 4.2 Considere el código 7-ario Reed-Solomon $\mathcal{C} = \mathfrak{B}(6, 4, 3, 1)$. Entonces $\mathcal{C}$ tiene polinomio generador

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

Entonces la dimensión de $\mathcal{C}$ es $k = 6 - grad(g(x)) = 6 - 3 = 3$. Por lo tanto, $\mathcal{C}$ es un $[6, 3]$-código 7-ario. Como se sabe, podemos formar una matriz generadora para $\mathcal{C}$ a partir de $g(x)$:

$$G = \begin{pmatrix} 6 & 1 & 3 & 1 & 0 & 0 \\ 0 & 6 & 1 & 3 & 1 & 0 \\ 0 & 0 & 6 & 1 & 3 & 1 \end{pmatrix}.$$

El polinomio de chequeo de $\mathcal{C}$ es:

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

entonces de $h(x)$ obtenemos una matriz de chequeo de paridad de $\mathcal{C}$:

$$H = \begin{pmatrix} 1 & 4 & 1 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 & 1 \end{pmatrix}.$$

Como cualesquiera 3 columnas de H son linealmente independientes y H tiene 4 columnas linealmente dependientes (en este caso, cualesquiera 4 columnas son linealmente dependientes), por el Corolario 2.21, la distancia mínima de $\mathcal{C}$ es $d = 4$. Observe que si $n = 6$ (que es la longitud de $\mathcal{C}$), entonces $k + d = 3 + 4 = 7 = 6 + 1 = n + 1$, es decir, $\mathcal{C}$ es un $[6, 3, 4]$-código 7-ario MDS.

Ejemplo 4.3 Considere el código 8-ario Reed-Solomon $\mathcal{C} = \mathfrak{B}(7, 3, \alpha, 1)$, donde $\alpha \in \mathbb{F}_8$ es raíz de $1 + x + x^3 \in \mathbb{F}_2[x]$. Entonces $\mathcal{C}$ tiene polinomio generador

$$g(x) = (x-\alpha)(x-\alpha^2) = (x+\alpha)(x+\alpha^2) = x^2 + (\alpha^2+\alpha)x + (\alpha+1).$$

Entonces la dimensión de $\mathcal{C}$ es $k = 7 - grad(g(x)) = 7 - 2 = 5$. Por lo tanto, $\mathcal{C}$ es un $[7, 5]$-código 8-ario. Nuevamente, de $g(x)$ obtenemos una matriz generadora para $\mathcal{C}$:

$$\begin{pmatrix} \alpha+1 & \alpha^2+\alpha & 1 & 0 & 0 & 0 & 0 \\ 0 & \alpha+1 & \alpha^2+\alpha & 1 & 0 & 0 & 0 \\ 0 & 0 & \alpha+1 & \alpha^2+\alpha & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha+1 & \alpha^2+\alpha & 1 & 0 \\ 0 & 0 & 0 & 0 & \alpha+1 & \alpha^2+\alpha & 1 \end{pmatrix}$$

y el polinomio de chequeo de paridad de $\mathcal{C}$ es:

$$h(x) = \tfrac{x^7-1}{g(x)} = \alpha^4 + (1+\alpha^4)x + (1+\alpha^4)x^2 + x^3 + \alpha^4 x^4 + x^5.$$

Así, a partir de $h(x)$, una matriz de chequeo de paridad para $\mathcal{C}$ es:

$$H = \begin{pmatrix} 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^4 & \alpha^4 & 0 \\ 0 & 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^4 & \alpha^4 \end{pmatrix}.$$

Ya que cualesquiera 2 columnas de H son linealmente independientes y H tiene 3 columnas linealmente dependientes (en este caso, cualesquiera 3 columnas son linealmente dependientes pues pueden ser vistas como elementos del espacio vectorial ${}_{\mathbb{F}_8}\mathbb{F}_8^2$), por el Corolario 2.21, la distancia mínima de $\mathcal{C}$ es $d = 3$. Si $n = 7$ (la longitud de $\mathcal{C}$), entonces $k + d = 5 + 3 = 8 = 7 + 1 = n + 1$, es decir, $\mathcal{C}$ es un $[7, 5, 3]$-código 8-ario MDS.

De los dos ejemplos anteriores podemos notar que los códigos Reed-Solomon resultan ser códigos MDS, y la distancia designada coincide con la distancia mínima. El siguiente teorema garantiza que ambas propiedades se cumplen en general para cualquier código Reed-Solomon.

Teorema 4.4 Los códigos Reed-Solomon son códigos de máxima distancia separable (MDS). Además, la distancia mínima de un código Reed-Solomon coincide con su distancia designada.

Demostración. Sea $\mathcal{C} = \mathfrak{B}(q-1, \delta, \alpha, b)$ un código Reed-Solomon. Si $g(x)$ es el polinomio generador de $\mathcal{C}$, entonces $grad(g(x)) = \delta - 1$, de ahí que $k = dim(\mathcal{C}) = n - grad(g(x)) = n - \delta + 1$, lo cual implica que $\delta = n - k + 1$. De la **cota BCH**, se sigue que la distancia mínima d de $\mathcal{C}$ cumple $d \geq \delta = n - k + 1$. Por la **cota de Singleton** se tiene que $d \leq n - k + 1$, y en consecuencia $d = n - k + 1 = \delta$. Por lo tanto, $\mathcal{C}$ es un código MDS ($\mathcal{C}$ es cíclico por definición) y su distancia mínima es igual a su distancia designada. ■

Ejemplo 4.5 Sea α una raíz de $1 + x + x^4 \in \mathbb{F}_2[x]$. Entonces α es un elemento primitivo de $\mathbb{F}_{16}$. Consideremos al código Reed-Solomon $\mathcal{C} = \mathfrak{B}(15, 5, \alpha, 3)$ cuyo polinomio generador es $g(x) = (x-\alpha^3)(a-\alpha^4)(x-\alpha^5)(x-\alpha^6) = x^4 + x^3 + \alpha^{10}x^2 + \alpha^9 x + \alpha^3$, y polinomio de chequeo de paridad $h(x) = \frac{x^{15}-1}{g(x)} = x^{11} + x^{10} + \alpha^5 x^9 + \alpha^7 x^8 + \alpha^3 x^7 + \alpha^{13} x^6 + \alpha^{10} x^5 + \alpha^9 x^4 + \alpha^7 x^3 + \alpha^{14} x^2 + \alpha^3 x + \alpha^{12}$. Se sigue que la dimensión de $\mathcal{C}$ es $k = 15 - grad(g(x)) = 15 - 4 = 11$, y de $h(x)$ podemos formar una matriz H de chequeo de paridad para $\mathcal{C}$, donde $H \in \mathcal{M}_{4x15}(\mathbb{F}_{16})$. En este caso no es fácil determinar la distancia mínima d de $\mathcal{C}$ a partir de H, sin embargo, por el Teorema 4.4 no hay que realizar cálculos, pues $d = \delta$ y en consecuencia, $\mathcal{C}$ es un $[15, 11, 5]$-código cíclico sobre $\mathbb{F}_{16}$.

Ejemplo 4.6 (i) Sea $\mathcal{C}$ como en el Ejemplo 4.2. Entonces $\mathcal{C}$ tiene matriz de chequeo de paridad

$$\mathrm{H} = \begin{pmatrix} 1 & 4 & 1 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 & 1 \end{pmatrix}.$$

Por el Teorema 2.24, la matriz

$$\overline{\mathrm{H}} = \begin{pmatrix} 1 & 4 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

es una matriz de chequeo de paridad del código extendido $\overline{\mathcal{C}}$. Entones, por el Corolario 2.21, el código extendido tiene distancia mínima igual a 5, y por lo tanto $\overline{\mathcal{C}}$ es un $[7, 3, 5]$-código MDS.

(ii) Sea $\mathcal{C}$ como en el Ejemplo 4.3. Entonces $\mathcal{C}$ tiene matriz de chequeo de paridad

$$\mathrm{H} = \begin{pmatrix} 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^2 & \alpha^4 & 0 \\ 0 & 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^4 & \alpha^4 \end{pmatrix}.$$

Por el Teorema 2.24 , la matriz

$$\overline{\mathrm{H}} = \begin{pmatrix} 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^2 & \alpha^4 & 0 & 0 \\ 0 & 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^4 & \alpha^4 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

es una matriz de chequeo de paridad del código extendido. Entonces, por el Corolario 2.21, el código extendido tiene distancia mínima igual a 4, y por lo tanto $\overline{\mathcal{C}}$ es un $[8, 5, 4]-$código MDS.

En los dos incisos del Ejemplo 4.6, notamos que el código extendido de los respectivos códigos Reed-Solomon preservan la propiedad de ser códigos MDS. El siguiente teorema muestra que, para ciertos códigos Reed-Solomon, el código extendido seguirá siendo un código MDS.

Teorema 4.7 Sea $\mathcal{C} = \mathfrak{B}(q-1, \delta, \alpha, 1)$ un código Reed-Solomon. Entonces el código extendido $\overline{\mathcal{C}}$ sigue siendo un código MDS.

Demostración. Ya que $\mathcal{C}$ es un $[q-1, q-\delta, \delta]$-código cíclico, tenemos que mostrar que $\overline{\mathcal{C}}$ tiene parámetros $[q, q-\delta, \delta+1]$. Para esto, por el Teorema 2.24, basta demostrar que la distancia mínima de $\overline{\mathcal{C}}$ es $\delta+1$. El polinomio generador de $\mathcal{C}$ es:

$$g(x) = \prod_{i=1}^{\delta-1}(x-\alpha^i).$$

Sea $c(x) + \langle x^{q-1}-1\rangle = \sum_{i=0}^{q-2} c_i x^i + \langle x^{q-1}-1\rangle \in \mathcal{C}$ una palabra-código distinta de cero. Es suficiente probar que el peso de $\bar{c} = (c_0, c_1, ..., c_{q-2}, -\sum_{i=0}^{q-2} c_i)$ es mayor o igual a $\delta+1$, ya que entonces se tendría que la distancia mínima del código extendido $\overline{\mathcal{C}}$ es mayor o igual a $\delta+1$, y por el Teorema 2.24 la distancia mínima de $\overline{\mathcal{C}}$ sería igual a $\delta+1$. Entonces para algún $f(x) \in \mathbb{F}_q[x]$ se tiene que $c(x) = f(x)g(x)$. Analicemos los siguientes casos:

Caso 1: $f(1) \neq 0$. Observe que 1 no puede ser raíz de $g(x)$, ya que las raíces de $g(x)$ son los elementos α^i, $i = 1, ..., \delta-1$, donde $2 \leq \delta \leq q-1$ y α es una raíz $(q-1)$-ésima de la unidad sobre $\mathbb{F}_q$. Entonces $c(1) = f(1)g(1) \neq 0$, y en consecuencia $wt(\overline{c}) = wt(c(x)) + 1 \geq \delta+1$.

Caso 2: $f(1) = 0$, es decir, $(x-1)$ es un factor lineal de $f(x)$. Sea $u(x) \in \mathbb{F}_q[x]$ tal que $f(x) = u(x)(x-1)$. Entonces, $c(x) = u(x)(x-1)g(x) = u(x)\prod_{i=0}^{\delta-1}(x-\alpha^i)$ es también una palabra-código del código BCH con distancia designada $\delta+1$ y con polinomio generador $\prod_{i=0}^{q-2}(x-\alpha^i)$. Entonces, por la **cota BCH**, se sigue que $wt(c(x)) \geq \delta+1$. Entonces, $wt(\overline{c}) \geq wt(c(x)) \geq \delta+1$. ■

Desafortunadamente, los códigos Reed-Solomon no son binarios, mientras que en las aplicaciones se requieren códigos binarios. En la práctica la técnica de concatenación se usa para producir códigos binarios a partir de códigos Reed-Solomon sobre extensiones de $\mathbb{F}_2$.

Sea $\mathcal{C} = \mathfrak{B}(2^m-1, \delta, \alpha, b)$ un código Reed-Solomon. Aplicando la técnica de concatenación como en el Teorema 2.26, concatenamos $\mathcal{C}$ con el código trivial $\mathbb{F}_2^m$, es decir, con el $[m, m, 1]$-código lineal sobre $\mathbb{F}_2$. Sea $\{v_1, ..., v_m\}$ una base del espacio vectorial ${}_{\mathbb{F}_2}\mathbb{F}_{2^m}$ y considere la función

$$\phi\colon \mathbb{F}_{2^m} \longrightarrow \mathbb{F}_2^m \text{ definida por } \phi(u_1v_1 + u_2v_2 + ... + u_mv_m) = (u_1, u_2, ..., u_m). \tag{4.1}$$

Observe que ϕ es un isomorfismo de $\mathbb{F}_2$-espacios vectoriales. Entonces, por el Teorema 2.26, tenemos el siguiente resultado.

Teorema 4.8 Sea $\mathcal{C} = \mathfrak{B}(n = 2^m - 1, \delta, \alpha, b)$ un código Reed-Solomon. Entonces

$$\phi^*(\mathcal{C}) = \{(\phi(c_0), \phi(c_1), ..., \phi(c_{n-1})) : (c_0, c_1, ..., c_{n-1}) \in \mathcal{C}\}$$

es un $[mn, mk]$-código lineal binario con distancia mínima $\geq \delta$.

Ejemplo 4.9 Considere el código Reed-Solomon $\mathcal{C} = \mathfrak{B}(7, 7, \alpha, 1)$, donde $\alpha \in \mathbb{F}_{2^3}$ es raíz de $1 + x + x^3 \in \mathbb{F}_2[x]$. Entonces, por la Observación 2) posterior a la Definición 4.1, el grado del polinomio generador de $\mathcal{C}$ es $7 - 1 = 6$, de ahí que $dim(\mathcal{C}) = 7 - 6 = 1$. Por el Teorema 4.4, en este caso, la distancia mínima de $\mathcal{C}$ es igual a 7. Así, $\mathcal{C}$ es un $[7, 1, 7]$-código lineal sobre $\mathbb{F}_{2^3}$, es decir,

$$\mathcal{C} = \{a(1, 1, 1, 1, 1, 1, 1) : a \in \mathbb{F}_8\}. \tag{4.2}$$

Al concatenar a $\mathcal{C}$ con el $[3, 3, 1]$-código lineal sobre $\mathbb{F}_2$, tenemos que

$$\phi^* : F_{2^3}^7 \longrightarrow \mathbb{F}_2^{21},$$

como se definió en el Teorema 2.26, es un isomorfismo de $\mathbb{F}_2$-espacios vectoriales.

Observe que $\mathbb{F}_8^7$ puede ser visto como espacio vectorial sobre $\mathbb{F}_2$, pues $\mathbb{F}_2$ es un subcampo de $\mathbb{F}_8$, entonces $\mathcal{C}$ también puede ser visto como $\mathbb{F}_2$-subespacio de $\mathbb{F}_8^7$. Ya que ϕ^* es isomorfismo de $\mathbb{F}_2$-espacios vectoriales, se tiene que $\phi^*(\mathcal{C}) \cong \mathcal{C}$ (como $\mathbb{F}_2$ espacios vectoriales). Dado que $\{1, \alpha, \alpha^2\}$ es una base de ${}_{\mathbb{F}_2}\mathbb{F}_8$, entonces de (4.2) se tiene la siguiente igualdad:

$$\mathcal{C} = \{\lambda_0(1, 1, 1, 1, 1, 1, 1) + \lambda_1(\alpha, \alpha, \alpha, \alpha, \alpha, \alpha, \alpha) + \lambda_2(\alpha^2, \alpha^2, \alpha^2, \alpha^2, \alpha^2, \alpha^2, \alpha^2) : \lambda_0, \lambda_1, \lambda_2 \in \mathbb{F}_2\}.$$

Entonces,

$$\phi^*(\mathcal{C}) = \{\lambda_0\phi^*(1, 1, 1, 1, 1, 1, 1) + \lambda_1\phi^*(\alpha, \alpha, \alpha, \alpha, \alpha, \alpha, \alpha) + \lambda_2\phi^*(\alpha^2, \alpha^2, \alpha^2, \alpha^2, \alpha^2, \alpha^2, \alpha^2) : \lambda_i \in \mathbb{F}_2\}$$

$$= \{\lambda_0(\phi(1), \phi(1), ..., \phi(1)) + \lambda_1(\phi(\alpha), \phi(\alpha), \phi(\alpha), ..., \phi(\alpha)) + \lambda_2(\phi(\alpha^2), \phi(\alpha^2), ..., \phi(\alpha^2)) : \lambda_i \in \mathbb{F}_2\}$$
$$= \{\lambda_0(1, 0, 0, 1, 0, 0, ..., 1, 0, 0) + \lambda_1(0, 1, 0, 0, 1, 0, ..., 0, 1, 0) + \lambda_2(0, 0, 1, 0, 0, 1, ..., 0, 0, 1) : \lambda_i \in \mathbb{F}_2\}.$$

De acuerdo al Teorema 2.26, la distancia mínima de $\phi^*(\mathcal{C})$ es mayor o igual a 7, pero $\phi^*(1, 1, 1, 1, 1, 1, 1) = (1, 0, 0, 1, 0, 0, ..., 1, 0, 0) \in \phi^*(\mathcal{C})$ tiene peso igual a 7. Por lo tanto, $\phi^*(\mathcal{C})$ es un $[21, 3, 7]$- código lineal sobre $\mathbb{F}_2$.

4.2. ERRORES RÁFAGA

En general los códigos correctores de errores aleatorios no son eficientes al corregir errores ráfaga. Este tipo de errores son bastante frecuentes en cuestiones practicas, como en telefonía, discos compactos, etc. Por ello es de interes construir códigos con buenas propiedades en la corrección de errores ráfaga. A través de los años se ha comprobado que los códigos cíclicos son eficientes en tal tarea. En particular, los códigos Reed-Solomon tienen propiedades destacadas respecto a la correción de los errores mencionados anteriormente, lo que hace de estos códigos muy socorridos en implementaciones prácticas.

Definición 4.10 (i) Una ráfaga de longitud $l \geq 1$ es un vector binario cuyas componentes distintas de cero están confinadas a l posiciones cíclicamente consecutivas, con la primera y última posición distinta de cero.

(ii) Un código es llamado un código corrector de errores l-ráfaga si puede corregir todos los errores-ráfaga de longitud l o menor,es decir, patrones de error que son ráfagas de longitud l o menor.

Ejemplo 4.11 (0011010000) es una ráfaga de longitud 4, mientras que (01000000000000100) es una ráfaga de longitud 5.

Teorema 4.12 Un código lineal $\mathcal{C}$ es un código corrector de errores l-ráfaga si y sólo si todos los errores ráfaga de longitud l o menor pertenecen a clases distintas de $\mathcal{C}$ (con la relación de equivalencia dada en el Lema 2.9).

Demostración. Si todos los errores ráfaga de longitud l o menor pertenecen a distintas clases, entonces cada error queda determinado por su síndrome. El error puede corregirse mediante su síndrome. Por otro lado, supongamos que dos errores ráfaga b_1 y b_2 de longitud l o menor pertenecen a la misma clase de $\mathcal{C}$. La diferencia $c = b_1 - b_2$ es una palabra código. Por lo tanto, si b_1 es recibida, entonces b_1 puede ser decodificada a 0 y c.

■

Teorema 4.13 Sea $\mathcal{C} = \mathfrak{B}(2^m - 1, \delta, \alpha, b)$ un código Reed-Solomon. Entonces, el código $\phi^*(\mathcal{C})$ puede corregir $m\lfloor \frac{n-k}{2} \rfloor - m + 1$ errores-ráfaga, donde $n = 2^m - 1$ y $k = n - \delta + 1$.

Demostración. Consideremos ϕ^* como se definió en el Teorema 2.26 con ϕ como está definida en (4.1). Sea $l = m\lfloor \frac{n-k}{2} \rfloor - m + 1$. Por el Teorema 4.12, es suficiente mostrar que cada error-ráfaga de longitud l o menor pertenece a una clase distinta de $\phi^*(\mathcal{C})$.

Sean $e_1, e_2 \in \mathbb{F}_2^{mn}$ dos errores ráfaga de longitud l_1 y l_2 respectivamente, donde $l_1, l_2 \leq l$, y supongamos que e_1 y e_2 pertenecen a la misma clase de $\phi^*(\mathcal{C})$. Observe que la función $\phi^* : \mathbb{F}_{2^m}^n \longrightarrow \mathbb{F}_2^{mn}$ como se definió en el Teorema 2.26, en este caso, es biyectiva. Sean $c_1, c_2 \in \mathbb{F}_{2^m}^n$ tales que $\phi^*(c_1) = e_1$ y $\phi^*(c_2) = e_2$. Observe que si $c_1 = (v_1, v_2, ..., v_n)$ entonces $e_1 = \phi^*(v_1, v_2, ..., v_n) = (\phi(v_1), \phi(v_2), ..., \phi(v_n))$, y también observe que $\phi(v_i)$ corresponden a m entradas del vector $e_1 = (\phi(v_1), \phi(v_2), ..., \phi(v_n)) \in \mathbb{F}_2^{mn}$ (ver (4.1)). A $\phi(v_i)$, $i = 1, ..., n$, les llamamos bloques del vector $e_1 = (\phi(v_1), \phi(v_2), ..., \phi(v))$. Sean $\phi(v_{1_1}), \phi(v_{1_2}), ..., \phi(v_{1_r})$ los bloques que dentro de sus m entradas en e_1 aportan elementos a la longitud de la ráfaga, es decir, al valor de l_1, donde el primer elemento de la ráfaga está en el bloque $\phi(v_{1_1})$ y el último en el bloque $\phi(v_{1_r})$. Observe que el número r^* de bloques que tienen dentro de sus m entradas algún elemento de e_1 distinto de cero es igual a $wt(c_1)$, por la definición de ϕ (ver (4.1)). Ya que puede haber bloques con sus m entradas iguales a cero, que aportan elementos a la ráfaga, y en consecuencia influyen en la longitud de la ráfaga, es decir, en el valor de l_1, se sigue que $r^* \leq r$, o bien, el número de bloques con al menos una entrada no cero que aportan elementos a la ráfaga es menor o igual que el número de bloques que aportan elementos a la ráfaga. Ahora, como la ráfaga empieza en $\phi(v_{1_1})$ y termina en $\phi^*(v_{1_r})$ entonces todos los m elementos de $\phi(v_{1_j})$, $1 < j < r$, están en la ráfaga y sólo algunos elementos de $\phi(v_{1_1})$ y $\phi(v_{1_r})$ (dependiendo en que posición de $\phi(v_{1_1})$ inicia la ráfaga y en que posición de $\phi(v_{1_r})$ termina la ráfaga). El menor valor de la longitud de la ráfaga, es decir, el menor valor de l_1, se obtiene cuando la ráfaga inicia en la m-ésima posición del bloque $\phi(v_{1_1})$ y simultáneamente termina en la primera posición del bloque $\phi(v_{1_r})$, es decir cuando las componentes distintas de cero están confinadas cíclicamente de la siguiente manera:

$$\underbrace{00, ..., 01}_{\phi(v_{1_1})} \phi(v_{1_2}), ..., \phi(v_{1_{r-1}}) \underbrace{10, ..., 0}_{\phi(v_r)}$$

Como los bloques constan de m elementos de e_1, en particular los bloques $\phi(v_{1_j})$, $1 < j < r$, se sigue que:

$$l \geq l_1 = m(r-2) + 2 \geq m(r^* - 2) + 2 = m(wt(c_1) - 2) + 2, \tag{4.3}$$

pues ya habíamos hecho notar que $r^* \leq r$, l_1, $wt(c_1) = r^*$ y por hipótesis $l_1 \leq l$. Ahora, supongamos que $\lceil \frac{l-1}{m} \rceil < wt(c_1)$, entonces $\frac{l-1}{m} \leq \lceil \frac{l-1}{m} \rceil \leq wt(c_1) - 2$, y en consecuencia $l \leq m(wt(c_1) - 2) + 1 <$

$m(wt(c_1)-2)+2$ lo cual contradice a (4.3). Por lo tanto $wt(c_1) \leq \lceil \frac{l-1}{m} \rceil + 1 = \lceil \frac{l-1}{m} + 1 \rceil = \lceil \frac{l-1+m}{m} \rceil \stackrel{*}{=} \lceil \lfloor \frac{n-k}{2} \rfloor \rceil = \lfloor \frac{n-k}{2} \rfloor \stackrel{**}{=} \lfloor \frac{\delta-1}{2} \rfloor$, donde la igualdad señalada con * se debe a la definición de l, y la igualdad señalada con ** se debe a que $\mathcal{C}$ es, en particular, un código MDS. Análogamente, se demuestra que $wt(c_2) \leq \lfloor \frac{\delta-1}{2} \rfloor$. Como $\phi^*(c_1 - c_2) = \phi^*(c_1) - \phi^*(c_2) = e_1 - e_2\phi^* \in (\mathcal{C})$ (pues e_1 y e_2 están en la misma clase de $\phi^*(\mathcal{C})$) y ϕ^* es biyectiva, entonces $c_1 - c_2 \in \mathcal{C}$, así que por el Lema 2.9, se tiene que $c_1 = c_2$, y en consecuencia $e_1 = e_2$. Entonces cada error ráfaga de longitud l o menor pertenece a una clase distinta de $\phi^*(\mathcal{C})$. Del Teorema 4.12 se sigue que $\phi^*(\mathcal{C})$ es un código corrector de errores l-ráfaga. ■

Ejemplo 4.14 Sea $\mathcal{C} = \mathfrak{B}(n = 2^3 - 1 = 7, \delta = 5, \alpha, b)$ un código Reed-Solomon. Ahora, sabemos que $k = n - \delta + 1$ y que δ es igual a la distancia mínima de $\mathcal{C}$, entonces $\mathcal{C}$ es un $[7, 3, 5]$-código lineal 8-ario. Así, el código $\phi^*(\mathcal{C})$, como en el Teorema 2.26, es un $[21, 9]$-código lineal binario. Por el Teorema 4.13, $\phi^*(\mathcal{C})$ es un codigo corrector de errores l-ráfaga, donde $l = 3\lfloor \frac{7-3}{2} \rfloor - 3 + 1 = 4$.

4.3. DECODIFICACIÓN DE CÓDIGOS REED-SOLOMON

Como se ha hecho notar, los códigos Reed-Solomon no son códigos binarios. Entonces el algoritmo presentado al final del capítulo 3 no es apto para los códigos Reed-Solomon y en general en códigos BCH no binarios. A continuación se verán algunos conceptos y resultados enfocados a tratar el problema de decodificación de códigos BCH de cierto tipo, siendo de interés los Reed-Solomon y BCH no binarios, pues hasta ahora no hemos presentado técnicas de decodificación para estos. Más precisamente, se trata de llegar al conocido algoritmo de Berlekamp-Massey y al algoritmo de Forney.

En el resto de la sección consideraremos códigos BCH de longitud n sobre $\mathbb{F}_q$ con $mcd(n, q) = 1$ y ω una raíz n-ésima primitiva sobre $\mathbb{F}_q$. Supongamos que $\mathbf{c}$ es una palabra-código transmitida y que $\mathbf{r}$ es recibida. Entonces $\mathbf{r} = \mathbf{c} + \mathbf{e}$, donde $\mathbf{e}$ es un vector error. La i-ésima componente de la palabara recibida está dada por

$$r_i = c_i + e_i,\ i = 0, ..., n-1,$$

y e_i es diferente de cero en a lo más t valores de i (es decir, se supone que $wt(\mathbf{e}) \leq t$). Por construcción de un código BCH de distancia designada $2t+1$, $\mathfrak{B}_q(n, 2t+1, \omega, b)$, donde $mcd(q, n) = 1$, hay $2t$ componentes de la Transformada de Fourier Discreta de $\mathbf{c}$ (ver (3.5) se la sección 3.3) iguales a cero:

$$\widehat{c}_j = \mathbf{c}(\omega^j) = 0,\ j = b, ..., b + 2t - 1\ ,$$

ya que $\omega^b, \omega^{b+1}, ..., \omega b + 2t - 1$ son raíces de $\mathbf{c(x)}$ según el Lema 3.10. La Transformada de Fourier Discreta de $\mathbf{r}$ tiene componentes:

$$\widehat{r}_j = \sum_{i=0}^{n-1} \omega^{ij} r_i = \sum_{i=0}^{n-1} \omega^{ij}(c_i + e_i) = \sum_{i=0}^{n-1} \omega^{ij} c_i + \sum_{i=0}^{n-1} \omega^{ij} e_i = \widehat{c}_j + \widehat{e}_j.$$

Los síndromes están definidos como aquellas $2t$ componentes $\widehat{r}_j$, $j = b, .., b+2t-1$, correspondientes a las componentes donde $\widehat{c}_j = 0$. Es conveniente indexar los síndromes iniciando en uno.

Definimos:

$$s_j = \widehat{r}_{j+b-1} = \widehat{e}_j,\ j = 1, ..., 2t.$$

Observe que si $b = 1$, la definición de síndrome coincide con la dada en la sección 3.3. El bloque de los $2t$ síndromes es una ventana a través de la cual podemos ver $2t$ componentes de $\widehat{\mathbf{e}}$.

Antes de proseguir con el problema de decodificación, es necesario ver algunas propiedades y conceptos relativos a la Transformada de Fourier discreta.

Definición 4.15 La convolución cíclica, denotada por $f * g$, de dos vectores $f, g \in \mathbb{F}_q^n$ se define como el vector con entradas

$$l_i = \sum_{k=0}^{n-1} f_{((i-k))} g_k,\ i = 0, ..., n-1,$$

donde los paréntesis dobles indican reducción módulo n.

Teorema 4.16 (de Convolución Cíclica) El vector $\mathbf{l}$ está dado por la convolución cíclica de los vectores $f, g \in \mathbb{F}_q^n$ si y sólo si las componentes de la Transformada de Fourier Discreta satisfacen

$$\widehat{l_i} = \widehat{f_j}\widehat{g_j},\ j = 0, ..., n-1.$$

Demostración. [⇒] Si $\mathbf{l}$ es la convolución cíclica de $f, g \in \mathbb{F}_q$, entonces se cumple lo siguiente:

$$l_i = \sum_{k=0}^{n-1} f_{((i-k))} g_k = \sum_{k=0}^{n-1} f_{((i-k))} (n^{-1} \sum_{j=0}^{n-1} \omega^{-jk} \widehat{g_j}) = \sum_{k=0}^{n-1} f_{((i-k))} (n^{-1} \sum_{j=0}^{n-1} \omega^{-jk+ij-ij} \widehat{g_j}) =$$
$$n^{-1} \sum_{j=0}^{n-1} \omega^{-ij} \widehat{g_j} (\sum_{k=0}^{n-1} \omega^{(i-k)j} f_{((i-k))}) = n^{-1} \sum_{j=0}^{n-1} \omega^{-ij} \widehat{g_j} \widehat{f_j}.$$

La última igualdad se justifica del hecho de que $((i-k))$ toma cada valor en $\{0, ..., n-1\}$ si corremos k en el conjunto $\{0, ..., n-1\}$, es decir, $\sum_{k=0}^{n-1} \omega^{(i-k)j} f_{((i-k))}$ es igual a $\sum_{k=0}^{n-1} \omega^{ij} f_i = \widehat{f_j}$ solo que los sumandos están en diferente orden. También, por el (3.7) de la sección 3.3, se tiene que $l_i = n^{-1} \sum_{j=0}^{n-1} \omega^{-ij} \widehat{l_j},\ i = 0, ..., n-1$. Se sigue que la Transformada de Fourier Discreta de $\mathbf{l}$ es el vector con entradas $\widehat{l_j}$ (por definición) y también el vector con entradas $\widehat{f_j}\widehat{g_j}$ con $j = 0, ..., n-1$. Por consiguiente, $\widehat{l_j} = \widehat{f_j}\widehat{g_j}\ j = 0, ..., n-1$.

[⇐] Si $\widehat{l_j} = \widehat{f_j}\widehat{g_j}\ j = 0, ..., n-1$. Entonces, como en la implicación anterior, obtenemos

$$n^{-1} \sum_{j=0}^{n-1} \omega^{-ij} \widehat{g_j} \widehat{f_j} = \sum_{k=0}^{n-1} f_{((i-k))} g_k\ i = 0, ..., n-1,$$

donde $l_i = n^{-1} \sum_{j=0}^{n-1} \widehat{l_j}$ (inciso (3.7) de la sección 3.3), es decir, $l_i = n^{-1} \sum_{j=0}^{n-1} \omega^{-ij} \widehat{g_j} \widehat{f_j}\ i = 0, ...n-1$ (por hipótesis). Por lo tanto, $l_i = \sum_{k=0}^{n-1} f_{((i-k))} g_k$, es decir, $\mathbf{l}$ es la convolución cíclica de $f, g \in \mathbb{F}_q^n$. ■

Definición 4.17 i)Una recursión lineal sobre un campo $\mathbb{F}$ es una expresión de la forma

$$V_k = \sum_{j=1}^{L} \Lambda_j V_{k-j}\ k = L, ...\ (\Lambda_j, V_{k-j} \in \mathbb{F}).$$

La recursión lineal está caracterizada por la longitud L y el vector $\Lambda = (\Lambda_1, ..., \Lambda_L)$ cuyas componentes son llamadas **pesos de conexión**, esto se denota por (Λ, L).

ii) La longitud de la recursión lineal es la longitud de la menor subsucesión de la cual la recursión puede calcular todos los términos de la sucesión.

iii) Se dice que la recursión lineal (Λ, L) produce a la sucesión $\mathbf{V} = \{V_0, V_1, ..., V_L, ...\}$ si los elementos de V satisfacen la expresión en i).

Definición 4.18 Dada una sucesión $\mathbf{V} = \{V_0, V_1, ..., V_{n-1}\}$ de elementos en $\mathbb{F}$, la longitud de la menor recursión lineal que produzca a la sucesión es llamada la **complejidad lineal** de $\mathbf{V}$.

Teorema 4.19 La complejidad lineal de $\mathbf{V} \in \mathbb{F}^n$ ($\{V_0, V_1, ..., V_{n-1}\}$ considerado como un vector $\mathbf{v}$ de longitud n), es igual al peso de la inversa $\mathbf{v}$ de la Transformada de Fourier Discreta ((3.7) de la sección 3.3).

Demostración. Sean $i_1, i_2, ..., i_d$ los índices de las d componentes distintas de cero del vector $\mathbf{v} = (v_0, v_1, ..., v_{n-1})$ y $\mathbf{V} = (V_0, ..., V_{n-1})$ la Transformada de Fourier Discreta de $\mathbf{v}$. Considere el polinomio $\Lambda(x) = \prod_{l=1}^{d}(1 - x^{i_l}) = \sum_{k=0}^{d} \Lambda_k x^k$. Sea $\Lambda = (\Lambda_0, ..., \Lambda_d, 0, 0, ..., 0) \in \mathbb{F}^n$. Sea λ la inversa de la Transformada de Fourier Discreta de Λ, cuyas entradas son:

$$\lambda_i = n^{-1} \sum_{k=0}^{n-1} \omega^{-ik} \Lambda_k = n^{-1} \Lambda(\omega^{-i}) = n^{-1} \prod_{l=1}^{d} (1 - \omega^{-i} \omega^{i_l}).$$

Se sigue que $\lambda_i = 0$ si y sólo si $i \in \{i_1, ..., i_d\}$. Esto es, $\lambda_i = 0$ si y sólo si $v_i \neq 0$. Observe que si $i \notin \{i_1, .., i_d\}$ entonces $v_i = 0$ y si $i \in \{i_1, ..., i_d\}$ entonces (como se hizo notar) $\lambda_i = 0$, de ahí que $\lambda_i v_i = 0$ para cada $i = 0, ..., n-1$. Ahora consideremos la Transformada de Fourier Discreta respecto a ω^{-1} de los vectores Λ y $\mathbf{V}$. Entonces $\widehat{\Lambda}_j = \sum_{i=0}^{n-1} (\omega^{-1})^{ij} \Lambda_i = \sum_{i=0}^{n-1} \omega^{-ij} \Lambda_i = n\lambda_j$, donde la última igualdad se tiene de la inversa de la Transformada de Fourier Discreta. Similarmente, $\widehat{V}_j = \sum_{i=0}^{n-1} \omega^{-ij} V_i = nv_i$. Entonces, $\widehat{\Lambda}_j \widehat{V}_j = (n\lambda_j)(nv_i) = n^2 \lambda_j v_j = 0$ para cada $j = 0, ..., n-1$. Del Teorema de Convolución, se sigue que la Transformada de Fourier Discreta de la convolución cíclica de los vectores Λ y $\mathbf{V}$ es 0, es decir, $\widehat{\Lambda * \mathbf{V}} = 0$, lo cual implica que $\Lambda * \mathbf{V} = 0$ (esto se verifica calculando la inversa de la Transformada de Fourier Discreta a $\widehat{\Lambda * \mathbf{V}} = 0$). De $\Lambda * \mathbf{V}$, se tiene que $\sum_{j=0}^{n-1} \Lambda_j V_{((k-j))} = 0$. Pero $\Lambda_0 = 1$ (las $d+1$ primeras entradas de Λ corresponden a los coeficientes del polinomio $\Lambda(x)$) y $\Lambda_k = 0$ si $k > d$, de ahí que:

$$V_k = -\sum_{j=1}^{d} \Lambda_j V_{((k-j))},$$

así que la complejidad lineal de $\mathbf{v}$ es menor o igual a d. Para probar que la complejidad lineal de $\mathbf{v}$ no es menor que d supongamos que

$$\sum_{j=0}^{L} \Lambda_j V_{((k-j))} = 0, \text{ para } \Lambda = (\Lambda_0, \Lambda_1, ..., \Lambda_L, 0, 0, ..., 0) \in \mathbb{F}^n \ (\Lambda_L \neq 0).$$

Se sigue que $\sum_{j=0}^{n-1} \Lambda_j V_{((k-j))} = 0$, es decir, $\Lambda * \mathbf{V} = 0$. Además, se exhibió que la Transformada de Fourier Discreta respecto a ω^{-1} de Λ y $\mathbf{V}$ son los vectores con entradas $n\lambda_i$ y nv_i, respectivamente. Entonces, por el Teorema de Convolución se sigue que $0 = (n\lambda_i)(nv_i) = n^2 \lambda_i v_i$ para cada $i = 0, ...n-1,$, de ahí que $\lambda_i = 0$ siempre que $v_i \neq 0$. En consecuencia, $d = wt(\mathbf{v}) = |\{i | 0 \leq i \leq n-1 \wedge v_i \neq 0\}| \leq |\{i | 0 \leq i \leq n-1 \wedge \lambda_i = 0\}|$. Consideremos $\Lambda(x) = \sum_{j=0}^{L} \Lambda_j x^j$. Ya que $\lambda_i = n^{-1}\Lambda(\omega^{-i})$ y $\Lambda(x)$ tiene a lo más $L = grad(\Lambda(x))$ ceros, se sigue que $|\{i | 0 \leq i \leq n-1 \wedge \lambda_i = 0\}| \leq grad(\Lambda(x)) = L$. Entonces $d \leq |\{i | 0 \leq i \leq n-1 \wedge \lambda_i = 0\}| \leq L$, de ahí que $d \leq L$. Por lo tanto, la compljejidad lineal de $\mathbf{V}$ es $d = wt(\mathbf{v})$, donde $\mathbf{v}$ es la inversa de la Transformada de Fourier Discreta.

■

4.3.1. ALGORITMO DE BERLEKAMP-MASSEY

Por la propiedad de complejidad lineal del Teorema 4.19, si el vector error $\mathbf{e}$ tiene peso ν, entonces $\widehat{\mathbf{e}}$ satisface la recursión lineal

$$\widehat{e}_k = -\sum_{j=1}^{\nu} \Lambda_j \widehat{e}_{((k-j))},\ k = 0, ..., n-1,$$

para pesos de conexión apropiados $\Lambda_1, ..., \Lambda_\nu$.

Para reescribir esto último supongamos que hay ν errores en la palabra recibida $\mathbf{r}$ en las posiciones $i_1, ..., i_\nu$ (es decir los elementos de $Sop(\mathbf{e})$). El polinomio localizador de errores

$$\Lambda(x) = \prod_{l=1}^{\nu} (1 - x\omega^{i_l})$$

define un vector $\Lambda = (\Lambda_0, \Lambda_1, ..., \Lambda_\nu, 0, ..., 0)$ cuyas entradas corresponden a sus coeficientes, y se tiene $\lambda = (\lambda_0, \lambda_1, ..., \lambda_{n-1})$ la inversa de la Transformada de Fourier Discreta de Λ, cuyas entradas son:

$$\lambda_i = n^{-1} \sum_{j=0}^{n-1} \Lambda_j \omega^{-ij} = n^{-1}\Lambda(\omega^{-i}).$$

Por la forma de $\Lambda(x)$, se tiene que $\Lambda(\omega^{-i}) = 0$ si y sólo si $i \in \{i_1, ..., i_\nu\}$. Entonces, $\lambda_i = 0$ siempre que $e_i \neq 0$. Por lo tanto $\lambda_i e_i = 0$ para cada $i = 0, ..., n-1$. Nuevamente, como en la demostración del Teorema 4.19, si consideramos la Transformada de Fourier respecto a ω^{-1}, se sigue del Teorema 4.16 (de convolución cíclica) lo siguiente:

$$\sum_{j=0}^{\nu} \Lambda_j \widehat{e}_{((k-j))},\ k = 0, ..., n-1,$$

donde el límite superior es ν ya que $\Lambda(x)$ tiene grado ν. Como $\Lambda_0 = 1$ se tiene lo antes mencionado. Además, si $\nu \leq t$, entonces $\Lambda_j = 0$ para $j = \nu + 1, ..., t$, así que la recursión lineal se puede escribir como:

$$\widehat{e}_k = -\sum_{j=1}^{t} \Lambda_j \widehat{e}_{((k-j))},\ k = 0, ..., n-1,$$

lo cual se puede preferir pues t es conocido y fijo. Este sistema de n ecuaciones involucra $2t$ componentes conocidas de $\widehat{e}$ dadas por los $2t$ síndromes, y $n - t$ variables, de las cuales t son los coeficientes desconocidos de $\Lambda(x)$ y $n - 2t$ componentes desconocidas de $\widehat{e}$. De las n ecuaciones, hay t ecuaciones que involucran sólo a las componentes conocidas de Λ, que son expresadas por los síndromes $s_j = \widehat{e}_{b+j-1}$ para $j = 1, ..., 2t$, y las t componentes desconocidas de Λ, que son los coeficientes del polinomio $\Lambda(x)$. Esto es, las t ecuaciones

$$s_k = -\sum_{j=1}^{t} \Lambda_j s_{((k-j))},\ k = t+1, ..., 2t,$$

involucran sólo a los síndromes, los cuales son conocidos, y las t componentes desconocidas de Λ.

Esta recursión lineal, escrita en la forma

$$s_k = -\sum_{j=1}^{\nu} \Lambda_j s_{k-j}\ k = \nu + 1, ..., 2t,$$

debe ser resuelta para Λ usando el menor valor posible de ν. Para decodificar un código BCH o un código Reed-Solomon, se resuelve el siguiente problema. Encontramos pesos de conexión $(\Lambda_1, ..., \Lambda_\nu)$ para el menor $\nu \leq t$ para el cual el sistema de ecuaciones

$$s_k = -\sum_{j=1}^{\nu} \Lambda_j s_{k-j} \ k = \nu+1, ..., 2t$$

tiene solución.

El algoritmo de Berlekamp-Massey resuelve el siguiente problema modificado. Encuentra pesos de conexión $(\Lambda_1, ..., \Lambda_\nu)$ para el menor $\nu \leq 2t$ para el cual el sistema de ecuaciones

$$s_k = -\sum_{j=1}^{\nu} \Lambda_j s_{k-j} \ k = \nu+1, ..., 2t$$

tiene una solución. En el problema modificado, valores del entero ν tan grandes como $2t$ son permitidos, así que para una sucesión arbitraria $s_1, ..., s_{2t}$, el problema debe tener solución. En particular, el valor $\nu = 2t$, junto con valores arbitrarios para $\Lambda_1, ..., \Lambda_{2t}$, siempre satisfacen el sistema de ecuaciones.

El problema modificado no es el mismo que el original, pero si el problema original tiene solución, será la misma para el problema modificado.

El problema modificado puede ser visto como la tarea de encontrar la menor recursión lineal $(\Lambda(x), \nu)$ que produzca a la sucesión de síndromes. Si $\Lambda(x) = 0$ ó $grad(\Lambda(x)) > t$ entonces tal solución no es de nuestro interés, pues no es solución del problema original y el objetivo es corregir a lo más t posiciones del error.

A continuación enunciamos el algoritmo de Berlekamp-Massey, que constituye la primera etapa en el problema de decodificación de un código BCH no binario ó un código Reed-Solomon. La demostración se puede consultar en [2].

Teorema 4.20 (Algoritmo de Berlekamp-Massey) Sean $s_1, ..., s_{2t} \in \mathbb{F}$. Con las condiciones iniciales $\Lambda^{(0)}(x) = 1$, $B^{(0)}(x) = 1$, y $L_0 = 0$, sean las siguientes ecuaciones para $r = 1, ..., 2t$ usadas iterativamente para calcular $\Lambda^{(2t)}(x)$:

$$\Delta_r = \sum_{j=0}^{n-1} \Lambda_j^{(r-1)} s_{r-j}$$

$$L_r = \delta_r(r - L_{r-1}) + (1 - \delta_r)L_{r-1}$$

$$\begin{pmatrix} \Lambda^{(r)}(x) \\ B^{(r)}(x) \end{pmatrix} = \begin{pmatrix} 1 & -\Delta_r x \\ \Delta_r^{-1}\delta_r & (1-\delta_r)x \end{pmatrix} \begin{pmatrix} \Lambda^{(r-1)}(x) \\ B^{(r-1)}(x) \end{pmatrix}$$

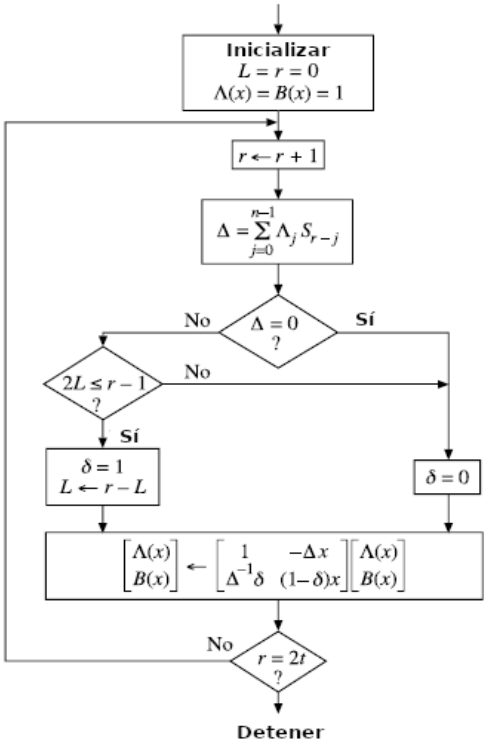
donde $\delta_r = 1$ si de manera simultánea $\Delta \neq 0$ y $2L_{r-1} \leq r-1$, y en otro caso $\delta_r = 0$. Entonces $(\Lambda^{(2t)}(x), L_{2t})$ es una recursión lineal de menor longitud que produce a $s_1, ..., s_{2t}$.

4.3.2. EXPLICACIÓN DEL ALGORITMO DE BERLEKAMP-MASSEY

El algoritmo es bastante creativo. Para nuestros fines, lo más valioso será obtener $\Lambda^{(2t)}(x)$ pues este es el polinomio localizador de errores, es decir, $\Lambda^{(2t)}(x) = \prod_{i \in Sop(\mathbf{e})} (1 - \omega^i x)$. Tener el polinomio localizador de errores es clave para determinar $Sop(\mathbf{e})$, ya que ω^{-i} es raíz de tal polinomio si y sólo si $i \in Sop(\mathbf{e})$. Sin embargo, en el caso no binario, determinar $Sop(\mathbf{e})$ no es suficiente para determinar a $\mathbf{e}$, pues hay varias posibilidades para sus entradas distintas de cero. Para ello, se complementará el algoritmo de Berlekamp-Massey con algún algoritmo que permita saber el valor de las entradas no cero del vector error $\mathbf{e}$.

En cuanto al algoritmo, se hacen $2t$ iteraciones. Cabe resaltar que $s_1, ..., s_{2t}$ no presumen de alguna propiedad especial. En la r-ésima iteración $(1 \leq r \leq 2t)$ se va encontrando a una recursión lineal $(\Lambda^{(r)}(x), L_r)$ de longitud menor tal que produzca $s_1, ..., s_r$. Al inicio de la r-ésima iteración se han construido las recursiones $(\Lambda^{(1)}, L_1), (\Lambda^{(2)}, L_2), ..., (\Lambda^{(r-1)}, L_{r-1})$. El principal truco del algoritmo de Berlekamp-Massey es usar esas iteraciones anteriores para producir una nueva recursión lineal de longitud menor $(\Lambda^{(r)}(x), L_r)$ que produzca a la sucesión $s_1, ..., s_r$.

La estructura del algoritmo puede visualizarse a través del siguiente diagrama de flujo:



Como se hizo notar, no es suficiente determinar mediante el algoritmo de Berlekamp-Massey a $Sop(\mathbf{e})$, por lo cual buscaremos un algoritmo complementario que termine por resolver la tarea de decodificar.

4.3.3. ALGORITMO DE FORNEY

Sean $s(x) = \sum_{j=1}^{2t} s_j x^j$ el síndrome polinomial, $\Lambda(x)$ el polinomio localizador de errores obtenido con el algoritmo de Berlekamp-Massey, donde $grad(\Lambda(x)) = \nu \leq t$, y $\Lambda = (\Lambda_0, \Lambda_1, ..., \Lambda_{2t}, 0, .., 0)$ el vector de longitud n de los coeficientes de $\Lambda(x)$. Entonces, $\Lambda(x)s(x)$ siempre se pude descomponer de la siguiente forma:

$$\Lambda(x)s(x) = \Gamma(x) + x^{\nu+1}O(x) + x^{2t+1}\Theta(x),$$

donde $grad(\Gamma(x)) \leq \nu$ $(\Lambda(x)s(x) = \sum_{i=0}^{\nu} a_i x^i + \sum_{i=\nu+1}^{2t} a_i x^i + \sum_{i=2t+1}^{grad(\Lambda(x)s(x))} a_i x^i$, de ahí se puede obtener la descomposición que se afirma). Ya que $(\Lambda(x), \nu)$ producen a $s_1, s_2, , , .s_{2t}$ entonces

$$\sum_{j=0}^{\nu} \Lambda_j s_{k-j} = 0,\ k = \nu+1, ..., 2t,$$

que son precisamente las entradas correspondientes al producto $x^{\nu+1}O(x)$ (pues $x^{\nu+1}O(x) = \sum_{j=\nu+1}^{2t} a_j x^j$, donde a_j son coeficientes del producto $\Lambda(x)s(x)$ para $j = \nu+1, ..., 2t$). Se sigue que

$$\Lambda(x)s(x) = \Gamma(x) + x^{2t+1}\Theta(x),\ grad(\Gamma(x)) \leq \nu.$$

El polinomio $\Lambda(x)$ es el polinomio evaluador del error. Observe que $s(x)$ es conocido y si $\Lambda(x)$ también lo es tras aplicar el algoritmo de Berlekamp-Massey, entonces para obtener a $\Gamma(x)$ simplemente podemos calcular $\Lambda(x)s(x) \pmod{x^{2t+1}}$, es decir, $\Gamma(x) = \Lambda(x)s(x) \pmod{x^{2t+1}}$.

Teorema 4.21 El polinomio evaluador del error puede escribirse como

$$\Gamma(x) = \sum_{i=1}^{\nu} xY_i X_i^b \prod_{j\neq i}(1 - X_j x),$$

donde $Y_i = e_i$ (la i-ésima entrada del vector error $\mathbf{e}$) y $X_i = \omega^i$.

Demostración. Recordemos que $s(x) = \sum_{j=1}^{2t} s_j x^j = \sum_{j=1}^{2t} \widehat{e}_{b+j-1}x^j = \sum_{j=1}^{2t} \sum_{i\in Sop(\mathbf{e})} Y_i X_i^{b+j-1} x^j$. Entonces, $\Gamma(x) = [\sum_{j=1}^{2t} \sum_{i\in Sop(\mathbf{e})} Y_i X_i^{b+j-1}x^j][\prod_{l\in Sop(\mathbf{e})}(1 - X_l x)] \pmod{x^{2t+1}}$

$= \sum_{i\in Sop(\mathbf{e})} xX_i^b Y_i[(1 - X_i x)\sum_{j=1}^{2t}(X_i x)^{j-1}] \prod_{\substack{l\in Sop(\mathbf{e})\\ l\neq i}}(1 - X_l x) \pmod{x^{2t+1}}$

$= \sum_{i\in Sop(\mathbf{e})} xX_i^b Y_i[1 - (X_i x)^{2t}] \prod_{\substack{l\in Sop(\mathbf{e})\\ l\neq i}}(1 - X_l x) \pmod{x^{2t+1}}$

$= \sum_{i\in Sop(\mathbf{e})} xX_i^b Y_i \prod_{\substack{l\in Sop(\mathbf{e})\\ l\neq i}}(1 - X_l x),$

donde la última igualdad se tiene de $x = x(1 - X_i x)^{2t} \pmod{x^{2t+1}}$. ■

Siguiendo con la notación del Teorema 4.21 se tiene el importante resultado conocido como el Algoritmo de Forney.

Teorema 4.22 (Algoritmo de Forney) Sea $l \in Sop(\mathbf{e})$. Entonces, el valor de l está dado por

$$Y_l = \frac{X_l^{1-b}\Gamma(X_l^{-1})}{\prod_{\substack{j\in Sop(\mathbf{e})\\ j\neq l}}(1 - X_j X_l^{-1})}.$$

Demostración. Por el Teorema 4.21, $\Gamma(x) = \sum_{i\in Sop(\mathbf{e})} xX_i^b Y_i \prod_{\substack{l\in Sop(\mathbf{e})\\ l\neq i}}(1 - X_l x)$. Se sigue que $\Gamma(X_l^{-1}) = \sum_{i\in Sop(\mathbf{e})} X_l^{-1} X_i^b Y_i \prod_{\substack{l\in Sop(\mathbf{e})\\ l\neq i}}(1 - X_l X_l^{-1})$. En la anterior suma, cada término se hace cero excepto cuando $i = l$ (si $i \neq l$ entonces $\prod_{\substack{l\in Sop(\mathbf{e})\\ l\neq i}}(1 - X_l X_l^{-1})$ tiene como factor a $(1 - X_l X_l^{-1}) = 0$). Entonces, $\Gamma(X_l^{-1}) = X_l^{b-1} Y_l \prod_{\substack{j\in Sop(\mathbf{e})\\ j\neq l}}(1 - X_j X_l^{-1})$, de ahí que $Y_l = \frac{X_l^{1-b}\Gamma(X_l^{-1})}{\prod_{\substack{j\in Sop(\mathbf{e})\\ j\neq l}}(1 - X_j X_l^{-1})}$. ■

Para concluir resumimos el proceso de decodificación. Primero calculamos los síndromes $s_1, ..., s_{2t}$. Posteriormente, implementamos el Algoritmo de Berlekamp-Massey para obtener al polinomio localizador de errores $\Lambda(x)$. Con $\Lambda(x)$ explícito, entonces determinamos $Sop(\mathbf{e})$ analizando que elementos de la forma ω^{-i} son raíces de $\Lambda(x)$ (pues esto índica que $i \in Sop(\mathbf{e})$). Una vez determinado $Sop(\mathbf{e})$, aplicamos el algoritmo de Forney para obtener los valores del error, es decir, se determina a $\mathbf{e}$. Finalmente, se estima la palabra transmitida como $\mathbf{c} = \mathbf{r} - \mathbf{e}$.

Como comentario adicional, si tenemos un código BCH binario entonces basta usar el Algoritmo de Berlekamp-Massey para decodificar, pues es suficiente con conocer a $Sop(\mathbf{e})$.

Capítulo 5

IMPLEMENTACIONES

Uno de los principales objetivos de este trabajo es aterrizar algunas ideas teóricas en implementaciones explicitas y concretas. En los Capítulos 3 y 4 se analizaron propiedades y características de los códigos BCH y códigos Reed-Solomon, respectivamente. Por sus propiedades, estos códigos son bastante útiles en distintos campos de la vida moderna, principalmente a los relacionados con cuestiones de transmisión de datos digitales, como lo es la telecomunicación. En ambos capítulos se analizaron técnicas que permiten afrontar la díficil tarea de decodificar, es decir, algoritmos que permiten detectar y corregir errores para los códigos pertinentes. Estos algoritmos pueden ser llevados a lenguajes de programación adecuados que permitan al usuario utilizarlos siempre que se desee, además de que esto permitiría una mejor comprensión de los conceptos abordados.

En la programación matemática existen diversas herramientas que son de bastante ayuda para resolver problemas en estos tiempos modernos. Entre ellos destacan varias por sus características y enfoque, además de ser un factor determinante en el uso de alguna herramienta u otra el ser software libre o no. SageMath es un sistema algebraico computacional que permite realizar distintas tareas en el campo de las matemáticas, el cual está constituido por paquetes matemáticos altamente constatados y es basado en Python. Dentro de las virtudes de SageMath se encuentran las capacidades de realizar actividades en el contexto del álgebra moderna, es decir, SageMath permite trabajar con estructuras algebraicas como grupos, anillos, campos, módulos, retículas y más. Con SageMath podemos accesar a conceptos propios de estas estructuras, como morfismos, construcciones especiales, etc. Con esta herramienta, es posible implementar o estructurar algunas ideas en el marco del álgebra moderna. En particular, los algoritmos analizados son posibles de implementar en esta herramienta computacional.

Para la implementación de los algoritmos de decodificación de los Capítulos 3 y 4 vamos a usar la capacidad de SageMath en el uso de campos finitos y anillos de polinomios sobre estos. Estos algoritmos se pueden implementar no sólo en SageMath, también en cualquier otra herramienta que cuente con librerias que soporten aritmética en campos finitos y construcciones relativas a estos.

5.1. ALGORITMO DE DECODIFICACIÓN DEL CAPÍTULO 3

En la Sección 3.5 se vio una técnica que permite la decodificación de códigos BCH binarios primitivos en el sentido estricto. Las ideas inmersas en este algoritmo se pueden aterrizar mediante la elaboración de un programa respectivo en SageMath.

A continuación se presenta el código fuente del programa respectivo al algoritmo de decodificación revisado en el Capítulo 3.

Código 5.1: Algoritmo de decodificación del Capítulo 3

```
from sage.functions.log import logb
import ast


R.<z> = GF(2)['z']


def SindromePolinomial(R,t):
    P = 0
    for i in range(len(R)):
        P += R[i]*x^i

    s = [None]*(2*t)
    for i in range(2*t):
        s[i]= P(y^(i+1))

    S = 0
    for i in range(2*t):
        S += s[i]*x^i

    return S

def AlgoritmoEuclides(a,S,mu,nu):
    r=[a,S]
    u=[x^0,0]
    v=[0,x^0]
    q=[]
    i=1

    rows = [["i","ui", "vi", "ri", "qi"],[-1,u[0],v[0],r[0], "----"]]
    rows.append([0,u[1],v[1],r[1],"----"])
    while( 0<=r[i].degree()):
        c,z=r[i-1].quo_rem(r[i])
        r.append(z)
        q.append(c)
        u.append(u[i-1]-q[i-1]*u[i])
        v.append(v[i-1]-q[i-1]*v[i])
        rows.append([i,u[i+1],v[i+1],r[i+1],q[i-1]])
        i+=1

    j=1
    if(mu+nu==(a.degree()-1) and 0<= mu and 0<=nu):
        while(mu<v[j].degree() or nu<r[j].degree()):
            j+=1

    Tabla = open("C:\\Users\\Luis\\Documents\\Euclides.txt","w")
    Tabla.write(str(table(rows)))
    Tabla.close()

    return (v[j],r[j])



def PalabraCorregida(R,v,r):
    sig = (v(0)^(-1))*v
    lamb = (r(0)^(-1))*r
    I=[]
    for i in range(len(R)):
        if (sig(y^(-i))==0):
            I.append(i)


    for i in range(len(I)):
        R[I[i]] += y^0
```

```
    C = R

    return C

print("DECODIFICACION DE CODIGOS BCH BINARIOS")

n = int(input("INGRESE LA LONGITUD DEL CODIGO: "))

l = logb(n+1,2)

while(l != floor(l)):
    n = int(input("LA LONGITUD NO ES DE LA FORMA n = 2^m -1"))
    l = logb(n+1,2)

delta = int(input("INGRESE LA DISTANCIA DESIGNADA:"))

while(delta%2 != 1):
    delta = int(input("LA DISTANCIA DESIGNADA DEBE SER IMPAR"))

t =int((delta-1)/2)
mu = t
nu=t-1


print "EN ESTE CASO, t = ",t

print "INGRESE EL POLINOMIO MINIMO QUE DEFINE AL CAMPO CON ", 2^l, "ELEMENTOS "
PPcadena = raw_input()
PolinomioPrimitivo = R(PPcadena)


while (PolinomioPrimitivo.is_primitive() == 0 or PolinomioPrimitivo.degree()!= l):
    print "EL POLINOMIO INGRESADO NO ES PRIMITIVO O NO ES DE GRADO ", l,"
    PPcadena = raw_input()
    PolinomioPrimitivo = R(PPcadena)

list = raw_input("INGRESE LA PALABRA RECIBIDA: R = ")
PalabraRecibida = ast.literal_eval(list)

while(len(PalabraRecibida)!= n):
    list = raw_input("LA PALABRA RECIBIDA NO TIENE LA LONGITUD DADA AL INICIO")
    PalabraRecibida = ast.literal_eval(list)


F.<y> = GF(2^l,modulus = PolinomioPrimitivo)
R.<x> = PolynomialRing(F)

a = x^(2*t)

S = SindromePolinomial(PalabraRecibida,t)


if S==0:
    print "LA PALABRA RECIBIDA YA ERA UNA PALABRA CODIGO"
else:
    v,r=AlgoritmoEuclides(a,S,mu,nu)
    C = PalabraCorregida(PalabraRecibida,v,r)
    print "LA PALABRA CORREGIDA ES C = ", C
```

5.1.1. EXPLICACIÓN DEL CÓDIGO

El código mostrado anteriormente consta de funciones y sentencias que de forma conjunta posibilitan realizar la tarea de decodificar códigos binarios BCH primitivos en el sentido estricto.

En el programa principal, es decir, después de las funciones definidas, se hacen sentencias sobre el ingreso de los datos pertinentes para implementar el algoritmo. Primero se tienen sentencias sobre ingresar la longitud del código, la cual debe ser de la forma 2^m-1 por el tipo de códigos que estamos tratando. Si la longitud no se ingresa adecuadamente se pedirá que se vuelva ingresar hasta que sea correcta. Posteriormente, las sentencias corresponden al ingreso de la distancia designada, la cual debe ser impar como se hizo en el análisis en el Capítulo 3. De manera similar a la longitud, si la distancia designada no es impar, se pedirá que se ingresé nuevamente hasta que sea correcta. En seguida, hay sentencias sobre el ingreso y verificación del polinomio primitivo sobre $\mathbb{F}_2$ que defina al campo $\mathbb{F}_{2^l}$, donde l es la longitud del código, es decir, el polinomio en $\mathbb{F}_2[x]$ que tiene como raíz a un elemento primitivo de $\mathbb{F}_{2^l}$. Luego, las sentencias son relativas al ingreso de la palabra recibida. A partir de aquí las sentencias son llamados de las funciones previamente definidas para aplicar las ideas del algoritmo de decodificación. Después de esto último, las sentencias relativas a la salida del programa, es decir, a presentar el estimado de la palabra transmitida. Respecto a las funciones definidas en el código podemos decir lo siguiente:

La función llamada SindromePolinomial tiene como objetivo calcular y devolver el síndrome polinomial. Los argumentos de esta función son la palabra recibida y la t implícita en la distancia designada impar $(2t+1)$. La función AlgoritmoEuclides realiza lo relativo a la Definición 3.25. Además, al final de está función se crea un archivo de texto que muestra el procedimiento completo, pues tal información puede ser de interés. Tal archivo se puede guardar en el directorio de la preferencia del usuario. Los argumentos de esta función son el polinomio x^{2t}, el síndrome polinomial, $\mu = t$ y $\nu = t-1$, esto en el sentido de la Definición 3.25. Finalmente, la función PalabraCorregida calcula el polinomio localizador de errores en base a los retornado por las anteriores funciones y con esto determina $Sop(\mathbf{e})$, lo cual es suficiente para encontrar a $\mathbf{e}$ (pues se están tratando códigos BCH binarios).

Ahora mostramos un ejemplo de como correr el programa dado anteriormente en base a los datos del Ejemplo 3.28.

Ejemplo 5.1

```
PROCESO DE DECODIFICACION DE CODIGOS BINARIOS PRIMITIVOS BCH EN EL SENTIDO ESTRICTO
INGRESE LA LONGITUD DEL CODIGO: 15
INGRESE LA DISTANCIA DESIGNADA:7
EN ESTE CASO, t =  3
INGRESE EL POLINOMIO MINIMO (DE GRADO 4 )SOBRE EL CAMPO BINARIO DEL ELEMENTO PRIMIVO DEL CAMPO CON  16 ELEMENTOS:
z^4+z+1
INGRESE LA PALABRA RECIBIDA: R = [1,1,0,0,0,0,1,1,0,1,1,0,1,0,1]
LA PALABRA CORREGIDA ES C =  [1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1]
```

Como se puede ver, hemos llegado a lo obtenido en el ejemplo antes mencionado.

Ejemplo 5.2 Este es un ejemplo en el que se ha transmitido la palabra-código cero del código BCH $\mathfrak{B}_2(15, 7, \alpha, 1)$, donde $\alpha \in \mathbb{F}_{16}$ es raíz del polinomio $x^4 + x + 1 \in \mathbb{F}_2[x]$ y se recibe la palabra [0,0,1,0,0,1,0,1,0,0,0,0,0,0,0].

```
PROCESO DE DECODIFICACION DE CODIGOS BINARIOS PRIMITIVOS BCH EN EL SENTIDO ESTRICTO
INGRESE LA LONGITUD DEL CODIGO: 15
INGRESE LA DISTANCIA DESIGNADA:7
EN ESTE CASO, t =  3
INGRESE EL POLINOMIO MINIMO (DE GRADO 4 )SOBRE EL CAMPO BINARIO DEL ELEMENTO PRIMIVO DEL CAMPO CON  16 ELEMENTOS:
z^4 + z+1
INGRESE LA PALABRA RECIBIDA: R = [0,0,1,0,0,1,0,1,0,0,0,0,0,0,0]
LA PALABRA CORREGIDA ES C =  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Eso muestra que el programa tuvo la salida esperada.

5.2. ALGORITMO DE DECODIFICACIÓN DEL CAPÍTULO 4

En el Capítulo 3 se desarrollo una técnica que permitía abordar el problema de la decodificación de códigos binarios BCH primitivos en el sentido estricto. Sin embargo, por definición, los códigos Reed-Solomon no son códigos binarios lo que hace buscar técnicas alternativas que permitan su decodificación. En el Capítulo 4 se reviso el conocido Algoritmo de Berlekamp-Massey que permite encontrar el polinomio localizador de errores, además de un algoritmo complementario que es el Algoritmo de Forney que permite determinar el valor de los errores. En particular, estos algoritmos son populares y útiles en la decodificación de códigos BCH no binarios, en particular los códigos Reed-Solomon.

A continuación se presenta el código fuente del programa respectivo a los algoritmos de decodificación revisados en el Capítulo 4.

Código 5.2: Algoritmo de decodificación del Capítulo 4

```
from sage.functions.log import logb
import ast

def RecibirPalabra(n):
    print "INGRESE LA PALABRA RECIBIDA (EN FORMA POLINOMIAL): "
    PalabraRecibidaPolinomial = R(raw_input())


    while n <= PalabraRecibidaPolinomial.degree():
        print "LA PALABRA RECIBIDA TIENE LONGITUD INCORRECTA, REINTENTELO:"
        PalabraRecibidaPolinomial = R(raw_input())

    PalabraRecibida = PalabraRecibidaPolinomial.list()

    for i in range(len(PalabraRecibida),n):
        PalabraRecibida.append(0)

    return PalabraRecibida

def Sindromes(r,t):
    n = len(r)
    S = vector([0*y^0]*(2*t))
    for i in range(2*t):
        for j in range(n):
            S[i] += r[j]*(y^(i+1))^j

    return S

def BerlekampMassey(S,t):
```

```
    L=0
    D = x^0
    B = x^0
    Dcoefs = vector(D.list())

    rows = [["r","Discrepancia","B(x)","Lambda(x)","L"],["0","---","1","1","0"]]

    for r in range(1,2*t+1):
        discrepancia = 0*y^0
        nu = len(Dcoefs)

        for i in range(nu):
            discrepancia += Dcoefs[i]*S[r-i-1]

        Daux = D

        D = D - discrepancia*x*B
        Dcoefs = vector(D.list())

        if discrepancia != 0*y^0 and 2*L <= r-1:
            L= r-L
            B = (discrepancia^-1)*Daux

        else:
            B = x*B

        rows.append([r,discrepancia,B,D,L])

    Tabla = open("C:\\Users\\Luis\\Documents\\BerlekampMassey.txt","w")
    Tabla.write(str(table(rows)))
    Tabla.close()

    return D

def LocalizacionErrores(D,n):
    Dcoefs = D.list()
    Localizaciones = []
    for i in range(n):
        evaluacion = 0*y^0

        for j in range(len(Dcoefs)):
            evaluacion += D[j]*(y^(-i))^j

        if evaluacion == 0:
            Localizaciones.append(i)

    return Localizaciones

def Forney(D,S,Localizaciones,t):
    ValorErrores = [None]*len(Localizaciones)
    SindromePol = 0*x^0
    for i in range(2*t):
        SindromePol += S[i]*x^(i+1)

    z,Gamma = (D*SindromePol).quo_rem(x^(2*t+1))

    for i in range(len(Localizaciones)):
        prod = y^0
        for j in range(len(Localizaciones)):
            if Localizaciones[j] != Localizaciones[i]:
                prod *= (1-y^(Localizaciones[j] - Localizaciones[i]))

        ValorErrores[i] = Gamma(y^(-Localizaciones[i]))/prod

    return ValorErrores
```

```
def PalabraCorregida(PalabraRecibida,Localizaciones,ValorErrores):
    c = PalabraRecibida

    for i in range(len(Localizaciones)):
        c[Localizaciones[i]] = PalabraRecibida[Localizaciones[i]] - ValorErrores[i]

    return c

print "DECODIFICACION DE CODIGOS q-ARIOS BCH PRIMITIVOS EN EL SENTIDO ESTRICTO"
q = int(input("INGRESE EL VALOR DE q: "))+0

while q.is_prime_power() != 1:
    q= int(input("q NO ES POTENCIA DE UN PRIMO, INGRESE SU NUEVAMENTE VALOR:"))+0

n = int(input("INGRESE LA LONGITUD n DEL CODIGO: "))

m = logb(n+1,q)

while m!=floor(m) :
    n = int(input("LA LONGITUD NO ES DE LA FORMA q^m-1 "))
    m = logb(n+1,q)

R.<z> = GF(q)['z']

delta = int(input("INGRESE LA DISTANCIA DESIGNADA: "))

while(delta%2 != 1):
    delta = int(input("LA DISTANCIA DESIGNADA DEBE SER IMPAR, REINTENTELO:"))

t = int((delta-1)/2)

F.<y> = GF(q^m)
R.<x> = PolynomialRing(F)


PalabraRecibida = RecibirPalabra(n)

print "PALABRA RECIBIDA ", PalabraRecibida
S = Sindromes(PalabraRecibida,t)

print "VECTOR DE SINDROMES: ",S
D = BerlekampMassey(S,t)
Localizaciones = LocalizacionErrores(D,n)

print "LOCALIZACIONES DE LOS ERRORES = ", Localizaciones


if q == 2:
    ValorErrores = [y^0]*len(Localizaciones)
else:
    ValorErrores = Forney(D,S,Localizaciones,t)

print "VALOR DE LOS ERRORES = ", ValorErrores

c = PalabraCorregida(PalabraRecibida,Localizaciones,ValorErrores)
print "EL ESTIMADO DE LA PALABRA TRANSMITIDA ES C = ", c
```

5.2.1. EXPLICACIÓN DEL CÓDIGO

El código mostrado anteriormente consta de funciones y sentencias que de forma conjunta posibilitan realizar la tarea de decodificar códigos BCH primitivos en el sentido estricto no binarios, en particular códigos Reed-Solomon.

En el programa principal (localizado después de la definición de las funciones) hay sentencias relativas al ingreso de los datos pertinentes y llamados de funciones definidas para la implementación del algoritmo. Primero, como no estamos asumiendo que el código sea binario, entonces la primera sentencia es para pedir el valor de q de nuestro código q-ario BCH. Ya que q es la cardinalidad del campo finito $\mathbb{F}_q$, entonces q debe ser una potencia de un número primo, en caso contrario se inicia un ciclo que termina hasta que q cumpla tal condición. Luego, se pide el ingreso de la longitud del código, la cual debe ser de la forma $q^m - 1$ por el tipo de códigos considerados, ya que en caso contrario se inicia un ciclo que termina hasta que la longitud del código cumple tal condición. Posteriormente, se pide el ingreso de la distancia designada del código, la cual debe ser impar por las consideraciones al desarrollar el algoritmo. En seguida se construyen el campo finito que es el campo de descomposición de nuestra raíz n-ésima primitiva sobre $\mathbb{F}_q$ y el anillo de polinomios con coeficientes en el campo mencionado. Finalmente, las últimas sentencias son llamados de funciones previamente definidas, más precisamente está el llamado al Algoritmo de Berlekamp-Massey para encontrar el polinomio localizador de errores y al Algoritmo de Forney para determinar el valor de los errores. Esto de forma conjunta permite hacer un estimado de la palabra transmitida y presentarla al usuario. Cabe destacar que al final del código tenemos una condicional respecto al valor de q. En caso de que q sea 2, es decir, el código sea binario, entonces sólo aplicamos el Algoritmo de Berlekamp-Massey, ya que sólo basta encontrar el polinomio localizador para encontrar $Sop(\mathbf{e})$ y así determinar $\mathbf{e}$. Si $q \neq 2$, como en el caso de los códigos Reed-Solomon, entonces es necesario utilizar de manera conjunta al Algoritmo de Berlekamp-Massey y el Algoritmo de Forney para determinar $\mathbf{e}$ y lograr estimar la palabra transmitida.

Respecto a las funciones definidas en el código podemos decir lo siguiente:

La función RecibirPalabra tiene como objetivo captar lo que el usuario ingrese como palabra recibida y verificar que sea consistente con los datos del código, más precisamente con la longitud. Aquí, a diferencia del algoritmo del Capítulo 3, el ingreso de la palabra recibida se hace en forma polinomial. El único argumento de esta función es la longitud del código. Teniendo la palabra recibida es posible calcular los síndromes, esto es lo que hace la función llamada Sindromes.
La función BerlekampMassey implementa el algoritmo de Berlekamp-Massey como se vio en el Teorema 4.20. La salida de esta función es el polinomio localizador de errores. Los argumentos son los síndromes (calculados con la función Sindromes) y el entero positivo t que está implícitio en la distancia designada impar $(2t + 1)$. Además, se crea un archivo de texto que contiene el proceso iterativo de Berlekamp-Massey el cual contiene información que puede ser de utilidad. Tal archivo se puede guardar en el directorio de preferencia del usuario. Luego tenemos la función LocalizacionErrores que determina a $Sop(\mathbf{e})$ analizando que potencias α^{-i} son raíces del polinomio localizador, pues esto equivale a tener que $i \in Sop(\mathbf{e})$. Los argumentos de esta función son el polinomio localizador y la longitud del código para tener control sobre los ciclos realizados. En seguida, se tiene la función llamada Forney que implementa el Algoritmo de Forney para determinar los valores del error. Esta función se basa en el Teorema 4.22. Los argumentos son el polinomio localizador, el vector de síndromes, un vector correspondiente a las posiciones donde el error es distinto de cero (calculado con la función LocalizacionErrores) y el entero positivo t. La salida de la función son los valores del error. Finalmente se tiene la función PalabraCorregida que con las salidas de la funciones BerlekampMassey y Forney realiza el estimado de la palabra transmitida. La salida es precisamente la palabra transmitida.

Ahora se presentarán ejemplos de como correr el programa con datos dados.

Ejemplo 5.3 En este caso se transmite la palabra-código cero del código BCH $\mathfrak{B}_2(15, 7, \alpha, 1)$ donde α es raíz del polinomio $x^4 + x + 1 \in \mathbb{F}_2[x]$. La palabra recibida es $[0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0]$.

```
DECODIFICACIÓN DE CODIGOS q-ARIOS BCH PRIMITIVOS EN EL SENTIDO ESTRICTO
INGRESE EL VALOR DE q: 2
INGRESE LA LONGITUD n DEL CODIGO: 15
INGRESE LA DISTANCIA DESIGNADA: 7
INGRESE LA PALABRA RECIBIDA (EN FORMA POLINOMIAL):
x^2 +x^5 + x^7
PALABRA RECIBIDA  [0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0]
VECTOR DE SINDROMES :  (y^3 + 1, y^3 + y^2 + 1, 1, y^3 + y^2 + y, y^2 + y, 1)
LOCALIZACIONES DE LOS ERRORES =  [2, 5, 7]
VALOR DE LOS ERRORES =  [1, 1, 1]
EL ESTIMADO DE LA PALABRA TRANSMITIDA ES C =  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Ejemplo 5.4 Ahora implementemos este algoritmo para el Ejemplo 5.1.

```
DECODIFICACIÓN DE CODIGOS q-ARIOS BCH PRIMITIVOS EN EL SENTIDO ESTRICTO
INGRESE EL VALOR DE q: 2
INGRESE LA LONGITUD n DEL CODIGO: 15
INGRESE LA DISTANCIA DESIGNADA: 7
INGRESE LA PALABRA RECIBIDA (EN FORMA POLINOMIAL):
1 + x + x^6 + x^7 + x^9 + x^10 + x^12 + x^14
PALABRA RECIBIDA  [1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1]
VECTOR DE SINDROMES :  (y^3 + y^2 + y + 1, y^3 + y, 0, y^3, 1, 0)
LOCALIZACIONES DE LOS ERRORES =  [2, 7]
VALOR DE LOS ERRORES =  [1, 1]
EL ESTIMADO DE LA PALABRA TRANSMITIDA ES C =  [1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1]
```

Observe que obtenemos la misma estimación de la palabra transmitida que en el Ejemplo 5.1, sin embargo, en tal ejemplo la técnica de decodificación es muy distinta a la que implementa los Algoritmos de Berlekamp-Massey y Forney.

Ejemplo 5.5 Ahora se transmite la palabra-código cero del código Reed-Solomon $\mathfrak{B}_{16}(15, 7, \alpha, 1)$ donde $\alpha \in \mathbb{F}_{16}$ es raíz del polinomio $x^4 + x + 1 \in \mathbb{F}_2[x]$. La palabra recibida es $[0, 0, \alpha^{11}, 0, 0, \alpha^5, 0, \alpha, 0, 0, 0, 0, 0, 0, 0]$.

```
DECODIFICACIÓN DE CODIGOS q-ARIOS BCH PRIMITIVOS EN EL SENTIDO ESTRICTO
INGRESE EL VALOR DE q: 16
INGRESE LA LONGITUD n DEL CODIGO: 15
INGRESE LA DISTANCIA DESIGNADA: 7
INGRESE LA PALABRA RECIBIDA (EN FORMA POLINOMIAL):
y*x^7 + (y^5)*x^5 + (y^11)*x^2
PALABRA RECIBIDA  [0, 0, y^3 + y^2 + y, 0, 0, y^2 + y, 0, y, 0, 0, 0, 0, 0, 0, 0]
VECTOR DE SINDROMES :  (y^3 + y^2 + y + 1, 1, y^3 + 1, y^3 + y^2 + 1, 1, y^3 + y^2 + y)
LOCALIZACIONES DE LOS ERRORES =  [2, 5, 7]
VALOR DE LOS ERRORES =  [y^3 + y^2 + y, y^2 + y, y]
EL ESTIMADO DE LA PALABRA TRANSMITIDA ES C =  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Como era de esperarse, el programa estimo correctamente a la palabra transmitida.

Ejemplo 5.6 Consideremos el código Reed-Solomon $\mathfrak{B}_8(7, 5, \alpha, 1)$ donde $\alpha \in \mathbb{F}_8$ es raíz del polinomio $x^3 + x + 1 \in \mathbb{F}_2[x]$.

```
DECODIFICACIÓN DE CODIGOS q-ARIOS BCH PRIMITIVOS EN EL SENTIDO ESTRICTO
INGRESE EL VALOR DE q: 8
INGRESE LA LONGITUD n DEL CODIGO: 7
INGRESE LA DISTANCIA DESIGNADA: 5
INGRESE LA PALABRA RECIBIDA (EN FORMA POLINOMIAL):
y^3 + y*x + x^2 + (y^2)*x^3 + (y^3)*x^5 + x^6
PALABRA RECIBIDA  [y + 1, y, 1, y^2, 0, y + 1, 1]
VECTOR DE SINDROMES :  (y + 1, y^2 + y, y^2 + y, 0)
LOCALIZACIONES DE LOS ERRORES =  [2, 3]
VALOR DE LOS ERRORES =  [y + 1, y^2 + 1]
EL ESTIMADO DE LA PALABRA TRANSMITIDA ES C =  [y + 1, y, y, 1, 0, y + 1, 1]
```

La variable "y" que aparece en la salida representa a α. Así que el estimado de la palabra-código es $c = [\alpha + 1, \alpha, \alpha, 1, 0, \alpha + 1, 1] = [\alpha^3, \alpha, \alpha, 1, 0, \alpha^3, 1]$. Este ejemplo es resuelto en [8]-pág 261, con técnicas extendidas de lo analizado en el Capítulo 3. Sin embargo, aún con técnicas distintas se llega al mismo estimado, lo cual es muy interesante por los conceptos que aborda cada técnica.

SageMath fue una herramienta bastante útil que permitió llevar a la programación ideas desarrolladas, en un principio, de forma teórica. Además, a través de estos programas los resultados analizados en todo este trabajo cobran aún más sentido. Los algoritmos programados fueron exitosos y eficientes, sin embargo, estos podrían ser modificados para extender las ideas a otro tipo de códigos BCH (que no sean en el sentido estricto), por lo cual el código se presento de manera explicita para que pueda ser de utilidad a personas que estudien este tipo de códigos y el problema de la decodificación.

CONCLUSIÓN

A lo largo del presente trabajo se comprobó que los códigos BCH y Reed-Solomon tienen propiedades bastante relevantes respecto a los parámetros importantes de un código. Esto hace ver la razón por la cual son una de las familias más importantes de los códigos lineales y más precisamente de los códigos cíclicos lineales. La construcción de estos códigos posibilita que la teoría respectiva sea profunda y lo suficientemente rica para seguir siendo estudiados con tanto interés. Los resultados presentados sirven de base para lograr un estudio más especializado de este tipo de códigos y sus posibles generalizaciones.

Las aplicaciones de la teoría matemática es uno de los principales intereses de una gran cantidad de profesionales, siendo más frecuente en lo relativo a cosas computacionales. Lo abordado en el Capítulo 5 hizo posible visualizar algunas aplicaciones mediante programación de las ideas teóricas manejadas en capítulos previos y motiva a seguir buscando posibles utilidades de los códigos estudiados en cosas más complejas y de una utilidad innegable en la vida cotidiana.

Bibliografía

[1] Berlekamp, E. (2015). *Algebraic Coding Theory* (Revised edition). Hackensack, NJ: World Scientific.

[2] Blahut, R. (2003). *Algebraic Codes for Data Transmission*. Cambridge, U.K : Cambridge University Press.

[3] Blahut, R. (2008). *Algebraic Codes on Lines, Planes, and Curves*. Cambridge, U.K : Cambridge University Press.

[4] Dummit, D., Foote, R. (2004). *Abstract Algebra* (3^a ed.). John Wiley and Sonns, Inc: Hoboken, USA.

[5] Hernández, H. (2018). *Códigos Cíclicos y de Residuos Cuadráticos sobre Anillos de Galois* (Tesis de Licenciatura). Benemérita Universidad Autónoma de Puebla, Puebla.

[6] Ling, S., Xing, C. (2004). *Coding Theory A First Course*. Cambridge, U.K: Cambridge University Press.

[7] MacWilliams, F., Sloane, N. (1978). *The Theory of Error Correcting Codes*. New York: North Holland Publishing.

[8] McEliece, R. (2004). *The Theory of Information and Coding*. Cambridge, U.K: Cambridge University Press.

[9] Niederreiter, H., Lidl, R. (1997). *Finite Fields*. New York: Cambridge.

[10] Niederreiter, H., Wintherhof, A. (2015). *Applied Number Theory*. New York: Springer.

[11] Roman, S. (1992). *Coding and Information Theory*. New York: Springer-Verlag.

[12] Roman, S. (2005). *Field Theory* (2^a ed.). New York: Springer.

[13] Rotman, J. (1990). *Galois Theory* (2^a ed.). New York: Springer.

[14] Stein et al. (2005). *SageMath* (Versión 8.6). Windows. `http://www.sagemath.org`

[15] Zimmermann et al.(2018) *Computational Mathematics with SageMath*. doi: `10.1137/1.9781611975468`