1er Coloquio del Departamento de Matemáticas

Introducción a Campos Finitos con Aplicación a Códigos Lineales

Horacio Tapia-Recillas

Introducción a Campos Finitos con Aplicación a Códigos Lineales

Horacio Tapia-Recillas

Departamento de Matemáticas, UAM-I

Casa abierta al tiempo

Universidad Autónoma Metropolitana

Contenido

Introducción

El estudio de los campos finitos tiene una larga historia y data de los siglos XVII y XVIII, con P. de Fermat (1601–1665), L. Euler (1707–1783), J.A. Lagrange (1736–1813) y A.M. Legendre (1752–1833), quienes enfocaron su estudio principalmente a los campos de la forma $\mathbb{Z}_p$, es decir, enteros modulares donde p es un número primo. La teoría de campos finitos como se conoce actualmente comenzó a fines del siglo XVIII y durante el siglo XIX, los principales impulsores fueron C. F. Gauss (1777–1855), E. Galois (1811–1832), R. Dedekind (1857) y E. H. Moore (1893), entre otros. El trabajo que realizó E. Galois referente al estudio de estos objetos matemáticos marca el comienzo de los campos finitos, conocidos también como campos de Galois, los cuales han jugado un papel importante en varias áreas de las matemáticas.

En las últimas décadas el estudio de los campos finitos se ha visto fuertemente impulsado, gracias en buena medida al desarrollo de la computación y las comunicaciones digitales, en especial por su aplicación a la detección y corrección de errores en la transmisión de información, es decir, en el estudio de la Teoría de Códigos, así como en el manejo de la seguridad de la información, particularmente en relación a los cifrados, es decir, a la Criptografía. Sin embargo, estas no son las únicas áreas relacionadas con los campos finitos, entre otras se encuentran las siguientes: combinatoria, geometrías proyectiva y finitas, así como el estudio de la transformada discreta de Fourier, teoría espectral, procesamiento digital de señales. El propósito de las presentes notas es dar al lector una introducción al estudio de los campos finitos. De particular importancia es la construcción de estos objetos matemáticos y sus propiedades. Además, como aplicación de los eneros modulares se darán los elemenos del sistema de cifrado conocido como RSA (por las iniciales de los apellidos de las personas que lo diseñaron: R. Rivest, A. Shamir y L. Adleman) y en el caso de los campos finitos se dará una introducción a la Teoría de Códigos Lineales Detectores-Correctores de Error, particularmente de una clase importante de estos: los códigos cíclicos.

Estas notas estan organizadas de la siguiente manera: en el capítulo 2 se recordarán algunas nociones y resultados básicos de Teoría de Números, particularmente sobre enteros modulares los cuales proporcionan ejemplos básicos de campos finitos, y quizá los primeros campos de esta naturaleza conocidos por P. de Fermat, L. Euler y E. Galois, como ya se mencionó antes. En el capítulo 3, basado en los enteros modulares y algunas de sus propiedades, se dán los elementos del sistema de cifrado RSA. En el capítulo 4 se presenta un método para la construcción de campos finitos. Este método esta basado en una de las formas algebraicas de construir los números complejos a partir del polinomio (irreducible) $x^2 + 1$ y en el capítulo 5 se presentan algunos resultados generales de los campos finitos. Existe una gran variedad de aplicaciones de estos campos, como ya se mencionó antes, de las cuales en el capítulo 7 se discute una relacionada con códigos detectortes-correctores de errores usados en la transmisión de información por canales convencionales. Además, a manera de ilustración, se mencionan ejemplos de algunos códigos que se usan en la vida cotidiana. De particular importancia por sus aplicaciones (por ejemplo en los aparatos reproductores de CD's) son los códigos cíclicos en los cuales se pone de manifisesto la estructura del anillo de polinomios en una indetermianda con coeficientes en un campo finito y el anillo cociente del anillo de polinomios. Esto se desarrollará en el capítulo 8.

El lector interesado en temas relacionados con los campos finitos y aplicaciones puede consultar las diversas referencias que se presentan al final de estas notas, las cuales se han agrupado en aquellas que tratan sobre diversos aspectos de los campos finitos, teoría de códigos, criptografía y teoría de números, entre otras. Se han agregado direcciones en internet donde se pueden consultar temas relacionados con algunas áreas afines a los campos finitos, así como algunas revistas relevantes al tema.

Deseo agradecer la colaboración de Araceli Sánchez Balbuena quién, entre otras cosas, realizó los ejemplos con el paquete computacional GAP (Groups, Algorithms and Programming) y las lecturas que hizo de texto de estas notas. También deseo agradecer al Departamento de Matemáticas de la Universidad Autónoma Metropolitana, Unidad Iztapalapa por la invitación a presentar este material en este Coloquio.

Capítulo 1

Un anillo importante

El propósito de este capítulo es el de recordar algunos conceptos de Teoría de Números, particularmente los enteros modulares, los cuales proporcionan ejemplos importantes de campos finitos. A partir de estos campos se construiran otros campos finitos como extensiones algebraicas. Además de proporcionar ejemplos de campos finitos, los enteros modulares son la base para el sistema de cifrado de llave pública conocido como RSA (por las iniciales de los apellidos de las personas que lo diseñaron: R. Rivest, A. Shamir y L. Adleman). Este sistema de cifrado se describirá en el siguiente capítulo para lo cual se recordarán algunos conceptos relacionados con el anillo de enteros modulares. Comenzaremos por recordar algunos conceptos básicos como son el de *grupo*, *anillo* y *campo*.

1.0.1. Conceptos básicos.

DEFINICIÓN 1.0.1. Un *grupo* es una pareja $(G, \circ)$ donde G es un conjunto no vacio y "$\circ$" es una operación binaria en G tal que:

(1) G es *cerrado* bajo la operación " $\circ$ ", es decir, si $a, b \in G$ entonces $a \circ b \in G$

(2) La operación "$\circ$" es *asociativa*: si $a, b, c \in G$ entonces

$$(a \circ b) \circ c = a \circ (b \circ c)$$

(3) Existe un elemento (único) $e \in G$ tal que

$$e \circ g = g \circ e = g$$

para todo $g \in G$. Al elemento e se le llama la **identidad** de G.

(4) Para cada elemento $a \in G$ existe un elemento (único) $x \in G$ tal que

$$a \circ x = x \circ a = e$$

Al elemento x con esta propiedad se le llama el **inverso** de a.

Obsérvese que la condición 4) es equivalente a decir que la ecuación $ax = e$ tiene solución (única) en el conunto G.

Si además la operación satisface:

$$a \circ b = b \circ a, \text{ para todo elemento } a, b \in G,$$

se dice que el grupo es *conmutativo*. El grupo $(G, \circ)$ es finito si la cardinalidad del conjunto G es finita, en otro caso el grupo es infinito.

Estamos seguros que el lector conoce los siguientes ejemplos de grupos: $(\mathbb{Z}, +), (\mathbb{R}, +), (\mathbb{Q}, +), (\mathbb{C}, +)$; los números enteros, reales, racionales y complejos, respectivamente, donde la operación es la suma de los elementos en cada caso. También $(\mathbb{R}^*, *), (\mathbb{Q}^*, *), (\mathbb{C}^*, *)$; donde $K^* = K \setminus \{0\}$, son grupos donde la operación es la multiplicación de los elementos en cada caso. Obviamente estos son grupos conmutativos de orden infinito.

A continuación se darán dos ejemplos de grupos finitos, uno conmutativo y el otro no-conmutativo.

Ejemplo 1. (raíces cuartas de la unidad). Sea $G_4 = \{1, i, -1, -i\}$ donde i es el número complejo tal que $i^2 = -1$. Dado que G_4 es un subconjunto de los números complejos, la operación en G_4 se toma como la multiplicación en $\mathbb{C}$. El hecho que $(G_4, *)$ cumple todas las propiedades para ser un grupo (finito, conmutativo) se puede ver de la siguiente tabla:

$*$	1	i	-1	$-i$
1	1	i	-1	$-i$
i	i	-1	$-i$	1
-1	-1	$-i$	1	i
$-i$	$-i$	1	i	-1

Observaciones.

(1) $G_4 = \{i^n : n \in \mathbb{Z}\} = \{1, i, i^2 = -1, i^3 = -i\}$, es decir, G_4 es un grupo cíclico generado por el elemento i. Otro generador de este grupo es $-i$.
(2) Los elementos de G_4 satisfacen la relación: $X^4 = 1$, es decir, son las raices (complejas) cuártas de la unidad.

Ejercicios.

(1) Probar que las raíces quíntas de la unidad, con el producto de números complejos, es un grupo cíclico. Determinar todos los generadores de este grupo.
(2) En general, si $n > 1$ es un entero, probar que las raíces n-ésimas de la unidad son un grupo cíclico de orden n y determinar todos sus generadores.

EJEMPLO 1. 2. (Grupo simétrico). Sea A un conjunto con n elementos y sea

$$S_n = \{\sigma : A \longrightarrow A,\ \sigma \text{ es funcion biyectiva}\}$$

Se define la operación en S_n como la composición de funciones "$\circ$". Entonces la pareja $(S_n, \circ)$ es un grupo finito de orden $n!$. Si $n \geq 4$, este gupo no es conmutativo. Este grupo es uno de los mas importantes en el estudio de grupos finitos. A continuación, a manera de ilustración, se describe la tabla de este grupo para el caso de $n = 3$.

Ahora recordemos la definición de un anillo.

DEFINICIÓN 1.0.2. Un *anillo* es un conjunto A con dos operaciones, " + " y " $*$ " tales que:

(1) $(A, +)$ es un grupo *conmutativo*.
(2) "$*$" es *asociativa*, es decir, para todo $a, b, c \in A$:

$$a * (b * c) = (a * b) * c$$

(3) La ley *distributiva* se satisface, es decir, para todo $a, b, c \in A$:

$$a * (b + c) = a * b + a * c, \quad (b + c) * a = b * a + c * a$$

Un anillo $(A, +, *)$ es:

- *conmutativo*, si la operación "$*$" es conmutativa.
- con *identidad* si A tiene una identidad multiplicativa.
- *de división* si $(R^* = R - \{0\}, *)$ es un grupo.

Como ejemplos de anillos se tienen: $(\mathbb{Z}, +, *)$, $(\mathbb{R}, +, *)$, $(\mathbb{Q}, +, *)$, $(\mathbb{C}, +, *)$; los números enteros, reales, racionales y complejos, donde las operaciones son la suma y producto usuales de los elementos en cada caso.

A continuación se darán otros ejemplos de anillos.

Ejemplo 1. (Anillo de polinomios.) Sea K alguno de los siguientes anillos: $\mathbb{R}, \mathbb{Q}, \mathbb{C}$ y sea

$$K[x] = \{f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,\ a_i \in K, n \geq 0, \text{ entero}\}$$

es decir, $K[x]$ consiste de todas las expresiones algebraicas en la indeterminada "x" con coeficientes en K. A estas expresiones se les llama *polinomios*. El entero n es el *grado* del polinomio y los elementos $a_n, a_{n-1}, ..., a_0$ son los coeficientes del polinomio.

Con las operaciones de suma y producto de polinomios definidos en la forma natural (como nos enseñaron en cursos básicos de algebra), se tiene que $(K[x], +, *)$ es un anillo conmutativo (infinito).

observación. Se puede definir el conjunto de expresiones algebraicas en varias indeterminadas con coeficientes en K, es decir, polinomios en varias variables con coeficientes en K, asi como una suma y producto con las cuales este conjunto es también un anillo conmutativo.

Ejemplo 2. (Anillo de matrices.) Sea $n \geq 1$ un entero, K como en el ejemplo anterior y denotemos por $\mathcal{M}_n(K)$ al conjunto de matrices $n \times n$ con entradas en K. Con las operaciones usuales de suma ($+$) y producto ($*$) de matrices se tiene que $(\mathcal{M}_n(K), +, *)$ es un anillo no-conmutativo (si $n > 1$).

En la siguiente sección se proporcionarán otros ejemplos de anillos (conmutativos), los cuales jugarán un papel muy importante, particularmente en el siguiente capítulo.

Ejercicio. Dar otros ejemplos de anillos.

Recordemos ahora la definición de otra estructura algebraica que será de muy importante a lo largo de estas notas.

DEFINICIÓN 1.0.3. Un campo es una terna $(\mathbb{F}, +, *)$ donde $(\mathbb{F}, +)$ es un grupo conmutativo y si $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$, $(\mathbb{F}^*, *)$ es tambien un grupo conmutativo. En particular todo elemento distinto de cero de $\mathbb{F}$ tiene inverso multiplicativo.

Como ejemplo de campos se tiene: $(K, +, *)$ donde $K = \mathbb{R}, \mathbb{Q}, \mathbb{C}$. El conjunto

$$\mathbb{Q}(i) = \{a + bi : a, b \in \mathbb{Q}\}$$

con las operaciones inducidas de los números complejos es tambien un campo (un subcampo del campo de los números complejos). Obviamente estos campos no son finitos. En el capítulo 4 se presentarán otros ejemplos de campos finitos.

Ejercicio. Dar otros ejemplos de campos.

Otro concepto que será de gran utilidad es el siguiente:

DEFINICIÓN 1.0.4. Un *espacio lineal* o vectorial sobre un campo K es una terna $(V, +, \cdot)$, donde $(V, +)$ es un grupo conmutativo y "$\cdot$" : Kx$V \longrightarrow V$ es una operación (multiplicación por escalares) tales que:

(1) $(\alpha + \beta) \cdot v = \alpha \cdot v + \beta \cdot v$, para todo escalar $\alpha, \beta \in K$, y todo elemento $v \in V$.
(2) $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$, para todo escalar $\alpha \in K$ y todo elemento $u, v \in V$.

Aunado al concepto de espacio vectorial esta el de *base* y *dimensión*. Como ejemplo de espacio vectorial de dimensión finita n se tiene el producto cartesiano $\mathbb{R}^n$ el cual es un $\mathbb{R}$-espacio vectorial de dimensión

n, y una base (canónica) esta dada por los vectores $e_i = (0, 0, ..., 1, ..., 0)$ donde la coordenada 1 aparece en el lugar i, para $i = 1, 2, ..., n$. Un ejemplo de espacio vectorial sobre el campo de los números complejos es $\mathbb{C}^n$, el cual es de dimensión n sobre $\mathbb{C}$ pero de dimensión $2n$ sobre $\mathbb{R}$. En las siguientes páginas se introducirá el campo de los números *binarios*, $\mathbb{F}_2$, y en forma análoga a los ejemplos anteriores se tiene el espacio vectorial $\mathbb{F}_2^n$ sobre los binarios el cual es tambien de dimensión n sobre este campo pero además es finito con 2^n elementos. Este espacio vectorial juega un papel muy importante en computación y en el estudio de diversos aspectos de la información (transmisión y seguridad) como se verá mas adelante.

1.1. Un anillo interesante

En esta sección se describirá un ejemplo de anillo (finito) que es muy importante desde varios puntos de vista y es usado en diversas áreas de la matemática, computación e ingeniería, entre otras, el cual se usará en estas notas, particularmente en los capítulos 3 y 4, nos referimos al anillo de *enteros modulares*.

1.1.1. Relación de equivalencia. Motivemos el concepto de relación de equivalencia con el siguiente ejemplo.

Sea M el conjunto de todos los ciudadanos mexicanos con acta de nacimiento emitida en alguna de las 32 entidades federativas.

En este conjunto se define la siguiente relación entre sus elementos:

dos ciudadanos estan relacionados siempre y cuando su acta de nacimiento sea de la misma entidad federativa.

Observemos lo siguiente:

- Cualquier ciudadano esta relacionado consigo mismo:

$$A \sim A$$

- Si el ciudadano A esta relacionado con el ciudadano B, entonces el ciudadano B esta relacionado con el ciudadano A:

$$\text{si } A \sim B, \text{ entonces } B \sim A$$

- Si el ciudadano A esta relacionado con el ciudadano B, y B esta relacionado con el ciudadano C, entonces el ciudadano A esta relacionado con el ciudadano C:

$$\text{si } A \sim B \text{ y } B \sim C, \text{ entonces } A \sim C$$

Ejercicio. Dar ejemplos de cojuntos y una relación entre sus elementos tal que satisfaga las propiedades anteriores.

Este ejemplo sugiere la siguiente:

DEFINICIÓN 1.1.1. Una relación " $\sim$" entre los elementos de un conjunto S es de *equivalencia* si:

- es **reflexiva**, es decir,

$$a \sim a$$

- es **simétrica**, es decir,

$$\text{si } a \sim b, \text{ entonces } b \sim a$$

- es **transitiva**, es decir,

$$\text{si } a \sim b \text{ y } b \sim c, \text{ entonces } a \sim c$$

para todo elemento $a, b, c \in S$.

Observación. El ejemplo anterior de la relación entre los ciudadanos, es de equivalencia.

A los subconjuntos S_i se les llama *clases de equivalencia* y a cada elemento en una clase de equivalencia se le llama un *representante* de la clase. En el ejemplo de los ciudadanos, las clases de equivalencia son 32 y corresponden a las entidades federativas de México. Por ejemplo, un representante de la clase de Michoacán es cualquier ciudadano con acta de nacimiento de esa entidad federativa.

Ejercicio. Dar dos ejemplos de relación de equivalencias y dos ejemplos en los cuales la relación definida no sea de equivalencia.

Observación. Toda relación de equivalencia en un conjunto S induce una *partición* del conjunto, es decir, existe una colección $\{S_i,\ i \in I\}$ de subconjuntos de S tal que:

- $S = \bigcup_{i \in I} S_i$
- $S_i \cap S_j = \emptyset$, $i \neq j$

1.1.2. Algunos resultados del anillo de enteros. En esta sección se recordarán algunos conceptos importantes del anillo de los enteros (racionales) $\mathbb{Z}$.

1.1.2.1. *Algoritmo de la División.* Este algoritmo dice lo siguiente:

Dados n, m enteros, existen enteros q, r tales que:

$$m = nq + r, \quad 0 \leq r \leq n - 1$$

1.1.2.2. *Máximo común Divisor (MDC). Dados los enteros n, m existe otro entero d tal que:*

- *es un divisor común de n y m.*
- *es el máximo de los divisores comunes de estos enteros.*

La siguiente es una notación común para denotar al máximo común divisor de los enteros m y n: $d = mcd(n, m)$ o simplemente $d = (m, n)$.

1.1.2.3. *Algoritmo de Euclides.* Un resultado importante en el estudio del anillo de los enteros $\mathbb{Z}$ es el siguiente:

*El MCD de dos enteros n, m se puede expresar, de manera única, como combinación lineal de estos enteros (***algoritmo extendido de Euclides***):*

$$d = an + bm$$

Los resultados anteriores se han enunciado para el anillo de los números enteros $\mathbb{Z}$, pero si en el lugar adecuado se cambia un entero por un polinomio, se tienen los mismos resultados.

1.1.2.4. *Otras propiedades.* Como consecuencia de algunas de las propiedades antes mencionadas para el anillo de los enteros y de polinomios en una indetermiadad con coeficientes en un campo, se tienen los siguientes resultados:

TEOREMA 1.1.2. *El anillo de los enteros $\mathbb{Z}$ (polinomios $K[x]$, K campo) es de factorización única, es decir, todo elemento se puede expresar de forma única como producto de elementos irreducibles.*

TEOREMA 1.1.3. *El anillo de los enteros $\mathbb{Z}$ (polinomios $K[x]$, K campo) es de ideales principales, es decir, todo ideal esta generado por un elemento.*

Ejercicio. Dar una demostración de los resultados anteriores.

Entre otras cosas los enteros $\mathbb{Z}$ y los polinomios $K[x]$ (K campo) son ejemplos de los llamados *anillos euclideanos*.

1.1.3. Enteros modulares. Ahora se introducirá una relación de equivalencia en el conjunto de los números enteros $\mathbb{Z}$, la cual va a ser muy importante a lo largo de estas notas, particularmente en el siguiente capítulo.

Con las propiedades de los números enteros mencionadas anteriormente, consideremos el siguiente ejemplo:

Sea n un entero (positivo), digamos $n = 4$ y defínase la siguiente relación entre los elementos de $\mathbb{Z}$ (enteros):

$$\text{para } a, b \in \mathbb{Z},\ a \sim b \text{ si } a - b = kn$$

es decir, $a \sim b$, si su diferencia es un múltiplo de n, en este caso $a - b = 4k$.

Por ejemplo, 19 y 5 no son equivalentes ya que 19-5=14, el cual no es múltiplo de 4. Sin embargo 19 y 3 son equivalentes: $19-3 = 16 = 4(4)$ y $3 - 19 = -16 = 4(-4)$, es decir, 3 y 19 son equivalentes también.

Tenemos ahora la siguiente afirmación: *la relación definida anteriormente es de equivalencia.*

Ejercicio. Comprobar que efectivamente esta relación es de equivalencia.

Como toda relación de equivalencia induce una partición, en este caso se tienen 4 clases de equivalencia:

$$\overline{0},\ \overline{1},\ \overline{2},\ \overline{3}$$

Así por ejemplo, la clase $\overline{2}$ consiste de todos los enteros m tales que $m - 2 = 4k$, es decir, la diferencia $m - 2$ es un múltiplo de 4. Por ejemplo, 2, 6, -2, -6 estan en la clase de equivalencia $\overline{2}$, es decir, son representantes de esa clase de equivalencia.

Ejercicio. Dar algunos representantes de las otras clases de equivalencia en este ejemplo.

El ejemplo anterior motiva introducir el siguiente concepto.

Sea n un entero (positivo) y defínase la siguiente relación en los números enteros $\mathbb{Z}$:

$$a \sim b \iff a - b = nk$$

es decir, los enteros a y b estan relacionado si su diferencia $a - b$ es un múltiplo del entero n.

Así como en el ejemplo anterior se puede ver fácilmente que esta relación es de **equivalencia**.

Ejercicio. Mostrar que esta relación es de equivalencia.

Notación. Esta relación se acostumbra expresarla como:

$$a \equiv b \ mod\ n \iff a - b = nk$$

y se lee de la siguiente manera:

a es congruente con b módulo n sí y sólo si su diferencia es un múltiplo de n

Las clases de equivalencia inducidas por esta relación (de equivalencia) se denotarán por:

$$\overline{0},\ \overline{1}, \ldots, \overline{n-1}$$

Al conjunto cuyos elementos son las clases de equivalencia módulo n se acostumbra denotarlo por:

$$\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z} = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}$$

y se le llama el **conjunto de enteros módulo** n o simplemente **enteros modulares** (módulo n).

Obsérvese que con la ayuda de la relación de congruencia módulo un entero n, definida en el conjunto de los números enteros, se obtiene un nuevo conjunto, los enteros modulares, cuya cardinalidad es n.

Por ejemplo, si $n = 6$:

$$\mathbb{Z}_6 = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}, \overline{5}\}$$

Ejercicios.

(1) Determinar en qué clase de equivalencia del ejemplo anterior estan los siguientes enteros: -1, -211, -48, $2^8 - 1$, 1111, 2^{20}.
(2) Dar un representante de las clases de equivalencia de $\mathbb{Z}_6$ que tengan al menos 5 dígitos, algunos positivos y otros negativos.
(3) ¿Cuál es la clase de equivalencia del entero $2^{30} - 1$ en $\mathbb{Z}_{10}$?
(4) Determinar el conjunto $\mathbb{Z}_{11}$ y dos representantes de cada una de sus clases de equivalencia.

Con la ayuda del algoritmo de la división, veamos como se obtiene de forma natural un representante de cada clase de equivalencia de los enteros modulares.

Sea n un entero (positivo) y sea a cualqier otro entero. Por el algoritmo de la división se tiene que:

$$a = r + kn, \text{ con } 0 \leq r < n$$

es decir, $a - r = kn$, equivalentemente, $a \equiv r \bmod n$

En otras palabras, el entero a es congruente módulo n al residuo que se obtiene de dividir a a por el entero n. Por consiguiente, la clase de equivalencia del entero a es la misma que la de su residuo: $\overline{a} = \overline{r}$. Por la propiedad que tiene el residuo, se sigue que sólo hay n posibilidades para las clases de equivalencia: $\overline{0}, \overline{1}, ..., \overline{n-1}$. Por consiguente:

$$\mathbb{Z}/n\mathbb{Z} = \{\overline{0}, \overline{1}, ..., \overline{n-1}\}$$

y como consecuencia se tiene que la cardinalidad de los enteros módulo n es: $|\mathbb{Z}/n\mathbb{Z}| = n$.

Como el conjunto de los enteros modulares $\mathbb{Z}_n$ se construyó a partir del anillo de los enteros con la ayuda de una relación de equivalencia (congruencia módulo n), una pregunta natural es si este conjunto de enteros modulares es tambien un anillo. Para tal propósito es necesario

definir la suma y producto y ver que estas operaciones satisfacen las propiedades para ser un anillo.

Veamos como se define la suma de los elementos $\overline{a}$ y $\overline{b}$ de $\mathbb{Z}/n\mathbb{Z}$, es decir, la suma de esas clases de equivalencia. Sea a un representante de la clase de equivalencia $\overline{a}$ y b un representante de la clase $\overline{b}$.

Como esos representantes son enteros, $a + b$ es otro entero. Aplicando el algoritmo de la división se tiene que $a + b = c + rn$, con $0 \leq c < n$ y por lo tanto los enteros $a + b$ y c determinan el mismo elemento de $\mathbb{Z}/n\mathbb{Z}$ que. De manera análoga, $\overline{a} * \overline{b}$ es el elemento de $\mathbb{Z}_n$ que se obtiene de tomar el producto de los enteros a y b y obtener el residuo al dividirlo por n.

Por facilidad de notación en lo sucesivo y esperando no haya confusión, se omitirá la barra para denotar los elementos de los enteros modulares. Las siguientes tablas muestran la suma y producto sobre los enteros módulo $n = 6$, $\mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}$:

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

*	1	2	3	4	5
1	1	2	3	4	5
2	2	4	0	2	4
3	3	0	3	0	3
4	4	2	0	1	2
5	5	4	1	2	1

Ejercicio. Probar que la terna $(\mathbb{Z}_n, +, *)$, donde " $+$ " y " $*$ " son las operaciones definidas anteriormente, es un anillo conmutativo, finito, con identidad.

Obsérvese que en el caso de $(\mathbb{Z}_6, +, *)$ no todos los elementos tienen inverso bajo la multiplicación. Veamos por ejemplo que en el caso $(\mathbb{Z}_7, +, *)$, todos los elementos distintos de cero tienen inverso multiplicativo. Para esto basta dar la tabla de multiplicar correspondiente:

*	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

En este caso la terna $(\mathbb{Z}_7, +, *)$ es un anillo (conmutativo) donde todo elemento distinto de cero tiene un inverso multiplicativo, es decir,

$(\mathbb{Z}_7, +, *)$ es un *campo finito* ya que su cardinalidad es 7. Obsérvese también que el conjunto $\mathbb{Z}_7^* = \mathbb{Z}_7 \setminus \{0\} = \{1, 2, 3, 4, 5, 6\}$ es un grupo cíclico (multiplicativo), es decir, esta generado por alguno de sus elementos: $\mathbb{Z}_7^* =< 2 >$.

Uno de los ejemplos mas sencillos e importantes de los enteros modulares es el campo de los *números binarios*: $(\mathbb{Z}_2, +, *)$. Otra notación para denotar a este campo es: $\mathbb{F}_2$ o $GF(2)$.

Normalmente un ingeniero en Electrónica considera a este conjunto pensando que "pasa" o "no pasa" corriente; una persona en Ciencias de la Computación lo piensa como "bits", los cuales se "encienden" o se "apagan". Las"tablas" de sumar y multiplicar de los números binarios son las siguientes:

$+$	0	1
0	0	1
1	1	0

$*$	0	1
0	0	0
1	0	1

Con la ayuda de los enteros modulares se puede dar bastantes ejemplos de campos finitos, de acuerdo al siguiente resultado:

TEOREMA 1.1.4. *Sea n un entero positivo. El anillo $(\mathbb{Z}/n\mathbb{Z}, +, *)$ es un campo sí y sólo si n es un número primo. Si n es primo $\mathbb{Z}_n^* = \mathbb{Z}_n \setminus \{0\}$, su grupo multiplicativo, es cíclico de orden $n - 1$.*

Ejercicio. Dar una demostración de este resultado.

El resultado anterior provee de un gran número de ejemplos de campos finitos: tantos como números primos existen. Pero hay muchas preguntas por hacer, por ejemplo, si el módulo n es suficientemente grande, digamos del orden de 50, 100 o 150 dígitos, ¿cómo se podrá hacer la aritmética en el anillo correspondiente? En particular, ¿cómo determinar si un elemento de $\mathbb{Z}_n$ tiene inverso multiplicativo y en caso afirmativo como se obtiene este? , o bien, ¿qué otro tipo de campos finitos existen, cómo se realiza su aritmética y qué propiedades tienen?

Una pregunta natural que se puede hacer el lector es la siguiente: ¿donde se usan los anillos de la forma $\mathbb{Z}/n\mathbb{Z}$ donde n es un entero del orden de 100 o 150 dígitos? o bien ¿donde se usan otros campos finitos? Una respuesta a la primera pregunta es que el anillo $\mathbb{Z}/n\mathbb{Z}$ con n grande, se usa en el sistema de cifrado (o encriptado) conocido como RSA, donde además varios de los resultados básicos de Teoría de Números como el Teorema (pequeño) de Fermat, la función de Euler y el Teorema de Euler, entre otros, son de suma importancia en este sistema criptográfico. Una respuesta a la segunda pregunta es que los campos finitos se usan, por ejemplo en la detección y correción de errores que

se adquieren en la transmisión de información por cualquier medio convencional (teléfono, satélite, internet, etc.), en el contexto de la Teoría de códigos lineales. Además de que tambien se usan en otros tipos de cifrado como el AES (Advance Encryption Standard, de llave privada) o bien en los de llave pública como es el basado en curvas elípticas o hiperelípticas definidas sobre campos finitos. En capítulos posteriores se dará una introducción a teoría de códigos lineales.

En el capítulo 3 se describirá el sistema de cifado RSA, el capítulo 4 se darán algunos ejemplos de campos finitos, se presentará una construcción general de estos objetos y se verán algunas de sus propiedades.

1.1.3.1. *Unidades de* $\mathbb{Z}_n$. En esta sección se recordarán varios conceptos importantes en el estudio del anillo $\mathbb{Z}_n$ de enteros modulares, como su grupo de unidades, Teorema (pequeño) de Fermat, función de Euler,Teorema de Euler y algunas de sus propiedades. Normalmente estos conceptos se estudian en un curso básico de Teoría de Números.

Comencemos con algunos ejemplos y observando que hay unas diferencias entre $\mathbb{Z}_6$ y $\mathbb{Z}_7$, las cuales se pueden ver fácilmente de sus tablas de multiplicar:

(1) En el caso $\mathbb{Z}_6$ se tiene que: (2)(3)=0 y en el caso $\mathbb{Z}_7$, $(a)(b) = 0$ sólo si a o b es igual a 0.
(2) La ecuación $aX = 1$ ($a \neq 0$) no siempre tiene solución en $\mathbb{Z}_6$ (por ejemplo si $a = 2, 3, 4$), en cambio esta ecuación siempre tiene solución en $\mathbb{Z}_7$ ($a \neq 0$), es decir, no todo elemento (distinto de cero) en $\mathbb{Z}_6$ tiene inverso bajo el producto, sin encambio en $\mathbb{Z}_7$, todo elemento distinto de cero tiene inverso bajo el producto.

Las observaciones anteriores permiten dar la siguiente:

DEFINICIÓN 1.1.5. Se dice que un elemento a de $\mathbb{Z}_n$ es **unidad** si tiene inverso multiplicativo.

El conjunto de unidades de $\mathbb{Z}_n$ se denotará por $(\mathbb{Z}_n)^*$ o bien por $U(\mathbb{Z}_n)$.

Ejemplos.

(1) El conjunto de unidades de $\mathbb{Z}_6$ es: $(\mathbb{Z}_6)^* = \{1, 5\}$.
(2) El conjunto de unidades de $\mathbb{Z}_7$ es:

$$(\mathbb{Z}_7)^* = \{1, 2, 3, 4, 5, 6\} = \mathbb{Z}_7 - \{0\}.$$

A continuación se dará un criterio para determinar cuando un elemento de $\mathbb{Z}_n$ es unidad.

PROPOSICIÓN 1.1.6. *Un elemento $\overline{a} \in \mathbb{Z}_n$ es unidad sí y sólo si cualquier representante de la clase $\overline{a}$ es primo relativo con n.*

DEMOSTRACIÓN. Veamos primero que si a es un representante de la clase $\overline{a}$ y es primo relativo con n, entonces esa clase es una unidad en $\mathbb{Z}_n$. Como se mencionó anteriormente, el MCD de dos elementos es combinación lineal de dichos elementos. En nuestro caso, dado que $(a, n) = 1$ existen enteros s, k tales que:

$$1 = as + nk$$

Tomando clases módulo n se tiene que: $\overline{as} = \overline{1}$, es decir $\overline{a}$ es invertible.

El recíproco se deja al lector como ejercicio. □

De acuerdo a este resultado, para determinar si un elemento de $\mathbb{Z}_n$ es invertible, basta ver que cualquiera de sus representantes es primo relativo con n, pero ¿como se logra esto? Aquí es donde el algoritmo de Euclides es de gran ayuda Supóngase ahora que ya se conoce que cualquier representante de una clase de $\mathbb{Z}_n$ es primo relativo con n, la pregunta ahora es: ¿como se obtiene su inverso? La respuesta esta dada por el Algoritmo Extendido de Euclides, por medio del cual se puede dar la combinación lineal de la forma $1 = as + nk$ de donde se obtiene fácilmente el inverso de la clase $\overline{a}$: $\overline{a}^{-1} = \overline{s}$.

Ejercicio. Determinar el conjunto de unidades de $\mathbb{Z}_n$ para $n = 10, 13, 16, 50$.

En base a estos ejemplos (y ejercicios) se tienen las siguientes preguntas:

(1) Si n es un entero grande, o en general, ¿como saber si un elemento de $\mathbb{Z}_n$ es unidad?, en caso afirmativo ¿cuál es el inverso de este elemento?
(2) ¿Cuántas unidades hay en $\mathbb{Z}_n$?

1.1.3.2. *Función de Euler.* Una forma de definir la función de Euler es la siguiente:

DEFINICIÓN 1.1.7. La función de Euler $\varphi : \mathbb{N} \longrightarrow \mathbb{N}$ esta definida como $\varphi(n) = |U(\mathbb{Z}_n)|$.

De la proposición anterior se sigue que:

$$\varphi(n) = |\{x : \ 1 \leq x < n, \ (x, n) = 1\}|$$

Por ejemplo si p es un número primo, entonces $\varphi(p) = U(\mathbb{Z}_p) = p-1$, es decir, todos los elementos distintos de cero de $\mathbb{Z}$ son invertibles, equivalentemente, $\mathbb{Z}_p$ es un campo (finito con p elementos).

Ejercicio. Determinar el número de unidades del anillo $\mathbb{Z}_n$ para $n = 8, 10, 15, 91$.

La función de Euler tiene varias propiedades interesantes, entre las cuales se pueden mencionar las siguientes:

TEOREMA 1.1.8. (1) *Si p es un primo y r es un entero positivo, entonces*

$$\phi(p^r) = p^r - p^{r-1} = p^r(1 - \frac{1}{p}).$$

(2) *Si n y m son enteros (positivos) primos relativos entre si, entonces*

$$\varphi(nm) = \varphi(n)\varphi(m)$$

(3) *Si la descomposición del entero n como producto de primos (distintos) es $p_1^{r_1} p_2^{r_2} \cdots p_t^{r_t}$, entonces*

$$\varphi(n) = \phi(p_1^{r_1})\phi(p_2^{r_2}) \cdots \phi(p_t^{r_t})$$

DEMOSTRACIÓN. La primera afirmación es fácil de probar y se deja como ejercicio al lector. La última es consecuencia de la segunda, y la demostración de esta también se deja como ejercicio al lector. □

Observación. La afirmación 2) del teorema anterior es equivalente a:

PROPOSICIÓN 1.1.9. *Si $(n, m) = 1$ entonces el grupo $\mathbb{Z}_{nm}$ es isomorfo al grupo $\mathbb{Z}_n \times \mathbb{Z}_m$.*

Observación. Un caso particular de la afirmación 2) del teorema anterior y la cual se usará en el siguiente capítulo es la siguiente:

Si p y q son primos distintos, entonces

$$\varphi(pq) = (p - 1)(q - 1)$$

1.1.4. Dos resultados importantes de los enteros modulares. En esta sección se recordarán dos resultados importantes de Teoría de Números: el Teorema (pequeño) de Fermat y el Teorema de Euler.

1.1.4.1. *El Pequeño Teorema de Fermat.* Este resultado de Fermat dice lo siguiente:

TEOREMA 1.1.10. *Sea p un número primo. Entonces para cualquier entero (positivo) a:*

$$a^{p-1} \equiv 1 \bmod p$$

Este resultado se puede enunciar también de la siguiente manera:

Si p es un número primo entonces para todo elemento distinto de cero $\overline{a} \in \mathbb{Z}_p$, se tiene que:

$$(\overline{a})^{p-1} = \overline{1}$$

1.1.4.2. *El Teorema de Euler.* Este resultado, debido a Euler es una generalización del Teorema de Fermat y se puede enunciar de la siguiente manera:

TEOREMA 1.1.11. *Sea n un entero (positivo) y a cualquier entero primo relativo con n. Entonces,*

$$a^{\varphi(n)} \equiv 1 \bmod n$$

Como se puede ver fácilmente, si n es un número primo se tiene el Teorema de Fermat.

El Teorema de Euler tambien se puede enunciar de la siguiente manera:

Si n es un entero (positivo), entonces para toda unidad $\overline{a} \in \mathbb{Z}_n$, se tiene que:

$$\overline{a}^{\varphi(n)} = \overline{1}$$

Capítulo 2

El sistema de cifrado RSA

En este capítulo se describirá el sistema de cifrado de llave pública conocido como RSA, por las iniciales de los apellidos de las personas que lo diseñaron. Este sistema esta basado en los enteros modulares y resultados clásicos de Teoría de Números.

2.0.5. Motivando el sistema RSA. Supóngase que **A**(nita) desea enviar por algun medio convencional de comunicación, por ejemplo internet, un mensaje a **B**(árbara) de tal manera que si este es interceptado, no se pueda conocer su contenido, es decir se mantenga en secreto. Una forma de hacerlo es que **A** lo transforme de manera que nadie mas conozca esa información, es decir, se cifre o encripte.

Ahora se tiene una pregunta: *B*(árbara) al recibir la información cifrada, ¿como recupera el mensaje original? Una solución es la siguiente:

- Ambas partes **A** y **B** acuerdan elegir un número entero, digamos $n = 21$.
- La parte **B** elige un entero, digamos $e = 5$ y hace del conociento de **A** la pareja $(21, 5)$.
- Si la parte **A** desea enviar a la parte **B** cierta información, digamos $m = 4$, que se mantenga en secreto durante el proceso de ser enviada, **A** envia:

  $$m^5 = 4^5 = 16$$

  es decir, 4^5 módulo 21.

Si en el proceso algúna entidad no autorizada intercepta la información enviada por **A**, ¿como recupera el mensaje original?

Recuperar el mensaje original, equivale a encontrar raices quíntas en el anillo de enteros modulares $\mathbb{Z}/21\mathbb{Z}$. Cuando el número n es pequeño como en el ejemplo que se esta trabajando, es fácil encontrar las raíces, pero cuando n es grande, digamos del orden de 100 dígitos) o mas, es bastante díficil encontrar raíces t-ésimas, a menos que se tenga alguna información adicional.

Entonces, la parte **B**, ¿como recupera el mensaje original que le envió **A**?

En este caso la información adicional y la solución para que **B** recupere el mensaje original esta basada en la siguiente observación:

$$(4^5)^5 = (4^2)^5 = (4^5)^2 = (4^2)^2 = 4^4 = 4$$

ya que $4^3 = 1$ en el anillo $\mathbb{Z}_{21}$.

Así, **B** recupera el mensaje original elevando nuevamente a la potencia 5 la información que recibió.

NOTA. Esta es la idea de las llamadas **funciones de un solo sentido** con "puerta secreta", es decir, fáciles de aplicar pero (muy) difíciles de obtener la inversa, excepto que se tenga algúna información adicional ("puerta secreta", "trapdoor").

El ejemplo anterior es un caso particular del sistema de cifrado RSA, el cual se describirá en la siguiente sección.

2.0.6. El sistema de cifrado RSA. Con los resultados sobre los enteros modulares recordados en el capítulo anterior, en esta sección se describirá el sistema de cifrado de llave pública RSA. El nombre "RSA" viene de las iniciales de los apellidos de las personas que lo diseñaron: Rivest, Shamir, Adleman.

Los siguientes son los principales ingredientes que se necesitan en el diseño del RSA:

(1) Elegir dos números primos distintos grandes p y q (del orden de 100 dígitos) y tomar el entero $n = pq$.
(2) Elegir dos elementos $e, d \in \mathbb{Z}_n$ tales que $d = e^{-1}$.
(3) La pareja (n, e) se hace pública.
(4) El elemento d es privado.
(5) La información a cifrar se toma como un elemento de $\mathbb{Z}_n$.

Si **A**(na) desea enviar el mensaje M a **B**(erta) cifrado con RSA, **B** usa la función:

$$\varphi : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n,\ \varphi(x) = x^e$$

Por consiguiente **A** determina el elemento $C = (M^e)_n$ y lo envia a **B**.

Para recuperar el mensaje original, **B** usa la siguiente función:

$$\psi : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n,\ \psi(y) = y^e$$

En particular se la aplica a la inforación recibida:$C = (M^e)_n$

$$(C^d)_n = (((M^e)_n)^d)_n$$

Como e y d son inverso uno del otro en $\mathbb{Z}_n$, se tiene que:

$$(C^d) = M$$

Veamos como se eligen los elementos e y d. Para esto observemos lo siguuiente:

dado que $|\mathbb{Z}_n^*| = \phi(n) = (p-1)(q-1)$, es decir, $\phi(n)$ es el orden del grupo $\mathbb{Z}_n^*$, se tiene que $x^{\phi(n)} = 1$ para todo $x \in \mathbb{Z}_n^*$ (Teorema de Euler), por lo tanto:

$$x^{\phi(n)+1} = x, \text{para todo } x \in \mathbb{Z}_n^*.$$

El elemento e se toma tal que $(e, \phi(n)) = 1$, es decir, $e \in \mathbb{Z}_n^*$, en otras palabras, e es una unidad de $\mathbb{Z}_n$.

Obsérvese que $(e, \phi(n)) = 1$ es equivalente a $(e, p-1) = (e, q-1) = 1$ ya que $n = pq$ y $\phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$.

Una vez elegido el elemento e, su inverso d se determina de la siguiente manera:

Como $(e, \phi(n)) = 1$, por el algoritmo extendido de Euclides, se pueden determinar enteros α y β tales que;

$$\alpha\phi(n) + \beta e = \alpha(p-1)(q-1) + \beta e = 1$$

Entonces, $(\beta)_n(e) - n = 1$, es decir, $(\beta)_n = e^{-1} = d$, en el anillo $\mathbb{Z}_n$.

Observación. Buena parte de la debilidad del sistema radica en que: si se puede determinar $\phi(n) = (p-1)(q-1)$ dado que se conoce e, con la ayuda del algoritmo extendido de Euclides, se puede obtener d.

Determinar $\phi(n)$ es tan difícil como determinar n. Conociendo n, e y d es suficiente para "adivinar" cual es p y q.

Ejemplo

A continuación se dará un ejemplo del sistema RSA, donde los números primos p y q no son grandes.

(1) Identificar el alfabeto con el conjunto $\{0, 1, 2, ..., 25\}$.
(2) Mensaje a cifrar: $m = SI$ (2 letras) $\longrightarrow 19 \cdot 26 + 9 = 503$
(3) Sea: $p = 71, q = 167, n = pq = 11857$, así $\phi(n) = (p-1)(q-1) = 11620$
(4) Llave para cifrar: $e = 117$ y para descifrar: $d = 9733, (d = e^{-1})$

(1) Se envia:

$(SI)^e = (503)^{117} \bmod n = 3702 = 5(26)^2 + 12(260 + 10 = "ELJ"$

(2) Para recuperar el mensaje:

$$(ELJ)^d = (3702)^{9733} \bmod n = 503 = "SI"$$

Algunas cuestiones prácticas.

- ¿como determinar los números primos (grandes, más de 100 dígitos) p y q?

- ¿como realizar aritmética eficiente en $\mathbb{Z}_n^*$, ($n = pq$), (exponenciación).
- Tener una forma eficiente para el algoritmo extendido de Euclides y obtener inversos en el anillo $\mathbb{Z}_n$.

En las siguientes figuras se muestra un texto en claro y el correspondiente cifrado con el sistema RSA. Para tal propósito se uso el software *CrypTool* de uso libre que se puede obtener de internet.

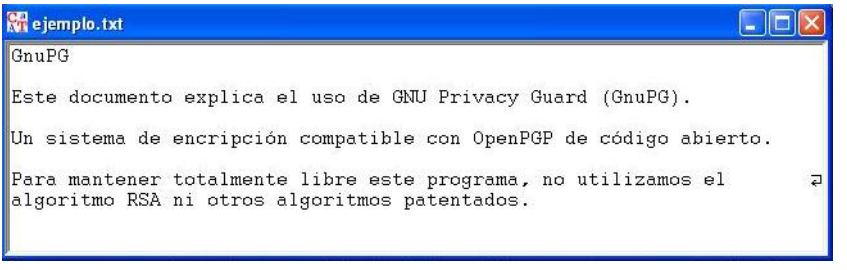

Figure 2.1. Texto en claro

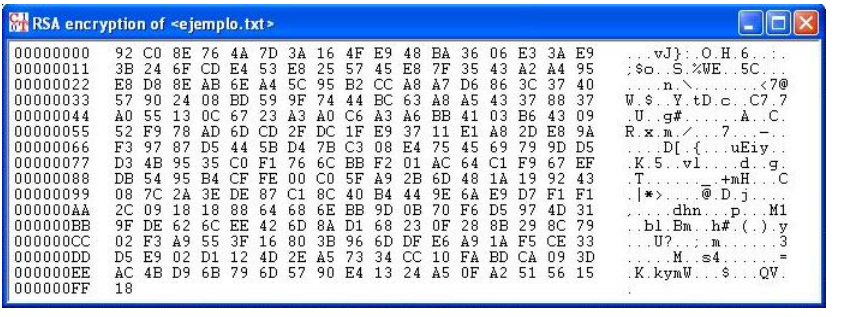

Figure 2.2. Texto cifrado con RSA

Capítulo 3

Construcción de Campos Finitos

En este capítulo se presentarán ejemplos de campos finitos distintos a los enteros modulares introducidos en el capítulo anterior, donde el módulo es un número primo. Estos ejemplos motivarán la introducción de campos finitos mas generales, de los cuales se dará su construcción. En este capítulo y en el resto de las notas $\mathbb{F}_p$ denotará al campo $\mathbb{Z}/p\mathbb{Z}$ de enteros modulares, donde p es un número primo.

3.1. Primeros ejemplos

Sea $\mathbb{F}_2$ el campo de los enteros módulo 2, también conocido como el campo de los *números binarios*. Sea n un entero positivo y sea $\mathbb{F}_2^n$ el producto cartesiano de $\mathbb{F}_2$ consigo mismo n veces decir,

$$\mathbb{F}_2^n = \{\mathbf{a} = (a_0, a_1, ..., a_{n-1}) : a_i \in \mathbb{F}_2\}$$

Es fácil ver que al conjunto $\mathbb{F}_2^n$ se le puede dar la estructura de $\mathbb{F}_2$-espacio lineal (vectorial), de la misma manera que se hace en el caso de los números reales con el producto cartesiano $\mathbb{R}^n$. Obsérvese que $\mathbb{F}_2^n$ tiene cardinalidad 2^n.

Ejemplo 1. Como primer ejemplo veamos como se construye un campo con 4 elementos. Consideremos el $\mathbb{F}_2$-espacio lineal:

$$\mathbb{F}_2^2 = \{(0,0), (1,0), (0,1), (1,1)\}$$

Como este es un espacio lineal, ya sabemos como se suman sus elementos (coordenada a coordenada), pero ahora nos gustaría "multiplicar", en forma natural, los elementos de este espacio lineal. Por ejemplo, ¿cual sería el producto de (0,1) y (1,1)?, de tal manera que el resultado sea un elemento del mismo espacio lineal, es decir, que $\mathbb{F}_2^2$ sea cerrado bajo esta multiplicación. Uno puede darse cuenta rápidamente que no es natural dar un producto entre los elementos de este espacio lineal de tal manera que tenga propiedades como por ejemplo los números reales, racionales, complejos o bien los enteros módulo un primo p. Para definir un producto entre los elementos de

este espacio lineal que tenga propiedades interesantes, identifiquemos a los elementos de $\mathbb{F}_2^2$ con expresiónes de la siguiente manera:

(0,0)	$\leftrightarrow$	0
(1,0)	$\leftrightarrow$	1
(0,1)	$\leftrightarrow$	α
(1,1)	$\leftrightarrow$	$1+\alpha$

es decir, se tiene una biyección entre el conjunto $\mathbb{F}_2^2$ y el conjunto $\mathbf{K} = \{0, 1, \alpha, 1+\alpha\}$. Es fácil ver que al conjunto $\mathbf{K}$ se le puede dar estructura de grupo aditivo el cual es conmutativo (de orden 4). Observése que si en $\mathbf{K}$ se toma el producto, por ejemplo, $\alpha * (1+\alpha) = \alpha + \alpha^2$, no es tal cual un elemento de $\mathbf{K}$. Sin embargo si consideramos la relación algebraica $\alpha^2 = \alpha + 1$, entonces $\mathbf{K}$ es cerrado bajo el producto natural de esas expresiones polinomiales. Mas aún, este conjunto es un campo como se puede ver de la siguiente tabla:

$*$	0	1	α	$\alpha+1$
0	0	0	0	0
1	0	1	α	$\alpha+1$
α	0	α	$\alpha+1$	1
$\alpha+1$	0	$\alpha+1$	1	α

Se acostumbra denotar a este campo finito como $\mathbb{F}_{2^2}$ o bien $GF(2^2)$.

Ejemplo 2. Siguiendo el mismo razonamiento, veamos ahora como se puede construir un campo finito a partir de los elementos del conjunto:

$$\mathbb{F} = \{0, 1, \alpha, \alpha^2, 1+\alpha, 1+\alpha^2, \alpha+\alpha^2, 1+\alpha+\alpha^2\}$$

y considerese la biyección entre estos dos conjuntos dado de la siguiente manera:

Si se trata de multiplicar dos elementos del conjunto $\mathbb{F}$, por ejemplo

$$(\alpha+\alpha^2)(1+\alpha^2) = \alpha + \alpha^2 + \alpha^3 + \alpha^4$$

usando las reglas de multiplicación de polinomios, se ve que la expresión resultante no es ningúno de los elementos del conjunto $\mathbb{F}$. Así como en el ejemplo anterior, para que el cojunto sea cerrado bajo este producto, se necesita que α satisfaga una relación algebraica. Por ejemplo $1+\alpha+\alpha^3 = 0$, o equivalentemente, $\alpha^3 = 1+\alpha$. Usando esta relación, se tiene que:

$$(\alpha+\alpha^2)(1+\alpha^2) = 1+\alpha = \alpha^3$$

Obsérvese que:

$$\alpha + \alpha^2 + \alpha^3 + \alpha^4 = (1+\alpha)(1+\alpha+\alpha^3) + (1+\alpha)$$

es decir, para obtener el resultado del producto $(\alpha+\alpha^2)(1+\alpha^2)$, se efectúa éste como si fueran polinomios, se divide por la expresión $1+\alpha+\alpha^3$ y el

residuo es el resultado de la multiplicación. De este ejemplo se puede inferir que para obtener el resultado del producto de dos elementos de $\mathbb{F}$, se efectúa éste tal cual (como si fueran polinomios), el resultado se divide por la expresión $1+\alpha+\alpha^3$ y el resultado de la multiplicación de los elementos en cuestión, es el residuo de esta división.

Se puede ver fácilmente que $(\mathbb{F}, +, ast)$, donde " $+$ " se toma como la suma usual de polinomios en una indeterminada y " $*$ " es la operación antes definida, tiene estructura de *campo* con cardinalidad 2^3. Este campo se acostumbra denotar por $\mathbb{F}_{2^3}$ o bien por $GF(2^3)$.

Obsérvese que los elementos del campo $\mathbb{F}_{2^3}$ se pueden poner en correspondencia biyectiva con los elementos del $\mathbb{F}_2$-espacio lineal $\mathbb{F}_2^3$, como se puede de la siguiente tabla:

(0,0,0)	$\leftrightarrow$	0
(1,0,0)	$\leftrightarrow$	1
(0,1,0)	$\leftrightarrow$	α
(0,0,1)	$\leftrightarrow$	α^2
(1,1,0)	$\leftrightarrow$	$1+\alpha$
(1,0,1)	$\leftrightarrow$	$1+\alpha^2$
(0,1,1)	$\leftrightarrow$	$\alpha+\alpha^2$
(1,1,1)	$\leftrightarrow$	$1+\alpha+\alpha^2$

Como $\mathbb{F}_{2^3}$ es también un $\mathbb{F}_2$-espacio lineal, se puede ver fácilmente que la biyección anterior es un isomorfismo de espacios vecoriales.

Ejercicios.

(1) Escribir la "tabla de multiplicar" de $\mathbb{F}_{2^3}$.
(2) Comprobar que efectivamente $(\mathbb{F}_{2^3}, +, *)$ es un campo.
(3) Probar que la biyección antes definida es un isomorfismo de espacios lineales.
(4) ¿Cuántas bases, como espacio lineal sobre $\mathbb{F}_2$, tiene el el campo $GF(2^3)$?

Un hecho interesante del campo $GF(2^3)$ es el siguiente: los elementos distintos de cero de este campo se pueden recuperar a partir de las potencias del elemento α:

$$\{\alpha^0 = 1, \alpha^1 = \alpha, \alpha^2, \alpha^3 = 1+\alpha, \alpha^4 = \alpha+\alpha^2, \alpha^5 = 1+\alpha+\alpha^2, \alpha^6 = 1+\alpha^2\}$$

es decir, $GF(2^3)^* = GF(2^3) - \{0\}$ es un *grupo cíclico* de orden $2^3 - 1 = 7$ generado $< \alpha >$.

Al elemento que es generador del grupo multiplicativo de un campo finito se le llama *primitivo*. La siguiente tabla describe a los elementos

del campo $GF(2^3)$ en términos de su estructura aditiva, polinomial y exponencial:

(0,0,0)	0	
(1,0,0)	1	α^0
(0,1,0)	α	α^1
(0,0,1)	α^2	α^2
(1,1,0)	$1+\alpha$	α^3
(1,0,1)	$1+\alpha^2$	α^6
(0,1,1)	$\alpha+\alpha^2$	α^4
(1,1,1)	$1+\alpha+\alpha^2$	α^5

Ejercicio. Determinar todos los elementos primitivos del campo $GF(2^3)$.

Obsérvese que los elementos del campo finito $GF(2^3)$ están expresados como potencias de un elemento (primitivo) α, por lo cual para efectuar la multiplicación de ellos es muy sencillo: basta sumar los exponentes y tomar en cuenta que $\alpha^7 = 1$. Otra observación importante es que para poder realizar todo lo anterior, el polinomio $f(x) = 1 + x + x^3 \in \mathbb{F}_2[x]$ es irreducible sobre $\mathbb{F}_2$ y que el elemento α es raíz de este polinomio, es decir, $f(\alpha) = 0$.

3.2. Costrucción de campos finitos

La construcción del campo $GF(2^3)$ descrita anteriormente, es equivalente a tomar un anillo cociente. En efecto, sea $\mathbb{Z}_2[x]$ el anillo de polinomios en la indeterminada x con coeficientes en el campo $\mathbb{Z}_2$ y sea $f(x) = 1 + x + x^3 \in \mathbb{Z}_2[x]$. Observése que este polinomio es irreducible y por lo tanto el ideal (principal) $\langle f(x)\rangle$ del anillo $\mathbb{Z}_2[x]$ generado por $f(x)$ es primo (de hecho es maximal). Por lo tanto el anillo cociente $R = mathbbZ_2[x]/\langle 1 + x + x^3\rangle$ cuyos elementos son las clases residuales módulo ese ideal, es el siguiente:

$$\mathbb{Z}_2[x]/(1 + x + x^3) = \{0, 1, \alpha, \alpha^2, 1 + \alpha, \alpha + \alpha^2, 1 + \alpha^2, 1 + \alpha + \alpha^2\}$$

donde $\alpha = x + \langle f(x)\rangle$ es la clase residual del polinomio x y satisface la relación $1 + \alpha + \alpha^2 = 0$ ya que el elemento cero de R es la clase de $f(x)$. Por consiguiente $(\mathbb{Z}_2[x]/(1 + x + x^3), +, *)$ es un campo finito con 2^3 elementos.

Ejercicio. Determinar la tabla de multiplicar del campo $GF(2^4)$ y determinar todos sus elementos primitivos (generadores del grupo multiplicativo de este campo).

Ejercicio. ¿Qué sucede si en lugar de tomar el polinomio $1+x+x^3$ en el caso discutido anteriormente, se toma el polinomio $1 + x^2 + x^3$? ¿habrá alguna relación entre los campos finitos que se obtienen con cada uno de esos polinomios?

Ejercicio. Determinar todos los polinomios irreducibles de grado 4 sobre $\mathbb{Z}_2$.

Si ahora se desea construir campos con 2^n elementos, o de manera mas general con p^n elementos, donde p es un primo, siguiendo las ideas descritas anteriormente, se requiere de polinomios $f(x)$ que sean irreducibles sobre el campo $\mathbb{Z}_2$ o en su caso sobre $\mathbb{Z}_p = GF(p)$. Antes recordemos algunas propiedades del anillo de polinomios con coeficientes en cualquier campo.

Sea $\mathbb{K}$ un campo y sea $\mathbf{K}[x]$ el anillo de polinomios en la indeterminada x con coeficientes en $\mathbb{K}$. Este anillo tiene propiedades similares al anillo de los enteros $\mathbb{Z}$, como se puede ver en el siguiente:

TEOREMA 3.2.1. (1) *En $\mathbb{K}[x]$ se tiene el algoritmo de la división: dados dos elementos $a(x), b(x) \in \mathbf{K}[x]$ existen elementos $q(x), r(x) \in \mathbb{F}_p[x]$ tales que:*

$$a(x) = b(x)q(x) + r(x),\ 0 \leq \mathrm{gr}(r(x)) < n$$

donde $n = \mathrm{gr}(b(x))$.

(2) *El anillo $\mathbb{F}_p[x]$ es euclidiano. En particular este anillo es de ideales principales, es decir, todo ideal de este anillo esta generado por un elemento.*

Ejercicio. Dar una demostración de este resultado.

Teniendo un polinomio irreducible $f(x)$ de grado m sobre el campo base $\mathbb{Z}_p$, una manera de construir un campo finito con p^m elementos es la siguiente:

Dado que $f(x) \in \mathbb{Z}_p[x]$ es un polinomio irreducible de grado m, sea $K_m = \mathbb{Z}_p[x]/\langle f(x)\rangle$ el anillo cociente módulo el ideal $\langle f(x)\rangle$ generado por $f(x)$. Como el polinomio $f(x)$ es irreducible entonces el ideal $\langle f(x)\rangle$ del anillo $\mathbb{Z}_p[x]$ es primo, de hecho es maximal. Por consiguiente el anillo cociente es un campo.

Gracias a las propiedades del anillo de polinomios con coeficientes en un campo, es fácil (y muy práctico) determinar un representante de cada elemento del anillo coiente, es decir, de cada clase de equivalencia módulo $\langle f(x)\rangle$. Dado que el anillo de los enteros (racionales) y el anillo de polinomios con coeficientes en un campo tienen muchas propiedades en común, la forma de determinar un representate natural de las clases de equivalencia, es básicamente la misma que se hizo para el caso de los enteros modulares.

Veampos como se hace esto. Sea $\overline{g(x)} = g(x) + \langle f(x)\rangle$ un elemento del anillo cociente K_m y sea $g(x) \in \mathbb{Z}_p[x]$ un representante de esta clase.

Por el algoritmo de la división se tiene que:

$$g(x) = f(x)q(x) + r(x),\ 0 \leq \text{gr}(r(x))\}$$

Por consiguiente: $g(x) \sim r(x) \mod \langle f(x)\rangle$, es decir, $r(x)$ es un representante de la clase $\overline{g(x)}$.

Si $\alpha = x + \langle f(x)\rangle$, es decir, α es la clase residual de x módulo el ideal $\langle f(x)\rangle$, por la observación anterior, los elementos de R_m se pueden identificar con:

$$\{a_0 + a + 1\alpha + a_2\alpha^2 + \cdots + a_{m-1}\alpha^{m-1},\ a_i \in \mathbb{Z}_p\}$$

es decir, cada clase residuale de K_m tienen como representante a un elemento de este conjunto. Como una consecuencia inmediata se tiene que K_m es un espacio lineal sobre $\mathbb{Z}_p$ de dimensión m y por lo tanto $|K_m| = p^m$.

Recordemos como se definen las operaciones en K_m.

La suma " + " de los elementos de K_m es la natural y se define de la siguiente manera: se toman representantes de las clases que se desea sumar, se suman estos como elementos del anillo $\mathbb{Z}_p[x]$ y se toma el residuo módulo el ideal $\langle f(x)\rangle$ de este elemento. Se puede ver fácilmente que esta operación esta bien definida, es decir, no depende de los represenantes de las clases que se tomen, como en el caso de los enteros modulares.

Para definir el producto "$*$ se procede de la misma manera: se toman representantes de las clases, se multiplican como elementos de $\mathbb{Z}_p[x]$ y se toma el residuo módulo el ideal $\langle f(x)\rangle$ de este producto. También es fácil ver que esta definición no depende de los representantes de las clases.

Resumiendo las observaciones anteriores se tiene el siguiente resultado:

TEOREMA 3.2.2. *Sea p un primo, $\mathbb{Z}_p$ el campo finito con p elementos y $f(x) \in \mathbb{Z}[x]$ un polinomio de grado m. Entonces con las operaciones definidas anteriormente la terna $(K_m, +, *)$ es un campo finito con p^m elementos sí y sólo si el polinomio $f(x)$ es irreducible.*

Veamos con un ejemplo que la condición de qe el polinomio $f(x)$ es irreducibe es muyimportante.

Sea $f(x) = x^2 + 1 \in \mathbb{Z}_2[x]$. Los elementos del anillo cociente $\mathbb{Z}_2[x]/\langle x^2 + 1\rangle$ y su tabla de multiplicar se pueden ver en la siguiente

tabla:

$*$	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	1	$x+1$
$x+1$	0	$x+1$	$x+1$	0

Como se puede observar de la tabla, no todos los elementos tiene inverso (bajo la multiplicación). Ademas este anillo tiene divisores cer cero propios: $(x+1)(x+1) = (x+1)^2 = 0$

Ejercicios.

(1) Escribir los elementos de un campo con 2^4 elementos y su tabla de multiplicar.
(2) Determinar dos polinomios irreducibles de grado 3 sobre los números binarios, usar estos para construir campos con 2^3 elementos, determinar en cada caso los elementos que generen al grupo multiplicativo, es decir, los elementos primitivos de cada campo. Ver que relación existe entre esos dos campos.
(3) ¿Cuántos polinomios irreducibles de grado 2,3 existen sobre el campo de los números binarios?

Del resultado anterior se sigue que si se tiene un polinomio irreducible de grado m sobre el camo $\mathbb{Z}$, se puede construir un campo fnito con p^m elementos. Por consiguiente la pregunta ahora es: ¿cómo determinar polinomios irreducibles sobre el campo $\mathbb{Z}_p$?

En el siguiente capítulo se describirá una forma de obtener polinomios irreducibles sobre el anillo $\mathbb{Z}_p[x]$ de polinomios en la indeterminada x con coeficientes en el campo base $\mathbb{Z}_p$ con p elementos. Como se vará, estos polinomios irreducibles no sólo ayudán a construir campos finitos, sino que también son útiles en la descripción de los llamados *códigos cíclicos*, de los cuales el código de Reed-Solomon es un ejemplo. Una de las aplicaciones de este código es en los lectores de CD's.

Capítulo 4

Algunos Resultados Generales

En este capítulo se mencionarán algunos resultados generales sobre los campos finitos introducidos en el Capítulo 4. El lector interesado en detalles y otros resultados relevantes puede consultar alguna de las referencias mencionadas en la bibliografía, o bien, dar una demostración de ellos.

Uno de los primeros resultados sobre campos finitos es el siguiente (Teorema de Fermat):

TEOREMA 4.0.3. Si $\mathbb{F}_p$ es un campo finito con p elementos, entonces todo elemento $a \in \mathbb{F}_p$ satisface la relación $a^p = a$.

Otro resultado que asegura la existencia de los campos finitos es el siguiente:

TEOREMA 4.0.4. Dado un primo p y un entero positivo n, existe un campo finito con p^n elementos. Este campo es único, salvo isomorfismos, y a p se le llama la característica del campo.

Ejemplo 1. Para ilustrar el resultado anterior tomemos el caso $p = 3$ y $n = 2$. Consideremos el anillo de polinomios $\mathbb{F}_3[x]$ con coeficientes en $\mathbb{F}_3 = \{0, 1, 2\}$, los enteros módulo 2, y el polinomio irreducible $f(x) = x^2 + 2x + 1 \in \mathbb{F}_3[x]$. De esta manera el anillo cociente $\mathbb{F}_3[x]/(x^2 + 1)$ tiene como elementos las clases residuales módulo el ideal generado por el polinomio $x^2 + 2x + 1$, es decir:

$$\mathbb{F}_3[x]/(x^2 + 1) = \{0, 1, 2, \beta, 2\beta, 1 + \beta, 1 + 2\beta, 2 + \beta, 2 + 2\beta\}$$

donde β representa la clase residual del polinomio x y satisface la relación $\beta^2 + 2\beta + 1 = 0$, o equivalentemente, $\beta^2 = \beta - 1$. A continuación se presentan las tablas de sumar y multiplicar:

De las tablas es fácil ver que $\mathbb{F}_3[x]/(x^2 + 1)$ es un grupo conmutativo bajo la suma y que los elementos diferentes de cero forman también un grupo bajo el producto. Así que $\mathbb{F}_3[x]/(x^2 + 1)$ es un campo y lo denotamos como $GF(3^2)$ o $\mathbb{F}_{3^2}$.

Obsérvese que $|GF(3^2)| = 3^2 = 9$, por lo tanto es un campo finito de característica 3.

$+$	0	1	2	x	$2x$	$1+x$	$1+2x$	$2+x$	$2+2x$
0	0	1	2	x	$2x$	$1+x$	$1+2x$	$2+x$	$2+2x$
1	1	2	0	$1+x$	$1+2x$	$2+x$	$2+2x$	x	$2x$
2	2	0	1	$2+x$	$2+2x$	x	$2x$	$1+x$	$1+2x$
x	x	$1+x$	$2+x$	$2x$	0	$1+2x$	1	$2+2x$	2
$2x$	$2x$	$1+2x$	$2+2x$	0	x	1	$1+x$	2	$2+x$
$1+x$	$1+x$	$2+x$	x	$1+2x$	1	$2+2x$	2	$2x$	0
$1+2x$	$1+2x$	$2+2x$	$2x$	1	$1+x$	2	$2+x$	0	x
$2+x$	$2+x$	x	$1+x$	$2+2x$	2	$2x$	0	$1+2x$	1
$2+2x$	$2+2x$	$2x$	$1+2x$	2	$2+x$	0	x	1	$1+x$

Tabla 1. Suma

$*$	1	2	x	$2x$	$1+x$	$1+2x$	$2+x$	$2+2x$
1	1	2	x	$2x$	$1+x$	$1+2x$	$2+x$	$2+2x$
2	2	1	$2x$	x	$2+2x$	$2+x$	$1+2x$	$1+x$
x	x	$2x$	2	1	$2+x$	$1+x$	$2+2x$	$1+2x$
$2x$	$2x$	x	1	2	$1+2x$	$2+2x$	$1+x$	$2+x$
$1+x$	$1+x$	$2+2x$	$2+x$	$1+2x$	$2x$	2	1	x
$1+2x$	$1+2x$	$2+x$	$1+x$	$2+2x$	2	x	$2x$	1
$2+x$	$2+x$	$1+2x$	$2+2x$	$1+x$	1	$2x$	x	2
$2+2x$	$2+2x$	$1+x$	$1+2x$	$2+x$	x	1	2	$2x$

Tabla 2. Producto

Ejercicio. Determinar los elementos primitivos del campo $GF(3^2)$.

Nótese que $GF(3) = \{0, 1, 2\}$ es un subcampo de $GF(3^2)$ y que $GF(3^2)$ es una *extensión* finita de grado 3 de $GF(3)$.

Ejercicio. Dar ejemplos de campos finitos con p^n elementos, donde p es un número primo y n es un entero positivo. Determinar sus subcampos.

Otro resultado importante sobre los campos finitos es el siguiente:

TEOREMA 4.0.5. El grupo multiplicativo de un campo finito es cíclico.

PROPOSICIÓN 4.0.6. *Sea $\mathbb{F}_p$ un campo finito con $q = p^r$ elementos y $n \geq 1$. Entonces para $a, b \in \mathbb{F}_p$ se tiene que:*

$$(a+b)^{q^n} = (a)^{q^n} + (b)^{q^n},$$

y

$$(a-b)^{q^n} = (a)^{q^n} - (b)^{q^n}.$$

Capítulo 5

Polinomios irreducibles

En este capítulo se introducirán algunos conceptos que permitirán dar ejemplos de polinomios irreducibles sobre el campo base $\mathbb{Z}_p$ de enteros módulo p (p, primo) los cuales se podrán usar para construir otros campos finitos (ver Cap. 4). Además los polinomios irreducibles se usarán en la descripción de códigos cíclicos (Cap. 8). A lo largo de este capítulo y el resto de estas notas, $\mathbb{F}_p$ también denotará al campo finito $\mathbb{Z}_p$.

5.1. Algunos ejemplos

Recordemos primero que un polinomio $f(x) \in \mathbb{F}_p[x]$ es *reducible* si se puede expresar como producto de dos polinomios propios, es decir, distintos del polinomio $f(x)$ y el polinomio constante igual a 1:

$$f(x) = g(x)h(x)$$

Se dice que $f(x)$ es *irreducible* si no es reducible.

Como ya se ha mencionado antes, el anillo de los enteros (racionales) $\mathbb{Z}$ y el anillo de polinomios $\mathbb{K}[x]$ en una indeterminada con coeficientes en un campo $\mathbb{K}$, tiene muchas propiedades en común. En el anillo de los enteros se tiene el llamado Teorema Fundamental de la Aritmética el cual dice que todo entero se puede expresar en forma única como un producto finito de potencias de números primos. El resultado análogo para el anillo de polinomios es el siguiente:

TEOREMA 5.1.1. Sea $\mathbb{F}_q$ un campo finito y $\mathbb{F}_q[x]$ el anillo de polinomios en la indeterminada x con coeficientes en $\mathbb{F}_q$. Entonces cualquier polinomio $f(x) \in \mathbb{F}_q[x]$ de grado positivo, se puede expresar en forma única como:

$$f(x) = cf_1(x)^{m_1} f_2(x)^{m_2} \cdots f_r(x)^{m_r}$$

donde $c \in \mathbb{F}_q$, $f_1(x), ..., f_r(x)$ son polinomios mónicos irreducibles y $m_1, ..., m_r$ son enteros positivos. Además ésta factorización es única no importando el orden en que aparezcan los factores.

Es obvio que si se tiene la descomposición de un polinomio como se menciona en este resultado, entonces se pueden conocer todos sus factores. Una aplicación interesante de este resultado es cuando se tiene la factorización del polinomio $x^n - 1$ el cual se usará para determinar todos los códigos cíclicos de longitud n definidos sobre un campo finito (ver Cap.8).

Observación. La definición anterior es válida para polinomios con coeficientes en cualquier campo.

Veamos algunos ejemplos de polinomios irreducibles.

Ejemplo 1. El polinomio $f(x) = x^2 + x + 1 \in \mathbb{F}_2[x]$ es irreducible.

En efecto, supóngase que no es irreducible, es decir, es reducible: $f(x) = g(x)h(x)$. Como $f(x)$ es mónico y de grado 2 se sigue que $g(x) = x - a$ y $h(x) = x - b$, para alguna $a, b \in \mathbb{F}_2$. Efectuando el produto $g(x)h(x)$ e igualando coeficientes de las potencias respectivas en $f(x) = g(x)h(x)$, se tienen las relaciones:

$$a + b = 1,\ ab = 1$$

Como se puede ver fácilmente, dado que $a, b \in \mathbb{F}_2$, este sistema no tiene solución en $\mathbb{F}_2$. Por consiguiente el polinomio $f(x)$ no es reducible, es decir, es irreducible.

Ejemplo 2. El polinomio $f(x) = x^3 - x + 1 \in \mathbb{F}_3[x]$ es irreducible.

Procedamos como en el ejemplo anterior: supóngase que $f(x) = g(x)h(x)$, con ambos $g(x), h(x) \in \mathbb{F}_3[x]$. Dado que $f(x)$ es mónico de grado 3, se puede suponer que $g(x) = x - c$ y $h(x) = x^2 + ax + b$ con $a, b, c \in \mathbb{F}_3$. Efectuando las operaciones e igualando coeficientes de las potencias respectivas de la indeterminada x, se tienen las siguientes relaciones:

$$a - c = 0,\ b - ac = 1,\ -bc = 1$$

Analizando los posibles valores que los coeficientes a, b, c pueden tomar en $\mathbb{F}_3$ se llega a una contradicción. Por consiguiente $f(x)$ no es reducible, es decir, es irreducible.

Otra manera de ver que $f(x)$ no es reducible es la siguiente:

Si $f(x) = (x - c)(x^2 + ax + b)$, con $a, b, c \in \mathbb{F}_3$, entonces $f(c) = 0$, es decir, c es una raíz de $f(x)$. Pero se puede ver directamente que ningún elemento de $\mathbb{F}_3$ es raíz de $f(x)$, y por lo tanto se tiene una contradicción.

Ejercicios.

(1) Determinar todos los polinomios irreducibles de grado 3 y 4 sobre los números binarios.
(2) Dar un polinomio irreducible de grado 4 sobre $\mathbb{F}_3$.

Como se puede apreciar de los ejemplos anteriores (y de los ejercicios), resulta complicado determinar polinomios irreducibles sobre el campo base $\mathbb{F}_p$ de grados relativamente grandes. Por consiguiente se necesitan otras técnicas que permitan obtener polinomios irreducibles sobre estos campos.

En el resto de este capítulo se presentarán algunas ideas que conducen a determinar polinomios irreducibles sobre $\mathbb{F}_p$.

El siguiente algoritmo, debido a Rabin, permite determinar polinomios irreducibles:

Entrada: Un polinomio arbitrario $f(x) \in \mathbb{F}_q[x]$ de grado n.
Salida: "$f(x)$ es irreducible" o "$f(x)$ es reducible".

Para $i = 1$ hasta $\lfloor n/2 \rfloor$
  si $\text{mcd}(x^{q^i} - x, f(x)) \neq 1$ entonces
    $f(x)$ es reducible y el algoritmo termina,
  de otra meanera $f(x)$ es irreducible.

Veamos un ejemplo.

Sea $f(x) = x^7 - 1 \in \mathbb{F}_2[x]$. En este caso $q = 2$ y se tiene que para $i = 1$, $\text{mcd}(x^2 - x, x^7 - 1) = x - 1$, por lo tanto $f(x)$ es reducible.

Existen otros algoritmos, en principio mas robustos y sofisticados, que permiten determinar cuando un polinomio es irreducible, aun mas, permiten dar la descomposición como producto de irreducibles. La descomposición de un polinomio tiene una gran variedad de aplicaciones, como ya se mencionó, en Teoría de Códigos (cíclicos), pero también en Criptografía así como en Álgebra y Teoría de Números computacional.

Ejercicio. Determinar la descomposición del polinomio $x^7 - 1$ como producto de irreducibles sobre $\mathbb{F}_2[x]$.

5.2. Polinomios minimales

Para determinar polinomios irreducibles primero se verá cual es el *polinomio minimal* de un elemento de un campo el cual es una extensión finita del campo base $\mathbb{F}_p$.

Sea $q = p^n$ y sea $\mathbb{F}_q$ el campo finito con q elementos (ver Cap. 4). Como el conjunto $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$ es un grupo cíclico de orden $q - 1$ entónces todo elemento α de $\mathbb{F}_q$ satisface la relación:

$$x^q - x = 0$$

(básicamente es una extensión del Teorema de Fermat conocido para los enteros modulares $\mathbb{F}_p$). Obsérvese que este polinomio es mónico

con coeficientes en $\mathbb{F}_p$. Una consecuencia de esta observación es la siguiente:

$$x^{p^n} - x = \prod_{\beta \in \mathbb{F}_{p^n}} (x - \beta)$$

Si α es un elemento del campo finito $\mathbb{F}_q$ con $q = p^r$ elementos, una pregunta natural es: ¿Habrá polinomios mónicos con coeficientes en $\mathbb{Z}_p$ de grado mínimo de los cuales α es raíz? Esta pregunta conduce a la siguiente:

DEFINICIÓN 5.2.1. El polinomio mínimo o minimal sobre $\mathbb{F}_p$ de un elemento $\alpha \in \mathbb{F}_q$ es el polinomio mónico de menor grado con coeficientes en $\mathbb{F}_q$, $M_\alpha(x)$, del cual α es raíz, es decir,

$$M_\alpha(\alpha) = 0$$

Ejercicio.

(1) Determinar el polinomio minimal de los elementos del campo con $\mathbb{F}_8$.
(2) Determinar el polinomio minimal de los elementos del campo $\mathbb{F}_{2^4}$.

Otra forma de definir el polinomio minimal de un elemento es la siguiente.

Sea $\mathbb{F}_q$ un campo finito con $q = p^n$ elementos (p primo y n un entero positivo), $\mathbb{F}_q[x]$ el anillo de polinomios en x con coeficientes en $\mathbb{F}_q$. Si $\gamma \in \mathbb{F}_q$ sea $\mathbb{F}_q[\gamma] = \{a_o + a_1\gamma + \cdots + a_m\gamma^m \ : a_i \in \mathbb{F}_q, m \in \mathbb{N}\}$, es decir, el conjunto de combinaciones lineales de potencias del elemento γ. Considerese la siguiente función:

$$\varphi_\alpha : \mathbb{F}_q[x] \longrightarrow \mathbb{F}_q[\alpha],\ \varphi(f(x)) = f(\alpha)$$

es decir, φ es la función evaluación en α. Es fácil ver que $\mathbb{F}_q[\alpha]$ es un anillo y que φ es un homomorfismo suprayectivo de anillos. El núcleo de φ, $ker(\varphi) = I_\alpha = \{f(x) \in \mathbb{F}_q[x] : \ f(\alpha) = 0\}$ es un ideal de $\mathbb{F}_q[x]$ y dado que este anillo es de ideales principales se tiene que existe un elemento $M_\alpha(x) \in I_\alpha$ tal que $I_\alpha = \langle M_\alpha(x) \rangle$.

Por lo tanto, cualquier polinomio $f(x) \in \mathbb{F}_q[x]$ tal que $f(\alpha) = 0$, es divisible por $M_\alpha(x)$. Además $M_\alpha(x)$ es irreducible y por lo tanto I_α es un ideal primo (y en consecuencia maximal).

Algunas otras propiedades del polinomio minimal de un elemento son las siguientes:

(1) $M_\alpha(x)$ divide a $x^q - x$.
(2) $\text{gr}(M_\alpha(x)) \leq n$.

(3) El polinomio mínimo de un elemento primitivo de $\mathbb{F}_q$, es decir, un generador del grupo multiplicativo $\mathbb{F}_q^*$ tiene grado n. Dicho polinomio se denomina *primitivo*.
(4) $I_\alpha = I_{\alpha^p}$ sí y sólo si $M_\alpha(x) = M_{\alpha^p}(x)$. En particular si α es un elemento primitivo de $\mathbb{F}_{p^n}$ y $M^{(i)}(x)$ es el polinomio mínimo de α^i, entonces

$$M^{(i)}(x) = M^{(pi)}(x)$$

5.3. Clases ciclotómicas

En esta sección se verá como las llamadas *clases ciclotómicas* ayudan a determinar el polinomio mínimal de un elemento.

Recordemos cuales son las clases ciclotómicas. Sean n y q enteros positivos y primos relativos, es decir, su máximo común divisor es igual a 1. Por consiguiente q es una unidad en el anillo de enteros módulo n y como las unidades de $\mathbb{Z}_n$ son un grupo de orden $\phi(n)$, existe un entero m tal que $q^m \equiv 1 \bmod n$. Equivalentemente, n divide a $q^m - 1$. Al entero m se le llama el *orden* multiplicativo de q. Obsérvese que m divide a $\phi(n)$.

Con la notación introducida anteriormente, sea $< q >$ el subgrupo cíclico de $\mathbb{Z}_n$ generado por q (obsérvese que este grupo tiene orden m). Ahora hagamos actuar este subgrupo sobre $\mathbb{Z}_n$:

$$\langle q \rangle \times \mathbb{Z}_n \longrightarrow \mathbb{Z}_n,\ (q^r, s) \longrightarrow sq^r$$

Ejercicio. Probar que esta es una acción de grupo.

Si $s \in \mathbb{Z}_n$, su *órbita* bajo esta acción es:

$$O_s = \{s, sq, sq^2, ..., sq^{m-1}\}$$

A esta órbita se acostumbra llamar la clase *ciclotómica* de s y denotarla por C_s.

Recordemos que si un grupo actua sobre un conjunto, la acción induce una relación de equivalencia en el conjunto: dos elementos son equivalentes si estan en la misma órbita. Por consiguiente las clases ciclotómicas son una partición de $\mathbb{Z}_n$. En particular se tiene que:

$$\mathbb{Z}_n = \bigcup_{s \in \mathbb{Z}_n} C_s$$

Ejemplo. Sea $n = 9$ y $q = 2$. Entonces $< q >= \{1, 2, 4, 8, 7, 5\}$ y las clases ciclotómicas son:

$$C_0 = \{0\},\ C_1 =< 2 >,\ C_3 = \{3, 6\}$$

Ejercicio. Determinar las clases ciclotómicas en los siguientes casos: a) $n = 2^3, 2^4$ y $q = 3$. b) $n = 3^2, 3^3$ $q = 5$.

Se tiene ahora el resultado:

PROPOSICIÓN 5.3.1. *Sea $\mathbb{F}_q$ el campo con $q = p^n$ elementos y $\alpha \in \mathbb{F}_q$ un elemento primitivo. Sea $M^{(i)}(x) \in \mathbb{F}_p[x]$ el polinomio mínimo del elemento α^i y C_i la clase ciclotómica de i módulo $p^n - 1$. Entonces:*

$$M^{(i)}(x) = \prod_{j \in C_i} (x - \alpha^j)$$

DEMOSTRACIÓN. Basta probar que la parte izquierda de la relación anterior es divisible por la parte derecha. □

Como consecuencia de este resultado se tiene el siguiente

TEOREMA 5.3.2. *Con la notación introducida anteriormente:*

$$x^{p^n-1} - 1 = \prod_s M^{(s)}(x)$$

donde s se toma en un conjunto de representantes de las clases ciclotómicas módulo $p^n - 1$.

A continuación se dará un ejemplo que ilustre los resultados anteriores.

Ejemplo 1. Factorizar en producto de polinomios irreducibles sobre $\mathbb{F}_2$ el polinomio $x^9 - 1$. En este caso $n = 9$, $q = 2$, es decir, $\mathbb{F}_q = \mathbb{F}_{3^2}$. Sea $\alpha \in \mathbb{F}_3$ un elemento primitivo el cual satisface la relación $z^2 + z + 1 = 0$. Como se vió anteriormente, las clases ciclotómicas son $C_0 = \{0\}$, $C_1 = \{1, 2, 4, 8, 7, 5\}$, $C_3 = \{3, 6\}$. Por lo tanto:

$$M^{(0)} = (x - 1)$$

$$M^{(1)}(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^4)(x - \alpha^5)(x - \alpha^7)(x - \alpha^8) = x^6 + x + 1$$

$$M^{(3)}(x) = (x - \alpha^3)(x - \alpha^6) = x^2 + x + 1$$

y se tiene que:

$$x^9 - 1 = (x - 1)(x^6 + x + 1)(x^2 + x + 1)$$

Ejercicio. Usando las clases ciclotómicas factorizar $x^7 - 1$ sobre $\mathbb{F}_2$.

Capítulo 6

Introducción a los códigos lineales

En este capítulo se motivará el concepto de *código*, se tratarán algunos aspectos de los códigos lineales y se darán varios ejemplos de tales códigos. El lector interesado en profundizar en el estudio de la teoría de códigos y su relación con otras áreas de la matemática como la geometría algebraica, campos finitos, combinatoria, criptografía, geometrias finitas, entre otras, le sugerimos consultar particularmente la referencia [3].

6.1. Consideraciones Generales

En esta primera parte se introducirá el concepto de *código*, se darán algunos ejemplos y varias definiciones que serán útiles a lo largo de este capítulo.

6.1.1. ¿Qué es una palabra? Comenzaremos con un concepto general:

DEFINICIÓN 6.1.1. Un **alfabeto de código** es simplemente un conjunto $A = \{a_1, ..., a_n\}$ de cardinalidad n. Una **palabra** sobre A es una colección (finita) de elementos de A. La **longitud** de una palabra es el número de elementos del alfabeto que la componen.

Ejemplos.

(1) Si $A = \{a, b, 1, 2, 3\}$, entonces $a, ab, a2b, aabb223$ son palabras sobre el alfabeto A, de longitud 1,2,3,7, respectivamente.
(2) Si $A = \mathbb{F}_2 = \{0, 1\}$, el conjunto de los números binarios, entonces $1, 0, 11, 1010, 1111$ son palabras sobre $\mathbb{F}_2$, de longitud 1,1,2,4,4, respectivamente.
(3) Si $A = \mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$, el conjunto de los números enteros módulo 5, entonces $1, 0, 113, 231010, 441111$ son palabras sobre $\mathbb{Z}_5$.
(4) Si $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2 = \alpha+1\}$ es un campo finito con 4 elementos, $\alpha\alpha, \alpha^2\alpha, 1\alpha^2$ son palabras sobre el alfabeto $\mathbb{F}_4$.

Ejercicio. Si A es un alfabeto de cardinalidad $n > 1$ probar que:

(1) Hay n^k palabras de longitud k.
(2) Hay $\frac{n^{k+1}-1}{n-1}$ palabras de longitud a lo mas k.
(3) El número de palabras de longitud n sobre $\mathbb{Z}_r$, los enteros módulo r, con exactamente k ceros es: $\binom{n}{r}(r-1)^{n-k}$.

Ejemplos.

(1) En $\mathbb{Z}_2^8$ hay $\binom{8}{3} = 56$ elementos con exactamente tres ceros.
(2) En $\mathbb{Z}_3^6$ hay $\binom{6}{2}2^4 = 240$ elementos con exactamente dos ceros.

6.1.2. ¿Qué es un código? La siguiente definición será importante a lo largo de estas notas.

DEFINICIÓN 6.1.2. Un **código** sobre un alfabeto $\mathcal{A}$ es un conjunto de palabras, que pueden ser de distinta longitud.

Por ejemplo,

$$\mathcal{C} = \{10, 1, 001, 1111, 101\}$$

es un código sobre el alfabeto de los números binarios $\mathbb{F}_2$.

En general los códigos se usan para codificar mensajes. Por ejemplo, el código $\mathcal{C}$ mencionado anteriormente se puede usar para codificar las primeras cinco letras del alfabeto, es decir,

$$\begin{array}{ccc} a & \longrightarrow & 10 \\ b & \longrightarrow & 1 \\ c & \longrightarrow & 001 \\ d & \longrightarrow & 1111 \\ e & \longrightarrow & 101 \end{array}$$

Por lo tanto se puede codificar palabras (o mensajes) a partir de esas letras codificadas, por ejemplo:

$$deba \longrightarrow 1111101110$$

Los códigos mas usados en la práctica son los binarios, es decir, donde el alfabeto es $\mathbb{F}_2$, el campo de los números binarios.

DEFINICIÓN 6.1.3. Sea $A = \{a_1, ..., a_n\}$ y $\mathcal{C}$ un código de longitud n sobre el alfabeto A. Una función de **codificación**, f, es simplemente una función biyectiva de A sobre $\mathcal{C}$, es decir, $f\colon A \longrightarrow \mathcal{C}$. A la pareja $(\mathcal{C}, f)$ se le llama un **esquema de codificación** para el alfabeto A.

Ejemplo. Las 26 letras del alfabeto inglés se pueden codificar de la siguiente manera: El alfabeto fuente es $S = \{a, b, c, ..., z\}$, el alfabeto

del código es $A = \{0, 1, 2, ..., 9\}$ y el código es $C = \{00, 01, 02, ..., 25\}$. La función de codificación se define como:

$$\begin{array}{lllll} f(a) = 00, & f(b) = 01, & f(c) = 02, & f(d) = 03, & f(e) = 04 \\ f(f) = 05, & f(g) = 06, & f(h) = 07, & f(i) = 08, & f(j) = 09 \\ f(k) = 10, & f(l) = 11, & f(m) = 12, & f(n) = 13, & f(o) = 14 \\ f(p) = 15, & f(q) = 16, & f(r) = 17, & f(s) = 18, & f(t) = 19 \\ f(u) = 20, & f(v) = 21, & f(w) = 22, & f(x) = 23, & f(y) = 24 \\ f(z) = 25 & & & & \end{array}$$

Esta función de codificación se puede usar para codificar mensajes:

$$f\text{: “un gran jefe”} \longrightarrow 201306171309040504$$

Obsérvese que si el código es $C = \{0, 1, 2, 3, \ldots, 25\}$ se tiene que distintos mensajes tienen la misma codificación. Por ejemplo:

$$\text{casa} \longrightarrow 20180, \quad \text{cabia} \longrightarrow 20180$$

Ejemplo (el código ASCII). Las iniciales “ASCII" significan *American Standard Code for Information Interchange*. Este código se usa en las computadoras, por ejemplo para almacenar caracteres en la memoria. Este código consiste de 256 caracteres que incluyen letras minúsculas, mayúsculas y signos puntuación, entre otros. La función de codificación f es que cada uno de esos caracteres es identificado con un elemento de $\mathbb{F}_2^8$. Por ejemplo la letra mayúscula “A" se identifica con la colección (10000001), (el número 65 en notación decimal), es decir, $f(A) = (10000001)$. A cada colección de 8 bits (ceros y unos) se le llama “byte", así los elementos del código ASCII estan en correspondencia con los bytes.

En la siguiente tabla se muestra una parte del código ASCII.

El código ASCII (parcial)								
A	$\longrightarrow$	01000001	J	$\longrightarrow$	01001010	S	$\longrightarrow$	01010011
B	$\longrightarrow$	01000010	K	$\longrightarrow$	01001011	T	$\longrightarrow$	01010100
C	$\longrightarrow$	01000011	L	$\longrightarrow$	01001100	U	$\longrightarrow$	01010101
D	$\longrightarrow$	01000100	M	$\longrightarrow$	01001101	V	$\longrightarrow$	01010110
E	$\longrightarrow$	01000101	N	$\longrightarrow$	01001110	W	$\longrightarrow$	01010111
F	$\longrightarrow$	01000110	O	$\longrightarrow$	01001111	X	$\longrightarrow$	01011001
G	$\longrightarrow$	01000111	P	$\longrightarrow$	01010000	Y	$\longrightarrow$	01011001
H	$\longrightarrow$	01000001	Q	$\longrightarrow$	01010001	Z	$\longrightarrow$	01011010
I	$\longrightarrow$	01001001	R	$\longrightarrow$	01010010	espacio	$\longrightarrow$	00100000

Los códigos se dividen en dos grandes grupos: **de bloque**, es decir aquellos cuyas palabras tiene la misma longitud n y los de **longitud variable**. Por ejemplo el código ASCII es de bloque ya que cada palabra

tiene la misma longitud $n = 8$. Al entero n se le llama la **longitud** del código $\mathscr{C}$ y la cardinalidad M de $\mathscr{C}$ el **tamaño** (o cardinalidad) del código. En este caso se dice que $\mathscr{C}$ es un (n, M) código. En particular si $A = \mathbb{F}_q$ es un campo finito con q elementos, se dice que $\mathscr{C}$ es un código sobre $\mathbb{F}_q$. Para cuestiones prácticas los códigos binarios, es decir, definidos sobre el campo $\mathbb{F}_2$ son los mas usados.

La longitud de un código así como su cardinalidad son algunos de los parámetros de los códigos. Mas adelante cuando se definan los códigos lineales, la dimensión, y su distancia mínima serán parámetros importantes (conociendo su dimensión se determina su cardinalidad).

Ejemplo (código de repetición). El código de repetición de longitud $n = 7$ es

$$\{0000000, 1111111\}$$

La cardinalidad de este código es $M = 2$, así es que se tiene un $(7, 2)$ código.

Ejercicios.

(1) Determinar el número de funciones de codificación del alfabeto $S = \{a, b\}$ en el código $C = \{0, 1\}$
(2) Determinar el número de funciones de codificación del alfabeto $S = \{a, b, c\}$ en el código $C = \{00, 01, 11\}$
(3) Hallar una fórmula para el número de funciones de codificación de un alfabeto de tamaño n en un código del mismo tamaño.

6.2. Códigos Lineales

Una de las clases mas interesantes de códigos de bloque es la de los *códigos lineales*. En esta sección se introducirán estos códigos y se darán algunas de sus principales propiedades. Requisitos mínimos para estudiar estos códigos son resultados básicos de Álgebra Lineal (equivalente a un primer curso de esta asignatura en un plan de estudios de una licenciatura en Matemáticas, Ingeniería o Ciencias de la Computación). Normalmente los conceptos básicos de este curso se introducen y desarrollan sobre espacios vectoriales (lineales) definidos sobre el campo de los números reales ($\mathbb{R}$) o sobre el campo de los números complejos (**C**). En el caso que nos interesa esos conceptos son tratados sobre los llamados *campos finitos* (o de Galois).

6.2.1. Definición de código lineal. Suponiendo conocidos conceptos básicos de Álgebra Lineal, procedemos ahora a introducir los *códigos lineales*. Para fijar ideas y motivar los conseptos consideremos el campo de los números binarios $\mathbb{Z}_2 = \mathbb{F}_2 = \{0, 1\}$. Como el lector podrá observar a lo largo de las siguientes páginas, la mayoría de las definiciones

y propiedades que se describirán son válidas cuando se reemplaza el campo de los números binarios por cualquier campo finito. Como ya se mencionó antes, desde un punto de vista de aplicaciones, el campo de los binarios es el mas apropiado, pero desde un punto de vista matemático se puede trabajar con cualquier campo finito.

DEFINICIÓN 6.2.1. Sea $\mathbb{F}_q$ un campo finito con $q = p^n$ elementos. Un código lineal sobre $\mathbb{F}_q$ de longitud n es simplemente un subespacio lineal del espacio (lineal) $\mathbb{F}_q^n$.

Ejemplos

(1)
$$\mathscr{C} = \{(0000), (0110), (1101), (1011)\}$$

(2)
$$\mathscr{C} = \{(00000), (11001), (01101), (10100)\}$$

Como es obvio, un subespacio lineal de $\mathbb{F}_2^n$ (o de $\mathbb{F}_q$) tiene una dimensión $k \leq n$. Así en el caso de un código lineal $\mathscr{C}$, a la dimensión se llama la dimensión del código.

Si un código lineal sobre el campo $\mathbb{F}_q$, $\mathscr{C}$, tiene longitud n y dimensión k, decimos que es un $[n, k]_q$-código lineal. Otro de los parámetros interesantes de un código lineal es su distancia mínima, la cual se definirá en la siguiente sección en relación a los códigos lineales detectores-correctores de errores.

6.2.2. Matriz generadora. Para describir varias de las propiedades de un espacio lineal de dimensión finita k sobre un campo, basta conocer una base de este espacio, en particular un conjunto de generadores. En el caso de un códigos lineal $\mathscr{C}$, si $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$, donde $\mathbf{v_i} = (v_{i1}, v_{i2}, \ldots, v_{in}) \in \mathbb{F}_q^n$, es un conjunto de generadores de $\mathscr{C}$, la matriz G cuyos renglones (de longitud n) son esos generadores de $\mathscr{C}$, se le llama matriz *generadora* del código $\mathscr{C}$. Como se puede observar, hay tantas matrices generadoras de un código lineal como conjunto de generadores tenga este espacio lineal.

Ejemplos

(1) Una matriz generadora del código lineal $\mathscr{C} = \{(0000), (0110), (1101), (1011)\}$ es:

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

(2) Una matriz generadora del código lineal $\mathscr{C} = \{(00000), (11001), (01101), (10100)\}$ es:

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

Ejercicio. Si la siguiente matriz es la matriz generadora de un código lineal binario, determinar todos los elementos de este código.

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

Ejercicios

(1) Dar otras matrices generadoras de los códigos lineales mencionados en los ejemplos anteriores.
(2) En estos ejemplos, ?'Cual de las matrices generadoras es la mas interesante (adecuada)?

6.3. Código lineal detector-corrector de errores

A continuación se motivará el concepto de *código lineal detector-corrector de errores*.

Supongase que se desea enviar, de una fuente emisora a otra receptora, el mensaje $\{\mathbf{SI}, \mathbf{NO}\}$ el cual se identifica, usando el alfabeto binario, por ejemplo con la siguiente cadena de bits (palabra):

$$\mathbf{SI} \longrightarrow \mathbf{0000}, \quad \mathbf{NO} \longrightarrow \mathbf{1111}$$

Supóngase que la estación receptora recibe la siguiente cadena de bits:

$$\mathbf{0101}\ ,\ \ \mathbf{1100}$$

La pregunta es: ?'cuál es la información enviada por la fuente emisora?

Veamos con un poco mas de detalle un ejemplo.

Supóngase que se desea enviar, de una estación emisora a una estación receptora, el siguiente mensaje de $k = 4$ símbolos binarios $\mathbf{u} = (0111)$. Antes de enviar este mensaje, se codifica en otro con $n = 6$ símbolos (binarios) $\mathbf{x} = (x_1 x_2 \cdots x_6)$ de la siguiente manera:

$$x_1 = 0,\ x_2 = 1,\ x_3 = 1,\ x_4 = 1, \tag{6.1}$$

A estos símbolos se les llama de *información*. Los restantes $n - k = 2$ símbolos se eligen como una combinación lineal de los símbolos de información, por ejemplo:

$$x_5 = x_1 + x_4,\ x_6 = x_1 + x_2 + x_3 + x_4 \tag{6.2}$$

A estos símbolos se les llama de *chequeo de paridad*.

Obsérvese que los símbolos de chequeo de paridad satisfacen las siguientes relaciones, tomando en cuenta que en el campo de los números binarios se tiene que $1 = -1$, es decir, el inverso (aditivo) de 1 es el mismo:

$$x_1 + x_4 + x_5 = 0,\ x_1 + x_2 + x_3 + x_4 + x_6 = 0 \tag{6.3}$$

es decir, los símbolos de información y de paridad satisfacen un sistema homogéneo de ecuaciones lineales con coeficientes en el campo de los números binarios $\mathbb{F}_2$, dados por las relaciones (1) y (3).

En términos matriciales este sistema se puede escribir de la siguiente manera:

$$H \cdot \mathbf{X}^t = 0 \tag{6.4}$$

donde

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

y $\mathbf{X} = (x_1, ..., x_6)$. La matriz H se denomina de *paridad*.

Así los símbolos de información y de paridad son soluciones del sistema lineal homogéneo (4). Como es bien sabido, el conjunto $\mathscr{C}$ de soluciones de un sistema de esta naturaleza es un subespacio lineal, el cual como tal tiene una dimensión. Por consiguiente en este caso, $\mathscr{C}$ es un código lineal de longitud 6. (?'cuál es su dimensión?).

Veamos otros ejemplos de cómo se codifica un mensaje original usando la matriz de paridad H descrita anteriormente.

En el ejemplo mencionado al inicio de esta sección, a la inforamación **SI** se le asoció (0000) y a **NO** se le asoció (1111). Usando la matriz H, esta información queda codificada como (000000) y (111111), respectivamente.

Si el mensaje original es $\mathbf{u} = (1010)$ entonces el mensaje codificado $\mathbf{x} = (x_1, ..., x_6)$ con la matriz H es: $\mathbf{x} = (101010)$; si $\mathbf{u} = (1110)$, el mensaje codificado es: $\mathbf{x} = (111011)$.

Ejercicio. Determinar el espacio de soluciones del sistema $H \cdot \mathbf{X}^t = 0$ del ejemplo anterior.

El ejemplo anterior se puede generalizar fácilmente de la siguiente manera:

Supóngase que se desea enviar, de una estación emisora a una estación receptora, un mensaje de k símbolos binarios $\mathbf{u} = u_1 u_2 \cdots u_k$. Antes de enviar este mensaje, se codifica como $\mathbf{x} = x_1 x_2 \cdots x_n$ con $n \geq k$ símbolos (binarios) de la siguiente manera:

$$x_1 = u_1, \quad x_2 = u_2, ..., x_k = u_k, \tag{6.5}$$

Los $n - k$ símbolos restantes $x_{k+1}, ..., x_n$ son combinación lineal de los símbolos de información $x_1, ..., x_k$, por lo tanto los elementos $x_1, ..., x_n$ satisfacen relaciones de la forma:

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = 0 \\ &a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = 0 \\ &\vdots \\ &a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n = 0 \end{aligned} \tag{6.6}$$

donde los coeficientes a_{ij} son elementos del campo de los números binarios, es decir, son 0 o 1. En otras palabras, los símbolos $x_1, ..., x_n$ satisfacen el sistema lineal homogéneo:

$$H \cdot \mathbf{X}^t = 0 \tag{6.7}$$

donde

$$H = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kn} \end{pmatrix}$$

es la matriz de *paridad* y $\mathbf{X} = (x_1, ..., x_n)$.

Nuevamente, el conjunto de vectores $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{F}_2^n$ que son solución del sistema (7) forman un subespacio vectorial $\mathscr{C}$ de $\mathbb{F}_2^n$, es decir, $\mathscr{C}$ es un código lineal (binario), el cual tiene dimensión finita $k \leq n$. A los elementos de este subespacio $\mathscr{C}$ se les llama **palabras codificadas.**

Las observaciones anteriores permiten dar la siguiente definición:

DEFINICIÓN 6.3.1. Un código lineal detector-corrector de errores sobre un campo finito $\mathbb{F}_q$, $\mathscr{C}$, con matriz de chequeo de paridad H, es el subespacio lineal de soluciones del sistema lineal homogéneo

$$H \cdot \mathbf{X}^t = 0$$

donde $\mathbf{X} = (x_1, ..., x_n)$.

Si la matriz H tiene n columnas y la dimensión del subespacio $\mathscr{C}$ es k, entonces se dice que $\mathscr{C}$ es un $[n, k]$-código lineal binario. Al entero n se le llama la *longitud* y al entero k la *dimensión* del código.

La longitud y la dimensión de un código lineal son dos de los principales parámetros. Otro de ellos es su *distancia mínima*, la cual se introducirá en breve, pero antes daremos algunos ejemplos de códigos lineales (binarios) detector-corrector de errores.

Observación. Si $\mathscr{C}$ es un $[n, k]$-código lineal (binario) entonces $\mathscr{C}$ tiene 2^k elementos.

Ejemplos

(1) Es fácil ver que el código lineal con matriz de chequeo de paridad

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

es:

$$\mathscr{C} = \{(0000), (0110), (1101), (1011)\}$$

el cuál es un $[4, 2]$-código.

(2) El $[5, 2]$-código lineal (binario) determinado por la matriz de chequeo de paridad

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

es:

$$\mathscr{C} = \{(00000), (11001), (01101), (10100)\}$$

(3) El código lineal $\mathscr{H}_3$ determinado por la matriz de chequeo de paridad

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

es un $[7, 4]$-código y es un ejemplo de una clase muy interesante de códigos lineales: los códigos de *Hamming* (ver Sección 7. 4.1) .

Ejercicio

(1) Determinar una base de los códigos lineales de los ejemplos 1 y 2 dados anteriormente.

(2) Determinar una base y todos los elementos (palabras) del código $\mathscr{H}_3$.

(3) ?'Qué propiedades puedes observar del código de Hamming $\mathcal{H}_3$?

6.3.1. Matriz de paridad y matriz generadora. Como se verá a continuación, es fácil obtener una matriz generadora de un código lineal a partir de una matriz de paridad y viceversa. Para esto es necesario la siguiente definición:

Definición 3.1.2 *Se dice que una matriz de paridad H de un código lineal $\mathscr{C}$ está en forma estándard si tiene la siguiente expresión:*

$$H = \begin{pmatrix} A & I_{n-k} \end{pmatrix}$$

donde A es una matriz de tamaño $k \times (n-k)$ e I_{n-k} es la matriz identidad de tamaño $n-k$.

Es fácil ver que si $\mathbf{u} = (u_1, u_2, ..., u_k)$ son los símbolos de información y $\mathbf{x} = (x_1, x_2, ..., x_n)$ es la palabra codificada, entonces

$$\mathbf{x} = \mathbf{u} \cdot G$$

donde

$$G = \begin{pmatrix} I_k & A^t \end{pmatrix}$$

y A^t es la matriz transpuesta de A. Es decir, G es una matriz generadora del código $\mathscr{C}$.

Obviamente si $G = \begin{pmatrix} I_k & B \end{pmatrix}$ es una matriz generadora de un código lineal $\mathscr{C}$ de longitud n, entónces $H = \begin{pmatrix} B^t & I_{n-k} \end{pmatrix}$ es una matriz de paridad de $\mathscr{C}$.

Veamos algunos ejemplos.

(1) Si la matriz de paridad de un código es (ver ejemplo 1 anterior):

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

la cual está en su forma estándard, entonces una matriz generadora de este código es:

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

(2) En el ejemplo 2 mencionado anteriormente se tiene que la matriz de paridad del $[5, 2]$-código es

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

la cual está en la forma estándard, entonces una matriz generadora de este código es:

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

(3) Una matriz generadora del código de Hamming $\mathcal{H}_3$ es:

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

Observación. Es fácil ver que en estos ejemplos se tiene que $G \cdot H = 0$ y que $H \cdot G = 0$.

Ejercicio. Probar en general que si un código lineal $\mathcal{C}$ tiene matriz de paridad H y matriz generadora G, entonces: $G \cdot H = 0$ y $H \cdot G = 0$.

Observación. Si T_H y T_G denotan la trasformación lineal determinada por H y G respectivamente, lo anterior quiere decir que ker(T_H)=Im(T_G), donde, como siempre, "ker", "Im" denotan el núcleo e imagen de una trasformación lineal respectivamente.

6.3.2. Distancia mínima. Otro de los parámetros interesantes de un código lineal detector-corrector de errores, y quizá, el mas importante para cuestiones de decodificación, es la *distancia mínima*. Para introducir este concepto, primero recordaremos qué es el *peso de Hamming* de una palabra.

DEFINICIÓN 6.3.2. El *peso de Hamming*, $p_H(\mathbf{u})$, de la palabra $\mathbf{u} = (u_1, u_2, ..., u_n) \in \mathbb{F}_q^n$ es el número de coordenadas distintas de cero de $\mathbf{u}$.

Si sop($\mathbf{u}$) = $\{i : u_i \neq 0\}$, es el *soporte* de $\mathbf{u}$, entonces $p_H(\mathbf{u})$=$|sop(\mathbf{u})|$, donde $|A|$ denota la cardinalidad del conjunto A.

Ejemplos

(1) La palabra (1011) $\in \mathbb{F}_2^4$ tiene peso de Hamming igual a 3.
(2) El peso de la palabra (0001111) del código de Hamming $\mathcal{H}_3$ introducido anteriormente es igual a 4.

Ejercicio Determinar el peso de las palabras de los códigos introducidos en los ejemplos anteriores.

Con la ayuda del peso de Hamming de una palabra se puede definir la distancia entre dos palabras de un código lineal:

La *distancia* de Hamming entre las palabras $\mathbf{u}$ y $\mathbf{v}$ de $\mathbb{F}_q^n$ es:

$$d_H(\mathbf{u}, \mathbf{v}) = \mathrm{p}_H(\mathbf{u} - \mathbf{v})$$

Ejercicio. Probar que la distancia de Hamming induce una métrica en el espacio $\mathbb{F}_q^n$.

Ahora se puede definir la distancia mínima de un código lineal:

DEFINICIÓN 6.3.3. La *distancia mínima del códigos lineal* $\mathscr{C}$ *es:*

$$d_H(\mathscr{C}) = \min_{\mathbf{u}\neq\mathbf{v}}\{d_H(\mathbf{u},\mathbf{v})\} = \min_{\mathbf{w}\neq 0}\{p_H(\mathbf{w})\}$$

Si un $[n,k]_q$-código lineal $\mathscr{C}$ tiene distancia mínima d, entonces decimos que es un $[n,k,d]_q$-código lineal.

Ejemplos

(1) La distancia mínima del código $\mathscr{C} = \{(0000), (0110), (1101), (1011)\}$ es 2.
(2) La distancia mínima del código $\mathscr{C} = \{(00000), (11001), (01101), (10100)\}$ es 2.

Ejercicios

(1) Determinar la distancia mínima del código de repetición de longitud 2^r, donde r es cualquier entero positivo.
(2) Determinar el peso de todas las palabras (en particular la distancia mínima) del código de Hamming $\mathscr{H}_3$.

La distancia mínima de un código lineal se puede obtener a partir de su matriz de paridad, como lo indica el siguiente resultado, para su demostración se puede consultar por ejemplo [3] o [4].

PROPOSICIÓN 6.3.4. *Si H es la matriz generadora de un código lineal $\mathscr{C}$, entonces su distancia mínima es el máximo entero d tal que cualesquiera $d-1$ columnas de la matriz H son linealmente independientes.*

Una de las principales propiedades de la distancia mínima de un $[n,k,d]$-código lineal está dada por el siguiente resultado:

TEOREMA 6.3.5. Un $[n,k,d]$-código lineal puede corregir a lo mas $[(d-1)/2]$ errores.

Para una demostración de este resultado sugerimos al lector consultar por ejemplo [3] o [4].

Otro de los resultados interesantes con respecto a la distancia mínima es el siguiente:

TEOREMA 6.3.6. *(La cota Singleton) La distancia mínima d de un $[n,k]$ código lineal $\mathscr{C}$ definido sobre un campo finito satisface la relación:*

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

DEMOSTRACIÓN. El rango de la matriz de paridad H del código es $n-k$ por lo tanto $n-k+1$ columnas de H son linealmente dependientes. por consiguiente el número mínimo de colunas linealmente dependientes, es decir d, debe ser menor o igual a $n-k+1$. □

Como consecuencia de este resultado y sabiendo que el código de Hamming $\mathcal{H}_3$ tiene distancia mínima igual a 3, se sigue que este código puede corregir a lo mas un error. De hecho, todos los código de Hamming (ver õ7.4.1) tienen distancia mínima 3 (dar una demostración !), pero a pesar de esta restricción fueron muy usados en la práctica y tiene muchas propiedades interesantes. Para mayores detalles sugerimos al lector interesado consultar por ejemplo [3],[4],[8].

6.4. Más ejemplos de códigos lineales

A continuación se introducirán otros ejemplos de códigos lineales detectores-correctores de errores, algunos de ellos usados en la práctica. Para describir y conocer las propiedades mas importantes de cada uno de ellos se requiere de más tiempo y algunos otros conceptos de teoría de códigos, álgebra y combinatoria entre otros, lo cual se sale del propósito de las presentes notas. El lector interesado en el estudio mas detallado de estos códigos puede consultar, por ejemplo, las siguientes referencias [3],[4],[8],[11]. Estos ejemplos ayudarán también a introducir algunos otros conceptos de teoría de códigos.

6.4.1. Los códigos binarios de Hamming. La familia de los códigos de Hamming, $\mathcal{H}_r$, es quizá una de las mas famosas y de donde surgieron muchos de los conceptos de la teoría de códigos detectores-correctores de errores. Estos códigos fueron descubiertos en forma independiente por Marcel Golay (1949) y Richard Hamming (1950).

La definición formal de estos códigos es la siguiente:

DEFINICIÓN 6.4.1. El código lineal binario de Hamming, $\mathcal{H}_r$, de longitud $n = 2^r - 1$, $r \geq 2$ está determinado por la matriz de paridad H cuyas columnas son todos los vectores diferentes de cero del espacio vectorial $\mathbb{F}_2^r$.

En este caso se tiene que $\mathcal{H}_r$ es un $[2^r-1, 2^r-1-r, 3]$-código lineal binario.

Por ejemplo, si $r = 3$, una matriz de chequeo de paridad del $[7, 4, 3]$-código lineal (binario) $\mathcal{H}_3$, es:

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

Obsérvese que las columnas de la matriz H son las representaciones binarias de los enteros $1, 2, 3, 4, 5, 6, 7$, respectivamente. Permutando las columnas de la matriz H se obtiene otra que tiene la forma estándard:

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

(las columnas de esta matriz son las representaciones binarias de los números 3,5,6,7,4,2,1, respectivamente).

Esta observación permite dar la siguiente definición:

DEFINICIÓN 6.4.2. Se dice que dos códigos lineales son equivalentes si difieren en el orden de sus coordenadas.

Por lo tanto las matrices H y H' definidas anteriormente, dan códigos lineales equivalentes.

Ejercicios

(1) Probar que las matrices

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

determinan códigos lineales equivalentes

(2) Probar que las matrices

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

determinan códigos lineales equivalentes.

Otra matriz de paridad del código de Hamming $\mathcal{H}_3$ es:

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

es decir, las columnas de H'' son los mismos elementos distintos de cero del espacio lineal $\mathbb{F}_2^3$, pero en otro orden. Obsérvese que en este caso cada renglón de la matriz H'' se puede obtener de uno de ellos haciendo un co-rrimiento hacia la derecha. Por ejemplo, el segundo renglón se obtiene del primero corriendo hacia la derecha un lugar; el tercer renglón se obtiene del segundo corriendo hacia la derecha un lugar, equivalentemente, haciendo un corrimiento hacia la derecha dos lugares en el primer renglón.

Esta propiedad conduce a otra de las clases mas interesantes de códigos lineales detectores-correctores de errores: los *códigos cíclicos*. Por ejemplo los códigos binarios de Hamming y los códigos de Reed-Solomon (usados por ejemplo en los lectores de CD's) son códigos cíclicos. Algúnas ideas y conceptos sobre los códigos cíclicos se presentarán en el Capítulo 8. El lector interesado puede consultar por ejemplo [3],[4],[8],[11].

Ejercicios

(1) Determinar todas las palabras de peso $i = 0, 1, ..., 7$ del código binario de Hamming $\mathcal{H}_3$.
(2) Dar dos matrices de paridad del código de Hamming $\mathcal{H}_4$, de las cuales una de ellas esté en la forma estándard y otra en la cual indique que dicho código es cíclico. Determinar las palabras de peso $i = 0, 1, ..., 15$ de este código y decir cual es su distancia mínima.

6.4.2. Códigos binarios de Reed-Muller. En esta sección se introducirá una de las clases de códigos lineales detectores-correctores de errores mas usados en la práctica, y la cual tiene propiedades interesantes, tanto desde un punto de vista matemático como computacional y de aplicación, en particular la forma de decodificación es sencilla. Por ejemplo el código de Reed-Muller, $R(1, 5)$ de orden 1 fue usado en 1979 por la nave espacial Mariner 9 para trasmitir fotografías en blanco y negro de Marte. En la literatura se pueden encontrar varias formas (equivalentes) de definir estos códigos. Para mayores detalles sugerimos al lector consultar por ejemplo [3],[6],[8].

Sea $\mathbb{F}_2$ el campo de los números binarios y sean $x_1, ..., x_n$ variables. Denotemos por A al conjunto $\mathbb{F}_2[x_1, ..., x_n]$ de polinomios en las indeterminadas $x_1, ..., x_n$ con coeficientes en $\mathbb{F}_2$. Es fácil ver que este conjunto tiene estructura de anillo (conmutativo con identidad) con las operaciones de suma y producto usuales. Sea R el conjunto de elementos de A con la propiedad de que ningúna variable aparece con exponente mayor que 1. En otras palabras, R es el anillo cociente de A módulo el ideal $< x_1^2 - x_1, x_2^2 - x_2, ..., x_n^2 - x_n >$ generado por los monomios $x_1^2 - x_1, x_2^2 - x_2, ..., x_n^2 - x_n$, es decir,

$$R = \mathbb{F}_2[x_1, ..., x_n]/ < x_1^2 - x_1, x_2^2 - x_2, ..., x_n^2 - x_n >$$

Por ejemplo si $n = 3$, $f(x_1, x_2, x_3) = x_1 + x_2 + x_3 + x_1x_3 + x_2x_3$ es un elemento de R, en cambio $g(x_1, x_2, x_3) = x_1^2 + x_1x_2x_3$ no es un elemento de R.

El anillo R se puede escribir de la siguiente manera:

$$R = R_0 + R_1 + \cdots + R_m + \cdots$$

donde R_d denota al conjunto de elementos de A de grado d (el grado de un polinomio se define en la forma usual, como el máximo de los grados de sus monomios, tomando en cuenta que ningúna variable aparece con exponente mayor a 1). Además, cada R_d es un grupo (aditivo conmutativo) y si $f(x_1, ..., x_n) \in R_s$ y $g(x_1, ..., x_n) \in R_t$, entonces el producto $fg \in R_{s+t}$. Obsérvese que cada R_d es casi un $\mathbb{F}_2$-subespacio vectorial: el cero no está en R_d si $d > 0$. Por abuso de notación, al subespacio lineal $< R_d >$ generado por R_d también se le denotará con el mismo símbolo, R_d. Lo anterior quiere decir que R es un anillo graduado con la graduación (natural) $R_0, R_1, \dots R_d, \dots$. Es fácil ver que el $\mathbb{F}_2$-subespacio lineal R_0 tiene dimensión 1, R_1 tiene dimensión n, R_2 tiene dimensión $\binom{n}{2}$, (el número de monomios de grado 2 en las variables $x_1, ..., x_n$). En general el subespacio lineal R_d tiene dimensión $\binom{n}{d}$, (el número de monomios de grado d en las variables $x_1, \dots, x_n$).

Si ρ es un entero tal que $0 \leq \rho \leq n$, entonces $R_{\leq \rho} = R_0 \oplus R_1 \oplus \cdots \oplus R_\rho$ es un $\mathbb{F}_2$-subespacio lineal de dimensión:

$$k = 1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{\rho - 1} + \binom{n}{\rho}$$

Por ejemplo si $n = 4$, R_0 denota a las constantes, es decir, $R_0 = \mathbb{F}_2$, R_1 es el $\mathbb{F}_2$-subespacio lineal de todos los polinomios de grado 1 en las variables $x_1, \dots, x_4$ y tiene dimensión 4; R_2 es el subespacio lineal de todos los polinomios de grado 2 en las variables $x_1, \dots, x_4$, por ejemplo $x_1x_3 + x_2x_4$ es un elemento de R_2; y en general R_d denota al subespacio lineal de polinomios de grado d en esas mismas variables ($d = 3, 4$). Así si $\rho = 3$ entonces $R_{\leq 3} = R_0 \oplus R_1 \oplus R_2 \oplus R_3$

Ahora estamos en condiciones de definir el código de Reed-Muller.

Definición *Sean $n \geq 1$ y ρ enteros tales que $1 \leq \rho \leq n$. El código binario de Reed-Muller $RM(\rho, n)$ de orden ρ en n variables es:*

$$RM(\rho, n) = R_0 \oplus R_1 \oplus \cdots \oplus R_\rho$$

es decir, el código RM(ρ, n) es el conjunto de los polinomios en n variables (donde cada variable aparece con exponente 1) de grado a lo mas ρ.

Las palabras del código RM(ρ, n) se obtienen de la siguiente manera: cada elemento $f(x_1, \dots, x_n) \in RM(\rho, n)$ se puede pensar como una función $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$, es decir, una función *booleana*, donde $\mathbb{F}_2^n$ se puede identificar con el espacio afín de dimensión n sobre el campo de los números binarios. De hecho,

$$R = \mathbb{F}_2[x_1, ..., x_n]/ < x_1^2 - x_1, x_2^2 - x_2, ..., x_n^2 - x_n >$$

es el anillo de coordenadas de este espacio afín.

Como $\mathbb{F}_2^n$ tiene 2^n elementos, la función (booleana) $f(x_1, ..., x_n) \in RM(\rho, n)$ se identifica con el vector de longitud 2^n de sus valores: $f \leftrightarrow (f(\mathbf{v}))_{\mathbf{v}\in\mathbb{F}_2^n}$. Así el código de Reed-Muller $RM(\rho, n)$ tiene longitud 2^n y su dimensión es la dimensión k de $R_{\leq\rho}$ dada anteriormente. Se puede ver que la distancia mínima de este código es $2^{m-\rho}$ por lo tanto, $RM(\rho, n)$ es $[2^n, k, 2^{m-\rho}]$ un código linear (binario).

Este código ha sido objeto de estudio de varios grupos de investiación y de un buen número de generalizaciones, y aún se esta muy lejos de conocer todas sus propiedades. Asimismo, las funciones booleanas, en particular las llamadas funciones *bent*, están muy relacionadas con algunos aspectos de la criptografía de llave secreta (o simétrica). El estudio de estas funciones, sobre todo en las últimas dos décadas, ha sido muy extenso, y así como en el caso de los códigos de Reed-Muller, aún hay muchas cosas por investigar. Los lectores interesados en el estudio de estos códigos y de funciones booleanas, puede consultar por ejemplo [3].

6.4.3. El código Simplex. El código *simplex*, $\mathcal{S}_r$, está íntimamente ligado al código (binario) de Hamming $\mathcal{H}_r$: es su código dual. Para entender un poco mas este código, a continuación se dá la definición del *código dual* asociado a un código lineal.

Recordemos que si $\mathbf{u}, \mathbf{v} \in \mathbb{F}_2^n$, su producto interno se define como:

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + \cdots + u_n v_n$$

donde $\mathbf{u} = (u_1, ..., u_n)$ y $\mathbf{v} = (v_1, ..., v_n)$

Si V es un subconjunto del espacio lineal $\mathbb{F}_2^n$ su dual se define como:

$$V^\perp = \{\mathbf{x} \in \mathbb{F}_2^n \mid \mathbf{v} \cdot \mathbf{x} = 0 \ \forall\ \mathbf{v} \in V\}$$

Como es fácil ver, en general $V^\perp$ no es un subespacio lineal de $\mathbb{F}_2^n$, pero en el caso en que V es un subespacio lineal, $V^\perp$ también lo es (dar una demostración de este hecho).

Ahora estamos en condiciones de dar la definición del código dual.

Definición 4.3.1 *El código dual de un código lineal binario $\mathcal{C}$ es simplemente $\mathcal{C}^\perp$.*

Obsérvese que si $\mathcal{C}$ es un $[n, k, d]_q$-código lineal, entonces $\mathcal{C}^\perp$ es un $[n, n-k, d']_q$-código lineal.

Dado que un código lineal tiene asociada una matriz generadora y de paridad, las correspondientes del código dual están dadas de la siguiente manera:

PROPOSICIÓN 6.4.3. *Si el código lineal binario $\mathscr{C}$ tiene como matriz generadora G y matriz de paridad H, entonces H es la matriz generadora y G la matriz de paridad del código dual $\mathscr{C}^{\perp}$.*

Ahora se puede definir el código simplex $\mathcal{S}_r$.

DEFINICIÓN 6.4.4. El código simplex, $\mathcal{S}_r$, es el código dual del código de Hamming $\mathcal{H}_r$, es decir:

$$\mathcal{S}_r = \mathcal{H}_r^{\perp}$$

Así, como $\mathcal{H}_r$ es un $[2^n - 1, 2^n - n - 1, 3]$-código, el código simplex $\mathcal{S}_r$ es un $[2^n - 1, n, d']$-código lineal binario.

A continuación se dará la matriz generadora de algúnos ejemplos del código simplex.

La matriz generadora del código simplex $\mathcal{S}_2$ es:

$$G_2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

y las palabras del código son:

$$\mathcal{S}_2 = \{(000), (011), (101), (110)\}$$

La matriz generadora del código simplex $\mathcal{S}_3$ es:

$$G_3 = \left(\begin{array}{c|c|c} 000 & 1 & 111 \\ \hline 011 & 0 & 011 \\ 101 & 0 & 101 \end{array}\right)$$

$$= \left(\begin{array}{c|c|c} 000 & 1 & 111 \\ \hline G_2 & 0 & G_2 \end{array}\right)$$

Ejercicios.

(1) Dar todas las palabras del código simplex $\mathcal{S}_3$.
(2) Dar una matriz generadora del código simplex $\mathcal{S}_4$.
(3) Dar una matriz generadora del código simplex $\mathcal{S}_r$.
(4) Determinar la distancia mínima del código simplex.

6.4.4. El código ISBN. Para finalizar esta serie de ejemplos, a continuación se describirá un código, el cual aunque no es lineal sí es detector de errores, y muy usado en la práctica: el código ISBN.

Es bien sabido que actualmente todos los libros tienen un número, el ISBN, (*International Standard Book Number*) el cual normalmente aparece junto con algúna otra información sobre el libro (autores, editorial, etc.), sobre todo en la primera página. El ISBN es un número

de 10 dígitos, de ahí que a este código se le llame *decimal*, el cual es de la siguiente forma:

$$0 - 387 - 97812 - 7$$

El primer dígito de este número indica el idioma en el cual el libro esta escrito, por ejemplo el "0" está asociado al idioma Inglés. El siguiente grupo de dígitos está asociado a la casa editorial, por ejemplo Springer Verlag tiene asociado el número "387", el siguiente grupo de dígitos, "97812", es el número que la casa editorial asigna a sus libros. El último dígito, "7", es un número detector de errores.

Veamos mas de cerca la función de este dígito checador de errores. A los 10 dígitos de un número ISBN, se les puede pensar como un elemento de $\mathbb{Z}_{11}^{10}$, donde $\mathbb{Z}_{11}$ es el campo de los enteros módulo 11. Si $\mathbf{x} = x_1 \cdots x_9$ son los primeros 9 dígitos de un número ISBN, el décimo dígito, x_{10} (el dígito checador), es la solución en $\mathbb{Z}_{11}$ de la ecuación:

$$x_1 + 2x_2 + 3x_3 + \cdots + 9x_9 + 10x_{10} = 0$$

Como $10 = -1$ en $\mathbb{Z}_{11}$, la relación anterior se puede expresar como:

$$x_{10} = x_1 + 2x_2 + 3x_3 + \cdots + 9x_9$$

Es fácil ver que en el ejemplo del número ISBN mencionado anteriormente, el dígito checador es $x_{10} = 7$.

Cuando el dígito checador es igual a 10, la convención es escribir una "X".

Desde el punto de vista de la teoría de códigos, el número ISBN se puede determinar de la siguiente manera:

Sea $\mathscr{C}$ el código lineal de longitud 10 sobre $\mathbb{Z}_{11}$ determinado por la matriz de paridad:

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

Es fácil ver que $\mathscr{C}$ es un $[10, 9, 2]$ código lineal sobre el campo $\mathbb{Z}_{11}$ con 11^9 elementos. Sea $\mathscr{T}$ el código que se obtiene de $\mathscr{C}$ quitando (removiendo) todas aquellas palabras que tengan un "10" en cualquier posición, excepto en la última. Obsérvese que $\mathscr{T}$ es un $(10, 10^9, 2)$-código no-lineal y que los elementos de este nuevo código son precisamente los números ISBN, por lo cual a $\mathscr{T}$ también se le llama el código ISBN.

Veamos ahora que el código ISBN $\mathscr{T}$ puede detectar un sólo error en una palabra de código (un número ISBN).

En efecto, supóngase que una palabra $\mathbf{c}$ (número) ISBN es enviado y que se recibe una palabra $\mathbf{x}$ que contiene un sólo error en la posición i. Así el error es $a\mathbf{e}_i$ para algún $a \in \mathbb{Z}_{11}$ y la palabra recibida es:

$$\mathbf{x} = \mathbf{c} + a\mathbf{e}_i$$

Como $\mathbf{c}$ es una palabra del código $\mathscr{C}$ y tiene la propiedad de que $H \cdot \mathbf{c} = 0$, se sigue que

$$H \cdot \mathbf{x} = H \cdot \mathbf{c} + H \cdot (a\mathbf{e}_i) = ai$$

el cual es un producto de dos elementos distintos de cero del campo $\mathbb{Z}_{11}$, por lo cual es distinto de cero. Por consiguiente, un sólo error es detectado en la palabra recibida $\mathbf{x}$ por el hecho de que $H \cdot \mathbf{x} \neq 0$.

Ejercicio. Verificar el dígito checador de errores de algunos de los libros que tienes.

Capítulo 7

Códigos cíclicos

En este capítulo se introducirá una de las clases mas importantes de códigos lineales, los *códigos cíclicos*, se darán algunas de sus propiedades y ejemplos. Una de las herramientas mas importantes para la descripción de estos códigos es el anillo de polinomios en una indeterminada con coeficientes en un campo finito.

7.1. Algunos ejemplos

En esta sección se presentarán algunos ejemplos sencillos que motiven el concepto de *código cíclico*.

Ejemplo 1. Sea $\mathbb{F}_q$ un campo finito. Dado un entero positivo n, consideremos el código de repetición de longitud n:

$$\mathscr{C} = \{(0, 0, \cdots, 0), (1, 1, \cdots, 1)\}$$

Obsérvese que $\mathscr{C}$ es un código lineal de dimensión 1 y que si la última coordenada de cada palabra de $\mathscr{C}$ se coloca al principio, se obtiene la misma palabra. Es decir, el código es invariante bajo este corrimiento.

Sea $\mathscr{H}_3$ el código lineal binario de Hamming cuya matriz de paridad es la siguiente:

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

Obsérvese que las columnas de esta matriz son los elementos distintos de cero del espacio lineal:

$$\mathbb{F}_2^3 = \{(000), (001), (010), (011), (100), (101), (110), (111)\}$$

y los elementos de este espacio lineal, en notación decimal son los números $\{0, 1, 2, 3, 4, 5, 6, 7\}$ (el bit mas significativo es el de la derecha).

Ejercicio. Determinar todos los elementos del código $\mathscr{H}_3$ asi como sus parámetros.

Ahora considerese el código lineal binario $\mathscr{C}$ cuya matriz de paridad es la siguiente:

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

Ejercicio. Determinar todos los elementos del código $\mathscr{C}$ asi como sus parámetros.

Observemos que las columnas de la matriz $\widetilde{H}$ son las mismas que las de la matriz H, pero en diferente orden. De hecho la siguiente permutación (o su inversa) determina el orden de las columnas de ambas matrices:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 2 & 1 & 6 & 3 & 7 & 5 \end{pmatrix}$$

Ejecicio. Escribir la permutacion σ como producto de cíclos y determinar si es una permutación par o impar.

El análisis anterior permite dar la siguiente definición:

DEFINICIÓN 7.1.1. Dos códigos lineales con matrices de paridad H y H' son *equivalentes* si las columnas de H' se pueden obtener permutando las columnas de la matriz H.

También observemos que los renglones de la matriz $\widetilde{H}$ tienen la siguiente propiedad: el segundo renglón se puede obtener del primero colocando la última entrada del primer renglón al principio de éste, y el tercer renglón se puede obtener del segundo efectuando la misma operación.

Ejercicio. Comprobar que los elementos del código $\mathscr{C}$ con matriz de paridad $\widetilde{H}$ se pueden obtener efectuando la operación descrita anteriormente con los renglones de la matriz $\widetilde{H}$.

En base al ejercicio anterior podemos decir que el código $\mathscr{C}$ tiene la siguiente propiedad:

Si $(a_0, a_1, a_2, a_3, a_4, a_5, a_6)$ *es cualquier elemento de* $\mathscr{C}$*, entonces el vector* $(a_6, a_0, a_1, a_2, a_3, a_4, a_5)$ *también es un elemento de* $\mathscr{C}$*.*

Este es un ejemplo de una clase de códigos lineales muy interesantes por sus propiedades, llamados *cíclicos*. La definición formal de un código cíclico es la siguiente:

DEFINICIÓN 7.1.2. Un $[n, k, d]$ código lineal $\mathscr{C}$ definido sobre un campo finito es *cíclico* si para cada palabra $(a_0, a_1, ..., a_{n-1})$ de $\mathscr{C}$ el vector $(a_{n-1}, a_0, a_1, ..., a_{n-2})$ también es un elemento del código $\mathscr{C}$.

Ejercicio. Sea $\mathscr{C}$ el código lineal (binario) generado por los renglones de la siguiente matriz:

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

es decir, si $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ son los renglones de la matriz G, entonces

$$\mathscr{C} = \{a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3 + a_4\mathbf{v}_4 \ : \ a_i \in \mathbb{F}_2\}$$

Ejercicios.

(1) Determinar los parámetros del código $\mathscr{C}$.
(2) Determinar la distribución de pesos del código $\mathscr{C}$.
(3) Determinar si existe alguna relación entre los códigos $\mathscr{H}_3$ y $\mathscr{C}$.

Si $\mathbb{R}$ denota al campo de los números reales, el producto cartesiano $\mathbb{R}^n$ es un espacio lineal (real) de dimensión n sobre $\mathbb{R}$. En este espacio se tiene el producto interno usual: si $\mathbf{x} = (x_1, x_2, ..., x_n), \mathbf{y} = (y_1, y_2, ..., y_n)$ son elementos de $\mathbb{R}^n$, entonces,

$$\mathbf{x} \cdot \mathbf{y} = x_1y_1 + x_2y_2 + \cdots + x_ny_n$$

con las propiedades usuales (de hecho es una forma bilineal en $\mathbf{R}^n$).

De manera similar al caso real, en el espacio lineal $\mathbb{F}_q^n$ también se tiene un producto interno y su definición es básicamente la misma que para el caso de ${\mathbb{F}_q}^n$: si $\mathbf{u} = (u_1, u_2, ..., u_n), \mathbf{v} = (v_1, v_2, ..., v_n)$ son elementos de $\mathbb{F}_q^n$, entonces,

$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2 + \cdots + u_nv_n$$

Este producto interno tiene las mismas propiedades que en el caso de $\mathbb{R}^n$.

En el caso de $\mathbb{R}^n$, con la ayuda del producto interno, dado un subespacio lineal V de $\mathbb{R}^n$, se puede definir el subespacio ortogonal $V^\perp$ de V. Dado que $\mathbb{F}_q^n$ tiene un producto interno con las mismas propiedades que en el caso de $\mathbb{R}^n$, tiene también el mismo concepto de subespacio ortogonal a uno dado: si U es un subespacio lineal de $\mathbb{F}_q^n$, entonces:

$$U^\perp = \{\mathbf{v} \in \mathbb{F}_q^n \ : \ \mathbf{v} \cdot \mathbf{u} = 0\}$$

Con el concepto de subespacio ortogonal se puede dar la siguiente definición:

DEFINICIÓN 7.1.3. Se dice que un código lineal C es *auto-ortogonal* si $C \subseteq C^\perp$, y el código C es *auto-dual* si $C = C^\perp$.

Ejercicios.

(1) Determinar el código dual de $\mathscr{H}_3$ y de $\mathscr{C}$.

(2) Determinar las matrices generadoras y de paridad de los códigos $\mathscr{H}_3^{\perp}$ y $\mathscr{C}^{\perp}$.
(3) Determinar si los códigos $\mathscr{H}_3$ y $\mathscr{C}$ son auto-ortogonal o auto-dual.

Una forma alternativa para describir y determinar algunas de las propiedades de los códigos cíclicos es por medio de algunos conceptos de Algebra Conmutativa, mas concretamente, de propiedades del anillo de polinomios $\mathbb{F}_q[x]$ y del anillo cociente $\mathbb{F}_q[x]/I$, donde I es un ideal del anillo $\mathbb{F}_q[x]$.

En las siguientes lineas se recordarán algunas de las propiedades de los anillos antes mencionados (el lector interesado puede consultar algún texto de Algebra Conmutativa).

Sea $\mathbb{F}$ un campo (el cual puede ser un campo finito $\mathbb{F}_q$). El conjunto de polinomios, $\mathbb{F}[x]$, donde $bbbf$ es un campo, tiene la estructura de anillo (conmutativo con identidad) con la suma y producto usual de polinomios. Este anillo tiene las siguientes propiedades:

(1) Es un anillo *euclidiano*, es decir, donde es válido el algoritmo de Euclides.
(2) Es un dominio de ideales principales (DIP), es decir, no tiene divisores de cero propios y cualquier ideal es generado por un elemento.

Si $I = \langle f(x)\rangle$ es un ideal de $\mathbb{F}_q[x]$, generado por el polinomio $f(x)$ de grado n, el anillo cociente o de clases residuales módulo $f(x)$, se puede identificar con los polinomios con coeficientes en $\mathbb{F}_q$ de grado a lo mas $n-1$, es decir,

$$\mathscr{R}_n = \mathbb{F}_q[x]/\langle f(x)\rangle = \{a_0 + a_1 x + \cdots + a_{n-1}x^{n-1},\ a_i \in \mathbb{F}_q\}$$

Este anillo tiene las siguientes propiedades:

(1) $\mathscr{R}_n$ es un $\mathbb{F}_q$-subespacio lineal de dimensión n. Una base natural está dada por $\{1, x, x^2, ..., x^{n-1}\}$.
(2) $\mathscr{R}_n$ es también un anillo de ideales principales.
(3) Los ideales de $\mathscr{R}_n$ estan en correspondencia biyectiva con los ideales del anillo $\mathbb{F}_q[x]$ que contienen al ideal $I = \langle f(x)\rangle$.

Como una consecuencia de esta última propiedad se tiene la siguiente observación:

Si $\bar{I} = \overline{g(x)}$ es un ideal de $\mathscr{R}_n$ entonces cualquier representante $h(x)$ de la clase $\overline{g(x)}$, es decir, cualquier polinomio $h(x) \in \mathbb{F}_q[x]$ congruente con $g(x)$ módulo $f(x)$, $h(x) \equiv g(x) \bmod f(x)$, es tal que $h(x)$ divide a $f(x)$ (en el anillo $\mathbb{F}_q[x]$), es decir,

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

para algún polinomio $q(x) \in \mathbb{F}_q[x]$ (el cociente de dividir a $f(x)$ por $h(x)$).

Con la ayuda del anillo cociente descrita anteriormente se puede dar la interpretación polinomial de los elementos del espacio lineal $\mathbb{F}_q^n$, de la siguiente manera:

PROPOSICIÓN 7.1.4. *Sea n un entero positivo y $\mathcal{R}_n = \mathbb{F}_q[x]/\langle x^n - 1\rangle$, entonces la función*

$$\begin{array}{ccc} \phi : \mathbb{F}_q^n & \longrightarrow & \mathcal{R}_n \\ \mathbf{a} = (a_0, a_1, ..., a_{n-1}) & \longrightarrow & a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \end{array}$$

es un isomorfismo de espacios vectoriales.

Ejercicio. Dar una demostración de la proposición anterior.

Como se puede ver del resultado anterior, a nivel de espacio lineal, lo que se puede hacer en $\mathbb{F}_q^n$ también se puede hacer en $\mathcal{R}_n$. Pero recordemos que $\mathcal{R}_n$ tiene ademas la estructura algebraica de anillo, y esta estructura adicional de $\mathcal{R}_n$ es la que se usará para dar una descripción alternativa de los códigos cíclicos.

TEOREMA 7.1.5. Sea $\mathcal{C} \subseteq \mathbb{F}_q^n$ un código lineal. Entonces $\mathcal{C}$ es cíclico sí y sólo si $\phi(\mathcal{C})$ es un ideal del anillo $\mathcal{R}_n$.

DEMOSTRACIÓN. Veamos primero que si $\mathcal{C}$ es un código cíclico, entonces $\phi(\mathcal{C})$ es un ideal del anillo $\mathcal{R}_n$. Si $\mathbf{a} = (a_0, a_1, ..., a_{n-1}) \in \mathcal{C}$ entonces $\phi(a) = a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \phi(\mathcal{C})$. Como $\mathcal{C}$ es cíclico, $\widetilde{\mathbf{a}} = (a_{n-1}, a_0, ..., a_{n-2}) \in \mathcal{C}$ y la imagen de esta palabra bajo la función ϕ es:

$$\widetilde{\mathbf{a}}(x) = a_{n-1} + a_0 x + a_1 x^2 + \cdots + a_{n-2}x^{n-x}$$

dado que $\widetilde{\mathbf{a}}(x) = xa(x)$ y $a(x) \in \phi(\mathcal{C})$ se sigue que $\widetilde{\mathbf{a}}(x) \in \phi(\mathcal{C})$. De manera similar se puede ver que $x^k a(x) \in \phi(\mathcal{C})$ para $k \geq 0$ entero, y como la suma y el producto de $\mathcal{R}_n$ son asociativos, se concluye que si $h(x)$ es cualquier elemento de $\mathcal{R}_n$, entonces $h(x)a(x) \in \phi(\mathcal{C})$ de lo cual se sigue que $\phi(\mathcal{C})$ es un ideal de $\mathcal{R}_n$.

Ahora veamos que si $\phi(\mathcal{C})$ es un ideal de $\mathcal{R}_n$, entonces el código lineal $\mathcal{C}$ es cíclico. Sea $\mathbf{a} = (a_0, a_1, ..., a_{n-1}) \in \mathcal{C}$ y $a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \phi(\mathcal{C})$ el elemento correspondiente de $\mathcal{R}_n$. Como $\phi(\mathcal{C})$ es un ideal se tiene en particular que $xa(x) = a_{n-1} + a_0 x + a_1 x^2 + \cdots + a_{n-2}x^{n-2} \in \phi(\mathcal{C})$. Pero la palabra de $\mathcal{C}$ correspondiente a $xa(x)$ es $(a_{n-1}, a_0, ..., a_{n-2})$ y dado que $xa(x) \in \phi(\mathcal{C})$ se sigue que $(a_{n-1}, a_0, ..., a_{n-2}) \in \mathcal{C}$, probando que $\mathcal{C}$ es un código cíclico. □

Con la ayuda de estos resultados, veamos ahora algunos ejemplos de códigos cíclicos.

De acuerdo a las observaciones anteriores, los códigos binarios de longitud n son precisamente los ideales (principales) del anillo $\mathcal{R}_n = \mathbb{F}_q^n[x]/\langle x^n - 1\rangle$. Si $I = \langle g(x)\rangle$ es un ideal de $\mathcal{R}_n$ entonces un representante de la clase de $g(x)$ módulo $f(x)$, digamos $g(x)$ divide al polinomio $x^n - 1$ en el anillo $\mathbb{F}_q[x]$. Por lo tanto si conocemos la descomposición de $x^n - 1$ como producto de factores irreducibles, se tienen todos los factores de $x^n - 1$.

Ejemplo 1. Códigos cíclicos binarios de longitud 3. De acuerdo a la observación anterior, veamos cual es la descomposición de $x^3 - 1$ como producto de factores irreducibles en $\mathbb{F}_2[x]$.

Observemos que $x^3 - 1 = (x - 1)(x^2 + a_1x + a_2)$ con $a_1, a_2 \in \mathbb{F}_2[x]$. ?'Qué posibilidades tenemos para el polinomio $q(x) = x^2 + a_1x + a_2$?, es decir, ?'Cuántos polinomios mónicos de grado 2 con coeficientes en $\mathbb{F}_2$ existen? Dado que los coeficientes a_0, a_1 pueden tomar solo dos valores, se tienen los siguientes polinomios:

$$x^2, \quad x^2 + 1, \quad x^2 + x, \quad x^2 + x + 1$$

¿Cuáles de estos polinomios son irreducibles sobre $\mathbb{F}_2$? Es fácil ver que los primeros tres no son irreducibles pero el tercero sí lo es. Por lo tanto la descomposición del polinomio $x^3 - 1$ como producto de irreducibles es:

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

Por consiguiente los factores de $x^3 - 1$ son: $1, x - 1, x^2 + x + 1, x^3 - 1$ y los códigos cíclicos binarios de longitud 3 son:

(1) $\mathcal{C}_0 = \langle 1\rangle$
(2) $\mathcal{C}_1 = \langle x - 1\rangle$
(3) $\mathcal{C}_2 = \langle x^2 + x + 1\rangle$
(4) $\mathcal{C}_3 = \langle x^3 - 1\rangle$

En general, a los códigos generados por los factores triviales 1 y $x^n - 1$ de $x^n - 1$ se les llama códigos triviales y para ciertos propósitos no se toman en cuenta, solo en ocasiones para cuestiones de conteo. Así podemos decir que hay 4 códigos cíclicos binarios de longitud 3 de los cuales dos de ellos son triviales.

Ahora surgen varias preguntas naturales:

(1) ¿Cuál es la matriz generadora de cada uno de esos códigos?
(2) ¿Cuáles son sus parámetros?
(3) ¿Existirá alguna relación entre el polinomio generador de un código cíclico y su matriz generadora?
(4) ¿El código dual de un cíclico es cíclico? ?'Cuál es su polinomio generador?

Ejercicios.

(1) Dar una respuesta a las preguntas anteriores.
(2) Determinar todos los códigos binarios de longitud 4,5,6.

Ejemplo 2. Códigos cíclicos binarios de longitud 7.

Como se vió en la sección anterior, un código cíclico de longitud n sobre un campo finito $\mathbb{F}_q$ es equivalente a un ideal en el anillo de polinomios $\mathbb{F}_q[x]/\langle x^n - 1\rangle$ y como este anillo es de ideales principales el ideal esta generado por un polinomio el cual es un divisor en $\mathbb{F}[x]$ de $x^n - 1$, el cual se conoce como el polinomio *generador* del código cíclico. En esta sección se obtendrán algunas propiedades del código a partir de su polinomio generador, en particular se describirá la matriz generadora y de paridad del código asi como su dimensión.

7.2. Polinomio y matriz generadora

Comenzaremos por decir cual es la dimensión y la matriz generadora de un código cíclico.

TEOREMA 7.2.1. *Sea $\mathscr{C}$ un código cíclico de longitud n sobre el campo $\mathbb{F}_q$ y sea $\mathscr{I}$ el ideal del anillo $\mathscr{R}_n = \mathbb{F}_q[x]/\langle x^n - 1\rangle$ correspondiente a $\mathscr{C}$ con polinomio generador $g(x)$ de grado r. Entonces:*

(1) *La dimensión de $\mathscr{C}$ es $n - r$.*
(2) *Si $g(x) = g_0 + g_1 x + \cdots + g_r x^r$, una matriz generadora de $\mathscr{C}$ es:*

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_r & \cdots & 0 \\ & g_0 & g_1 & \cdots & g_{r-1} & g_r & \cdots \\ & & & \cdots & & & \\ & & g_0 & \cdots & \cdots & \cdots & g_r \end{pmatrix}$$

$$= \begin{pmatrix} g(x) & & & & \\ & xg(x) & & & \\ & & & \cdots & \\ & & & & x^{n-r-1}g(x) \end{pmatrix}$$

donde en los espacios en blanco aparece un cero.

DEMOSTRACIÓN. Si $c(x) \in \mathscr{I}$, $\mathrm{gr}(c(x)) < n$, entonces $c(x) = q(x)g(x)$ en $\mathscr{R}_n$ y por lo tanto:

$$\begin{aligned} c(x) &= q(x)g(x) + a(x)(x^n - 1) \text{ en } \mathbb{F}_q[x] \\ &= (q(x) + a(x)h(x))g(x) \text{ en } \mathbb{F}_q[x] \\ &= f(x)g(x) \text{ en } \mathbb{F}_q[x]. \end{aligned}$$

donde $\mathrm{gr}(f(x)) \leq n - r - 1$. Por lo tanto el código (ideal) consiste de los múltiplos del polinomio generador por polinomios de $\mathbb{F}_q[x]$ (no de $\mathscr{R}_n$) de grado $\leq n - r - 1$. Los polinomios $g(x), xg(x), ..., x^{n-r-1}g(x)$, un total de $n-r$ y múltiplos de $g(x)$, son linealmente independientes sobre

$\mathbb{F}_q$. Por consiguiente el código tiene dimensión $n - r$ y los vectores correspondientes son las hileras de la matriz generadora G. □

7.3. Polinomio y matriz de paridad

Veamos ahora como se obtiene el polinomio de chequeo de paridad y la matriz de paridad de un código cíclico.

Sea $\mathscr{C} = \langle g(x) \rangle$ un código cíclico de longitud n sobre el campo $\mathbb{F}_q$ con polinomio generador $g(x)$. Como se vió antes, $g(x)$ divide a $x^n - 1$ en $\mathbb{F}_q[x]$. Así

$$h(x) = (x^n - 1)/g(x) = h_0 + h_1 x + \cdots + h_t x^t, (h_t \neq 0).$$

se llama el *polinomio de paridad* del código $\mathscr{C}$. Este polinomio tiene la siguiente propiedad (de ahí el nombre):

Si $c(x) = c_0 + c_1 x + \cdots c_{n-1} x^{n-1} = f(x)g(x)$ es un elemento de $\mathscr{C}$, entonces:

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

lo cual implica que $c(x)h(x) = 0$ en el anillo $\mathscr{R}_n$.

Efectuando el producto de la relación anterior, los coeficientes de los polinomios satisfacen las siguientes relaciones (en $\mathscr{R}_n$):

$$\sum_{i=0}^{n-1} c_i h_{j-i} = 0, \;\; j = 0, 1, ..., n-1$$

(los sub-índices se toman módulo n). Las relaciones anteriores son llamadas de *paridad*. Si

$$H = \begin{pmatrix} & & h_k & \cdots & h_2 & h_1 & h_0 \\ & h_k & \cdots & h_2 & h_1 & h_0 & \\ & & & \cdots & & & \\ h_k & \cdots & h_2 & h_1 & h_0 & & \end{pmatrix}$$

$$= \begin{pmatrix} & & & & & & h(x) \\ & & & & & xh(x) & \\ & & & \cdots & & & \\ x^{n-k-1}h(x) & & & & & & \end{pmatrix},$$

de las relaciones anteriores se tiene que si $c \in \mathscr{C}$ entonces $Hc^t = 0$. Como

$$k = \mathrm{gr}(h(x)) = n - \mathrm{gr}(g(x)) = dim_{\mathbb{F}_q} \mathscr{C}$$

y los renglones de H son linealmente independientes se tiene que $Hc^t = 0$ implica $c \in \mathscr{C}$.

7.4. El código dual

Si $\mathscr{C}$ es un código cíclico, el siguiente resultado muestra que el código dual también es cíclico.

TEOREMA 7.4.1. *Sea $\mathscr{C}$ un código cíclico con polinomio generador $g(x)$. Entonces el código dual $\mathscr{C}^{\perp}$ es cíclico con polinomio generador*

$$g^{\perp}(x) = x^{\mathrm{gr}(h(x))}h(x^{-1})$$

donde $x^n - 1 = g(x)h(x)$.

DEMOSTRACIÓN. La demostración se sigue de la expresión para la matriz de paridad H mencionada anteriorente. □

A $g^{\perp}(x)/h(0)$ se le conoce como el polinomio *recíproco* de $h(x)$.

7.5. Algunos ejemplos

En esta sección se darán algunos ejemplos de códigos cíclicos, su dual y las matrices generdoras y de paridad.

Recordemos que las columnas de la matriz de paridad del código binario de Hamming $\mathscr{H}_m$ son los elementos no-cero del espacio lineal $\mathbb{F}_2^m$, siendo un total de $2^m - 1$ elementos. Si α es un elemento primitivo del campo $\mathbb{F}_{2^m}$, los elementos no-cero de este campo son $\{1, \alpha, \alpha^2, ..., \alpha^{2^m-2}\}$, los cuales estan en correspondencia biyectiva con las columnas de la matriz H. Por consiguiente la matriz H se puede identificar con la matriz

$$H = (1, \alpha, \alpha^2, ..., \alpha^{2^m-2})$$

donde cada entrada se reemplaza por la columna correspondiente de la matriz de paridad de $\mathscr{H}_m$. Por ejemplo, para el código $\mathscr{H}_3$ se tiene que:

$$H = (1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6) = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

donde α es un elemento primitivo del campo $\mathbb{F}_{2^3}$ y satisface la relación $\alpha^3 + \alpha + 1 = 0$.

Por consiguiente, un elemento $c = (c_0, c_1, ..., c_{n-1}) \in \mathscr{H}_m$ sí y sólo si $Hc^t = 0$ sí y sólo si $c_0 + c_1\alpha + \cdots + c_{n-1}\alpha^{n-1} = 0$ sí y sólo si $c(\alpha) = 0$, donde $c(x) = c_0 + c_1x + \cdots c_{n-1}x^{n-1}$ es el polinomio asociado a la palabra c.

El código de Hamming $\mathscr{H}_3$ generado por $g(x) = x^3 + x + 1$ tiene como polinomio de paridad: $h(x) = (x^7 - 1)/(x^3 + x + 1) = (x + 1)(x^3 + x^2 + 1) = x^4 + x^2 + x + 1$ y por lo tanto la matriz de paridad correspondiente es:

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

7.6. El código de Reed-Solomon

Uno de los códigos lineales detectores-correctores de error mas usados en la práctica (CD's, DVD's, comunicación inhalambrica, etc.) es el código (binario) de *Reed-Solomon.* En esta sección se recordará su definición y algunas propiedades básicas. Para mayores detalles el lector puede consultar, por ejemplo los libros de MacWilliams-Sloane o Huffman-Pless.

En la literatura existen varias definiciones (equivalentes) del código de Reed-Solomon. En las siguientes líneas se describirá éste usando una definición que permite definir otros códigos (pero que no se hará aquí por razones de espacio).

Sea $\mathbb{F}_q$ un campo finito con $q = p^r$ elementos y sea $\Gamma = \mathbb{F}_q \setminus 0$. Sea k un entero tal que $0 \leq k \leq n = q-1$ y $\mathcal{R}_k = \mathbb{F}_q[x]/\langle x^k - 1\rangle$ el anillo de las clases residuales de polinomios con coeficientes en $\mathbb{F}_q$ módulo el ideal generado por el polinomio $x^k - 1$. Este anillo se puede identificar con el siguiente conjunto:

$$\mathcal{R}_k = \{f(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}, a_i \in \mathbb{F}_q\}$$

Considerese la función evaluación sobre el conjunto Γ:

$$ev_\Gamma : \mathcal{R}_k \longrightarrow \mathbb{F}_q^n,\ ev_\Gamma(f(x)) = (f(\gamma))_{\gamma\in\Gamma}.$$

Es fácil ver que esta función es $\mathbb{F}_q$-lineal y por consiguiente su imagen es un subespacio lineal sobre este campo. El código de *Reed-Solomon* sobre $\mathbb{F}_q$ se define como:

$$RS_q(k) = Im(ev_\Gamma).$$

Si $\alpha \in \mathbb{F}_q$ es un elemento primitivo entonces $\Gamma = \{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\} = \mathbb{F}_q \setminus \{0\}$, la función evaluación se puede escribir como:

$$ev_\Gamma(f(x)) = (f(1), f(\alpha), f(\alpha^2), \ldots, f(\alpha^{q-2})).$$

Los parámetros de este código estan dados en el siguiente

TEOREMA 7.6.1. *Con la notación anterior $RS_q(k)$ es un $[q-1, k, n-k+1]$ código lineal sobre $\mathbb{F}_q$. Mas aún $RS_q(k)$ es cíclico.*

DEMOSTRACIÓN. Es obvio que la longitud del código es $q-1$. Como la función evaluación es $\mathbb{F}_q$-lineal e inyectiva y $\mathcal{R}_k$ es de dimensión k, el código tiene dimensión k. Como cualquier polinomio de grado $< k$ puede tener a lo mas $k-1$ raices, se sigue que la distancia mínima d es tal que $d \geq n-k+1$. De esta relación y la cota Singleton se concluye que $d = n-k+1$. □

Este es un ejemplo de la clase de códigos llamados MDS (Maximum Distance Separable), es decir, donde la longitud n, dimensión k y

distancia mínima d del código satisfacen la relación:

$$d = n - k + 1.$$

A continuación se dará un ejemplo del código de Reed-Solomon.

Sea $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$, donde $\alpha^2 = \alpha + 1$, el campo con 4 elementos y sea $\mathcal{R}_3 = \{a_0 + a_1 x + a_2 x^2 : \ a_i \in \mathbb{F}_4\}$. Es fácil ver que este espacio vectorial tiene como base $B = \{1, x, x^2\}$, así que tiene dimensión $k = 3$ y por lo tanto tiene $4^3 = 64$ elementos. Como $RS_4(3) = Im(ev_\Gamma)$ y la función ev_Γ es inyectiva, para tener una matriz generadora de este código basta ver cual es la imagen de B bajo este mapeo. Efectuando los cálculos se tiene que una matriz generadora del código $RS_4(3)$ es:

$$G = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \end{pmatrix}.$$

En este ejemplo se puede ver que el tercer renglón de la matriz se obtiene del segundo haciendo un corrimiento y por lo tanto que el código es cíclico.

Ejercicio. Determinar la distribución de pesos del código $RS_4(3)$. En particular dar (al menos) una palabra cuyo peso de Hamming sea igual a la distancia mínima del código.

Capítulo 8

GAP

GAP (Group, Algorithm, Programming) es un sistema libre de restricciones de uso que contiene una librería de funciones la cual implementa operaciones algebraicas, entre otros temas. Es usado para el estudio de grupos, anillos, espacios vectoriales, algebras, etc. El usuario puede fácilmente investigar, modificar y extender las funciones de la librería para su uso especial. Además cuenta con paquetes que implementan funciones para el estudio de áreas específicas como teoría de códigos, factorización de enteros, algebras, etc. Se puede descargar de la página http://www.gap-system.org/, en la cual se encuentran manuales, así como mayor información sobre este sistema y los paquetes con los que cuenta. En este capítulo mostramos algunos ejemplos de campos finitos y códigos utilizando GAP y el paquete GUAVA que contiene implementación de funciones para la construcción y análisis de códigos detectores-correctores de error.

8.0.1. Operaciones básicas. Para familiarizarse con GAP primero veremos algunas operaciones básicas. Al iniciar una sesión en GAP aparece un promp (gap>), después del cual podemos dar instrucciones al sistema. Al final de cada instrucción se escribe un punto y coma (;). Para terminar una sesión se utiliza la instrucción *quit;*.

Los operadores aritméticos son: +, *, –, /, mod, ^
Ejemplo:

```
gap> 3+4;
7
gap> 5*9;
45
gap> 4/16;
1/4
gap> Int(7/3);
2
gap> 7 mod 3;
1
gap> 5^9;
1953125
gap> 9.75;
Syntax error: ; expected
9.75;
```

```
75
```

Para cada fracción GAP calcula su equivalente irreducible y no acepta números con punto decimal.

Operadores lógicos: *and, or, not* . Estos operadores actúan sobre valores booleanos (true o false). Por ejemplo

```
gap> true and false;
false
gap>not(true);
false
gap> true or false;
true
```

Operadores de comparación: =, <>, <, <=, >, >=. Al emplear estos operadores obtenemos como resultado un valor booleano.

```
gap> 9>17;
false
gap> 13=13;
true
```

Las variables en GAP se definen como una sucesión de letras y dígitos que tienen asignado un valor. Para asignar un valor a una variable se utiliza el operador :=. Por ejemplo:

```
gap> x:= 29; y:= 31;
29
31
gap> x^2+y^2;
1802
```

Se pueden dar varias instrucciones en una misma línea las cuales deben separarse con un punto y coma (;), el resultado de cada instrucción aparecerá en el mismo orden en que fueron introducidas.

Una *lista* es una colección de objetos separados por comas y encerrados entre paréntesis rectangulares ([...]). Los elementos de una lista pueden ser de tipos diferentes o del mismo tipo:

```
gap> l:= [35,98,76,7];
[35,98,76,7]
gap> l1:= ["hola",5,32,'r'];
["hola",5,32,'r']
```

Para agregar un elemento al final de una lista utilizamos la instrucción *Add*, por ejemplo:

```
gap> Add(l,9);
[35,98,76,7,9]
```

Por último, podemos crear funciones, escritas en lenguaje de GAP, que devuelven algún resultado. GAP incluye funciones que pueden ser de nuestra utilidad, como:

```
gap> Factorial(5);
120
gap> Print("HOLA\n");
HOLA
gap> IsPrime(15687031);
true
```

Para definir una función sencilla la sintaxis es: nombre := entrada -> salida. Por ejemplo:

```
gap> cuad:=x -> x^2;
function( x ) ... end
gap> cuad(150);
22500
```

8.0.2. Códigos. El paquete GUAVA de GAP cuenta con funciones útiles para la creación de códigos lineales y cíclicos, para construir un nuevo código a partir de otros dos, y para obtener los parámetros importantes de los códigos, sin embargo el usuario puede implementar sus propias funciones.

Comenzaremos con algunos ejemplos de funciones que permiten obtener códigos lineales sobre el campo de los números binarios a partir de la matriz generadora o de la matriz de chequeo de paridad en su forma estándard y algunas otras que permiten obtener el peso de una palabra del código, calcular la distancia entre dos palabras y obtener la distribución de pesos de un código lineal binario y su dual.

Las siguientes funciones permiten obtener los elementos de un código binario a partir de su matriz de chequeo de paridad o su matriz generadora en su forma estándard.

```
gap> InputLogTo("ElementsCode.g");
gap> ElementsCodeG:=function(G)
> local c,i,j;
> c:=[0*G[1]];
> for i in [1..Length(G)] do
>   for j in [(i+1)..Length(c)] do
>       if ((c[i]+c[j]) mod 2) in c then;
>       else Add(c,((c[i]+c[j]) mod 2));
>       fi;
>   od;
> od;
> return c;;
> end;
function( G ) ... end
gap> ElementsCodeH:=function(H)
> local I,S,C,i;
> I:=IdentityMat(Length(H[1]));
> S:=SourceStrings(I);;
> C:=[];
> for i in [1..Length(S)] do
>   if 1 in Syndrome(H,S[i]) then;
>   else Add(C,S[i]);
>   fi;
> od;
> return C;
> end;
function( H ) ... end
gap> InputLogTo();
```

Estas funciones son almacenadas en el archivo ElementsCode.g, para leer el archivo y poder utilizar las funciones en Gap debemos utilizar la instrucción `Read`(“ElementsCode.g"). Por ejemplo:

```
gap> Read("ElementsCode.g");
InputLogTo: can not close the logfile at
CLOSE_INPUT_LOG_TO();
gap> G:=[[1,0,0,1,1],[0,1,0,0,1],[0,0,1,1,0]];;
gap> PrintArray(G);
[ [  1,  0,  0,  1,  1 ],
  [  0,  1,  0,  0,  1 ],
  [  0,  0,  1,  1,  0 ] ]
gap> C:=ElementsCodeG(G);;
gap> PrintArray(C);
[ [  0,  0,  0,  0,  0 ],
  [  1,  0,  0,  1,  1 ],
  [  0,  1,  0,  0,  1 ],
  [  0,  0,  1,  1,  0 ],
  [  1,  1,  0,  1,  0 ],
  [  1,  0,  1,  0,  1 ],
  [  0,  1,  1,  1,  1 ],
  [  1,  1,  1,  0,  0 ] ]
gap> H:=[[1,0,1,1,0],[0,1,0,0,1]];;
gap> PrintArray(H);
[ [  1,  0,  1,  1,  0 ],
  [  0,  1,  0,  0,  1 ] ]
gap> C1:=ElementsCodeH(H);;
PrintArray(C1);
[ [  0,  0,  0,  0,  0 ],
  [  1,  0,  1,  0,  0 ],
  [  1,  0,  0,  1,  0 ],
  [  0,  1,  0,  0,  1 ],
  [  0,  0,  1,  1,  0 ],
  [  1,  1,  1,  0,  1 ],
  [  1,  1,  0,  1,  1 ],
  [  0,  1,  1,  1,  1 ] ]
```

Ejercicios.

(1) Comprobar que los elementos de los códigos C y C1 son correctos.

(2) Obtener los elementos del código dual de C y C1, respectivamente.

Si tenemos la matriz generadora en su forma estándard podemos obtener la matriz de chequeo de paridad y viceversa. Las siguientes funciones permiten hacer ésto. Recuérdese que al hacer la implementación de una función en GAP se debe guardar en un archivo y para poder utilizar las funciones hay que leer el archivo.

```
ParityCheckMatrix:= function(G)
local H,I,M,i,j,k;
H:= [];
M:= [];
for i in [1..Length(G)] do
    for j in [(Length(G)+1)..Length(G[i])] do
        if j=(Length(G)+1) then M[i]:= [G[i][j]];
        else Add(M[i],G[i][j]);
        fi;
    od;
od;
M:= (-1*TransposedMat(M)) mod 2;
I:= IdentityMat(Length(M));
for k in [1..Length(M)] do
    H[k]:= Concatenation(M[k],I[k]);
od;
```

```
return H;
end;


GeneratorMatrix:= function(H)
local G,I,M,i,j,k;
G:= [];
M:= [];
for i in [1..Length(H)] do
    for j in [(Length(G)+1)..(Length(H[1])-Length(H))] do
        if j=(Length(G)+1) then M[i]:= [H[i][j]];
        else Add(M[i],H[i][j]);
        fi;
    od;
od;
M:= (-1*TransposedMat(M)) mod 2;
I:= IdentityMat(Length(H[1])-Length(H));
for k in [1..Length(I)] do
    G[k]:= Concatenation(I[k],M[k]);
od;
return G;
end;
```

Por ejemplo:

```
gap> H:= ParityCheckMatrix(G);;
gap> PrintArray(H);
[ [  1,  0,  1,  1,  0 ],
  [  1,  1,  0,  0,  1 ] ]
gap> LogTo();
```

Teniendo la matriz de chequeo de paridad y la matriz generadora de un código lineal $\mathcal{C}$ es fácil obtener los elementos del código dual $\mathcal{C}^{\perp}$. Ahora nos interesa obtener la distribución de pesos de estos códigos, para lo cual necesitamos las siguientes funciones:

```
TableWeight:= function(C)
local i,j,w,n;
w:= []; for i in [1..Length(C)] do
    n:= 0;
    for j in [1..Length(C[i])] do
        if C[i][j] <> 0 then
            n:= n+1;
        fi;
    od;
    w[i]:= n;
od;
return w;;
end;



Weight:= function(C)
local W;
W:= TableWeight(C);;
Unbind(W[1]);
return Minimum(W);
end;

WeightDistribution:= function(C)
local W,A,n,i,j,k;
A:=[];
W:= TableWeight(C);;
for i in [0..Length(C[1])] do
```

```
    n:= 0;
    for j in [1..Length(W)] do
        if i=W[j] then n:=n+1;
        fi;
    od;
    Add(A,n);
od;
return A;
end;

WeightTable:= function(C,CD)
local A,A1,i;
A:= WeightDistribution(C);;
A1:= WeightDistribution(CD);;
Print("\n\t\t\tW |C   | CD\n"); Print("\t\t\t------------------\n");
for i in [1..Length(A)] do
    if (i-1) < 10 and A[i] < 10 then
        Print("\t\t\t",i-1,"   |   ",A[i],"   |   ",A1[i],"\n");
    elif (i-1) < 10 and A[i] > 10 and A[i] < 100 then
        Print("\t\t\t",i-1,"   |   ",A[i],"  |   ",A1[i],"\n");
    elif (i-1) < 10 and A[i] >= 100 then
        Print("\t\t\t",i-1,"   |   ",A[i]," |   ",A1[i],"\n");
    elif (i-1) >=10 and A[i] >= 100 then
        Print("\t\t\t",i-1,"  |   ",A[i]," |   ",A1[i],"\n");
    elif (i-1) > 10 and A[i] < 10 then
        Print("\t\t\t",i-1,"  |   ",A[i],"   |   ",A1[i],"\n");
    elif (i-1) > 10 and A[i] > 10 and A[i] < 100 then
        Print("\t\t\t",i-1,"  |   ",A[i],"  |   ",A1[i],"\n");
    elif (i-1) > 10 and A[i] >= 100 then
        Print("\t\t\t",i-1,"  |   ",A[i]," |   ",A1[i],"\n");
    else Print("falta");
    fi;
od;
end;
```

Tomando el mismo código de los ejemplos anteriores tenemos lo siguiente:

```
gap> PrintArray(H);
[ [  1,  0,  1,  1,  0 ],
  [  1,  1,  0,  0,  1 ] ]

gap> CD:=ElementsCodeG(H);;
gap> PrintArray(CD);
[ [  0,  0,  0,  0,  0 ],
  [  1,  0,  1,  1,  0 ],
  [  1,  1,  0,  0,  1 ],
  [  0,  1,  1,  1,  1 ] ]

gap> WeightTable(C,CD);

            W   |   C   |   CD
            ------------------
            0   |   1   |   1
            1   |   0   |   0
            2   |   2   |   0
            3   |   4   |   2
            4   |   1   |   1
            5   |   0   |   0
```

Se obtiene el código dual y obtenemos la tabla de distribución de los pesos, en donde la columna `W` representa el peso de las palabras, la

columna C contiene el número de palabras del código con peso i, y la columna CD contiene el número de palabras del código dual con peso i. A partir de ésto podemos obtener el polinomio enumerador de pesos, tanto del código $\mathcal{C}$ como de su dual $\mathcal{C}^{\perp}$.

Otros ejemplos de funciones útiles para códigos lineales son las siguientes:

```
Distance:= function(c,d)
local i,x,n;
n:=0;
x:=(c-d) mod 2;
for i in [1..Length(x)] do
    if x[i] <> 0 then n:= n+1;
    fi;
od;
return n;
end;

Syndrome:= function(H,codeword)
return (H*codeword) mod 2;
end;
```

La primer función permite obtener la distancia entre dos palabras del código (*métrica de Hamming*) y, una vez calculada la matriz de chequeo de paridad, la segunda función ayuda a obtener el sindrome de una palabra del código.

Ahora bien, utilizaremos el paquete GUAVA para el estudio y construcción de códigos lineales y cíclicos sobre cualquier campo finito. Para poder trabajar con este paquete es necesario cargarlo desde el sistema GAP:

```
gap> LoadPackage("guava","2.1");
```

Con ésto ya podemos emplear las funciones contenidas en GUAVA. Empezaremos por dar un ejemplo de la construcción de un código cíclico de longitud n a partir de su polinomio generador. Para ésto debemos definir el campo finito sobre el cual se construirá el código, declarar la indeterminada con la cual se representará el polinomio generador y la longitud del código.

```
gap> F:=GF(2);;
gap> x:=Indeterminate(F,"x");;
gap> p:=RandomPrimitivePolynomial(F,3);
x^3+x^2+Z(2)^0
gap> n:= 7;;
gap> C:= GeneratorPolCode(p,n,F);
a cyclic [7,4,1..3]1 code defined by generator polynomial over GF(2)
```

En este caso su polinomio irreducible sobre el campo $GF(2)$ es x^3+x^2+1, generador de un [7,4,3]-código cíclico, cuyos elementos son:

```
gap> Elements(C);
[ [ 0 0 0 0 0 0 0 ], [ 0 0 0 1 0 1 1 ], [ 0 0 1 0 1 1 0 ], [ 0 0 1 1 1 0 1 ],
  [ 0 1 0 0 1 1 1 ], [ 0 1 0 1 1 0 0 ], [ 0 1 1 0 0 0 1 ], [ 0 1 1 1 0 1 0 ],
  [ 1 0 0 0 1 0 1 ], [ 1 0 0 1 1 1 0 ], [ 1 0 1 0 0 1 1 ], [ 1 0 1 1 0 0 0 ],
```

```
  [ 1 1 0 0 0 1 0 ], [ 1 1 0 1 0 0 1 ], [ 1 1 1 0 1 0 0 ], [ 1 1 1 1 1 1 1 ] ]
```

Podemos obtener fácilmente la matriz de chequeo de paridad y la matriz generadora, así como la distancia mínima del código cíclico.

```
gap> MinimumDistance(C);
3
gap> G:=GeneratorMat(C);;
gap> Display(G);
1 . 1 1 . . .
. 1 . 1 1 . .
. . 1 . 1 1 .
. . . 1 . 1 1
gap> H:=CheckMat(C);;
gap> Display(H);
1 1 1 . 1 . .
. 1 1 1 . 1 .
. . 1 1 1 . 1
gap> r:= CodewordNr(C,3);
[ 0 0 1 0 1 1 0 ]
```

Ejercicio Dar otro polinomio irreducible sobre $GF(2)$ de grado 3 y obtener los elementos del código cíclico generado por este polinomio.

Los elementos de un $[n, k]$-código lineal sobre algún campo finito GF(p), pueden ser obtenidos de la siguiente forma

```
gap> F:= GF(2);
gap> n:= 10;; k:= 3;;
gap> L:= RandomLinearCode(n,k,F);
a  [10,3,?] randomly generated code over GF(2)
gap> Elements(L);
[ [ 0 0 0 0 0 0 0 0 0 0 ], [ 0 0 0 0 0 1 0 1 0 1 ], [ 0 0 0 0 1 0 0 1 1 0 ],
  [ 0 0 0 0 1 1 0 0 1 1 ], [ 1 1 1 0 0 0 1 1 1 0 ], [ 1 1 1 0 0 1 1 0 1 1 ],
  [ 1 1 1 0 1 0 1 0 0 0 ], [ 1 1 1 0 1 1 1 1 0 1 ] ]
gap> Size(L);
8
gap> MinimumDistance(L);
3
```

Teniendo los elementos de un código podemos obtener sus matrices generadora y de chequeo de paridad, sus parámetros (distancia mínima, dimensión, longitud de las palabras), la distribución de pesos, el polinomio enumerador de pesos, la tabla de sindromes, su código dual, el arreglo estándard, codificar y decoficar palabras. Además podemos verificar si se trata de un código cíclico, de ser así podemos obtener su polinomio generador y su polinomio de chequeo. Por ejemplo, para el código de Hamming $\mathcal{H}_3$ tenemos lo siguiente

```
gap> H:=HammingCode(3);
a linear [7,4,3]1 Hamming (3,2) code over GF(2)
gap> WordLength(H);
7
gap> Redundancy(H);
3
gap> MinimumDistance(H);
3
gap> IsPerfectCode(H);
true
gap> MinimumWeightWords(H);
[ [ 1 0 0 0 0 1 1 ], [ 0 1 0 1 0 1 0 ], [ 0 1 0 0 1 0 1 ],
  [ 1 0 0 1 1 0 0 ], [ 0 0 1 0 1 1 0 ], [ 0 0 1 1 0 0 1 ],
```

```
  [ 1 1 1 0 0 0 0 ] ]
gap> WeightDistribution(H);
[ 1, 0, 0, 7, 7, 0, 0, 1 ]
gap> CodeWeightEnumerator(H);
x^7+7*x^4+7*x^3+1
gap> CodeMacWilliamsTransform(H);
7*x^4+1
gap> c:=Random(H);
[ 0 0 0 1 1 1 1 ]
gap> SyndromeTable(H);
[ [ [ 0 0 0 0 0 0 0 ], [ 0 0 0 ] ], [ [ 1 0 0 0 0 0 0 ], [ 0 0 1 ] ],
  [ [ 0 1 0 0 0 0 0 ], [ 0 1 0 ] ], [ [ 0 0 1 0 0 0 0 ], [ 0 1 1 ] ],
  [ [ 0 0 0 1 0 0 0 ], [ 1 0 0 ] ], [ [ 0 0 0 0 1 0 0 ], [ 1 0 1 ] ],
  [ [ 0 0 0 0 0 1 0 ], [ 1 1 0 ] ], [ [ 0 0 0 0 0 0 1 ], [ 1 1 1 ] ] ]
gap> IsLinearCode(H);
true
gap> IsPerfectCode(H);
true
```

Obsérvese que la función `CodeMacWilliamsTransform(H)` permite obtener un polinomio de la forma:

$$f(x) = \sum_{i=0}^{n} C_i x^i$$

donde C_i es el número de palabras con peso i del código dual $\mathcal{H}_3^{\perp}$. De manera similar, la función `CodeWeightEnumerator(H)` da como resultado un polinomio en la forma:

$$f(x) = \sum_{i=0}^{n} A_i x^i$$

donde A_i es el número de palabras con peso i de $\mathcal{H}_3$.

El siguiente ejemplo muestra como obtener los elementos de un [10,3]-código lineal sobrel el campo $GF(3)$.

```
L:= RandomLinearCode(10,3,GF(3));
a  [10,3,?] randomly generated code over GF(3)
gap> Size(L);
27
gap> MinimumDistance(L);
4
gap> LD:=DualCode(L);
a linear [10,7,1]2..3 dual code
gap> Size(LD);
2187
gap> L!.Dimension;
3
gap> LD!.Dimension;
7
```

Una de las ventajas del paquete GUAVA es que contiene funciones para la construcción de códigos sobre un campo finito arbitrario. Por ejemplo, es fácil obtener un [7,2]-código lineal sobre el campo $GF(11)$:

```
gap> L:= RandomLinearCode(7,2,GF(11));
a  [7,2,?] randomly generated code over GF(11)
gap> Size(L);
121
```

```
gap> LD:=DualCode(L);
a linear [7,5,1..2]2 dual code
gap> Size(LD);
161051
gap> L!.Dimension;
2
gap> LD!.Dimension;
5
```

Note que el número de elementos del código dual, LD, es grande comparado con el número de elementos de L. El lector puede cambiar los parámetros (*n*,*k*) para generar códigos lineales o cíclicos con más elementos.

Otro código que es posible construir con GUAVA, es el código de repetición de longitud 7 sobre el campo de los números binarios, se puede hacer sobre cualquier campo finito, y es cíclico, ésto lo podemos verificar de la siguiente manera

```
gap> R:=RepetitionCode(7,GF(2));
a cyclic [7,1,7]3 repetition code over GF(2)
gap> IsCyclicCode(R);
true
gap> CheckPol(R);
x_1+Z(2)^0
gap> GeneratorPol(R);
x_1^6+x_1^5+x_1^4+x_1^3+x_1^2+x_1+Z(2)^0
gap> G:=GeneratorMat(R);; Display(G);
1 1 1 1 1 1 1
gap> H:=CheckMat(R);; Display(H);
1 1 . . . . .
. 1 1 . . . .
. . 1 1 . . .
. . . 1 1 . .
. . . . 1 1 .
. . . . . 1 1
gap> Elements(R);
[ [ 0 0 0 0 0 0 0 ], [ 1 1 1 1 1 1 1 ] ]
gap> RD:=DualCode(R);
a cyclic [7,6,2]1 dual code
gap> IsCyclicCode(RD);
true
gap> GeneratorPol(RD);
x_1+Z(2)^0
gap> CheckPol(RD);
x_1^6+x_1^5+x_1^4+x_1^3+x_1^2+x_1+Z(2)^0
gap> IsPerfectCode(R);
true
gap> Size(RD);
64
gap> c:=Random(RD);
[ 1 0 1 1 0 1 0 ]
gap> PolyCodeword(c);
x_1^5+x_1^3+x_1^2+Z(2)^0
```

La instrucción PolyCodeword(c) permite expresar una palabra del código como un polinomio, de esta manera podemos hacer operaciones entre los elementos del código. Ahora que tenemos la matriz generadora y el polinomio generador del código una pregunta

natural es ¿qué relación hay entre ambos?, lo mismo ocurre para la matriz de chequeo de paridad y el polinomio de chequeo.

A partir de los códigos que ya tenemos, es posible construir otros códigos, un ejemplo es el código extendido de Hamming.

```
gap> H:=HammingCode(3);
a linear [7,4,3]1 Hamming (3,2) code over GF(2)
gap> HE:= ExtendedCode(H);
a linear [8,4,4]2 extended code
gap> G1:=GeneratorMat(H);; Display(G1);
1 1 1 . . . .
1 . . 1 1 . .
. 1 . 1 . 1 .
1 1 . 1 . . 1
gap> G2:=GeneratorMat(HE);; Display(G2);
1 1 1 . . . . 1
. 1 1 1 1 . . .
. . 1 . 1 1 . 1
. . . 1 1 1 1 .
gap> Size(HE);
16
gap> MinimumDistance(HE);
4
gap> HE!.Dimension;
4
```

Otro ejemplo, es el código de Reed-Muller

```
gap> R:=ReedMullerCode(1,3);
a linear [8,4,4]2 Reed-Muller (1,3) code over GF(2)
gap> Size(R);
16
gap> MinimumDistance(R);
4
gap> R!.Dimension;
4
gap> RG:=GeneratorMat(R);; Display(RG);
1 1 1 1 1 1 1 1
. . . . 1 1 1 1
. . 1 1 . . 1 1
. 1 . 1 . 1 . 1
gap> R2:=AugmentedCode(R,["00000011","00000101","00010001"]);
a linear [8,7,1..2]1 code, augmented with 3 word(s)
gap> G:=GeneratorMat(R2);; Display(G);
1 1 1 1 1 1 1 1
. . . . 1 1 1 1
. . 1 1 . . 1 1
. 1 . 1 . 1 . 1
. . . . . . 1 1
. . . . . 1 . 1
. . . 1 . . . 1
gap> R2=ReedMullerCode(2,3);
true
gap> Size(R2);
128
gap> R2!.Dimension;
7
gap> MinimumDistance(R2);
2
```

Obsérvese que a partir del código (1,3) de Reed-Muller se construye otro utilizando la instrucción `AugmentedCode()` y el código resultante es igual al (2,3) de Reed-Muller.

En general, el paquete GUAVA contiene funciones que permiten obtener otros códigos tales como Reed-Solomon, BCH, Goppa, geométrico-algebraicos, entre otros.

Bibliografía

[1] R. Lidl and G. Pilz, *Applied Abstrac Algebra*, Springer Verlag (1984)
[2] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Applications*, Cambridge University Press (1986)
[3] R. Lidl and H. Niederreiter, *Finite Fields*, Cambridge University Press (1987)
[4] R.J. McEliece, *Finite Fields for Computer Scientist and Engineers*, Kluwer Academic Publishers (1987)
[5] A. Menezes (ed.) *Applications of Finite Fields*, Kluwer Academic Publishers (1993)
[6] M. Wang and I. Blake, ‘‘Bit-serial multiplication in Finite Fields", SIAM, J. Disc. Math.3 (1990), pp.140-148
[7] I. Shparlinski, *Computational Problems in Finite Fields*, Kluwer Academic Publisher (1992)
[8] J. I. Muñoz, *Aritmética rápida sobre campos finitos*, Tesis de Maestría, UAM-I (2001)

Teoría de Códigos

[9] E. Berlekamp, ‘‘Bit-serial Reed-Solomon encoders", IEEE. Trans. Inf. Theory 28, (1982), pp. 869-874
[10] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.* Amsterdam, The Netherlands: North-Holland (1977)
[11] V. Pless, *Introduction to the Theory of Error-Correcting Codes*, 2nd ed., Wiley, New York (1989)
[12] V.D. Goppa *Geometry and Codes*, Kluwer Academic Publishers (1988)
[13] J. H. van Lint and G. van der Geer, *Introduction to Coding Theory and Algebraic Geometry*, DMV Seminar, Band 12, Birhauser Verlag (1988)
[14] C. Rentería, H. Tapia, W.Y. Velez, *Breve introducción a códigos detectores-correctores de errores*, Aportaciones Matemáticas, Comunicaciones No.7, SMM (1990)

Cripotgrafía

[15] W. Diffie, M.E. Hellman, ‘‘New Directions in Cryptography", IEEE. Trans. Inf. Theory 22, (1976), pp. 644-654
[16] T. El Gamal, ‘‘A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE. Trans. Inf. Theory 31, (1985), pp. 469-472
[17] D.E. Robling Denning, *Cryptography and Data Security*, Addison-Wesley Publishing Co. (1983)
[18] P. Caballero, *Seguridad Informática: Técnicas criptográficas*, AlfaOmega Grupo Editor, México (1997)
[19] N. Koblitz, *Algebraic Aspects of Cryptography*, Algorithms and Computation in Mathematics, Vol.3, Springer Verlag (1998)
[20] R. Rivest, A. Shamir and L. Adleman, ‘‘A method for obtaining digital signature and PKC", Communications of the ACM, vol.21, No.2, (1978), pp.120-128
[21] B. Schneier, *Applied Cryptography: Protocols, Algorithms and Source Code in C*, John Wiley & Sons (1994)

[22] D. Stinson, *Cryptography, Theory and Practice*, The CRC Press Series in Discrete Mathematics and its Applications (1995)

[23] A. Menezes et al., *Handbook of Applied Cryptography*, The CRC Press Series in Discrete Mathematics and its Applications (1997)

Teoría de Números

[24] M.R. Schroeder *Number Theory in Science and Communications*, Springer Verlag (1986)

Principales revistas en el tema

Finite Fields and their Applications
IEEE Transactions on Inforamtion Theory
Designs, Codes and Cryptography
Applicable Algebra in Engenineering, Communication and Computing

Algunas direcciones en la Red

`http://www.certicom.com`
`http://www.seguridata.com.mx`
`http://mx.geocities.com/valdesmarrero`
`http://cypher_cobaya.tripod.com`
`http://www.cd.hut.fi`
`http://theory.lcs.mit.edu`
`http://www.digicash.com`