XX ESCUELA VENEZOLANA DE MATEMÁTICAS

BASES DE GRÖBNER: APLICACIONES A LA CODIFICACIÓN ALGEBRAICA

Edgar Martínez Moro
Carlos Munuera Gómez
Diego Ruano Benito

MÉRIDA, VENEZUELA, 2 AL 7 DE SEPTIEMBRE DE 2007

XX ESCUELA VENEZOLANA DE MATEMÁTICAS

BASES DE GRÖBNER: APLICACIONES A LA CODIFICACIÓN ALGEBRAICA

Edgar Martínez Moro
Carlos Munuera Gómez
Dpto. Matemática Aplicada, Universidad de Valladolid
edgar@maf.uva.es, cmunuera@modulor.arq.uva.es
Diego Ruano Benito
Fachbereich Mathematik, Technische Universität Kaiserslautern
ruano@mathematik.uni-kl.de

MÉRIDA, VENEZUELA, 2 AL 7 DE SEPTIEMBRE DE 2007

XX ESCUELA VENEZOLANA DE MATEMÁTICAS

La Escuela Venezolana de Matemáticas es una actividad de los postgrados en matemáticas de las instituciones siguientes: Centro de Estudios Avanzados del Instituto Venezolano de Investigaciones Científicas, Facultad de Ciencias de la Universidad Central de Venezuela, Facultad de Ciencias de la Universidad de Los Andes, Universidad Simón Bolívar, Universidad Centroccidental Lisandro Alvarado y Universidad de Oriente, y se realiza bajo el auspicio de la Asociación Matemática Venezolana. La XX ESCUELA VENEZOLANA DE MATEMÁTICAS recibió financiamiento de la Academia de Ciencias Físicas, Matemáticas y Naturales, la Corporación Andina de Fomento (CAF), el Fondo Nacional de Ciencia, Tecnología e Innovación (FONACIT), la Fundación TALVEN, el Instituto Venezolano de Investigaciones Científicas (Departamento de Matemáticas y Ediciones IVIC), la Universidad de los Andes (CEP, CDCHT, Facultad de Ciencias y Departamento de Matemáticas) y el Rectorado de la Unversidad Centroccidental Lisandro Alvarado.

2000 Mathematics Subject Classification: 13P10, 94B05, 20B40.

Instituto Venezolano de Investigaciones Científicas

Bases de Gröbner: aplicaciones a la codificación algebraica
Edgar Martínez Moro, Carlos Munuera Gómez y Diego Ruano Benito

Diseño y edición: Escuela Venezolana de Matemáticas
Preprensa e impresión: Editorial Texto
Depósito legal lf66020076002491
ISBN 978-980-261-087-7
Caracas, Venezuela
2007

Prefacio

En los últimos años nuestra habilidad para manipular sistemas de ecuaciones expresadas mediante polinomios ha experimentado algunas transformaciones cruciales. Comenzando con el descubrimiento de las bases de Gröbner por B. Buchberger a finales de los años 60 [Buc70] y apoyado por el espectacular crecimiento de las capacidades de los ordenadores modernos muchas herramientas de la geometría algebraica clásica han ganado una gran importancia y, a su vez, la han hecho más asequible y aplicable. Recientemente, las bases de Gröbner han sido aplicadas en multitud de problemas por su capacidad de resolver sistemas de ecuaciones polinomiales y como modelo algebraico de computación.

No es casualidad que en el mismo periodo, desde el artículo seminal de C. Shannon en 1948 [Sha48], muchas herramientas del álgebra aplicada han sido aprovechadas para encontrar códigos correctores de errores con buenas propiedades y para implementar esquemas de codificación y descodificación de los mismos.

Este transcurrir de ambas disciplinas ha proporcionado diversas y fructíferas colaboraciones entre ellas hasta la actualidad, donde las interacciones mutuas son múltiples. Por este motivo proponemos el curso titulado

Bases de Gröbner: aplicaciones a la codificación algebraica

dictado en la XX Escuela Venezolana de Matemáticas como una introducción a la investigación matemática en la codificación algebraica moderna de suma utilidad para aquellos investigadores y alumnos que eventualmente deseen comenzar a trabajar en dicho campo.

Los docentes del curso han venido realizando su trabajo de investigación en este área y cuentan con más de una cincuentena de artículos internacionales de investigación en la misma.

Objetivos del curso

- Introducir las ténicas algebraicas básicas de bases de Gröbner necesarias para la comprensión de la situación actual en la teoría de códigos.

- Mostrar las principales aplicaciones de las bases de Gröbner en la teoría de la codificación, tanto en su parte estructural como en los procesos de codificación y descodificación.

- Mostrar algunas líneas de trabajo en la interacción de ambas disciplinas, tanto en el marco teórico como aplicado.

Prerrequisitos y notas al lector

El curso puede ser seguido teniendo un conocimiento básico de estructuras algebraicas que comprenda cuerpos finitos, anillos de polinomios y álgebra conmutativa básica. También sería conveniente (aunque no necesario) conocer una introducción de bases de Gröbner y de codificación algebraica para poder profundizar más en el contenido del curso. Este mínimo se puede alcanzar con una lectura previa al curso del capítulo 2 "Groebner bases" (pp. 47–73) del libro [CLO97] y el capítulo 2 "Error detection, correction and decoding" (pp. 5–14) de [LX04] o textos similares, o bien los dos primeros capítulos de estas notas que serán repasados de forma somera durante el comienzo del curso.

Hemos tratado que estas notas sean lo más autocontenidas posible, no obstante, dada la extensión del curso a impartir el la XX Escuela Venezolana de Matemáticas se hacen llamadas a resultados que no están cubiertos en las notas y que han sido suplidos con la correspondiente bibliografía. También, al ser la temática muy amplia, hay relacciones entre las bases de Gröbner y códigos correctores de errores que no han podido ser tratados como por ejemplo el tratamiento de los códigos quasi-cíclicos o los códigos álgebro-geométricos y otros que no han sido tratados con la profundidad deseada. Algunas secciones marcadas con ($\star$) no han sido cubiertas durante el curso pero nos han parecido pertinente incluirlas para una lectura posterior.

Agradecimientos

Los autores desean agradecer a La Escuela Venezolana de Matemáticas y todas las organizaciones que la auspician la invitación para dictar este curso y la ayuda brindada para la elaboración de este material. Este agradecimiento se extiende de forma particular a Stella Brasesco presidenta del comité organizador de la XX Escuela Venezolana de Matemáticas.

Septiembre de 2007
Los autores

Índice general

Capítulo 1

Introducción a las bases de Gröbner

Las bases de Gröbner son una herramienta fundamental y básica en muchos aspectos del álgebra computacional. En este capítulo mostraremos una breve introducción a la teoría de las bases de Gröbner forzosamente sesgada, por motivos de tiempo y espacio, hacia aquellos aspectos más relacionados con otras partes del texto. Existe varias referencias excelentes sobre bases de Gröbner y sus aplicaciones tales como [AL96, BW93, CLO97, CLO05, GP02, Win96] y una recopilación teórica en [Mor05]. Este capítulo sigue en gran parte el primer capítulo de [AL96] y de [CLO97].

1.1. Primer contacto

A lo largo de de todas las notas denotaremos por $R = \mathbb{K}[x_1, \ldots, x_n] = \mathbb{K}[\mathbf{x}]$ al anillo de polinomios sobre el cuerpo $\mathbb{K}$. La mayor parte del tiempo nuestro cuerpo será un cuerpo finito de característica p, con p primo, y lo denotaremos por $\mathbb{F}_q$ donde $q = p^l$ y l es un entero mayor que 0 (Véase [LN86] para una introducción a los cuerpos finitos y sus aplicaciones). Un subconjunto no vacio $I \subseteq R$ es un ideal de R si es cerrado bajo la suma de polinomios (esto es, es un subgrupo aditivo) y es cerrado por la multiplicación de elementos de R (esto es, si $f \in R$, $g \in I$ entonces $fg \in I$). Diremos que el ideal I está finitamente generado si existen

$f_1, \ldots, f_s \in R$ tales que

$$I = \langle f_1, \ldots, f_s \rangle \doteqdot \left\{ \sum_{i=1}^{t} h_i f_i \mid h_i \in R \right\} \tag{1.1}$$

El siguiente teorema nos muestra una propiedad importante del anillo de polinomios sobre un cuerpo.

Teorema 1.1 (Teorema de la base de Hilbert)**.**

1. *Si I es un ideal de R entonces está finitamente generado.*
2. *(Condición de cadena ascendente) Si $I_1 \subseteq I_2 \subseteq \cdots \subseteq I_n \subseteq \cdots$ es una cadena ascendente de ideales entonces existe un N tal que $I_N = I_M$ para todo $M > N$.*

Una demostración de este teorema se puede encontrar en la mayoría de los libros citados al comienzo de este capítulo. En este capítulo definiremos y construiremos un sistema de generadores finito $f_1, \ldots, f_s$ de un ideal $I \subseteq R$ que denominaremos base de Gröbner, con ciertas propiedadades deseables que nos ayude a resolver los siguientes problemas:

Problema 1: Determinar si un polinomio f pertenece a un ideal.

Problema 2: En caso de que pertenezca, determinar $u_1, \ldots, u_s \in R$ tales que

$$u_1 f_1 + \cdots + u_s f_s = f\,.$$

Problema 3: Un ideal $I \subseteq R$ define una relación de equivalencia sobre R dada por $f \equiv g \mod I$ si $f - g \in I$ (la comprobación se deja como ejercicio). En conexión con esta construcción el problema es determinar un conjunto de representantes de las clases de equivalencia de R/I y una base de R/I como $\mathbb{K}$-espacio vectorial (posiblemente no finita).

Ejemplo 1.2. *En el caso $R = \mathbb{K}[x]$ el anillo de polinomios en una única variable la respuesta a los tres problemas anteriores es inmediata pues R es un dominio de ideales principales y se tiene que*

$$I = \langle f_1, \ldots, f_s \rangle = \langle g = \mathrm{mcd}(f_1, \ldots, f_s) \rangle$$

donde mcd *denota el máximo común divisor. Por lo tanto f pertenece a I si y sólo si su resto r al dividirlo por g es* 0 *(es decir, f es un múltiplo de g). También $r + I = f + I$ y r es un representante de la clase de equivalencia. Por último, $1, x, x^2, \ldots, x^{d-1}$ donde $d = \text{grado}(g)$ son una base de R/I como $\mathbb{K}$-espacio vectorial.*

Ejemplo 1.3. *En el caso del anillo $R = \mathbb{K}[x, y]$ el problema es más complicado pues no es un dominio de ideales principales. Veremos como tratarlo con posterioridad. Por ejemplo, si consideramos el ideal*

$$I = \langle xy - x,\, y + 1 \rangle$$

y el polinomio $f = xy$, el "resto" de f entre $xy - x$ es x y entre $y + 1$ es $-x$ (en ninguno de los dos casos 0*) sin embargo f pertenece al ideal I pues*

$$f = \frac{1}{2}(xy - x) + \frac{x}{2}(y + 1)$$

Otra forma de definir un ideal en R es la siguiente: Consideremos un subconjunto V (posiblemente infinito) de puntos en el espacio $\mathbb{K}^n$, el conjunto de polinomios

$$\mathcal{I}(V) \doteqdot \{f \in R \mid f(a_1, \ldots, a_n) = 0 \quad \forall (a_1, \ldots, a_n) \in V\} \subseteq R \tag{1.2}$$

es un ideal de R (la demostración se deja como ejercicio). El teorema de la base de Hilbert nos dice que $\mathcal{I}(V)$ está finitamente generado, en otras palabras, cualquier sistema de ecuaciones polinómicas (posiblemente infinito) es equivalente a un sistema con un número finito de ecuaciones. La construcción anterior es muy importante, pues nos proporciona un "diccionario" entre la geometría y el álgebra:

$$\begin{array}{ccc} \text{Subconjuntos de } R & \longrightarrow & \text{Variedades de } \mathbb{K}^n \\ S & \longmapsto & \mathcal{V}(S) \doteqdot \{\mathbf{a} \in \mathbb{K}^n \mid f(\mathbf{a}) = 0 \quad \forall f \in S\} \end{array} \tag{1.3}$$

y la aplicación anterior

$$\begin{array}{ccc} \text{Subconjuntos de } \mathbb{K}^n & \longrightarrow & \text{Ideales en } R \\ V & \longmapsto & \mathcal{I}(V) \end{array} \tag{1.4}$$

Es fácil comprobar (se deja como ejercicio) que $I \subseteq \mathcal{I}(\mathcal{V}(I))$ pero la igualdad no se da siempre. A cada polinomio $f \in R$ se le asocia una

aplicación polinómica de evaluación que, con abuso de notación notaremos por f siempre que no de lugar a error dada por

$$\begin{array}{rccc} f: & \mathbb{K}^n & \longrightarrow & \mathbb{K} \\ & \mathbf{a} & \mapsto & f(\mathbf{a}) \end{array}$$

Nótese que dos funciones f, g son idénticas sobre cierta variedad $\mathcal{V}(I)$ si los correspondientes polinomios cumplen $f - g \in \mathcal{I}(\mathcal{V}(I))$, es decir, si pertenecen a la misma clase de equivalencia en $R/\mathcal{I}(\mathcal{V}(I))$.

1.2. Órdenes monomiales

A la hora de definir una división en polinomios de más de una variable es importante especificar un orden en el conjunto de los monomios

$$\mathbb{T}^n = \left\{ \mathbf{x}^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n} \mid \alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n \right\} \tag{1.5}$$

Definición 1.4. *Definimos un orden monomial sobre $\mathbb{T}^n$ como un orden total $\prec$ que satisface:*

1. *$1 \prec \mathbf{x}^\alpha$ para todo $\mathbf{x}^\alpha \in \mathbb{T}^n$ distinto de 1.*
2. *Si $\mathbf{x}^\alpha \prec \mathbf{x}^\beta$ entonces $\mathbf{x}^\gamma \mathbf{x}^\alpha \prec \mathbf{x}^\gamma \mathbf{x}^\beta$ para todo $\mathbf{x}^\gamma \in \mathbb{T}^n$.*

Ejercicio 1.5. *Comprueba que el único orden monomial en $\mathbb{K}[x]$ el anillo de polinomios en una sola variable es el que viene dado por el grado, esto es, extiende a*

$$\dots \succ x^{n-1} \succ x^n \succ \dots \succ x^2 \succ x \succ 1.$$

Como es usual, escribiremos $\mathbf{x}^\alpha \preceq \mathbf{x}^\beta$ para denotar que $\mathbf{x}^\alpha \prec \mathbf{x}^\beta$ o $\mathbf{x}^\alpha = \mathbf{x}^\beta$. El resultado siguiente nos proporciona dos propiedades importantes de un orden monomial. La primera de ellas nos dice que un orden monomial es compatible con la relación de divisibilidad.

Proposición 1.6. *Sea $\prec$ un orden monomial sobre $\mathbb{T}^n$.*

1. *Sean $\mathbf{x}^\alpha, \mathbf{x}^\beta \in \mathbb{T}^n$, si $\mathbf{x}^\alpha$ divide a $\mathbf{x}^\beta$ entonces $\mathbf{x}^\alpha \preceq \mathbf{x}^\beta$.*
2. *Un orden monomial es un buen orden, esto es, para todo subconjunto $T \subseteq \mathbb{T}^n$ existe $\mathbf{x}^\alpha \in T$ tal que $\mathbf{x}^\alpha \preceq \mathbf{x}^\beta$ para todo $\mathbf{x}^\beta \in T$.*

Demostración.

1. Como $\mathbf{x}^\alpha$ divide a $\mathbf{x}^\beta$ entonces existe $\mathbf{x}^\gamma$ tal que $\mathbf{x}^\alpha\mathbf{x}^\gamma = \mathbf{x}^\beta$, y como $1 \preceq \mathbf{x}^\gamma$ del apartado 2) de la definición 1.4 se sigue el resultado.

2. Supongamos no se cumple la condición en 2). Entonces existe una cadena descendente en $\mathbb{T}^n$ de forma que

$$\mathbf{x}^{\alpha_1} \succ \mathbf{x}^{\alpha_2} \succ \mathbf{x}^{\alpha_3} \succ \dots$$

   De donde la cadena de ideales de R dada por

$$\langle \mathbf{x}^{\alpha_1} \rangle \subsetneq \langle \mathbf{x}^{\alpha_1}, \mathbf{x}^{\alpha_2} \rangle \subsetneq \langle \mathbf{x}^{\alpha_1}, \mathbf{x}^{\alpha_2}, \mathbf{x}^{\alpha_3} \rangle \subsetneq \dots$$

   es una cadena extrictamentetamente creciente de ideales, lo que contradice el teorema de la base de Hilbert. Para ver que las contenciones son extrictas basta ver que no se da el caso $\mathbf{x}^{\alpha_i} = \sum_{j=1}^{i-1} \mathbf{x}^{\alpha_j} f_j$ con $f_j \in R$, pero esto implica que $\mathbf{x}^{\alpha_i}$ es divisible por algún $\mathbf{x}^{\alpha_j}$ que contradice la cadena ascendente de monomios por la parte 1) de la proposición.

□

Definimos dos de los órdenes monomiales más importantes:

Definición 1.7 (Orden lexicográfico)**.** *Sean $\mathbf{x}^\alpha, \mathbf{x}^\beta \in \mathbb{T}^n$ decimos que $\mathbf{x}^\alpha \succ_{\text{lex}} \mathbf{x}^\beta$ si la componente no nula más a la izquierda del vector $\alpha - \beta$ es positiva.*

Definición 1.8 (Orden graduado reverso-lexicográfico)**.** *Sean $\mathbf{x}^\alpha, \mathbf{x}^\beta \in \mathbb{T}^n$ decimos que $\mathbf{x}^\alpha \succ_{\text{grevlex}} \mathbf{x}^\beta$ si $\sum_i \alpha_i > \sum_i \beta_i$ o si $\sum_i \alpha_i = \sum_i \beta_i$ entonces la componente no nula más a la derecha del vector $\alpha - \beta$ es negativa.*

Ejercicio 1.9. *Demuestra que los órdenes definidos en las definiciones 1.7 y 1.8 son dos órdenes monomiales.*

Dado un polinomio $f = \sum_\alpha c_\alpha \mathbf{x}^\alpha \in R$, $0 \neq c_\alpha \in \mathbb{K}$, llamaremos término líder de f respecto al orden monomial $\prec$ y lo notaremos por $\text{lt}_\succ(f)$ al producto $c_\beta \mathbf{x}^\beta$ donde $\mathbf{x}^\beta$ es el mayor monomio que aparece en el polinomio f para el orden monomial $\succ$. Análogamente llamaremos

monomio líder de f respecto al orden monomial $\prec$ a $\mathrm{lm}_\succ(f) = \mathbf{x}^\beta$ y coeficiente líder a $\mathrm{lc}_\succ(f) = c_\beta$. Llamaremos multigrado de f respecto al orden monomial $\prec$ a $\beta = (\beta_1, \ldots, \beta_n)$. Podemos extender de forma obvia el orden monomial $\prec$ a un orden sobre $\mathbb{Z}^n_{\geq 0}$.

La siguiente proposición nos proporciona un resultado que extiende la división euclídea de polinomios en una sola variable al caso multivariable.

Proposición 1.10. *Fijado un orden monomial $\succ$ en $\mathbb{T}^n$ y sea $f_1, \ldots, f_s$ una s-upla* ***ordenada*** *de polinomios de R. Cada $f \in R$ puede ser expresado de la forma*

$$f = u_1 f_1 + \cdots + u_s f_s + r \tag{1.6}$$

donde $u_i, r \in R$ y donde o bien $r = 0$ o bien r es una combinación lineal de monomios no divisibles por los monomios $\{\mathrm{lt}_\succ(f_i)\}_{i=1}^s$.

En el capítulo 1 §5 de [AL96] se muestra un algoritmo que nos proporciona el resultado de la proposición anterior, el llamado algoritmo de división multivariable que reproducimos a continuación

Algoritmo 1.11 (Algoritmo de división multivariable)**.**

Input: *$f_1, \ldots, f_s$ con $f_i \neq 0$ $(1 \leq i \leq s)$ y un orden monomial $\prec$.*

Output: *$u_1 \ldots, u_s, r$ tales que*

$$f = u_1 f_1 + \ldots + u_s f_s + r$$

donde o bien $r = 0$ o bien r es una combinación lineal de monomios no divisibles por los monomios $\{\mathrm{lm}_\succ(f_i)\}_{i=1}^s$

1: $u_i \leftarrow 0$ $1 \leq i \leq s$, $r \leftarrow 0$, $h \leftarrow f$
2: ***while*** $h \neq 0$ ***do***
3: ***if*** *existe un i tal que $\mathrm{lm}_\succ(f_i)$ divide a $\mathrm{lm}_\succ(h)$* ***then***
4: *toma el menor i tal que $\mathrm{lm}_\succ(f_i)$ divide a $\mathrm{lm}_\succ(h)$*
5: $u_i \leftarrow u_i + \dfrac{\mathrm{lt}_\succ(h)}{\mathrm{lt}_\succ(f_i)}$
6: $h \leftarrow h - \dfrac{\mathrm{lt}_\succ(h)}{\mathrm{lt}_\succ(f_i)} f_i$
7: ***else***
8: $r \leftarrow r + \mathrm{lt}_\succ(h)$
9: $h \leftarrow h - \mathrm{lt}_\succ(h)$

10: ***end if***
11: ***end while***

Debemos notar que hay dos diferencias fundamentales con la división euclídea sobre el anillo $\mathbb{K}[x]$ de polinomios de una sola variable. La primera es que permitimos la división por más de un elemento, esto es debido a que nuestro anillo ya no es un dominio de ideales principales como en el caso de una variable. La segunda y más importante es que nuestro resultado $u_1 \ldots, u_s, r$ depende del orden en la s-upla $f_1, \ldots, f_s$ (ver primera condición despues del "if" en el algoritmo) y del orden monomial $\prec$ elegido.

Ejercicio 1.12. *Implementa el algoritmo 1.11 en tu lenguaje de programación favorito.*

1.3. Bases de Gröbner

Hemos visto en la sección anterior que dado un orden monomial y una s-upla ordenada $f_1, \ldots, f_s$ de polinomios de R podemos calcular el resto r de la división multivariable de un polinomio $f \in R$. En el caso de una sola variable un polinomio f pertenece al ideal $\langle g \rangle$ si y sólo si el resto de la división euclídea de f por g es 0. Supongamos ahora el ideal $I = \langle f_1, \ldots, f_s \rangle \subset R$, es claro que si $r = 0$ entonces $f \in I$ pues

$$f = u_1 f_1 + \cdots + u_s f_s$$

pero, ¿Es el recíproco cierto?, la respuesta es negativa en general, como muestra el siguiente ejemplo

Ejemplo 1.13. *Consideremos los polinomios de $\mathbb{Q}[x, y]$ del Ejemplo 1.3, $f_1 = xy - x, f_2 = y + 1$, $f = xy$ y el orden lexicográfico con $y \succ x$. El resto de f entre f_1, f_2 es $r = x \neq 0$ sin embargo f pertenece al ideal $\langle f_1, f_2 \rangle$.*

¿Qué está ocurriendo en el ejemplo anterior? Es claro que el resto $r = x$ pertenece también al ideal $\langle f_1, f_2 \rangle$ pues $r = f - f_1$, pero no existe ningún término líder en f_1, f_2 que pueda cancelarlo. Es decir, para poder conseguir mediante la división multivariable por $f_1, \ldots, f_s$ que un resto no nulo implique la no pertenencia al ideal $I = \langle f_1, \ldots, f_s \rangle$, debemos

poder eliminar **todos** los términos líder de los elementos de I mediante los términos líder de $f_1, \ldots, f_s$.

La discusión anterior motiva la siguiente definición

Definición 1.14 (Base de Gröbner)**.** *Dado un orden monomial $\succ$ y un ideal $I \subset R$ diremos que el conjunto $\{f_1, \ldots, f_s\} \subset R$ es una base de Gröbner de I para $\succ$ si se cumple*

$$\langle \mathrm{lt}_\succ(f_1), \ldots, \mathrm{lt}_\succ(f_s) \rangle = \langle \mathrm{lt}_\succ(I) \rangle \tag{1.7}$$

donde $\mathrm{lt}_\succ(I) = \{\mathrm{lt}_\succ(f) \mid f \in I\}$.

En el siguiente resultado veremos que efectivamente una base de Gröbner es un sistema de generadores del ideal I y demostraremos su existencia.

Teorema 1.15. *Dado un orden monomial $\succ$ y un ideal $I \subset R$ existe una base de Gröbner de I y es un sistema de generadores del ideal I.*

Demostración. El teorema de la base de Hilbert (veáse Teorema 1.1) asegura que existe un sistema de generadores finito $h_1, \ldots, h_s$ para el ideal $\langle \mathrm{lt}_\succ(I) \rangle$. Como $\langle \mathrm{lt}_\succ(I) \rangle$ está generado por términos líderes de I podemos encontrar polinomios $g_1, \ldots, g_s \in I$ tales que

$$h_i \in \langle \mathrm{lt}_\succ(g_1), \ldots, \mathrm{lt}_\succ(g_s) \rangle, \quad 1 \leq i \leq s$$

esto es

$$\langle \mathrm{lt}_\succ(I) \rangle = \langle h_1, \ldots, h_s \rangle \subset \langle \mathrm{lt}_\succ(g_1), \ldots, \mathrm{lt}_\succ(g_s) \rangle \subset \langle \mathrm{lt}_\succ(I) \rangle$$

de donde se sigue que $\langle \mathrm{lt}_\succ(I) \rangle = \langle \mathrm{lt}_\succ(g_1), \ldots, \mathrm{lt}_\succ(g_s) \rangle$ y por lo tanto el conjunto de polinomios $\{g_1, \ldots, g_s\}$ es una base de Gröbner.

Para probar que $\{g_1, \ldots, g_s\}$ es un sistema de generadores de I sólo debemos probar que $\langle g_1, \ldots, g_s \rangle \supset I$ pues los elementos g_i pertenecen a I para $1 \leq i \leq s$. Sea f un elemento de I y consideremos su división multivariable por $\{g_1, \ldots, g_s\}$ utilizando el algoritmo de división multivariable para el orden $\succ$

$$f = u_1 g_1 + \ldots + u_s g_s + r.$$

Supongamos que $r \neq 0$, entonces

$$r = f - (u_1 g_1 + \ldots + u_s g_s) \in I \setminus \{0\}$$

de donde $\mathrm{lt}_\succ(r) \in \langle \mathrm{lt}_\succ(I)\rangle = \langle \mathrm{lt}_\succ(g_1), \ldots, \mathrm{lt}_\succ(g_s)\rangle$ por lo que $\mathrm{lt}_\succ(r)$ debe dividir algún $\mathrm{lt}_\succ(g_i)$ $1 \leq i \leq s$ lo que contradice la Proposición 1.10. □

La última parte de la demostración anterior nos sirve como demostración de la siguiente proposición que veníamos buscando desde el principio de esta sección

Proposición 1.16. *Si $G = \{g_1, \ldots, g_s\}$ es una base de Gröbner del ideal $I \subset R$ entonces $f \in I$ si y sólo si el resto de la división multivariable de f por G es 0.*

El resultado anterior nos muestra una forma algorítmica (una vez que sepamos construir una base de Gröbner para un ideal) para contestar a los Problemas 1 y 2 planteados al comienzo del capítulo. Además, el resto que obtenemos por la división es único como nos muestra el siguiente resultado

Proposición 1.17. *Si $G = \{g_1, \ldots, g_s\}$ es una base de Gröbner del ideal $I \subset R$ para el orden monomial $\succ$ y $f \in R$, entonces f puede ser escrito de forma única como*

$$f = g + r \tag{1.8}$$

donde $g \in I$ y ningún término de r se puede dividir por $\mathrm{lt}_\succ(g_i)$ $1 \leq i \leq s$.

Demostración. Supongamos existen dos polinomios $r, r' \in R$ tales que $f = g + r$ y $f = g' + r'$ con $g, g' \in I$. Esto es

$$r - r' = g' - g \in I.$$

Supongamos $r - r' \neq 0$, entonces existe algún $\mathrm{lt}_\succ(g_i)$ $1 \leq i \leq s$ tal que divide a $\mathrm{lt}_\succ(r - r')$ lo que es imposible, pues ningún $\mathrm{lt}_\succ(g_i)$ $1 \leq i \leq s$ divide a r o a r'. □

Notaremos por

$$\bar{f}^G_\succ = r$$

al único resto de la división multivariante de f por G para el orden monomial $\succ$. En una sección posterior veremos que estos restos nos proporcionan un conjunto de representantes del anillo cociente R/I que dan respuesta al Problema 3 planteado al principio de este capítulo.

1.4. El algoritmo de Buchberger

Las proposiciones 1.16 y 1.17 nos proporcionan dos resultados valiosos siempre y cuando conozcamos una base de Gröbner del ideal $I \subset R$. Sin embargo el teorema 1.15 sólo nos muestra un resultado existencial. En esta sección mostraremos el algoritmo de Buchberger [Buc85] para el cálculo de una base de Gröbner de un ideal dado por un sistema de generadores. La herramienta fundamental son los S-polinomios que esencialmente son la combinación más simple de dos polinomios que cancela sus términos líderes.

Definición 1.18 (S-polinomio)**.** *Sean $f, g \in R$ dos polinomios no nulos y $\succ$ un orden monomial. El S-polinomio de f y g es*

$$S(f,g) = \frac{\mathbf{x}^\gamma}{\mathrm{lt}_\succ(f)} \cdot f - \frac{\mathbf{x}^\gamma}{\mathrm{lt}_\succ(g)} \cdot g, \tag{1.9}$$

donde $\mathbf{x}^\gamma = \mathrm{mcm}\,\{\mathrm{lm}_\succ(f), \mathrm{lm}_\succ(g)\}$.

Notar que el S-polinomio depende del orden monomial $\succ$ elegido, aunque para aliviar la notación no utilizaremos $S(f,g)_\succ$ más que cuando sea extrictamente necesario. El siguiente resultado muestra que cualquier cancelación de términos líderes entre polinomios con el mismo multigrado es producto de una cancelación de S-polinomios.

Proposición 1.19. *Fijado un orden monomial $\succ$, supongamos que $f = \sum_{i=1}^s c_i f_i$, donde $c_i \in \mathbb{K}$ y $f_i \in R$ para $1 \le i \le s$; y el multigrado de f_i es $\delta = (\delta_1, \ldots, \delta_n) \in \mathbb{Z}^n_{\ge 0}$ para todo $1 \le i \le s$. Si el multigrado de f es menor que δ entonces f es una combinación lineal con coeficientes en $\mathbb{K}$ de los S-polinomios*

$$S(f_j, f_k), \quad 1 \le j, k \le s.$$

Además cada S-polinomio tiene multigrado menor o igual que δ.

Demostración. Sea $d_i = \mathrm{lc}_\succ(f_i) \in \mathbb{K}$. Como $c_i f_i$ tiene multigrado δ y f tiene multigrado extrictamente menor que δ entonces $\sum_{i=1}^s d_i c_i = 0$. Sea p_i el polinomio mónico $p_i = f_i/d_i$ $1 \le i \le s$, entonces

$$\begin{aligned} f = \sum_{i=1}^s c_i d_i p_i = c_1 d_1 (p_1 - p_2) + (c_1 d_1 + c_2 d_2)(p_2 - p_3) + \cdots + \\ (c_1 d_1 + \cdots + c_{s-1} d_{s-1})(p_{s-1} - p_s) + (c_1 d_1 + \cdots + c_s d_s) p_s. \end{aligned} \tag{1.10}$$

También se tiene que

$$S(f_j, f_k) = \frac{\mathbf{x}^\delta}{\mathrm{lt}_\succ(f_j)} f_j - \frac{\mathbf{x}^\delta}{\mathrm{lt}_\succ(f_k)} f_k = p_j - p_k, \quad 1 \leq j, k \leq s. \tag{1.11}$$

Utilizando $\sum_{i=1}^{s} d_i c_i = 0$ y la ecuación (1.11) en la ecuación (1.10) se tiene que

$$\begin{aligned} f =& c_1 d_1 S(f_1, f_2) + (c_1 d_1 + c_2 d_2) S(f_2, f_3) + \cdots + \\ & (c_1 d_1 + \cdots + c_{s-1} d_{s-1}) S(f_{s-1}, f_s). \end{aligned} \tag{1.12}$$

que es una suma de la forma deseada. □

Utilizando la proposición anterior podemos probar el siguiente criterio para comprobar cuándo una conjunto de polinomios es una base de Gröbner.

Proposición 1.20 (Criterio de Buchberger)**.** *Fijado un orden monomial $\succ$ sea I un ideal en R. Un sistema de generadores $G = \{g_1, \ldots, g_s\}$ de I es una base de Gröbner si y sólo si*

$$\overline{S(g_i, g_j)}_\succ^G = 0, \quad 1 \leq i, j \leq s, \quad i \neq j.$$

Demostración.

⇒) $S(g_i, g_j)_\succ \in I$, si G es una base de Gröbner aplicando la proposición 1.16 se tiene que $\overline{S(g_i, g_j)}_\succ^G = 0$.

⇐) Sea

$$f = \sum_{i=1}^{s} u_i g_i \tag{1.13}$$

un polinomio de I. Es claro que el multigrado de f es menor o igual que el máximo de los multigrados de los polinomios $u_i g_i$ con $1 \leq i \leq s$. Si la igualdad no ocurre debe existir alguna coincidencia en los multigrados de los polinomios $\{u_i g_i\}_{i=1}^s$.

Consideremos ahora todas las posibles combinaciones del tipo (1.13) en que puede ser escrito f. Como $\succ$ es un buen orden podemos

elegir aquella en que el máximo de los multigrados de los polinomios $\{u_i g_i\}_{i=1}^s$ sea mínimo. Sea ese máximo δ. Supongamos que el multigrado de f es menor que δ, podemos reescribir (1.13) como

$$\begin{aligned} f &= \sum_{m(i)=\delta} u_i g_i + \sum_{m(i)\prec\delta} u_i g_i \\ &= \sum_{m(i)=\delta} \mathrm{lt}_\succ(u_i) g_i + \sum_{m(i)=\delta} (u_i - \mathrm{lt}_\succ(u_i)) g_i + \sum_{m(i)\prec\delta} u_i g_i \end{aligned} \tag{1.14}$$

donde los monomios que aparecen en el segundo y tercer sumando tienen todos multigrado menor que δ. Por lo tanto, que el multigrado de f sea menor que δ implica que el primer sumando también ha de tener multigrado menor que δ.

Sea $\mathrm{lt}_\succ(u_i) = c_i \mathbf{x}^{\alpha(i)}$, esto es

$$\sum_{m(i)=\delta} \mathrm{lt}_\succ(u_i) g_i = \sum_{m(i)=\delta} c_i \mathbf{x}^{\alpha(i)} g_i,$$

la proposición 1.19 implica que esta suma es una combinación lineal de S-polinomios de la forma

$$S(\mathbf{x}^{\alpha(j)} g_j, \mathbf{x}^{\alpha(k)} g_k) = \mathbf{x}^{\delta - \gamma_{jk}} S(g_j, g_k) \quad 1 \le j, k \le s \tag{1.15}$$

donde $\mathbf{x}^{\gamma_{jk}} = \mathrm{mcm}(\mathrm{lm}_\succ(g_j), \mathrm{lm}_\succ(g_k))$. Aplicando la proposición 1.19 existen constantes $c_{jk} \in \mathbb{K}$ tales que

$$\sum_{m(i)=\delta} \mathrm{lt}_\succ(u_i) g_i = \sum_{1 \le j,k \le s} c_{jk} \mathbf{x}^{\delta - \gamma_{jk}} S(g_j, g_k) \tag{1.16}$$

Por hipótesis sabemos que $\overline{S(g_i, g_j)}_\succ^G = 0, \quad 1 \le i, j \le s, \quad i \ne j$, es decir, existen polinomios $a_{ijk} \in R$ tales que

$$S(g_j, g_k) = \sum_{i=1}^s a_{ijk} g_i, \quad 1 \le i, j, k \le s, \quad i \ne j \tag{1.17}$$

donde el multigrado de $S(g_j, g_k)$ es mayor o igual que el de cada producto $a_{ijk} g_i$, $1 \le i, j, k \le s$. Si multiplicamos ambos términos de la igualdad anterior por $\mathbf{x}^{\delta - \gamma_{jk}}$ obtenemos

$$\mathbf{x}^{\delta - \gamma_{jk}} S(g_j, g_k) = \sum_{i=1}^s b_{ijk} g_i, \quad 1 \le i, j, k \le s, \quad i \ne j \tag{1.18}$$

donde $b_{ijk} = \mathbf{x}^{\delta-\gamma_{jk}} a_{ijk}$, y sabemos por la proposición 1.19 que el multigrado de $b_{ijk}g_i$ es menor que el de $\mathbf{x}^{\delta-\gamma_{jk}} S(g_j, g_k)$ y por tanto menor que δ. Sustituyendo en la ecuación (1.16) se tiene

$$\sum_{m(i)=\delta} \mathrm{lt}_\succ(u_i)g_i = \sum_{j,k} c^s_{j,k=1} \left(\sum_{i=1}^{s} b_{ijk} g_i \right) = \sum_{i=1}^{s} \tilde{h}_i g_i \qquad (1.19)$$

con multigrado del polinomio $\tilde{h}_i g_i$ menor que δ para $1 \leq i \leq s$.

Si sustituimos la ecuación (1.19) en la ecuación (1.14) obtenemos f como una combinación de g_i con $1 \leq i \leq s$ y donde todos los términos tienen multigrado menor que δ, lo que contradice la minimalidad de δ. Finalmente hemos llegado a una contradicción, por lo que el multigrado de f ha de ser δ, por lo tanto en la ecuación (1.13) existe algún i tal que $\mathrm{lt}_\succ(g_i)$ divide a f lo que muestra que $\mathrm{lt}_\succ(f) \in \langle \{\mathrm{lt}_\succ(g_i)\}_{i=1}^s \rangle$.

□

A partir de los dos resultados anteriores parece una idea natural que para construir una base de Gröbner de un ideal $I \subset R$ dado por un sistema de generadores $\{f_1, \ldots, f_r\}$ debemos añadir cierta redundancia de generadores. No es una sorpresa, tras la discusión anterior, que los candidatos a nuevos generadores a añadir sean los S-polinomios. Veamos un ejemplo antes de enunciar el algoritmo de Buchberger.

Ejemplo 1.21. *Continuando con los ejemplos 1.3 y 1.13 consideremos los polinomios $f_1 = xy - x, f_2 = y + 1$ en $\mathbb{Q}[x, y]$ y el ideal $I = \langle f_1, f_2 \rangle$. El conjunto $\{f_1, f_2\}$ no es una base de Gröbner del ideal para el orden lexicográfico con $x \prec y$ pues*

$$S(f_1, f_2) = \frac{xy}{xy} f_1 - \frac{xy}{y} f_2 = (xy - x) - x(y + 1) = -2x,$$

cuyo resto no es 0 al dividir por $\{f_1, f_2\}$. Además es claro que $f_3 = S(f_1, f_2) = -2x$ pertenece a I. Si añadimos el nuevo polinomio $f_3 \in I$ al conjunto de generadores $F = \{f_1, f_2, f_3\}$ se tiene $\overline{S(f_1, f_2)}^F_\succ = 0$. Calculemos ahora los S-polinomios restantes:

$$S(f_1, f_3) = 2x, \quad \Rightarrow \overline{S(f_1, f_3)}^F_\succ = 0$$

$$S(f_2, f_3) = -2x, \quad \Rightarrow \overline{S(f_2, f_3)}^{F}_{\succ} = 0$$

por lo que F es una base de Gröbner de I por la proposición 1.20.

El resultado siguiente nos asegura que mediante el procedimiento anterior (llamado algoritmo de Buchberger) siempre se llega a una base de Gröbner en un número finito de pasos.

Algoritmo 1.22 (Algoritmo de Buchberger)**.**

Input: *$F = \{f_1, \ldots, f_s\}$ con $I = \langle \{f_i\}_{i=1}^{s} \rangle \neq \{0\}$ y $\succ$ un orden monomial.*

Output: *Una base de Gröbner para el ideal I en el orden monomial $\succ$.*

1: $G \leftarrow F$, $G' \leftarrow \{0\}$
2: ***while*** $G \neq G'$ ***do***
3: $G' \leftarrow G$
4: ***for*** *cada par* $\{p, q\} \subset G'$ *con* $p \neq q$ ***do***
5: $S \leftarrow \overline{S(p, q)}^{G'}_{\succ}$
6: ***if*** $S \neq 0$ ***then***
7: $G \leftarrow G \cup \{S\}$
8: ***end if***
9: ***end for***
10: ***end while***

Proposición 1.23. *Sea $I = \langle f_1, \ldots, f_s \rangle \subset R$ un ideal distinto de $\{0\}$, entonces se puede construir una base de Gröbner del ideal I para un orden monomial $\succ$ mediante el algoritmo 1.22*

Demostración. Mostraremos primero que $G \subset I$ en cada paso del algoritmo, lo que es claro pues $\overline{S(p, q)}^{G'}_{\succ} \in I$ por pertenecer p y q a I. Además, G es un sistema de generadores de I en cada paso pues contiene a F.

El algoritmo termina cuando $G = G'$, es decir $\overline{S(p, q)}^{G'}_{\succ} = 0$ para todo par $p, q \in G'$, es decir, por el criterio de Buchberger (proposición 1.20) el resultado es una base de Gröbner.

Falta comprobar que el algoritmo termina.Notemos por $\mathrm{lt}_{\succ}(G) = \{\mathrm{lt}_{\succ}(g) \mid g \in G\}$. Es claro que

$$\left\langle \mathrm{lt}_{\succ}(G') \right\rangle \subseteq \langle \mathrm{lt}_{\succ}(G) \rangle \,. \tag{1.20}$$

Es más, en el caso $G \neq G'$ la contención es extricta pues supongamos que un nuevo resto $r \neq 0$ se añade a G en el paso 7 del algoritmo, $\mathrm{lt}_\succ(r)$ no es divisible por los téminos líderes de los elementos de G' y entonces

$$\mathrm{lt}_\succ(r) \in \langle \mathrm{lt}_\succ(G) \rangle , \quad \mathrm{lt}_\succ(r) \notin \left\langle \mathrm{lt}_\succ(G') \right\rangle .$$

Esto es, la ecuación (1.20) muestra una cadena ascendente de ideales que ha de estabilizarse (ver teorema 1.1) en algún momento en que $G = G'$. □

Hemos visto que en el algoritmo 1.22 en cada paso efectivo se añade un nuevo generador al conjunto G. Esto lleva a que la base de Gröbner resultante sea altamente redundante en generadores, el siguiente resultado nos ayuda a reducir su tamaño.

Proposición 1.24. *Sea G una base de Gröbner del ideal $I \subset R$ y el orden monomial $\succ$. Sea $g \in G$ tal que $\mathrm{lt}_\succ(g) \in \langle \mathrm{lt}_\succ(G \setminus \{g\}) \rangle$. Entonces $G \setminus \{g\}$ también es una base de Gröbner del ideal $I \subset R$ y el orden monomial $\succ$.*

Demostración. Por definición de base de Gröbner $\langle \mathrm{lt}_\succ(G) \rangle = \langle \mathrm{lt}_\succ(I) \rangle$, pero $\langle \mathrm{lt}_\succ(G) \rangle = \langle \mathrm{lt}_\succ(G \setminus \{g\}) \rangle$, por lo tanto $G \setminus \{g\}$ es una base de Gröbner del ideal $I \subset R$. □

También podemos ajustar las constantes de forma que tengamos todos los polinomios mónicos (es decir con coeficiente líder 1):

Definición 1.25. *Llamaremos base de Gröbner minimal de un ideal $I \subset R$ respecto del orden monomial $\succ$ a una base de Gröbner G tal que:*

1. $\mathrm{lc}_\succ(g) = 1$ *para todo* $g \in G$.
2. *Para todo* $g \in G$ *se tiene* $\mathrm{lt}_\succ(g) \notin \langle \mathrm{lt}_\succ(G \setminus \{g\}) \rangle$.

Ejercicio 1.26. *Comprueba que si tenemos dos bases de Gröbner minimales G y $\tilde{G}$ de un ideal $I \subset R$ respecto del orden monomial $\succ$ se tiene que $\mathrm{lt}_\succ(G) = \mathrm{lt}_\succ(\tilde{G})$. En particular, contienen el mismo número de elementos.*

Aún con los condicionantes de una base minimal respecto de un orden dado la base de Gröbner no es única, por ejemplo los polinomios

$$f_1 = y + ax + 1, \quad f_2 = x, \quad a \in \mathbb{Q}$$

son una base minimal de Gröbner para el ideal del ejemplo 1.21 para el orden lexicográfico, con $x \prec y$, para cualquier elección del parámetro $a \in \mathbb{Q}$. Afortunadamente podemos distinguir una como la más adecuada.

Definición 1.27. *Llamaremos base de Gröbner reducida de un ideal $I \subset R$ respecto del orden monomial $\succ$ a una base de Gröbner G tal que:*

1. $\mathrm{lc}_\succ(g) = 1$ *para todo* $g \in G$.
2. *Para todo $g \in G$ ningún monomio de g pertence a $\langle \mathrm{lt}_\succ(G \setminus \{g\})\rangle$.*

Es claro que una base reducida es minimal. En el caso anterior cuando el parámetro $a = 0$ la base es reducida.

Proposición 1.28. *Sea $\{0\} \neq I \subset R$ un ideal. Para un orden monomial $\succ$ dado existe una única base de Gröbner reducida.*

Demostración. Sea G una base de Gröbner minimal del ideal I. Diremos que el polinomio g es reducido respecto a G si ningún monomio de g pertenece a $\langle \mathrm{lt}_\succ(G \setminus \{g\})\rangle$.

Dado un elemento $g \in G$ definimos $g' = \bar{g}_\succ^{G\setminus\{g\}}$, es fácil comprobar que $G' = (G \setminus \{g\}) \cup \{g'\}$ es también una base minimal de I. Ahora tomemos todos los elementos de G y repetimos el procedimiento anterior hasta que la base es reducida, como los términos líderes nunca cambian, una vez un polinomio es reducido permanece reducido hasta el final del proceso, de donde el resultado final es una base de Gröbner reducida.

Supongamos ahora que existen dos bases reducidas G y $\tilde{G}$ del ideal I para el orden $\succ$, por lo tanto $\mathrm{lt}_\succ(G) = \mathrm{lt}_\succ(\tilde{G})$ (ver ejercicio 1.26). Consideremos los elementos $g \in G$ y $\tilde{g} \in \tilde{G}$ tales que $\mathrm{lt}_\succ(g) = \mathrm{lt}_\succ(\tilde{g})$. Claramente, por pertenecer al ideal,

$$\overline{(g - \tilde{g})}_\succ^G = 0,$$

como ambos tienen igual monomio líder y son reducidos respecto el resto de su correspondiente base se tiene que $\overline{(g - \tilde{g})}_\succ^G = (g - \tilde{g})$ y se sigue que $g = \tilde{g}$. □

No es el objetivo de estas notas hacer un análisis de la complejidad del cálculo de una base de Gröbner, simplemente notar que esisten refinamientos del algoritmo de Buchberger que permiten una mayor eficiencia.

En general si n es el número de variables y d es el grado máximo de los polinomios que generan el ideal se tiene que el mayor grado de un polinomio que aparezca en una base reducida del ideal para cualquier orden $D(n,d)$ está acotado por (ver [Dub89, MM84])

$$cd^{2^n} \leq D(n,d) \leq d^{2^n}$$

donde c es una constante.

Para finalizar esta sección en el siguiente ejercicio se hacen notar las semejanzas entre las bases de Gröbner para el orden lexicográfico y la eliminación gaussiana.

Ejercicio 1.29. *Compara los pasos necesarios de reducción por filas para realizar la eliminación gaussiana (hasta su forma reducida) del sistema de ecuaciones lineales*

$$\begin{pmatrix} 2 & -3 & 4 & 0 \\ 1 & 4 & 0 & 3 \\ -1 & -15 & 4 & -9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

con el cálculo de la base reducida de Gröbner del ideal

$$I = \langle 2x - 3y + 4z,\ x + 4y + 3w,\ -x - 15y + 4z - 9w \rangle$$

para el orden lexicográfico con $x \succ y \succ z \succ w$.

1.5. Introducción a la eliminación

La eliminación de variables, es decir, encontrar ecuaciones que sólo relacionen las soluciones de ciertas variables excluyendo el resto, es una de las técnicas más utilizadas en la resolución de sistemas de ecuaciones. Por ejemplo, consideramos el sistema de ecuaciones

$$\begin{cases} xy - x + 1 &= 0 \\ x - 2y &= 0 \end{cases}$$

y el polinomio

$$(xy - x + 1) + \frac{1}{2}x(x - 2y) = \frac{1}{2}x^2 - x + 1 = 0$$

que elimina la variable y y nos permite calcular los valores de x que satisfacen el sistema de ecuaciones.

En general dado un sistema de s ecuaciones

$$f_1 = 0, \ldots, f_s = 0 \tag{1.21}$$

donde $f_i \in R$ para $1 \leq i \leq s$ podemos considerar el ideal $I = \langle f_1, \ldots, f_s \rangle$ y las intersecciones

$$\begin{aligned} &I \cap \mathbb{K}[x_2, x_3, \ldots, x_n] \text{ que elimina } x_1, \\ &I \cap \mathbb{K}[x_3, \ldots, x_n] \text{ que elimina } x_1, x_2, \\ &\vdots \\ &I \cap \mathbb{K}[x_n] \text{ que elimina } x_1, \ldots, x_{n-1}. \end{aligned} \tag{1.22}$$

A estos ideales se les denomina *ideales de eliminación*, el objetivo principal de la teoría de la eliminación es buscar un sistema de generadores para cada uno de ellos. Si nosotros disponemos de una base de Gröbner para el orden lexicográfico.

Proposición 1.30 (Teorema de eliminación)**.** *Sea $I = \langle f_1, \ldots, f_s \rangle \subset R$ un ideal y $G = \{g_1, \ldots, g_t\}$ una base de Gröbner de I respecto del orden lexicográfico $\succ_{\text{lex}}$, entonces para cada k con $2 \leq k \leq n$ el conjunto*

$$G \cap \mathbb{K}[x_k, \ldots, x_n]$$

es una base de Gröbner respecto del orden lexicográfico del ideal de eliminación

$$I \cap \mathbb{K}[x_k, \ldots, x_n].$$

Demostración. Consideremos un elemento $f \in I \cap \mathbb{K}[x_k, \ldots, x_n]$. Como $f \in I$ existe un elemento $g \in G$ tal que $\text{lt}_{\succ_{\text{lex}}}(g)$ divide a $\text{lt}_{\succ_{\text{lex}}}(f)$, además $\text{lt}_{\succ_{\text{lex}}}(f)$ no posee divisores entre $x_1, \ldots, x_{k-1}$ y por lo tanto tampoco $\text{lt}_{\succ_{\text{lex}}}(g)$. Como en el orden lexicográfico $x_1 \succ_{\text{lex}} x_2 \succ_{\text{lex}} \cdots \succ_{\text{lex}} x_n$ un monomio que sea divisible por $x_1, \ldots, x_{k-1}$ es mayor que uno en $\mathbb{K}[x_k, \ldots, x_n]$,por lo tanto

$$\text{lt}_{\succ_{\text{lex}}}(g) \in \mathbb{K}[x_k, \ldots, x_n] \Rightarrow g \in \mathbb{K}[x_k, \ldots, x_n].$$

Por lo tanto para un $0 \neq f \in I \cap \mathbb{K}[x_k, \ldots, x_n]$ se tiene que existe un elemento $g \in G \cap \mathbb{K}[x_k, \ldots, x_n]$ tal que $\text{lt}_{\succ_{\text{lex}}}(g)$ divide a $\text{lt}_{\succ_{\text{lex}}}(f)$, esto es, $G \cap \mathbb{K}[x_k, \ldots, x_n]$ es una base de Gröbner del ideal de eliminación $\square$

Una de las principales desventajas de esta aproximación a la eliminación es que el orden lexicográfico a menudo nos lleva a bases de Gröbner muy grandes y costosas de calcular. Una alternativa consiste en crear ordenes de eliminación para cierto número de variables basados en otro orden monomial que tenga buenas propiedades de cálculo de la base de Gröbner como el graduado reverso lexicográfico (ver [BS87]). Otra alternativa para ideales 0-dimensionales (ver sección 1.6) es calcular la base de Gröbner para el orden graduado reverso lexicográfico y posteriormente mediante técnicas de álgebra lineal (ver [FGLM93]) calcular la base de Gröbner para el orden lexicográfico.

Otro aspecto a tener en cuenta en la teoría de la eliminación es el de la sustitución progresiva de las soluciones sistema a partir de los ideales de eliminación anteriores, es decir, cuando una solución parcial $(s_k, \ldots, s_n) \in \mathbb{K}^{n-k+1}$ en $\mathcal{V}(I \cap \mathbb{K}[x_k, \ldots, x_n])$ se puede extender a una solución $(s_{k-1}, s_k, \ldots, s_n) \in \mathbb{K}^{n-k+2}$ en $\mathcal{V}(I \cap \mathbb{K}[x_{k-1}, \ldots, x_n])$. El siguiente resultado nos proporciona esta información cuando $\mathbb{K}$ es algebraicamente cerrado.

Proposición 1.31 (Teorema de extensión)**.** *Sea $I = \langle f_1, \ldots, f_s \rangle \subset R$ y $\mathbb{K}$ un cuerpo algebraicamente cerrado. Consideremos el ideal de eliminación $I_1 = I \cap \mathbb{K}[x_2, x_3, \ldots, x_n]$. Para cada $1 \leq i \leq s$ se puede escribir*

$$f_i = \tilde{f}_i(x_2, x_3, \ldots, x_n) x_1^{N_i} + (\textit{términos con grado de } x_1 \ < N_i)\,,$$

con $N_i \geq 0$ y $\tilde{f}_i(x_2, x_3, \ldots, x_n) \neq 0$. Sea $(s_2, \ldots, s_n) \in \mathcal{V}(I_1)$ una solución de I_1, si $f_i(s_2, s_3, \ldots, s_n) \neq 0$ para al menos un valor de $1 \leq i \leq s$ entonces existe $s_1 \in \mathbb{K}$ tal que $(s_1, s_2, s_3, \ldots, s_n) \in \mathcal{V}(I)$.

Una demostración de la proposición anterior requiere el uso de resultantes y excede el objetivo de estas notas, el lector interesado puede consultar [CLO97, §6].

1.6. Álgebras de dimensión finita

En esta sección estudiaremos el problema 3 de la sección 1.1 en el caso finito dimensional. Es decir, la "aritmética de los restos" asociada a una base de Gröbner $G \subset R$ respecto a un orden dado $\succ$ o, más formalmente, estudiaremos el anillo $R/\langle G \rangle$. Sea $I = \langle G \rangle$, dado cualquier polinomio

$f \in R$ sabemos que $\bar{f}^G_\succ$ es una combinación lineal de aquellos monomios $\mathbf{x}^\alpha \notin \langle \mathrm{lt}_\succ(I)\rangle$, además

$$f, g \in R, \qquad \bar{f}^G_\succ = \bar{g}^G_\succ \Leftrightarrow f - g \in I. \tag{1.23}$$

Es fácil comprobar que existe una relación uno a uno entre los restos y los representantes del anillo cociente R/I compatible con la suma y el producto.

Ejercicio 1.32. *Comprueba la frase anterior, es decir, comprueba que dados $f, g \in R$ se tiene*

1. $\overline{f}^G_\succ + \overline{g}^G_\succ = \overline{(f+g)}^G_\succ$.
2. $\overline{f}^G_\succ \cdot \overline{g}^G_\succ = \overline{(f \cdot g)}^G_\succ$.

Podemos dotar al anillo cociente R/I de una estructura de $\mathbb{K}$-espacio vectorial, esto es, un $\mathbb{K}$-álgebra que notaremos por $A = R/I$. Una base de A como $\mathbb{K}$-espacio vectorial es la base cuyos representantes son los monomios en el conjunto (monomios estándar o canónicos)

$$\mathcal{B} = \{\mathbf{x}^\alpha \notin \langle \mathrm{lt}_\succ(I)\rangle\}. \tag{1.24}$$

El siguiente resultado caracteriza cuándo el $\mathbb{K}$-álgebra $A = R/I$ posee dimensión finita como $\mathbb{K}$-espacio vectorial (una demostración se puede encontrar en [AL96, Teorema 2.2.7]):

Teorema 1.33. *Sea $\mathbb{K}$ un cuerpo algebraicamente cerrado y $I \subset R$ un ideal. Las siguientes condiciones son equivalentes:*

1. *El álgebra $A = R/I$ tiene dimensión finita como $\mathbb{K}$-espacio vectorial.*
2. *La variedad $\mathcal{V}(I)$ tiene un número finito de puntos.*
3. *Si G es una base de Gröbner del ideal I, para cada i, $1 \leq i \leq n$ existe un entero positivo m_i con $x_i^{m_i} = \mathrm{lt}(g)$, donde $g \in G$.*

Capítulo 2

Códigos correctores de errores

Cuando utilizamos un objeto tan cotidiano como un disco compacto (por ejemplo, de audio) pocas veces somos conscientes de su complejidad. Para su fabricación se ha discretizado la función "sonido" a partir de tomas discretas de datos: 2×44100 tomas por segundo (el sistema es estereofónico y permite reproducir frecuencias de hasta 20000 Hz.) cada una de las cuales se ajusta a una escala de 2^{16} niveles. Esto se traduce en $44100 \times 16 \times 2 = 1411200$ bits de información por segundo para almacenar en el disco. Además, por motivos técnicos, cada 8 bits de datos (*audiobytes*) se graban en el disco mediante 17 bits. Con esto, cada segundo de música requiere almacenar 2998800 bits en el disco.

Con estos enormes volumenes de datos, no es extraño que uno de los problemas más importantes que genera la manipulación y transmisión de la información digital sea el de los errores. Basta una pequeña alteración del soporte que contiene o transmite la información (un rayón sobre un disco o una onda parásita) para que una parte del mensaje se corrompa: algunos de los 0 serán leídos como 1 y viceversa. En el caso del disco compacto, un pequeño rayón de 1 mm. puede alterar entre 2000 y 4000 bits. Por tanto es preciso desarrollar algún mecanismo que permita detectar cuando se han producido errores y, si es posible, corregirlos recuperando la información original. Con este propósito nacieron en los años 50 (coincidiendo con el desarrollo de las primeras computadoras) los *códigos correctores de errores*.

En los inicios de la telefonía, la calidad del sonido transmitido era muy pobre. A menudo, para trasmitir una palabra se utilizaba un códi-

go basado en iniciales. Por ejemplo, la palabra *error* se codificaba como "Eco-Romeo-Romeo-Omega-Romeo". Desde la invención de estos primeros códigos, hasta los que usamos hoy en día, se ha experimentado una tremenda evolución. Sin embargo la filosofía subyacente sigue siendo la misma: el mensaje original se trocea en partes, cada una de las cuales se codifica mediante la inclusión sistemática de información redundante. En base a esa redundancia es posible detectar los errores producidos y, si no sobrepasan ciertas cotas, corregirlos. Como contrapartida, el precio que pagamos es el aumento del tamaño de los mensajes transmitidos (o, visto de otra forma, la reducción en la cantidad de información 'neta' que contiene cada mensaje recibido).

La *Teoría de los códigos correctores de errores* forma hoy un extenso y fructífero campo de interacción entre las matemáticas y las tecnologías de la información, en el que conceptos y resultados matemáticos abstractos permiten dar elegantes soluciones al problema de transmitir información de forma eficiente y segura. Entre estos conceptos matemáticos juegan un papel relevante el álgebra lineal y la teoría de bases de Gröbner. Para un tratamiento más completo de esta teoría pueden consultarse [MS85, MT97, JH04, PH98].

2.1. La información y los errores

2.1.1. La información digital

La información digital se caracteriza por presentarse en un formato discreto. Sean $\mathcal{A}$ un conjunto finito (o *alfabeto*) y $\mathcal{A}^*$ el conjunto de secuencias finitas de elementos de $\mathcal{A}$. Una *información digital* (o *palabra*, o *mensaje*) es simplemente una secuencia $m = x_1x_2\cdots \in \mathcal{A}^*$. Un texto escrito (este libro) es un buen ejemplo de información digital. Por conveniencia, el alfabeto $\mathcal{A}$ suele identificarse con algún sistema numérico y muy a menudo con $\{0, 1\}$, conjunto que interpretaremos como el cuerpo finito $\mathbb{F}_2$. De manera análoga, si $\mathcal{A}$ contiene q elementos y q es la potencia de un número primo, entonces $\mathcal{A}$ se identifica con el cuerpo finito con q elementos $\mathbb{F}_q$. Esta identificación permite aplicar a los problemas de codificación todos los recursos del álgebra y la geometría sobre cuerpos finitos.

Una vez se tiene la información en el formato digital adecuado (binario o no) es apta para su manipulación o transmisión, siguiendo un esquema

del tipo

Figura 2.1: Esquema de una transmisión de información.

En un sentido amplio, el canal puede ser espacial o temporal: envío por línea telefónica, óptica, almacenamiento en un disco, etc. Al recibir el mensaje, el receptor no puede estar seguro de que alguna parte del mismo haya sido corrompida durante la transmisión. Sí puede, sin embargo, conocer la frecuencia con que se producen los errores y, por tanto, determinar cuantos errores (en media) cabe esperar que hayan ocurrido.

En lugar de enviar la información directamente, la transformamos (*codificamos*) añadiéndole cierta redundancia con arreglo a unas reglas sistemáticas. Es esta información codificada la que realmente se transmite por el canal (Figura 2). En base a la redundancia añadida el receptor puede detectar, y eventualmente corregir, los errores producidos y devolverla a su formato original. Este proceso recibe el nombre de *descodificación.*

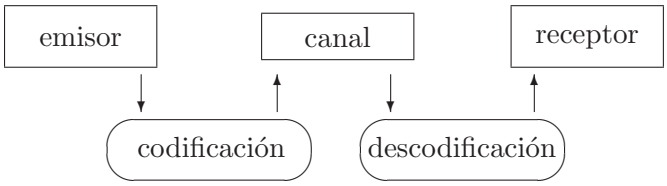

Figura 2.2: Esquema de una transmisión de información codificada.

2.1.2. Códigos correctores

Como hemos señalado, la información se presenta originalmente como una secuencia $m = x_1x_2\cdots \in \mathcal{A}^*$. Fijamos dos enteros $k < n$ y troceamos m en bloques de longitud $k : m = (x_1\cdots x_k)(x_{k+1}\cdots x_{2k})\cdots$. Cada uno de estos bloques se codifica (y posteriormente se enviará y descodificará) independientemente de los demás, como su imagen mediante una

aplicación inyectiva $c : \mathcal{A}^k \longrightarrow \mathcal{A}^n$. La codificación del mensaje completo se obtiene concatenando la codificación de los bloques que lo integran:

$$c(m) = c(x_1, \ldots, x_k)c(x_{k+1}, \ldots, x_{2k}) \cdots$$

El conjunto $\mathcal{C} = \mathrm{Im}(c)$ es, por definición, el *código* utilizado.

Definición 2.1. *Un* código corrector de errores *es un subconjunto* $\mathcal{C} \subseteq \mathcal{A}^n$*, siendo* $\mathcal{A}$ *un alfabeto finito y* n *un entero positivo. Los elementos de* $\mathcal{C}$ *son llamados* palabras *y* n *es su* longitud.

Cada palabra de $\mathcal{C}$ contiene k símbolos de información y $n-k$ símbolos redundantes: el número k/n se llama *tasa de transmisión* de $\mathcal{C}$.

Supongamos que se ha enviado una palabra $\mathbf{c} \in \mathcal{C}$ y recibido un vector $\mathbf{x} \in \mathcal{A}^n$. Si p es la probabilidad de que un símbolo resulte alterado en la transmisión, podemos esperar una media de np símbolos erróneos en $\mathbf{x}$. La capacidad de corrección de errores de $\mathcal{C}$ debe superar al menos esa cota. Cuando $\mathbf{x} \notin \mathcal{C}$, ciertamente ha habido errores; de hecho, aún en el caso de que $\mathbf{x} \in \mathcal{C}$, nunca podremos estar realmente seguros de que no hayan existido errores. Ahora bien, si el código se ha diseñado correctamente, sus palabras serán muy 'diferentes' unas de otras, de manera que resulte 'suficientemente improbable' que $\mathbf{x} \in \mathcal{C}$ a causa de los errores aleatorios sufridos en el canal.

La forma adecuada de medir la diferencia entre dos palabras (o dos vectores de $\mathcal{A}^n$) es la distancia de Hamming.

Definición 2.2. *Dados* $\mathbf{x}, \mathbf{y} \in \mathcal{A}^n$*, llamamos* distancia de Hamming *entre* $\mathbf{x}$ *e* $\mathbf{y}$ *al número de coordenadas distintas que poseen,*

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

Obsérvese que la función d es realmente una distancia en $\mathcal{A}^n$. El hecho de que d *no* sea invariante por cambios de base, hace que la teoría de códigos no sea una parte trivial del álgebra lineal. La capacidad de corrección de errores de $\mathcal{C}$ viene determinada por su *distancia mínima*, definida como

$$d = d(\mathcal{C}) = \mathrm{mín}\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in \mathcal{C},\, \mathbf{x} \neq \mathbf{y}\}.$$

En efecto, recibido el vector $\mathbf{x} \in \mathcal{A}^n$, se descodifica $\mathbf{x}$ por la palabra $\mathbf{c} \in \mathcal{C}$ 'más parecida' a $\mathbf{x}$, es decir, que minimiza $d(\mathbf{x}, \mathbf{c})$. Como las bolas (para

la métrica de Hamming) de radio $(d-1)/2$ centradas en las palabras del código son disjuntas, si el número de errores en $\mathbf{x}$ no supera $\lfloor (d-1)/2 \rfloor$, entonces la palabra corregida coincide con la realmente enviada. Así, nuestra estrategia permite detectar $d-1$ errores y corregir $\lfloor (d-1)/2 \rfloor$ errores.

El objetivo principal (o mejor, uno de los objetivos principales) de la teoría de códigos correctores de errores es encontrar *buenos* códigos, es decir, códigos que maximicen solidariamente los parámetros k/n y d/n. Sin embargo estas demandas son mutuamente contradictorias: al aumentar uno de los parámetros, el otro tiende siempre a disminuir. En la práctica habremos de conformarnos con un cierto equilibrio entre ellos.

Otro requerimiento importante para un buen código es que posea algún método de descodificación computacionalmente efectivo. El sistema al que nos hemos referido anteriormente –evaluar la distancia de $\mathbf{x}$ a todas las palabras de $\mathcal{C}$ y quedarnos con la más cercana– es inviable en la práctica (excepto para códigos de pequeño tamaño). Relativamente pocos códigos permiten estos métodos efectivos. En el lenguaje de la Teoría de la Complejidad Computacional, el problema de descodificar un código es NP-Completo. Retomaremos el tema de los buenos códigos un poco más adelante.

2.1.3. Algunos ejemplos

Ejemplo 2.3 (El código ASCII)**.** *En su versión habitual (no extendida), ASCII permite codificar* $128 = 2^7$ *símbolos (letras, números, signos y controles no imprimibles) de uso general para computadoras. A cada uno de ellos se le asigna un número de orden y se le codifica mediante la escritura binaria (con 7 bits) de ese número. Para aumentar la fiabilidad de esta codificación, a cada 7-upla* $x_1, \ldots, x_7 \in \mathbb{F}_2^7$ *se le añade un bit control* x_8*, calculado de manera que* $x_1 + \cdots + x_7 + x_8 \equiv 0$ *(mod 2). Este sistema permite detectar, aunque no corregir, cualquier número impar de errores.*

Ejemplo 2.4 (Códigos de Hamming)**.** *Los códigos de Hamming constituyen una familia doblemente infinita de códigos (para cada potencia* q *de un número primo existe una familia infinita de ellos). Vamos a describir con cierto detalle al más pequeño.*

Deseamos codificar una 4*-upla* $(x_1, x_2, x_3, x_4) \in \mathbb{F}_2^4$*. Vamos a hacerlo añadiendo a estos cuatro bits otros tres redundantes,* $c(x_1, x_2, x_3, x_4) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$*. Para ello consideremos tres circunferencias cortándose en posición general, tal y como se ve en el dibujo. Estas tres circunferencias determinan* 7 *regiones (más la exterior no acotada). Numerémoslas.*

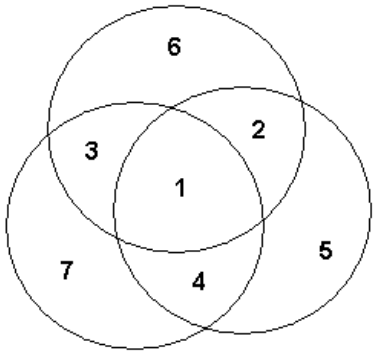

Figura 2.3: Tres circunferencias.

Colocamos cada bit x_i *en la región* i *de la figura,* $(i = 1, \ldots, 7)$*. Los* x_5, x_6, x_7*, se calculan de manera que cada círculo contenga un número par de 1. Por ejemplo, (1010) se codifica como (1010100). El conjunto* $\mathcal{C} = \{c(x_1, x_2, x_3, x_4) \mid (x_1, x_2, x_3, x_4) \in \mathbb{F}_2^4\}$ *se llama* código de Hamming binario de redundancia 3 *y habitualmente se denota* $\mathcal{H}_2(3)$*.*

Es un ejercicio instructivo y fácil comprobar que dos palabras cualesquiera de $\mathcal{H}_2(3)$ *se diferencian en al menos tres coordenadas (es decir, que* $\mathcal{H}_2(3)$ *tiene distancia mínima* $d = 3$*) y diseñar un algoritmo de corrección de errores. Se observará que en este código (a diferencia de lo que ocurre en general) la descodificación de cualquier vector recibido es siempre posible; será correcta cuando el vector recibido contenga un error como máximo e incorrecta en otro caso.*

Ilustremos con un ejemplo como aumenta la fiabilidad de la transmisión mediante el empleo de $\mathcal{H}_2(3)$*. Si, durante la transmisión, la probabilidad de error por bit es de* 0,1 *(suposición únicamente académica y –afortunadamente– muy poco realista), entonces un sencillo cálculo muestra que*

- *la probabilidad de transmisión sin error en 4 bits de información es* 0,6561 *sin codificar y* 0,8503 *codificando;*

- *la probabilidad de transmisión incorrecta no detectada en 4 bits de información es* 0,3439 *sin codificar y* 0,0257 *codificando.*

Claro está que esta ganancia se consigue al precio de enviar un volumen de datos $7/4$ *veces mayor.*

2.2. Códigos lineales

En todo lo que sigue supondremos que el alfabeto usado $\mathcal{A}$ tiene por cardinal, q, la potencia de un número primo e identificaremos $\mathcal{A}$ con $\mathbb{F}_q$ el cuerpo finito con q elementos.

Hemos citado ya que entre los requisitos de un buen código $\mathcal{C}$ está el de poseer algoritmos eficaces de codificación y descodificación. Por lo general, esta condición pasa por que $\mathcal{C}$ posea alguna estructura algebraica. Por ejemplo, ¿posee el código de Hamming alguna estructura algebraica? No hay más que transcribir la condición de que cada uno de los tres círculos contenga un número par de 1 en términos de ecuaciones,

$$\begin{cases} x_1 & +x_2 & +x_3 & & & +x_6 & & \equiv 0 \pmod 2 \\ x_1 & +x_2 & & +x_4 & +x_5 & & & \equiv 0 \pmod 2 \\ x_1 & & +x_3 & +x_4 & & & +x_7 & \equiv 0 \pmod 2. \end{cases}$$

y $\mathcal{H}_2(3)$ es un subespacio vectorial de $\mathbb{F}_2^7$. También la aplicación de codificación es lineal:

$$c(x_1, x_2, x_3, x_4) = (x_1, x_2, x_3, x_4) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

En general, si la aplicación de codificación $c : \mathbb{F}_q^k \to \mathbb{F}_q^n$ es lineal, decimos asimismo que el código $\mathrm{Im}(c)$ es lineal.

Definición 2.5. *Un* código lineal q-ario de longitud n *es un subespacio vectorial* $\mathcal{C} \subseteq \mathbb{F}_q^n$.

Para abreviar, de un código lineal de longitud n, dimensión k y distancia mínima d, diremos que es de tipo $[n, k, d]$. Los códigos utilizados en la práctica (excepto algunos de pequeño tamaño) son siempre lineales. A continuación veremos como los procesos de codificación y descodificación, y el cálculo de la distancia mínima, son mucho más simples para los códigos lineales que para aquellos que no lo son.

2.2.1. Matriz generatriz

Todo subespacio de $\mathbb{F}_q^n$ de dimensión k puede ser interpretado como imagen de una (no única) aplicación lineal inyectiva $c : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$.

Definición 2.6. *Llamaremos* matriz generatriz *de $\mathcal{C}$ a la matriz de una aplicación lineal biyectiva $c : \mathbb{F}_q^k \longrightarrow \mathcal{C} \subset \mathbb{F}_q^n$, es decir, a una matriz $k \times n$ cuyas filas son una base de $\mathcal{C}$.*

Como una base de $\mathcal{C}$ no es única, tampoco lo es una matriz generatriz. Cualquiera de ellas, G, proporciona no solamente un código sino una codificación. En efecto, como $\mathcal{C} = \{\mathbf{a}G \mid \mathbf{a} \in \mathbb{F}_q^k\}$, (escribimos los vectores en forma de filas) un mensaje $\mathbf{a} \in \mathbb{F}_q^k$ se codifica por $\mathbf{a}G \in \mathbb{F}_q^n$. Así la codificación es para los códigos lineales de máxima simplicidad y sólo requiere el almacenamiento en memoria de la matriz G (es decir, de nk elementos de $\mathbb{F}_q$, y no de nq^k como sería el caso de un código en bloque no lineal con el mismo cardinal).

Cuando se codifica una palabra $\mathbf{a} \in \mathbb{F}_q^k$ mediante un código lineal, es a veces interesante que la palabra codificada contenga como subpalabra a $\mathbf{a}$ (podemos suponer que al comienzo de la palabra codificada), es decir sea de la forma $(\mathbf{a}, \mathbf{z})$, $\mathbf{z} \in \mathbb{F}_q^{n-k}$. Así los k primeros símbolos de la palabra contienen la información y los siguientes son de control. Este tipo de codificación es llamado *sistemático*. Evidentemente la codificación es sistemática si y sólo si la matriz G es de la forma $G = (I_k, C)$, donde I_k denota la matriz identidad $k \times k$. Esta forma de G es conocida como *forma estándar*, y el código es llamado *sistemático* si posee alguna matriz generatriz en forma estándar.

Ejemplo 2.7. *Una matriz generatriz estándar del código de Hamming*

$\mathcal{H}_2(3)$ *es*

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

Definición 2.8. *Diremos que dos códigos $\mathcal{C}_1, \mathcal{C}_2$, de la misma longitud, n, sobre $\mathbb{F}_q$, son* equivalentes *si existe una permutación σ del conjunto $\{1, \ldots, n\}$ tal que $\mathcal{C}_2 = \{\sigma(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C}_1\}$.*

Nota 2.9. *Una permutación σ, actúa realmente sobre los índices $\{1, \ldots, n\}$ y no sobre los elementos de $\mathbb{F}_q^n$. Cuando por abuso de notación escribimos $\sigma(\mathbf{x})$, deberíamos escribir $(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$. Denotaremos por $\mathcal{S}_n$ el* grupo simétrico *de orden n, es decir, el grupo de todas las $n!$ permutaciones del conjunto $\{1, \ldots, n\}$.*

Códigos equivalentes tienen los mismos parmetros k y d. Recíprocamente, dado el código $\mathcal{C}$, para cada permutación σ de $\{1, \ldots, n\}$, el conjunto

$$\sigma(\mathcal{C}) = \{\sigma(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C}\}$$

es un código equivalente a $\mathcal{C}$. Eventualmente puede suceder que $\sigma(\mathcal{C}) = \mathcal{C}$. De hecho podemos considerar el conjunto

$$\mathrm{Aut}(\mathcal{C}) = \{\sigma \in \mathcal{S}_n \mid \sigma(\mathcal{C}) = \mathcal{C}\}.$$

$\mathrm{Aut}(\mathcal{C})$ es un subgrupo de $\mathcal{S}_n$, llamado *grupo de automorfismos* de $\mathcal{C}$.

Proposición 2.10. *Todo código es equivalente a uno sistemático.*

2.2.2. Matriz de control

Un subespacio vectorial de $\mathbb{F}_q^n$ puede describirse no sólo mediante un sistema de generadores (lo que da lugar al concepto de matriz generatriz), sino también mediante unas ecuaciones implícitas. Esta forma de caracterización origina la siguiente definición.

Definición 2.11. *Diremos que una matriz H es una* matriz de control *del código $\mathcal{C}$ si para todo vector $\mathbf{x} \in \mathbb{F}_q^n$ se verifica que $\mathbf{x} \in \mathcal{C}$ si y sólo si $H\mathbf{x}^t = \mathbf{0}$.*

Si $\mathcal{C}$ es de tipo $[n,k]$, entonces H es de tamaño $(n-k)\times n$ y rango $n-k$.

Ejemplo 2.12. *Una matriz del control del código de Hamming es*

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

A veces se toma como código de Hamming uno equivalente a éste, disponiendo las columnas de H en orden creciente como representación binaria de los enteros 1,2,$\cdots$,7. Como veremos más adelante, esta disposición es aprovechada en la descodificación.

Proposición 2.13. *Si G y H son matrices generatriz y de control de $\mathcal{C}$, entonces $GH^t=0$.*

Si G es una matriz generatriz de $\mathcal{C}$ dada en forma estándar, $G=(I_k,C)$, entonces es fácil ver que la matriz $H=(-C^t,I_{n-k})$ tiene tamaño $(n-k)\times n$, rango $n-k$ y verifica $GH^t=0$, luego es una matriz de control para $\mathcal{C}$. Diremos que una matriz de control está en forma estándar si es de la forma (B,I_{n-k}).

La distancia mínima de un código puede ser obtenida a partir de su matriz de control. Para poder probar este resultado nos es preciso un nuevo concepto.

Definición 2.14. *Sea $\mathbf{x}=(x_1,\cdots,x_n)\in\mathbb{F}_q^n$. Llamaremos* soporte *de $\mathbf{x}$ al conjunto $sop(\mathbf{x})=\{i\mid 1\le i\le n, x_i\neq 0\}$. Llamaremos* peso de Hamming *de $\mathbf{x}$ a $w(\mathbf{x})=\#sop(\mathbf{x})=d(\mathbf{x},\mathbf{0})$ siendo $\mathbf{0}$ el vector $\mathbf{0}=(0,0,\cdots,0)$.*

La aplicación w, así definida, es una norma en $\mathbb{F}_q^n$ y d es la distancia asociada a esta norma. Análogamente a la distancia mínima de un código, podemos definir su *peso mínimo* como

$$w(\mathcal{C})=\text{mín}\{w(\mathbf{c})\mid \mathbf{c}\in\mathcal{C},\mathbf{c}\neq\mathbf{0}\}. \tag{2.1}$$

Lema 2.15. *En un código lineal, la distancia mínima es igual al peso mínimo.*

Demostración. $d(\mathbf{x},\mathbf{y})=w(\mathbf{x}-\mathbf{y})$. □

Proposición 2.16. *Sea $\mathcal{C}$ un código lineal de matriz de control H y distancia mínima d. Entonces $d > r$ si y sólo si cualesquiera r columnas de H son linealmente independientes. Por tanto, la distancia mínima de $\mathcal{C}$ coincide con el menor cardinal de un conjunto de columnas linealmente dependientes en H.*

Demostración. Cualesquiera r columnas de H son independientes si y sólo si para ningún vector de peso $\leq r$ sucede que $H\mathbf{x}^t = \mathbf{0}$. □

Ejemplo 2.17. *Consideremos el código de Hamming y sea H su matriz de control. Las columnas de H son todos los elementos de $\mathbb{F}_2^3$, excepto $(0,0,0)$. En particular todas son distintas, luego (siendo $\mathbb{F}_2$ el cuerpo base) linealmente independientes dos a dos. Por tanto $d \geq 3$. Como $1110000 \in \mathcal{C}$, concluimos que $d = 3$. Puede comprobarse que $\mathcal{H}_2(3)$ es el código binario de distancia 3 y dimensión 4 con la mayor longitud posible.*

Corolario 2.18 (Cota de Singleton)**.** *La distancia mínima de un código lineal $[n,k]$ verifica $d \leq n - k + 1$.*

Los códigos lineales para los que se alcanza la igualdad en la cota anterior, $d = n-k+1$, son llamados de *máxima distancia de separación* (o MDS) y juegan un papel preponderante tanto a nivel teórico como práctico.

2.2.3. Dualidad

La matriz de control, H, de un código lineal $\mathcal{C}$, puede ser interpretada como matriz generatriz de otro código sobre $\mathbb{F}_q$, llamado *dual* de $\mathcal{C}$ y denotado $\mathcal{C}^\perp$. Obviamente, si $\mathcal{C}$ tiene dimensión k, entonces $\mathcal{C}^\perp$ tiene dimensión $n-k$. Además, si G es una matriz generatriz de $\mathcal{C}$, como la igualdad $GH^t = 0$ implica $HG^t = 0$, se deduce que G es una matriz de control para $\mathcal{C}^\perp$.

Proposición 2.19. *Si $\mathcal{C}$ es un código lineal, entonces su dual $\mathcal{C}^\perp$ es el ortogonal de $\mathcal{C}$ con respecto a la forma bilineal*

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^{n} u_i v_i \in \mathbb{F}_q. \tag{2.2}$$

Demostración. Sean G y H matrices generatriz y de control de $\mathcal{C}$. El resultado es consecuencia de la igualdad $GH^t = 0$, ya que rango(G) + rango$(H) = n$. □

Como la forma bilineal $\langle , \rangle$ es simétrica y no degenerada, se verifica que $(\mathcal{C}^\perp)^\perp = \mathcal{C}$, es decir, el dual del dual de un código es el propio código. Obsérvese que puede darse la situación $\mathcal{C} \cap \mathcal{C}^\perp \neq \{\mathbf{0}\}$. El caso extremo se presenta cuando $\mathcal{C} = \mathcal{C}^\perp$.

Definición 2.20. *Diremos que un código lineal es* autodual *cuando coincide con su código dual.*

A diferencia de lo que ocurre con la dimensión, no es posible, en general, determinar la distancia mínima de $\mathcal{C}^\perp$ únicamente en términos de la distancia mínima de $\mathcal{C}$.

2.2.4. Descodificación de los códigos lineales

En esta sección vamos a exponer un método general de descodificación para códigos lineales. Sean $\mathcal{C}$ un código lineal $[n, k, d]$ sobre $\mathbb{F}_q$ y H una matriz de control. Como sabemos $\mathcal{C}$ corrige $t = \lfloor \frac{d-1}{2} \rfloor$ errores. Supongamos enviada una palabra $\mathbf{c} \in \mathcal{C}$ y recibido un vector $\mathbf{y} \in \mathbb{F}_q^n$. El error cometido durante la transmisión ha sido $\mathbf{e} = \mathbf{y} - \mathbf{c}$. La estrategia que seguiremos para descodificar $\mathbf{y}$ es simple (en esencia la misma que en 2.1.2): calculamos la distancia de $\mathbf{y}$ a todas las palabras de $\mathcal{C}$ y la descodificamos por la más próxima (si existe). Si durante la transmisión se han cometido a lo más t errores (es decir, si $w(\mathbf{e}) \leq t$), entonces $d(\mathbf{c}, \mathbf{y}) = w(\mathbf{e}) \leq t$ y $\mathbf{c}$ es la única palabra del código con tal propiedad; la descodificación es por tanto correcta. Si $t < w(\mathbf{e}) < d$, podemos detectar que se han producido errores (puesto que $\mathbf{y} \notin \mathcal{C}$), pero no corregirlos en general. Si $w(\mathbf{e}) \geq d$ la descodificación fallará eventualmente.

Llevar a cabo este proceso es mucho más simple y computacionalmente económico para los códigos lineales, debido a su estructura algebraica.

Definición 2.21. *Llamaremos* síndrome *de* $\mathbf{y}$ *al vector*

$$s(\mathbf{y}) = H\mathbf{y}^t \in \mathbb{F}_q^{n-k}. \tag{2.3}$$

Notemos que $\mathbf{y} \in \mathcal{C}$ si y sólo si $s(\mathbf{y}) = \mathbf{0}$. Por tanto, al ser el síndrome una aplicación lineal, $s(\mathbf{y}) = s(\mathbf{c} + \mathbf{e}) = s(\mathbf{c}) + s(\mathbf{e}) = s(\mathbf{e})$. Así, recibido $\mathbf{y}$, conocemos inmediatamente el síndrome del error cometido.

Proposición 2.22. *El síndrome del vector recibido* $\mathbf{y}$ *es una combinación lineal de las columnas de* H *correspondientes a las posiciones de error.*

Para ver en que modo puede ayudarnos el síndrome a detectar y corregir errores examinemos un caso simple. Supongamos que $\mathcal{C}$ corrige al menos un error y que durante la transmisión ha ocurrido un único error (es decir, que $w(\mathbf{e}) = 1$, pongamos $\mathbf{e} = (0, \ldots, 0, e_i, 0, \ldots, 0)$). Por ser $d \geq 3$, cualesquiera dos columnas de H son linealmente independientes, es decir ninguna columna de H es múltiplo de ninguna otra. Así el síndrome del vector recibido $\mathbf{y}$ será múltiplo de una y sólo una columna de H. Según la proposición anterior, la posición de esa columna es precisamente la posición en que se ha cometido el error, es decir, si $\mathbf{h}_i$ es la i-sima columna de H, $s(\mathbf{y}) = e_i\mathbf{h}_i$, de donde pueden deducirse inmediatamente $\mathbf{e}$ y el mensaje enviado $\mathbf{c} = \mathbf{y} - \mathbf{e}$.

Ejemplo 2.23. *Utilizando el código de Hamming binario* $\mathcal{H}_2(3)$ *de matriz de control*

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

se envía el mensaje $\mathbf{c} = 0010110$*. Durante la transmisión ocurre un error, de manera que se recibe* $\mathbf{y} = 0011110$*. El síndrome del vector recibido es*

$$H(0011110)^t = (100)^t.$$

Como 100 *es la representación binaria de 4, el error ha ocurrido en la cuarta posición, luego* $\mathbf{y}$ *se descodifica por* 0010110*, que era el mensaje enviado.*

Consideremos en $\mathbb{F}_q^n$ la relación de equivalencia: $\mathbf{u} \sim \mathbf{v}$ si y sólo si $\mathbf{u} - \mathbf{v} \in \mathcal{C}$. El espacio vectorial cociente obtenido, módulo tal relación, se denota por $\mathbb{F}_q^n/\mathcal{C}$. Los elementos de $\mathbb{F}_q^n/\mathcal{C}$ son clases de equivalencia $\mathbf{u} + \mathcal{C} = \{\mathbf{u} + \mathbf{x} \mid \mathbf{x} \in \mathcal{C}\}$. Como cada clase posee $\#\mathcal{C} = q^k$ elementos (o *representantes*), el cardinal de $\mathbb{F}_q^n/\mathcal{C}$ es q^{n-k} y su dimensión es $n - k$. Nótese que $\mathbf{u} - \mathbf{v} \in \mathcal{C}$ si y sólo si $s(\mathbf{u}) = s(\mathbf{v})$, luego recibido $\mathbf{y}$, al calcular $s(\mathbf{y})$ conocemos la clase a la que pertenece el error.

Definición 2.24. *Si en una clase existe un único elemento de peso mínimo, éste recibe el nombre de* líder *de la clase.*

Algunos autores exigen, además, que el peso de tal elemento sea $\leq t$ (siendo t la capacidad de corrección del código) para darle el nombre de líder. En cualquier caso, en general no toda clase tendrá líder ya que el elemento de peso mínimo no será, en general, único. Sin embargo, si una clase contiene un elemento de peso $\leq t$, éste es el líder de la clase.

Proposición 2.25. *Cada clase de $\mathbb{F}_q^n/\mathcal{C}$ posee a lo más un elemento de peso $\leq t$.*

Demostración. Si existen $\mathbf{u}, \mathbf{v}$ en la misma clase, ambos de peso $\leq t$, entonces $\mathbf{u} - \mathbf{v} \in \mathcal{C}$ y $w(\mathbf{u} - \mathbf{v}) \leq w(\mathbf{u}) + w(\mathbf{v}) \leq 2t < d(\mathcal{C})$, lo cual implica que $\mathbf{u} = \mathbf{v}$. □

Algoritmo del líder

Recibido un vector $\mathbf{y}$, como todos los vectores $\mathbf{y} - \mathbf{x}$, $\mathbf{x} \in \mathcal{C}$, están en la misma clase de $\mathbb{F}_q^n/\mathcal{C}$, que es la de $\mathbf{y}$, el mínimo de $d(\mathbf{y}, \mathbf{x}) = w(\mathbf{y} - \mathbf{x})$ se obtiene cuando $\mathbf{y} - \mathbf{x}$ es el líder de la clase. Por tanto la descodificación es posible si y sólo si la clase del vector recibido posee líder, y el error es asumido como el líder de la clase. La proposición 2.25 garantiza que si el número de errores no supera la capacidad correctora del código, entonces la descodificación es correcta.

Para llevar a cabo este proceso, construimos una tabla con dos columnas y tantas filas como clases hay en $\mathbb{F}_q^n/\mathcal{C}$ (es decir, q^{n-k} filas). En la primera columna escribimos el síndrome de un elemento cualquiera de cada una de las clases; en la segunda el líder de la clase correspondiente (si existe). Esta tabla se construye de una vez por todas y sirve para la descodificación de cualquier vector. Ahora, recibido $\mathbf{y}$, hacemos

Algoritmo 2.26. *Recibido un vector $\mathbf{y}$,*

1. *calcular $s(\mathbf{y})$ y buscarlo en la columna de síndromes*
2. *si la clase correspondiente no posee líder, la descodificación falla. Fin.*
3. *Si la clase posee líder, $\mathbf{e}$, se decide que $\mathbf{e}$ es el error cometido. La palabra descodificada es $\mathbf{y} - \mathbf{e}$. Fin.*

Ejemplo 2.27. *El código binario de matriz generatriz*

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

tiene distancia mínima 5, luego corrige dos errores. La tabla de síndromes y líderes asociada a este código es la dada en la página siguiente.

Supongamos recibido el mensaje 11011011. *A su síndrome,* 111000^t, *según la tabla le corresponde el líder* 10000100. *El mensaje descodificado es, por tanto,* $01011111 = 11011011 + 10000100$. *Como el líder tiene peso 2, podemos estar seguros de que la decodificación es correcta si han ocurrido a lo más dos errores.*

Supongamos recibido el mensaje 01110010. *Su síndrome es* 101101^t *y el líder de esta clase es* 10010001. *Aunque el líder tiene peso mayor que la capacidad teórica de corrección del código, es posible descodificar el mensaje recibido por* 11100011. *En cualquier caso sabemos que han ocurrido al menos tres errores durante la transmisión, y que nuestra descodificación es correcta si y sólo si han ocurrido exáctamente tres errores.*

Supongamos recibido 01011000. *Su síndrome es* 000111^t. *Como esta clase no tiene líder, no es posible la descodificación. Sabemos además que han ocurido al menos tres errores.*

síndrome	líder	síndrome	líder
000000	00000000	100000	00100000
000001	00000001	100001	00100001
000010	00000010	100010	00100010
000011	00000011	100011	11000000
000100	00000100	100100	00100100
000101	00000101	100101	00100101
000110	00000110	100110	00100110
000111		100111	11000100
001000	00001000	101000	00101000
001001	00001001	101001	00101001
001010	00001010	101010	00101010
001011		101011	11001000
001100	00001100	101100	10010000
001101		101101	10010001
001110		101110	10010010
001111	01010000	101111	01110000
010000	00010000	110000	00110000
010001	00010001	110001	00110001
010010	00010010	110010	00110010
010011		110011	11010000
010100	00010100	110100	10001000
010101		110101	10001001
010110		110110	10001010
010111	01001000	110111	01101000
011000	00011000	111000	10000100
011001		111001	10000101
011010		111010	10000110
011011	01000100	111011	01100100
011100	10100000	111100	10000000
011101	01000010	111101	10000001
011110	01000001	111110	10000010
011111	01000000	111111	01100000

2.3. Códigos cíclicos

Los códigos cíclicos constituyen la familia más ampliamente utilizada de códigos correctores de errores. Para su estudio alteraremos ligeramente las notaciones que venimos utilizando, y escribiremos las coordenadas de los vectores desde 0 a $n-1$. Así pondremos $\mathbf{x} = (x_0, \ldots, x_{n-1})$ en lugar de $(x_1, \ldots, x_n)$. Enseguida podrá apreciarse la utilidad de este

cambio de notación.

2.3.1. Noción de código cíclico

Definición 2.28. *Un código lineal $\mathcal{C}$ de longitud n sobre $\mathbb{F}_q$, es* cíclico *si para cada $(c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$ se verifica que $(c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}$.*

En otros términos, se exige a $\mathcal{C}$ ser invariante por permutaciones cíclicas.

Sean $\mathbb{F}_q[X]_{(n-1)}$ el espacio vectorial de los polinomios sobre $\mathbb{F}_q$ con grado menor que n y A el anillo cociente $A = \mathbb{F}_q[X]/\langle X^n - 1\rangle$. En virtud de los isomorfismos de espacios vectoriales

$$\mathbb{F}_q^n \cong \mathbb{F}_q[X]_{(n-1)} \cong A \tag{2.4}$$

podemos identificar cada vector $(a_0, \ldots, a_{n-1})$ con el polinomio $a_0 + a_1X + \cdots + a_{n-1}X^{n-1}$ y con la clase, en A, $a_0 + a_1X + \cdots + a_{n-1}X^{n-1} + \langle X^n - 1\rangle$. Consecuentemente, un código sobre $\mathbb{F}_q$ puede considerarse como un subconjunto de A. En lo que sigue utilizaremos libremente estas identificaciones. Por otra parte, impondremos la restricción adicional $\mathrm{mcd}(q, n) = 1$. Esto garantiza que el polinomio $X^n - 1$ tiene todos sus factores irreducibles distintos y sus raíces forman un grupo cíclico de orden n. La propiedad fundamental de los códigos cíclicos es la siguiente.

Teorema 2.29. *Sea $\mathcal{C}$ un código lineal no nulo de longitud n sobre el cuerpo finito $\mathbb{F}_q$. $\mathcal{C}$ es cíclico si y sólo si, considerado inmerso en A, es un ideal.*

Demostración. Supongamos que $\mathcal{C}$ es cíclico. Puesto que $\mathcal{C}$ es ya un subgrupo abeliano de A, basta probar que si $c(X) \in \mathcal{C}$ entonces $Xc(X) \in \mathcal{C}$. Ahora bien

$$X(c_0 + c_1X + \cdots + c_{n-1}X^{n-1}) = c_{n-1} + c_0X + \cdots + c_{n-2}X^{n-1}$$

y el hecho de que este último polinomio pertenezca al código no es sino la definición de código cíclico interpretada en lenguaje polinómico. El recíproco se demuestra de manera idéntica. □

Es conocido que todo ideal del anillo A es principal, es decir, consiste en el conjunto de múltiplos de un polinomio $g(X)$ divisor de $X^n - 1$.

Corolario 2.30. *Dado un código cíclico no nulo $\mathcal{C}$ de longitud n, existe un único polinomio mónico $g(X) \in \mathbb{F}_q[X]$ divisor de $X^n - 1$, tal que $\mathcal{C} = \langle g(X) \rangle$. En consecuencia, los elementos de $\mathcal{C}$ pueden identificarse con los polinomios de grado menor que n múltiplos de $g(X)$.*

2.3.2. Matrices generatriz y de control

Proposición 2.31. *Sea $\mathcal{C}$ un código cíclico de longitud n sobre $\mathbb{F}_q$ con polinomio generador $g(X)$ de grado $n-k$. El conjunto*

$$\{g(X), Xg(X), \ldots, X^{k-1}g(X)\} \tag{2.5}$$

es una base de $\mathcal{C}$. En particular, $\mathcal{C}$ tiene dimensión k.

Demostración. Basta probar la primera afirmación. Tomemos $f(X)g(X) \in \mathcal{C}$. En virtud del corolario 2.30, podemos suponer $\deg f(X) < k$ (en caso contrario puede encontrarse en A un polinomio $f'(X)$ con $\deg f'(X) < k$ y $f(X)g(X) = f'(X)g(X)$). Sea pues $f(X) = a_0 + a_1X + \cdots + a_{k-1}X^{k-1}$. La escritura

$$f(X)g(X) = a_0g(X) + a_1Xg(X) + \cdots + a_{k-1}X^{k-1}g(X) \tag{2.6}$$

es la combinación lineal buscada, luego $\{g(X), Xg(X), \ldots, X^{k-1}g(X)\}$ es un sistema de generadores. Veamos que es un conjunto libre. Una relación de dependencia lineal

$$b_0g(X) + b_1Xg(X) + \cdots + b_{k-1}X^{k-1}g(X) = 0 \tag{2.7}$$

puede escribirse también $b(X)g(X) = 0$, siendo $b(X) = b_0 + b_1X + \cdots + b_{k-1}X^{k-1}$. Como $\deg(b(X)g(X)) < n$ y $g(X) \neq 0$, se verifica (en A) que $b(X) = 0$, luego $b_i = 0$ para $i = 0, 1, \ldots, k-1$. □

Corolario 2.32. *Un codigo cíclico de longitud n y polinomio generador $g(X) = g_0 + g_1X + \cdots + g_{n-k}X^{n-k}$ tiene matriz generatriz*

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & & & & & \\ & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & & & & \\ & & \ddots & \ddots & & \ddots & \ddots & & & \\ & & & & \ddots & \ddots & & \ddots & \ddots & \\ & & & & & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} \end{pmatrix}.$$

La codificación con un código cíclico $\mathcal{C}$ de parámetros $[n, k]$, puede hacerse del modo usual –a partir de cualquiera de las matrices generatrices descritas anteriormente– o utilizando la notación polinómica. Con esta notación el mensaje sin codificar puede identificarse con un polinomio $a(X)$ de grado menor que k y su codificación es simplemente

$$g(X)a(X) \in \mathcal{C} \tag{2.8}$$

siendo $g(X)$ el polinomio (de grado $n-k$) generador de $\mathcal{C}$. Si se desea que la codificación sea sistemática, entonces realizamos la división euclídea

$$X^{n-k}a(X) = g(X)q(X) + r(X) \tag{2.9}$$

con $\deg r(X) < \deg g(X) = n - k$. El mensaje $a(X)$ se codifica por

$$X^{n-k}a(X) - r(X) \in \mathcal{C}. \tag{2.10}$$

Nótese que la codificación es sistemática en las últimas k posiciones. Vamos ya con la matriz de control.

Definición 2.33. *Si $\mathcal{C}$ es un código cíclico de longitud n, con polinomio generador $g(X)$ de grado $n-k$, llamaremos* polinomio de control *de $\mathcal{C}$ a*

$$h(X) = \frac{X^n - 1}{g(X)} = h_0 + h_1 X + \cdots + h_k X^k. \tag{2.11}$$

Proposición 2.34. *Con las notaciones de la definición anterior, la matriz (de tamaño $(n-k) \times n$)*

$$H = \begin{pmatrix} & & & & & h_k & h_{k-1} & \dots & h_1 & h_0 \\ & & & & h_k & h_{k-1} & h_{k-2} & \dots & h_0 & \\ & & & . & . & . & . & . & & \\ & & & & & & & & & \\ & & . & . & . & . & & & & \\ h_k & h_{k-1} & \dots & h_1 & h_0 & & & & & \end{pmatrix}$$

es una matriz de control de $\mathcal{C}$.

Demostración. Basta con probar la identidad $GH^t = 0$. Para todo $1 \leq i \leq k, 1 \leq j \leq n-k$, el elemento (i, j) de la matriz producto GH^t, es el coeficiente de $X^{n-i-j+1}$ en el polinomio $g(X)h(X) = X^n - 1$. □

Teniendo en cuenta la definición 2.33, resulta evidente que $h(X)$ es un generador de $\mathcal{C}^\perp$. Esto demuestra que el dual de un código cíclico es también cíclico.

2.3.3. Ceros de un código cíclico

Sea $X^n - 1 = f_1(X)f_2(X)\cdots f_m(X)$ la descomposición de $X^n - 1$ en factores irreducibles y sea α_i una raíz de $f_i(X)$. Para el código cíclico $\mathcal{C}_i$ engendrado por $f_i(X)$, se verifica $\mathcal{C}_i = \langle f_i(X)\rangle = \{c(X) \in A \mid c(\alpha_i) = 0\}$. En general, para el código cíclico $\mathcal{C}$ engendrado por $g(X) = f_{i_1} f_{i_2} \cdots f_{i_r}$, se tendrá

$$\mathcal{C} = \langle g(X)\rangle = \{c(X) \mid c(\alpha_{i_1}) = c(\alpha_{i_2}) = \cdots = c(\alpha_{i_r}) = 0\}, \qquad (2.12)$$

lo que muestra que los códigos cíclicos pueden definirse, alternativamente, como conjuntos de polinomios con ciertas raíces n-ésimas de 1 como ceros. Esto permite invertir el proceso: en lugar de partir del polinomio generador $g(X)$ y tomar los ceros adecuados (uno en cada factor irreducible de $g(X)$), podemos tomar, a priori, un conjunto de elementos $\{\alpha_1, \ldots, \alpha_r\}$ en una extensión finita $\mathbb{F}_{q^t}$ de $\mathbb{F}_q$ y definir

$$\mathcal{C} = \{c(X) \in A \mid c(\alpha_1) = \cdots = c(\alpha_r) = 0\}. \qquad (2.13)$$

Tal código es automáticamente cíclico, pues si $f_i(X)$ es el polinomio irreducible de α_i, se verifica que $\mathcal{C} = \langle g(X)\rangle = \langle \text{mcm}(f_1, \ldots, f_r)\rangle$. Si n_i es el orden de α_i en $\mathbb{F}_{q^t}^*$ y $n = \text{mcm}\{n_1, \ldots, n_r\}$, es claro que $g(X)|X^n - 1$ y por tanto $\mathcal{C}$ es un código cíclico de longitud n.

Con esta caracterización de los códigos cíclicos puede comprobarse fácilmente si una palabra recibida está o no en el código. Para ello consideramos la matriz

$$H' = \begin{pmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha_r & \cdots & \alpha_r^{n-1} \end{pmatrix}. \qquad (2.14)$$

Si, para un polinomio $f(X) = f_0 + f_1 X + \cdots + f_{n-1}X^{n-1}$, convenimos en considerar –forzando las notaciones– que

$$H'f(X) = (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_r)) \qquad (2.15)$$

entonces $f(X) \in \mathcal{C}$ si y sólo si $H'f(X) = 0$, lo que autoriza a considerar a H' como un tipo de matriz de control del código. Nótese, sin embargo, que H' no tiene ni coeficientes en $\mathbb{F}_q$ ni dimensiones $(n-k) \times n$, por lo que no es una matriz de control en sentido estricto.

A continuación veremos como los códigos de Hamming pueden obtenerse de esta forma.

Ejemplo 2.35. *Si* $\mathrm{mcd}(q-1, r) = 1$*, entonces el código de Hamming q-ario* $\mathcal{H}_q(r)$ *es equivalente a un código cíclico. En particular, todo código de Hamming binario es equivalente a un código cíclico. En efecto,* $\mathcal{H}_2(r)$ *tiene longitud* $n = 2^r - 1$ *y dimensión* $k = 2^r - r - 1$*. Sea* α *un elemento primitivo de* $\mathbb{F}_{2^r}$ *(una raíz primitiva n-ésima de la unidad). Puesto que los elementos de* $\mathbb{F}_{2^r}$ *son las potencias de* α*, una matriz de control de* $\mathcal{H}_2(r)$ *es la matriz*

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \end{pmatrix}$$

(identificando cada α^i *con el vector columna de sus coordenadas en la base* $1, \alpha, \ldots, \alpha^{r-1}$*). Por tanto* $\mathcal{H}_2(r)$ *puede identificarse con el conjunto de los polinomios que tienen a* α *por raíz o, dicho de otra forma, con el código cíclico generado por el polinomio* $Irr(\alpha, \mathbb{F}_2)$*.*

2.3.4. Descodificación de los códigos cíclicos

El proceso general de descodificación de los códigos cíclicos sigue el esquema síndrome-líder válido para cualquier código lineal. Como sabemos, el mayor inconveniente de la descodificación mediante síndromes y líderes es el enorme tamaño de la tabla necesaria para llevarlo a cabo. Sin embargo, la estructura cíclica permite una notable reducción de esta tabla. En efecto, si $\mathcal{C}$ es cíclico, basta corregir los errores producidos en una posición determinada (fija) de los vectores recibidos. Generalmente esta posición se toma la última (es decir, la de coordenada $n-1$).

Construimos una tabla *reducida* de síndromes y líderes que contenga unicamente las entradas correspondientes a líderes con coordenada $n-1$ no nula. Recibido un vector $\mathbf{y} = (y_0, \ldots, y_{n-1})$, la tabla reducida permite descodificar la última coordenada y_{n-1} de $\mathbf{y}$ de la forma usual: se calcula el síndrome $s(\mathbf{y})$; si aparece en la tabla se corrige $\mathbf{y}$ mediante el líder correspondiente. En otro caso se asume y_{n-1} correcto. Una vez terminado el proceso con y_{n-1}, se comienza con y_{n-2}. Para ello se considera el vector $\mathbf{y}^{(1)}$, obtenido de $\mathbf{y}$ permutando cíclicamente sus coordenadas, $\mathbf{y}^{(1)} = (y_{n-1}, y_0, \ldots, y_{n-2})$. Si $\mathbf{y} = \mathbf{c} + \mathbf{e}$, entonces $\mathbf{y}^{(1)} = \mathbf{c}^{(1)} + \mathbf{e}^{(1)}$, con lo que la tabla reducida permite la descodificación de la última coordenada de $\mathbf{y}^{(1)}$, que es y_{n-2}. Iterando el proceso n veces con los vectores $\mathbf{y}, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n-1)}$, se obtiene la descodificación completa de $\mathbf{y}$.

Ejemplo 2.36. *Trabajando con el código (cíclico como se sabe) binario de Hamming* $\mathcal{H}_2(3)$*, de matriz de control*

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

las tablas completa y reducida de síndromes y líderes son

síndrome	*líder*
000	*0000000*
001	*1000000*
010	*0100000*
011	*0010000*
100	*0001000*
101	*0000100*
110	*0000010*
111	*0000001*

y

síndrome	*líder*
111	*0000001*

Supongamos emitida la palabra 1001100 *y recibido el vector* $\mathbf{y} = 1011100$*. El proceso de descodificación sigue las siete iteraciones siguientes:*

1. $s(\mathbf{y}) = s(1011100) = 011$*, que no aparece en la tabla reducida. La última coordenada, 0, de* $\mathbf{y}$ *es correcta.*
2. $s(\mathbf{y}^{(1)}) = s(0101110) = 100$*, que no aparece en la tabla reducida. La última coordenada, 0, de* $\mathbf{y}^{(1)}$ *es correcta.*
3. $s(\mathbf{y}^{(2)}) = s(0010111) = 101$*, que no aparece en la tabla reducida. La última coordenada, 1, de* $\mathbf{y}^{(2)}$ *es correcta.*
4. $s(\mathbf{y}^{(3)}) = s(1001011) = 110$*, que no aparece en la tabla reducida. La última coordenada, 1, de* $\mathbf{y}^{(3)}$ *es correcta.*
5. $s(\mathbf{y}^{(4)}) = s(1100101) = 111$*, que sí aparece en la tabla reducida. La última coordenada, 1, de* $\mathbf{y}^{(4)}$ *es incorrecta.*
6. $s(\mathbf{y}^{(5)}) = s(1110010) = 001$*, que no aparece en la tabla reducida. La última coordenada, 0, de* $\mathbf{y}^{(5)}$ *es correcta.*
7. $s(\mathbf{y}^{(6)}) = s(0111001) = 010$*, que no aparece en la tabla reducida. La última coordenada, 1, de* $\mathbf{y}^{(6)}$ *es correcta.*

El mensaje se descodifica por 1001100 *que era la palabra enviada.*

Este método de descodificación, debido a J. Meggitt, permite dividir por n el tamaño de la tabla necesaria, si bien requiere el cálculo de n veces más síndromes.

2.3.5. Captura del error

Trabajando con códigos cíclicos es interesante manejar los vectores en forma de polinomios.

Definición 2.37. *Sea $\mathcal{C}$ un código cíclico generado por el polinomio $g(X)$. Recibido un vector $\mathbf{y}$, llamaremos* polinomio síndrome de $\mathbf{y}$*, y lo representaremos $s[\mathbf{y}](X)$, al resto de la división euclídea de $y(X)$ entre $g(X)$.*

Así pues, recibido $\mathbf{y}$, se tiene la escritura $y(X) = g(X)q(X)+s[\mathbf{y}](X)$, con $\deg s[\mathbf{y}](X) < \deg g(X)$. Si el vector enviado es $\mathbf{c}$, entonces $\mathbf{y} = \mathbf{c}+\mathbf{e}$. En algunos casos, el polinomio síndrome proporciona inmediatamente el error $\mathbf{e}$.

Proposición 2.38. *Sea $\mathcal{C}$ un código cíclico que corrige t errores. Si, enviada la palabra $\mathbf{c}$ y recibida $\mathbf{y} = \mathbf{c}+\mathbf{e}$ han ocurrido a lo más t errores y si el polinomio síndrome de $\mathbf{y}$, $s[\mathbf{y}](X)$ tiene peso t a lo más, entonces $e(X) = s[\mathbf{y}](X)$.*

Demostración. Como $s[\mathbf{y}](X) = y(X) - g(X)q(X) = c(X) + e(X) - g(X)q(X)$, en cualquier caso $s[\mathbf{y}](X) - e(X) = c(X) - g(X)q(X) \in \mathcal{C}$. Si $s[\mathbf{y}](X)$ tiene peso t a lo más y han ocurrido $\leq t$ errores, entonces $s[\mathbf{y}](X)-e(X)$ tiene peso a lo más $2t < d$, luego $s[\mathbf{y}](X)-e(X) = 0$. □

Por supuesto este método de corrección de errores requiere que el síndrome $s[\mathbf{y}](X)$ tenga peso $\leq t$, lo que no siempre sucede. Sin embargo, aún en el caso de no verificar $\mathbf{y}$ esta condición, puede que sí la verifique alguna de sus permutaciones cíclicas, $\mathbf{y}^{(j)}$. En este caso, como

$$y^{(j)}(X) = c^{(j)}(X) + e^{(j)}(X) \tag{2.16}$$

y la proposición anterior garantiza que $e^{(j)}(X) = s[\mathbf{y}^{(j)}](X)$, podemos recuperar el error como

$$e(X) = s[\mathbf{y}^{(j)}]^{(n-j)}(X). \tag{2.17}$$

Este método de descodificación es conocido como de *captura del error*.

2.3.6. Errores a ráfagas

Definición 2.39. *Una* ráfaga *es un vector* $\mathbf{x} \in \mathbb{F}_q^n$ *tal que todas sus coordenadas no nulas son consecutivas. Se llama* longitud *de la ráfaga a* $w(\mathbf{x})$.

Los códigos cíclicos son particularmente eficientes en la detección de los errores a ráfagas.

Proposición 2.40. *Un código cíclico* $\mathcal{C}$ *de parámetros* $[n, k]$ *no contiene ninguna ráfaga de longitud* $l \leq n-k$*, luego detecta cualquier error ráfaga de longitud* $l \leq n-k$.

Demostración. Una ráfaga de longitud l corresponde a un polinomio de la forma $X^i l(X)$ con $\deg l(X) < l$. En particular $l(X) \notin \mathcal{C}$ puesto que $\deg l(X) < \deg g(X)$ y $l(X)$ no puede ser múltiplo de $g(X)$. Por tanto $X^i l(X) \notin \mathcal{C}$. Para la segunda afirmación, si $\mathbf{e}$ es una ráfaga de longitud $l \leq n-k$, entonces $\mathbf{c}+\mathbf{e} \notin \mathcal{C}$ (pues en otro caso $\mathbf{e} \in \mathcal{C}$), luego puede detectarse el error. □

Si un código lineal de parámetros $[n, k]$ puede detectar ráfagas de longitud l, es fácil probar que $l \leq n-k$. Por tanto, la capacidad de detección de errores a ráfagas es óptima en los códigos cíclicos. La descodificación de las ráfagas puede hacerse mediante la captura de errores, ya que se verifica el siguiente resultado.

Proposición 2.41. *Sea* $\mathcal{C}$ *un código cíclico de parámetros* $[n, k]$*. Si los errores de un vector recibido* $\mathbf{y}$ *constituyen una ráfaga de longitud a lo más* $n-k$*, entonces existe* j *tal que* $e^{(j)}(X) = s[\mathbf{y}^{(j)}](X)$.

Demostración. Bajo las condiciones indicadas, algún $\mathbf{y}^{(j)}$ tiene errores únicamente en las coordenadas $0, \ldots, n-k-1$, con lo que $\deg e^{(j)}(X) < n-k$. Ahora bien, por la unicidad de la división euclídea

$$s[\mathbf{y}^{(j)}](X) = s[\mathbf{e}^{(j)}](X) = e^{(j)}(X) \tag{2.18}$$

y se obtiene el resultado. □

Otra técnica usada para combatir los errores a ráfagas es el *intercalado* (ó *'interleaving'*). Dado un código $\mathcal{C}$ y dadas palabras $\mathbf{c}^1, \ldots, \mathbf{c}^m$ de $\mathcal{C}$, podemos construir por intercalado la palabra

$$(c_0^1, c_0^2, \ldots, c_0^m; c_1^1, \ldots, c_1^m; \ldots; c_{n-1}^1, \ldots, c_{n-1}^m).$$

Se denota por $\mathcal{C}^{(m)}$ el código obtenido intercalando todas las posibles elecciones de m palabras de $\mathcal{C}$.

Proposición 2.42. *Si $\mathcal{C}$ es un código cíclico de parámetros $[n,k]$ y polinomio generador $g(X)$, entonces el código intercalado $\mathcal{C}^{(m)}$ tiene polinomio generador $g(X^m)$ y parámetros $[mn,mk]$, luego detecta ráfagas de longitud $m(n-k)$.*

2.4. Códigos BCH y RS

Los códigos BCH (denominados así en honor de sus descubridores, Bose, Chaudhuri y Hocquenghem) constituyen la más importante familia de códigos cíclicos. Se ha visto en la sección anterior, que los códigos cíclicos pueden determinarse prescribiendo un conjunto de elementos como ceros de su polinomio generador. En concreto, si tomamos como ceros los elementos $\alpha_1,\ldots,\alpha_r$, entonces la matriz

$$H' = \begin{pmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{n-1} \\ \vdots & \vdots & & \cdots \\ 1 & \alpha_r & \cdots & \alpha_r^{n-1} \end{pmatrix} \tag{2.19}$$

se comporta como una matriz de control del código obtenido, $\mathcal{C}$. En particular, la distancia mínima de $\mathcal{C}$ es $\geq d$ si cualesquiera $d-1$ columnas de H' son linealmente independientes. No es fácil, en general, determinar este número para una elección arbitraria de los α_i. Una excepción la constituye el caso en que los α_i son potencias consecutivas de una raíz primitiva n-ésima de la unidad, $\alpha_i = \alpha^i$, $i = 1,\ldots,r < n$, pues entonces todo menor de la correspondiente matriz H' se reduce a un determinante de tipo Vandermonde y $d(\mathcal{C}) \geq r+1$.

2.4.1. Construcción y parámetros

Fijemos un cuerpo $\mathbb{F}_q$ y números naturales n, b y δ, $2 \leq \delta \leq n$. Sean m el orden multiplicativo de q módulo n (es decir, el menor número natural tal que $q^m \equiv 1 \pmod{n}$) y $\alpha \in \mathbb{F}_{q^m}$ una raíz primitiva n-ésima de la unidad.

Definición 2.43. *Llamaremos* código BCH *de longitud n y* distancia mínima prevista δ*, al código cíclico de longitud n cuyo polinomio generador tiene por raíces $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-2}$.*

Si se toma $b = 1$, el código se denomina BCH *en sentido estricto*. Si la longitud n es de la forma $n = q^m - 1$, entonces se habla de códigos BCH *primitivos* (en este caso el exponente m coincide con el orden multiplicativo de q módulo n y α es un elemento primitivo de $\mathbb{F}_{q^m}$); si, además, $m = 1$, (es decir, $n = q - 1$ y por tanto $\alpha \in \mathbb{F}_q$) el código se denomina *Reed-Solomon*. Los códigos Reed-Solomon son importantes por derecho propio y volveremos sobre ellos más adelante.

Proposición 2.44. *Un código BCH de distancia prevista δ, posee distancia mínima $d \geq \delta$.*

Demostración. Cualquier menor $(\delta - 1) \times (\delta - 1)$ de la matriz

$$H' = \begin{pmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{n-1} \\ \vdots & \vdots & & \cdots \\ 1 & \alpha_r & \cdots & \alpha_r^{n-1} \end{pmatrix}$$

se reduce a un determinante de tipo Vandermonde, luego cualesquiera $\delta - 1$ de sus columnas son linealmente independientes. □

Dado que, en general, no es factible conocer la auténtica distancia de un código BCH, en la práctica se utiliza δ como un sustituto de la misma. Claro está que la distancia mínima real puede ser mayor que la prevista (como veremos a continuación en los ejemplos).

Un polinomio generador del código puede obtenerse del modo siguiente: para $i = b, \ldots, b + \delta - 2$, sea $m_i(X)$ el polinomio irreducible de α^i sobre $\mathbb{F}_q$. Entonces

$$g(X) = \text{ mcm}\{m_b(X), \ldots, m_{b+\delta-2}(X)\} \tag{2.20}$$

es el polinomio generador buscado. La dimensión del código es, como en todos los códigos cíclicos, $n - \deg g(X)$.

Ejemplo 2.45. *Por simplicidad trabajaremos con códigos BCH binarios, primitivos y en sentido estricto. Así, sean $m \in \mathbb{Z}$, $n = 2^m - 1$ y $\alpha \in \mathbb{F}_{2^m}$ una raíz primitiva n-ésima de la unidad (es decir, un elemento primitivo de $\mathbb{F}_{2^m}$). El caso más simple de código BCH se da para $\delta = 2$, obteniéndose*

$$\mathcal{C}_2 = \{c(X) \in A = \mathbb{F}_2[X]/\langle X^n - 1\rangle \mid c(\alpha) = 0\}.$$

Como $c(X) \in \mathbb{F}_2[X]$, *si* $c(\beta) = 0$, *entonces* $c(\beta^2) = 0$. *Por tanto*

$$\mathcal{C}_2 = \mathcal{C}_3 = \{c(X) \in A \mid c(\alpha) = c(\alpha^2) = 0\}.$$

Como ya sabemos, $\mathcal{C}_2 = \mathcal{C}_3$ *es un código de Hamming, de distancia mínima* 3.
El siguiente caso a considerar es $\delta = 4$. *El código que se obtiene es*

$$\begin{aligned} \mathcal{C}_4 &= \{c(X) \in A \mid c(\alpha) = c(\alpha^2) = c(\alpha^3) = 0\} \\ &= \{c(X) \in A \mid c(\alpha) = c(\alpha^3) = 0\}. \end{aligned}$$

Como en el caso anterior, los polinomios de $\mathcal{C}_4$ *verifican automáticamente la condición* $c(\alpha^4) = 0$, *luego* $\mathcal{C}_4 = \mathcal{C}_5$ *y este código tiene distancia mínima* ≥ 5. *Una matriz de control es*

$$H' = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)} \end{pmatrix}$$

Si $m_1(X) = Irr(\alpha, \mathbb{F}_2)$ *y* $m_3(X) = Irr(\alpha^3, \mathbb{F}_2)$, *el polinomio generador de* $\mathcal{C}_4$ *es*

$$g(X) = mcm\{m_1(X), m_3(X)\}.$$

Para poder realizar un cálculo concreto fijemos $m = 4$ *(luego* $n = 15$*). En este caso, los polinomios* $m_1(X), m_3(X)$ *son*

$$\begin{aligned} m_1(X) &= 1 + X + X^4 \\ m_3(X) &= 1 + X + X^2 + X^3 + X^4 \end{aligned}$$

por lo que

$$g(X) = (1 + X + X^4)(1 + X + X^2 + X^3 + X^4) = 1 + X^4 + X^6 + X^7 + X^8$$

y la dimensión de $\mathcal{C}_4$ *es* $n - \deg g(X) = 7$. *Las matrices generatriz y de control del código pueden obtenerse por el procedimiento habitual, a partir de* $g(X)$, *o directamente a partir de* H'. *Tomando la base* $\{1, \alpha, \alpha^2, \alpha^3\}$ *de* $\mathbb{F}_{2^4}$ *sobre* $\mathbb{F}_2$, *las coordenadas de cada elemento de* $\mathbb{F}_{2^4}$ *son*

$$\begin{array}{cccc} 1 = 1000 & \alpha^4 = 1100 & \alpha^8 = 1010 & \alpha^{12} = 1111 \\ \alpha = 0100 & \alpha^5 = 0110 & \alpha^9 = 0101 & \alpha^{13} = 1011 \\ \alpha^2 = 0010 & \alpha^6 = 0011 & \alpha^{10} = 1110 & \alpha^{14} = 1001 \\ \alpha^3 = 0001 & \alpha^7 = 1101 & \alpha^{11} = 0111 & \end{array}$$

(recuérdese que $Irr(\alpha, \mathbb{F}_2) = 1 + X + X^4$*, luego* $1 + \alpha = \alpha^4$*). Con esto,* H' *es*

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \end{pmatrix}$$

una matriz de control, en sentido estricto, queda

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

2.4.2. Descodificación de los Códigos BCH

El interés de los códigos BCH radica, de una parte en la posibilidad de elegir a priori la capacidad correctora deseada (determinada por δ) y de otra en la existencia de un algoritmo efectivo de descodificación.

Sea $\mathcal{C}$ el código BCH sobre $\mathbb{F}_q$ de longitud n y distancia prevista $\delta = 2t + 1$ (luego $\mathcal{C}$ puede corregir t errores). Sea α una raíz n-ésima primitiva de la unidad. Por simplicidad, vamos a descodificar el código determinado por las raíces $\alpha, \ldots, \alpha^{\delta-1}$, luego con

$$H' = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{\delta-1} & \alpha^{2(\delta-1)} & \cdots & \alpha^{(n-1)(\delta-1)} \end{pmatrix}. \tag{2.21}$$

Supongamos enviada una palabra $\mathbf{c} \in \mathcal{C}$ y recibido un vector $\mathbf{y} = \mathbf{c} + \mathbf{e}$ con $w(\mathbf{e}) = r \leq t$. Sean $0 \leq i_1 < \ldots < i_r \leq n - 1$, las posiciones en que han ocurrido errores y $e_{i_1}, \ldots, e_{i_r}$ las coordenadas del error $\mathbf{e}$ en esas posiciones. El primer paso en la descodificación consiste en calcular el síndrome del vector recibido

$$s = s(\mathbf{y}) = s(\mathbf{e}) = H\mathbf{y}^t = (s_0, \ldots, s_{\delta-2})^t. \tag{2.22}$$

Utilizando notación polinómica, podemos escribir $s(X) = s_0 + \cdots + s_{\delta-2}X^{\delta-2}$. Obsérvese que para cada $h = 0, \ldots, \delta - 2$,

$$s_h = y(\alpha^{h+1}) = e(\alpha^{h+1}) = \sum_{j=1}^{r} e_{i_j}(\alpha^{h+1})^{i_j} = \sum_{j=1}^{r} e_{i_j}(\alpha^{i_j})^{h+1}. \quad (2.23)$$

Si, para simplificar la notación, convenimos en poner $\eta_j = \alpha^{i_j}$; $\epsilon_j = e_{i_j}$ para $j = 1, \ldots, r$, resulta $s_h = \epsilon_1\eta_1^{h+1} + \cdots + \epsilon_r\eta_r^{h+1}$. Los η_j son llamados *localizadores del error* y los ϵ_j, *valores del error* (siempre con respecto al vector recibido $\mathbf{y}$). Por supuesto, conocer estos $2r$ números es equivalente a conocer el error. Si $s(X) = 0$ entonces $\mathbf{y} \in \mathcal{C}$, con lo que el mensaje recibido es dado por válido. En lo que sigue supondremos $s(X) \neq 0$.

Definición 2.46. *Llamaremos* polinomio localizador de errores *al polinomio*

$$L(X) = (1 - \eta_1 X) \cdots (1 - \eta_r X). \quad (2.24)$$

Llamaremos polinomio evaluador de errores *al polinomio*

$$E(X) = \sum_{j=1}^{r} \epsilon_j \eta_j \prod_{i \neq j} (1 - \eta_i X). \quad (2.25)$$

$L(X)$ y $E(X)$ son polinomios de grados r y $r - 1$ respectivamente, cuyo conocimiento implica el de los η_j y ϵ_j, ya que

Proposición 2.47. *En las condiciones anteriores,*
a) si $\rho_1, \ldots, \rho_r$ son las raíces de $L(X)$, entonces sus inversos $\rho_1^{-1}, \ldots, \rho_r^{-1}$ son los localizadores del error;
b) conocidos $\eta_1, \ldots, \eta_r$, los valores del error son

$$\epsilon_j = \frac{-E(\eta_j^{-1})}{L'(\eta_j^{-1})} \quad (2.26)$$

siendo $L'(X)$ la derivada (formal) de $L(X)$.

Demostración. La parte a) es evidente. Probemos b). Como

$$L'(X) = \sum_{h=1}^{r} (-\eta_h) \prod_{i \neq h} (1 - \eta_i X),$$

η_j^{-1} es raíz de todos los sumandos de $L'(X)$ excepto del j-ésimo. Por tanto

$$\epsilon_j L'(\eta_j^{-1}) = (-\eta_j)\epsilon_j \prod_{i \neq j}(1 - \eta_i \eta_j^{-1}) = -E(\eta_j^{-1}).$$

□

El algoritmo de descodificación no proporciona directamente el vector $\mathbf{e}$, sino los polinomios $L(X)$ y $E(X)$. A partir de ellos, según la proposición anterior, deducimos los η_j y ϵ_j y, finalmente, las posiciones y valores del error. El polinomio evaluador de errores admite una escritura alternativa en términos de series de potencias. En concreto, como

$$\frac{1}{1 - aX} = \sum_{i=0}^{\infty} a^i X^i \tag{2.27}$$

podemos poner

$$\begin{aligned} E(X) &= \sum_{j=1}^{r} \epsilon_j \frac{L(X)}{1 - \eta_j X} \eta_j \\ &= L(X) \sum_{j=1}^{r} \epsilon_j \eta_j \sum_{i=0}^{\infty} \eta_j^i X^i \\ &= L(X) \sum_{i=0}^{\infty} \left(\sum_{j=1}^{r} \epsilon_j \eta_j^{i+1} \right) X^i \\ &= L(X) \sum_{i=0}^{\infty} e(\alpha^{i+1}) X^i. \end{aligned}$$

Teorema 2.48 (Ecuación clave)**.**

$$E(X) \equiv L(X)s(X) \pmod{X^{\delta-1}}. \tag{2.28}$$

Demostración. Según lo computado anteriormente

$$\begin{aligned} E(X) \equiv\ & L(X) \sum_{i=0}^{\delta-2} e(\alpha^{i+1}) X^i = \\ & L(X) \sum_{i=0}^{\delta-2} s_i X^i = L(X)s(X) \pmod{X^{\delta-1}} \end{aligned}$$

como queríamos demostrar. □

Nota 2.49. *La ecuación clave muestra que, una vez conocidos $s(X)$ y $L(X)$, el polinomio evaluador de errores, $E(X)$, se obtiene inmediatamente. Así, el proceso de descodificación quedará completo en cuanto seamos capaces de determinar $L(X)$.*

Consideremos el polinomio

$$B(X) = \prod_{i=1}^{r}(X - \eta_i). \tag{2.29}$$

Como es bien conocido, $B(X)$ puede escribirse en función de los polinomios simétricos elementales $\sigma_1, \ldots, \sigma_r$ en los η_i:

$$B(X) = X^r - \sigma_1 X^{r-1} + \cdots + (-1)^r \sigma_r. \tag{2.30}$$

Las raíces de $B(X)$ son $\eta_1, \ldots, \eta_r$, luego de la escritura anterior se deducen las r ecuaciones

$$\eta_i^r - \sigma_1 \eta_i^{r-1} + \cdots + (-1)^r \sigma_r = 0,\ i = 1, \ldots, r. \tag{2.31}$$

Multiplicando la ecuación i-ésima por $\epsilon_i \eta_i^j$, j fijo, $1 \leq j \leq r$, y sumando todas las relaciones así obtenidas, se tiene

$$s_{j+r-1} - \sigma_1 s_{j+r-2} + \cdots + (-1)^r \sigma_r s_{j-1} = 0. \tag{2.32}$$

Repitiendo el proceso para cada $j = 1, \ldots, r$, finalmente se obtiene el sistema lineal de r ecuaciones en las r incógnitas $l_i = (-1)^i \sigma_i$

$$[S] \begin{cases} s_r & + & s_{r-1}l_1 & + & \cdots & + & s_0 l_r & = & 0 \\ s_{r+1} & + & s_r l_1 & + & \cdots & + & s_1 l_r & = & 0 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ s_{2r-1} & + & s_{2r} l_1 & + & \cdots & + & s_{r-1} l_r & = & 0. \end{cases}$$

Lema 2.50. *El sistema anterior tiene solución única en las l_i si y solamente si el error* **e** *tiene justamente peso r.*

Demostración. La matriz de coeficientes del sistema,

$$C = \begin{pmatrix} s_0 & s_1 & \cdots & s_{r-1} \\ s_1 & s_2 & \cdots & s_r \\ \vdots & \vdots & & \vdots \\ s_{r-1} & s_r & \cdots & s_{2r} \end{pmatrix}$$

se factoriza en la forma $C = VDV^t$, siendo

$$V = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \eta_1 & \eta_2 & \cdots & \eta_r \\ \vdots & \vdots & & \vdots \\ \eta_1^{r-1} & \eta_2^{r-1} & \cdots & \eta_r^{r-1} \end{pmatrix}, \; D = \begin{pmatrix} \epsilon_1\eta_1 & 0 & \cdots & 0 \\ 0 & \epsilon_2\eta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \epsilon_r\eta_r \end{pmatrix}.$$

Dado que V es una matriz de tipo Vandermonde y D una matriz diagonal, el que ambas sean no singulares equivale a la condición enunciada. □

Una vez conocidos los $l_1, \ldots, l_r$ (y, por supuesto, el número de errores r), el polinomio localizador de errores se deduce inmediatamente

Proposición 2.51. *El polinomio localizador de errores es*

$$L(X) = 1 + l_1X + \cdots + l_rX^r. \tag{2.33}$$

Demostración. Es suficiente probar que el polinomio, así obtenido, verifica que $L(\eta_i^{-1}) = 0$ para $i = 1, \ldots, r$. Ahora bien

$$\eta_i^r L(\eta_i^{-1}) = \eta_i^r + l_1\eta_i^{r-1} + \cdots + l_r = B(\eta_i) = 0 \tag{2.34}$$

por definición de $B(X)$. □

Como el número de errores de la palabra recibida es desconocido, sería necesario calcularlo. En virtud del lema 2.50 basta determinar el máximo r tal que el sistema $[S]$ tenga solución única. Una vez hecho esto, basta resolver tal sistema para obtener los l_i. Sin embargo, el proceso anterior se revela computacionalmente impracticable. Por ello han sido propuestos diversos métodos alternativos, que proporcionan algoritmos eficientes para calcular los l_i, el más conocido de los cuales es debido a Berlekamp y Massey. Lo que este algoritmo calcula, en realidad, es el polinomio mínimo de una sucesión recurrente de orden r, supuesto conocidos $2r$ términos de dicha sucesión. No pudiendo abordar aquí toda la teoría de sucesiones recurrentes sobre cuerpos finitos, necesaria para la justificación del método, nos limitaremos a exponer las etapas del algoritmo resultante. Señalemos únicamente, que el sistema $[S]$ muestra que los s_i constituyen realmente una sucesion recurrente de orden r, con ecuación de recurrencia,

$$s_r + s_{r-1}l_1 + \cdots + s_0l_r = 0 \tag{2.35}$$

y que realmente se conocen $2r$ términos de tal sucesión, puesto que $r \leq t = \lfloor \delta - 1/2 \rfloor$, y $s_0, s_1, \ldots, s_{\delta-2}$ se deducen del síndrome del vector recibido. A partir de él se construyen recursivamente las cuatro sucesiones

$$\{L_j(X)\}_{j=0}^{2t} \ , \ \{E_j(X)\}_{j=0}^{2t} \ , \ \{m_j\}_{j=0}^{2t} \ , \ \{b_j\}_{j=0}^{2t}. \tag{2.36}$$

Para ello, partimos de los valores iniciales $L_0(X) = 1, E_0(X) = X, m_0 = 0, b_0 = s_0$ y utilizamos la ley de recurrencia

$$\begin{aligned}
L_{j+1}(X) &= L_j(X) - b_j E_j(X). \\
E_{j+1}(X) &= \begin{cases} b_j^{-1} X L_j(X) & \text{si } b_j \neq 0 \text{ y } m_j \geq 0; \\ X E_j(X) & \text{en otro caso.} \end{cases} \\
m_{j+1} &= \begin{cases} -m_j & \text{si } b_j \neq 0 \text{ y } m_j \geq 0; \\ m_j + 1 & \text{en otro caso.} \end{cases} \\
b_{j+1} &= \text{coeficiente de } X^{j+1} \text{ en } L_{j+1}(X) s(X).
\end{aligned}$$

Una vez construidas estas sucesiones, sea $s = \lfloor t + 1/2 - m_{2t}/2 \rfloor$. El polinomio

$$m(X) = X^s L_{2t}(1/X) \tag{2.37}$$

resulta ser el polinomio mínimo de la sucesión recurrente, y tiene por coeficientes precisamente los l_i buscados.

Una vez conocidos estos l_i, se construye el polinomio localizador de errores $L(X)$ según 2.51 y, a partir de él y de $s(X)$, el polinomio evaluador de errores $E(X)$, como ya se vió. Las posiciones del error se determinan a partir de las raíces de $L(X)$. Para encontrar tales raíces se evalua este polinomio en todos los $1, \alpha, \ldots, \alpha^{n-1}$ (método de Chien). Si $\alpha^{h_1}, \ldots, \alpha^{h_r}$ son tales raíces, entonces sus inversos son los localizadores del error, $\eta_1 = \alpha^{n-h_1}, \ldots, \eta_r = \alpha^{n-h_r}$. El valor del error asociado al localizador η_j es

$$\epsilon_j = \frac{E(\alpha^{h_j})}{L'(\alpha^{h_j})} \tag{2.38}$$

siendo L' la derivada (formal) de L. Finalmente se corrige el mensaje recibido: para $i = 0, \ldots, n-1$, la i-ésima coordenada del mensaje corregido es

$$\begin{cases} y_i & \text{si } i \neq n - h_1, \ldots, n - h_r; \\ y_i - \epsilon_j & \text{si } i = n - h_j. \end{cases} \tag{2.39}$$

Ejemplo 2.52. *Sean* $q = 3, n = 8$ *y* α *una raíz del polinomio* $2+X+X^2$. *Como* $Irr(\alpha, \mathbb{F}_3) = Irr(\alpha^3, \mathbb{F}_3) = 2 + X + X^2$, $Irr(\alpha^2, \mathbb{F}_3) = 1 + X^2$ *e* $Irr(\alpha^4, \mathbb{F}_3) = 1 + X$, *el código BCH de longitud 8 y distancia mínima prevista* $\delta = 5$ *sobre* $\mathbb{F}_3$, *tiene polinomio generador* $g(X) = 2 + X^2 + X^3 + 2X^4 + X^5 \sim 201121$, *luego dimensión* $k = 3$. *El mensaje fuente* $\mathbf{m}= 010 \sim X$, *se codifica por*

$$m(X)g(X) = 2X + X^3 + X^4 + 2X^5 + X^6 \sim 02011210 = \mathbf{c}.$$

Supongamos que durante la transmisión de esta palabra se produce el error $\mathbf{e} = 10000100 \sim 1 + X^5$, *con lo que el mensaje recibido es*

$$\mathbf{c} + \mathbf{e} = \mathbf{y} = 12011010 \sim 1 + 2X + X^3 + X^4 + X^6$$

que contiene errores en las posiciones 0 y 5. Naturalmente, estos errores son desconocidos por el receptor y es necesario calcularlos a partir de $\mathbf{y}$ *y del código usado. El proceso es el siguiente.*
Etapa 1. *El síndrome del vector recibido es*

$$\begin{aligned} s_0 &= e(\alpha) = y(\alpha) = 2\alpha + 1 \\ s_1 &= e(\alpha^2) = y(\alpha^2) = 2\alpha + 2 \\ s_2 &= e(\alpha^3) = y(\alpha^3) = \alpha + 2 \\ s_3 &= e(\alpha^4) = y(\alpha^4) = 0 \end{aligned}$$

Etapa 2. *El método de Berlekamp-Massey proporciona los siguientes resultados:*

j	L_j	E_j	m_j	b_j
0	*1*	X	*0*	$1 + 2\alpha$
1	$1 + (\alpha + 2)X$	$(\alpha + 2)X$	*0*	2α
2	$1 + 2\alpha X$	$(2 + 2\alpha)X + \alpha X^2$	*0*	α
3	$1 + (1 + 2\alpha)X + (2 + \alpha)X^2$	$(1 + \alpha)X + 2X^2)$	*0*	$1 + \alpha$
4	$1 + (2 + \alpha)X + 2\alpha X^2$	$\alpha X + (2 + 2\alpha)X^2$	*0*	$2 + 2\alpha$

con lo cual $m(X) = X^2 + (2 + \alpha)X + 2\alpha$ *y* $l_1 = 2 + \alpha$, $l_2 = 2\alpha$.
Etapa 3. *El polinomio localizador de errores es* $L(X) = 1 + (2 + \alpha)X + 2\alpha X^2$, *cuyas raíces son* α^8 *y* α^3. *Por lo tanto,* $\eta_1 = \alpha^0$ *y* $\eta_2 = \alpha^5$. *Así* $e(X) = 1 + X^5 \sim \mathbf{e} = 10000100$ *y, consecuentemente*

$$\mathbf{c} = \mathbf{y} - \mathbf{e} = 02011210 \sim m(X)g(X) = 2X + X^3 + X^4 + 2X^5 + X^6.$$

Finalmente recuperamos el mensaje original $\mathbf{m} \sim m(X)g(X)/g(X) = 010$.

2.4.3. Códigos de Reed-Solomon

Repitamos la definición, ya citada, de código de Reed-Solomon:

Definición 2.53. *Un código* Reed-Solomon *sobre* $\mathbb{F}_q$ *es un código BCH primitivo de longitud* $n = q - 1$.

Como caso particular de los BCH que son estos códigos, siguen los mismos procesos de codificación y descodificación que aquellos. Su característica distintiva más notable es que la raíz n-ésima, α, es un elemento de $\mathbb{F}_q$ y, por tanto, todas las manipulaciones con el código implican sólo operaciones en el propio cuerpo $\mathbb{F}_q$. Como contrapartida a esta simplicidad de manejo, queda limitada a $q-1$ la longitud de un código Reed-Solomon sobre $\mathbb{F}_q$.

Una primera justificación del interés de estos códigos es el siguiente resultado.

Proposición 2.54. *Los códigos Reed-Solomon son MDS.*

Demostración. Como en todo código BCH, fijada la distancia prevista δ y la raíz n-ésima α, su distancia mínima y dimensión, d y k, verifican $d \geq \delta$ y $k = n - \deg g(X)$, siendo $g(X) = \operatorname{mcm}\{\operatorname{Irr}(\alpha^i, \mathbb{F}_q) \mid i = 1, \ldots, \delta - 1\}$. Ahora bien, dado que $\alpha^i \in \mathbb{F}_q$ para todo i, se tendrá $\deg g(X) = \delta - 1$ luego, teniendo en cuenta la cota de Singleton, $k = n - \delta - 1$, de donde $d = n - k + 1$. □

Los códigos de Reed-Solomon son habitualmente utilizados en la detección de errores (aleatorios y a ráfagas) sobre canales binarios. Dado que, como se ha dicho, su longitud es muy pequeña, para 'alargarla' se utiliza a menudo la estrategia de descenso de cuerpo, que describimos a continuación.

Dado un cuerpo finito $\mathbb{F}_{q^r}$, extensión de $\mathbb{F}_q$, fijemos una base $\{1, \alpha, \ldots, \alpha^{r-1}\}$ de $\mathbb{F}_{q^r}$ sobre $\mathbb{F}_q$. Como sabemos, cada elemento de $\mathbb{F}_{q^r}$ puede identificarse con el vector de $\mathbb{F}_q^r$ de sus coordenadas en la base anterior. Aplicando este procedimiento a cada componente de un vector de $\mathbb{F}_{q^r}^n$, obtenemos un vector de $\mathbb{F}_q^{rn}$

$$\begin{array}{c}\text{vector original sobre } \mathbb{F}_{q^r} \\ \overbrace{*\cdots*}^{*}\overbrace{*\cdots*}^{*}\cdots\cdots\overbrace{*\cdots*}^{*} \\ \text{vector obtenido sobre } \mathbb{F}_q\end{array} \tag{2.40}$$

En particular, si $\mathcal{C}$ es un código de longitud n sobre $\mathbb{F}_{q^r}$, a partir de él podemos conseguir un código sobre $\mathbb{F}_q$ de longitud rn; se dice que este cuerpo se obtiene del original *por descenso de cuerpo.*

Esta construcción se realiza habitualmente con códigos Reed-Solomon definidos sobre $\mathbb{F}_{2^r}$. Si el código inicial tiene longitud n, entonces el código binario obtenido por descenso de cuerpo sobre $\mathbb{F}_2$ tiene longitud rn y gran capacidad de detección de errores a ráfagas.

Proposición 2.55. *Sea $\mathcal{C}$ un código de Reed-Solomon sobre $\mathbb{F}_{2^r}$ con distancia $d = 2t + 1$. Entonces, el código binario obtenido por descenso de cuerpo sobre $\mathbb{F}_2$ corrige todos los errores a ráfagas de longitud $l \leq (t-1)r + 1$.*

Demostración. Recibida una palabra binaria, podemos transformarla en un vector de $\mathbb{F}_{2^r}$ siguiendo el proceso contrario al descrito anteriormente (es decir, ascendiendo de cuerpo). Si los errores forman un ráfaga de longitud $l \leq (t-1)r+1$, teniendo en cuenta que cada r símbolos binarios se colapsan en uno solo de $\mathbb{F}_{2^r}$, la palabra transformada contiene a lo más t coordenadas erróneas. Pero t es precisamente la capacidad de corrección del código. □

2.5. Códigos polinomiales ($\star$)

Sea $\mathcal{P} = \{P_1, \cdots, P_n\}$ un conjunto de puntos pertenecientes a cierto 'objeto geométrico' $\mathcal{X}$. Si V es un espacio vectorial de funciones $f : \mathcal{X} \longrightarrow \mathbb{F}_q$, podemos considerar la aplicación de evaluación en $\mathcal{P}$

$$ev_{\mathcal{P}} : V \longrightarrow \mathbb{F}_q^n \ , \ ev_{\mathcal{P}}(f) = (f(P_1), \cdots, f(P_n)).$$

Si $ev_{\mathcal{P}}$ es lineal, su imagen es un subespacio vectorial de $\mathbb{F}_q^n$, es decir, un código lineal sobre $\mathbb{F}_q$ de longitud n. Sus palabras son los $ev_{\mathcal{P}}(f)$, $f \in V$. Diremos que éste código es obtenido *por evaluación en $\mathcal{P}$ de las funciones de V*. Podemos aclarar el significado y alcance de esta definición mediante unos ejemplos.

2.5.1. Códigos Reed-Solomon

Sea $\mathcal{C}$ el código Reed-Solomon de distancia δ y longitud $n = q - 1$ sobre $\mathbb{F}_q$, con polinomio generador

$$g(X) = \prod_{i=0}^{\delta-2} (X - \alpha^i)$$

donde α es una raiz n-ésima primitiva de la unidad. Su matriz de control es

$$H = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{n-1} \\ \vdots & \vdots & & \cdots \\ 1 & \alpha^{\delta-2} & \cdots & \alpha^{(n-1)(\delta-2)} \end{pmatrix}.$$

Por consiguiente, el dual de $\mathcal{C}$, cuya matriz generatriz es H, puede ser visto como el código obtenido por evaluación de las funciones de $V = \{f(X) \in \mathbb{F}_q[X] \mid \deg(f) \leq \delta - 2\}$ en los puntos $\{1, \alpha, \ldots, \alpha^{n-1}\}$ de $\mathbb{F}_q$ (o de la recta afín sobre $\mathbb{F}_q$).

Esta interpretación puede servirnos para generalizar los códigos RS e intentar remediar su principal problema que, como sabemos, consiste en su escasa longitud.

2.5.2. Códigos Reed-Muller

Tomemos ahora $V = \{f \in \mathbb{F}_q[X_1, \ldots, X_m] \mid \deg(f) \leq r\}$ y evaluemos estos polinomios en los puntos de $\mathcal{P} = \mathbb{F}_q^m$ El código obtenido se llama *Reed-Muller q-ario de orden r y longitud* q^m, $\mathcal{RM}_q(r, m)$. De nuevo, el dual de un código Reed-Muller es también Reed-Muller, ya que

$$\mathcal{RM}_q(r, m)^\perp = \mathcal{RM}_q(m(q-1) - r - 1, m)$$

lo que se deja como ejercicio al lector.

2.5.3. Códigos algebraico-geométricos

Otra posibilidad consiste sustituir la recta afín $\mathcal{X} = \mathbb{F}_q$ por una curva $\mathcal{X}$ sobre $\mathbb{F}_q$, $\mathcal{P}$ por un conjunto de puntos racionales de $\mathcal{X}$ y tomar como V un conjunto de funciones racionales sobre $\mathcal{X}$. Los códigos de evaluación

para estos datos reciben el nombre de *códigos algebraico-geométricos de Goppa*.

Más concretamente, sea $\mathcal{X}$ una curva proyectiva, lisa, absolutamente irreducible definida sobre $\mathbb{F}_q$. Para cada divisor racional E de $\mathcal{X}$, el conjunto

$$\mathcal{L}(E) = \{\text{funciones racionales } f \in \mathbb{F}_q(\mathcal{X})^* \mid (f) + E \geq 0\} \cup \{0\}$$

(siendo (f) el divisor de f), es un espacio vectorial sobre $\mathbb{F}_q$, cuya dimensión siempre es finita y denotaremos por $\ell(D)$. Sea $\mathcal{P} = \{P_1, \ldots, P_n\}$, un conjunto de n puntos de $\mathcal{X}$ racionales y distintos, con los que construimos el divisor $D = P_1 + \cdots + P_n$. Sea, finalmente, G otro divisor sobre $\mathcal{X}$, racional y con soporte disjunto del de D, $\text{sop(G)} \cap \text{sop(D)} = \emptyset$. Con estos elementos, consideramos la aplicación de evaluación

$$ev : \mathcal{L}(G) \longrightarrow \mathbb{F}_q^n \; ; \; ev(f) = (f(P_1), \cdots, f(P_n)).$$

La aplicación ev está bien definida. En efecto, como $P_i \notin \text{sop(G)}$, P_i no puede ser un polo de $f \in \mathcal{L}(G)$. Por otro lado, tanto P_i como G son racionales sobre $\mathbb{F}_q$, luego $f(P_i) \in \mathbb{F}_q$. Además, es evidente que ev es una aplicación lineal de espacios vectoriales.

Llamaremos *código algebraico-geométrico* (o *código geométrico de Goppa*) asociado a $\mathcal{X}, D$ y G, a $\mathcal{C}(\mathcal{X}, D, G) = ev(\mathcal{L}(G))$. Obsérvese que podemos limitarnos al caso $0 \leq \deg(G) \leq n + 2g - 2$, siendo g el género de la curva $\mathcal{X}$. En efecto, si $\deg G < 0$ entonces $\mathcal{L}(G) = \{0\}$, mientras que si $\deg(G) > n + 2g - 2$ entonces $\mathcal{C} = \mathbb{F}_q^n$.

Teorema 2.56. *$\mathcal{C}(\mathcal{X}, D, G)$ es un código lineal de parámetros $[n, k, d]$, con*
a) $k = \ell(G) - \ell(G - D)$;
b) $d \geq n - \deg(G)$.

Demostración. $\mathcal{C}(\mathcal{X}, D, G)$ es claramente lineal, ya que ev lo es, y su longitud es n. Como el núcleo de ev es

$$\ker(ev) = \{f \in \mathcal{L}(G) \mid f(P_i) = 0, i = 1, \cdots, n\} = \mathcal{L}(G - D)$$

la dimensión de $\mathcal{C}$ es $\dim(\mathcal{L}(G)) - \dim(\ker(ev)) = \ell(G) - \ell(G - D)$. Comprobemos la acotación para la distancia. Sea $\mathbf{c} \in \mathcal{C}$ un elemento de peso mínimo, d. Si $\mathbf{c} = ev(f)$, entonces f se anula en $n - d$ puntos del soporte de D, pongamos $P_1, \cdots, P_{n-d}$. Por tanto $f \in \mathcal{L}(G - P_1 - \cdots - P_{n-d})$, con lo que $\deg(G) - (n - d) \geq 0$. □

Si imponemos restricciones adicionales en el grado del divisor G podemos refinar el resultado obtenido para k.

Corolario 2.57. *Si* $\deg(G) < n$, *entonces* $k = \ell(G)$. *Si* $2g - 2 < \deg(G) < n$, *entonces* $k = \deg(G) + 1 - g$.

Demostración. Si $\deg(G) < n$ entonces $\deg(G-D) < 0$ luego $\ell(G-D) = 0$. Si, ademas, $2g-2 < \deg(G)$ entonces, de nuevo por razones de grado, $\ell(W - G) = 0$, siendo W un divisor canónico de $\mathcal{X}$. El resultado se deduce del teorema de Riemann-Roch. □

Para estimar la calidad de los parámetros de un código algebraico-geométrico, limitándonos al caso $2g - 2 < \deg(G) < n$, observemos que, según el Teorema 2.56 y la cota de Singleton

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

luego los códigos obtenidos a partir de curvas racionales (de género 0) son siempre MDS (de hecho, RS extendidos). La acotación va 'empeorando' a medida que crece el género de la curva.

2.6. Funciones orden, códigos de evaluación ($\star$)

En esta sección vamos a formalizar el método de construcción de los códigos de evaluación. Además, esto nos permitirá prescindir de los formalismos de la geometría algebraica, utilizando unicamente métodos elementales.

2.6.1. Ordenes y pesos

Sean $\mathbb{F}_q$ un cuerpo finito, R una $\mathbb{F}_q$-álgebra conmutativa con unidad y $\mathbb{N}_0$ el conjunto de los enteros no negativos.

Definición 2.58. *Una* función orden *en* R *es una función* $\rho : R \to \mathbb{N}_0$ *verificando las siguientes propiedades: para todos* $f, g \in R$ *y* $\lambda \in \mathbb{F}_q$,

(O.0) $\rho(f) = -\infty$ *si y sólo si* $f = 0$;

(O.1) $\rho(\lambda f) = \rho(f)$;

(O.2) $\rho(f + g) \leq \max\{\rho(f), \rho(g)\}$; *y*

(O.3) *si* $\rho(f) = \rho(g)$, *existe* $\lambda \in \mathbb{F}_q$ *tal que* $\rho(f + \lambda g) < \rho(f)$.

Si, además, se verifica la condición suplementaria

(O.4) $\rho(fg) = \rho(f) + \rho(g)$,

diremos que ρ *es un peso en* R.

Nota 2.59. *A partir de la definición anterior se verifican facilmente las siguientes propiedades:*
1. Si R *admite una función orden, entonces* R *es un dominio de integridad. Este hecho justifica el nombre de* dominios con orden *que daremos a estas álgebras.*
2. $\{f \in R : \rho(f) = 0\} = \mathbb{F}_q^*$.
3. Si $\rho(f) < \rho(g)$, *entonces* $\rho(f + g) = \rho(g)$.

Veamos algunos ejemplos

Ejemplo 2.60. *(1) La función grado,* $\deg : \mathbb{F}_q[X] \to \mathbb{N}_0$, *es una función orden. (2) Sea* $\mathcal{X}$ *una curva sobre* $\mathbb{F}_q$ *y* Q *un punto racional. Sea* R *el conjunto de funciones racionales sobre* $\mathcal{X}$ *con polos sólo en* Q. *La valoración asociada a* Q, *cambiada de signo,* $\rho = -v_Q$, *es una función orden en* R. *(4) El álgebra* $R = \mathbb{F}_q[X, Y]/(XY - 1)$ *es íntegra pero no admite ninguna función orden. Esto prueba que no todo dominio es un dominio de orden.*

Proposición 2.61. *Sea* R *una* $\mathbb{F}_q$*-álgebra que admite una función orden* ρ *y sea* $\{\rho_i : i \in \mathbb{N}\}$ *la sucesión creciente de enteros que aparecen como orden de algún elemento de* R. *Para cada* $i \in \mathbb{N}$ *sea* $f_i \in R$ *tal que* $\rho(f_i) = \rho_i$. *Entonces*
(a) $\{f_i \ : \ i \in \mathbb{N}\}$ *es una base de* R;
(b) sea $L_i = \langle f_1, \ldots, f_i \rangle$. *Si* $f \in L_i \setminus L_{i-1}$ *y* $f \neq 0$, *entonces* $\rho(f_i) = \rho_i$;
(c) para $i, j \in \mathbb{N}$ *sea* $l(i, j)$ *el menor entero* l *tal que* $f_i f_j \in L_l$. *La función* l *es estrictamente creciente en sus dos argumentos.*

Las tres propiedades de esta proposición son fáciles de ver y se dejan como ejercicio al lector.

2.6.2. Los códigos

Sea R una $\mathbb{F}_q$-álgebra que admite una función orden ρ. Sean $\{f_i\}$ y $l(i, j)$ como en la Proposición anterior. Consideremos en $\mathbb{F}_q^n$ el producto $*$

coordenada a coordenada, $(x_1, \ldots, x_n) * (y_1, \ldots, y_n) = (x_1y_1, \ldots, x_ny_n)$, y sea finalmente $\phi : R \to \mathbb{F}_q^n$ un morfismo suprayectivo de $\mathbb{F}_q$-álgebras (es decir, ϕ es lineal y $\phi(fg) = \phi(f) * \phi(g)$). Pongamos $\mathbf{h}_i = \phi(f_i)$ y para $l = 1, 2, \ldots$ consideremos los códigos

$$E_l = \phi(L_l) = \langle \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_l \rangle$$

y sus duales $C_l = E_l^{\perp}$. Se obtiene así una secuencia $C_0 = \mathbb{F}_q^n \supseteq C_1 \supseteq \ldots$ decreciente de códigos y, dado que ϕ es suprayectiva, existirá un N tal que $C_l = \{0\}$ para $l \geq N$.

Vamos a calcular una cota sobre la distancia mínima de C_l, la llamda *cota del orden*. Para ello, dado un vector $\mathbf{y} \in \mathbb{F}_q^n$, introducimos los *síndromes 2-dimensionales relativos a* $\mathbf{y}$, $s_{ij} = s_{ij}(\mathbf{y}) = \mathbf{y}(\mathbf{h}_i * \mathbf{h}_j)$, y la *matriz de síndromes* $S(\mathbf{y}) = (s_{ij} \mid 1 \leq i, j \leq N)$.

Proposición 2.62. *rang*$(S(\mathbf{y})) = w(\mathbf{y})$.

Demostración. Sean H la matriz $N \times n$ cuya fila i-ésima es $\mathbf{h}_i$, y $D(\mathbf{y})$ la matriz diagonal cuya diagonal es $\mathbf{y}$. Como H posee rango máximo, $\operatorname{rang}(HD(\mathbf{y})H^t) = \operatorname{rang}(D(\mathbf{y})) = w(\mathbf{y})$, y el resultado es consecuencia de la igualdad $S(\mathbf{y}) = HD(\mathbf{y})H^t$. □

Consideremos ahora el conjunto $N_l = \{(i, j) \in \mathbb{N} \mid l(i, j) = l + 1\}$.

Proposición 2.63. *Sea* $N_l = \{(i_1, j_1), \ldots, (i_t, j_t)\}$. *Se verifica que*
(a) los $i_1, \ldots i_t$ *son todos ellos distintos; y*
(b) si $\mathbf{y} \in C_l \setminus C_{l+1}$, *entonces el menor formado por las filas y columnas* $i_1, \ldots i_t$ *de* $S(\mathbf{y})$, *es no nulo.*

Demostración. En primer lugar notemos que, siendo $l(i, j)$ simétrica, $\{i_1, \ldots, i_t\} = \{j_1, \ldots, j_t\}$. Ordenémolos de manera que $i_1, \leq i_2 \leq \cdots \leq i_t$ (luego $j_1 \geq \cdots \geq j_t$). Si $i_u = i_{u+1}$ entonces $j_u < j_{u+1}$ con lo que, al ser l estrictamente creciente en cada uno de sus argumentos, $l + 1 = l(i_u, j_u) < l(i_{u+1}, j_{u+1}) = l + 1$, lo que es imposible y prueba (a). Para (b) veamos que

$$s_{i_u j_v} = \begin{cases} 0 & \text{si } u < v; \\ \neq 0 & \text{si } u = v. \end{cases}$$

Si $u < v$ entonces $l(i_u, j_v) < l(i_v, j_v) = l + 1$, con lo que $\mathbf{h}_{i_u} * \mathbf{h}_{j_v} \in C_l^{\perp}$ y $s_{i_u j_v} = 0$. Recíprocamente, si $u = v$ entonces $l(i_u, j_v) = l + 1$ con lo que $\mathbf{h}_{i_u} * \mathbf{h}_{j_v} \notin C_l^{\perp}$ y $s_{i_u j_v} \neq 0$. □

Corolario 2.64. *Si* $\mathbf{y} \in C_l \setminus C_{l+1}$ *entonces* $w(\mathbf{y}) \geq \#N_l$.

Llamaremos *cota del orden* en la distancia mínima de C_l a

$$d_{ORD}(l) = \text{mín}\{\#N_t \mid t \geq l\}.$$

Es claro que, en virtud de los razonamientos anteriores, $d_{ORD}(l)$ es una cota para la distancia mínima de C_l, es decir, $d(C_l) \geq d_{ORD}(l)$.

Capítulo 3

Descodificación y eliminación

En este capítulo mostraremos un algoritmo para la descodificación de los códigos cíclicos utilizando las bases de Gröbner. Como vimos en el capítulo anterior en la sección 2.3, los códigos cíclicos tienen una estructura algebraica muy rica que nos permite diseñar algoritmos de descodificación algebraicos (véase sección 2.3.4 del capítulo anterior). Varios autores han utilizado la teoría de las base de Gröbner para resolver el problema de la descodificación de los códigos cíclicos: en [Fit95] P. Fitzpatrick utiliza las bases de Gröbner sobre módulos para encontrar una solución de la ecuación clave mediante un método con complejidad similar al algoritmo de Berlekamp-Massey que vimos en el apartado 2.3.4, este método lo estudiaremos en el capítulo siguiente. A.B. Cooper [Co90, Co91] presenta un algoritmo capaz de descodificar un código cíclico hasta su distancia mínima basado en la teoría de la eliminación (véase sección 1.5). El algoritmo de Cooper requiere calcular una base de Gröbner para cada síndrome no nulo recibido lo que, desde el punto de vista de la complejidad, no es el escenario más adecuado. Posteriormente X. Chen, I.S. Reed, T. Helleseth y K. Truong [CRHT94a, CRHT94b] revisan la idea original de Cooper y proponen el cálculo de una base de Gröbner genérica que elimina la necesidad del cálculo de una base de Gröbner para cada síndrome recibido. A pesar de todo, el cálculo de dicha base de Gröbner genérica es muy costoso. Repasaremos este enfoque en la sección 3.2 en la que exponemos la descodificación de códigos sobre variedades afines propuesta por J. Fitzgerald y R.F. Lax en [FL98]. Nosotros seguiremos en este capítulo la aportación de P. Loustaunau

y E.V. York [LY97] (en particular en la tesis doctoral de este último [Yor94]) basada en la aportación de X. Chen, I.S. Reed, T. Helleseth y K. Truong [CRHT94a, CRHT94b].

3.1. La variedad síndrome de un código cíclico

Sea α una raíz primitiva de $x^n - 1$ en el correspondiente cuerpo de descomposición de $\mathbb{F}_q$ (asumimos en todo este capítulo que $\mathrm{mcd}(n, q) = 1$ como es habitual). Consideremos $g(x)$ el polinomio generador de un código cíclico con raíces $\alpha^{i_1}, \ldots, \alpha^{i_r}$ donde $\{i_1, \ldots, i_r\} \subseteq \{1, \ldots, n-1\}$. El código $\mathcal{C}$ generado por $g(x)$ puede ser visto como el núcleo en $\mathbb{F}_q^n$ de la matriz de control (véase en la sección 2.3.3 la ecuación (2.14))

$$H = \begin{pmatrix} 1 & \alpha^{i_1} & \cdots & \alpha^{i_1(n-1)} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha^{i_r} & \cdots & \alpha^{i_r(n-1)} \end{pmatrix}. \tag{3.1}$$

Sea $\tilde{\mathbf{c}} \in \mathbb{F}_q^n$ un vector recibido con $\tilde{\mathbf{c}} = \mathbf{c} + \mathbf{e}$, donde $\mathbf{c} \in \mathcal{C}$ y $\mathbf{e}$ es el error cometido (Por conveniencia utilizaremos durante esta sección $\tilde{\mathbf{c}}$ para el vector recibido en lugar de $\mathbf{y}$ que reservaremos para los localizadores del error). El siguiente sistema de ecuaciones relaciona el síndrome $\mathbf{s}$ de $\tilde{\mathbf{c}}$ con el error cometido

$$e_0 + e_1\alpha^{i_j} + e_2\alpha^{2i_j} + \cdots + e_{n-1}\alpha^{(n-1)i_j} = s_j, \quad j = 1, \ldots, r. \tag{3.2}$$

Durante el resto de esta sección asumiremos que el error cometido tiene peso $w(\mathbf{e}) = \tau \leq t$, donde $2t + 1 = d$ la distancia mínima de $\mathcal{C}$. Expresaremos a continuación las soluciones del sistema expresado en (3.2) como una variedad algebraica. Consideremos los polinomios siguientes:

$$f_j = y_1 z_1^{i_j} + y_2 z_2^{i_j} + \cdots + y_t z_t^{i_j} - x_j, \quad j = 1, \ldots, r. \tag{3.3}$$

$$h_k = z_k^{n+1} - z_k, \quad k = 1, \ldots, t. \tag{3.4}$$

$$l_k = y_k^{q-1} - 1, \quad k = 1, \ldots, t. \tag{3.5}$$

Consideremos el conjunto de polinomios

$$F = \{f_j \mid j = 1, \ldots, r\} \cup \{h_k \mid k = 1, \ldots, t\} \cup \{l_k \mid k = 1, \ldots, t\} \tag{3.6}$$

y el ideal $I \subset \mathbb{F}_q[\mathbf{x}, \mathbf{z}, \mathbf{y}]$ generado por F. Llamaremos *variedad síndrome* a $\mathcal{V}(I)$, es decir:

$$\left\{\mathbf{p} \in \mathbb{F}_{q^m}^{r+2t} \mid f_j(\mathbf{p}) = h_k(\mathbf{p}) = l_k(\mathbf{p}) = 0,\ \forall j = 1, \ldots, r,\ k = 1, \ldots, t\right\} \tag{3.7}$$

y a I *ideal síndrome*. Es fácil comprobar que el cardinal de $\mathcal{V}(I)$ es $(q-1)^t(n+1)^t$ por lo tanto I es un ideal cero dimensional. Además $\mathcal{V}(I)$ contiene toda la información necesaria para descodificar el vector $\tilde{\mathbf{c}}$. Supongamos que tenemos

$$\mathbf{s} = H\tilde{\mathbf{c}} = H\mathbf{e}, \quad w(\mathbf{e}) = \tau \leq t,$$

existen puntos en la variedad $\mathcal{V}(I)$ que determinan los valores y posiciones del error $\mathbf{e}$. Dichos puntos vienen dados por

$$\mathbf{p} = (s_1, s_2, \ldots, s_r, 0, \ldots, 0, \alpha^{l_1}, \alpha^{l_2}, \ldots, \alpha^{l_\tau}, *, \ldots, *, \beta_1, \beta_2, \ldots, \beta_\tau), \tag{3.8}$$

donde $s_1, s_2, \ldots, s_r$ corresponde al síndrome (coordenadas en $\mathbf{x}$), los valores no nulos de $\mathbf{e}$ (posiciones del error) están localizados en las coordenadas $l_1, l_2, \ldots, l_\tau$ y $\beta_1, \beta_2, \ldots, \beta_\tau$ su valor. El símbolo $*$ puede ser cualquier valor no nulo del cuerpo $\mathbb{F}_q$.

Para cada síndrome $\mathbf{s}$ con $w(\mathbf{e}) = \tau \leq t$ existen

$$\binom{t}{\tau} \tau! \, (q-1)^{t-\tau}$$

puntos en la variedad $\mathcal{V}(I)$ correspondientes a las permutaciones de las variables en $\mathbf{z}$ y la correspondiente permutación de las variables en $\mathbf{y}$. Notaremos por $\mathcal{V}_\mathbf{s}$ al conjunto de puntos correspondientes al síndrome $\mathbf{s}$ y por $S \subset \mathbb{F}_q^r$ al conjunto de todos los síndromes posibles no nulos que corresponden a errores con peso menor que t. Sea

$$\mathcal{E} = \bigcup_{\mathbf{s} \in S} \mathcal{V}_\mathbf{s}, \tag{3.9}$$

el conjunto $\mathcal{E}$ contiene

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

puntos con toda la información necesaria para descodificar cualquier mensaje recibido con menos de t errores. Sin embargo, la variedad síndrome contiene muchos más puntos que $\mathcal{E}$.

Ejemplo 3.1. *Consideremos un código BCH primitivo con $q = 2$, longitud 31, dimensión 11 y capacidad correctora 5, se tiene que*

$$|\mathcal{V}(I)| = 33554432$$

$$|\mathcal{E}| = 24444275.$$

Para que nuestro ideal sólo considere aquellos puntos que se encuentran en el conjunto $\mathcal{E}$ añadiremos a F los $\binom{t}{2}$ polinomios

$$z_k z_\lambda \left(\frac{z_k^n - z_\lambda^n}{z_k - z_\lambda} \right), \quad k, \lambda = 1, \ldots, t. \tag{3.10}$$

que fuerzan a las coordenadas correspondientes a z_k, z_λ o bien a ser una de ellas nula o bien ambas no nulas y distintas. Por lo tanto

$$\mathcal{E} = \mathcal{V}\left(F \bigcup \left\{ z_k z_\lambda \left(\frac{z_k^n - z_\lambda^n}{z_k - z_\lambda} \right) \right\}_{k,\lambda=1}^{t} \right). \tag{3.11}$$

El cálculo directo de $\mathcal{E}$ con la variedad anterior puede ser computacionalmente muy costoso, P. Loustaunau y E.V. York [LY97] diseñaron una estrategia alternativa que sigue a continuación.

La observación clave es que el número de errores en el punto $\mathbf{p}$ son t menos el número de coordenadas nulas de $\mathbf{p}$ en las posiciones correspondientes a $\mathbf{z}$ y este número de ceros puede ser calculado mirando las diversas proyecciones de la variedad.

Definición 3.2. *Dado el conjunto $X \subseteq \mathbb{F}_{q^m}^b$, para cada $a \leq b$ definimos la* proyección *de X sobre las primeras a coordenadas como*

$$\prod_a (X) = \left\{ \mathbf{p} \in \mathbb{F}_{q^m}^a \mid \exists \mathbf{p}' \in \mathbb{F}_{q^m}^{b-a}, \text{ tal que } (\mathbf{p}, \mathbf{p}') \in X \right\} \tag{3.12}$$

Teorema 3.3. *Sea $\mathbf{0}_k \in \mathbb{F}_q^k$ el vector con todas sus coordenadas nulas. Dado $\tilde{\mathbf{c}}$ y su correspondiente síndrome $\mathbf{s}$, los errores cometidos en $\tilde{\mathbf{c}}$ son exactamente τ si y sólo si*

$$(\mathbf{s}, \mathbf{0}_k) \in \prod_{r+k} (\mathcal{V}(F)), \quad \forall k \leq t - \tau$$

y

$$(\mathbf{s}, \mathbf{0}_{t-\tau+1}) \notin \prod_{r+t-\tau+1} (\mathcal{V}(F)).$$

Demostración. Sea $\mathbf{s}$ un síndrome correspondiente a un error $\mathbf{e}$ con $w(\mathbf{e}) = \tau \leq t$. Entonces existe un punto $\mathbf{p}$ con una expresión como la mostrada en la ecuación (3.8), esto es $(\mathbf{s}, \mathbf{0}_k) \in \prod_{r+k}(\mathcal{V}(F))$ para todo $k \leq t - \tau$. Supongamos que

$$\mathbf{p}' = (\mathbf{s}, \mathbf{0}_{t-\tau+1}) \in \prod_{r+t-\tau+1} (\mathcal{V}(F)),$$

esto es, $\mathbf{p}'$ extiende a un punto

$$\mathbf{p}_0 = \big(\mathbf{p}', \gamma_1, \ldots, \gamma_{\tau-1}, \eta_1, \ldots, \eta_t\big) = \big(\mathbf{p}', \gamma, \eta\big) \in \mathcal{V}(F).$$

El vector γ no es el vector nulo (pues implicaría $\mathbf{s} = \mathbf{0}$), además no puede tener todas sus coordenadas distintas (pues representaría un error), y por lo tanto tendríamos dos vectores error distintos (con peso menor que t) asociados a un mismo síndrome, lo que es una contradicción. Por lo tanto, se tiene que existen i, j distintos con $\gamma_i = \gamma_j$ (esto es $\mathbf{p}_0 \notin \mathcal{E}$). Supondremos, sin pérdida de generalidad, que $\gamma_1 = \gamma_2$ y tomamos

$$\begin{aligned} \mathbf{p}_1 &= \big(\mathbf{p}', 0, \gamma_2, \ldots, \gamma_{\tau-1}, \eta_1, \eta_1 + \eta_2, \ldots, \eta_t\big) \text{ si } \eta_1 + \eta_2 \neq 0 \\ \mathbf{p}_1 &= \big(\mathbf{p}', 0, 0, \gamma_3, \ldots, \gamma_{\tau-1}, \eta_1, \eta_1, \ldots, \eta_t\big) \text{ en otro caso.} \end{aligned}$$

De nuevo $\mathbf{p}_1 \in \mathcal{V}(F)$. Podemos proseguir con esta construcción hasta que tengamos coordenadas no nulas en el vector γ que no se repiten, pero éstas representarán a un error con peso extrictamente menor que τ lo que es una contradicción, por lo tanto $(\mathbf{s}, \mathbf{0}_{t-\tau+1}) \notin \prod_{r+t-\tau+1}(\mathcal{V}(F))$.

Para demostrar la otra implicación supongamos que

$$(\mathbf{s}, \mathbf{0}_k) \in \prod_{r+k} (\mathcal{V}(F)), \quad \forall k \leq t - \tau$$

y $(\mathbf{s}, \mathbf{0}_{t-\tau+1}) \notin \prod_{r+t-\tau+1}(\mathcal{V}(F))$. Entonces $(\mathbf{s}, \mathbf{0}_{t-\tau})$ se extiende a un punto en $\mathcal{V}(F)$, y dicho punto puede ser relacionado con un único vector error de peso exactamente τ. □

Corolario 3.4. *Con la notación del teorema anterior consideremos el conjunto*

$$\Gamma = \{\mathbf{p} \in \mathcal{V}(F) \mid \mathbf{p} = (\mathbf{s}, \mathbf{0}_{t-\tau}, *, *, \ldots, *)\}. \tag{3.13}$$

Entonces el conjunto $\prod_{r+t-\tau+1}(\Gamma)$ *contiene exactamente* τ *puntos distintos de la forma* $(\mathbf{s}, \mathbf{0}_{t-\tau}, \alpha^{l_i})$*, y las posiciones del error correspondiente vienen dadas por* l_i *con* $i = 1, \ldots, \tau$.

Relacionaremos ahora los resultados anteriores con la teoría de la eliminación expuesta en la sección 1.5. Para ello necesitaremos el siguiente lema previo:

Lema 3.5.

$$\prod_{r+k} (\mathcal{V}(F)) = \mathcal{V}\left(I \cap \mathbb{F}_q[\mathbf{x}, z_1, \ldots, z_k]\right)$$

Demostración. La demostración se puede encontrar en el teorema 2.5.3 de [AL96]. □

Consideremos ahora un orden lexicográfico $\prec_{lex}$ en $\mathbb{F}_q[\mathbf{x}, \mathbf{z}, \mathbf{y}]$ tal que

$$x_1 \prec_{lex} \cdots \prec_{lex} x_r \prec_{lex} z_1 \prec_{lex} \cdots \prec_{lex} z_t \prec_{lex} y_1 \prec_{lex} \cdots \prec_{lex} y_t,$$

y sea G una base de Gröbner del ideal síndrome I respecto del orden $\prec_{lex}$, si tenemos en cuenta el teorema 1.30 entonces

$$G_k = G \cap \mathbb{F}_q[\mathbf{x}, z_1, \ldots, z_k] \tag{3.14}$$

es una base de Gröbner del ideal de eliminación $I \cap \mathbb{F}_q[\mathbf{x}, z_1, \ldots, z_k]$, esto es, podemos reinterpretar el teorema 3.3 y el corolario 3.4 de la siguiente forma:

Teorema 3.6. *Con la notación adoptada en el teorema 3.3, dado $\tilde{\mathbf{c}}$ y su correspondiente síndrome $\mathbf{s}$, los errores cometidos en $\tilde{\mathbf{c}}$ son exactamente τ si y sólo si para cada elemento $g \in G_k$ se tiene que*

$$g(\mathbf{s}, \mathbf{0}_k) = 0, \quad \forall k \leq t - \tau$$

y además existe un elemento $g \in G_{t-\tau+1}$ tal que

$$g(\mathbf{s}, \mathbf{0}_{t-\tau+1}) \neq 0.$$

Corolario 3.7. *Sea $G_{t-\tau+1} = \{g_1, \ldots, g_s\}$ y consideremos el vector $\xi_{t-\tau} = (\mathbf{s}, \mathbf{0}_{t-\tau}, z)$ donde z es una nueva variable. El ideal*

$$\langle G_{t-\tau+1}(\xi_{t-\tau})\rangle = \langle g_1(\xi_{t-\tau}), \ldots, g_s(\xi_{t-\tau})\rangle \subset \mathbb{F}_q[z] \tag{3.15}$$

está generado por el polinomio cuyas raíces localizan los errores, es más, dicho polinomio será uno de los polinomios evaluados

$$g_1(\xi_{t-\tau}), \ldots, g_s(\xi_{t-\tau}).$$

Demostración. La primera parte del corolario se sigue directamente del teorema anterior y del corolario 3.4. Para ver la segunda parte basta comprobar que $G_{t-\tau+1}(\xi_{t-\tau})$ es una base de Gröbner, lo que es cierto en el caso cero dimensional respecto a un orden puramente lexicográfico (ver [Gia89]). □

3.1.1. Descodificación por la variedad síndrome

En las condiciones del corolario anterior se podría calcular el polinomio localizador de errores con el ideal $\langle G_{t-\tau}(\xi_{t-\tau-1})\rangle$ pues este último ideal está generado por z multiplicado por el polinomio cuyas raíces localizan los errores. Esto se debe a que en este caso la variedad corresponde sólo a la proyección de los puntos de $\mathcal{E}$. Aquellos puntos en $\mathcal{V}(F) \setminus \mathcal{E}$ cuyas coordenadas en $\mathbf{x}$ corresponden a un síndrome $\mathbf{s}$ son aquellos puntos de $\mathcal{V}_{\mathbf{s}}$ donde dos o más coordenadas nulas en $\mathbf{z}$ fueron sustituidas por el mismo elemento de $\mathbb{F}_q$. Sin embargo, el ideal de eliminación $t-\tau$-ésimo sólo deja una variable en $\mathbf{z}$ nula por lo que la situación anterior no se puede dar. De la misma manera podemos obtener que el generador del ideal $\langle G_k(\xi_{k-1})\rangle$ es $z^{n+1} - z$ para todo $k < t - \tau$.

De la discusión anterior se sigue el siguiente **método de Loustaunau-York de descodificación** de un código cíclico dado un síndrome $\mathbf{s}$

1. Calcular los ideales de eliminación

$$G_k, \quad k = 1, \ldots, t.$$

   (Esto sólo requiere un único cálculo de una base de Gröbner aunque previsiblemente muy costoso si utilizamos directamente el algoritmo de Buchberger).

2. Evaluar los generadores de G_k sucesivamente en $\mathbf{x} = \mathbf{s}$ y comprobar si los términos independiente son nulos o no.

3. Cuando encontremos el primer término independiente no nulo calcular el polinomio cuyas raíces localizan los errores y factorizarlo.

Nota 3.8. *En todo este capítulo, y en concreto en el método de descodificación expuesto anteriormente suponemos conocida la distancia mínima del código cíclico, ésto en el caso general no es un problema trivial de resolver, en los trabajos de M. Sala [Sal02, OS05] se puede ver cómo realizar dicho cálculo utilizando la variedad síndrome.*

Ejemplo 3.9 ([Yor94] pág. 61)**.** *Consideremos el código BCH binario con parámetros* $[15, 5, 7]$*. Los polinomios que definen la variedad síndrome son*

$$\begin{aligned} F &= \{z_1+z_2+z_3+x_1,\, z_1^3+z_2^3+z_3^3+x_2,\, z_1^5+z_2^5+z_3^5+x_3,\\ &\quad z_1^{16}-z_1,\, z_2^{16}-z_2,\, z_3^{16}-z_3\} \end{aligned}$$

donde no se incluyen variables en $\mathbf{y}$ *pues estamos en el caso binario. Calculando una base de Gröbner para el orden lexicográfico con* $x_1 \prec x_2 \prec x_3 \prec z_1 \prec z_2 \prec z_3$ *tenemos los siguientes resultados (se omiten los polinomios que sólo involucran las variables* $\mathbf{x}$ *pues evaluados en un síndrome son nulos).*

$$\begin{aligned} G_1 &= \{z_1^{16}-z_1,\, z_1^3x_2+z_1^3x_1^3+z_1^2x_1x_2+z_1^2x_1^4+z_1x_3+z_1x_1^2x_2+\\ &\quad x_1x_3+x_2^2+x_1^3x_2+x_1^6,\, z_1^3x_3+z_1^3x_1^5+z_1^2x_1x_3+z_1^2x_1^6+\\ &\quad z_1x_2^9x_3^2+z_1x_1^3x_2^8x_3^2+z_1x_2^4x_3^2+z_1x_1^9x_2x_3^2+z_1x_1^2x_3+z_1x_1^{10}x_2^9+\\ &\quad z_1x_1^{13}x_2^8+z_1x_1^{10}x_2^4+z_1x_1^4x_2+z_1x_1^7+x_1x_2^9x_3^2+x_1^4x_2^8x_3^2+\\ &\quad x_1x_2^4x_3^2+x_1^{10}x_2x_3^2+x_2x_3+x_1^{11}x_2^9+x_1^{14}x_2^8+x_1^{11}x_2^4\}\\ G_2 &= G_1 \cup \{z_2^{16}-z_2,\, z_1z_2^2+z_2^2x_1+z_1^2z_2+z_2x_1^2+z_1^2x_1+z_1x_1^2+\\ &\quad x_2+x_1^3,\, z_2^2x_2+z_2^2x_1^3+z_1z_2x_2+z_1z_2x_1^3+z_2x_1x_2+z_2x_1^4+\\ &\quad z_1^2x_2+z_1^2x_1^3+z_1x_1x_2+z_1x_1^4+x_3+x_1^2x_2,\, z_2^2x_3+z_2^2x_1^5+\\ &\quad z_1z_2x_3+z_1z_2x_1^5+z_2x_1x_3+z_2x_1^6+z_1^2x_3+z_1^2x_1^5+z_1x_1x_3+\\ &\quad z_1x_1^6+x_2^9x_3^2+x_1^3x_2^8x_3^2+x_2^4x_3^2+x_1^9x_2x_3^2+x_1^2x_3+x_1^{10}x_2^9+\\ &\quad x_1^{13}x_2^8+x_1^{10}x_2^4+x_1^4x_2+x_1^7\}\\ G_3 &= G_2 \cup \{z_3+z_2+z_1+x_1\}. \end{aligned}$$

Sea α *el elemento primitivo del cuerpo* $\mathbb{F}_{16}$ *tal que* $\alpha^4+\alpha+1=0$*. Se envía la palabra* $\mathbf{0}$*.*

- *Finalmente, supongamos que se ha cometido un error en la segunda posición, el síndrome correspondiente es* $\mathbf{s}=(\alpha,\alpha^3,\alpha^5)$*, se tiene*

$$\begin{aligned} G_1(\xi_0) &= \{z^{16}+z\},\\ G_2(\xi_1) &= \{z^{16}+z,\, \alpha^2z+\alpha z^2\},\\ G_3(\xi_2) &= \{z+\alpha\}. \end{aligned}$$

Las variedades generadas por $G_1(\xi_0)$ *y* $G_2(\xi_1)$ *contienen a* 0 *y* $G_3(\xi_2)$ *nos da el polinomio cuya raiz localiza el error* $z+\alpha$. *Además como hacíamos notar anteriormente al principio de la sección* $G_2(\xi_1)$ *está generado por* $z(z+\alpha)$ *y* $G_1(\xi_0)$ *por* $z^{15+1}+z$.

- *Supongamos ahora que se cometen dos errores en las posiciones segunda y cuarta. El síndrome correspondiente es* $\mathbf{s}=(\alpha^9,\alpha,\alpha^{10})$ *y*

$$\begin{aligned} G_1(\xi_0) &= \{z^{16}+z,\ \alpha^{13}z^3+\alpha^7z^2+\alpha^2z,\ \alpha^5z^3+\alpha^{14}z^2+\alpha^9z\},\\ G_2(\xi_1) &= \{z^{16}+z,\ \alpha^9z^2+\alpha^3z+\alpha^{13},\\ &\quad \alpha^{13}z^2+\alpha^7z+\alpha^2,\ \alpha^5z^2+\alpha^{14}z+\alpha^9\}. \end{aligned}$$

De nuevo la variedad que genera $G_1(\xi_0)$ *contiene a* 0 *y los elementos distintos de* $z^{16}+z$ *en* $G_2(\xi_1)$ *factorizan como*

$$\begin{aligned} \alpha^9z^2+\alpha^3z+\alpha^{13} &= \alpha^9(z+\alpha)(z+\alpha^3)\\ \alpha^{13}z^2+\alpha^7z+\alpha^2 &= \alpha^{13}(z+\alpha)(z+\alpha^3)\\ \alpha^5z^2+\alpha^{14}z+\alpha^9 &= \alpha^5(z+\alpha)(z+\alpha^3) \end{aligned}$$

esto es, los localizadores del error son α *y* α^3.

- *Supongamos ahora que se cometen tres errores en las posiciones segunda, cuarta y séptima. El síndrome correspondiente es* $\mathbf{s}=(\alpha+\alpha^2,\alpha+\alpha^3,\alpha^5)$ *y*

$$G_1(\xi_0)=\{z^{16}+z,\ \alpha^7z^3+\alpha^{12}z^2+\alpha^8z+\alpha^2,\ z^3+\alpha^5z^2+\alpha z+\alpha^{10}\}$$

los elementos distintos de $z^{16}+z$ *factorizan como*

$$\begin{aligned} \alpha^7z^3+\alpha^{12}z^2+\alpha^8z+\alpha^2 &= \alpha^7(z^3+\alpha^5z^2+\alpha z+\alpha^{10})\\ &= \alpha^7(z+\alpha)(z+\alpha^3)(z+\alpha^6). \end{aligned}$$

Por lo tanto el polinomio cuyas raíces localizan los errores es $(z+\alpha)(z+\alpha^3)(z+\alpha^6)$.

3.1.2. Técnicas FGLM

Como ya dijimos en el capítulo primero, la complejidad del cálculo de una base de Gröbner mediante el algoritmo de Buchberger es doblemente

exponencial lo que resulta un grave inconveniente para la aplicación del algoritmo descrito en la sección anterior (el lector puede tratar de calcular la base de Gröbner en el ejemplo 3.9 en algún sistema de álgebra computacional al que tenga acceso mediante el algoritmo de Buchberger, ¿qué sucede?).

En esta sección describiremos someramente cómo nos puede ayudar en nuestro caso la técnica de reordenamiento o FGLM (de J.C. Faugère, P. Gianni, D. Lazard, T. Mora [FGLM93]) para ideales cero dimensionales. Dichas técnicas aprovechan la estructura lineal en el caso cero dimensional (ver sección 1.6) para calcular una base de Gröbner de un ideal para cierto orden monomial, supuesta conocida una base de Gröbner del ideal para otro orden monomial distinto, mediante álgebra lineal.

El siguiente lema cuya demostración se deja como un fácil ejercicio, nos proporciona el primer ingrediente, una base de Gröbner del ideal síndrome:

Lema 3.10. *El conjunto F de generadores del ideal síndrome $I \subset \mathbb{F}_q[\mathbf{x}, \mathbf{z}, \mathbf{y}]$ expresado en la ecuación (3.6) son una base de Gröbner de I respecto del orden lexicográfico con*

$$y_1 \prec_1 \ldots \prec_1 y_t \prec_1 z_1 \ldots \prec_1 z_t \prec_1 x_1 \ldots \prec_1 x_r.$$

El único inconveniente es que para realizar la descodificación de Loustaunau-York hemos de calcular la base de Gröbner respecto del orden lexicográfico dado por

$$x_1 \prec_2 \cdots \prec_2 x_r \prec_2 z_1 \prec_2 \cdots \prec_2 z_t \prec_2 y_1 \prec_2 \cdots \prec_2 y_t.$$

Dado que F es una base de Gröbner la aplicación

$$\begin{array}{rcl} r : \mathbb{F}_q[\mathbf{x}, \mathbf{z}, \mathbf{y}] & \longrightarrow & \mathbb{F}_q[\mathbf{x}, \mathbf{z}, \mathbf{y}]/I \\ f & \longmapsto & \overline{f}^F_{\prec_1} + I, \end{array}$$

es un homomorfismo de espacios vectoriales (ver sección 1.6). Además una base de dicho espacio vectorial son todos los monomios de la forma

$$\prod_{i=0}^{t} y^{a_i} z^{b_i}, \quad 0 \leq a_i \leq q-2,\, 0 \leq b_i \leq n,$$

y la dimensión de $\mathbb{F}_q[\mathbf{x},\mathbf{z},\mathbf{y}]/I$ como $\mathbb{F}_q$-espacio vectorial es $(n+1)^t(q-1)^t$. Ordenemos ahora los monomios de $\mathbb{F}_q[\mathbf{x},\mathbf{z},\mathbf{y}]$ conforme al orden monomial $\prec_2$, cualquier monomio $\mathbf{x}^{\mathbf{a}}\mathbf{z}^{\mathbf{b}}\mathbf{y}^{\mathbf{c}}$ es o bien reducido respecto de F, o bien un múltiplo de un término líder de F. Por lo tanto, cualquier polinomio $f = \sum_{i=0}^{\nu} \sigma_i \mathbf{x}^{\mathbf{a}_i}\mathbf{z}^{\mathbf{b}_i}\mathbf{y}^{\mathbf{c}_i}$ con término líder $\mathbf{x}^{\mathbf{a}_\nu}\mathbf{z}^{\mathbf{b}_\nu}\mathbf{y}^{\mathbf{c}_\nu}$ respecto a $\prec_2$ y $(\sigma_0,\ldots,\sigma_\nu) \in \mathbb{F}_q^\nu$ está en el ideal I si y sólo si

$$r(\sum_{i=0}^{\nu} \sigma_i \mathbf{x}^{\mathbf{a}_i}\mathbf{z}^{\mathbf{b}_i}\mathbf{y}^{\mathbf{c}_i}) = \sum_{i=0}^{\nu} \sigma_i r(\mathbf{x}^{\mathbf{a}_i}\mathbf{z}^{\mathbf{b}_i}\mathbf{y}^{\mathbf{c}_i}) = 0. \tag{3.16}$$

Esto es, el problema de encontrar qué términos líder respecto del orden monomial $\prec_2$ se reduce a calcular un elemento σ en el núcleo de una matriz $(\nu+1)\times(n+1)^t(q-1)^t$, además, como el ideal es cero dimensional, hay sólo un número finito de sistemas a resolver. También si incluimos un término líder no tenemos que considerar ya sus múltiplos (ver el artículo original [FGLM93] para un estudio más detallado). Este procedimiento se puede llevar a cabo algorítmicamente como sigue:

Algoritmo 3.11 (FGLM Loustaunau-York)**.**

Input: *F base de Gröbner respecto $\prec_1$ y un orden monomial $\prec_2$.*

Output: *G base de Gröbner reducida de $\langle F\rangle$ respecto $\prec_2$.*

1: LPP $\leftarrow$ [], $G \leftarrow$ [], Mbasis $\leftarrow$ []
2: PowerProducts $\leftarrow$ [], $\mathbf{x}_{(i)} \leftarrow 1$
3: ***while*** $\mathbf{x}_{(i)} \neq$ null ***do***
4: ***if*** $\mathbf{x}_{(i)}$ *no es divisible por algún elemento de* LPP ***then***
5: $r(\mathbf{x}_{(i)}) \leftarrow \overline{\mathbf{x}_{(i)}}^{F}_{\prec_1}$,
6: ***if*** *si existe una relación lineal* $r(\mathbf{x}_{(i)}) = \sum_{r(\mathbf{x}_{(j)})\in \text{Mbasis}} \sigma_j r(\mathbf{x}_{(j)})$ ***then***
7: $g_i \leftarrow \mathbf{x}_{(i)} - \sum_{r(\mathbf{x}_{(j)})\in \text{Mbasis}} \sigma_j r(\mathbf{x}_{(j)})$,
8: $G \leftarrow [g_i, G]$,
9: LPP $\leftarrow [\mathbf{x}_{(i)}, \text{LPP}]$
10: ***else***
11: Mbasis $\leftarrow [[\mathbf{x}_{(i)}, r(\mathbf{x}_{(i)})], \text{Mbasis}]$,
12: InsertNext$(\mathbf{x}_{(i)})$.
13: ***end if***

14: $\mathbf{x}_{(i)} \leftarrow$ NextMonom.
15: ***end if***
16: ***end while***

Donde las ***variables locales*** *son*

- PowerProducts*: lista de términos a ser considerados ordenados respecto al orden monomial* $\prec_2$.
- LPP*: Lista que contiene los términos líder de* G.
- Mbasis*: Lista de pares* $[\mathbf{x}_{(i)}, r(\mathbf{x}_{(i)})]$ *donde* $\mathbf{x}_{(i)}$ *es un elemento de la base de* $\mathbb{F}_q[\mathbf{x}, \mathbf{z}, \mathbf{y}]/\langle G \rangle$

y las ***subrutinas***

- NextMonom*: Elimina el primer elemento de la lista* PowerProducts *y retorna "null" si queda vacía.*
- InsertNext($\mathbf{x}_{(i)}$)*: Añade a la lista* PowerProducts *los productos* $x_1\mathbf{x}_{(i)}, \ldots, x_r\mathbf{x}_{(i)}$, $z_1\mathbf{x}_{(i)}, \ldots, z_t\mathbf{x}_{(i)}$, $y_1\mathbf{x}_{(i)}, \ldots, y_t\mathbf{x}_{(i)}$ *y los ordena respecto* $\prec_2$.

Para nuestro propósito no es necesario calcular la base de Gröbner completa pues sólo estamos interesados en G_k, esto es, se puede modificar InsertNext de forma que no incluya en la lista aquellos monomios que contengan algún y_i. Otras mejoras pueden incorporarse cuando el cuerpo base es $\mathbb{F}_2$ (ver [LY97, Yor94]). La complejidad del algoritmo anterior es $\mathcal{O}((n+1)^{3t+1})$.

3.2. Códigos de evaluación sobre variedades afines

J. Fitzgerald y R.F. Lax en [FL98] proponen la siguiente construcción de códigos de evaluación. Consideremos un ideal $I \subseteq \mathbb{F}_q[x_1, \ldots, x_s]$ y consideramos el ideal

$$I_q = I + \langle x_1^q - x_1, \ldots, x_s^q - x_s \rangle \,. \tag{3.17}$$

La variedad $\mathcal{V}(I_q)$ corresponde a los puntos $\mathbb{F}_q$-racionales de la variedad $\mathcal{V}(I)$ y claramente I_q es un ideal de dimensión cero (pues contiene a

$x_1^q - x_1, \ldots, x_s^q - x_s$) y radical. Sea

$$\mathcal{V}(I_q) = \{P_1, P_2, \ldots, P_n\}, \tag{3.18}$$

la aplicación de evaluación

$$\begin{array}{ccc} ev : \mathbb{F}_q[x_1, \ldots, x_s]/I_q & \longrightarrow & \mathbb{F}_q^n \\ f + I_q & \longmapsto & (f(P_1), f(P_2), \ldots, f(P_n)) \end{array} \tag{3.19}$$

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

Definición 3.12. *Sea L un $\mathbb{F}_q$-subespacio vectorial de $\mathbb{F}_q[x_1, \ldots, x_s]/I$. Definimos el código de evaluación $C(I_q, L)$ sobre la variedad afín $\mathcal{V}(I_q)$ como la imagen de L por la aplicación ev en la ecuación (3.19).*

Ejemplo 3.13. *Sea $I = \langle x^{q-1} - 1 \rangle \subset \mathbb{F}_q[x]$, (esto es $I_q = I$), y*

$$L = \operatorname{span}_{\mathbb{F}_q}\{1 + I, x + I, x^2 + I, \ldots, x^k + I\} \subseteq \mathbb{F}_q[x]/I.$$

$C(I, L)$ es el código Reed-Solomon de dimensión k sobre $\mathbb{F}_q$.

Una observación importante es que la clase de códigos que acabamos de definir incluyen todos los códigos lineales:

Teorema 3.14. *Todo código lineal $\mathcal{C}$ sobre el cuerpo $\mathbb{F}_q$ puede ser representado como un código de evaluación sobre una variedad afín.*

Demostración. Consideremos una matriz generatriz de $\mathcal{C}$ dada por $G = (g_{ij})$, $i = 1, \ldots, k$, $j = 1, \ldots, n$. Sea s un entero tal que $q^s > n$ y tomemos un conjunto de n puntos $\mathbf{P} = \{P_1, P_2, \ldots, P_n\} \subseteq \mathbb{F}_q^s$ con

$$P_j = (p_{j1}, p_{j2}, \ldots, p_{js}).$$

Consideremos $I = \mathcal{I}(\mathbf{P}) \subset \mathbb{F}_q[x_1, \ldots, x_s]$. Los polinomios

$$\chi_j(x_1, \ldots, x_s) = \prod_{l=1}^{s} \left(1 - (x_l - p_{jl})^{q-1}\right), \quad j = 1, \ldots, n \tag{3.20}$$

cumplen (ver [DGM70])

$$\chi_j(P) = \begin{cases} 0 \text{ si } P \in \mathbb{F}_q^s \setminus \{P_j\} \\ 1 \text{ si } P = P_j \end{cases}.$$

Consideremos las clases de equivalencia $\chi_j + I_q \in F_q[x_1, \ldots, x_s]/I_q$ para $j = 1, \ldots, n$ y construimos

$$f_i + I_q = \left[\sum_{j=1}^{n} g_{ij} \left(\chi_j + I_q \right) \right] \in F_q[x_1, \ldots, x_s]/I_q, \quad i = 1, \ldots, k.$$

Si tomamos $L = \mathrm{span}_{\mathbb{F}_q} \{f_i + I_q\}_{i=1}^{k}$ se tiene $\mathcal{C} = C(I_q, L)$. □

En la demostración hemos hecho referencia al cálculo del ideal $I = \mathcal{I}(\mathbf{P})$ para un conjunto de puntos $\mathbf{P}$, un algoritmo para dicho cálculo puede encontrarse en [MB82].

3.2.1. Descodificación mediante bases de Gröbner

Consideremos el código dual $\mathcal{C} = C(I_q, L)^{\perp}$ de un código de evaluación sobre la variedad afín $\mathcal{V}(I_q)$ con

$$\begin{aligned} I &= \langle g_1, g_2, \ldots, g_m \rangle \subset \mathbb{F}_q[x_1, \ldots, x_s] \\ L &= \mathrm{span}_{\mathbb{F}_q} \{f_1 + I_q, \ldots, f_r + I_q\}, \ f_i \in \mathbb{F}_q[x_1, \ldots, x_s], \ i = 1, \ldots, r \\ \mathcal{V}(I_q) &= \{P_1, P_2, \ldots, P_n\} \subset \mathbb{F}_q^s. \end{aligned}$$

Consideremos que hemos recibido la palabra $\mathbf{y} = (y_1, \ldots, y_n)$, su síndrome $\mathbf{s}$ viene dado por

$$s_i = \sum_{j=1}^{n} y_i f_i(P_j), \quad i = 1, \ldots, r. \tag{3.21}$$

Si $\mathbf{y} = \mathbf{e} + \mathbf{c}$ donde $\mathbf{c} \in \mathcal{C}$ y $w(\mathbf{e}) = t$ entonces

$$s_i = \sum_{j=1}^{n} e_i f_i(P_j), \quad i = 1, \ldots, r. \tag{3.22}$$

Consideremos el anillo de polinomios (notar ahora que $e_1, \ldots, e_t$ son indeterminadas)

$$T = \mathbb{F}_q[x_{11}, \ldots, x_{1s}, \ldots, x_{t1}, \ldots, x_{ts}, e_1, \ldots, e_t] \tag{3.23}$$

y los polinomios en T

$$h_i = \sum_{j=1}^{t} e_j f_i(x_{j1}, \ldots, x_{js}) - s_i, \quad i = 1, \ldots, r \tag{3.24}$$

donde los $s_i \in \mathbb{F}_q$ son las coordenadas del vector síndrome. Consideremos el ideal

$$E_{\mathbf{y}} = \left(\left\langle g_l(x_{j1}, \ldots, x_{js}), h_i, e_j^{q-1} - 1 \right\rangle \right)_q \tag{3.25}$$

con $i = 1, \ldots, r$, $j = 1, \ldots, t$ y $l = 1, \ldots, m$. Nótese que a pesar de la notación $E_{\mathbf{y}}$ el ideal es igual para todas las palabras de igual síndrome, esto es depende del síndrome.

Teorema 3.15. *Si han ocurrido exactamente t errores (t la capacidad correctora del código) en las posiciones correspondientes a P_{i_j} valores de error e_{i_j} $j = 1, \ldots, t$ entonces existen precisamente $t!$ puntos en la variedad $\mathcal{V}(E_{\mathbf{y}})$ correspondientes a*

$$\left\{ \left(P_{i_{\sigma(1)}}, \ldots, P_{i_{\sigma(t)}}, e_{i_{\sigma(1)}}, \ldots, e_{i_{\sigma(t)}} \right) \right\}_{\sigma \in \mathcal{S}_t} \tag{3.26}$$

donde $\mathcal{S}_t$ es el grupo simétrico de orden t.

Demostración. Dada la simetría de los polinomios en $E_{\mathbf{y}}$ es claro que los puntos del conjunto en (3.26) están en $\mathcal{V}(E_{\mathbf{y}})$. La existencia de otro punto en la variedad no contenido en (3.26) contradice el hecho de la existencia de un único vector error $\mathbf{e}$ de peso t. □

Consideremos ahora el orden monomial $\prec_1$ que extiende el orden lexicográfico

$$x_{11} \prec_1 x_{12} \prec_1 \ldots \prec_1 x_{1s} \prec_1 e_1$$

en las variables $x_{11}, x_{12}, \ldots, x_{1s}, e_1$ y sea $\prec_2$ cualquier otro orden monomial en el resto de las variables. Dados dos monomios en T los descompondremos como M_1N_1 y M_2N_2, donde M_1, M_2 involucran sólo las variables $x_{11}, x_{12}, \ldots, x_{1s}, e_1$ y N_1, N_2 el resto de variables. Definiremos el orden de eliminación $\prec$ (para las variables $x_{11}, x_{12}, \ldots, x_{1s}, e_1$) como

$$M_1N_1 \prec M_2N_2 \Longleftrightarrow \begin{cases} M_1 \prec_1 M_2 \\ \text{Si } M_1 = M_2 \text{ entonces } N_1 \prec_2 N_2. \end{cases} \tag{3.27}$$

Teorema 3.16. *Sea G una base de Gröbner para el ideal $E_{\mathbf{y}}$ respecto del orden monomial definido en (3.27) Se pueden calcular las posiciones del error y sus valores aplicando la eliminación en $E_{\mathbf{y}}$ sobre las variables $x_{11}, x_{12}, \ldots, x_{1s}, e_1$.*

Demostración. Consideremos los ideales de eliminación

$$\begin{aligned} J &= E_{\mathbf{y}} \cap \mathbb{F}_q[x_{11}, x_{12}, \ldots, x_{1s}, e_1] \\ J_i &= E_{\mathbf{y}} \cap \mathbb{F}_q[x_{11}, x_{12}, \ldots, x_{1i}],\ i = 1, \ldots, s. \end{aligned}$$

El conjunto $G \cap \mathbb{F}_q[x_{11}, x_{12}, \ldots, x_{1s}, e_1]$ es una base de Gröbner del ideal J respecto del orden $\prec_1$ (ver [AL96] Capítulo 2) y por lo tanto $G \cap \mathbb{F}_q[x_{11}, x_{12}, \ldots, x_{1i}]$ es una base de Gröbner del ideal J_i respecto del orden $\prec_1$ para cada $i = 1, \ldots, s$.

Consideremos $\{g_1(x_{11})\} = G \cap \mathbb{F}_q[x_{11}]$ el generador del ideal principal J_1. Las raíces del polinomio g_1 son las primeras coordenadas de los puntos en $\mathcal{V}(E_{\mathbf{y}})$. Sustituyendo cada una de dichas coordenadas en el conjunto $G \cap \mathbb{F}_q[x_{11}, x_{12}]$ obtenemos polinomios en la variable x_{12} que podemos resolver y así podemos continuar el proceso hasta encontrar todas las coordenadas correspondientes a las variables $x_{11}, x_{12}, \ldots, x_{1s}, e_1$ para cada punto en $\mathcal{V}(E_{\mathbf{y}})$. □

Ejemplo 3.17 ([FL98])**.** *Sea* $I = \langle y^2 + y - x^3 \rangle \subset \mathbb{F}_4[x, y]$ *y* $\mathbb{F}_4 = \mathbb{F}_2[\alpha]$ *con* $\alpha^2 = \alpha + 1$*. Los puntos de la variedad* $\mathcal{V}(I)$ *son*

$$\begin{array}{llll} P_1 = (0,0), & P_2 = (0,1), & P_3 = (1,\alpha), & P_4 = (1,\alpha^2), \\ P_5 = (\alpha,\alpha), & P_6 = (\alpha,\alpha^2), & P_7 = (\alpha^2,\alpha), & P_8 = (\alpha^2,\alpha^2). \end{array}$$

Consideremos el subespacio vectorial

$$L = \mathrm{span}_{\mathbb{F}_4} \left\{ 1 + I_4, x + I_4, y + I_4, x^2 + I_4, xy + I_4 \right\}.$$

Una matriz de control para el código $\mathcal{C} = (L, I_4)^{\perp}$ *es*

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

y su distancia mínima es 5. Si hemos recibido el vector $\mathbf{y} = (0, 0, 1, 0, 0, \alpha, 0, 0)$ *su síndrome es* $\mathbf{s} = (\alpha^2, \alpha, \alpha^2, 0, 0)$*. Consideremos el ideal* $E_{\mathbf{y}}$ *en* $\mathbb{F}_4[x_1, y_1, x_2, y_2, e_1, e_2]$ *generado por los polinomios*

$$\begin{aligned} & x_1^4 - x_1, y_1^4 - y_1, e_1^3 - 1, x_2^4 - x_2, y_2^4 - y_2, e_2^3 - 1, \\ & y_1^2 + y_1 - x_1^3, y_2^2 + y_2 - x_2^3, \\ & e_1 + e_2 - \alpha^2, e_1 x_1 + e_2 x_2 - \alpha, e_1 y_1 + e_2 y_2 - \alpha^2, e_1 x_1^2 + e_2 x_2^2, \\ & e_1 x_1 y_1 + e_2 x_2 y_2. \end{aligned}$$

Una base de Gröbner de $E_{\mathbf{y}}$ para el orden lexicográfico que extiende el orden $x_1 \prec y_1 \prec e_1 \prec x_2 \prec y_2 \prec e_2$ es

$$\begin{aligned} G &= \{x_1^2 + \alpha^2 x_1 + \alpha, y_1 + \alpha x_1, e_1 + x_1, x_2 + x_1 + \alpha^2, \\ &\quad y_2 + \alpha x_1 + 1, e_2 + x_1 + \alpha^2\}. \end{aligned}$$

Las primeras coordenadas de los puntos que localizan el error son las raíces del polinomio $x_1^2 + \alpha^2 x_1 + \alpha$ que son 1 y α. Sustituyendo las raíces en el polinomio $y_1 + \alpha x_1$ se obtienen las segundas coordenadas α y α^2 respectivamente, esto es corresponden a P_3 y P_6. Finalmente, de la ecuación $e_1 + x_1$ se obtiene que el valor del error en cada punto es la primera coordenada del punto.

3.2.2. Cálculo previo de los localizadores de errores

El algoritmo de descodificación anterior dista mucho de ser práctico pues requiere el cálculo de una base de Gröbner para cada síndrome distinto. Para evitar este problema J. Fitzgerald y R.F. Lax en [FL98] proponen añadir nuevas variables $s_1, \ldots, s_r$ al anillo de polinomios correspondientes a los síndromes. Ahora el cálculo de la base de Gröbner es más complejo (al ser un anillo de polinomios "mayor") pero tiene la ventaja de realizarse una única vez.

Con la notación de la sección anterior consideremos el anillo de polinomios

$$\mathcal{T} = T[s_1, \ldots, s_r] \tag{3.28}$$

donde T es el anillo de polinomios definido en la ecuación (3.23). Consideremos los polinomios en $\mathcal{T}$

$$h_i = \sum_{j=1}^{t} e_j f_i(x_{j1}, \ldots, x_{js}) - s_i, \quad i = 1, \ldots, r \tag{3.29}$$

donde, a diferencia de en la ecuación (3.24), ahora s_i $i = 1, \ldots, r$ son indeterminadas y, análogamente a la sección anterior definimos el ideal

$$\mathcal{E} = \left(\left\langle g_l(x_{j1}, \ldots, x_{js}), h_i, e_j^{q-1} - 1 \right\rangle \right)_q \subset \mathcal{T} \tag{3.30}$$

con $i = 1, \ldots, r$, $j = 1, \ldots, t$ y $l = 1, \ldots, m$. Análogamente a la discusión de la ecuación (3.9) en la sección 3.1 podemos ver la variedad $\mathcal{V}(\mathcal{E})$ como

la unión de las variedades $\mathcal{V}(E_{\mathbf{y}})$ para los distintos síndromes del código correspondientes a errores de peso t.

Sea $\prec_s$ un orden monomial cualquiera en las variables $s_1, \ldots, s_r$ y $\prec$ el orden monomial definido sobre T en la ecuación (3.27). Dados dos monomios en $\mathcal{T}$ los descompondremos como M_1N_1 y M_2N_2 donde M_1, M_2 involucran sólo las variables $s_1, \ldots, s_r$ y N_1, N_2 son monomios en T. Definiremos el orden monomial sobre $\mathcal{T}$ $\prec'$ como

$$M_1N_1 \prec' M_2N_2 \Longleftrightarrow \begin{cases} M_1 \prec_s M_2 \\ \text{Si } M_1 = M_2 \text{ entonces } N_1 \prec N_2. \end{cases} \tag{3.31}$$

Teorema 3.18. *Sea G una base de Gröbner para el ideal $\mathcal{E}$ con respecto del orden monomial $\prec'$ definido en (3.31) y supongamos se han cometido exactamente t errores. Se pueden calcular las posiciones del error y sus valores sustituyendo el valor del síndrome en las variables $s_1, \ldots, s_r$ y posteriormente utilizando eliminación.*

Demostración. Consideremos los ideales de eliminación

$$\begin{aligned} J &= \mathcal{E} \cap \mathbb{F}_q[s_1, \ldots, s_r, x_{11}, x_{12}, \ldots, x_{1s}, e_1] \\ J_i &= \mathcal{E} \cap \mathbb{F}_q[s_1, \ldots, s_r, x_{11}, x_{12}, \ldots, x_{1i}],\ i = 1, \ldots, s \\ J_0 &= \mathcal{E} \cap \mathbb{F}_q[s_1, \ldots, s_r]. \end{aligned}$$

De forma similar a la demostración del teorema anterior El conjunto $G \cap \mathbb{F}_q[s_1, \ldots, s_r, x_{11}, x_{12}, \ldots, x_{1s}, e_1]$ es una base de Gröbner del ideal J con respecto del orden $\prec'$ y $G \cap \mathbb{F}_q[s_1, \ldots, s_r, x_{11}, x_{12}, \ldots, x_{1i}]$ es una base de Gröbner del ideal J_i respecto del orden $\prec'$ para cada $i = 0, 1, \ldots, s$.

Si sustituimos las variables $s_1, \ldots, s_r$ por las coordenadas de un síndrome (correspondiente a un error de peso t) estamos obteniendo una solución parcial en $\mathcal{V}(J_0)$ de un punto de $\mathcal{V}(\mathcal{E})$. Como el número de soluciones es finito la proyección de $\mathcal{V}(\mathcal{E})$ sobre las variables $s_1, \ldots, s_r, x_{11}, x_{12}, \ldots, x_{1s}, e_1$ es $\mathcal{V}(J)$ y sobre las variables $s_1, \ldots, s_r, x_{11}, x_{12}, \ldots, x_{1i}$ es $\mathcal{V}(J_i)$. Se sigue, como en la demostración del teorema anterior, que la solución parcial se puede extender a todos los puntos en $\mathcal{V}(\mathcal{E})$ que se encuentran sobre ella (ver [CLO97], pág. 122–123). □

Ejemplo 3.19 ([FL98])**.** *Dado el mismo código que en el ejemplo anterior el ideal $\mathcal{E}$ en $\mathbb{F}_4[s_1, s_2, s_3, s_4, s_5, x_1, y_1, x_2, y_2, e_1, e_2]$ generado por*

los polinomios

$$x_1^4 - x_1, y_1^4 - y_1, e_1^3 - 1, x_2^4 - x_2, y_2^4 - y_2, e_2^3 - 1,$$
$$y_1^2 + y_1 - x_1^3, y_2^2 + y_2 - x_2^3,$$
$$e_1 + e_2 - s_1, e_1x_1 + e_2x_2 - s_2, e_1y_1 + e_2y_2 - s_3, e_1x_1^2 + e_2x_2^2 - s_4,$$
$$e_1x_1y_1 + e_2x_2y_2 - s_5$$

nos permite corregir cualquier combinación de dos errores. Una base de Gröbner calculada con Macaulay (ver [FL98]) para el orden $\prec'$ donde $\prec_s$ y $\prec_2$ se toman el orden monomial graduado reverso lexicográfico contiene 119 polinomios. Una inspección detallada de los mismos lleva a poder descodificar el código.

Nota 3.20. *Como el lector se habrá dado cuenta, el método anterior se convierte en altamente impracticable por la base de Gröbner a calcular (recuérdese que la complejidad de dicho cálculo es doblemente exponencial). El método teórico es similar al de la sección 3.1 , pero al generalizarse a un código lineal cualquiera no pueden utilizarse técnicas FGLM al no conocerse una base de Gröbner a priori del ideal como ocurre en el lema 3.5 para los códigos cíclicos.*

3.3. Transformada de Matson-Solomon ($\star$)

El lector habrá comprobado en la descodificación de los códigos cíclicos mediante eliminación que la transformada discreta de Fourier y el hecho que el código sea un ideal (no sólo un espacio vectorial como en el caso de los códigos de evaluación) forma parte importante del método de descodificación. En esta sección generalizaremos ambos conceptos y estudiadremos los códigos definidos como ideales de un álgebra semisimple en cuyo caso el papél jugado por la transformada discreta de Fourier correrá a cargo de la transformada de Matson-Solomon.

3.3.1. Códigos semisimples

La mayor parte de esta sección se puede encontrar en [Mar07]. En toda esta sección $\mathcal{A}$ será un álgebra semisimple conmutativa de dimensión finita sobre $\mathbb{F}_q$. Como $\mathbb{F}_q$ es perfecto el álgebra es separable, esto es el polinomio mínimo $m_a(x)$ de $a \in \mathcal{A}$ no tiene raíces múltiples.

Definición 3.21. *Un código semisimple sobre $\mathcal{A}$ es un subálgebra de $\mathcal{A}$.*

Ejemplo 3.22. *Un código cíclico puede ser visto como una subálgebra de*

$$\mathcal{A}_1 = \mathbb{F}_q[x]/\langle x^n - 1\rangle$$

con $\mathrm{mcd}(n, q) = 1$.

Citaremos ahora varias propiedades relativas a $\mathcal{A}$, para una descripción detallada ver [Chi95, Mar04]. Sea $\mathfrak{B} = \{b_1 = 1, \ldots, b_n\}$ una base de $\mathcal{A}$ y consideremos su tabla de multiplicación dada por

$$b_i b_j = \sum_{k=1}^{n} m_{i,k}(b_j, \mathfrak{B}) b_k \qquad 1 \leq i, j, k \leq n, \quad m_{i,k}(b_j, \mathfrak{B}) \in \mathbb{F}_q. \quad (3.32)$$

Los polinomios en el anillo $\mathbb{F}_q[x_1, \ldots, x_n]$

$$F = \left\{ x_i x_j - m_{i,1}(b_j, \mathfrak{B}) - \sum_{k=2}^{n} m_{i,k}(b_j, \mathfrak{B}) x_k \right\}_{2 \leq i \leq j \leq n} \cup \{x_1 - 1\} \quad (3.33)$$

se denominan *polinomios estructurales del álgebra $\mathcal{A}$.*

Proposición 3.23. *Sea $\mathcal{A}$ un álgebra semisimple conmutativa de dimensión finita n sobre $\mathbb{F}_q$ y F el conjunto de polinomios estructurales del álgebra $\mathcal{A}$, entonces*

$$\mathcal{A} \cong \mathbb{F}_q[x_1, \ldots, x_n]/I \quad (3.34)$$

donde I es el ideal generado por F. Además, F es una base de Gröbner de I respecto de un orden monomial compatible con el grado total y el ideal I es radical y cero dimensional.

Una demostración de la proposición anterior puede encontrarse en [Mar04]. El hecho de que los polinomios en F sean una base de Gröbner se sigue de forma fácil de su estructura. Es importante recalcar que la base de Gröbner se obtiene directamente de la tabla de multiplicación del álgebra y que para calcular cualquier otra base de Gröbner (o para calcular la tabla de multiplicación respecto de otra base del álgebra) se pueden utilizar técnicas FGLM [FGLM93] similares a las empleadas en el algoritmo 3.11 (ver [Mar04] para una discusión con mayor detalle).

Ejemplo 3.24. *Consideremos el siguente álgebra sobre el cuerpo* $\mathbb{F}_5$

$$\mathcal{A}_2 = \mathbb{F}_5[x, y, z]/J \tag{3.35}$$

donde el ideal J *es*

$$\begin{aligned} J = \langle & x + y + z + 1, y^2 + y + z + 1, yz^2 + 2z^3 + yz + z^2 - y, \\ & z^4 + z^3 - 2yz - y + 1 \rangle \end{aligned} \tag{3.36}$$

Notar que por conveniencia los generadores de J *forman una base de Gröbner con respecto del orden graduado reverso lexicográfico con* $x \prec y \prec z$ *y* J *está generado por*

$$\{(x-3)(x-1)(x-4), y^2 - x, z + x + y + 1\}.$$

Esto no es el caso general y el álgebra suele estar presentado mediante una tabla de multiplicación.
Una base de $\mathcal{A}_2$ *como espacio vectorial sobre* $\mathbb{F}_5$ *(ver capítulo 1, sección 1.6) viene dada por*

$$\mathfrak{B}_2 = \left\{x_1 = 1, x_2 = z^3, x_3 = z^2, x_4 = yz, x_5 = z, x_6 = y\right\} \tag{3.37}$$

y los polinomios estructurales del álgebra $\mathcal{A}_2$ *son*

$$\begin{aligned} F_2 = \{ & x_1 - 1, x_2^2 - (2x_2 - 2x_4 + 2x_3 - x_6 + x_5 - 2), \\ & x_2x_3 - (2x_2 + 2x_4 - 2x_3 + x_6 - x_5 + 1), \\ & x_2x_4 - (2x_2 + 2x_3 + x_6 + 2x_5 + 2), \\ & x_2x_5 - (-x_2 + 2x_4 + x_6 - 1), x_2x_6 - (-2x_2 - 2x_4 + x_3 + 2x_6 + 2), \\ & x_3^2 - (-x_2 + 2x_4 + x_6 - 1), x_3x_4 - (-2x_2 - 2x_4 + x_3 + 2x_6 + 2), \\ & x_3x_5 - (x_2), x_3x_6 - (-2x_2 - x_4 - x_3 + x_6), \\ & x_4^2 - (x_2 + x_4 - x_6), x_4x_5 - (-2x_2 - x_4 - x_3 + x_6), \\ & x_4x_6 - (-x_4 - x_3 - x_5), x_5^2 - (x_3) \\ & x_5x_6 - (x_4), x_6^2 - (-x_5 - x_6 - 1)\} \end{aligned} \tag{3.38}$$

En este caso los elementos entre paréntesis corresponden a las formas normales de los productos de la base en (3.37) respecto de la base de Gröbner en (3.36).

Tomemos la variedad $\mathcal{V}(I) = (P_1, P_2, \ldots, P_n)$, esto es, las raíces comunes de los polinomios en F (notar que se encontrarán en alguna extensión algebraica $\mathbb{F}$ de $\mathbb{F}_q$). Les denotaremos como vectores fila $P_i = (p_{i1}, \ldots, p_{in})$. Definimos la *matriz de Mattson-Solomon* asociada a la situación descrita como

$$M = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ p_{21} & \cdots & p_{2n} \\ & \ddots & \\ p_{n1} & \cdots & p_{nn} \end{pmatrix} \tag{3.39}$$

M es una matriz no singular y para cada $a(\mathbf{x}) \in \mathbb{F}[x_1, \ldots, x_n]/I$, la aplicación

$$\Theta : \mathbb{F}[x_1, \ldots, x_n]/I \longrightarrow \mathbb{F}[x_1, \ldots, x_n]/I$$

definida por

$$[\Theta(a)](\mathbf{x}) = \sum_{i=1}^{n} a(P_i)x_i \tag{3.40}$$

es la *transformada de Mattson-Solomon*. Si $\circ$ es la multiplicación de polinomios módulo el ideal I y $\star$ es el producto componente a componente

$$\Big(\sum a_i x_i\Big) \star \Big(\sum b_i x_i\Big) = \Big(\sum a_i b_i x_i\Big)$$

entonces $\Theta : (\mathbb{F}[x_1, \ldots, x_n]/I, +, \circ) \to (\mathbb{F}[x_1, \ldots, x_n]/I, +, \star)$ es un isomorfismo de anillos (ver por ejemplo [Chi95]).

La base de monomios asociada a un orden compatible con el grado total del ideal $I = \langle F \rangle$ son $\{x_1, x_2, \ldots, x_n\}$, por lo tanto, los elementos de $\mathbb{F}[x_1, \ldots, x_n]/I$ se pueden escribir como $\sum_{i=1}^{n} a_i x_i$, y la transformada de Mattson-Solomon es equivalente al producto de matrices $M \cdot \mathbf{a}^T$ donde $\mathbf{a} = (a_1, \ldots, a_n)$.

Ejemplo 3.25. *Esta construcción es análoga a la de los códigos cíclicos presentada en el capítulo 2 sección 2.4. En aquel caso M era la matriz de Fourier*

$$M_1 = \left(\alpha^{ij}\right)_{1 \le i,j \le n}$$

con α una raíz primitiva n-ésima de la unidad.

Ejemplo 3.26. *En el caso del ejemplo 3.24 la matriz de Mattson-Solomon es*

$$M_2 = \begin{pmatrix} 1 & 3 & 4 & 2 & 2 & 1 \\ 1 & 4 & 1 & 1 & 4 & 4 \\ 1 & -\alpha & 4-2\alpha & 2+\alpha & 1-\alpha & \alpha \\ 1 & +\alpha & 4+2\alpha & 2-\alpha & 1+\alpha & -\alpha \\ 1 & 2 & 4 & 1 & 3 & 2 \\ 1 & 3 & 4 & 1 & 2 & 3 \end{pmatrix} \tag{3.41}$$

donde α es una raíz de x^2+2 y $\mathbb{F}_5(\alpha) \cong \mathbb{F}_{5^2}$.

Podemos escoger una matriz de control de un código semisimple H seleccionando alguna de las filas de M (no aleatoriamente). Por ejemplo, si transformamos F a un orden de eliminación de la variable x_i podemos calcular el polinomio mínimo $m(x_i)$ para x_i cuyas raíces son los elementos de la columna de M que corresponden a x_i.

Más formalmente, si $\mathbb{F}$ es una extensión de $\mathbb{F}_q$, y tenemos una factorización del polinomio $m(x_i) = f_1 \cdot f_2 \cdot \ldots \cdot f_s$ en $\mathbb{F}[x_i]$, si una fila correspondiente a un valor ν en la entrada i se incluye en la matriz de control H, entonces existe un j $1 \leq j \leq s$ tal que $f_j(\nu) = 0$ y todas las filas cuya entrada i corresponda a raíces de f_j debe ser también incluida en H.

Ejemplo 3.27. *Consideremos la base de Gröbner F_2 en el ejemplo 3.24 y $\mathcal{G}_5$ una base Gröbner para $\langle F_2 \rangle$ en un orden lexicográfico con $x_5 < x_i$, $i \neq 2$, $1 \leq i \leq n$, y consideremos el ideal de eliminación*

$$\langle m_5(x_5) \rangle = \mathbb{F}_5[x_5] \cap \mathcal{G}_5$$

Se tiene que

$$m_5(x_5) = (x_5+1)(x_5+2)(x_5+3)(x_5^2+3x_5+3) \text{ en } \mathbb{F}_5[x_2].$$

Esto es, si la primera fila de la matriz M_2 se incluye en la matriz de control H también debe ser incluida la última pues ambas corresponden al factor x_5-2. Tenemos la misma situación con la tercera y cuarta fila correspondientes al factor $x_5^2+3x_5+3$. En una extensión $\mathbb{F}(\alpha)$ de $\mathbb{F}_2$ con α raíz de $x_5^2+3x_5+3$ estas dos últimas filas se pueden separar.

En el caso de tener un *elemento separador* en la base del álgebra $\mathcal{A}$, es decir, un elemento cuyo polinomio mínimo factoriza completamente en factores lineales, podemos elegir cualquier combinación de columnas. Obsérvese que ésto siempre es cierto si permitimos que el código esté sobre una extensión del cuerpo $\mathbb{F}_q$ suficientemente grande.

3.3.2. Ideal generador de un código semisimple

En esta sección generalizaremos la idea de polinomio generador y polinomio de control para un código cíclico al caso semisimple. Esto nos permitirá manipular los códigos semisimples sin necesidad de calcular su matriz de Mattson-Solomon.

Sea $\mathbb{F}$ una extensión de cuerpos de $\mathbb{F}_q$ y consideremos la variedad $\mathcal{V}(I')$ donde $I' = I(F')$ es el ideal generado por

$$F' = F \cup \{g(x_{i_1}), \ldots, g(x_{i_s})\}, \tag{3.42}$$

con F la base de Gröbner en (3.33) asociada al álgebra semisimple $\mathcal{A}$ y $g(x_{i_j})|m_{\mathbb{F}}(x_{i_j})$ un divisor del polinomio mínimo del elemento $b_{i_j} \in \mathcal{A}$ correspondiente a la variable x_{i_j} $1 \leq i_j \leq n$. Esto es, $\mathcal{V}(I')$ contiene (como puntos) las filas de alguna matriz de control H de un código sobre $\mathcal{A}$. Sea $\mathcal{C}$ el código en

$$\mathcal{A} \cong \mathbb{F}_q[x_1, \ldots, x_n]/I$$

cuya matriz de control es la submatriz de la matriz de Mattson-Solomon de $\mathcal{A}$ dada por los puntos de $\mathcal{V}(I')$ que llamaremos *variedad de control*. Llamaremos *ideal generador* $I_{\mathcal{C}}$ del código $\mathcal{C}$ al ideal generado por

$$F_{\mathcal{C}} = F \cup \{g_i(x_i)\}_{i=1}^{n} \tag{3.43}$$

donde $g_i(x_i)|m_{\mathbb{F}}(x_i)$ el polinomio mínimo de x_i (nótese que puede que $g_i(x_i)$ sea el polinomio mínimo para algún valor de i).

Como en la teoría de códigos cíclicos consideramos los códigos definidos por la preimagen por la aplicación Θ de una subálgebra formada por aquellos elementos que se anulan en algunas posiciones prefijadas en el codominio de Mattson-Solomon (véase sección 2.3.3). Ésto nos permite poder definir cotas del tipo BCH (véase [Mar07]).

Ejemplo 3.28. *Sea $\mathcal{A}_2$ el álgebra en el ejemplo 3.24. Los polinomios mínimos para los elementos de la base (3.37) sobre $\mathbb{F}_5$ son:*

$$\begin{aligned} m(x_2) =&(x_2+1)(x_2+2)(x_2+3)(x_2^2+2) \\ m(x_3) =&(x_3+1)(x_3+4)(x_3^2+2x_3+4) \\ m(x_4) =&(x_4+3)(x_4+4)(x_4^2+x_4+1) \\ m(x_5) =&(x_5+1)(x_5+2)(x_5+3)(x_5^2+3x_5+3) \\ m(x_6) =&(x_6+1)(x_6+2)(x_6+3)(x_6+4)(x_6^2+2) \end{aligned} \tag{3.44}$$

Sea $\mathcal{C}$ el código con ideal generador

$$F_{\mathcal{C}} = F \cup \{(x_6+1)(x_6+2)(x_6+3)(x_6+4), (x_3+1)\}$$

Mediante técnicas FGLM se puede calcular una base de Gröbner con respecto a un orden lexicográfico (véase [Mar04])

$$\begin{aligned} I_{\mathcal{C}} = \langle & x_4 - 2x_5 - 2x_6 - 1, x_3 + 1, x_2 - 2x_4 - x_6 + 2 \\ & x_1 - 1, x_6^2 + x_5 + x_6 + 1, x_5x_6 - x_4, x_5^2 + 1 \rangle \end{aligned} \tag{3.45}$$

por lo tanto la matriz de control de $\mathcal{C}$ corresponde a las filas 1,5 y 6 de la matriz M_2 del ejemplo 3.26. Nótese que la elección de los $g_i(x_i)$ no es única, por ejemplo $F \cup \{(x_5^2+1)\}$ define el mismo código. Sin embargo el ideal $I_{\mathcal{C}}$ es único.

3.3.3. Codificación y descodificación

Una construcción para un codificador sistemático de un código semi-simple $\mathcal{C}$ puede hacerse como sigue (véase en [Lit98] una construcción similar). Sea G una base de Gröbner de $I_{\mathcal{C}}$ generado por $F_{\mathcal{C}}$ para un orden compatible con el grado total $\prec$ y el conjunto

$$\Delta_{\prec}(I) = \{\mathbf{x^a} \text{ monomio en } F_q[x_1, \ldots, x_n] \mid \mathbf{x^a} \text{ no es polinomio líder en } I\}.$$

1. Las posiciones de la información son los monomios en $\Delta_{\prec}(I) \setminus \Delta_{\prec}(I_{\mathcal{C}})$.
2. Las posiciones de control corresponden a $\Delta_{\prec}(I_{\mathcal{C}})$.
3. Para codificar $c \in \mathcal{C}$

- **Input**: c una combinación lineal de elementos de $\Delta_\prec(I) \setminus \Delta_\prec(I_\mathcal{C})$.
- Calcula $w = \overline{c}^G_\prec$.
- **Output**: $c - w$.

Ejemplo 3.29. *Consideremos el código dado por $I_\mathcal{C}$ en (3.45). $\Delta_\prec(I) \setminus \Delta_\prec(I_\mathcal{C}) = \{x_2, x_3, x_4\}$. Si queremos codificar $c = 3x_2 + 3x_3 + 2x_4$ calculamos*

$$\overline{3x_2 + 3x_3 + 2x_4}^G_\prec = x_5 - x_6 - 1$$

y la palabra correspondiente es $x_1 + 3x_2 + 3x_3 + 2x_4 - x_5 + x_6$. (pues $x_1 = 1$).

Dado un código semisimple $\mathcal{C}$ definido mediante su ideal generador $I_\mathcal{C}$ tenemos que si hemos recibido la palabra $r = \sum_{i=1}^n r_i x_i$ los síndromes de r son

$$s_j = \sum_{i=1}^{n} r_i P_{ji} \tag{3.46}$$

donde j recorre las filas permitidas de la matriz de Mattson-Solomon (esto es, la variedad $\mathcal{V}(I_\mathcal{C})$). Supongamos han ocurrido t errores y sea G una base de Gröbner de $I_\mathcal{C}$. Consideremos los polinomios

$$f_j = \sum_{l=1}^{t} y_l x_l - z_j$$

y $\sigma_j = z_j^{q^m} - z_j$, $\lambda_i = y_i^{q-1} - 1$ para índices adecuados dependiendo de la extensión de cuerpos donde se defina $\mathcal{C}$. El conjunto de polinomios

$$\{f_j, \sigma_j, \lambda_i\}_{i,j} \cup G \tag{3.47}$$

define el análogo a la variedad síndrome para el caso de códigos semisimples.

Capítulo 4

La ecuación clave

En el capítulo 2 presentamos un método para descodificar códigos BCH mediante sucesiones recurrentes, el método de Berlekamp-Massey. Además, probamos que los polinomios síndrome, localizador y evaluador del error verifican una ecuación denominada ecuación clave.

En este capítulo presentamos una solución de la ecuación clave, utilizando bases de Gröbner sobre módulos, y mostramos cómo dicha solución nos permite descodificar códigos BCH. Este método de descodificación fue desarrollado por P. Fitzpatrick [Fit95, Fit97].

Presentamos la estructura algebraica denominada módulo, puesto que trabajaremos con módulos definidos sobre un anillo de polinomios. Existe una fuerte analogía entre las bases de Gröbner en anillos y las definidas en módulos sobre un anillo de polinomios; este fenómeno nos va a permitir obtener los resultados de forma inmediata, una vez definidos los conceptos asociados a las bases de Gröbner para módulos.

La solución de la ecuación clave de P. Fitzpatrick tiene una complejidad similar a la del algoritmo de Berlekamp-Massey. La solución de la ecuación clave $(E(X), L(X))$ será el elemento mínimo de una base de Gröbner en $\mathbb{F}_q[X]^2$ con respecto a cierto orden monomial.

Este capítulo sigue en parte los capítulos 5 y 9 de [CLO05] y [Fit95, Fit97].

4.1. Bases de Gröbner sobre módulos

Definición 4.1. *Un módulo M sobre un anillo A (conmutativo con elemento neutro) es un conjunto dotado de una operación $+$ y una multiplicación escalar que verifican las siguiente propiedades*

1. $(M, +)$ *es un grupo abeliano.*
2. $a(f+g) = af + ag$
3. $(a+b)f = af + bf$
4. $(ab)f = a(bf)$
5. $1f = f$

para todo $a, b \in R$, $f, g \in M$.

Ejemplo 4.2. *El ejemplo más sencillo de módulo es A^m, siendo A un anillo.*

Como a lo largo de las notas, $R = \mathbb{K}[x_1, \ldots, x_n] = \mathbb{K}[\mathbf{x}]$ es el anillo de polinomios sobre el cuerpo $\mathbb{K}$. En esta sección trabajaremos siempre con módulos definidos sobre el anillo de polinomios R, concretamente con R^m, $m \geq 1$, y sus submódulos.

Consideramos notación multi-índice, es decir $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_s^{\alpha_s}$ en $\mathbb{K}[x_1, \ldots, x_n] = \mathbb{K}[\mathbf{x}]$, con $\alpha = (\alpha_1, \ldots, \alpha_s)$. Además, usaremos negrita para los elementos de R^m. Para saber si estamos considerando un exponente de $\mathbf{x}$ en notación multiíndice o un elemento de R^m, fijamos la siguiente notación: $\mathbf{f}$, $\mathbf{g}$ para elementos de R^m; a, b para polinomios de R; α, β, γ (en notación multiíndice $\mathbf{x}^\alpha, \mathbf{x}^\beta, \mathbf{x}^\gamma$) para los exponentes de los monomios de R; y c para los elementos de $\mathbb{K}$. Además, indicaremos en cada momento si cada símbolo representa un elemento de R^m, R o $\mathbb{K}$.

Un submódulo de un módulo M sobre R es un subconjunto de R^m que es cerrado para la suma y el producto por elementos de R, es decir, aquellos subconjuntos que son módulos en sí mismos. Denotamos por $\langle \mathbf{f}_1, \ldots, \mathbf{f}_s \rangle \subset M$ el submódulo generado por $\mathbf{f}_1, \ldots, \mathbf{f}_s$, es decir, el conjunto de combinaciones lineales de la forma $a_1\mathbf{f}_1 + \cdots + a_s\mathbf{f}_s$, con $a_1, \ldots, a_s \in R$.

Una base de un módulo es un conjunto de generadores linealemente independientes. Un espacio vectorial siempre tiene una base, en cambio, para módulos esta propiedad no se cumple en general. A los módulos que tienen base se los denomina libres. Por ejemplo, R^m es un módulo libre, una base -denominada base estándar de R^m- es $\mathbf{e}_1 = (1, 0, \ldots, 0)$, $\mathbf{e}_2 = (0, 1, 0, \ldots, 0)$, ..., $\mathbf{e}_m = (0, , \ldots, 0, 1)$.

En esta sección introducimos los órdenes monomiales en los anillos libres R^m, $m \geq 1$, y las bases de Gröbner para submódulos $M \subset R^m$. Al igual que en el caso de anillos e ideales, presentado en el capítulo 1, las bases de Gröbner permiten resolver el problema de pertenencia a un submódulo, es decir, cuándo $\mathbf{f} \in R^m$ pertenece a $M \subset R^m$, y dar un sistema de generadores que se comporta "bien" para la división.

El proceso para desarrollar bases de Gröbner en R^m será dar una definición "adecuada" de monomio y orden monomial (y sus conceptos asociados). Una vez que se tengan estos dos conceptos el resto de definiciones y resultados se obtendrá de forma inmediata por su analogía con las bases de Gröbner de ideales en anillos.

Definición 4.3. *Decimos que* $\mathbf{m} \in R^m$ *es un monomio si* $\mathbf{m} = \mathbf{x}^\alpha \mathbf{e}_i$, *con* $\mathbf{x}^\alpha$, $i \in \{1, \ldots, m\}$.

De esta forma todo elemento $f \in R^m$ se escribe de forma única como combinación $\mathbb{K}$-lineal de monomios. Cada sumando de la combinación lineal, es decir, el producto $c\mathbf{m}$ de un elemento de $\mathbb{K}$ por un monomio se denomina término y $c \in \mathbb{K}$ se denomina coeficiente.

Sean $\mathbf{m} = \mathbf{x}^\alpha \mathbf{e}_i, \mathbf{n} = \mathbf{x}^\beta \mathbf{e}_j$ dos monomios. Decimos que $\mathbf{n}$ divide a $\mathbf{m}$, y lo denotamos por $\mathbf{n}|\mathbf{m}$, si $i = j$ y x^β divide a $\mathbf{x}^\alpha$. Por tanto, el cociente $\mathbf{m}/\mathbf{n} = \mathbf{x}^{\alpha-\beta}$ es un elemento de R. En este caso, el máximo común divisior y el mínimo común múltiplo de $\mathbf{m}$ y $\mathbf{n}$, $\mathrm{mcd}(\mathbf{m}, \mathbf{n})$ y $\mathrm{mcm}(\mathbf{m}, \mathbf{n})$, son $\mathrm{mcd}(\mathbf{x}^\alpha, \mathbf{x}^\beta) \cdot \mathbf{e}_i$ y $\mathrm{mcm}(\mathbf{x}^\alpha, \mathbf{x}^\beta) \cdot \mathbf{e}_i$ (respectivamente). En caso contrario ($i \neq j$), se definen el máximo común divisor y el mínimo común múltiplo como $\mathbf{0}$.

Al igual que con anillos e ideales, para desarrollar la teoría de bases de Gröbner sobre módulos necesitaremos definir órdenes monomiales, construir el algoritmo de división y extender el algoritmo de Buchberger para módulos.

La definición de orden monomial en R^m es la extensión natural de la definición para anillos (véase definición 1.4), para $m = 1$ ambas definiciones coinciden.

Definición 4.4. *Un orden monomial sobre los monomios de R^m es un orden total $\prec$ que verifica:*

1. $\mathbf{e}_i \prec \mathbf{x}^\alpha \mathbf{e}_i$ *para todo monomio* $\mathbf{x}^\alpha$ *de* R *distinto de* 1. *Esta condición es equivalente a que* $\prec$ *sea un buen orden.*
2. *Si dos monomios de* R^m *verifican* $\mathbf{m} \prec \mathbf{n}$ *entonces* $\mathbf{x}^\gamma \mathbf{m} \prec \mathbf{x}^\gamma \mathbf{n}$ *para todo monomio* $\mathbf{x}^\gamma$ *de* R.

Los dos órdenes monomiales habitualmente utilizados en R^m son una extensión de los órdenes monomiales de R. Para ello debemos considerar un orden monomial en R (véase 1.7 y 1.8) y fijar un orden en la base estándar de R^m, nosotros consideraremos $\mathbf{e}_1 \succ \mathbf{e}_2 \succ \cdots \succ \mathbf{e}_m$.

Definición 4.5. *(Orden TOP-Term Over Position, término sobre la posición). Sea* $\prec$ *un orden monomial en* R *y sean* $\mathbf{x}^\alpha \mathbf{e}_i$ *y* $\mathbf{x}^\beta \mathbf{e}_j$ *dos monomios de* R^m. *Decimos que* $\mathbf{x}^\alpha \mathbf{e}_i \succ_{\mathrm{TOP}} \mathbf{x}^\beta \mathbf{e}_j$ *si* $\mathbf{x}^\alpha \succ \mathbf{x}^\beta$, *o si* $\mathbf{x}^\alpha = \mathbf{x}^\beta$ *y* $i < j$.

Definición 4.6. *(Orden POT-Position Over Term, posición sobre el término). Sea* $\prec$ *un orden monomial en* R *y sean* $\mathbf{x}^\alpha \mathbf{e}_i$ *y* $\mathbf{x}^\beta \mathbf{e}_j$ *dos monomios de* R^m. *Decimos que* $\mathbf{x}^\alpha \mathbf{e}_i \succ_{\mathrm{POT}} \mathbf{x}^\beta \mathbf{e}_j$ *si* $i < j$, *o si* $i = j$ *y* $\mathbf{x}^\alpha \succ \mathbf{x}^\beta$.

Ejercicio 4.7. *Demostrar que los órdenes definidos en 4.5 y 4.6 son órdenes monomiales.*

Sea $\mathbf{f} \in R^m$. Podemos escribir $\mathbf{f}$ como $\mathbf{f} = \sum c_i \mathbf{m}_i$, con $0 \neq c_i \in \mathbb{K}$. Llamamos término líder de $\mathbf{f}$ con respecto al orden monomial $\prec$ y lo denotamos por $\mathrm{lt}_\succ(\mathbf{f})$ al término $c_j \mathbf{m}_j$ donde $\mathbf{m}_j$ es el mayor monomio que aparece en $\mathbf{f}$ para el orden monomial $\prec$. Análogamente, definimos el coeficiente líder de $\mathbf{f}$ como $\mathrm{lc}_\succ(\mathbf{f}) = c_i$ y el monomio líder de $\mathbf{f}$ como $\mathrm{lm}_\succ(\mathbf{f}) = \mathbf{m}_i$.

Ejemplo 4.8. *En este ejemplo comprobamos cómo, para un mismo elemento, se tienen términos líderes diferentes para una extensión POT y TOP. Consideramos* R^2, *con* $R = \mathbb{F}_q[x, y]$. *Sean*

- $\mathbf{f}_1 = (y^2, x^2 - y) = y^2 \mathbf{e}_1 + (x^2 - y)\mathbf{e}_2 \in \mathbb{F}_q[x, y]^2$
- $\mathbf{f}_2 = (x^2 + 1, y^3) = (x^2 + 1)\mathbf{e}_1 + y^3 \mathbf{e}_2 \in \mathbb{F}_q[x, y]^2$

Consideramos $\mathbf{e}_1 \succ \mathbf{e}_2$*, el orden lexicográfico en* $\mathbb{F}_q[x, y]$ *con* $x \succ y$*. Entonces se tiene que*

- $\mathrm{lt}_{\succ_{\mathrm{POT}}}(\mathbf{f}_1) = y^2\mathbf{e}_1$
- $\mathrm{lt}_{\succ_{\mathrm{POT}}}(\mathbf{f}_2) = (x^2)\mathbf{e}_1$
- $\mathrm{mcm}(y^2\mathbf{e}_1, (x^2+1)\mathbf{e}_1) = (x^2y^2 + y^2)\mathbf{e}_1$

En cambio, si consideramos la extensión TOP, tenemos

- $\mathrm{lt}_{\succ_{\mathrm{TOP}}}(\mathbf{f}_1) = (x^2)\mathbf{e}_2$
- $\mathrm{lt}_{\succ_{\mathrm{TOP}}}(\mathbf{f}_2) = (x^2)\mathbf{e}_1$
- $\mathrm{mcm}((x^2 - y)\mathbf{e}_2, (x^2+1)\mathbf{e}_1) = \mathbf{0}$

En R^m también se tiene un algoritmo de división que extiende el algoritmo de división multivariable (ver proposición 1.10 y algoritmo 1.11).

Proposición 4.9. *Sea* $\succ$ *un orden monomial en* R^m *y sea* $\mathbf{f}_1, \ldots, \mathbf{f}_s$ *una s-upla ordenada de elementos de* R^m*. Entonces* $\mathbf{f} \in R^m$ *puede expresarse de la forma siguiente*

$$\mathbf{f} = a_1\mathbf{f}_1 + \cdots + a_s\mathbf{f}_s + \mathbf{r}$$

donde $a_i \in R$ *y el resto* $\mathbf{r} \in R^m$ *es o bien* $\mathbf{0}$*, o bien una combinación lineal de monomios no divisibles por los monomios* $\{\mathrm{lm}(\mathbf{f}_i)\}_{i=1}^s$*.*

Omitimos la prueba con su algoritmo debido a que es idéntica a la presentada en el capítulo 1 (ver algoritmo 1.11).

Al igual que en anillos, el resultado de la división depende del orden monomial elegido y de la ordenación de la s-upla $\mathbf{f}_1, \ldots, \mathbf{f}_s$. El algoritmo de división por sí solo tampoco resuelve el problema de pertenencia a un submódulo $M = \langle \mathbf{f}_1, \ldots, \mathbf{f}_s \rangle$, como muestra el ejemplo 4.10. Si al dividir $\mathbf{f}$ por $\mathbf{f}_1, \ldots, \mathbf{f}_s$ obtenemos resto $\mathbf{r} = \mathbf{0}$ entonces $\mathbf{f} \in M$, pero el recípropo no es cierto en general. Para tener el resultado recíproco necesitaremos unos generadores especiales de M, la denominada base de Gröbner de un submódulo.

Ejemplo 4.10. *Consideramos el módulo* $\mathbb{F}_q[x,y]^2$ *y* $\mathbf{f}_1 = (y^2, x^2 - y)$, $\mathbf{f}_2 = (x^2+1, y^3)$ *como en el ejemplo 4.8. Sea* $\mathbf{f}_3 = (0, x^4 - x^2y + x^2 - y^5 - y) \in \mathbb{F}_q[x,y]^2$.

Consideramos $\mathbf{e}_1 \succ \mathbf{e}_2$, *el orden lexicográfico en* $\mathbb{F}_q[x,y]$ *con* $x \succ y$, *y la extensión POT. La división de* $\mathbf{f}_3$ *por* $\mathbf{f}_1, \mathbf{f}_2$ *produce*

$$\mathbf{f}_3 = 0\mathbf{f}_1 + 0\mathbf{f}_2 + \mathbf{r}$$

donde $\mathbf{r} = \mathbf{f}_3$ *puesto que el término líder de* $\mathbf{f}_3$ *no es divisible por los términos líderes de* $\mathbf{f}_1$ *y de* $\mathbf{f}_2$.

En cambio, si consideramos el orden TOP y dividimos $\mathbf{f}_3$ *por* $\mathbf{f}_1, \mathbf{f}_2$ *tenemos que*

$$\mathbf{f}_3 = (x^2+1)\mathbf{f}_1 - y^2\mathbf{f}_2 + \mathbf{0}$$

y por tanto $\mathbf{f}_3$ *pertenece al submódulo de* $\mathbb{F}_q[x,y]^2$ *generado por* $\{\mathbf{f}_1, \mathbf{f}_2\}$.

Definición 4.11. *Para un orden monomial* $\prec$ *y un submódulo* $M \subset R^m$, *decimos que el conjunto* $\{\mathbf{f}_1, \ldots, \mathbf{f}_s\} \subset R$ *es una base de Gröbner si*

$$\langle \mathrm{lt}_\succ(\mathbf{f}_1), \ldots, \mathrm{lt}_\succ(\mathbf{f}_s)\rangle = \langle \mathrm{lt}_\succ(M)\rangle$$

donde $\langle \mathrm{lt}_\succ(M)\rangle$ *es el submódulo generado por los términos líderes de los elementos de* M *con respecto a* $\prec$.

Todas las propiedades del capítulo 1 para bases de Gröbner son válidas también para submódulos de R^m con idénticas demostraciones (una demostración detallada del primer apartado de la siguiente demostración puede encontrarse en [CLO05], proposición 5.2.3). En particular, las bases de Gröbner permiten resolver el problema de pertenencia a un submódulo, proporcionando un sistema de generadores.

Proposición 4.12. *Sea* $\prec$ *un orden monomial,* G *una base de Gröbner de un submódulo* $M \subset R^m$ *y* $\mathbf{f} \in R^m$. *Entonces se tiene que*

1. M *es finitamente generado, equivalentemente, los submódulos de* R^m *verifican la condición de cadena ascendente.*
2. $\mathbf{f} \in M$ *si y sólo si el resto* $\mathbf{r}$ *de la división por* G *es cero.*
3. G *genera* M *como módulo.*

Aunque se utiliza la terminología de base de Gröbner para el sistema de generadores G anteriormente definido (por su analogía con el caso de anillos e ideales), cabe destacar que G no es una base del submódulo M en general, es decir, los elementos de una base de Gröbner de M son un sistema de generadores pero no tienen por qué ser linealmente independientes.

Finalmente, extendemos el algoritmo de Buchberger de anillos (ver sección 1.4). Para ello definimos los S-polinomios y obtenemos el criterio y el algoritmo de Buchberger.

Definición 4.13. *(S-polinomio) Sean* $\mathbf{f}, \mathbf{g} \in R^m$ *y* $\prec$ *un orden monomial. El S-polinomio de* $\mathbf{f}$ *y* $\mathbf{g}$ *es*

$$S(\mathbf{f}, \mathbf{g}) = \frac{\mathbf{m}}{\mathrm{lt}_{\succ}(\mathbf{f})} \cdot \mathbf{f} - \frac{\mathbf{m}}{\mathrm{lt}_{\succ}(\mathbf{g})} \cdot \mathbf{g}$$

donde $\mathbf{m} = \mathrm{mcm}(\mathrm{lm}_{\succ}(\mathbf{f}), \mathrm{lm}_{\succ}(\mathbf{g}))$.

Proposición 4.14. *(Criterio de Buchberger) Sea* $\prec$ *un orden monomial y sea* M *un submódulo de* R^m. *Un sistema de generadores* $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_s\}$ *de* I *es una base de Gröbner de* M *si y sólo si*

$$\overline{S(\mathbf{g}_i, \mathbf{g}_j)}^{G}_{\succ} = \mathbf{0}, \quad 1 \leq i, j \leq s, \quad i \neq j$$

donde para $\mathbf{f} \in R^m$, $\bar{\mathbf{f}}^{G}_{\succ}$ *es el resto de la división de* $\mathbf{f} \in R^m$ *por* G *para el orden monomial* $\prec$.

La demostración del resultado anterior es la misma que la del criterio de Buchberger en anillos (proposición 1.20). Además, tenemos un algoritmo, completamente análogo al de ideales, que calcula una base de Gröbner de un submódulo en un número finito de pasos .

Algoritmo 4.15. *(Algoritmo de Buchberger)*

Input: $F = \{\mathbf{f}_1, \ldots, \mathbf{f}_s\}$ *con* $M = \langle \{\mathbf{f}_i\}_{i=1}^{s} \rangle$ *distinto de cero y* $\succ$ *un orden monomial.*

Output: *Una base de Gröbner para el módulo* M *con respecto al orden monomial* $\succ$.

1: $G \leftarrow F$, $G' \leftarrow \{0\}$

2: ***while*** $G \neq G'$ ***do***
3: $G' \leftarrow G$
4: ***for*** *cada par* $\{\mathbf{f}, \mathbf{g}\} \subset G'$ *con* $\mathbf{f} \neq \mathbf{g}$ ***do***
5: $S \leftarrow \overline{S(\mathbf{f}, \mathbf{g})}^{G'}_{\succ}$
6: ***if*** $S \neq 0$ ***then***
7: $G \leftarrow G \cup \{S\}$
8: ***end if***
9: ***end for***
10: ***end while***

Ejemplo 4.16. *Consideramos el módulo* $\mathbb{F}_q[x,y]^2$ *y* $\mathbf{f}_1 = (y^2, x^2 - y)$, $\mathbf{f}_2 = (x^2+1, y^3)$ *como en los ejemplos 4.8 y 4.10. También consideramos* $\mathbf{e}_1 \succ \mathbf{e}_2$, *el orden lexicográfico en* $\mathbb{F}_q[x,y]$ *con* $x \succ y$, *y el orden monomial en* $\mathbb{F}_q[x,y]^2$ *que da la extensión POT. Sea* $M = \langle \mathbf{f}_1, \mathbf{f}_2 \rangle$. *El S-Polinomio de* $\mathbf{f}_1$ *y* $\mathbf{f}_2$ *es*

$$S(\mathbf{f}_1, \mathbf{f}_2) = (x^2)\mathbf{f}_1 - y^2\mathbf{f}_2 = (-y^2, x^4 - x^2y - y^5)$$

Luego, como vimos en el ejemplo 4.10, $\{\mathbf{f}_1, \mathbf{f}_2\}$ *no es una base de Gröbner para* $\prec_{\mathrm{POT}}$ *puesto que* $\overline{S(\mathbf{f}_1, \mathbf{f}_2)}^{G}_{\succ_{\mathrm{POT}}} \neq \mathbf{0}$. *Siguiendo el algoritmo de Buchberger, consideramos como generadores de* M $\{\mathbf{f}_1, \mathbf{f}_2, S(\mathbf{f}_1, \mathbf{f}_2)\}$. *El lector puede comprobar fácilmente que la clase de cada S-polinomio de dos generadores es igual a* $\mathbf{0}$, *y por tanto* $\{\mathbf{f}_1, \mathbf{f}_2, S(\mathbf{f}_1, \mathbf{f}_2)\}$ *es una base de Gröbner de* M *para* $\succ_{\mathrm{POT}}$.

En cambio, si consideramos la extensión TOP del orden monomial en $\mathbb{F}_q[x,y]^2$, *tenemos que*

$$S(\mathbf{f}_1, \mathbf{f}_2) = \mathbf{0}$$

puesto que $\mathrm{mcm}(\mathrm{lm}_{\succ_{\mathrm{TOP}}}(\mathbf{f}_1), \mathrm{lm}_{\succ_{\mathrm{TOP}}}(\mathbf{f}_1)) = \mathbf{0}$ *(ver ejemplo 4.8). Por tanto, siguiendo el criterio de Buchberger, tenemos que* $\{\mathbf{f}_1, \mathbf{f}_2\}$ *es una base de Gröbner para* $\succ_{\mathrm{TOP}}$.

Fijado un orden monomial, se pueden definir bases de Gröbner minimales y reducidas de la misma forma que para ideales de anillos. Igualmente, se tiene que existe una única base de Gröbner reducida para cada orden monomial.

El siguiente resultado muestra cuándo R^m/M tiene dimensión finita como espacio vectorial sobre $\mathbb{K}$ para un submódulo $M \subset R^m$.

Proposición 4.17. *Sea M un submódulo de R^m y $\prec$ un orden monomial en M. El $\mathbb{K}$-espacio vectorial R^m/M tiene dimensión finita si y sólo si para todo $i \in \{1, \ldots, m\}$ y para todo $j \in \{1, \ldots, n\}$, existe $\mathbf{f} \in M$ tal que $\mathrm{lm}_{\succ}(\mathbf{f}) = x_j^u \mathbf{e}_i$, para algún $a \geq 0$.*

Demostración. Sea G una base de Gröbner de M para el orden $\prec$. Los elementos de R^m/M son combinaciones lineales de monomios pertenecientes al complementario de $\langle \mathrm{lt}_{\succ}(M) \rangle$, al igual que para ideales en anillos. Y se tiene el resultado, puesto que el número de monomios en el complmentario de $\langle \mathrm{lt}_{\succ}(M) \rangle$ es finito si y sólo si se verifica la condición del enunciado. □

Ejemplo 4.18. *Sean $R^2 = \mathbb{F}_q[X]^2$ y el submódulo $M = \langle X^u \mathbf{e}_1, X^v \mathbf{e}_2 \rangle$, con u, v enteros no nulos. Entonces R^m/M es un $\mathbb{K}$-espacio vectorial de dimensión finita.*

4.2. Solución de la ecuación clave

Sea $\mathcal{C}$ un código BCH sobre $\mathbb{F}_q$ de longitud n y distancia $\delta = 2t + 1$. De la misma forma que en el capítulo 2, sea α una raíz n-ésima de la unidad, y consideremos el código determinado por las raíces α, ..., $\alpha^{\delta-1}$. En la sección 2.4.2 hemos definido el polinomio localizador de errores $L(X)$ y el polinomio evaluador $E(X)$ para una palabra recibida de un código BCH. Éstos nos permiten saber en qué posiciones ha ocurrido un error durante la transmisión y cuál es el valor del error en cada posición (respectivamente).

Una vez calculados los polinomios ($E(X)$ y $L(X)$) podemos descodificar una palabra recibida $\mathbf{y} = \mathbf{c} + \mathbf{e}$, puesto que las raíces $\alpha^{-i}, i \in I$ del polinomio localizador $L(X)$ en $\mathbb{F}_q^*$ muestran las posiciones erróneas de la palabra recibida ($e_i = 0, i \notin I$). Una vez conocidas las posiciones del error la proposición 2.47 permite obtener los valores del error, es decir, $\mathbf{e}$. Por tanto, podemos descodificar $\mathbf{y}$ si calculamos $E(X)$ y $L(X)$.

El teorema 2.48 proporciona la siguiente igualdad, denominada ecuación clave,

$$E(X) \equiv L(X)s(X) \pmod{X^{2t}}, \tag{4.1}$$

donde la distancia mínima prevista del código BCH es $\delta = 2t + 1$ y podemos corregir t errores. En la ecuación clave $\delta - 1 = 2t$ y $s(X)$ son

conocidos a partir de la palabra recibida, mientras que $E(X)$ y $L(X)$ no lo son a priori.

En esta sección se propone un método, alternativo al propuesto en la sección 2.4.2, para descodificar códigos BCH. Dicho método utiliza bases de Gröbner sobre módulos, en concreto sobre $\mathbb{F}_q[X]^2$, y fue desarrollado por P. Fitzpatrick [Fit95, Fit97]. El siguiente resultado muestra que, bajo ciertas condiciones, la solución de la ecuación clave (4.1) es única (considerando $E(X)$ y $L(X)$ como indeterminadas).

Teorema 4.19. *Sea $\mathcal{C}$ un código BCH con $\delta = 2t+1$ y sea $s(X)$ un polinomio síndrome asociado a una palabra recibida que tiene a lo sumo t errores. Salvo multiplicación por escalar existe una única solución $(\overline{E}(X), \overline{L}(X))$ de la ecuación clave que verifica que $\overline{E}(X)$ y $\overline{L}(X)$ son relativamente primos (no tienen factores comunes),* $\deg(\overline{L}(X)) \leq t$ *y* $\deg(\overline{E}(X)) < \deg(\overline{L}(X))$.

Demostración. Sean $(E(X),L(X))$ y $(\overline{E}(X),\overline{L}(X))$ dos soluciones de la ecuación clave 4.1 que verifican las condiciones del enunciado, es decir, se tiene

$$E(X) \equiv L(X)s(X) \pmod{X^{2t}}.$$
$$\overline{E}(X) \equiv \overline{L}(X)s(X) \pmod{X^{2t}}.$$

Multiplicamos la primera ecuación por $\overline{L}(X)$, la segunda por $L(X)$ y restamos ambas ecuaciones para obtener

$$\overline{E}(X)L(X) \equiv E(X)\overline{L}(X) \pmod{X^{2t}}.$$

Debido a la hipótesis sobre los grados de las soluciones, se tiene que $\overline{E}(X)L(X)$ y $E(X)\overline{L}(X)$ tienen grado menor o igual que $2t-1$. Por tanto,

$$\overline{E}(X)L(X) = E(X)\overline{L}(X)$$

y puesto que $E(X)$, $L(X)$ y $\overline{E}(X)$, $\overline{L}(X)$ son relativamente primos se tiene el resultado. □

Ejercicio 4.20. *Demostrar que los polinomios localizador y evaluador del error definidos en el capítulo 2 verifican las condiciones del teorema 4.19. Por lo tanto, la solución de la ecuación clave que verifica las condiciones del teorema 4.19 nos proporciona los polinomios localizador y evaluador del error (salvo multiplicación por constante).*

El método que describimos en este capítulo calcula esta solución particular de la ecuación clave que resuelve el problema de la descodificación. Para ello definimos el siguiente submódulo de $\mathbb{F}_q[X]^2$,

Definición 4.21. *Sean $t \in \mathbb{N}$ y $s \in \mathbb{F}_q[X]$, definimos el módulo $M \subset \mathbb{F}_q[X]^2$*

$$M = \{(E(X), L(X)) \mid E(X) \equiv L(X)s(X) \pmod{X^{2t}}\}$$

Ejercicio 4.22. *Demostrar que M es un submódulo de $\mathbb{F}_q[X]^2$.*

Vamos a calcular una base de Gröbner de M para resolver la ecuación clave con respecto a un orden determinado que definimos a continuación. Para un valor de $r \in \mathbb{Z}$ concreto tendremos que uno de los miembros de la base de Gröbner es la solución buscada.

Definición 4.23. *Sean $r \in \mathbb{Z}$ y $X^\alpha \mathbf{e}_i$, $X^\beta \mathbf{e}_j \in \mathbb{F}_q[X]^2$. Decimos que*

1. *$X^\alpha \mathbf{e}_i \succ_r X^\beta \mathbf{e}_j$ si $\alpha > \beta$, con $i = j$*
2. *$X^\alpha \mathbf{e}_2 \succ_r X^\beta \mathbf{e}_1$ si $\alpha + r \geq \beta$*

Ejercicio 4.24.

- *Demostrar que $\succ_r$ es un orden monomial, para todo $r \in \mathbb{Z}$.*
- *Demostrar que $\succ_0$ y $\succ_{-1}$ son órdenes monomiales que coinciden con un orden TOP.*

Consideramos $\mathbf{g}_1 = (X^{2t}, 0)$, $\mathbf{g}_2 = (s(X), 1) \in \mathbb{F}_q[X]^2$, se tiene que $\mathbf{g}_1, \mathbf{g}_2 \in M$ puesto que verifican la ecuación clave (4.1)

$$X^{2t} \equiv 0 \pmod{X^{2t}}$$

$$s \equiv s \pmod{X^{2t}}$$

Además, $\mathbf{g}_1$ y $\mathbf{g}_2$ generan M: sea $(E(X), L(X)) \in M$

$$(E(X), L(X)) = \frac{E(X) - L(X)s(X)}{X^{2t}}(X^{2t}, 0) + L(S, 1)$$

puesto que $(E(X) - L(X)s(X))/X^{2t} \in \mathbb{F}_q[X]$. Se tiene que $\mathbb{F}_q[X]^2/M$ es un $\mathbb{F}_q$-espacio vectorial de dimensión finita en virtud de la proposición 4.17, porque

$$\mathrm{lt}_{\succ \deg(s(X))}((X^{2t}, 0) = (X^{2t}, 0) = X^{2t}\mathbf{e}_1$$

$$\mathrm{lt}_{\succ_{\deg(s(X))}}(s(X),1) = (0,1) = \mathbf{e}_2.$$

El siguiente resultado nos da un criterio para saber si dos elementos son una base de Gröbner de M para el orden monomial $\succ_r$, con $r \in \mathbb{Z}$.

Proposición 4.25. *Consideramos $M \subset \mathbb{F}_q[X]^2$ y $r \in \mathbb{Z}$. Puesto que $\mathbb{F}_q[X]^2/M$ es un $\mathbb{F}_q$-espacio vectorial de dimensión finita, existen $\mathbf{x}^u\mathbf{e}_1$, $\mathbf{x}^v\mathbf{e}_2 \in M$ que generan M (ver proposición 4.17). Un conjunto $G = \{\mathbf{g}_1, \mathbf{g}_2\} \subset M$ es una base de Gröbner reducida de M para el orden $\succ_r$ si $\mathbf{g}_1 = g_{11}(X)\mathbf{e}_1 + g_{12}(X)\mathbf{e}_2$ y $\mathbf{g}_2 = g_{21}(X)\mathbf{e}_1 + g_{22}(X)\mathbf{e}_2$ verifican*

1. $\mathrm{lt}_{\succ_r}(\mathbf{g}_1) = \mathbf{x}^u\mathbf{e}_1$, $\mathrm{lt}_{\succ_r}(\mathbf{g}_2) = \mathbf{x}^v\mathbf{e}_2$
2. $\deg(g_{21}(X)) < u$, $\deg(g_{12}(X)) < v$.

Demostración. Sea G un subconjunto de M que verifica las condiciones de la proposición. Por la primera condición se tiene que los términos líderes de G generan $\langle \mathrm{lt}_{\succ_r}(M)\rangle$, por lo que forman una base de Gröbner. La segunda condición asegura que la base es reducida puesto que $\deg(g_{12}(X)) + r < u$ y $\deg(g_{21}(X)) \leq v + r$.

Recíprocamente, si G es una base de Gröbner de M para $\succ_r$, debe verificar la primera condición del enunciado y si, además, es la base de Gröbner reducida, debe verificar la segunda condición. □

Corolario 4.26. *Se tiene que $g_1 = (X^{2t}, 0)$, $g_2 = (s, 1)$ es la base de Gröbner reducida de M para el orden $\prec_{\deg(s(X))}$.*

La solución de la ecuación clave va a ser el elemento mínimo, concepto que definimos a continuación, con respecto de un determinado orden monomial de una base de Gröbner.

Definición 4.27. *Sea M un submódulo de $\mathbb{F}_q[X]^2$, decimos que $\mathbf{m} \in M$ no nulo es un elemento mínimo de M con respecto a un orden monomial $\succ$ si $\mathrm{lt}_{\succ}(\mathbf{m})$ es minimal con respecto a $\succ$.*

Proposición 4.28. *Sea M un submódulo de $\mathbb{F}_q[X]^2$. Se tiene que un elemento mínimo de M es único salvo multiplicación por escalar. Además, cada base de Gröbner de M contiene un elemento mínimo.*

Demostración. Sean $\mathbf{f}_1$ y $\mathbf{f}_2$ elementos mínimos de M, se tiene que el resto de la división de $\mathbf{f}_1$ por $\mathbf{f}_2$ no puede ser menor que $\mathrm{lt}(\mathbf{f}_2)$, por lo que

el resto es $(0,0)$. De la misma forma, el resto al dividir $\mathbf{f}_2$ por $\mathbf{f}_1$ es $(0,0)$. Por tanto, su cociente es una constante y se tiene la primera parte del enunciado.

La segunda parte del enunciado se deduce trivialmente de la definición de base de Gröbner, puesto que una base de Gröbner es un sistema de generadores del módulo. □

Proposición 4.29. *Sea* $\mathbf{g} = (\overline{E}(X), \overline{L}(X))$ *la única (salvo multiplicación por escalar) solución de la ecuación clave que satisface las condiciones del teorema 4.19. Entonces* $\mathbf{g} = (\overline{E}(X), \overline{L}(X))$ *es un elemento mínimo de* M *para* $\succ_{-1}$.

Demostración. Un elemento $\mathbf{g} = (\overline{E}(X), \overline{L}(X)) \in M$ verifica $\deg(\overline{E}(X)) < \deg(\overline{L}(X))$ si y sólo si su monomio líder con respecto a $\succ_{-1}$ es un múltiplo de $\mathbf{e}_2$. La solución de la ecuación clave que verifica las condiciones del teorema 4.19 tiene esta propiedad y grado de $\overline{L}(X)$ mínimo, por tanto su término líder es mínimo con respecto a los elementos cuyo término líder es un múltiplo de $\mathbf{e}_2$.

Razonamos por reducción al absurdo, suponiendo que $\mathbf{g} = (\overline{E}(X), \overline{L}(X))$ no es minimal en M, es decir existe $\mathbf{f} = (a(X), b(X)) \in M$ no nulo tal que $\mathrm{lt}_{\succ_{-1}}(\mathbf{f}) \prec_{-1} \mathrm{lt}_{\succ_{-1}}(\mathbf{g})$. Por la discusión anterior, el monomio líder de $\mathbf{f}$ debe ser un múltiplo de $\mathbf{e}_1$. Por tanto

$$\deg(\overline{L}(X)) > \deg(a(X)) \geq \deg(b(X)) \tag{4.2}$$

Como $\mathbf{g}$ y $\mathbf{f}$ son soluciones de la ecuación clave (4.1), se tiene

$$\overline{E}(X) \equiv s(S)\overline{L}(X) \pmod{X^{2t}}$$

$$a(X) \equiv s(S)b(X) \pmod{X^{2t}}$$

Multiplicando la primera ecuación por $b(X)$ y la segunda por $\overline{L}(X)$ y restándolas

$$\overline{L}(X)a(X) \equiv \overline{E}(X)b(X) \pmod{X^{2t}} \tag{4.3}$$

Como $\mathbf{g} = (\overline{E}(X), \overline{L}(X))$ es la solución del teorema 4.19, se verifica $\deg(\overline{L}(X)) \leq t$ y $\deg(\overline{E}(X)) < \deg(\overline{L}(X))$. Por tanto, de (4.2) se tiene que $\deg(a(X)) \leq t-1$, lo cuál contradice la ecuación (4.3), puesto que el primer término de la congruencia tiene grado menor o igual que $2t-1$ y el segundo término tiene grado estrictamente menor que $2t-1$.

□

De la proposición anterior deducimos que, para resolver la ecuación clave, debemos calcular una base de Gröbner de M para el orden $\succ_{-1}$. Para ello, hay dos formas de proceder:

1. Consideramos $\{(s(X),1),(X^{2t},0)\}$ como generadores de M y calculamos una base de Gröbner para el orden $\succ_{-1}$ usando el algoritmo de Buchberger (o una de sus modificaciones que aumentan la eficiencia del algoritmo). Una vez calculada la base de Gröbner para el orden $\succ_{-1}$, la solución de la ecuación clave es el elemento mínimo de la base de Gröbner calculada con respecto a $\succ_{-1}$.

2. Consideramos $\{(s(X),1),(X^{2t},0)\}$ como base de Gröbner de M para el orden $\succ_{\deg(s)}$. Usando técnicas FGLM [FGLM93], podemos obtener una base de Gröbner de M para el orden $\succ_{-1}$. Igualmente, el elemento mínimo de la base de Gröbner obtenida es la solución de la ecuación clave.

El primer método es más eficiente para códigos "grandes". En [CLO05] (proposición 9.4.19 y ejercicio 9.5) puede encontrarse, de forma detallada, la segunda alternativa para descodificar (usando FGLM). Nosotros presentamos un ejemplo del primer método, calculando directamente una base de Gröbner. Además, ilustramos los cálculos usando el sistema de álgebra polinomial SINGULAR [GPS05].

Ejemplo 4.30. *Consideramos el código, la palabra transmitida y el error del ejemplo 2.52. En dicho ejemplo se descodifica usando el método de Berlekamp-Massey; en este ejemplo descodificaremos la palabra recibida mediante la resolución de la ecuación clave descrita en este capítulo.*

Sean $q=3, n=8$ *y* α *una raíz del polinomio* $2+X+X^2$. *El código BCH de longitud* 8 *y distancia mínima prevista* $\delta=5$ *sobre* $\mathbb{F}_3$ *tiene polinomio generador* $g(X)=2+X^2+X^3+2X^4+X^5 \sim 201121$ *y dimensión* $k=3$. *Codificamos el mensaje fuente* $\mathbf{m}=010 \sim X$,

$$m(X)g(X)=2X+X^3+X^4+2X^5+X^6 \sim 02011210=\mathbf{c}.$$

Supongamos, al igual que en el ejemplo 2.52, que durante la transmisión de esta palabra se produce el error $\mathbf{e}=10000100 \sim 1+X^5$, *con lo que el mensaje recibido es*

$$\mathbf{c}+\mathbf{e}=\mathbf{y}=12011010 \sim 1+2X+X^3+X^4+X^6$$

Realizamos los cálculos usando SINGULAR*:*

```
                    SINGULAR
 A Computer Algebra System for Polynomial Computations

     by: G.-M. Greuel, G. Pfister, H. Schoenemann
FB Mathematik der Universitaet, D-67653 Kaiserslautern
>ring R=(3^2,alpha),X,(lp,c);
> minpoly;
1*alpha^2+1*alpha^1+2*alpha^0
```

Consideramos el anillo de polinomios R en una variable sobre $\mathbb{F}_3[\alpha]$, con α una raíz del polinomio X^2+X+2. Consideramos el orden $\prec_{-1}$ sobre R^2, dicho orden viene dado por `(lp,c)` *puesto que es un orden TOP. Para mayor información sobre* SINGULAR *puede consultarse [GP02].*

```
> [X^2,0]>[X,0];
1 //1 indica si, 0 indica no
> [X,0]>[0,X];
1
> [0,X]>[1,0];
1
> [1,0]>[0,1];
1
```

Para la palabra recibida $\mathbf{y}$ *vamos a aplicar el método de resolución de la ecuación clave. El síndrome de la palabra recibida es*

$$s(X) = \sum_{i=0}^{4} \mathbf{y}(\alpha^i)x^i = (2\alpha+1)+(2\alpha+2)x+(\alpha+2)x^2 = \alpha^2+\alpha^3X+\alpha^6X^2$$

en este ejemplo escribimos los elementos de $\mathbb{F}_q^*$ como potencia del elemento primitivo α.

```
> poly y=1+2*X+X^3+X^4+X^5;
> poly s0=subst(y,X,alpha);  poly s1=subst(y,X,alpha2);
> poly s2=subst(y,X,alpha3); poly s3=subst(y,X,alpha4);
> poly s = s0 + (s1)*X + (s2)*X^2 + (s3)*X^3;
> s;
```

```
alpha6*X2+alpha3*X+alpha2
> (2alpha +1) + (2alpha +2)*X + (alpha +2)*X^2; //Ej 2.52
alpha6*X2+alpha3*X+alpha2
```

Definimos el módulo M a partir de los generadores $\mathbf{g}_1 = (X^{2t}, 0)$, $\mathbf{g}_2 = (s(X), 1) \in \mathbb{F}_q[X]^2$. Tenemos dos formas de representar un módulo: a partir de la base estándar o escribiéndolo como vector columna.

```
> vector g1=[X4,0];
> vector g2=[s,1];
> module M=g1,g2;
> M;
M[1]=X4*gen(1)
M[2]=alpha6*X2*gen(1)+alpha3*X*gen(1)+alpha2*gen(1)+gen(2)
> print(M);
X4,alpha6*X2+alpha3*X+alpha2,
0, 1
```

Calculamos la base de Gröbner reducida de M para el orden $\prec_{-1}$. El primer generador de la base de Gröbner $(X+\alpha^5, X^2+\alpha X+\alpha^3)$ es menor que el segundo elemento $(X^2+\alpha^5 X+2, \alpha^2)$, por tanto $(\overline{E}(X), \overline{L}(X)) = (X+\alpha^5, X^2+\alpha X+\alpha^3)$ es la solución de la ecuación clave que verifica las condiciones del teorema 4.19.

```
> option(redSB);
> module G=std(M);
> G;
G[1]=X2*gen(2)+X*gen(1)+alpha*X*gen(2)+alpha5*gen(1)+alpha3*gen(2)
G[2]=X2*gen(1)+alpha5*X*gen(1)-gen(1)+alpha2*gen(2)
> print(G);
X+alpha5,          X2+alpha5*X-1,
X2+alpha*X+alpha3,alpha2
> G[1]<G[2];
1
> poly E=G[1][1];
> poly L=G[1][2];
> E;
X+alpha5
```

```
> L;
X2+alpha*X+alpha3
```

Las posiciones del error vienen determinadas por los raíces del polinomio $L(X) = X^2 + \alpha X + \alpha^3$*, para calcularlas podemos factorizar* $L(X)$*, o hacer una búsqueda exhaustiva. Obtenemos que* α^3 *y* α^8 *son las raíces de* $\overline{L}(X)$*. Por tanto, se tiene que los errores se han producido en las posiciones* 0 *y* 5 *de la palabra recibida (o equivalentemente en el término independiente y en* X^5*), puesto que* $\eta_1 = \alpha^0 = (\alpha^8)^{-1}$ *y* $\eta_2 = \alpha^5 = (\alpha^3)^{-1}$*.*

```
> int ii; for(ii=1;ii<=8;ii++)
{"exponente",ii,"valor de L",subst(L,X,alphaˆii);}
exponente 1 valor de L 1
exponente 2 valor de L alpha
exponente 3 valor de L 0
exponente 4 valor de L alpha
exponente 5 valor de L alpha3
exponente 6 valor de L -1
exponente 7 valor de L -1
exponente 8 valor de L 0
> 1/alpha3;
alpha5
> 1/alpha8;
1
```

Una vez determinadas las posiciones del error, podemos obtener los valores del error mediante el polinomio evaluador $\overline{E}(X) = X + \alpha^5$*, haciendo uso de la proposición 2.47.*

$$e_0 = \frac{-E(\alpha^8)}{L'(\alpha^8} = \frac{\alpha^6}{\alpha^6} = 1$$

$$e_5 = \frac{-E(\alpha^3)}{L'(\alpha^3} = \frac{\alpha^2}{\alpha^2} = 1$$

```
> poly LP=2*X + alpha;
> LP;
-X+alpha
```

```
> subst(-E,X,alpha^8);
alpha6
> subst(LP,X,alpha^8);
alpha6
> subst(-E,X,alpha^3);
alpha2
> subst(LP,X,alpha^3);
alpha2
```

Por tanto, el error es $e(X) = 1 + X^5 \sim \mathbf{e} = 10000100$ *y la palabra descodificada es*

$$c(X) - e(X) = 2X + X^3 + X^4 + 2X^5 + X^6 \sim 02011210$$

.

Nota 4.31. *En la sección 2.4.2 se ha definido el polinomio localizador de errores de forma que su término independiente sea* 1*. El polinomio localizador de errores obtenido en el ejemplo anterior es el polinomio definido en la sección 2.4.2 multiplicado por* α^3*.*

Este fenómeno se debe a que el elemento mínimo, o la solución del teorema 4.19, es único salvo multiplicación por escalar. Por tanto para obtener el verdadero polinomio localizador debemos dividirlo por su término independiente. Formalmente, $(E(X), L(X))$ *de la sección 2.4.2 es igual a*

$$(\alpha^3)^{-1}((X + \alpha^5, X^2 + \alpha X + \alpha^3)) = (\alpha^5 X + \alpha^2, \alpha^5 X^2 + \alpha^6 X + 1)$$

En el ejemplo anterior no hemos considerado esta división porque para hacer los cálculos en la práctica no es necesario: las raíces del polinomio $L(X)$ *son las mismas aunque lo multipliquemos por una constante. Además, la fórmula dada por la proposición 2.47 para obtener los valores del error proporciona el mismo valor para cualquier multiplicación por una constante de* $(E(X), L(X))$ *(puesto que se anula al hacer el cociente).*

El método presentado en este capítulo tiene complejidad similar al algoritmo de Berlekamp-Massey del capítulo 2. Además, el método de Berlekamp-Massey puede entenderse dentro de este esquema, para mayor referencia puede consultarse [Fit95].

Capítulo 5

El ideal asociado a un código

En este capítulo final describiremos algunas técnicas relacionadas con las bases de Gröbner basadas en el algoritmo de Möller y las técnicas FGLM [FGLM93]. Esta técnicas nos permiten describir las clases de equivalencia de un código y descodificar mediante un método de gradiente. Por relajación de la notación y de las demostraciones y ejemplos nos limitaremos a códigos binarios, aunque la mayor parte de los resultados de este capítulo se pueden extender al caso general (véase [BBM06, BBM07a, BBFM06, BBM07b])

5.1. La representación de Gröbner de un código

En esta sección estudiaremos una representación de las relaciones de equivalencia dadas por un código lineal binario $\mathcal{C}$ de dimensión k y longitud n. El acercamiento es similar al uso de técnicas de tipo FGLM y a su vez, el algoritmo principal es una instancia concreta del algoritmo de Möller [BBM07b], al igual que el algoritmo 3.11.

Consideremos el conjunto de monomios $\mathbb{T}^n$ generado por las n variables $X = \{x_1, \ldots, x_n\}$, y el morfismo de monoides de $\mathbb{T}^n$ sobre $\mathbb{F}_2^n$ dado por:

$$\begin{aligned} \psi : \mathbb{T}^n &\to \mathbb{F}_2^n \\ x_i &\mapsto (0, \ldots, 0, \underbrace{1}_{i}, 0, \ldots, 0). \end{aligned}$$

Podemos extender el morfismo de forma que

$$\prod_{i=1}^{n} x_i^{\beta_i} \mapsto (\beta_1 \mod 2, \ldots, \beta_n \mod 2) \tag{5.1}$$

Diremos que $\prod_{i=1}^{n} x_i^{\beta_i} \in \mathbb{T}^n$ está expresado en forma estándar si $\beta_i < 2$ para todo i, esto es, si está libre de cuadrados.

Como ya hemos visto, $\mathcal{C}$ define una relación de equivalencia $R_{\mathcal{C}}$ en $\mathbb{F}_2^n$ dada por

$$(\mathbf{u}, \mathbf{v}) \in R_{\mathcal{C}} \Leftrightarrow \mathbf{u} - \mathbf{v} \in \mathcal{C}. \tag{5.2}$$

Dada una matriz H de control del código, podemos traducir esta relación en términos de monomios $\mathbf{x^a}, \mathbf{x^b} \in \mathbb{T}^n$

$$\mathbf{x^a} \equiv_{\mathcal{C}} \mathbf{x^b} \Leftrightarrow (\psi(\mathbf{x^a}), \psi(\mathbf{x^b})) \in R_{\mathcal{C}} \Leftrightarrow \xi_{\mathcal{C}}(\mathbf{x^a}) = \xi_{\mathcal{C}}(\mathbf{x^b}) \tag{5.3}$$

donde $\xi_{\mathcal{C}}(\mathbf{x^a}) = H \cdot (\psi(\mathbf{x^a}))^t$ representa la transición del monoide $\mathbb{T}^n$ al conjunto de síndromes del código $\mathcal{C}$.

Llamaremos *soporte* de $\mathbf{x^a} \in \mathbb{T}^n$ al conjunto de variables en X que dividen a $\mathbf{x^a}$ y lo denotaremos por $\mathrm{sop}(\mathbf{x^a})$. Existe una clara analogía con la definición 2.14 del soporte de la palabra asociada a $\mathbf{x^a}$ en $\mathbb{F}_2^n$ si el término está expresado en forma estándar.

Definición 5.1 (Orden de Hamming)**.** *Dados dos monomios $\mathbf{x^a}, \mathbf{x^b} \in \mathbb{T}^n$ diremos que $\mathbf{x^a}$ es menor que $\mathbf{x^b}$ respecto del orden de Hamming y lo denotaremos por $\mathbf{x^a} \prec_H \mathbf{x^b}$, si alguna de las siguientes condiciones se cumple:*

1. $|\mathrm{sop}(\mathbf{x^a})| < |\mathrm{sop}(\mathbf{x^b})|$.
2. $|\mathrm{sop}(\mathbf{x^a})| = |\mathrm{sop}(\mathbf{x^b})|$ *y* $\mathbf{x^a} \prec_{ad} \mathbf{x^b}$*, donde $\prec_{ad}$ denota un orden admisible arbitrario en $\mathbb{T}^n$.*

Definición 5.2 (Representación de Gröbner)**.** *Sea $\mathcal{C}$ un código lineal sobre $\mathbb{F}_2$ con dimensión k y longitud n. Llamaremos representación de Gröbner del código $\mathcal{C}$ a un par (N, ϕ) dado por*

- *Un conjunto de términos $N = \{\tau_1, \ldots, \tau_{q^{n-k}}\} \subseteq \mathbb{T}^n$*
- *y una aplicación $\phi : N \times X \to N$*

tal que cumple las siguientes condiciones

1. $1 \in N$.
2. *Si* $\tau_1, \tau_2 \in N$ *y* $\tau_1 \neq \tau_2$ *entonces* $\xi_{\mathcal{C}}(\tau_1) \neq \xi_{\mathcal{C}}(\tau_2)$.
3. *Para todo* $\tau \in N \setminus \{1\}$ *existe un elemento* $x \in X$ *tal que* $\tau = \tau' x$ *y* $\tau' \in N$.
4. $\xi_{\mathcal{C}}(\phi(\tau, x_i)) = \xi_{\mathcal{C}}(\tau x_i)$.

El cardinal del conjunto N es el mismo que el de clases de equivalencia (síndromes) del código $\mathcal{C}$. La condición (2) nos muestra que dos elementos diferentes de N pertenecen a distinta clase de equivalencia. La aplicación ϕ nos muestra la estructura multiplicativa y es independiente de la elección de los elementos del conjunto N. Nótese que ϕ puede ser vista como una función sobre las clases de equivalencia, que envía cada clase $\mathbf{c} + \mathcal{C}$ en la clase $(\mathbf{c} + \mathbf{e}_i) + \mathcal{C}$, donde $\mathbf{e}_i = (0, \ldots, \underbrace{1}_{i}, \ldots, 0)$ corresponde al elemento x_i por el que se multiplica.

El siguiente algoritmo nos muestra una forma de calcular una instancia de N y la correspondiente representación de la aplicación ϕ, una vez hemos prefijado un orden en las variables en X.

Algoritmo 5.3 (FGLM Borges-Borges-Martínez)**.**

Input: *H matriz de control de $\mathcal{C}$.*

Output: *N, ϕ para $\mathcal{C}$ como en la definición 5.2.*

```
1:  List ← {1}, N ← ∅, r ← 0
2:  while List ≠ ∅ do
3:     τ ← NextTerm[List], v ← ξ_C(τ)
4:     j ← Member[v, {v_1, ..., v_r}]
5:     if j ≠ false then
6:        for k such that τ = τ'x_k with τ' ∈ N do
7:           φ(τ', x_k) = τ_j
8:        end for
9:     else
10:       r ← r + 1, v_r ← v, τ_r ← τ, N ← N ∪ {τ_r}
11:       List ← InsertNext[τ_r, List]
12:       for k such that τ_r = τ'x_k with τ' ∈ N do
```

13: $\phi(\tau', x_k) = \tau_r$
14: ***end for***
15: ***end if***
16: ***end while***

Donde las funciones internas del algoritmo se corresponden con:

1. InsertNext$[\tau, \text{List}]$ inserta todos los productos τx en la lista ordenada List, donde $x \in X$, y mantiene la lista ordenada con respecto al orden $\prec_H$.
2. NextTerm[List] devuelve el primer elemento de la lista List y le borra de la misma.
3. Member$[obj, G]$ devuelve la posición j de obj en G si $obj \in G$ y "false" en caso contrario.

Para encontrar una demostración del algoritmo ver [BBM07a].

Ejemplo 5.4. *Consideremos el código $\mathcal{C}$ en $\mathbb{F}_2^6$ con matriz generadora*

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

Las palabras del código son

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

Consideremos $\prec_{ad}$ el orden graduado reverso-lexicográfico compatible con $x_1 \prec x_2 \prec \cdots \prec x_6$ y su correspondiente extensión a $\prec_H$. El algoritmo 5.3 calcula el siguiente conjunto de representantes de las clases de equivalencia

$$N = \{1, x_1, x_2, x_3, x_4, x_5, x_6, x_1x_6\}$$

y la aplicación ϕ se representa como una matriz de posiciones como sigue:

$$\begin{bmatrix} [[0,0,0,0,0,0],\ 1,\ [2,3,4,5,6,7]], & [[1,0,0,0,0,0],\ 1,\ [1,6,5,4,3,8]], \\ [[0,1,0,0,0,0],\ 1,\ [6,1,8,7,2,5]], & [[0,0,1,0,0,0],\ 1,\ [5,8,1,2,7,6]], \\ [[0,0,0,1,0,0],\ 1,\ [4,7,2,1,8,3]], & [[0,0,0,0,1,0],\ 1,\ [3,2,7,8,1,4]], \\ [[0,0,0,0,0,1],\ 1,\ [8,5,6,3,4,1]], & [[1,0,0,0,0,1],\ 0,\ [7,4,3,6,5,2]] \end{bmatrix}$$

En cada tripleta de la lista anterior la primera entrada corresponde a los elementos $\psi(\tau)$ *con* $\tau \in N$ *(*$\tau = N[i]$*). La segunda entrada es* 1 *si* $\psi(\tau) \in B(\mathcal{C}, t)$ *(esto es,* τ *corresponde a un síndrome corregible) o* 0 *en otro caso. La tercera componente contiene los valores en la lista ordenada* N *que corresponden a* $\phi(\tau, x_j)$*, para* $j = 1, \ldots, 6$*.*
Por ejemplo, si nos fijamos en la última tripleta de la matriz

$$[[1,\ 0,\ 0,\ 0,\ 0,\ 1],\ 0,\ [7, 4, 3, 6, 5, 2]]$$

La lista $[1,\ 0,\ 0,\ 0,\ 0,\ 1]$ *representa el elemento de* N *dado por* x_1x_6*. La segunda entrada es* 0*, esto es,* x_1x_6 *corresponde a un síndrome no corregible. Por último la entrada* $[7, 4, 3, 6, 5, 2]$ *representa los siguientes resultados*

$$\begin{aligned}
\phi(x_1x_6, x_1) &= x_1x_6\\
\phi(x_1x_6, x_2) &= x_3\\
\phi(x_1x_6, x_3) &= x_2\\
\phi(x_1x_6, x_4) &= x_5\\
\phi(x_1x_6, x_5) &= x_4\\
\phi(x_1x_6, x_6) &= x_1.
\end{aligned}$$

Nota 5.5. *El algoritmo 5.3 que genera la salida mostrada en el ejemplo anterior, así como otras utilidades, se encuentra en el paquete de GAP[GAP06] "Gröbner basis by linear algebra and codes" [GBLA06].*

Nota 5.6. *Compruébese que dada la construcción y estructura del algoritmo 5.3, los representantes de las clases de equivalencia en* N *tales que corresponden a vectores en* $B(\mathcal{C}, t)$ *son los términos en* $\mathbb{T}^n$ *mínimos con respecto al orden* $\prec_H$*, por lo tanto son los términos estándar cuyas imágenes por la aplicación* ψ *corresponden a los vectores síndromes.*

Un producto importante del algoritmo es el siguiente teorema que nos permite calcular la capacidad correctora del código $\mathcal{C}$ a partir del primer término que se incorpora a la lista (véase su demostración en [BBM07a]).

Teorema 5.7. *Sean* List *la lista de términos en el paso 3 del algoritmo 5.3 y* τ *el primer elemento que se analiza en* NextTerm[List] *tal que* τ *no pertenece al conjunto* N *y* τ *se encuentra en representación estándar. Entonces*

$$t = |\mathrm{sop}(\tau)| - 1, \tag{5.4}$$

donde t es la capacidad correctora del código.

Obsérvese que no es necesario ejecutar completamente el algoritmo para calcular el elemento τ correspondiente al teorema anterior pues el algoritmo es incremental y sólamente necesitamos el primer elemento que lo cumpla.

Nota 5.8. *La construcción en esta sección puede ser generalizada de forma no trivial al caso de cuerpos finitos con característica distinta de 2, véase [BBM07a].*

Como es usual en la teoría de bases de Gröbner, definiremos una reducción en las formas canónicas N de la forma siguiente

Definición 5.9 (Reducción en un paso)**.** *Consideremos el par N, ϕ dados en la definición 5.2 representación de Gröbner del código $\mathcal{C}$ y $\tau \in N$, $x \in X$, diremos que $\phi(\tau, x)$ es la forma canónica de τx, esto es τx reduce en un paso a $\phi(\tau, x)$.*

La reducción anterior se puede extender a $\mathbb{T}^n$ como sigue: Sea $\mathbf{x^a} = x_{i_1} \dots x_{i_k} \in \mathbb{T}^n$, $x_{i_j} \prec_H x_{i_k}$ para todo $j \leq k-1$ y consideremos la función recursiva

$$\begin{array}{rccc} \mathrm{Can}_{\prec_H}: & \mathbb{T}^n & \longrightarrow & N \\ & 1 & \mapsto & 1 \\ & \mathbf{x^a} & \mapsto & \phi(\mathrm{Can}_{\prec_H}(x_{i_1} \dots x_{i_{k-1}}), x_{i_k}). \end{array} \tag{5.5}$$

donde el caso inicial es la palabra nula $\mathbf{0}$ representada por 1. El elemento $\mathrm{Can}_{\prec_H}(\mathbf{x^a}) \in N$ tiene el mismo síndrome que $\mathbf{x^a}$ pues

$$\begin{aligned} \xi_{\mathcal{C}}\left(\mathrm{Can}_{\prec_h}(x_{i_1} \dots x_{i_k})\right) &= \xi_{\mathcal{C}}\left(\phi\left(\mathrm{Can}_{\prec_H}(x_{i_1} x_{i_2} \dots x_{i_{k-1}}), x_{i_k}\right)\right) \\ &= \xi_{\mathcal{C}}\left(\mathrm{Can}_{\prec_H}(x_{i_1} x_{i_2} \dots x_{i_{k-1}}) x_{i_k}\right) \\ &= \xi_{\mathcal{C}}\left(\mathrm{Can}_{\prec_H}(x_{i_1} x_{i_2} \dots x_{i_{k-1}})\right) + \xi_{\mathcal{C}}(x_{i_k}) \end{aligned} \tag{5.6}$$

La segunda igualdad en la ecuación (5.6) se cumple por la definición de ϕ y la tercera igualdad se debe a la aditividad de $\xi_{\mathcal{C}}$. Podemos calcular por recursión

$$\xi_{\mathcal{C}}\left(\mathrm{Can}_{\prec_H}(x_{i_1} \dots x_{i_k})\right) = \xi_{\mathcal{C}}\left(\mathrm{Can}_{\prec_H}(1) x_{i_1} x_{i_2} \dots x_{i_k}\right) = \xi_{\mathcal{C}}\left(x_{i_1} x_{i_2} \dots x_{i_k}\right)$$

por lo tanto ambos síndromes son iguales. Finalmente, que el elemento $\mathrm{Can}_{\prec_H}$ está bien definido para los elementos de $\mathbb{T}^n$ por la recurrencia en (5.5) se sigue de los pasos 10 y 11 en el algoritmo 5.3.

5.2. El ideal binomial asociado a un código

Consideremos el ideal binomial

$$\mathrm{I}(\mathcal{C}) := \langle \{\tau_1 - \tau_2 \mid (\psi(\tau_1), \psi(\tau_2)) \in R_{\mathcal{C}}\} \rangle \subset \mathbb{F}[X] \tag{5.7}$$

donde $\mathbb{F}$ es un cuerpo arbitrario. Nótese que como el código $\mathcal{C}$ es binario se tiene que $x_i^2 - 1 \in \mathrm{I}(\mathcal{C})$ para todo $x_i \in X$.
Sea $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ el conjunto formado por los vectores fila correspondientes a la matriz generadora del código (o con más generalidad, una matriz cuyas filas generen el código $\mathcal{C}$) y sea

$$I = \langle \{\mathbf{x}^{\mathbf{w}_1} - 1, \ldots, \mathbf{x}^{\mathbf{w}_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\} \rangle \tag{5.8}$$

el ideal generado por los conjuntos de binomios $\{\mathbf{x}^{\mathbf{w}_1} - 1, \ldots, \mathbf{x}^{\mathbf{w}_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\} \in \mathbb{F}[X]$. Como los vectores $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ generan $\mathcal{C}$ se sigue que $I = I(\mathcal{C})$.

Consideremos $G_\prec$ la base de Gröbner reducida del ideal $I(\mathcal{C})$ con respecto a un orden $\prec$ lexicográfico compatible con el orden dado por grado. El conjunto de polinomios $G_\prec$ puede ser calculado mediante el algoritmo de Buchberger partiendo del conjunto inicial de polinomios $\{\mathbf{x}^{\mathbf{w}_1} - 1, \ldots, \mathbf{x}^{\mathbf{w}_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\}$. En este caso hay algunas ventajas computacionales:

1. El cuerpo en que se encuentran los coeficientes puede ser $\mathbb{F}_2$ (por lo tanto no hay crecimiento de los coeficientes).

2. La longitud (número de variables) máxima de un binomio en el ideal es n pues los binomios $x_i^2 - 1$ previene el crecimiento de los exponentes.

Los dos principales inconvenientes del algoritmo de Buchberger para el cálculo de bases de Gröbner desaparecen en este caso. Para encontrar una mayor información sobre la complejidad y un algoritmo de Buchberger modificado adecuado a este caso véase [BBFM06].

Consideremos ahora el anillo cociente

$$\mathbb{F}[X]/\mathrm{I}(\mathcal{C}), \tag{5.9}$$

se puede demostrar (véase [BBM07b]) que los elementos en N calculados en el algoritmo 5.3 son representantes de las clases de equivalencia en la

ecuación (5.9) y la aplicación ϕ corresponde a la expresión de la tabla de multiplicación en el álgebra (5.9). Por lo tanto hay una traducción entre las clase de equivalencia del código y las del anillo cociente en (5.9).

Si $\prec = \prec_H$, teniendo en cuenta el teorema 5.7, si el código es al menos 1-corrector los polinomios $x_i^2 - 1$ se encuentran en la base de Gröbner reducida del ideal $\mathrm{I}(\mathcal{C})$. Por lo tanto todas las variables x_i se corresponden con elementos del conjunto N. Sin embargo, si el código es 0 corrector existe al menos una x_i tal que no se encuentra en N.

Ejemplo 5.10. *Tomando el mismo código $\mathcal{C}$ que en el ejemplo 5.4 la base de Gröbner reducida del ideal* $\mathrm{I}(\mathcal{C})$ *es*

$$\begin{aligned}\{&x_1^2-1, x_2^2-1, x_3^2-1, x_4^2-1, x_5^2-1, x_6^2-1,\\ &x_1x_2-x_5, x_1x_3-x_4, x_1x_4-x_3, x_1x_5-x_2,\\ &x_2x_3-x_1x_6, x_2x_4-x_6, x_2x_5-x_1, x_2x_6-x_4,\\ &x_3x_4-x_1, x_3x_5-x_6, x_3x_6-x_5,\\ &x_4x_5-x_1x_6, x_4x_6-x_2, x_5x_6-x_3\}.\end{aligned}$$

5.3. Aplicaciones

Existen múltiples aplicaciones de las dos seciones anteriores. En el artículo [BBM06] se puede encontrar una aplicación al problema de determinar si dos códigos son equivalentes. En [BBFM06] se encuentran varias aplicaciones a distintos problemas de ciclos sobre un grafo. En esta sección mostraremos brevemente cómo podemos utilizar la información en una base de Gröbner de $\mathrm{I}(\mathcal{C})$ para descodificar el código $\mathcal{C}$.

La descodificación completa para un código lineal [MS85] es un problema computacional NP-completo (véase [BMT78]), por consiguiente es bastante improbable que exista un algoritmo que en tiempo (espacio) polinomial lo resuelva. En la literatura se encuentran varios intentos para mejorar la idea del descodificación mediante el síndrome expuesta en la seción 2.2.4. Estos intentos buscan una estructura menor que la tabla de síndromes para efectuar la descodificación, la idea es encontrar en cada relación de equivalencia el menor peso de las palabras en dicho conjunto, en lugar de almacenar el vector error candidato. Entre estos métodos se pueden destacar por ejemplo el "Step-by-Step algorithm" en [PW72], el "test set decoding" en [Bar98] los algoritmos basados en

los "zero-neighbors" y los "zero-guards" en [Han98, HH97, LH85]. Siguendo la notación en [Bar98] llamaremos a estod métodos *algoritmos de descodificación por gradiente*.

De el mismo modo que las estructuras citadas en el párrafo anterior, la aplicación ϕ y el conjunto N, calculados en la sección 5.1 para el caso binario y en [BBM07a] para el caso general, pueden ser utilizadas para descodificar por gradiente un código lineal. En el caso binario la base reducida de Gröbner calculada en la sección anterior también puede ser utilizada para descodificar, y además suele ser bastante más "pequeña" que la expresión de la ϕ.

El siguiente teorema es independiente del tipo de reducción a utilizar (mediante la aplicación ϕ o mediante la base reducida en el caso de un código binario).

Teorema 5.11 ([BBM07a])**.** *Sea $\mathcal{C}$ un código lineal de longitud n y capacidad correctora t. Sea $\tau \in \mathbb{T}^n$ y $\tau' \in N$ su forma canónica correspondiente.*

- *Si $\mathrm{w}(\psi(\tau')) \leq t$ entonces $\psi(\tau')$ es el vector error correspondiente a $\psi(\tau)$.*
- *En otro caso, si $\mathrm{w}(\psi(\tau')) > t$, $\psi(\tau)$ contiene más de t errores.*

Ejemplo 5.12. *Consideremos $\mathcal{C}$ como en el ejemplo 5.4 y el ideal $\mathrm{I}(\mathcal{C})$ y su base de Gröbner reducida en el ejemplo 5.10. Supongamos que hemos recibido la palabra $\mathbf{y} \in \mathbb{F}_2^n$*

Descodificación utilizando ϕ.

1. *Caso 1 $\mathbf{y} \in B(\mathcal{C}, t)$:*
   $\mathbf{y} = (1,1,0,1,1,0)$*;* $w_y := x_1x_2x_4x_5$*;*

$$\phi(1, x_1) = x_1,$$
$$\phi(x_1, x_2) = x_5,$$
$$\phi(x_5, x_4) = x_2x_3,$$
$$\phi(x_2x_3, x_5) = x_4,$$

   Esto es $\mathrm{w}(\psi(x_4)) = 1$, por lo tanto la palabra correspondiente a $\mathbf{y}$ es $\mathbf{c} = \mathbf{y} - \psi(x_4) = (1,1,0,0,1,0)$.

2. *Caso 2* $\mathbf{y} \notin B(\mathcal{C}, t)$*:*
   $y = (0, 1, 0, 0, 1, 1)$*;* $w_y := x_2x_5x_6$*;*

$$\phi(1, x_2) = x_2,$$

$$\phi(x_2, x_5) = x_1,$$

$$\phi(x_1, x_6) = x_2x_3.$$

   Se tiene que $\mathrm{w}(\psi(x_2x_3)) > 1$ *y se comunica que se ha cometido un error. El lector puede comprobar que la palabra* $\mathbf{y}$ *no se encuentra en el conjunto* $B(\mathcal{C}, 1)$ *observando las palabras del código* $\mathcal{C}$ *dadas en el ejemplo 5.4.*

Descodificación usando la base de Gröbner reducida

Consideremos los dos casos anteriores. Por $w \xrightarrow{g} v$ *representaremos la reducción de* w *a* v *modulo el binomio* g *de la base reducida.*

1. $x_1x_2x_4x_5 \xrightarrow{x_1x_2 - x_5} x_4x_5^2$, $x_4x_5^2 \xrightarrow{x_5^2 - 1} x_4$.
2. $x_2x_5x_6 \xrightarrow{x_2x_5 - x_1} x_1x_6$.

Bibliografía

[AL96] W.W. Adams, P. Loustaunau. An introduction to Gröbner bases. *AMS, Graduate Studies in Mathematics*, **3**,(1996).

[Bar98] A. Barg. Complexity issues in coding theory. In *Handbook of Coding Theory, Elsevier Science*, vol. 1, (1998).

[BS87] D. Bayer, M. Stillman. A theorem on refining division orders by the reverse lexicographic order. *Duke J. Math.*,**55**, 321–328 (1987).

[BW93] T. Becker, V Weispfenning. Gröbner bases. *Springer Verlag*, (1993).

[BMT78] E. Berlekamp, R. McEliece, H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory*, IT-24, 384–386, (1978).

[BBFM06] M. Borges-Quintana, M. Borges-Trenard, P. Fitzpatrick, and E. Martínez-Moro. Gröbner bases and combinatorics for binary codes. *Submitted to Appl. Algebra Engrg. Comm. Comput.*, (2006).

[BBM06] M. Borges-Quintana, M. Borges-Trenard, and E. Martínez-Moro. A general framework for applying FGLM techniques to linear codes. *AAECC 16, Springer Lecture Notes in Computer Science*, **3857**, 76–86, (2006).

[GBLA06] M. Borges-Quintana, M. A. Borges-Trenard and E. Martínez-Moro. GBLA-LC: Gröbner basis by linear algebra and codes. International Congress of Mathematicians 2006 (Madrid), Mathematical Software, *EMS (Ed)*, 604–605, (2006). Avaliable at `http://www.math.arq.uva.es/ edgar/GBLAweb/`.

[BBM07a] M. Borges-Quintana, M. Borges-Trenard, and E. Martínez-Moro. On a Gröbner bases structure associated to linear codes. *Journal of Discrete Mathematical Sciences and Cryptography*, April 2007.

[BBM07b] M. Borges-Quintana, M. Borges-Trenard, and E. Martínez-Moro. A Gröbner Representation for Linear Codes. *Advances in Coding Theory and Cryptography*, Series on Coding Theory and Cryptology vol. 3, World Scientific, (2007).

[Buc70] B. Buchberger. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. *Aequationes Math.*, 4:374–383, (1970).

[Buc85] B. Buchberger. Gröbner bases: an algorithmic method in polynomial ideal theory. *Multidimensional system theory, N.K. Bose (ed.) D. Reidel Pub. Comp., Dordrecht*, 184–232 (1985).

[Chi95] D. Chillag Regular Representation of Semisimple Algebras, Separable Field Extensions, Group Characters, Generalized Circulants and Generalized Cyclic Codes, Linear Algebra and its Applications, **218**, 147–183 (1995).

[CRHT94a] X. Chen, I.S. Reed, T. Helleseth, K. Truong. Use of Gröbner bases to decode binary cyclic codes up to the true minimum distance. *IEEE Trans. Inf. Theory*, **40**,1654–1661 (1994).

[CRHT94b] X. Chen, I.S. Reed, T. Helleseth, K. Truong. Algebraic decoding of cyclic codes: a polynomial ideal point of view. *Contemp. Math.*, **168**,15–22, (1994).

[Co90] A.B. Cooper III. Direct solution of BCH decoding equations. *E. Arikan (Ed.) Communication, Control and Singal Processing*, 281–286 (1990).

[Co91] A.B. Cooper III. Finding BCH error locator polynomials in one step. *Electronic Letters*, **27**, 2090–2091 (1991).

[CLO97] D. Cox, J. Little,D. O'Shea. Ideals,verieties and algorithms. An introducction to computational algebraic geometry and commutative algebra. *2nd. edition. Springer*,(1997).

[CLO05] D. Cox, J. Little,D. O'Shea. Using algebraic geometry. *2nd. edition. Springer, Graduate text in mathematics*, **185**,(2005).

[DGM70] P. Delsarte, J.M. Goethals, F.J. MacWilliams. On generalized Reed-Muller codes and their relatives. *Information and control*, **16**, 403–442, (1970).

[Dub89] T. Dubé. Quantitative analysis problems in computer algebra: Gröbner basis and the Nullstellensatz. *PhD. Thesis. Courant Institute, New York University*,(1989).

[FGLM93] J.C. Faugère, P. Gianni, D. Lazard, T. Mora. Efficient Computation of Zero-Dimensional Grobner Bases by Change of Ordering. *J. Symbolic Comput.*, **16(4)**, 329–344, (1993).

[FL98] J. Fitzgerald, R.F. Lax. Decoding affine variety codes using Gröbner bases *Des. Codes Cryptogr.*, **13**, 147–158, (1998).

[Fit95] P. Fitzpatrick. On the key equation. *IEEE Trans. Inform. Theory*, **41-5**, 1290–1302, (1995).

[Fit97] P. Fitzpatrick. Solving a multivariable congruence by change of term order. *J. Symbolic Comput.*, **24**, 575–589, (1997).

[GAP06] The GAP Group. GAP – Groups, Algorithms, and Programming, Version 4.4.9, (2006). `http://www.gap-system.org`.

[Gia89] P. Giani. Properties of Gröbner basis under specializations. *EUROCAL'87. Lect. Notes in Computer Sci.*, **378**, 293–297, (1989).

[GP02] G.-M. Greuel, G. Pfister . *A Singular introduction to commutative algebra*. Springer-Verlag, (2002).

[GPS05] G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR 3.0. A Computer Algebra System for Polynomial Computations. Centre for Computer Algebra, University of Kaiserslautern (2005). `http://www.singular.uni-kl.de`.

[Han98] Y. Han. A New Decoding Algorithm for Complete Decoding of Linear Block Codes. *SIAM J. Discrete Math.*, vol. 11(4), 664–671, (1998).

[HH97] Y. Han, C. Hartmann. The zero-guards algorithm for general minimum-distance decoding problems. *IEEE Trans. Inform. Theory*, vol. 43, 1655–1658, (1997).

[JH04] J. Justensen, T. Høholdt. *A course in error-correcting codes*. European Mathematical Society (EMS),(2004).

[LH85] L. Levitin, C. Hartmann. A new approach to the general minimum distance decoding problem: the zero-neighbors algorithm. *IEEE Trans. Inform. Theory*, vol. 31, 378–384, (1985).

[LN86] H. Lidl, H. Niederreiter. Introduction to Finite Fields and their Applications. Cambridge University Press, (1986).

[LX04] S. Ling, C. Xing. *Coding theory, a first course*. Cambridge University Press, Cambridge, (2004).

[Lit98] Little, J.B. Applications to coding theory. In Applications of computational algebraic geometry. Proceedings of Symposia in Applied Mathematics vol. 53. American Mathematical Society, (1998).

[LY97] P. Loustaunau, E.V. York. On the decoding of cyclic codes using Gröbner bases *AAECC*,**8**, 469–483, (1997).

[MS85] F.J. MacWilliams, N.J.A. Sloane. The theory of error-correcting codes. Parts I, II. (3rd repr.). North-Holland Mathematical Library, *North- Holland (Elsevier)*, **16**, (1985).

[Mar04] E. Martínez-Moro *Regular Representations of Finite-dimensional Separable Semisimple Algebras and Gröbner Bases*, Journal of Symbolic Computation **37**, 575–587, (2004)

[Mar07] E. Martínez-Moro *On semisimple algebra codes: generator theory*, Algebra and Discrete Mathematics, To appear. 14 pag (2007).

[MB82] H.M. Möller, B. Buchberger. The construction of multivariate polynomials with preasigned zeros. *Computer Algebra EUROCAM'82, Springer Verlag. Lecture Notes in Comput. Sci.* ,24–31 (1982).

[MM84] H.M. Möller, T. Mora. Upper and lower bounds for the degree of Groebner basis *EUROSAM 1984, J. Fich (ed),Springer Verlag. Lecture Notes in Comput. Sci.* , **174**, 172–183 (1984).

[Mor05] T. Mora. Solving polynomial equation systems II. Macaulay's paradigm and Gröbner technology *Cambridge University Press, Encyclopedia of Mathematics and its Applications*, **99**,(2005).

[MT97] C. Munuera, J. Tena. Codificación de la información. *Pub. Universidad de Valladolid,* ,(1997).

[OS05] E. Orsini and M. Sala. Correcting errors and erasures via the syndrome variety. *J. Pure Appl. Algebra*, 200(1-2), 191–226, (2005).

[PW72] W. W. Peterson, E. J. Jr. Weldon. *Error-Correcting Codes (2nd ed.)*. MIT Press, (1972).

[PH98] V. Pless, W.C. Huffman (editores). Handbook of Coding Theory. Elsevier, (1998).

[Sal02] M. Sala. Groebner bases and distance of cyclic codes. *Appl. Algebra Engrg. Comm. Comput.*, 13(2), 137–162, (2002).

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 623–656, (1948).

[Win96] F. Winkler. Polinomial algorithms in computer algebra *Springer Verlag Wien*,(1996).

[Yor94] E.V. York. Algebraic Description and Construction of Error Correcting Codes, a Systems Theory Point of View. *PhD Thesis*, University of Notre Dame, (1997).

Índice alfabético

La Asociación Matemática Venezolana fue fundada en 1990 como una organización civil sin fines de lucro, cuya finalidad es trabajar por el desarrollo de la matemática en Venezuela.

Asociación Matemática Venezolana
Apartado 47.898, Caracas 1041-A, Venezuela
http://amv.ivic.ve/