Irene Márquez Corbella

Presentaciones binomiales de problemas lineales modulares: Aplicación a los códigos correctores de errores.

Trabajo de investigación dirigido por:
Dr. D. Edgar Martínez Moro

UVa

Diploma de Estudios Avanzados 2010

Dr. D. Edgar Martínez Moro,
Profesor Titular del Departamento de Matemática Aplicada de la Universidad de Valladolid.

CERTIFICA:

Que el presente trabajo, *Presentaciones binomiales de programas lineales modulares: Aplicaciones a los códigos correctores de errores*, ha sido realizado bajo su dirección por Irene Márquez Corbella en el Departamento de Álgebra, Geometría y Topología y constituye su memoria de Investigación para la obtención del Diploma de Estudios Avanzados del programa de Doctorado en Matemáticas de la Universidad de Valladolid.

En Valladolid, a 1 de Julio de 2010.

Fdo: Dr. D. Edgar Martínez Moro.

Índice general

Introducción

El presente trabajo tiene por objetivo presentar una memoria con los resultados obtenidos y material estudiado durante el periodo de investigación del Programa de Doctorado en Matemáticas de la Universidad de Valladolid, curso $2009 - 2010$, bajo la tutela de los profesores Dr. D. Antonio Campillo López y Dr. D. Edgar Martínez Moro, requisito indispensable para la obtención del Diploma de Estudios Avanzados.
La investigación que se presenta ha dado lugar a las siguientes publicaciones:

- M. Borges-Quintana, M. A. Borges-Trenard, I. Márquez-Corbella and E. Martínez-Moro. *An Algebraic View to Gradient Descent Decoding*. Acepted to IEEE Information Theory Workshop, 2010. [12].

- M. Borges-Quintana, M. A. Borges-Trenard, I. Márquez-Corbella and E. Martínez-Moro.*Descodificación por gradiente como reducción*. XII Encuentro de Álgebra Computacional y Aplicaciones (Santiago de Compostela, 19-21 de julio de 2010). [13].

- I. Márquez-Corbella and E. Martínez-Moro. *Combinatorics of minimal codewords of some linear codes*. Submitted to Advances in Mathematics of Communications, 2010. [46].

- I. Márquez-Corbella and E. Martínez-Moro. *Programación lineal modular y bases de Graver: Cálculo de soportes minimales de códigos lineales*. VII Jornadas de Matemática Discreta y Algorítmica (Castro Urdiales, 7-9 de julio de 2010), 451-458, 2010. [47].

Es bien conocido que la descodificación de códigos lineales binarios se puede ver como un problema de programación lineal entera modular. Esta relación

motivó el inicio del trabajo que tenía como objetivo describir algebraicamente el conjunto de palabras de soporte mínimo de un código definido sobre $\mathbb{Z}_q$ o, equivalentemente, circuitos mínimos en matroide $\mathbb{Z}_q - representable$. El interés del conjunto de palabras de soporte mínimo de un código viene dado por su relación con los algoritmos de descodificación por gradientes. Existen dos algoritmos de descodificación por gradientes propuestos para códigos binarios de forma independiente por Liebler [42] y por Ashikhmin y Barg [6]. Liebler menciona en su artículo [42] que los dos algoritmos, a pesar de compartir la filosofía de descodificación por gradiente, son diferentes. En el capítulo 4 de este trabajo se probará que estos dos algoritmos son duales en el sentido de que son dos formas de entender la representación de Gröbner de un código.

El conjunto de palabras de soporte mínimo también tiene interés en el campo de la criptografía, en particular en el esquema para compartir secretos basados en códigos correctores de errores, ya que describen el conjunto de coaliciones minimales que acceden al mismo, véanse los textos [1, 6].

El cálculo de las palabras de soporte mínimo de un código lineal arbitrario es NP-completo, ya que está relacionado con la descodificación completa, incluso si se permite preprocesamiento de los datos (Véanse [7, 10, 20]).

Durante este trabajo se utilizaran las siguientes aplicaciones de *cambio de característica*:

$$\blacktriangledown : \mathbb{Z}^s \longrightarrow \mathbb{Z}_q^s \quad \text{y} \quad \blacktriangle : \mathbb{Z}_q^s \longrightarrow \mathbb{Z}^s$$

donde s se determina del contexto y ambas aplicaciones actúan coordenada a coordenada sobre vectores y matrices. La aplicación $\blacktriangledown$ se corresponde con la reducción módulo q y $\blacktriangle$ sustituye la clase de los elementos $0, 1, \ldots, q-1$ por el mismo símbolo considerado como un entero.

Teniendo en cuenta estas aplicaciones y si se considera un entero $q \in \mathbb{Z}_{\geq 2}$, una matriz de coeficientes $A \in \mathbb{Z}_q^{m\times n}$ y los vectores $\mathbf{b} \in \mathbb{Z}_q^n$ y $\mathbf{w} \in \mathbb{R}^n$, definimos un problema de programación lineal entera modular, y se denotará por $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$, como un problema de programación lineal en aritmética modular cuyo objetivo es encontrar un vector $\mathbf{u} \in \mathbb{Z}_q^n$ que minimice la función de costos $\mathbf{w} \cdot \blacktriangle\mathbf{u}$ y verifique que $A\mathbf{u}^t \equiv \mathbf{b} \mod q$. Es decir:

$$\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b}) = \begin{cases} \text{Minimizar:} & \mathbf{w} \cdot \blacktriangle\mathbf{u} \\ \text{Sujeto a:} & \begin{cases} A\mathbf{u}^t \equiv \mathbf{b} \mod q \\ \mathbf{u} \in \mathbb{Z}_q^n \end{cases} \end{cases}$$

Conti y Traverso [22] propusieron en 1991 un algoritmo eficiente que utiliza bases de Gröbner para resolver problemas de programación lineal entera. En términos generales este algoritmo se basa en el cálculo de una base de

Gröbner del ideal asociado al núcleo de la matriz de coeficientes del problema, de forma que el exponente de la forma normal del monomio asociado a cualquier solución no óptima del problema coincide con la solución óptima buscada.
Luego, en 2002 Ikegami y Kaji en [37] adaptaron las ideas de Conti y Traverso para resolver un problema de programación lineal modular.
Sea $\mathbb{K}$ un cuerpo arbitrario, se denota por $\mathbb{K}[\mathbf{x}] := \mathbb{K}[x_1, \ldots, x_n]$ y por $\mathbb{K}[\mathbf{y}] := \mathbb{K}[y_1, \ldots, y_m]$ al anillo, en n y m indeterminadas respectivamente, sobre el cuerpo $\mathbb{K}$. Consideramos el siguiente homomorfismo de anillos Θ definido por:

$$\begin{array}{rccc} \Theta: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{K}[\mathbf{y}] \\ & \mathbf{x^u} & \longmapsto & \Theta(\mathbf{x^u}) = \mathbf{y}^{\blacktriangle A\mathbf{u}^t} \end{array}$$

En el [37] se muestra que el ideal generado por los vectores pertenecientes al núcleo de la matriz A, que se corresponde con el núcleo de Θ, viene dado por el ideal de eliminación $I = I_A \cap \mathbb{K}[\mathbf{x}]$ donde:

$$I_A = \langle \{\Theta(x_i) - x_i\}_{i=1}^n, \{y_j^q - 1\}_{j=1}^m \rangle \subseteq \mathbb{K}[\mathbf{x}, \mathbf{y}]$$

El algoritmo extendido de Conti y Traverso presentado por Ikegami y Kaji muestra que el proceso de reducción, para un orden monomial adecuado, dado por una base de Gröbner reducida $\mathcal{G}$ del ideal $I_A \cap \mathbb{K}[\mathbf{x}]$, permite resolver el problema $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$. Es decir, el exponente de la forma normal respecto de la base de Gröbner calculada del monomio asociado al vector $\mathbf{b}$, $\mathrm{nf}_{\mathcal{G}}(\mathbf{x^b})$, se corresponde con la solución óptima del problema.
La principal desventaja del algoritmo de Ikegami y Kaji es que involucra $m \times n$ variables en el cálculo de la base de Gröbner y además no muestra ningún sistema específico para acelerar el algoritmo clásico de Buchberger, cuya complejidad crece de forma exponencial en el número de variables. Aunque cabe destacar que uno de los problemas clásicos de la eficiencia del cálculo de bases de Gröbner, el crecimiento de los coeficientes, no debe ser considerado, pues sin falta de generalidad podemos suponer que $\mathbb{K} = \mathbb{F}_2$, ya que la información necesaria se encuentra codificada en los exponentes de los binomios.
La soluciones que se proponen, en el capítulo 3 del presente trabajo, para mejorar la eficacia del algoritmo de Ikegami y Kaji, es utilizar la filosofía expuesta por Di Biase-Urbanke en [26] para reducir el número de variables definiendo un ideal directamente en el anillo $\mathbb{K}[\mathbf{x}]$. La reducción del número de variables que propuesta es una generalización al caso modular no binario del artículo [15].

Sea $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\} \subseteq \mathbb{Z}_q^n$ un subconjunto de $\mathbb{Z}_q$- generadores del espacio vectorial generado por las filas de la matriz A definimos el siguiente ideal:

$$\blacktriangle I = \langle \{\mathbf{x}^{\blacktriangle \mathbf{w}_1} - 1, \ldots, \mathbf{x}^{\blacktriangle \mathbf{w}_k} - 1\} \cup \{x_i^q - 1\}_{i=1}^n \rangle \subseteq \mathbb{K}[\mathbf{x}].$$

En el mencionado capítulo 3 se probará que este ideal es equivalente al ideal de eliminación $I_A \cap \mathbb{K}[\mathbf{x}]$ definido por Ikegami y Kaji y que además es igual al ideal $I(A^\perp)$ definido como:

$$I(A^\perp) = \langle \{\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}} \mid A^\perp \blacktriangledown (\mathbf{a} - \mathbf{b})^t \equiv 0 \mod q\} \rangle$$

donde $A^\perp$ representa la matriz cuyas filas generan el siguiente subespacio vectorial: $\{\mathbf{u} \in \mathbb{Z}_q^n \mid \mathbf{u} \cdot a \equiv 0 \mod q, \ \forall a \text{ fila de la matriz } A\}$. La matriz $A^\perp$ coincide con la matriz no negativa que buscan Di Biase y Urbanke en [26]. Además, se recomienda utilizar la extensión al caso q-ario del algoritmo FGLM adaptado presentado en [15] para el cálculo de una base de Gröbner reducida.

Ikegami y Kaji en [37] muestran, en el caso binario $q = 2$, la vinculación entre el problema $\mathrm{IP}_{A,\mathbf{w},2}(\mathbf{b})$ con la descodificación completa del código generado por la matriz $A^\perp$ tomando como vector peso $\mathbf{w} = 1$. Este problema también es abordado en [15], donde se demuestra que las palabras representadas en la base de Gröbner reducida del ideal asociado a un código son minimales. Como ya se ha indicado estos dos métodos son equivalentes.

Desafortunadamente, para el caso $q > 2$ la distancia de Hamming no se ha conseguido expresar como la función objetivo de un problema lineal. En [14] se presentan algunas modificaciones para solucionar este problema en el caso $q = p^r$ siendo p un número primo, considerando el borde de un código y un orden especial en los monomios. El resto de supuestos es un problema que queda sin resolver con este trabajo pero que constituirá una de las líneas de investigación para el futuro proyecto de tesis.

Es natural asociar al ideal construido a partir de los problemas de programación lineal entera modular $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ la base de Graver formada por el conjunto de binomios primitivos de dicho ideal. En el capítulo 4 se muestra que la base de Graver del ideal asociado a un código $\mathcal{C} \subseteq \mathbb{Z}_q^n$ se corresponde con el conjunto de palabras de soporte mínimo de dicho código. Además se aporta un algoritmo para el cálculo de dichas palabras, así como distintos ejemplos que clarifican lo expuesto.

En resumen, este trabajo está dividido en 4 capítulos. En los dos primeros no se presenta ningún resultado nuevo sino que se limita a revisar los conceptos básicos de la Teoría de Códigos y de bases de Gröbner. Al final del capítulo 2 también se puede encontrar un resumen de las ideas expuestas por Di

Biase-Urbanke en [26] para reducir el número de variables involucradas en el cálculo del núcleo de un homomorfismo entre anillos de polinomios, así como una breve introducción a bases de Graver asociadas a ideales tóricos siguiendo los resultados del libro [60].
En la primera parte del capítulo 3 se estudia la relación entre bases de Gröbner y la programación lineal entera introducida por Conti y Traverso en [22] y la adaptación de Ikegami y Kaji [37] de las ideas de Conti y Traverso para resolver un problema de programación lineal modular. La segunda parte del capítulo 3 aporta resultados novedosos que permiten mejorar la eficacia del algoritmo propuesto por Ikegami y Kaji, reduciendo el número de variables involucradas. Es decir se muestra cómo calcular el $\mathbb{Z}_q$-núcleo de la matriz de coeficientes del problema de programación linear modular $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ involucrando sólo las variables del espacio ambiente que contiene el ideal de eliminación que definen Ikegami y Kaji. En realidad, los resultados obtenidos pueden verse como una generalización al caso modular de los resultados presentados en [15] o como una adaptación de la filosofía del algoritmo de Di Biase y Urbanke [26].
En el capítulo 4 se exponen las aplicaciones en Teoría de Códigos y Criptografía obtenidos de los resultados del capítulo anterior. En la primera parte del capítulo se estudian diferentes métodos de descodificación de códigos lineales y en particular métodos específicos del caso binario como el método de descodificación completa que proponen Ikegami y Kaji en [37], y que se ha probado en el capítulo 3 que es equivalente al método propuesto en [15]. También se describen dos algoritmos de descodificación por gradiente propuestos de forma independiente por Liebler en [42] y por Ashikhmin y Barg en [6] y se prueba que ambos algoritmos son duales. En otra de las secciones de este capítulo se presenta la aplicación principal de los resultados del capítulo anterior dando un algoritmo para calcular las palabras de soporte mínimo de un código lineal definido en $\mathbb{Z}_q^n$ y se ilustra este resultado con dos ejemplos realizados con el programa Sage [54]. Por último se describe brevemente la importancia del conjunto de palabras de soporte mínimo de un código dentro de la criptografía, particularmente en el campo de los esquemas para compartir secretos basados en códigos correctores de errores.

Antes de acabar con esta introducción me gustaría, en primer lugar, agradecer a mis tutores, al Dr. D. Antonio Campillo y al Dr. D. Edgar Martínez-Moro, por permitirme realizar este trabajo bajo su dirección como inicio de mi tesis doctoral, por su disponibilidad, su paciencia y su capacidad para guiarme en este proceso de formación como investigadora. También me gustaría agradecer al Dr. D. Alberto Vigneron (Universidad de Cádiz) por sus

comentarios y sugerencias sobre el cálculo del levantamiento de Lawrence para semigrupos con torsión. Finalmente, también quiero expresar mi agradecimiento a la Dra. Dª Evelia García Barroso (Universidad de La Laguna) por el apoyo y el ánimo que siempre me ha transmitido en el difícil pero a la vez apasionante campo de la investigación matemática.

Capítulo 1

Códigos lineales

En este capítulo presentamos algunas nociones fundamentales de la Teoría de Códigos. Tras una breve introducción en la que se recuerdan algunos conceptos y resultados básicos de la Teoría de códigos, se pasa a estudiar con mayor detalle una familia particular de códigos, los llamados códigos lineales. Este capítulo sigue en gran parte el capítulo 2 de [48], así como el libro [36]. También aconsejamos la lectura de Niederreiter [50], autor ya clásico de la Teoría de Códigos, y otros textos como [7, 38, 52, 44].

1.1. Definiciones básicas

La *Teoría de Códigos Correctores* busca la forma de transmitir información de manera fiable y eficiente a través de canales afectados de ruido y que, por lo tanto, pueden distorsionar la información. El objetivo es determinar un sistema de codificación /descodificación que permita recuperar la información emitida a pesar de las alteraciones sufridas por el mensaje en la transmisión.

Se puede atribuir los orígenes de la Teoría de Códigos al artículo [58] que publicó Claude Shannon en 1948. Luego esta disciplina fue creciendo y conectando muchas más áreas de las matemáticas como el álgebra y la combinatoria. En la vida cotidiana, convivimos con muchos códigos, los más comunes son el código de barras, el ISBN usado en los libros y el código ASCII usado en los ordenadores. Los primeros ejemplos de códigos son el código Morse, usado en telegrafía desde el siglo XIX, y el sistema Braille. Además, cualquier mecanismo que transmita o almacene información digital, sonidos o imágenes involucra al menos un código.

En la transmisión de la información intervienen varios factores que precisa-

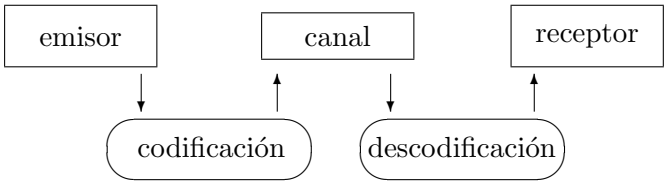

Figura 1.1: Esquema de una transmisión de información codificada.

mos a continuación. Llamamos *canal* al medio de transmisión utilizado como puede ser el cable telefónico, un disco, ondas de radio... Es habitual suponer que los canales satisfacen algunas hipótesis suplementarias. En nuestro caso supondremos que el canal no tiene *pérdidas*, lo que quiere decir que, si se emite un símbolo, siempre se recibe un símbolo; tampoco tiene *memoria*, lo que implica que la transmisión no depende de los símbolos previamente enviados, y, además, cada símbolo tiene una probabilidad p de ser erróneo, lo que se conoce habitualmente como *probabilidad de error del canal*. Supondremos que el canal es *simétrico* es decir, la probabilidad de error en la transmisión de un símbolo es independiente del símbolo emitido; y además el canal *envía información redundante*, lo que significa que, para poder detectar y corregir los errores de transmisión, se utiliza un código llamado del tipo $[n, k]$ donde un mensaje formado por $k \in \mathbb{Z}_{\geq 0}$ símbolos se transforma en una palabra de $n \in \mathbb{Z}_{\geq 0}$, $n > k$, símbolos con $n-k$ *símbolos redundantes*.
El conjunto de símbolos o la información digital que es posible transmitir se caracteriza por presentarse en un formato discreto, llamado *alfabeto* y que denotaremos por $\mathcal{A}$. Si $\mathcal{A}$ tiene q letras (elementos distintos) y q es potencia de un número primo, entonces $\mathcal{A}$ se identifica con el cuerpo finito $\mathbb{F}_q$ de q elementos. Esta identificación permite aplicar a los problemas de codificación las nociones y propiedades de cuerpos finitos.
A una secuencia finita de letras se denomina *palabra*. Denotaremos por $\mathcal{A}^n$ el conjunto de palabras de longitud n formadas con los elementos de $\mathcal{A}$. Un conjunto de palabras con la misma longitud forman un *código de bloque* $\mathcal{C} \subseteq \mathcal{A}^n$.
Al recibir un mensaje, el receptor no puede estar seguro de que alguna parte del mismo haya sido corrompida durante la transmisión, pero puede conocer la frecuencia con que se producen los errores y, por tanto, determinar cuántos errores cabe esperar que hayan ocurrido. Llamaremos *fuente* al dispositivo que emite mensajes elegidos de un conjunto finito de mensajes. La *codificación* es el proceso en el que cada mensaje, antes de ser enviado, se

transforma en una sucesión de palabras del código. Es decir, fijamos dos enteros positivos $k < n$ y troceamos el mensaje $\mathbf{m}$, que se presenta originalmente como una secuencia $\mathbf{m} = x_1 x_2 \cdots x_r \in \mathcal{A}^r$, en bloques de longitud k, es decir:

$$\mathbf{m} = (x_1 \cdots x_k) \cdot (x_{k+1} \cdots x_{2k}) \cdot \ldots \cdot (x_{r-k+1} \cdots x_r).$$

Cada uno de estos bloques se codifica independientemente de los demás 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 constituyen:

$$c(\mathbf{m}) = c(x_1 \cdots x_k) \cdot (x_{k+1} \cdots x_{2k}) \cdot \ldots \cdot (x_{r-k+1} \cdot x_r).$$

El conjunto $\mathcal{C}$ definido por la imagen de la aplicación c forma un código de longitud n.
La *descodificación* es la parte más importante y delicada del proceso. Debemos tener en cuenta que, si en la transmisión no se cometen errores, entonces la palabra recibida es la misma que la emitida, pero esto sólo ocurre en un canal sin ruido. En este caso el proceso de descodificación es sencillo y consiste simplemente en invertir la codificación. En el caso general de tener un canal con ruido, es necesario crear un sistema que nos permita asignar una palabra del código a cualquier palabra recibida, es decir, asignar un elemento del código $\mathcal{C}$ a cualquier elemento de $\mathcal{A}^n$.
Para ello debemos tener en cuenta que si se envía una palabra $\mathbf{c} \in \mathcal{C}$, recibimos un vector $\mathbf{x} \in \mathcal{A}^n$ y p es la probabilidad de que un símbolo resulte alterado en la transmisión, entonces esperamos una media de np símbolos erróneos en x. De ahí que, para que un código sea factible, la *capacidad correctora* del mismo debe superar al menos dicha cota. Observemos que si $\mathbf{x} \notin \mathcal{C}$ deducimos que se han producido errores, sin embargo, aún cuando $\mathbf{x} \in \mathcal{C}$, no podemos estar seguros de que no se hayan cometido errores.
Por todo lo expuesto es necesario que el código se diseñe correctamente de manera que las palabras sean muy *diferentes* unas de otras para que resulte improbable que $\mathbf{x} \in \mathcal{C}$ cuando se hayan cometido errores en el canal.

Definición 1.1. *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. A los elementos de $\mathcal{C}$ se les llama* palabras *de longitud n. Cada palabra de $\mathcal{C}$ contiene k símbolos de información y $n - k$ símbolos redundantes o símbolos de control.*

Ejemplo 1.1. *Consideramos un código binario $\mathcal{C} \subseteq \mathbb{F}_2^n$ y supongamos que queremos enviar los mensajes:*

$$ARRIBA, \quad ABAJO, \quad IZQUIERDA, \quad DERECHA.$$

Vamos a utilizar diferentes códigos, definidos en la siguiente tabla:

Código	*Longitud*	*ARRIBA*	*ABAJO*	*IZQUIERDA*	*DERECHA*
C_1	*2*	*00*	*10*	*01*	*11*
C_2	*3*	*000*	*110*	*011*	*101*
C_3	*6*	*000000*	*111000*	*001110*	*110011*

Observamos que:

- *El código $\mathcal{C}_1$ no tiene capacidad de detectar errores ya que alterando un bit del mensaje obtenemos otra palabra de código.*
- *El código $\mathcal{C}_2$ es capaz de detectar errores individuales ya que alterando un bit obtenemos una palabra que no pertenece al código, sin embargo, no puede corregir errores, ya que si recibimos la palabra* 111*; dicha palabra puede ser resultado de un error en las tres siguientes palabras del código:* 110, 011 *ó* 101*.*
- *El código $\mathcal{C}_3$ es capaz de detectar un error individual y corregirlo (ya que si se comete un único error dos palabras distintas del código no se pueden transformar en la misma).*

Definición 1.2. *Dados dos elementos $\mathbf{x} = (x_1, \dots, x_n)$, $\mathbf{y} = (y_1, \dots, y_n) \in \mathcal{A}^n$, llamamos distancia de Hamming entre $\mathbf{x}$ e $\mathbf{y}$ al número de coordenadas distintas que poseen, es decir al número natural:*

$$d_H(\mathbf{x}, \mathbf{y}) = \sharp\{i \mid 1 \leq i \leq n,\ x_i \neq y_i\}$$

Proposición 1.1. *La función distancia de hamming, d_H, es una distancia en el espacio euclídeo $\mathcal{A}^n$.*

Demostración. En efecto,

1. Sean $\mathbf{a},\ \mathbf{b} \in \mathcal{A}^n$, entonces $d_H(\mathbf{a}, \mathbf{b}) = 0$ si y solamente si $\mathbf{a} = \mathbf{b}$.
2. Además $d_H(\mathbf{a}, \mathbf{b}) = d_H(\mathbf{b}, \mathbf{a})$.
3. De manera que nos bastaría con demostrar que para todo $\mathbf{a},\ \mathbf{b},\ \mathbf{c} \in \mathcal{A}^n$ se tiene que $d_H(\mathbf{a}, \mathbf{b}) \leq d_H(\mathbf{a}, \mathbf{c}) + d_H(\mathbf{c}, \mathbf{b})$.

Observamos que si $\mathbf{a} = \mathbf{b}$, entonces $d_H(\mathbf{a}, \mathbf{b}) = 0$ luego,

$$d_H(\mathbf{a}, \mathbf{c}) + d_H(\mathbf{c}, \mathbf{b}) \geq 0.$$

Además si $\mathbf{a} = \mathbf{c}$ ó $\mathbf{b} = \mathbf{c}$, entonces $d_H(\mathbf{a}, \mathbf{b}) = d_H(\mathbf{a}, \mathbf{c}) + d_H(\mathbf{c}, \mathbf{b}) \geq 0$.
Para el resto de los casos procedamos por inducción sobre el número de coordenadas en que difieren $\mathbf{a}$ y $\mathbf{b}$.

- Supongamos que $\mathbf{a},\ \mathbf{b}$ difieren en una sola coordenada ($d_H(\mathbf{a},\mathbf{b}) = 1$) además, sabemos que $\mathbf{a} \neq \mathbf{b}$ y $\mathbf{b} \neq \mathbf{c}$, entonces

$$d_H(\mathbf{a},\mathbf{c}) + d_H(\mathbf{c},\mathbf{b}) \geq 2 \geq d_H(\mathbf{a},\mathbf{b}).$$

- Supongamos, por hipótesis de inducción, que si $\mathbf{a},\ \mathbf{b}$ se diferencian en k coordenadas con $k < n$, entonces

$$d_H(\mathbf{a},\mathbf{c}) + d_H(\mathbf{c},\mathbf{b}) \geq d_H(\mathbf{a},\mathbf{b}) = k.$$

- Veamos qué ocurre si $d_H(\mathbf{a},\mathbf{b}) \geq k+1$.

  Consideramos cualquier palabra $\mathbf{c} \in \mathcal{A}^n$ y supongamos que existe un índice k tal que $a_k \neq b_k$ y $a_k = c_k$. En este caso definimos:

$$\mathbf{b}' = (b_1, \ldots, b_k, a_{k+1}, b_{k+2}, \ldots, b_n),$$

  Aplicando la hipótesis de inducción, concluimos que:

$$\begin{aligned} d_H(\mathbf{a},\mathbf{b}) = d_H(\mathbf{a},\mathbf{b}') + 1 &\geq d_H(\mathbf{a},\mathbf{c}) + d_H(\mathbf{b}',\mathbf{c}) + 1 \\ &= d_H(\mathbf{a},\mathbf{c}) + d_H(\mathbf{b},\mathbf{c}) \end{aligned}$$

  Si no existe dicho índice, es decir, si para todo índice k tal que $a_k \neq b_k$ se tiene que $a_k \neq c_k$, entonces $d_H(\mathbf{a},\mathbf{b}) \leq d_H(\mathbf{a},\mathbf{c})$. De donde se tiene el resultado.

□

Definición 1.3. *Dada una palabra $\mathbf{x} \in \mathcal{A}^n$, el método de descodificación por mínima distancia consiste en codificar dicha palabra por una palabra del código $\mathbf{c} \in \mathcal{C}$ que verifique que $d_H(\mathbf{x},\mathbf{c}) = min\{d_H(\mathbf{x},\mathbf{a}) \mid \mathbf{a} \in \mathcal{C}\}$.*
En el caso de que la distancia mínima se alcance para más de una palabra del código se deberá optar por un criterio que decida cuál de ellas es la elegida para la descodificación.

Este sistema de descodificación (evaluar la distancia de x a todas las palabras del código $\mathcal{C}$ y quedarnos con la más cercana) es impracticable para códigos grandes debido a su enorme costo; se trata de un problema NP-completo, es decir, que no se puede resolver en tiempo polinomial. Relativamente pocos códigos permiten estos métodos efectivos, por ello uno de los problemas más importantes de la teoría de códigos en la actualidad es encontrar algoritmos rápidos y eficaces para descodificar.

Definición 1.4. *Dado un código $\mathcal{C} \subseteq \mathcal{A}^n$, se llama distancia mínima al entero*

$$d(\mathcal{C}) = min\{d_H(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in \mathcal{C}\}$$

Definición 1.5. *El peso Hamming de un vector $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{A}^n$, que denotaremos por $w_H(\mathbf{x})$, se define como:*

$$w_H(\mathbf{x}) = \sharp\{i \mid 1 \leq i \leq n \ , \ x_i \neq 0\} = d_H(\mathbf{x}, 0)$$

Proposición 1.2. *La aplicación w_H es una norma en $\mathcal{A}^n$ y la distancia hamming d_H es la distancia asociada a dicha norma.*

Demostración. En efecto, observamos que:

1. Para todo $\mathbf{x} \in \mathcal{A}^n$ se tiene que $w_H(\mathbf{x}) = 0$ si y sólo si $\mathbf{x} = 0$.
2. Para todo $\mathbf{x} \in \mathcal{A}^n$ y para todo $k \in \mathcal{A}$ se tiene que $w_H(k\mathbf{x}) = kw_H(\mathbf{x})$.
3. Para todo $\mathbf{x},\ \mathbf{y} \in \mathcal{A}^n$ se verifica que $w_H(\mathbf{x} + \mathbf{y}) \leq w_H(\mathbf{x}) + w_H(\mathbf{y})$

□

Definición 1.6. *Podemos asociar a un código $\mathcal{C} \subseteq \mathcal{A}^n$ su peso mínimo, definido como*

$$w(\mathcal{C}) = min\{w_H(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C},\ \mathbf{c} \neq 0\}$$

Definición 1.7. *Diremos que el código $\mathcal{C} \subseteq \mathcal{A}^n$* detecta *hasta s errores si para cualquier palabra $\mathbf{c} \in \mathcal{C}$ y para cualquier vector $\mathbf{y} \in \mathcal{A}^n$ tal que $d_H(\mathbf{c}, \mathbf{y}) \leq s$ se tiene que $\mathbf{y} \notin \mathcal{C}$.*
Diremos que la capacidad correctora de un código $\mathcal{C} \subseteq \mathcal{A}^n$ es t si siempre que se cometan a lo sumo t errores la descodificación por mínima distancia proporciona la palabra correcta.

Teorema 1.1. *Si en el código $\mathcal{C} \subseteq \mathcal{A}^n$ el número de errores no supera $\left\lfloor \frac{d_H(\mathcal{C})-1}{2} \right\rfloor$, entonces la palabra original coincide con la recibida, donde $\lfloor a \rfloor$ indica el mayor entero menor que a.*

Demostración. Sea $\mathbf{c} \in \mathcal{C}$ la palabra original y sea $\mathbf{z} \in \mathcal{A}^n$ la palabra recibida que contiene e errores, es decir:

$$d_H(\mathbf{c}, \mathbf{z}) = e \leq \left\lfloor \frac{d_H(\mathcal{C}) - 1}{2} \right\rfloor.$$

Entonces, para cada $\mathbf{b} \in \mathcal{C}$ tenemos que:

$$d_H(\mathcal{C}) \leq d_H(\mathbf{b}, \mathbf{c}) \leq d_H(\mathbf{b}, \mathbf{z}) + d_H(\mathbf{z}, \mathbf{c}) \leq d_H(\mathbf{b}, \mathbf{z}) + e$$

Como $e \leq \left\lfloor \frac{d_H(\mathcal{C})-1}{2} \right\rfloor$, entonces $d_H \geq 2e+1$ y tenemos que:

$$2e+1 \leq d_H(\mathbf{b}, \mathbf{z}) + e,$$

es decir, $d_H(\mathbf{b}, \mathbf{z}) \geq e+1$, de donde deducimos que $\mathbf{c}$ es la única palabra del código $\mathcal{C}$ que se encuentra a una distancia e de $\mathbf{z}$. □

Definición 1.8. *Al entero* $t := \left\lfloor \frac{d_H(\mathcal{C})-1}{2} \right\rfloor$ *se le denomina capacidad correctora del código.*

Con la estrategia de descodificación por mínima distancia, podemos detectar $d_H(\mathcal{C})-1$ errores y corregir $\left\lfloor \frac{d_H(\mathcal{C})-1}{2} \right\rfloor$, tal y como probamos en el Teorema 1.1.
La proporción entre el número de mensajes M y el número posible de palabras código q^n (donde q es el número de letras del alfabeto $\mathcal{A}$) es una forma de medir el exceso de información que se transmite, es decir el costo del proceso. Observemos que si el número de mensajes es $M = q^k$, entonces la forma de expresar esta cantidad es $R(\mathcal{C}) = \frac{k}{n}$.

Definición 1.9. *Se llama tasa de transmisión al número* $R(\mathcal{C}) = \frac{k}{n}$ *que mide la proporción entre los símbolos que contienen información y el número total de símbolos empleados.*

De esta forma, cuanto mayor es el número de símbolos de control (o redundancia) menor será esta tasa. Observemos que $0 < R(\mathcal{C}) \leq 1$ y que en el caso de que $R(\mathcal{C}) = 1$ se emite exactamente el mínimo posible de información para poder transmitir el mensaje; por el contrario, cuanto más cerca del 0 está esta medida, más caro es el código. Como es lógico, lo deseable es que esta medida esté cerca del 1.

Ejemplo 1.2. *El código ASCII (siglas en inglés de American Standard Code for Information Interchange) es un código de caracteres basado en el alfabeto latino que fue creado en* 1963 *por el Comité Estadounidense de Estándares, como una evolución de los conjuntos de códigos utilizados en telegrafía.*
En su versión habitual este código permite codificar $128 = 2^7$ *símbolos. A cada símbolo se le asigna un número de orden y se codifica mediante la escritura binaria de dicho número utilizando* 7 *bits. Para aumentar la fiabilidad de esta codificación, a cada upla* $(x_1, \ldots, x_7) \in \mathbb{F}_2^7$ *se le añade un bit de control* x_8 *de manera que:*

$$x_1 + \ldots + x_7 + x_8 \equiv 0 \bmod 2$$

Este sistema permite detectar, pero no corregir, cualquier número impar de errores. Observemos que la tasa de transmisión de este código es $\frac{7}{8} = 0,875$*, luego no se trata de un código muy costoso.*

1.2. Códigos lineales

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

Definición 1.10. *Un subconjunto* $\mathcal{C} \subseteq F_q^n$ *diremos que es un* $[n, k, d]$ *código lineal si* $\mathcal{C}$ *es un subespacio vectorial de dimensión* k *de* $\mathbb{F}_q^n$*, su longitud es* n *y su distancia mínima es* d*.*
Es decir, si la aplicación de codificación $c : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$ *es lineal, diremos que el conjunto imagen,* $Im(c)$*, es un código lineal.*

Veamos cómo se caracterizan los procesos de codificación y descodificación de este tipo de códigos. Dado que un código lineal es un subespacio de $\mathbb{F}_q^n$, para describirlo basta con dar una base del mismo.

Definición 1.11. *Llamamos matriz generatriz del código* $\mathcal{C} \subseteq \mathbb{F}_q^n$ *a la matriz de dimensión* $k \times n$ *que tiene por filas los vectores de la base elegida.*
De esta forma, si $\{\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(k)}\}$ *es una base del código* $\mathcal{C}$ *y consideramos* $\mathbf{c}^{(i)} = (c_1^{(i)}, \ldots, c_n^{(i)})$ *para todo* $i \in \{1, \ldots, n\}$*, podemos definir la matriz generatriz* $G_{\mathcal{C}}$ *del código* $\mathcal{C}$ *como:*

$$G_{\mathcal{C}} = \begin{pmatrix} \mathbf{c}^{(1)} \\ \mathbf{c}^{(2)} \\ \vdots \\ \mathbf{c}^{(k)} \end{pmatrix} = \begin{pmatrix} c_1^{(1)} & c_2^{(1)} & \cdots & c_n^{(1)} \\ c_1^{(2)} & c_2^{(2)} & \cdots & c_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ c_1^{(k)} & c_2^{(k)} & \cdots & c_n^{(k)} \end{pmatrix} \in \mathbb{F}_q^{k \times n}$$

Dado que el código $\mathcal{C}$ es el conjunto de las combinaciones lineales de los vectores de la base, es decir:

$$\mathcal{C} = \{\lambda_1 \cdot \mathbf{c}^{(1)} + \ldots + \lambda_k \cdot \mathbf{c}^{(k)} \mid \lambda_i \in \mathbb{F}_q,\ i \in \{1, \ldots, k\}\},$$

se tiene que cualquier código lineal $\mathcal{C} \subseteq \mathbb{F}_q^n$ se puede describir en función de su matriz generatriz como $\mathcal{C} = \{\mathbf{x} \cdot G \mid \mathbf{x} \in \mathbb{F}_q^k\}$. De esta forma, dado un mensaje $\mathbf{m} \in \mathbb{F}_q^k$, dicho mensaje se codifica por $\mathbf{m} \cdot G_{\mathcal{C}} \in \mathbb{F}_q^n$.
Como una base de $\mathcal{C}$ no es única, tampoco lo es una matriz generatriz, por ello tiene sentido buscar dentro de las matrices posibles la más sencilla.

A veces es interesante cuando se codifica una palabra $\mathbf{m} \in \mathbb{F}_q^n$, que la palabra codificada contenga como subpalabra a $\mathbf{m}$, esto es que sea de la forma $(\mathbf{m}, \mathbf{z})$ con $\mathbf{z} \in \mathbb{F}_q^{n-k}$. De este modo los primeros k símbolos contienen la información y los siguientes serían símbolos de control o redundantes.

Definición 1.12. *Diremos que una matriz generatriz de un código $\mathcal{C}$ es estándar si tiene la forma $G_{\mathcal{C}} = (I_k, A)$ donde I_k denota la matriz identidad de tamaño $k \times k$. De esta forma, si disponemos de una matriz estándar, la operación de codificación asociada es:*

$$\mathbf{m} \cdot G_{\mathcal{C}} = (m_1, \ldots, m_k, x_{k+1}, \ldots x_n)$$

Esta codificación denominada codificación sistemática conserva los símbolos del mensaje original en k posiciones fijas.

Definición 1.13. *Diremos que dos códigos $\mathcal{C}_1, \mathcal{C}_2$ de la misma longitud n son equivalentes si uno se puede obtener del otro mediante permutaciones de las coordenadas y multiplicación de ciertas coordenadas por un escalar fijo no nulo. Dichas operaciones no alteran los parámetros del código.*

Proposición 1.3. *Todo código es equivalente por permutaciones a un código sistemático. Es decir, todo código $\mathcal{C}$ es equivalente a un código $\mathcal{C}'$ que admite una matriz generatriz en forma estándar.*

Demostración. La demostración consiste en la descripción del algoritmo que, mediante operaciones elementales, permite pasar de una matriz generatriz cualquiera, $G_{\mathcal{C}}$, a una matriz generatriz estándar.
Partimos de una matriz $G_{\mathcal{C}}$ que escribiremos de la forma:

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

A partir de esta matriz, aplicamos el algoritmo de Gauss para obtener una matriz de la forma:

$$\begin{pmatrix} 1 & g_{12} & g_{13} & \cdots & g_{1k} & \cdots & g_{1n} \\ 0 & 1 & g_{23} & \cdots & g_{2k} & \cdots & g_{2n} \\ 0 & 0 & 1 & \cdots & g_{3k} & \cdots & g_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \cdots & g_{kn} \end{pmatrix}$$

Para terminar basta con convertir en cero los elementos que se encuentran sobre los unos de la diagonal, lo que se consigue restando a la fila i-ésima todas las filas situadas por debajo multiplicada por la constante necesaria. □

Gracias a este teorema no se pierde generalidad trabajando con códigos lineales en forma estándar.
Sabemos que otra forma de caracterizar un subespacio vectorial es mediante sus ecuaciones implícitas (conjunto de soluciones de un sistema lineal homogéneo). Esta caracterización origina la siguiente definición:

Definición 1.14. *Dado $[n, k, d]$ código lineal $\mathcal{C} \subseteq \mathbb{F}_q^n$ diremos que una matriz $H_\mathcal{C}$ de tamaño $(n-k) \times n$ es una matriz de control del código $\mathcal{C}$ si para todo vector $\mathbf{x} \in \mathbb{F}_q^n$ se verifica que $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{C}$ si y solamente si $H_\mathcal{C} \cdot \mathbf{x}^t = 0$.*

En otras palabras, si escribimos

$$H_\mathcal{C} = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-k,1} & h_{n-k,2} & \cdots & h_{n-k,n} \end{pmatrix},$$

entonces el código $\mathcal{C}$ es el subespacio de soluciones del sistema de ecuaciones homogéneo:

$$\begin{gathered} h_{1,1}x_1 + h_{1,2}x_2 + \dots + h_{1,n}x_n = 0 \\ h_{2,1}x_1 + h_{2,2}x_2 + \dots + h_{2,n}x_n = 0 \\ \cdots \\ h_{n-k,1}x_1 + h_{n-k,2}x_2 + \dots + h_{n-k,n}x_n = 0 \end{gathered}$$

A cada una de las ecuaciones anteriores se les llama *ecuaciones de control del código*. Como en el caso de las matrices generatrices, las matrices de control no son únicas.

Proposición 1.4. *$G_\mathcal{C}$ y $H_\mathcal{C}$ son las matrices generatriz y de control de un código $\mathcal{C}$ si y sólo si $H_\mathcal{C}^t \cdot G_\mathcal{C} = 0$*

Demostración. Decir que $H_\mathcal{C}^t \cdot G_\mathcal{C} = 0$ equivale a decir que para todo vector columna $\mathbf{c}$ de $G_\mathcal{C}^t$ se tiene que $H_\mathcal{C} \cdot \mathbf{c} = 0$. Como los vectores columna de $G_\mathcal{C}^t$ que equivalen a los vectores filas de $G_\mathcal{C}$ forman una base del espacio vectorial $\mathcal{C}$, entonces la condición de que $H_\mathcal{C}^t \cdot G_\mathcal{C} = 0$ equivale a decir que

$\mathcal{C} \subseteq \ker(H_{\mathcal{C}})$. Sabemos que $H_{\mathcal{C}}$ es de rango $n-k$ y que la dimensión de $\ker(H_{\mathcal{C}})$ es $n-(n-k) = dim(\mathcal{C})$, de donde se sigue la igualdad $\ker(H_{\mathcal{C}}) = \mathcal{C}$ □

Supongamos que $H_{\mathcal{C}} \in \mathbb{F}_q^{(n-k)\times n}$ es una matriz de control de un código lineal $\mathcal{C} \subseteq \mathbb{F}_q^n$. Determinar una matriz generatriz de $\mathcal{C}$ es equivalente a encontrar una base del subespacio vectorial $\mathcal{C}$ de dimensión k. Esta base coincide con el núcleo de la función lineal definida entre $\mathbb{F}_q$-espacios vectoriales como sigue:

$$\begin{array}{rccc} \psi: & \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^{n-k} \\ & \mathbf{x} = (x_1, \ldots, x_n) & \longmapsto & (H \cdot \mathbf{x}^t)^t \end{array}$$

Lema 1.1. *En un código lineal, la distancia mínima es igual al peso mínimo.*

Demostración. En efecto, pues $d_H(\mathbf{x}, \mathbf{y}) = w_H(\mathbf{x} - \mathbf{y})$, por lo tanto:

$$d(\mathcal{C}) = min\{d_H(\mathbf{x}, \mathbf{y}) \mid \mathbf{x},\ \mathbf{y} \in \mathcal{C}\} = min\{w_H(\mathbf{x} - \mathbf{y}) \mid \mathbf{x},\ \mathbf{y} \in \mathcal{C}\} = w(\mathcal{C})$$

□

La distancia mínima de un código se puede obtener mediante su matriz de control.

Proposición 1.5. *Si $H_{\mathcal{C}}$ es una matriz de control del código lineal $\mathcal{C} \subseteq \mathbb{F}_q^n$, entonces $d(\mathcal{C}) = d$ si y sólo si d es el mayor entero para el que $d-1$ columnas cualesquiera de $H_{\mathcal{C}}$ son linealmente independientes.*

Demostración. Para probar este resultado, denotemos $\mathbf{h}^{(i)}$ a la i-ésima columna de la matriz de control $H_{\mathcal{C}}$ del código $\mathcal{C} \subseteq \mathbb{F}_q^n$, de forma que:

$$H_{\mathcal{C}} = (\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(n)}).$$

Sea $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$, entonces tenemos que

$$H_{\mathcal{C}} \cdot \mathbf{x}^t = x_1 \cdot \mathbf{h}^{(1)} + x_2 \cdot \mathbf{h}^{(2)} + \ldots + x_n \cdot \mathbf{h}^{(n)}.$$

Supongamos que el código $\mathcal{C} \subseteq \mathbb{F}_q^n$ tiene distancia mínima d, y sea $\mathbf{x} \in \mathcal{C}$ un vector de peso d. Como $\mathbf{x} \in \mathcal{C}$ y $H_{\mathcal{C}}$ es una matriz de control de dicho código, se tiene que: $0 = H_{\mathcal{C}}\mathbf{x}^t = x_1 \cdot \mathbf{h}^{(1)} + x_2 \cdot \mathbf{h}^{(2)} \ldots + x_n \cdot \mathbf{h}^{(n)}$.
Es decir, como $\mathbf{x} \in \mathcal{C}$ tiene peso d esto quiere decir que tiene exactamente d coordenadas distintas de cero, por tanto, en la expresión anterior, tenemos una combinación lineal de los vectores columnas en la cual aparecen involucrados exactamente d de dichos vectores. De todo ello se deduce que hay d columnas de H que son linealmente dependientes.

Recíprocamente, si tenemos una combinación lineal de $m < d$ columnas de $H_{\mathcal{C}}$, es decir, si suponemos que: $0 = x_1 \cdot \mathbf{h}^{(1)} + \ldots + x_m \cdot \mathbf{h}^{(m)}$ completando el resto de coeficientes con ceros tenemos una expresión de la forma:

$$0 = x_1 \cdot \mathbf{h}^{(1)} + \ldots + x_n \cdot \mathbf{h}^{(n)}.$$

Así $\mathbf{x} = (x_1, \ldots, x_n)$ es una palabra del código con peso igual a m. De lo que se deduce que $d = d(\mathcal{C}) = w(\mathcal{C}) \geq m$. □

Corolario 1.1. *La distancia mínima de un código lineal $[n, k, d]$ verifica que $d \leq n - k + 1$.*

Demostración. Este resultado es consecuencia inmediata de que el rango de una matriz de control de un código $\mathcal{C} \subseteq \mathbb{F}_q^n$ es $(n - k)$. □

Definición 1.15. *A la cota que define el corolario 1.1 se le denomina cota Singleton.*

Definición 1.16. *Los códigos lineales que alcanzan la igualdad en la cota Singleton, es decir $d = n - k + 1$, se llaman códigos de máxima distancia de separación (o MSD).*

Definición 1.17. *Sea $\mathcal{C} \subseteq \mathbb{F}_q^n$ un $[n, k, d]$ código lineal. Se llama código dual del código $\mathcal{C}$ al código $\mathcal{C}^\perp$ definido por:*

$$\mathcal{C}^\perp = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{x} \cdot \mathbf{c} = 0 \ \forall \mathbf{c} \in \mathcal{C}\}.$$

De esta definición se deduce que si $G_{\mathcal{C}}$ (respectivamente $H_{\mathcal{C}}$) es una matriz generatriz (respectivamente de control) del código $\mathcal{C} \in \mathbb{F}_q^n$, entonces $H_{\mathcal{C}}$ es una matriz generatriz de $C^\perp \subseteq \mathbb{F}_q^n$ y $G_{\mathcal{C}}$ es una matriz de control de $\mathcal{C}^\perp$. Obviamente si $\mathcal{C}$ tiene dimensión k, entonces $\mathcal{C}^\perp$ tiene dimensión $(n-k)$. Con respecto a la distancia mínima de $\mathcal{C}^\perp$, en general no es posible determinarla únicamente en términos de la distancia mínima de $\mathcal{C}$.

Definición 1.18. *Diremos que un código lineal es autodual cuando coincide con su código dual, esto es: $\mathcal{C} = \mathcal{C}^\perp$.*

Observemos que si $\mathcal{C}$ es autodual, entonces $n - k = k$, de lo que se deduce que $n = 2k$. Luego, para que un código sea autodual es necesario que n sea par. La dimensión de un código autodual es $\frac{n}{2}$. Además, en un código autodual $\mathcal{C}$ toda matriz generatriz de $\mathcal{C}$ es al mismo tiempo una matriz de control.

Capítulo 2

Bases de Gröbner

En este capítulo abordaremos el estudio de las bases de Gröbner, para comprender su aplicación a la programación lineal entera. Por motivos de espacio y el objetivo de esta memoria, presentaremos de manera sucinta estos temas, pero existen varias referencias excelentes sobre bases de Gröbner y sus aplicaciones tales como [4, 9, 24, 25, 35, 62] y una recopilación teórica en [49]. Este capítulo sigue en gran parte el primer capítulo de [4] y de [24].

2.1. Introducción a las Bases de Gröbner

2.1.1. Preliminares

En esta sección fijaremos la notación que utilizaremos a lo largo de este capítulo y repasaremos algunas definiciones y resultados conocidos.
Denotaremos por $\mathbb{N}$ al conjunto de números enteros no negativos es decir $\mathbb{N} = \{0, 1, 2 \ldots\}$ y $\mathbb{K}$ a un cuerpo conmutativo arbitrario. En la mayoría de los casos trabajaremos con 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.
Un *monomio* en las variables $x_1, \ \ldots, \ x_n$ es un producto de la forma

$$\mathbf{x}^{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \ \cdots \ x_n^{\alpha_n}$$

que identificaremos con los vectores $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$.
El *grado total de un monomio* $\mathbf{x}^{\alpha}$ lo denotaremos por $|\alpha| = \alpha_1 + \ldots + \alpha_n$.
Si $\alpha = (0, \ldots, 0)$, entonces $\mathbf{x}^{\alpha} = 1$ que es el único monomio de grado 0.
Un *polinomio* f en $x_1, \ldots, x_n$ con coeficientes en $\mathbb{K}$, es una combinación lineal finita de monomios en las variables $x_1, \ldots, x_n$ con coeficientes en $\mathbb{K}$. Al

conjunto de todos los polinomios en las variables $x_1, \ldots, x_n$ con coeficientes en $\mathbb{K}$, lo denotaremos por $\mathbb{K}[\mathbf{x}] := K[x_1, \ldots, x_n]$ y lo llamaremos anillo de polinomios en n indeterminadas con coeficientes en $\mathbb{K}$. Es decir, todo polinomio $f(\mathbf{x}) \in \mathbb{K}[\mathbf{x}]$ se escribe como

$$f(\mathbf{x}) = \sum_{\alpha \in \Lambda} \lambda_\alpha \mathbf{x}^\alpha \in \mathbb{K}[\mathbf{x}] \text{ con } \lambda_\alpha \in \mathbb{K} \ \forall \alpha \in \Lambda,$$

donde Λ es un subconjunto finito de $\mathbb{N}^n$.

2.1.2. Órdenes monomiales

Hemos visto que podemos establecer una aplicación biyectiva entre el conjunto de monomios en $\mathbb{K}[\mathbf{x}]$ y $\mathbb{N}^n$ identificando cada monomio $\mathbf{x}^\alpha \in \mathbb{K}[\mathbf{x}]$ con su exponente $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$. Luego, cualquier orden $>$ en $\mathbb{N}^n$ define un orden en el conjunto de monomios de $\mathbb{K}[\mathbf{x}]$:

$$\alpha > \beta \ \Leftrightarrow \ \mathbf{x}^\alpha \succ \mathbf{x}^\beta.$$

Los órdenes totales en $\mathbb{N}^n$ que sean compatibles con la estructura algebraica de anillo de polinomios en varias variables, $\mathbb{K}[\mathbf{x}]$, son conocidos como *órdenes monomiales*.

Definición 2.1. *Un* orden monomial *sobre* $\mathbb{K}[\mathbf{x}]$ *es una relación* $\succ$ *sobre el conjunto de los monomios* $\mathbb{K}[\mathbf{x}]$*, o equivalentemente, una relación sobre* $\mathbb{N}^n$*, que verifica:*

1. $\succ$ *es un orden total sobre* $\mathbb{N}^n$*.*
2. *Dados* $\alpha, \ \beta, \ \gamma \in \mathbb{N}^n$*, se tiene que* $\alpha \succ \beta \ \Leftrightarrow \ \alpha + \gamma \succ \beta + \gamma$*.*

Si además el orden monomial $\succ$ *verifica la siguiente propiedad, diremos que se trata de un* orden monomial global*.*

3. $\succ$ *es un buen orden sobre* $\mathbb{N}^n$*, es decir, que todo subconjunto no vacío de* $\mathbb{N}^n$ *tiene un elemento minimal para* $\succ$*.*

El siguiente Lema nos permite simplificar la definición de orden monomial global.

Lema 2.1. *Sea* $\succ$ *un orden monomial sobre* $\mathbb{K}[\mathbf{x}]$*, diremos que* $\succ$ *es un buen orden si y sólo si*

3'. $0 \prec \alpha$ *para todo* $\alpha \in \mathbb{N}^n \setminus \{0\}$*.*

Como todos los órdenes monomiales que vamos a considerar en este trabajo son globales, para simplificar la notación, denominaremos órdenes monomiales aquellos que verifican las condiciones 1, 2 de la definición 2.1 y la condición $3'$ del Lema 2.1.
Definimos a continuación los órdenes monomiales más comunes sobre $\mathbb{K}[\mathbf{x}]$.

- *Orden Lexicográfico (lex)*

  Dados $\alpha,\ \beta \in \mathbb{N}^n$, diremos que $\alpha \succ_{lex} \beta$ si el primer término no nulo del vector $\alpha - \beta \in \mathbb{N}^n$ es positivo.

- *Orden Lexicográfico Graduado (grlex)*

  Dados $\alpha,\ \beta \in \mathbb{N}^n$, diremos que $\alpha \succ_{grlex} \beta$ si:

$$\begin{cases} |\alpha| = \sum_{i=1}^n \alpha_i \succ \sum_{i=1}^n \beta_i = |\beta| \\ \text{O bien, } |\alpha| = |\beta| \text{ y } \alpha \succ_{lex} \beta. \end{cases}$$

- *Orden Lexicográfico Graduado Inverso (grevlex)* Dados $\alpha,\ \beta \in \mathbb{N}^n$, diremos que $\alpha \succ_{grlex} \beta$ si:

$$\begin{cases} |\alpha| = \sum_{i=1}^n \alpha_i \succ \sum_{i=1}^n \beta_i = |\beta|, \text{ o bien,} \\ |\alpha| = |\beta| \text{ y el primer término no nulo de } \alpha - \beta \in \mathbb{N}^n \text{ es negativo.} \end{cases}$$

Definición 2.2. *Sea $r \in \{1, \ldots, n\}$. Se dice que un orden monomial $\succ$ sobre $\mathbb{K}[\mathbf{x}]$ es de* eliminación *para r primeras variables si cualquier monomio en el que aparezca alguna de las variables $x_1, \ldots, x_r$ es mayor que todos los monomios de $\mathbb{K}[x_{r+1}, \ldots, x_n]$.*

Observemos que el orden lexicográfico es un orden de eliminación para las r primeras variables, con $r \in \{1, \ldots, n\}$.

Definición 2.3. *Sea $f = \sum_\alpha \lambda_\alpha \mathbf{x}^\alpha$ un polinomio no nulo en $\mathbb{K}[\mathbf{x}]$ y $\succ$ un orden monomial sobre $\mathbb{K}[\mathbf{x}]$. Se define:*

- Exponente *de f como* $\exp(f) = \max\{\alpha \in \mathbb{N}^n \mid \lambda_\alpha \neq 0\}$.
- Coeficiente líder o inicial *de f a* $\mathrm{LC}_\succ(f) = \lambda_{\exp(f)} \in \mathbb{K}$.
- Monomio líder o inicial *de f a* $\mathrm{LM}_\succ(f) = \mathbf{x}^{\exp(f)} \in \mathbb{K}[\mathbf{x}]$.
- Término líder o inicial *de f a*

$$\mathrm{LT}_\succ(f) = \mathrm{LC}_\succ(f) \cdot \mathrm{LM}_\succ(f) = \lambda_{\exp(f)} \cdot \mathbf{x}^{\exp(f)}.$$

Lema 2.2. *Sean* $f,\ g \in \mathbb{K}[\mathbf{x}]$ *polinomios no nulos. Se verifica que:*

1. $\exp(f \cdot g) = \exp(f) + \exp(g)$.
2. $\exp(f+g) \leq \max\{\exp(f), \exp(g)\}$.
   Además, se verifica la igualdad si $\mathrm{LT}(f) - \mathrm{LT}(g) \neq 0$.

2.1.3. Ideales monomiales y el Lema de Dickson

Definición 2.4. *Un* ideal I *de* $\mathbb{K}[\mathbf{x}]$ *es un subgrupo aditivo de* $\mathbb{K}[\mathbf{x}]$ *cerrado para el producto por elementos de* $\mathbb{K}[\mathbf{x}]$. *Es decir,* $I \subseteq \mathbb{K}[\mathbf{x}]$ *es un ideal si y sólo si:*

1. $f - g \in I$, *para todo* $f,\ g \in I$.
2. $f \cdot g \in I$, *para todo* $f \in \mathbb{K}[\mathbf{x}]$ *y* $g \in I$.

Definición 2.5. *Un conjunto de polinomios* $\{f_i\}_{i\in\Lambda}$ *de* $\mathbb{K}[\mathbf{x}]$ *es un* sistema de generadores de un ideal I *si todos los polinomios* $f \in I$ *son de la forma*

$$\sum_{i\in\Lambda} \lambda_i f_i \ \text{ con } \lambda_i \in \mathbb{K}.$$

En este caso diremos que el ideal I *está generado por el conjunto de polinomios* $\{f_i\}_{i\in\Lambda}$ *y lo denotaremos:*

$$I = \langle f_i \rangle_{i\in\Lambda} = \langle f_i \mid i \in \Lambda \rangle .$$

Un ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ *se dice que está* finitamente generado *si posee un sistema de generadores formado por un número finito de polinomios.*

Definición 2.6. *Un ideal* $I \subseteq \mathbb{K}[\mathbf{x}]$ *es un* ideal monomial *si admite un sistema de generadores formado por monomios. Es decir, si existe un subconjunto* $A \subseteq \mathbb{N}^n$ *tal que:*

$$I = \langle \mathbf{x}^\alpha \mid \alpha \in A \rangle$$

Definición 2.7. *Se dice que un monomio* $\mathbf{x}^\beta$ *es divisible por* $\mathbf{x}^\alpha$ *si*

$$\mathbf{x}^\beta = \mathbf{x}^\gamma \cdot \mathbf{x}^\alpha \ \text{, para algún } \gamma \in \mathbb{N}^n.$$

Es decir, si $\beta = \gamma + \alpha$, *para algún* $\gamma \in \mathbb{N}^n$.

El siguiente Lema nos permite caracterizar los monomios que pertenecen a un ideal monomial.

Lema 2.3. *Sea $I = \langle \mathbf{x}^\alpha \mid \alpha \in A \rangle$ un ideal monomial de $\mathbb{K}[\mathbf{x}]$. Entonces $\mathbf{x}^\beta \in I$ si y sólo si $\mathbf{x}^\beta$ es divisible por $\mathbf{x}^\alpha$ para algún $\alpha \in A$.*

Demostración. Véase [24]. Capítulo 2.4, Lema 2. □

El siguiente Lema nos demuestra que para determinar si un polinomio f pertenece a un ideal monomial basta estudiar los monomios de f.

Lema 2.4. *Sea I un ideal monomial de $\mathbb{K}[\mathbf{x}]$ y sea $f \in \mathbb{K}[\mathbf{x}]$. Entonces las siguientes afirmaciones son equivalentes:*

- *$f \in I$.*
- *Todos los términos de f están en I.*
- *f es combinación finita de monomios de I.*

Demostración. Véase [24]. Capítulo 2.5, Lema 3. □

Corolario 2.1. *Sean $I,\ J \in \mathbb{K}[\mathbf{x}]$ dos ideales monomiales, se tiene que $I = J$ si y sólo si I y J contienen los mismos monomios.*

Demostración. Este Corolario es consecuencia inmediata del Lema 2.4 que nos dice que todo ideal monomial está determinado de forma única por sus monomios. □

El resultado principal de esta sección es que todo ideal monomial de $\mathbb{K}[\mathbf{x}]$ está finitamente generado. Este resultado se recoge en el Lema de Dickson.

Teorema 2.1 (Lema de Dickson)**.**
Todo ideal monomial $I = \langle \mathbf{x}^\alpha \mid \alpha \in A \rangle \subseteq \mathbb{K}[\mathbf{x}]$ se puede expresar de la forma $I = \langle \mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_s} \rangle$ con $\alpha_1, \ldots, \alpha_s \in A$.
Es decir I posee un sistema finito de generadores monomiales.

Demostración. Véase [24]. Capítulo 2.4, Teorema 5. □

2.1.4. Algoritmo de la división

Fijado un orden monomial $\succ$ en $\mathbb{N}^n$ y dada $(f_1, \ldots, f_s)$ una s-upla ordenada de polinomios de $\mathbb{K}[\mathbf{x}]$, introducimos la siguiente notación:

$$\begin{cases} \Delta_1 & := & (\exp(f_1) + \mathbb{N}^n) \\ \Delta_2 & := & (\exp(f_1) + \mathbb{N}^n) - \Delta_1 \\ \cdots & & \\ \Delta_s & := & (\exp(f_s) + \mathbb{N}^n) - \bigcup_{i=1}^{s-1} \Delta_i \\ \overline{\Delta} & := & \mathbb{N}^n - \bigcup_{i=1}^{s} \Delta_i \end{cases}$$

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

Teorema 2.2 (Teorema de la división)**.**
Fijado un orden monomial $\succ$ en $\mathbb{N}^n$ y dada una s-upla ordenada de polinomios $(f_1, \ldots, f_s)$ de $\mathbb{K}[\mathbf{x}]$. Para todo polinomio $f \in \mathbb{K}[\mathbf{x}]$, existen unos únicos $h_1, \ldots, h_s,\ r \in \mathbb{K}[\mathbf{x}]$ tales que:

1. $f = h_1 \cdot f_1 + \ldots + h_s \cdot f_s + r$.
2. *Si $r \neq 0$: entonces r es una combinación lineal de monomios no divisibles por los monomios $\{\mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s)\}$.*
   Es decir, $r = \sum_{\alpha \in B} c_\alpha \cdot \mathbf{x}^\alpha$ con $\alpha \in \overline{\Delta}$.
3. *Si $h_i \cdot f_i \neq 0$ entonces $\exp(f) \geq \exp(h_i \cdot f_i)$.*
   Es decir, $h_i = \sum_{\beta_i \in A} c_{\beta_i} \cdot \mathbf{x}^{\beta_i}$ con $\beta_i + \exp(f_i) \in \Delta_i$, $\forall i \in \{1, \ldots, s\}$.

Demostración. Véase [24]. Capítulo 2.3, Teorema 3. □

Para entender el Algoritmo 1 basta observar que:

- La variable d representa el dividendo, la variable r el resto y las variables $h_1, \ldots, h_s$ los cocientes en cada etapa.
- La variable booleana `SeRealizaDivision` nos indica si el término inicial del dividendo es divisible por algún $\mathrm{LT}(f_i)$.
- El segundo bucle WHILE realiza las siguientes acciones:
  - Si algún $\mathrm{LT}(f_i)$ divide a $\mathrm{LT}(d)$, entonces el algoritmo procede como en el caso de una variable.
  - Si ningún $\mathrm{LT}(f_i)$ divide a $\mathrm{LT}(d)$, entonces el algoritmo añade $\mathrm{LT}(d)$ a r y se lo sustrae a d.

Hay dos diferencias fundamentales con la división euclídea sobre el anillo de polinomios de una variable, $\mathbb{K}[x]$:

1. La primera es que realizamos la división por más de un elemento.
2. La segunda es que nuestro resultado depende del orden de la s-upla $f_1, \ldots, f_s$ y del orden monomial $\succ$ elegido.

Nótese que en el caso de una sola variable existe un único orden admisible dado por el grado.

Algoritmo 1 Algoritmo de la división

INPUT: $f_1, \ldots, f_s, f$.
OUTPUT: $h_1, \ldots, h_s, r$, tales que $f = h_1 \cdot f_1 + \ldots + h_s \cdot f_s + r$

```
h_1 := 0, ..., h_s := 0, r := 0
d := f
while p ≠ 0 do
  i := 1
  SeRealizaDivision := false
  while i ≤ s AND SeRealizaDivision = false do
    if LT(f_i) divide LT(p) then
      h_i := h_i + LT(p)/LT(f_i)
      d := d − (LT(p)/LT(f_i)) · f_i
      SeRealizaDivision := true
    else
      i := i + 1
    end if
  end while
  if SeRealizaDivision := false then
    r := r + LT(p)
    d := d − LT(p)
  end if
end while
Devolver h_1, ..., h_s, r.
```

Ejemplo 2.1. *Consideramos los polinomios* $f_1 = y^2 - 1$, $f_2 = xy - 1$ *y* $g = x^2y + xy^2 + y^2 \in \mathbb{R}[x, y]$. *Fijamos como orden monomial el orden lexicográfico y como orden en las variables* $x > y$.
Dividimos g *entre el par* $\{f_1, f_2\}$ *utilizando el algoritmo de la división (Algoritmo 1).*
En primer lugar calculamos Δ_1, Δ_2 *y* $\overline{\Delta}$.

$$\begin{array}{lll} \exp(f_1) = (0,2); & \mathrm{LT}(f_1) = y^2 & \Rightarrow \quad \Delta_1 = (0,2) + \mathbb{N}^2 \\ \exp(f_2) = (1,1); & \mathrm{LT}(f_2) = xy & \Rightarrow \quad \Delta_2 = \big((1,1) + \mathbb{N}^2\big) - \Delta_1 \\ \Rightarrow & \overline{\Delta} = \mathbb{N}^2 - (\Delta_1 \cup \Delta_2) & \end{array}$$

Observamos que:

1. $\exp(g) = (2,1) \in \Delta_2$, *definimos:*
   - $h^{(1)} = \frac{\mathrm{LT}(g)}{\mathrm{LT}(f_2)} = \frac{x^2y}{xy} = x$
   - $g^{(1)} = g - h^{(1)} f_2 = x^2y + xy^2 + y^2 - x^2y + x = xy^2 - x + y^2 \neq 0$
2. $\exp(g^{(1)}) = (1,2) \in \Delta_1$, *definimos:*
   - $h^{(2)} = \frac{\mathrm{LT}(g^{(1)})}{\mathrm{LT}(f_1)} = \frac{xy^2}{y^2} = x$
   - $g^{(2)} = g^{(1)} - x f_1 = xy^2 + x + y^2 - xy^2 + x = y^2 + 2x \neq 0$
3. $\exp(g^{(2)}) = (1,0) \in \overline{\Delta}$, *definimos:*
   - $r^{(1)} = \mathrm{LT}(g^{(2)}) = 2x$
   - $g^{(3)} = g^{(2)} - r^{(1)} = y^2 + 2x - 2x = y^2 \neq 0$
4. $\exp(g^{(3)}) = (0,2) \in \Delta_1$, *definimos:*
   - $h^{(4)} = \frac{\mathrm{LT}(g^{(3)})}{\mathrm{LT}(f_1)} = \frac{y^2}{y^2} = 1$
   - $g^{(4)} = g^{(3)} - f_1 = y^2 - y^2 + 1 = 1 \neq 0$
5. $\exp(g^{(5)}) = (0,0) \in \overline{\Delta}$, *definimos:*
   - $r^{(2)} = \mathrm{LT}(g^4) = 1$
   - $g^{(5)} = g^{(4)} - r^{(2)} = 0$. *Fin del algoritmo.*

Por lo tanto:

$$\begin{aligned} g &= g^{(1)} + h^{(1)} f_2 \\ &= \left(g^{(2)} + h^{(2)} f_1\right) + x f_2 \\ &= \left(g^{(3)} + r^{(1)}\right) + x f_1 + x f_2 \\ &= \left(g^{(4)} + f_1\right) + x f_1 + x f_2 + r^{(1)} \\ &= \left(g^{(5)} + r^{(2)}\right) + (x+1) f_1 + x f_2 + r^{(1)} \\ &= \underbrace{(x+1)}_{h_1} f_1 + \underbrace{(x)}_{h_2} f_2 + \underbrace{(2x+1)}_{r = r^{(1)} + r^{(2)}} \end{aligned}$$

Sabemos que el algoritmo de la división euclídea en el anillo de polinomios de una variable, $\mathbb{K}[x]$, determina de forma efectiva la pertenencia de un polinomio a un ideal. En efecto, si I es un ideal de $\mathbb{K}[x]$, entonces existe $f_1 \in \mathbb{K}[x]$ tal que $I = \langle f_1 \rangle$ y por lo tanto $f \in I$ si y sólo si el resto de la división de f entre f_1 es cero.
Sin embargo, el algoritmo de división no tiene la misma propiedad en el caso de varias variables, debido a que el resto de la división en un anillo de polinomios con más de una variable no está determinado de forma única por dividendo y divisores.
Consideramos un ideal I generado por los polinomios $f_1, \ldots, f_s$. Es claro que, si tras dividir f entre $f_1, \ldots, f_s$ obtenemos resto cero, entonces

$$f = h_1 \cdot f_1 + \ldots, h_s \cdot f_s,$$

y por tanto $f \in \langle f_1, \ldots, f_s \rangle$.
Sin embargo, que el resto de la división de f entre $f_1, \ldots, f_s$ sea cero no es una condición necesaria para que $f \in I$. Esta propiedad se muestra en el siguiente ejemplo.

Ejemplo 2.2. *Consideramos los polinomios $f_1, f_2, g \in \mathbb{R}[x, y]$, siendo $f_1 = xy + 1$, $f_2 = y^2 - 1$ y $g = xy^2 - x$. Fijamos como orden monomial el orden lexicográfico y como orden en las variables $x > y$.*
Al efectuar la división de g entre $\{f_1, f_2\}$ se obtiene que $g = yf_1 - x - y$. Es decir, el resto de la división de g entre el ideal generado por f_1 y f_2 es distinto de cero. Sin embargo, $g = xf_2$ luego $g \in I = \langle (f_1, f_2 \rangle$.

2.1.5. Teorema de la base de Hilbert

Definición 2.8. *Sea $I \neq \{0\}$ un ideal de $\mathbb{K}[\mathbf{x}]$.*

1. *Se define el ideal inicial de I, que se denota por* $\mathrm{LT}(I)$*, como el conjunto de formas iniciales de los elementos de I.*

$$\mathrm{LT}(I) = \{c \cdot x^\alpha : \exists f \in I \text{ con } \mathrm{LT}(f) = c \cdot x^\alpha\}.$$

2. *Denotamos por* $\langle \mathrm{LT}(I) \rangle$ *al ideal generado por los elementos de* $\mathrm{LT}(I)$.

En el siguiente ejemplo veremos que si $\{f_1, \ldots, f_s\}$ es un sistema generador de un ideal I, esto no implica que $\{\mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_2)\}$ sea un sistema generador para $\mathrm{LT}(f)$.

Ejemplo 2.3. *Consideramos el ideal $I = \langle f_1, f_2 \rangle \subseteq \mathbb{R}[x, y]$, siendo $f_1 = x^3 - 2xy$ y $f_2 = x^2y - 2y^2 + x$. Fijamos como orden monomial el orden lexicográfico y como orden en las variables $x > y$.*
Observamos que $\mathrm{LT}(f_1) = x^3$ y $\mathrm{LT}(f_2) = x^2y$, sin embargo

$$\mathrm{LT}(I) \neq \langle \mathrm{LT}(f_1), \mathrm{LT}(f_2) \rangle.$$

En efecto:

$$xf_2 - yf_1 = x^3y - 2xy^2 + x^2 - x^3y + 2xy^2 = x^2 \in I.$$

Por lo tanto, $x^2 \in \mathrm{LT}(I)$ pero $x^2 \notin (x^3, x^2y)$.

Proposición 2.1. *Sea $I \subseteq \mathbb{K}[\mathbf{x}]$ un ideal.*

1. $\langle \mathrm{LT}(I) \rangle$ *es un ideal monomial.*

2. *Existen $g_1, \ldots, g_t \in I$ tales que $\langle \mathrm{LT}(I) \rangle = \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t) \rangle$.*

Demostración.

1. Hemos definido el ideal $\mathrm{LT}_\succ(I)$ como el conjunto de términos líderes de los elementos de I. Es decir:

$$\langle \mathrm{LT}_\succ(I) \rangle \;=\; \langle \{\mathrm{LT}_\succ(f) \mid f \in I \setminus \{0\}\} \rangle.$$

   Como $\mathrm{LM}_\succ(f)$ y $\mathrm{LT}_\succ(f)$ difieren en una constante no nula, entonces:

$$\langle \mathrm{LT}_\succ(I) \rangle \;=\; \langle \{\mathrm{LM}_\succ(f) \mid f \in I \setminus \{0\}\} \rangle.$$

   Luego $\langle \mathrm{LT}_\succ(I) \rangle$ es un ideal monomial.

2. Hemos visto que el ideal $\langle \mathrm{LT}_\succ(f)\rangle$ está generado por los monomios $\mathrm{LM}(f)$ tales que $f \in I \setminus \{0\}$. El Lema de Dickson (Teorema 2.1), nos garantiza que todo ideal monomial posee un sistema de generadores monomiales finitos, por lo tanto sabemos que existen $f_1, \ldots, f_s \in I$ tales que:

$$\langle \mathrm{LT}_\succ(I)\rangle \quad = \quad \langle \{\mathrm{LM}_\succ(f) \mid f \in I \setminus \{0\}\}\rangle.$$

Como $\mathrm{LM}_\succ(f)$ y $\mathrm{LT}_\succ(f)$ difieren en una constante no nula, se tiene el resultado.

□

Con la Proposición anterior y el algoritmo de la división podemos probar que todo ideal de $\mathbb{K}[\mathbf{x}]$ está finitamente generado.

Teorema 2.3 (Teorema de la Base de Hilbert)**.**
Todo ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ posee un sistema de generadores finito. Es decir,

$$I = \langle f_1, \ldots, f_s\rangle \text{ para ciertos } f_1, \ldots, f_s \in I.$$

Demostración. Véase [24]. Capítulo 2.5, Teorema 4. □

El Teorema 2.3 garantiza la existencia de un sistema de generadores del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$, $\{f_1, \ldots, f_s\}$, con la propiedad:

$$\langle \mathrm{LT}(I)\rangle = \langle \mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s)\rangle.$$

Como vimos en el Ejemplo 2.3, no todos los sistemas generadores de un ideal en $\mathbb{K}[\mathbf{x}]$ tienen esta propiedad, por ello reciben un nombre especial.

Definición 2.9 (Bases de Gröbner)**.**
Fijado un orden monomial. Un subconjunto finito $G = \{g_1, \ldots, g_t\}$ de un ideal I se dice que es una base de Gröbner si:

$$\langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t)\rangle = \langle \mathrm{LT}(I)\rangle.$$

Con esta definición se puede deducir que un subconjunto $\{f_1, \ldots, f_s\}$ de polinomios de un ideal I es una base de Gröbner de I respecto de $\succ$ si y sólo si el término líder de cualquier elemento $f \in I$ respecto de $\succ$ es divisible por $\mathrm{LT}_\succ(f_i)$ para algún $i \in \{1, \ldots, s\}$.

Corolario 2.2. *Fijado un orden monomial. Todo ideal $I \neq \{0\}$ de $\mathbb{K}[\mathbf{x}]$ tiene una base de Gröbner. Además toda base de Gröbner del ideal I es un sistema de generadores de I.*

Demostración. Fijamos un orden monomial $\succ$. Por la Proposición 2.1 sabemos que podemos encontrar polinomios $g_1, \ldots, g_s \in I$ tales que:

$$\langle \mathrm{LT}_\succ(I) \rangle = \langle \mathrm{LT}_\succ(g_1), \ldots, \mathrm{LT}_\succ(g_s) \rangle.$$

Por lo tanto el conjunto de polinomios $\{g_1, \ldots, g_s\}$ forman una base de Gröbner del ideal I. Nos faltaría probar que el conjunto $\{g_1, \ldots, g_s\}$ es un sistema de generadores de I. Es claro que $\langle g_1, \ldots, g_s \rangle \subseteq I$.
Recíprocamente, consideremos un polinomio $f \in I$, aplicando el algoritmo de la división de varias variables (Algoritmo 1) de f por $\{g_1, \ldots, g_s\}$ para el orden monomial $\succ$, sabemos que existen $h_1, \ldots, h_s, r \in \mathbb{K}[\mathbf{x}]$ tales que:

$$f \quad = \quad h_1 \cdot g_1 + \ldots + h_s \cdot g_s + r$$

Supongamos que $r \neq 0$, entonces $r = f - (h_1 \cdot g_1 + \ldots + h_s \cdot g_s) \in I \setminus \{0\}$, de donde se deduce que $\mathrm{LT}_\succ(r) \in \langle \mathrm{LT}_\succ(I) \rangle = \langle \mathrm{LT}_\succ(g_1), \ldots, \mathrm{LT}_\succ(g_s) \rangle$.
Es decir $\mathrm{LT}_\succ(r)$ divide a algún $\mathrm{LT}_\succ(g_i)$ con $i \in \{1, \ldots, s\}$, lo que contradice el Teorema 2.2.

□

2.1.6. Propiedades de las bases de Gröbner

Las bases de Gröbner tienen ciertas propiedades que nos permiten resolver los siguientes problemas:

1. Determinar si un polinomio f pertenece a un ideal.
2. En el caso de que un polinomio pertenezca al ideal $I = \langle f_1, \ldots, f_s \rangle$, determinar $h_1, \ldots, h_s \in \mathbb{K}[\mathbf{x}]$ tales que
$$f = h_1 f_1 + \ldots + h_s f_s.$$
3. Todo ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ define una relación de equivalencia sobre los polinomios de $\mathbb{K}[\mathbf{x}]$ dada por
$$f \equiv g \mod I \iff f - g \in I.$$
El problema que se nos plantea 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.

Nota 2.1. *En el caso del anillo $\mathbb{K}[x]$ de polinomios en una variable sabemos la respuesta a los problemas anteriores, pues $\mathbb{K}[x]$ es un dominio de ideales principales y por tanto se tiene que $I = \langle f_1, \ldots, f_s \rangle = \langle g \rangle$, donde g denota el máximo común divisor de los polinomios $f_1, \ldots, f_s$. Por lo tanto:*

1. *$f \in I$ si y sólo si el resto de dividir f por g es cero.*
2. *Aplicando el algoritmo de Euclides obtenemos*

$$f = \lambda g + r \quad con \quad \exp(r) \leq \exp(f).$$

Por lo tanto $r+I = f+I$ es un representante de la clase de equivalencia de f y $\{1, x, x^2, \ldots, x^{d-1}\}$ donde $d = \exp(g)$ es una base de R/I como $\mathbb{K}$-espacio vectorial.

La siguiente Proposición nos confirma que el resto que obtenemos por la división de un polinomio respecto a una base de Gröbner es único.

Proposición 2.2. *Fijado un orden monomial $\succ$. Sean $G = \{g_1, \ldots, g_t\}$ una base de Gröbner del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto de $\succ$ y $f \in \mathbb{K}[\mathbf{x}]$. Entonces existe un único $r \in \mathbb{K}[\mathbf{x}]$ que verifica las siguientes propiedades:*

1. *Ningún término de r es divisible por el conjunto $\{\mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t)\}$.*
2. *Existe $g \in I$ que verifica que $f = g + r$.*

En particular r es el resto de la división de f por cualquier elemento de G.

Demostración. Procedamos por reducción al absurdo suponiendo que existen dos polinomios $r,\ r' \in \mathbb{K}[\mathbf{x}]$ tales que:

$$f = g + r \ \text{ y } \ f = g' + r' \text{ con } g,\ g' \in I \Rightarrow r - r' = g - g' \in I.$$

Supongamos que $r - r' \neq 0$, entonces existe algún $g_i \in G$ con $i \in \{1, \ldots, t\}$ tal que $\mathrm{LT}_{\succ}(g_i)$ divide a $\mathrm{LT}_{\succ}(r - r')$. Esto es imposible ya que por hipótesis ningún $\mathrm{LT}_{\succ}(g_i)$ con $i \in \{1, \ldots, t\}$ divide a r o a r'. □

Definición 2.10. *Al resto de la división de f por G, donde G es una base de Gröbner del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto del orden monomial $\succ$, se denomina* forma normal de f módulo I respecto de $\succ$, *y lo denotaremos por*

$$\mathrm{nf}_{G_{\succ}}(f) = r.$$

A los monomios que pertenecen a $\mathbb{K}[\mathbf{x}] \setminus \langle \mathrm{LT}(I) \rangle$ se les denomina monomios estándar módulo I.

La Proposición 2.2 nos dice que todo polinomio $f \in \mathbb{K}[\mathbf{x}]$ módulo I se escribe de forma única como combinación lineal sobre $\mathbb{K}$ de monomios estándar. Es decir, los monomios estándar forman una base de $\mathbb{K}[\mathbf{x}]/I$ como $\mathbb{K}$-espacio vectorial. Este resultado da respuesta al Problema 3, planteado al principio de esta sección.

Lema 2.5. *Fijamos un orden monomial $\succ$. Sean G una base de Gröbner del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto de $\succ$ y $f \in \mathbb{K}[\mathbf{x}]$. Entonces:*

1. $\mathrm{nf}_{G_\succ}(f)$ *es una combinación lineal de los monomios*

$$\{\mathbf{x}^\alpha \mid \mathbf{x}^\alpha \notin \langle \mathrm{LT}_\succ(I)\rangle\}.$$

2. $\mathrm{nf}_{G_\succ}(f) = \mathrm{nf}_{G_\succ}(g) \Leftrightarrow f - g \in I$, *para todo* $f,\ g \in \mathbb{K}[\mathbf{x}]$.

3. *Los representantes del anillo cociente $\mathbb{K}[\mathbf{x}]/I$ son compatibles con la suma y el producto. Es decir:*

   a) $\mathrm{nf}_{G_\succ}(f) + \mathrm{nf}_{G_\succ}(g) = \mathrm{nf}_{G_\succ}(f+g)$.

   b) $\mathrm{nf}_{G_\succ}(f) \cdot \mathrm{nf}_{G_\succ}(g) = \mathrm{nf}_{G_\succ}(f \cdot g)$.

4. *Podemos dotar al anillo cociente $\mathbb{K}[\mathbf{x}]/I$ de una estructura de $\mathbb{K}$-espacio vectorial. Una base de este espacio vectorial está formada por los monomios estándar, es decir, los monomios del conjunto:*

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

Sin embargo, aunque el resto r sea único, los cocientes h_i producidos por el algoritmo de la división tales que $f = h_1 f_1 + \ldots + h_s f_s + r$ pueden cambiar si ordenamos los generadores de I con otro orden monomial.
El siguiente corolario nos aporta un criterio para determinar si un polinomio pertenece a un ideal.

Corolario 2.3. *Sean $G = \{g_1, \ldots, g_t\}$ una base de Gröbner del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ y $f \in \mathbb{K}[\mathbf{x}]$. Entonces $f \in I$ si y sólo si el resto de la división de f por G es cero.*

Demostración. Véase [24]. Capítulo 2.6, Corolario 2. □

Nota 2.2. *En general, el resultado no es cierto si se trata de un sistema de generadores cualquiera del ideal I, tal como vimos en el ejemplo 2.3.*

Definición 2.11. *Sean $f,\ g \in k[x_1, \ldots, x_n]$ dos polinomios no nulos.*

1. *Sea $\exp(f) = \alpha$ y $\exp(g) = \beta$, definimos x^γ el mínimo común múltiplo de $\mathrm{LT}(f)$ y $\mathrm{LT}(g)$. Donde $\gamma = (\gamma_1, \ldots, \gamma_n)$ con $\gamma_i = \mathrm{máx}(\alpha_i, \beta_i)$.*

2. *Definimos el polinomio S de f y g como el polinomio:*

$$S(f,g) = \frac{x^\gamma}{\mathrm{LT}(f)} \cdot f - \frac{x^\gamma}{\mathrm{LT}(g)} \cdot g$$

Los polinomios S de f y g están definidos para producir la cancelación de los términos líderes de f y g.

Lema 2.6. *Sean $f_1, \dots f_s \in \mathbb{K}[\mathbf{x}]$ tales que $\exp(f_i) = \alpha \neq 0$, para todo $i \in \{1, \dots, s\}$. Sea $f = \sum_{i=1}^{s} c_i \cdot f_i$ con $c_i \in \mathbb{K}$, $i \in \{1, \dots, s\}$. Si $\exp(f) > \alpha$, entonces f es una combinación lineal de polinomios $S(f_i, f_j)$ con $1 \leq i < j \leq s$.*

Demostración. Véase [24]. Capítulo 2.6, Lema 5. □

El siguiente Teorema nos aporta un criterio para determinar si un conjunto de polinomios constituye una base de Gröbner del ideal que generan.

Teorema 2.4 (Criterio de Buchberger)**.**
Sea $I \neq \{0\}$ un ideal de $\mathbb{K}[\mathbf{x}]$. Un sistema de generadores $\{g_1, \dots, g_t\}$ de I es una base de Gröbner de I si y sólo si para cualesquiera $i, j \in \{1, \dots, t\}$ se verifica que el resto de la división del polinomio $S(g_i, g_j)$ por G es cero.

Demostración. Véase [24]. Capítulo 2.6, Teorema 6. □

Este Teorema llevó a Bruno Buchberger en 1966 a crear un algoritmo que nos permite obtener una base de Gröbner de un ideal I de $\mathbb{K}[\mathbf{x}]$ a partir de un sistema de generadores de I en un número finito de etapas.

Definición 2.12. *Sea $\succ$ un orden monomial sobre $\mathbb{K}[\mathbf{x}]$. Una* base de Gröbner minimal *del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto de $\succ$ es una base de Gröbner G de I respecto de $\succ$ tal que:*

- $\mathrm{LC}(p) = 1$ *para todo $p \in G$*
- *Para todo $p \in G$,* $\mathrm{LT}(p) \notin \langle \mathrm{LT}(G - \{p\}) \rangle$

A partir de una base de Gröbner G se puede construir una base de Gröbner minimal utilizando el siguiente lema que elimina información redundante de G.

Lema 2.7. *Sean $\succ$ un orden monomial sobre $\mathbb{K}[\mathbf{x}]$ y G una base de Gröbner del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto de $\succ$. Si $p \in G$ es un elemento de G tal que $\mathrm{LT}_\succ(p) \in \langle \mathrm{LT}_\succ(G \setminus \{p\}) \rangle$, entonces $G \setminus \{p\}$ es también una base de Gröbner de I respecto de $\succ$.*

Algoritmo 2 Algoritmo de Buchberger

INPUT: $F = \{f_1, \ldots, f_s\} \subseteq \mathbb{K}[\mathbf{x}]$ con $f_i \neq 0$.
OUTPUT: Una base de Gröbner $G = \{g_1, \ldots, g_t\}$ del ideal I tal que $F \subset G$.

$G := F$
$\mathcal{G} := \{\{f_i, f_j\}, f_i \neq f_j \in G\}$
while $\mathcal{G} \neq \emptyset$ **do**
  Elegir $\{f, g\} \in \mathcal{G}$
  $\mathcal{G} := \mathcal{G} \setminus \{\{f, g\}\}$
  Sea h el resto de la división entre $S(f, g)$ y G.
  if $h \neq 0$ **then**
    $\mathcal{G} := \mathcal{G} \cap \{\{u, h\}, \forall u \in G\}$
    $G := G \cap \{h\}$
  end if
end while
Devolver G.

Algoritmo 3 Algoritmo de obtención de una base de Gröbner minimal

INPUT: $G = \{g_1, \ldots, g_s\}$ una base de Gröbner del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$
OUTPUT: $\mathcal{G}$ una base de Gröbner minimal del ideal I

$\mathcal{G} := G$
for $i = 1$ TO n **do**
  if $\mathrm{LC}(g_i) \in in(I)$ **then**
    $\mathrm{LC}(I) := \left(\mathrm{LC}(g_1), \ldots, \widehat{\mathrm{LC}(g_i)}, \ldots, \mathrm{LC}(g_t)\right)$
    $\mathcal{G} := \mathcal{G} \setminus \{g_i\}$
  end if
end for
Devolver $\mathcal{G}$.

Demostración. Véase [24]. Capítulo 2.7, Lema 3. □

Definición 2.13. *Sea $\succ$ un orden monomial sobre $\mathbb{K}[\mathbf{x}]$. Una* base de Gröbner reducida *del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto de $\succ$ es una base de Gröbner G de I respecto de $\succ$ tal que:*

- $\mathrm{LC}(p) = 1$ *para todo* $p \in G$.
- *Ningún término de p está en $\langle \mathrm{LT}(G - \{p\}) \rangle$ para cada $p \in G$.*

En otras palabras, si ningún monomio en la base de Gröbner G es redundante, diremos que se trata de una base de Gröbner minimal. Y si para cada dos elementos distintos de la base de Gröbner minimal G, $g,\ g' \in G$, ningún término de g' es divisible por $\mathrm{LT}_{\succ}(g)$, entonces diremos que G es reducida. Las bases de Gröbner reducidas tienen la siguiente propiedad.

Proposición 2.3. *Sea I un ideal no nulo en $\mathbb{K}[\mathbf{x}]$. Entonces para cada orden monomial sobre $\mathbb{K}[\mathbf{x}]$ existe una única base de Gröbner reducida de I.*

Demostración. Véase [24]. Capítulo 2.7, Proposición 6. □

Como consecuencia de la Proposición 2.3 podemos deducir un algoritmo para determinar si dos ideales son iguales. Supongamos que tenemos dos ideales $I,\ J \subseteq \mathbb{K}[\mathbf{x}]$ generados por los conjuntos de polinomios $\{f_1, \ldots, f_s\}$ y $\{f'_1, \ldots, f'_t\}$, fijamos un orden monomial $\succ$ y calculamos una base de Gröbner reducida de $\langle f_1, \ldots, f_r \rangle$ y $\langle f'_1, \ldots, f'_t \rangle$. Entonces los ideales son iguales si sus bases de Gröbner reducidas son iguales.

Algoritmo 4 Algoritmo de obtención de una base de Grôbner reducida

INPUT: $G = \{g_1, \ldots, g_s\}$ una base de Gröbner del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto del orden monomial $\succ$.
OUTPUT: $\mathcal{G}$ una base de Gröbner reducida del ideal I
  for $i = 1$ TO s **do**
    $g'_i := \frac{1}{\mathrm{LC}_{\succ}(g_i)} g_i$
  end for
  for $i = 1$ TO s **do**
    Sea g''_i el resto de la división entre g'_i y $\{g'_1, \ldots, \widehat{g'_i}, \ldots, g'_s\}$
    $G := G \cup \{g''_i\}$
  end for
  Devolver G.

Proposición 2.4. *Todo ideal I de $\mathbb{K}[\mathbf{x}]$ tiene un número finito de generadores con términos líderes distintos.*

Demostración. Véase [60]. Capítulo 1, Teorema 1.2. □

Definición 2.14 (Base de Gröbner Universal)**.**
Un subconjunto finito $\mathcal{U}$ de I es una base de Gröbner Universal, si $\mathcal{U}$ es una base de Gröbner de I respecto de todos los órdenes monomiales simultáneamente.

Corolario 2.4. *Todo ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ tiene una base de Gröbner universal finita.*

Demostración. Por la Proposición 2.4 sabemos que existe una cantidad finita de bases de Gröbner reducidas de I. Su unión es una base de Gröbner universal de I. □

2.1.7. Teorema de eliminación

Definición 2.15. *Dado un ideal I de $\mathbb{K}[\mathbf{x}]$, llamaremos r-ésimo* ideal de eliminación de I *al ideal de $\mathbb{K}[x_{r+1}, \ldots, x_n]$ definido por $I \cap \mathbb{K}[x_1, \ldots, x_n]$.*

Desde un punto de vista algebraico, el r-ésimo ideal de eliminación de I consiste en todos los polinomios $f \in I$ que dependen solamente de las últimas $n - r$ variables.
El siguiente Teorema permite determinar un algoritmo para calcular el r-ésimo ideal de eliminación de un ideal $I \subseteq \mathbb{K}[\mathbf{x}]$.

Teorema 2.5 (Teorema de Eliminación)**.**
Sean $I \subseteq \mathbb{K}[\mathbf{x}]$ un ideal y G una base de Gröbner de I respecto de un orden de eliminación. Entonces, el conjunto $G_r = G \cap \mathbb{K}[x_{r+1}, \ldots, x_n]$ es una base de Gröbner del ideal de eliminación r-ésimo para cada $r \in \{0, \ldots, n\}$.

Demostración. Véase [24]. Capítulo 3,1, Teorema 2. □

2.1.8. Complejidad del cálculo de una base de Gröbner

La dificultad que involucra el cálculo de una base de Gröbner ha sido objeto de numerosos estudios. Como una base de Gröbner puede ser utilizada para resolver un sistema de ecuaciones polinomiales, la complejidad de su cálculo es al menos la misma que la de este problema. Por otra parte, no es difícil codificar algunos problemas NP-completos (Knapsack, k-SAT ...) como sistemas de ecuaciones polinomiales, lo que demuestra que en general es un

problema duro y que la complejidad del peor caso no puede esperarse que sea buena incluso en característica 2, sin una cota en el número de variables (ver [3]). A pesar de que dicha complejidad es doblemente exponencial $O(2^{2^{\frac{n}{10}}})$, donde n es el número de variables, el comportamiento genérico es mucho mejor.
Por ejemplo, si el sistema algebraico tiene un número finito de soluciones, entonces su base de Gröbner puede ser calculada en tiempo polinomial en $d \times n$ operaciones con $d = \text{máx}_i\, d_i$, siendo d_i el exponente del término líder de cada una de las ecuaciones. En este caso para el orden degree-reverse-lexicographical donde el mayor grado de los elementos de una base de Gröbner está acotado por la cota de Macaulay [40, 31] :

$$\sum_{i=1}^{n}(d_i - 1) + 1.$$

Esta cota se puede comparar con el teorema de Bézout, que dice que el número de soluciones está acotado (cuando es finito) por $\prod_i d_i$ y alcanza con exactitud la cota en el caso homogéneo.
Otro problema que suele surgir a la hora del cálculo de bases de Gröbner es el crecimiento de los coeficientes en los cálculos intermedios; este crecimiento es tan grande que puede provocar fallos en los cálculos. Por ejemplo, si consideramos:

$$f_1 = 8x2y^2 + 5xy^3 + 3x^3z + x^2yz, \quad f_2 = x^5 + 2y^3z^2 + 13y^2z^3 + 5yz^4$$

$$f_3 = 8x^3 + 12y^3 + xz^2 + 3, \quad f_4 = 7x^2y^4 + 18xy^3z^2 + y^3z^3.$$

El ideal generado por los polinomios anteriores tiene como base de Gröbner para el orden *deglex* y con orden de las variables $x > y > z$

$$g_1 = x, \quad g_2 = y^3 + 1/4, \quad g_3 = z^2.$$

Sin embargo en el cálculo aparece el siguiente polinomio

$$y^3 - 173590650429045129076474718 2\ldots$$

Diversas alternativas modulares se han propuesto para solucionar este último problema, véase por ejemplo [5].

2.1.9. Algoritmo de conversión de bases de Gröbner

Dada una base de Gröbner G_1 de un ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto de un orden monomial $\succ_1$, el algoritmo que describen Faugère, Gianni, Lazard y Mora

en [27], que se conoce como algoritmo FGLM, permite obtener otra base G_2 del mismo ideal respecto de cualquier otro orden monomial $\succ_2$. La idea principal consiste en utilizar la estructura de espacio vectorial asociada al espacio cociente $\mathbb{K}[\mathbf{x}]/I$.
Ilustramos el algoritmo FGLM partiendo de una base de Gröbner arbitraria del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto de un orden monomial $\succ$ y convirtiéndola en una base de Gröbner G_{lex} de dicho ideal respecto de un orden lexicográfico. Utilizaremos dos listas de polinomios de $\mathbb{K}[\mathbf{x}]$:

- La nueva base que queremos obtener, G_{lex}, formada por polinomios del ideal I que se inicializa como una lista vacía.
- Una lista de monomios del anillo cociente $\mathbb{K}[\mathbf{x}]/I$ que denotamos por B_{lex} y que también se inicializa como una lista vacía.

En cada iteración consideramos el monomio más pequeño respecto del orden $\succ_2$ que no es divisible por ningún monomio $\mathrm{LT}_{\succ_2}(g_i)$ con $g_i \in G_{lex}$, denotamos a dicho monomio Ψ. Necesariamente el primer monomio que tenemos que considerar es $\Psi = 1$ ya que si fuese de la forma $\mathbf{x}^\alpha$, entonces se tendría que $\mathbf{x}^\alpha \prec 1$, lo que contradice la definición de orden monomial en $\mathbb{K}[\mathbf{x}]$.
El algoritmo consiste en 3 etapas:

1. **Etapa principal**: Calculamos la forma normal del monomio Ψ respecto de la base de Gröbner G_1.
   - Si existe una combinación lineal de la forma:
   $$\mathrm{nf}_{G_1}(\Psi) = \sum_j \lambda_j \mathrm{nf}_{G_1}(\mathbf{x}^{\alpha_j}) \ \text{ con } \ \mathbf{x}^{\alpha_j} \in B_{lex} \ \text{ y } \ \lambda_j \in \mathbb{K}.$$
   Entonces se tiene que $g = \Psi - \sum_j \lambda_j \mathbf{x}^{\alpha_j} \in I$ y añadimos g a la nueva base G_{lex}.
   - Si $\mathrm{nf}_{G_1}(\Psi)$ es linealmente independiente de las formas normales de los monomios de la lista G_{lex} respecto de la base de Gröbner G_1, entonces añadimos Ψ como último elemento de la lista B_{lex}.
2. **Averiguamos si hemos terminado:** Sea x_1 la mayor variable en el orden lexicográfico que estamos considerando. Si en la etapa anterior hemos añadido un nuevo elemento g a la lista G_{lex} entonces comprobamos si $\mathrm{LT}_{\succ_2}(g)$ es una potencia x_1. En dicho caso, el algoritmo termina.

3. **Calculamos el nuevo término:** Si el algoritmo no ha terminado entonces volvemos a la primera etapa considerando como nuevo término el menor monomio de $\mathbb{K}[\mathbf{x}]$ respecto del orden lexicográfico elegido que no sea divisible por ningún monomio $\mathrm{LT}_{\succ_2}(g_i)$ con $g_i \in G_{lex}$.

Véase [25], capítulo 2,3, Teorema 3,4 para una prueba de este algoritmo.

Ejemplo 2.4. *Consideramos el ideal*

$$I = \langle xy + z - xz, x^2 - z, 2x^3 - x^2yz - 1\rangle \subseteq \mathbb{Q}[x, y, z].$$

Para el orden lexicográfico graduado inverso $\succ_{grlex}$ y con orden en las variables $x > y > z$ obtenemos la siguiente base de Gröbner:

$$\begin{aligned} G \;=\; & \{f_1 = z^4 - 3z^3 - 4yz + 2z^2 - y + 2z - 2, f_2 = yz^2 + 2yz - 2z^2 + 1, \\ & f_3 = y^2 - 2yz + z^2 - z, f_4 = x + y - z\}. \end{aligned}$$

Utilizamos el algoritmo FGLM para encontrar una base de Gröbner del ideal I respecto del orden lexicográfico con $z > y > x$.

- *Inicializamos las listas G_{lex} y B_{lex} como listas vacías y consideramos como primer término de estudio $\Psi = 1$.*

  Observamos que $\mathrm{nf}_G(\Psi) = 1$ es linealmente independiente con las formas normales de los monomios de la lista B_{lex} respecto de la base de Gröbner G. Por lo tanto añadimos el término 1 a la lista B_{lex}.

- *Según el algoritmo, el siguiente término que tenemos que estudiar es $\Psi = x$.*

  Observamos que $\mathrm{nf}_G(\Psi) = -y+z$ de nuevo es linealmente independiente con las formas normales de los monomios de la lista B_{lex} respecto de la base de Gröbner G. Por lo que añadimos x a la lista B_{lex}.

- *Procedemos de forma análoga con los términos x^2, x^3, x^4, x^5 cuyas formas normales respecto de la base de Gröbner G son:*

  $$\begin{aligned} &\mathrm{nf}_G(x^2) = z, \quad \mathrm{nf}_G(x^2) = -yz, \\ &\mathrm{nf}_G(x^2) = z^2, \quad \mathrm{nf}_G(x^2) = z^3 + 2yz - 2z^2 + 1. \end{aligned}$$

  Estas formas normales son linealmente independientes con las formas normales de los monomios de la lista B_{lex} respecto de la base de Gröbner G.

Tras estas etapas la lista $B_{lex} = \{1, x, x^2, x^3, x^4, x^5\}$ *ha sido actualizada, mientras que la lista* G_{lex} *continúa vacía.*

- *Sin embargo, la forma normal del siguiente elemento* $\Psi = x^6$ *respecto de la base de Gröbner* G *verifica que:*

$$\mathrm{nf}_G(x^6) = \mathrm{nf}_G(x^5) + 2\mathrm{nf}_G(x^3) - \mathrm{nf}_G(1) = z^3.$$

  Por lo tanto añadimos el elemento $g_1 = x^6 - x^5 - 2x^3 + 1$ *a la nueva base de Gröbner,* G_{lex}*.*

- *El siguiente monomio que tenemos que estudiar es* $\Psi = y$*. Observamos que:*

$$\mathrm{nf}_G(y) = \mathrm{nf}_G(x^2) - \mathrm{nf}_G(x) = y.$$

  Por lo tanto $g_2 = y + x - x^2$ *forma un nuevo elemento de la lista* G_{lex}*.*

- *El siguiente monomio que tenemos que estudiar es* $\Psi = z$*. Observamos que:*

$$\mathrm{nf}_G(z) = \mathrm{nf}_G(x^2) = z.$$

  Por lo tanto $g_3 = z - x^2$ *forma un nuevo elemento de la lista* G_{lex}*.*

- *Observamos que el término líder respecto del orden lexicográfico definido del elemento que hemos añadido en la etapa anterior a la nueva base de Gröbner es múltiplo de la variable* z*, que es la mayor variable de dicho orden. Por lo tanto el algoritmo termina.*

Tras finalizar el algoritmo hemos obtenido una base de Gröbner del ideal I *respecto del orden lexicográfico definido. La nueva base es:*

$$G_{lex} = \{x^6 - x^5 - 2x^3 + 1, y + x - x^2, z - x^2\}.$$

De forma general, dada una base de Gröbner G_1 del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto del orden monomial $\succ_1$, el algoritmo FGLM (Algoritmo 5) crea una nueva base de Gröbner G_2 del mismo ideal respecto de otro orden monomial $\succ_2$. Para formular el algoritmo FGLM (Algoritmo 5) de forma general utilizamos las siguientes listas y funciones:

- La lista `Newbasis` que contiene los polinomios que forman la nueva base de Gröbner G_2.

- La lista `NexTerms` cuyo primer elemento es el que tenemos que estudiar en cada iteración.

- La lista `LeadTerms` que contiene los términos líderes de la lista `Newbasis` ordenados de forma creciente por el orden $\succ_2$.
- La lista `RedTerms` que contiene las formas normales de los términos de la lista `Newbasis`.
- La función $\mathrm{nf}_{G_1}(\mathbf{t})$ que nos devuelve la forma normal del elemento $\mathbf{t} \in \mathbb{K}[\mathbf{x}]$ respecto de la base de Gröbner G_1.
- La función `Insert`$(S, \mathbf{t})$ que añade a la lista S los términos $\{x_i \cdot \mathbf{t}\}_{i=1}^{n}$.
- La función `Order`(S) que ordena los elementos de la lista S respecto del orden $\succ_2$.
- La función `Next`(S) que nos devuelve el primer elemento de la lista S y lo borra de dicha lista.

En primer lugar inicializamos la lista `NextTerms` con el término más pequeño respecto del orden $\succ_2$, que necesariamente es 1, y el resto de listas como listas vacías.
En cada iteración escogemos el elemento $\mathbf{t} :=$ `Next`(`NextTerms`) y calculamos su forma normal respecto de la base de Gröbner inicial G_1. En el caso de que la forma normal obtenida se pueda escribir como combinación lineal de las formas normales de los elementos de `RedTerms`, entonces añadimos un nuevo elemento a la base de Gröbner G_2; en caso contrario, se añade $\mathbf{t}$ a la lista `RedTerms`. El nuevo término que escogemos es el término más pequeño respecto del orden $\succ_2$ del siguiente conjunto:

$$\{\mathbf{t} \in \mathbb{T}_1 \mid \mathbf{t} \text{ no es divisible por } \mathrm{LT}_{\succ_2}(g_i) \text{ con } g_i \in \texttt{NewBasis}\}.$$

Donde $\mathbb{T}_1$ denota el conjunto de monomios de $\mathbb{K}[\mathbf{x}]$.

2.2. Cálculo del núcleo de un homomorfismo

Consideramos un cuerpo arbitrario $\mathbb{K}$, denotaremos por $\mathbb{K}[\mathbf{x}] := \mathbb{K}[x_1, \ldots, x_n]$ y por $\mathbb{K}[\mathbf{y}] := \mathbb{K}[y_1, \ldots, y_m]$ al anillo, en n y m indeterminadas respectivamente, sobre el cuerpo $\mathbb{K}$.
Vamos a estudiar en esta sección distintos algoritmos que nos permiten calcular el núcleo del siguiente homomorfismo de anillos:

$$\begin{array}{rccc} \Theta: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{K}[\mathbf{y}] \\ & x_i & \longmapsto & \Theta(x_i) = f_i \end{array}$$

Algoritmo 5 Algoritmo FGLM

INPUT: Una base de Gröbner G_1 del ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ respecto del orden $\succ_1$ y otro orden monomial $\succ_2$ de $\mathbb{K}[\mathbf{x}]$.

OUTPUT: Una Base de Gröbner reducida G_2 de I respecto del orden $\succ_2$.

Inicializamos Ψ con el término más pequeño en el conjunto de monomios $\mathbb{T}_1$ de $\mathbb{K}[\mathbf{x}]$ respecto del orden $\succ_2$.

`RedTerms` $:= [\Psi]$;
`NextTerms` $:= [\Psi]$;
`NewBasis` $:= []$;
`LeadTerms` $:= []$;
`Insert`(`NextTerms`, Ψ);
`Order`(`NextTerms`);
while `NextTerms` $\neq []$ **do**
  $\Psi :=$ `Next`(`NextTerms`);
  if Ψ no es un múltiplo de un elemento de la lista `LeadTerms` **then**
    if existe una combinación lineal de la forma:

$$\mathrm{nf}_{G_1}(\Psi) = \sum_{\mathbf{s} \in \texttt{RedTerms}} \alpha_{\mathbf{s}} \cdot \mathrm{nf}_{G_1}(\mathbf{s}) \text{ con } \alpha_{\mathbf{s}} \in \mathbb{K}$$

    then
      `NewBasis` $:=$ `NewBasis` $\cup \{\Psi - \sum_{\mathbf{s} \in \texttt{RedTerms}} \alpha_{\mathbf{s}} \cdot \mathbf{s}\}$;
      `LeadTerms` $:=$ `LeadTerms` $\cup \{\Psi\}$;
    else
      `RedTerms` $:=$ `RedTerms` $\cup \{\Psi\}$;
      `Insert`(`NextTerms`, Ψ);
      `Order`(`NextTerms`);
    end if
  end if
end while

2.2.1. Algoritmo clásico

Lema 2.8. *Sea G una base de Gröbner reducida del ideal*

$$\langle \{x_1 - \Theta(x_1), \ldots, x_n - \Theta(x_n)\} \rangle \subseteq \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$$

respecto a un orden monomial que elimine las variables y.
Entonces $G' = G \cap \mathbb{K}[\mathbf{x}]$ es una base de Gröbner reducida de $\ker(\Theta)$.

Demostración. La demostración se deduce del Teorema de eliminación (Teorema 2.5). □

El lema anterior nos indica que el problema que nos habíamos planteado se puede abordar aplicando el algoritmo de Buchberger sobre el anillo de polinomios $\mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ (Algoritmo 2) y luego utilizar la teoría de eliminación para obtener una base de Gröbner en $\mathbb{K}[\mathbf{x}]$.
La complejidad del algoritmo de Buchberger aumenta considerablemente al incrementar el número de variables. Nótese que estaríamos trabajando en un anillo con $n + m$ variables, luego lo óptimo sería encontrar un algoritmo que trabaje directamente en $\mathbb{K}[\mathbf{x}]$.

2.2.2. Método de Di Biase-Urbanke

En esta sección presentamos el método de Fausto Di Biase y Rüdiger Urbanke basado en calcular bases de Gröbner con un número menor de variables, lo que en general acelera el proceso computacional. Véase [26].
A continuación, fijamos la notación que utilizaremos a lo largo de esta sección. Definimos $M_{\mathbf{x}}$ como el conjunto de monomios en el anillo $\mathbb{K}[\mathbf{x}]$. Análogamente, definimos $M_{\mathbf{y}}$ como el conjunto de monomios en el anillo $\mathbb{K}[\mathbf{y}]$.
Nuestro objetivo es calcular el núcleo del homomorfismo de anillos Θ definido por:

$$\begin{array}{rccc} \Theta: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{K}[\mathbf{y}] \\ & x_i & \longmapsto & \Theta(x_i) = f_i \in M_{\mathbf{y}} \end{array}$$

El homomorfismo Θ viene definido por la matriz

$$M = (m_{ij})_{\substack{i \in \{1,\ldots,m\} \\ j \in \{1,\ldots,n\}}} \in \mathbb{Z}^{m \times n}$$

donde la fila i-ésima se corresponde con el exponente del monomio $f_i \in M_{\mathbf{y}}$, por lo tanto podemos reescribir la aplicación Θ como:

$$\begin{array}{rccc} \Theta: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{K}[\mathbf{y}] \\ & \mathbf{x}^{\mathbf{u}} & \longmapsto & \Theta(\mathbf{x}^{\mathbf{u}}) = \mathbf{y}^{M\mathbf{u}^t} \in M_{\mathbf{y}} \end{array}$$

A partir del homomorfismo anterior podemos definir la aplicación $\mathbb{Z}$-lineal Π:

$$\begin{array}{rccc} \Pi: & \mathbb{Z}^n & \longrightarrow & \mathbb{Z}^m \\ & \mathbf{u} & \longmapsto & \Pi(u) = M \cdot \mathbf{u}^t \end{array}$$

Utilizando la *forma normal de Hermite* es fácil obtener una base de $\ker \Pi$.
Definimos soporte de un vector $\mathbf{u} \in \mathbb{Z}^n$ como el conjunto de coordenadas distintas de cero, es decir, $\text{supp}(\mathbf{u}) = \{i \mid u_i \neq 0\}$.
Sabemos que todo vector $\mathbf{u} \in \mathbb{Z}^n$ se puede escribir de forma única como $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^-$ donde $\mathbf{u}^+,\ \mathbf{u}^- \in \mathbb{N}^n$ y tienen soportes disjuntos.
El siguiente Lema nos permite establecer una relación entre los vectores $\mathbf{u} \in \mathbb{Z}^n$ que pertenecen al $\mathbb{Z}$-núcleo de la aplicación Π y los monomios que pertenecen al núcleo de la aplicación Θ.

Lema 2.9. $\mathbf{u} \in \ker \Pi$ *si y sólo si* $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in \ker \Theta$.

Demostración. Consideramos un vector $\mathbf{u} \in \mathbb{Z}^n$ perteneciente al núcleo de la aplicación lineal Π. Por definición tenemos que:

$$0 = \Pi(\mathbf{u}) = \Pi(\mathbf{u}^+ - \mathbf{u}^-) = \Pi(\mathbf{u}^+) - \Pi(\mathbf{u}^-) = M \cdot (\mathbf{u}^+)^t - M \cdot (\mathbf{u}^-)^t.$$

Por lo tanto:

$$0 = \mathbf{y}^{M(\mathbf{u}^+)^t} - \mathbf{y}^{M(\mathbf{u}^-)^t} = \Theta(\mathbf{x}^{\mathbf{u}^+}) - \Theta(\mathbf{x}^{\mathbf{u}^-}) = \Theta(\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-})$$

De donde se deduce que el binomio asociado al vector $\mathbf{u} \in \mathbb{Z}^n$, $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$, pertenece al núcleo del homomorfismo Θ. □

La relación entre $\ker \Pi$ y $\ker \Theta$ viene dada por la siguiente aplicación que asocia a cada vector $\mathbf{u} \in \mathbb{Z}^n$ un binomio en $\mathbb{K}[\mathbf{x}]$.

$$\begin{array}{rccc} \varphi: & \mathbb{Z}^n & \longrightarrow & \mathbb{K}[\mathbf{x}] \\ & \mathbf{u} & \longmapsto & \varphi(\mathbf{u}) = \mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \end{array}$$

Teorema 2.6. *El núcleo del homomorfismo* Θ *está generado como* $\mathbb{K}$*-espacio vectorial por el conjunto de binomios*

$$\{\mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} \mid \mathbf{u},\ \mathbf{v} \in \mathbb{N}^n \quad y \quad \Pi(\mathbf{u}) = \Pi(\mathbf{v})\}.$$

Es decir,

$$\ker \Theta = \langle \mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \mid \mathbf{u} \in \ker \Pi \quad y \quad \mathbf{u},\ \mathbf{v} \in \mathbb{N}^n \rangle.$$

Demostración. En el Lema 2.9 vimos que

$$\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in \ker\Theta \Leftrightarrow \mathbf{u} \in \ker\Pi \text{ y } \mathbf{u},\ \mathbf{v} \in \mathbb{N}^n.$$

Faltaría probar que todo polinomio $f \in \ker\Theta$ es combinación lineal de binomios de la forma $\{\mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} \mid \mathbf{u},\ \mathbf{v} \in \mathbb{N}^n \text{ y } \Pi(\mathbf{u}) = \Pi(\mathbf{v})\}$.
Procedamos por reducción al absurdo suponiendo que $\exists f \in \ker\Theta$ tal que f no es combinación lineal de binomios con la forma requerida.
Elegimos f entre todos los que verifican la propiedad anterior de manera que $\mathrm{LT}_{\succ}(f) = \mathbf{x}^{\mathbf{u}}$ es mínimo respecto del orden $\succ$.
Como $f \in \ker\Theta$, entonces $\Theta(f(x_1, \ldots, x_n)) = 0$. En particular:

$$0 = \Theta(\mathrm{LT}_{\succ}(f)) = \Theta(\mathbf{x}^{\mathbf{u}}) = \mathbf{y}^{\Pi(\mathbf{u})},$$

es decir, debe existir un monomio $\mathbf{x}^{\mathbf{v}} \succ \mathbf{x}^{\mathbf{u}}$ en f tal que $\Pi(\mathbf{v}) = \Pi(\mathbf{u})$.
Por hipótesis $f' = f - \mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}}$ tampoco debe poderse escribir como combinación lineal de binomios con la forma requerida, sin embargo:

$$\mathrm{LT}_{\succ}(f) \succ \mathrm{LT}_{\succ}(f'),$$

lo que contradice la hipótesis de minimalidad realizada. □

Nota 2.3. *No toda base de* $\ker\Pi$ *genera a* $\ker\Theta$ *como* $\mathbb{K}$*-espacio vectorial.*

Ejemplo 2.5. *En efecto, consideramos el homomorfismo*

$$\begin{array}{rccl} \Theta: & \mathbb{K}[x_1,x_2,x_3,x_4] & \longrightarrow & \mathbb{K}[y_1,y_2] \\ & x_1 & \longmapsto & \Theta(x_1) = y_1^3 \\ & x_2 & \longmapsto & \Theta(x_2) = y_1^2 y_2 \\ & x_3 & \longmapsto & \Theta(x_3) = y_1 y_2^2 \\ & x_4 & \longmapsto & \Theta(x_1) = y_2^3 \end{array}$$

Por el Lema 2.8 sabemos que si G es una base reducida del ideal

$$I = \langle x_1 - \Theta(x_1), \ldots, x_4 - \Theta(x_4)\rangle,$$

entonces $G' = G \cap \mathbb{K}[\mathbf{x}]$ es una base reducida de $\ker\Theta$.
Fijamos como orden monomial el orden lexicográfico y como orden en las variables: $y_1 > y_2 > x_1 > x_2 > x_3 > x_4$.
Aplicando el algoritmo de Buchberger (Algoritmo 2) sobre el ideal I obtenemos:

$$\begin{aligned} G = \{&-x_3^2 + x_2x_4, -x_2x_3 + x_1x_4, -x_2^2 + x_1x_3, x_4 - y_2^3, -x_4y_1 + x_3y_2, \\ &-x_3y_1 + x_2y_2, -x_2y_1 + x_1y_2, x_3 - y_1y_2^2, x_2 - y_1^2y_2, x_1 - y_1^3\}. \end{aligned}$$

Por lo tanto $G' = G \cap \mathbb{K}[\mathbf{x}]$ *es un ideal de* $\ker \Theta$ *respecto del orden lexicográfico dado por el orden en las variables* $x_1 > x_2 > x_3 > x_4$.

$$G' = \{-x_3^2 + x_2x_4, -x_2x_3 + x_1x_4, -x_2^2 + x_1x_3\} \tag{2.1}$$

Por otra parte, la matriz asociada a la aplicación Θ *es:*

$$M = \begin{pmatrix} 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix}.$$

Observamos que una base de $\ker \Pi$ *es:*

$$S = \begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & 2 & -1 & 0 \end{pmatrix}.$$

Aplicando el algoritmo de Buchberger (Algoritmo 2) a $\varphi(S) = \{x_1x_4 - x_2x_3, x_2^2 - x_1x_3\}$ *obtenemos una base de Gröbner del ideal* $\varphi(\langle \ker \Pi \rangle)$ *respecto del orden lexicográfico dado por el orden en las variables* $x_1 > x_2 > x_3 > x_4$.

$$G = \{x_2x_3^2 - x_2^2x_4, -x_2x_3 + x_1x_4, x_2^2 - x_1x_3\}. \tag{2.2}$$

Observamos que las bases 2.1 y 2.2 son distintas, luego:

$$\langle \varphi(\ker \Pi) \rangle \subsetneq \langle \ker \Theta \rangle.$$

La clave del algoritmo está en el siguiente resultado.

Teorema 2.7. *Si* $G \in \mathbb{N}^{k \times n}$ *es una base de* $\ker \Pi$ *entonces*

$$\ker \Theta = \langle \varphi(\text{Span } G) \rangle.$$

Demostración. Este resultado se deduce como consecuencia del Teorema 2.6. □

Definición 2.16. *Diremos que dos matrices* M, $M' \in \mathbb{Z}^{n \times m}$ *están relacionadas* ($M \sim M'$) *si existe una matriz* $A \in \mathbb{Z}^{n \times n}$ *tal que* $M' = AM$.
En este caso, el espacio vectorial que generan M *y* M' *coincide, i.e.*

$$\text{Span}(M) = \text{Span}(M').$$

En general, dada una matriz $M \in \mathbb{Z}^{m \times n}$ no siempre existe una matriz $M' \in \mathbb{N}^{m \times n}$ tal que $M \sim M'$. Sin embargo, tal y como expone el siguiente lema, siempre podemos encontrar una matriz $M' \in \mathbb{Z}^{m \times n}$ tal que cada columna de M' está en $\mathbb{N}^m$ o en $(-\mathbb{N})^m$.

Lema 2.10. *Sea $M \in \mathbb{Z}^{m\times n}$, existe $M' \in \mathbb{Z}^{m\times n}$ tal que $M \sim M'$ y cada columna de M' está en $\mathbb{N}^m$ o en $(-\mathbb{N}^m)$.*

Demostración. Consideramos $M \in \mathbb{Z}^{m\times n}$ y denotamos por $M_i \in \mathbb{Z}^m$, $\forall i = \{1,\dots n\}$ a las columnas de M.
Siguiendo la notación definimos $M' \in \mathbb{Z}^{m\times n}$ tal que:

$$\begin{cases} M'_1 = M_1 \\ M'_i = M_i + (1 + \text{máx}\,|m_{ij} \mid j \in \{1,\dots,n\}|) \cdot M'_{i-1} \end{cases}$$

Es fácil comprobar que $M \sim M'$, pues $\exists A \in \mathbb{Z}^{n\times n} : M' = AM$ con

$$A = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ b_2 & 1 & 0 & \dots & 0 \\ b_3b_2 & b_3 & 1 & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ b_k \cdots b_2 & b_k \cdots b_3 & b_k \cdots b_4 & \dots & 1 \end{pmatrix}$$

donde $b_i = 1 + \text{máx}\,|m_{ij} \mid j \in \{1,\dots,n\}|$, $\forall i \in \{2,\dots,k-1\}$. □

El algoritmo se basa en dos ideas principales:

1. Realizar una serie de cambios de signo en la matriz M', de manera que del último cambio de signo obtenemos una matriz que verifica el Teorema 2.7. Los cambios de signo se llevarán a cabo mediante la siguiente aplicación:

$$\begin{array}{rccc} T_j: & \mathbb{Z}^n & \longrightarrow & \mathbb{Z}^n \\ & \mathbf{u} = (u_1,\dots,u_n) & \longmapsto & T_j(\mathbf{u}) = (u_1,\dots,-u_j,\dots,u_n) \end{array}$$

2. Deshacer los cambios realizados en los ideales asociados, a través de bases de Gröbner, para obtener un sistema de generadores de $\ker\Theta$.

El siguiente teorema nos permite deshacer los cambios de signo en los ideales asociados. Los cambios de signo en los ideales asociados se realizarán utilizando la siguiente aplicación:

$$\begin{array}{rccc} T_j^*: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{K}[\mathbf{x}] \\ & p = \varphi(\mathbf{u}) = \mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} & \longmapsto & T_j^*(p) = \varphi(T_j(\mathbf{u})) \end{array}$$

Sea $A \in \mathbb{Z}^{n\times m}$, denotaremos A_i a la matriz formada por las columnas de A en la que reemplazamos la columna a_i por la columna $-a_i$. Utilizaremos

también la notación $\mathbf{x}$ para los monomios que no contienen a la variable x_j. Siguiendo esta notación, se tiene que:

$$x_i^r \mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} \in \ker(A_i) \iff \mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} x_i^r \in \ker(A). \tag{2.3}$$

Fijamos como orden monomial $\succ$ cualquier orden de eliminación en $\mathbb{K}[\mathbf{x}]$ para la variable x_i.

Teorema 2.8. *Sea* $G_i = \{x_i^{r_j}\mathbf{x}^{\mathbf{u}_j} - \mathbf{x}^{\mathbf{v}_j} \mid j \in \{1,\dots,m\}\}$ *una base de Gröbner de* $\ker(A_i)$ *respecto del orden* $\succ$*, entonces*

$$G = \{\mathbf{x}^{\mathbf{u}_j} - \mathbf{x}^{\mathbf{v}_j} x_i^{r_j}, \forall j \in \{1,\dots,m\}\}$$

es un sistema de generadores de $\ker(A)$*.*

Demostración. Teniendo en cuenta la forma de G_i, es fácil comprobar que $\ker(A)$ está generado por binomios de la forma $f = \mathbf{x}^{\mathbf{u}} - x_i^r \mathbf{x}^{\mathbf{v}}$. Por definición de base de Gröbner se tiene que $f \equiv 0 \mod G$, es decir, existen $f_1,\dots,f_m \in \mathbb{K}[\mathbf{x}]$ tales que:

$$x_i^r \mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} = \sum_{j=1}^{m} f_j(x_1,\dots,x_m)(x_i^{r_j}\mathbf{x}^{\mathbf{u}_j} - \mathbf{x}^{\mathbf{v}_j}). \tag{2.4}$$

Por la elección del orden monomial $\succ$ sabemos que la variable x_i aparece con grado a lo sumo $r - r_j$ en f_j, en particular $f_j = 0$ si $r < r_j$.
Reemplazamos x_i por $\frac{1}{x_i}$ en la ecuación 2.4 y obtenemos:

$$\left(\frac{1}{x_i}\right)^r \mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} = \sum_{j=1}^{m} f_j(x_1,\dots,x_{i-1},\frac{1}{x_i},x_{i+1},\dots,x_m)\left(\left(\frac{1}{x_i}\right)^{r_j}\mathbf{x}^{\mathbf{u}_j} - \mathbf{x}^{\mathbf{v}_j}\right)$$

Multiplicamos ambos lados de la igualdad por x_i^r en la ecuación anterior:

$$\mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} x_i^r = \sum_{j=1}^{m} f_j(x_1,\dots,x_{i-1},\frac{1}{x_i},x_{i+1},\dots,x_m) x_i^{r-r_j}\left(\mathbf{x}^{\mathbf{u}_j} - \mathbf{x}^{\mathbf{v}_j}\right).$$

Luego hemos visto que $\mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} x_i^r$ es combinación lineal de polinomios de G. □

Algoritmo 6 Método de Di Biase - Urbanke

INPUT: Un homomorfismo de anillos Θ.
OUTPUT: Un sistema de generadores de $\ker \Theta$.

❶ Calcular una base $M \in \mathbb{Z}^{m\times n}$ de $\ker \Pi$.
❷ Encontrar $M' \in \mathbb{Z}^{m\times n}$ tal que $M' \sim M$ y todas las columnas de M' están en $\mathbb{N}^m$ o en $(-\mathbb{N})^m$.
❸ Sea $J \subseteq \{1,\dots,n\}$ los índices de las columnas de M' con elementos en $(-\mathbb{N})^m$. Definimos M'_J la matriz obtenida a partir de la matriz M' cambiando el signo de las columnas indexadas por $j \in J$.
❹ Por el Teorema 2.7 sabemos que $\ker \Theta = \langle \varphi(\text{Span } M'_J)\rangle$.
while $J \neq \emptyset$ **do**
  Elegir $j \in J$.
  Definir una base de Gröbner G_J del ideal $\langle \varphi(M'_J)\rangle$ respecto a cualquier orden de eliminación para la variable x_j.
  Definir $G_{J\setminus\{j\}} = T_j(G_J)$, donde $G_{J\setminus\{j\}}$ es el resultado de llevar a cabo la aplicación T_j^* en todos los elementos de G_J.
  $J := J \setminus \{j\}$.
end while
Devolver $G_\emptyset$ es un sistema de generadores de $\ker \pi$.

2.3. Bases de Graver

2.3.1. Introducción a las Bases de Graver

A lo largo de esta sección continuaremos con la notación establecida en los apartados anteriores, es decir consideramos:

- Sea $\mathbb{K}$ un cuerpo arbitrario, denotaremos por $\mathbb{K}[\mathbf{x}] := \mathbb{K}[x_1, \ldots, x_n]$ y por $\mathbb{K}[\mathbf{y}] := \mathbb{K}[y_1, \ldots, y_m]$ al anillo, en n y m indeterminadas respectivamente, sobre el cuerpo $\mathbb{K}$.
- Una matriz $M = (m_{ij})_{\substack{i \in \{1,\ldots,m\} \\ j \in \{1,\ldots,n\}}} \in \mathbb{Z}^{m \times n}$
- La aplicación Π $\mathbb{Z}$-lineal definida por:

$$\begin{array}{rccc} \Pi: & \mathbb{Z}^n & \longrightarrow & \mathbb{Z}^m \\ & \mathbf{u} & \longmapsto & \Pi(\mathbf{u}) = M \cdot \mathbf{u}^t \end{array}$$

- El homomorfismo de anillos Θ definido por:

$$\begin{array}{rccc} \Theta: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{K}[\mathbf{y}] \\ & \mathbf{x}^{\mathbf{u}} & \longmapsto & \Theta(x_i) = \mathbf{y}^{M\mathbf{u}^t} \end{array}$$

El núcleo del homomorfismo Θ forma un ideal tórico asociado a la matriz M, que denotaremos por el ideal $I_M = \ker \Theta$.
Como ya vimos en el Teorema 2.6, el ideal tórico I_M está generado como $\mathbb{K}$-espacio vectorial por el conjunto de binomios

$$\begin{aligned} I_M &= \langle \{\mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} \mid \mathbf{u},\ \mathbf{v} \in \mathbb{N}^n \text{ y } \Pi(\mathbf{u}) = \Pi(\mathbf{v})\} \rangle \\ &= \langle \{\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \mid \mathbf{u} \in \ker(\Pi)\} \rangle \end{aligned}$$

Corolario 2.5. *Sea $\succ$ un orden monomial sobre $\mathbb{K}[\mathbf{x}]$. Existe un conjunto finito $G_\succ \subseteq \ker(\Pi)$, tal que $\{\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \mid \mathbf{u} \in G_\succ\}$ es una base de Gröbner reducida del ideal I_M respecto de $\succ$.*

Demostración. Por el Teorema de la Base de Hilbert (Teorema 2.3) sabemos que existe un subconjunto finito de vectores de $\ker \Pi$ tal que sus correspondientes binomios generan I_M.
Aplicamos el algoritmo de Buchberger (Algoritmo 2) a este conjunto finito de binomios, sabiendo que las operaciones que se realizan en el algoritmo de

Buchberger conservan la estructura de los binomios. Es decir, la salida de este algoritmo será un subconjunto contenido en el conjunto de binomios

$$\{\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \mid \mathbf{u} \in \ker(\Pi)\}.$$

□

Denotaremos por $\mathcal{U}_M$ a la base de Gröbner Universal del ideal I_M, es decir, siguiendo la definición 2.14, $\mathcal{U}_M$ es la unión de todas las bases de Gröbner reducidas del ideal I_M.
Como consecuencia del Corolario 2.5 podemos identificar $\mathcal{U}_M$ con un conjunto finito de vectores que generan $\ker \Pi$.

Definición 2.17. *Un binomio* $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in I_M$ *se dice que es un* binomio primitivo *en el ideal* I_M *si no existe otro binomio* $\mathbf{x}^{\mathbf{v}^+} - \mathbf{x}^{\mathbf{v}^-} \in I_M$ *tal que* $\mathbf{x}^{\mathbf{v}^+}$ *divida a* $\mathbf{x}^{\mathbf{u}^+}$ *ni* $\mathbf{x}^{\mathbf{v}^-}$ *divida a* $\mathbf{x}^{\mathbf{u}^-}$.

Definimos a continuación el orden $\sqsubset$ en $\mathbb{Z}^n$ que utilizaremos a lo largo de esta sección.

Definición 2.18. *Sean* $\mathbf{u},\ \mathbf{v} \in \mathbb{Z}^n$, *diremos que* $\mathbf{u} \sqsubset \mathbf{v}$ *si* $|u_i| \leq |v_i|$ *y* $u_i \cdot v_i \geq 0$ *para todo* $i = \{1, 2, \ldots, n\}$.

Observemos que un binomio $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$ es primitivo en el ideal I_M si y sólo si se tiene que $\mathbf{u} = \mathbf{u}^+ + \mathbf{u}^- \in \mathbb{Z}^n$ es minimal respecto al orden $\sqsubset$.

Lema 2.11. *El binomio* $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in \mathbb{K}[\mathbf{x}]$ *es primitivo en el ideal* I_M *si y sólo si el vector* $\mathbf{u} = \mathbf{u}^+ + \mathbf{u}^- \in \mathbb{Z}^n$ *es minimal respecto al orden* $\sqsubset$.

Demostración. Supongamos que el binomio $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$ es primitivo en el ideal I_M. Procedamos por reducción al absurdo suponiendo que el vector $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^-$ no es minimal respecto del orden $\sqsubset$.
Es decir, existe un vector $\mathbf{v} = \mathbf{v}^+ - \mathbf{v}^- \in \mathbb{Z}^n$ tal que $\mathbf{v} \sqsubset \mathbf{u}$. Por definición tenemos que:

1. $v_i \cdot u_i \leq 0\ \forall i \in \{1, \ldots, n\} \ \Rightarrow \ \begin{cases} \text{Si } v_i \in \mathbf{v}^+ & \Rightarrow \ u_i \in \mathbf{u}^+ \\ \text{Si } v_i \in \mathbf{v}^- & \Rightarrow \ u_i \in \mathbf{u}^- \end{cases}$

2. $|v_i| \leq |u_i| \Rightarrow |\mathbf{v}^+| \leq |\mathbf{u}^+| \Rightarrow \mathbf{v}^+ \leq \mathbf{u}^+$

$$\Rightarrow \exists q, r \in \mathbb{Z}^r \mid \mathbf{u}^+ = q\mathbf{v}^+ + r \Rightarrow \mathbf{x}^{\mathbf{u}^+} = \mathbf{x}^{q\mathbf{v}^+ + r} = \mathbf{x}^{\mathbf{v}^+}\mathbf{x}^{(q-1)\mathbf{v}^+ + r}$$

Lo que contradice la hipótesis de que el binomio $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$ sea primitivo en el ideal I_M.
Recíprocamente, si suponemos que el vector $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^-$ es minimal respecto del orden dado por $\sqsubset$ y procedemos por reducción al absurdo suponiendo que el binomio $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$ no es primitivo en el ideal I_M, entonces sabemos que existe otro binomio $\mathbf{x}^{\mathbf{v}^+} - \mathbf{x}^{\mathbf{v}^-} \in I_M$ tal que verifica las siguientes dos condiciones:

1. $\mathbf{x}^{\mathbf{v}^+}$ divide a $\mathbf{x}^{\mathbf{u}^+}$

$$\begin{aligned} &\Rightarrow \quad \exists f \in \mathbb{K}[\mathbf{x}] \mid \mathbf{x}^{\mathbf{u}^+} = f\mathbf{x}^{\mathbf{v}^+} \\ &\Rightarrow \quad \exp(\mathbf{x}^{\mathbf{u}^+}) \leq \exp(f\mathbf{x}^{\mathbf{v}^+}) \leq \exp(f) + \exp(\mathbf{x}^{\mathbf{v}^+}) \\ &\Rightarrow \quad \exp(\mathbf{x}^{\mathbf{u}^+}) \leq \exp(\mathbf{x}^{\mathbf{v}^+}) \Rightarrow |\mathbf{u}^+| \leq |\mathbf{v}^+|. \end{aligned}$$

   Como $\mathbf{u}^+,\ \mathbf{v}^+ \in \mathbb{N}^n$ y $|\mathbf{u}^+| \leq |\mathbf{v}^+|$ entonces se tiene que

$$|u_i| \leq |v_i| \ \text{ y } u_i \cdot v_i \geq 0 \ \text{ para todo } \ u_i \in \mathbf{u}^+ \ \text{ y } v_i \in \mathbf{v}^+.$$

2. $\mathbf{x}^{\mathbf{v}^-}$ divide a $\mathbf{x}^{\mathbf{u}^-}$

   Siguiendo el mismo razonamiento se prueba que:

$$|u_i| \leq |v_i| \ \text{ y } \ u_i \cdot v_i \geq 0 \ \text{ para todo } \ u_i \in \mathbf{u}^- \ \text{ y } v_i \in \mathbf{v}^-.$$

Luego hemos visto que existe $\mathbf{v} \in \mathbb{Z}^n$ tal que $\mathbf{v} \sqsubset \mathbf{u}$, lo que contradice la hipótesis de que el vector $\mathbf{u}$ sea minimal respecto del orden $\sqsubset$. □

Definición 2.19. *Llamaremos* base de Graver del ideal I_M*, y la denotaremos por* Gr_M*, al conjunto de binomios primitivos del ideal* I_M

Observemos que, teniendo en cuenta la definición, la *base de Graver* de una matriz $M \in \mathbb{Z}^{m\times n}$ es el conjunto de dependencias minimales respecto al orden $\sqsubset$.

Definición 2.20. *Llamamos* vectores de soporte mínimo *o* circuitos *asociados al ideal* I_M *a las palabras que tienen soporte minimal.*
Definimos el conjunto C_M *como el conjunto de circuitos asociados al ideal* I_M*.*

Recordemos que hemos denotado:

- C_M como el conjunto de circuitos del ideal I_M.

- Gr_M como el conjunto de elementos primitivos del ideal I_M que se denomina *Base de Graver*.
- $\mathcal{U}_M$ como la base de Gröbner del ideal I_M respecto de todos los órdenes monomiales simultáneamente. A $\mathcal{U}_H$ se denomina *Base de Gröbner Universal* y está formada por la unión de todas las bases de Gröbner reducidas del ideal I_M.

Proposición 2.5. *Todo binomio de la forma* $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in \mathcal{U}_M$ *es primitivo.*

Demostración. Fijamos un orden monomial $\succ$ en $\mathbb{K}[\mathbf{x}]$ y consideramos un binomio cualquiera $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$ de la base de Gröbner reducida $G_\succ$ del ideal I_M respecto de $\succ$. Supongamos, sin pérdida de generalidad, que $\mathbf{x}^{\mathbf{u}^+} \succ \mathbf{x}^{\mathbf{u}^-}$. Entonces $\mathbf{x}^{\mathbf{u}^+}$ es un generador minimal del ideal monomial $\mathrm{LT}_\succ(I_M)$ y $\mathbf{x}^{u^-}$ es un monomio estándar.
Procedamos por reducción al absurdo suponiendo que el binomio $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$ no es primitivo. Entonces sabemos que existe un vector $\mathbf{v} \in \ker \Pi \setminus \{u\}$ tal que $\mathbf{x}^{\mathbf{v}^+}$ divide a $\mathbf{x}^{\mathbf{u}^+}$ y $\mathbf{x}^{\mathbf{v}^-}$ divide a $\mathbf{x}^{\mathbf{u}^-}$.
Tenemos dos posibles casos:

- Si $\mathbf{v}^+ \succ \mathbf{v}^-$, entonces $\mathbf{x}^{\mathbf{u}^+}$ no es un generador minimal del ideal $\mathrm{LT}_\succ(I_M)$.
- Si $\mathbf{v}^+ \succ \mathbf{v}^-$, entonces $\mathbf{x}^{\mathbf{u}^-}$ no es un monomio estándar del ideal $\mathrm{LT}_\succ(I_M)$.

En ambos supuestos tenemos una contradicción. □

Proposición 2.6. *Sea* $M \in \mathbb{Z}^{m\times n}$ *y siguiendo con la notación establecida se tiene que:*

$$C_M \subseteq \mathcal{U}_M \subseteq Gr_M.$$

Demostración. Por la Proposición 2.5 todo binomio $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in \mathcal{U}_A$ es primitivo, luego queda probado que $\mathcal{U}_M \subseteq Gr_M$.
Faltaría demostrar que todo binomio asociado a un circuito del ideal I_M pertenece a una base de Gröbner reducida del ideal I_M. Procedamos por reducción al absurdo suponiendo que existe un circuito $\mathbf{u} \in \ker \Pi$, tal que su binomio asociado $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$ no pertenece a ninguna base de Gröbner reducida del ideal I_M. En particular no pertenece a la base de Gröbner reducida del ideal I_M respecto del orden $\sqsubset$.
Sin pérdida de generalidad, supongamos que en todo binomio $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$ se verifica que $\mathbf{u}^- \sqsubset \mathbf{u}^+$.
Es decir, existe $\mathbf{v} \in \ker \Pi \setminus \{0, \mathbf{u}\}$ tal que $\mathbf{v}^+ \sqsubset \mathbf{u}^+$. Por lo tanto observamos que:

- $\text{supp}(\mathbf{v}^+) \subseteq \text{supp}(\mathbf{u}^+) \subseteq \text{supp}(\mathbf{u})$
- $\text{supp}(\mathbf{v}^-) \subseteq \text{supp}(\mathbf{v}^+) \subseteq \text{supp}(\mathbf{u})$

Es decir, hemos visto que $\text{supp}(\mathbf{u}) \subseteq \text{supp}(\mathbf{v})$, como $\mathbf{u}$ es un circuito esto es posible si y sólo si $\mathbf{u} = \mathbf{v}$. □

Definición 2.21. *Sean $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^n$. Diremos que el vector $\mathbf{u}$ es conforme a $\mathbf{v}$ si $\text{supp}(\mathbf{u}^+) \subseteq \text{supp}(\mathbf{v}^+)$ y $\text{supp}(\mathbf{u}^-) \subseteq \text{supp}(\mathbf{v}^-)$.*

Lema 2.12. *Todo vector $\mathbf{v} \in \ker(\Pi)$ puede escribirse como $(n-m)$ circuitos conformes con $\mathbf{v}$.*

Demostración. Véase [60]. Capítulo 4, Lema 4.10. □

2.3.2. Cálculo de Bases de Graver

En esta sección presentaremos un algoritmo para el cálculo de la base de Graver (Gr_M) del ideal tórico asociado al núcleo de la matriz $M \in \mathbb{Z}^{m\times n}$. Para ello utilizaremos la elevación de Lawrence de la matriz M.

Definición 2.22. *Sea $M \in \mathbb{Z}^{m\times n}$, llamaremos* elevación de Lawrence *de la matriz M a la matriz*

$$M' = \begin{pmatrix} M & 0 \\ 1 & 1 \end{pmatrix} \in \mathbb{Z}^{(m+n)\times 2n}$$

Donde $1 \in \mathbb{Z}^{n\times n}$ es la matriz identidad y $0 \in \mathbb{Z}^{m\times n}$ es la matriz nula. Toda matriz de la forma de M' se dice que es de tipo Lawrence.

Lema 2.13. *La matriz M y su elevación de Lawrence M' tienen núcleos isomorfos verificando*

$$\ker(M') = \{(\mathbf{u}, -\mathbf{u}) \mid \mathbf{u} \in \ker(M)\}.$$

El ideal asociado a $\ker(M')$ es

$$I_{\ker(M')} = \langle \mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} \mid \mathbf{u} \in \ker(M)\rangle \subseteq \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_n].$$

Demostración. En efecto,

$$\begin{aligned} \ker(M') &= \left\{(\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{Z}^{2n} \mid \begin{pmatrix} A & 0 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right\} \\ &= \{(\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{Z}^{2n} \mid A \cdot \mathbf{u}_1^t = 0 \ \text{ y } \ \mathbf{u}_1 + \mathbf{u}_2 = 0\} \\ &= \{(\mathbf{u}, -\mathbf{u}) \mid \mathbf{u} \in \ker(M)\}. \end{aligned}$$

□

Teorema 2.9. *En una matriz de tipo Lawrence $M' \in \mathbb{Z}^{(m+n)\times n}$, los siguientes conjuntos de binomios coinciden:*

1. *$Gr_{M'}$, la base de Graver asociada a M'.*
2. *$\mathcal{U}_{M'}$, la base de Gröbner Universal asociada a M'.*
3. *Una base de Gröbner reducida del ideal asociado a $\ker(M')$.*
4. *Un sistema minimal de generadores del ideal asociado a $\ker(M')$.*

Demostración. Es fácil probar que:

$$u \in \ker(M) \text{ es primitivo } \iff (\mathbf{u}, -\mathbf{u}) \in \ker(M') \text{ es primitivo.} \quad (2.5)$$

Por lo tanto las bases de Graver asociadas al núcleo de la matriz M y al núcleo de su correspondiente elevación de Lawrence M' están relacionadas de la siguiente forma:

$$Gr_{M'} = \{\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} \mid \mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in Gr_M\}.$$

- Veamos en primer lugar que $Gr_{M'}$ genera al ideal $I_{M'}$.

  Consideramos cualquier binomio $\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} \in I_{M'}$.

  - Si el vector $\mathbf{u}$ es primitivo entonces por la fórmula 2.5 se tiene que:
    $$\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} \in Gr_{M'}.$$
  - En caso contrario si $\mathbf{u}$ no es primitivo por el Lema 2.12 sabemos que podemos escribir $\mathbf{u}$ como combinación lineal de $(n-m)$ circuitos conformes con $\mathbf{u}$.

- Veamos que $Gr_{M'}$ es un sistema minimal de generadores del ideal $I_{M'}$.

  Consideramos cualquier binomio $g = \mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} \in Gr_{M'}$. Definimos el conjunto de binomios B como:

  $$B = \{\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} \in Gr_{M'}\} \setminus \{g\}.$$

  Procedamos por reducción al absurdo suponiendo que el conjunto B genera al ideal $I_{M'}$. Es decir, podemos escribir g como combinación lineal de elementos del conjunto B, por lo tanto existe un binomio $\mathbf{x}^{\mathbf{v}^+}\mathbf{y}^{\mathbf{v}^-} - \mathbf{x}^{\mathbf{v}^-}\mathbf{y}^{\mathbf{v}^+} \in B$ tal que uno de sus términos divide a $\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-}$.

  Reemplazando $\mathbf{v}$ por $-\mathbf{v}$ si fuese necesario, podemos suponer que $\mathbf{x}^{\mathbf{v}^+}\mathbf{y}^{\mathbf{v}^-}$ divide a $\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-}$. Es decir, el vector $\mathbf{u}$ no es primitivo en $\ker(M)$, lo que constituye una contradicción.

- Finalmente probemos que $Gr_{M'}$ es una base de Gröbner reducida respecto de cualquier orden monomial.

  En efecto, fijemos un orden monomial y consideremos $G_\succ$ una base de Gröbner reducida del ideal $I_{M'}$ respecto de $\succ$. Por el Teorema 2.6 sabemos que $G_\succ \subseteq Gr_{M'}$.

  Procedamos por reducción al absurdo suponiendo que existe un binomio $\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} \in Gr_{M'}$ tal que $\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} \notin G_\succ$.

  Esto significa que existe un binomio $\mathbf{x}^{\mathbf{v}^+}\mathbf{y}^{\mathbf{v}^-} - \mathbf{x}^{\mathbf{v}^-}\mathbf{y}^{\mathbf{v}^+} \in G_\succ$ tal que $\mathrm{LT}(\mathbf{x}^{\mathbf{v}^+}\mathbf{y}^{\mathbf{v}^-} - \mathbf{x}^{\mathbf{v}^-}\mathbf{y}^{\mathbf{v}^+})$ divide a $\mathrm{LT}(\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+})$.

  Reemplazando $\mathbf{v}$ por $-\mathbf{v}$ y $\mathbf{u}$ por $-\mathbf{u}$ si fuese necesario podemos suponer que

$$\begin{aligned} \mathrm{LT}(\mathbf{x}^{\mathbf{v}^+}\mathbf{y}^{\mathbf{v}^-} - \mathbf{x}^{\mathbf{v}^-}\mathbf{y}^{\mathbf{v}^+}) &= \mathbf{x}^{\mathbf{v}^+}\mathbf{y}^{\mathbf{v}^-} \\ \mathrm{LT}(\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+}) &= \mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-}. \end{aligned}$$

  De donde se deduce que $\mathbf{x}^{\mathbf{v}^+}$ divide a $\mathbf{x}^{\mathbf{u}^+}$ y $\mathbf{x}^{\mathbf{v}^-}$ divide a $\mathbf{x}^{\mathbf{u}^-}$.

  Pero esto implicaría que $\mathbf{u}$ no es primitivo lo que es una contradicción.

□

El Teorema 2.9 sugiere el Algoritmo 7 para calcular Bases de Graver.

Algoritmo 7 Algoritmo para el cálculo de Bases de Graver

INPUT: Una matriz entera $M \in \mathbb{Z}^{m\times n}$

OUTPUT: Una base de Graver del ideal I_M (Gr_M).

Definimos la elevación de Lawrence de la matriz M como la matriz M'.
Fijamos un orden monomial en el anillo $\mathbb{K}[x_1,\ldots,x_n,y_1,\ldots,y_n]$.
Calculamos una base de Gröbner reducida G del ideal $I_{M'}$.
Sustituimos las variables $y_1,\ldots,y_n \longmapsto 1$ en G.
El resultado es una base de Graver $Gr_M \subseteq \mathbb{K}[\mathbf{x}]$ del ideal I_M.

Capítulo 3

Aplicaciones a la programación lineal entera

En esta sección estudiaremos la relación entre bases de Gröbner y la programación lineal entera introducida por Conti y Traverso en [22] y su interpretación combinatoria que se puede encontrar en [61].

3.1. Programación lineal entera

Se llama programación lineal al conjunto de técnicas que pretenden optimizar (maximizar o minimizar) una función lineal de tal forma que las variables de dicha función estén sujetas a una serie de restricciones expresadas por inecuaciones lineales.
La formulación general de un problema de programación lineal, que denotaremos por $\mathrm{LP}_{A,\mathbf{c}}(\mathbf{b})$, es la siguiente:

$$\mathrm{LP}_{A,\mathbf{c}}(\mathbf{b}) = \begin{cases} \text{Minimizar:} & z = \mathbf{c}\cdot\mathbf{x} \in \mathbb{K}[\mathbf{x}] \\ \text{Sujeto a:} & \begin{cases} A\mathbf{x}^t = \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} \end{cases}$$

Donde $A \in \mathbb{R}^{d\times n}$, $\mathbf{c} \in \mathbb{R}^n$ y $\mathbf{b} \in \mathbb{R}^d$ y $\cdot$ denota el producto escalar entre vectores.
Por tanto en un problema de programación lineal intervienen:

- La función $z \in \mathbb{K}[\mathbf{x}]$ llamada *función objetivo*. En esta expresión las variables $x_1, \ldots, x_n$ son las *variables de decisión*, mientras que al vector $\mathbf{c} \in \mathbb{R}^n$ se le conoce por *vector coste*.

- Las restricciones que vienen determinadas por la *matriz de coeficientes* $A \in \mathbb{R}^{d\times n}$ y el vector $\mathbf{b} \in \mathbb{R}^d$.

Al conjunto de valores $\mathbf{x} = (x_1, \ldots, x_n)$ que verifican todas las restricciones se denominan *soluciones factibles* del problema. Todo vector $\mathbf{x} \in \mathbb{R}^n$ que no sea una solución factible del problema no es una solución del problema.
Llamaremos *solución óptima* del problema a la solución $\mathbf{x}_0$ que maximice o minimice la función objetivo.
El conjunto de soluciones factibles del problema $\mathrm{LP}_{A,\mathbf{c}}(\mathbf{b})$, que denotaremos por $P_{\mathbf{b}} = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x}^t = \mathbf{b}, \mathbf{x} \geq 0\} \subseteq \mathbb{R}^n$, es un *politopo*, es decir, está definido por la intersección de un número finito de semiespacios cerrados. Diremos que el problema $\mathrm{LP}_{A,\mathbf{c}}(\mathbf{b})$ tiene solución si $P_{\mathbf{b}} \neq \emptyset$.
El politopo $P_{\mathbf{b}}$ puede estar acotado o no acotado. Para simplificar el problema supondremos que $P_{\mathbf{b}}$ está acotado, es decir, se trata de un poliedro que también puede ser definido como la envolvente convexa de un número finito de puntos llamados vértices del poliedro.
Todo punto del poliedro $P_{\mathbf{b}}$ cumple todas las restricciones dadas por el problema, de esta forma, resolver el problema lineal $\mathrm{LP}_{A,\mathbf{c}}(\mathbf{b})$ implica minimizar la función objetivo sobre el conjunto de puntos del poliedro $P_{\mathbf{b}}$. Esto implica que la solución óptima del problema $\mathrm{LP}_{A,\mathbf{c}}(\mathbf{b})$ se encuentra en un vértice del poliedro $P_{\mathbf{b}}$.
Probablemente el *método del símplex* creado en 1947 por el matemático George Dantzig es el más conocido para resolver problemas de programación lineal. Se trata de un método iterativo que, partiendo del valor de la función objetivo en un vértice cualquiera, permite obtener otra solución del problema que optimice a la anterior. Este proceso finaliza cuando no es posible seguir mejorando dicha solución. Como el número de vértices es finito, siempre puede encontrarse una solución si $P_{\mathbf{b}} \neq \emptyset$. Geométricamente, el método símplex traza un camino de aristas en el poliedro $P_{\mathbf{b}}$, comenzando en el vértice inicial y terminando en el vértice que nos aporta la solución óptima del problema.
Un problema de *programación lineal entera* es un problema de programación lineal con la restricción adicional de que los valores de la solución deben ser enteros. Podemos formular de forma general un problema de programación lineal entera que denotaremos por $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$ de la siguiente forma:

$$\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b}) = \begin{cases} \text{Minimizar:} & z = \mathbf{c}\cdot\mathbf{x} \in \mathbb{K}[\mathbf{x}] \\ \text{Sujeto a:} & \begin{cases} A\mathbf{x}^t = \mathbf{b} \\ \mathbf{x} \in \mathbb{N}^n \end{cases} \end{cases}$$

Donde podemos suponer que $A \in \mathbb{Z}^{d\times n}$, $\mathbf{c} \in \mathbb{R}^n$ y $\mathbf{b} \in \mathbb{Z}^d$.

Aunque ambos modelos sólo varían en la restricción a los enteros, los problemas de programación lineal pueden ser resueltos en tiempo polinomial, mientras que los problemas de programación lineal entera son NP-completos. Si seguimos con la misma notación, la región factible del problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$ es el conjunto discreto de puntos $\{\mathbf{x} \in \mathbb{N}^n \mid A\mathbf{x}^t = \mathbf{b}\}$. Denotaremos por $P^I_{\mathbf{b}}$ a la envolvente convexa de la región factible del problema entero. Se tiene que $P^I_{\mathbf{b}}$ es un poliedro y que $P^I_{\mathbf{b}} \subseteq P_{\mathbf{b}}$. Luego, si tenemos el problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$ y resolvemos el modelo de programación lineal asociado $\mathrm{LP}_{A,\mathbf{c}}(\mathbf{b})$, estaremos obteniendo la solución de la *relajación continua* del modelo entero. En el caso de que el problema busque maximizar (minimizar) la función objetivo, la relajación continua nos proporciona una cota superior (respectivamente inferior) del valor óptimo del modelo de programación entera asociado. Si la solución que nos proporciona la relajación continua es entera, entonces esta solución también es una solución del modelo de programación entera asociado. En caso contrario se hace necesario la aplicación de métodos específicos para resolver este tipo de problemas.

Los dos métodos más representativos son el *método de los planos de corte de Gomory* y el *método de ramificación y acotación* (*branch and bound*).

En el *método de planos de corte de Gomory* en primer lugar se resuelve la relajación continua del modelo entero. Si la solución obtenida $\mathbf{x}_0$ es entera, entonces ésta es la solución de nuestro problema original. En caso contrario se construye un *plano de corte*, un hiperplano $\alpha\mathbf{x}^t = \beta$, que separa la solución obtenida $\mathbf{x}_0$ del politopo $P^I_{\mathbf{b}}$. Es decir, se verifica que:

$$\alpha\mathbf{x}_0^t > \beta \text{ y } P^I_{\mathbf{b}} \subseteq \{\mathbf{x} \mid \alpha\mathbf{x}^t \leq \beta\}.$$

Se añade la desigualdad $\alpha\mathbf{x}^t \leq \beta$ al conjunto de restricciones del problema y comenzamos de nuevo la búsqueda de la solución óptima, así sucesivamente hasta que una solución entera haya sido alcanzada o se demuestre que no existe. Este proceso termina en un número finito de pasos construyendo planos de corte, a partir de una solución no entera, de tal forma que los cortes asociados generan de forma iterada la solución entera buscada, si existe.

Para estudiar este método dirigimos al lector al primer artículo de Gomory [34] y a su segunda versión [33]. También al artículo de Beale [8] y a la lectura de [55].

En el *método de ramificación y acotación* se obtiene la solución óptima a un problema de programación lineal entera realizando dos operaciones básicas sobre la región factible de la relajación continua del modelo entero. La primera operación es la *ramificación* en la que se divide la región factible del

problema en distintos subconjuntos, obteniendo subproblemas del problema original. La segunda operación es la *acotación* que determina una cota que se utiliza para ordenar las soluciones óptimas de los distintos subproblemas y, así, determinar la solución óptima del problema entero.
A diferencia del método de Gomory, el método de ramificación y acotación permite resolver problemas de programación lineal entera con relativa facilidad. Como primer antecedente se puede consultar el artículo de A. H. Land y A. G. Doig [41].

3.1.1. Algoritmo Conti-Traverso

Aunque los problemas de programación entera son NP-completos, existen algunos algoritmos que mejoran la complejidad en determinados casos. En 1991 Conti y Traverso presentaron un algoritmo que permite resolver problemas de programación entera utilizando bases de Gröbner, véase [22].
La relación entre bases de Gröbner y problemas de programación entera fue estudiada de forma independiente por primera vez por Ollivier en [51] y por Pottier en [53]. Luego Conti en [23] desarrolló un algoritmo utilizando bases de Gröbner asociadas a ideales tóricos que aporta la solución óptima de un problema de programación entera.
En términos generales este algoritmo necesita una matriz de coeficientes $A \in \mathbb{Z}^{m\times n}$, un vector $b \in \mathbb{Z}^m$ y un vector de peso $\mathbf{w} \in \mathbb{R}^n$. De esta forma define un ideal tórico I_A, un orden monomial $<_{\mathbf{w}}$ y un monomio f_b, calcula una base de Gröbner del ideal I_A respecto del orden $<_{\mathbf{w}}$ y devuelve el exponente de la forma normal del monomio f_b respecto de la base de Gröbner calculada. Este exponente coincide con la solución óptima del problema de programación lineal entera $\mathrm{IP}_{A,\mathbf{w}}(b)$. Estudiaremos a continuación los detalles de este algoritmo.
Siguiendo con la notación de las secciones anteriores, sea $\mathbb{K}$ un cuerpo arbitrario, denotaremos por $\mathbb{K}[\mathbf{x}] := \mathbb{K}[x_1, \ldots, x_n]$ y por $\mathbb{K}[\mathbf{y}] := \mathbb{K}[y_1, \ldots, y_m]$ al anillo, en n y m indeterminadas respectivamente, sobre el cuerpo $\mathbb{K}$.

Definición 3.1. *Sea $<$ un orden monomial en $\mathbb{K}[\mathbf{x}]$ y $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ un vector peso. Se define en $\mathbb{K}[\mathbf{x}]$ el* orden de peso $<_{\mathbf{w}}$ *como:*

$$\mathbf{x}^\alpha <_{\mathbf{w}} \mathbf{x}^\beta \Leftrightarrow \mathbf{w}\cdot\alpha < \mathbf{w}\cdot\beta \quad ó \quad \mathbf{w}\cdot\alpha = \mathbf{w}\cdot\beta \quad y \quad \mathbf{x}^\alpha < \mathbf{x}^\beta,$$

donde $\cdot$ denota el producto escalar entre vectores.

En primer lugar estudiamos el caso en que los elementos de la matriz $A \in \mathbb{Z}^{m\times n}$ y del vector $\mathbf{b} \in \mathbb{Z}^m$ son todos positivos. Si denotamos a las columnas

de la matriz A como $\mathbf{a}^j = (a_{1j}, \ldots, a_{mj})$ para todo $j = 1, \ldots, n$, resolver el problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$ se puede entender como encontrar una representación del vector $\mathbf{b}$ como elemento del submonoide de $\mathbb{N}^m$ generado por las columnas de A. Sólo como recordatorio exponemos la siguiente definición.

Definición 3.2. *Una operación binaria definida sobre un conjunto A es una función*

$$\begin{array}{rccc} \oplus: & A \times A & \longrightarrow & A \\ & (\mathbf{a}, \mathbf{b}) & \longmapsto & \mathbf{a} \oplus \mathbf{b}. \end{array}$$

Diremos que $\oplus$ es asociativa si $\mathbf{a} \oplus (\mathbf{b} \oplus \mathbf{c}) = (\mathbf{a} \oplus \mathbf{b}) \oplus \mathbf{c}$, para todo $\mathbf{a},\ \mathbf{b},\ \mathbf{c} \in A$. Diremos que es commutativa si $\mathbf{a} \oplus \mathbf{b} = \mathbf{b} \oplus \mathbf{a}$, para todo $\mathbf{a},\ \mathbf{b} \in A$.
Un monoide es un conjunto A provisto de una operación interna. Un monoide es asociativo o commutativo si lo es su operación y es finito si lo es su conjunto subyacente. Un elemento $\mathbf{e} \in A$ es neutro si $\mathbf{e} \oplus \mathbf{a} = \mathbf{a} \oplus \mathbf{e} = \mathbf{a}$, para todo $\mathbf{a} \in A$. Diremos que un monoide es unitario si tiene neutro. Un semigrupo es un monoide unitario y asociativo.

Sea $\mathbb{K}$ un cuerpo arbitrario y consideremos $\mathbb{K}[\mathbf{y}] := \mathbb{K}[y_1, \ldots, y_m]$ y $\mathbb{K}[\mathbf{x}] := \mathbb{K}[x_1, \ldots, x_n]$ el anillo de polinomios en m y n indeterminadas respectivamente y $N = \langle \alpha_1, \ldots, \alpha_m \rangle$ un submonoide de $\mathbb{N}^m$, con $\alpha_j = \mathbf{y}^{\mathbf{a}_j} = \prod_{i=1}^m y_i^{a_{ij}}$. De manera que se tiene que:

$$\beta = \mathbf{y}^{\mathbf{b}} = \prod_{i=1}^m y_i^{b_i} \in \mathbb{K}[N] \Leftrightarrow \mathbf{b} \in N.$$

Por lo tanto resolver el problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$ es equivalente a resolver el problema de pertenencia a una subálgebra. Para más referencias sobre cómo resolver el problema de pertenencia, véanse [57] y [59].
Definimos el siguiente homomorfismo entre anillos:

$$\begin{array}{rccl} \Theta: & \mathbb{K}[\mathbf{x}, \mathbf{y}] & \longrightarrow & \mathbb{K}[\mathbf{y}] \\ & \mathbf{x}^u & \longmapsto & \Theta(\mathbf{x}^u) = \mathbf{y}^{Au^t} \\ & x_j & \longmapsto & \Theta(x_j) = \alpha_j = \mathbf{y}^{\mathbf{a}^j} = \prod_{i=1}^m y_i^{a_{ij}} \\ & y_j & \longmapsto & \Theta(y_j) = y_j \end{array}$$

El núcleo del homomorfismo Θ forma un ideal tórico asociado a la matriz A, que denotaremos por el ideal $I_A = \ker \Theta \subseteq \mathbb{K}[\mathbf{x}, \mathbf{y}]$. Como ya vimos en el Teorema 2.9, el ideal tórico I_A está generado, como $\mathbb{K}$-espacio vectorial, por el siguiente conjunto de binomios:

$$I_A \ = \ \langle \{ \mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \mid \mathbf{u} = \mathbf{u}^+ - \mathbf{u}^- \in \ker(A) \} \rangle.$$

Es fácil ver que $\mathbb{K}[\mathbf{y}] \cong \mathbb{K}[\mathbf{x},\mathbf{y}]/I_A$ y que $\mathbb{K}[N] \cong \mathbb{K}[\mathbf{x}]/I_A \cap \mathbb{K}[\mathbf{x}]$.
Por el Corolario 2.5, fijado $<$ un orden monomial en $\mathbb{K}[\mathbf{x}]$, la base de Gröbner reducida del ideal I_A respecto de $<$ consiste en un conjunto finito de binomios de la forma $\{\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \mid \mathbf{u} \in \ker A\}$.

Lema 3.1. *Denotamos por* $G_{\mathbf{c}} = \{\mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i} \mid i = 1,\dots,s\}$ *la base de Gröbner reducida del ideal* I_A *respecto del orden* $<_{\mathbf{c}}$. *Si* $\mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i} \in G_{\mathbf{c}}$, *asumiremos que* $\mathbf{c}\cdot\alpha_i > \mathbf{c}\cdot\beta_i$. *Entonces para cada binomio* $\mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i} \in G_{\mathbf{c}}$, β_i *es la única solución óptima del problema lineal entero* $\mathrm{IP}_{A,\mathbf{c}}(A\alpha_i^t)$.

Demostración. En primer lugar observemos que, por definición de base de Gröbner reducida, se tiene que el conjunto de binomios

$$\{\mathbf{x}^{\alpha_i} \mid i = 1,\dots,s\}$$

es un conjunto de generadores minimal del ideal inicial $\mathrm{LT}_{<_{\mathbf{c}}}(I_A)$.
Además, también por definición, por cada binomio $\mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i} \in G_{\mathbf{c}}$ se tiene que $A\alpha_i^t = A\beta_i^t$, $\alpha_i,\ \beta_i \in \mathbb{N}^n$ y que $\mathbf{c}\cdot\alpha_i > \mathbf{c}\cdot\beta_i$.
Procedamos por reducción al absurdo suponiendo que β_i no es la solución óptima del problema $\mathrm{IP}_{A,\mathbf{c}}(A\alpha_i^t)$ entonces $\mathbf{x}^{\beta_i}$ pertenece al ideal inicial $\mathrm{LT}(I_A)$ y, por tanto, existe un $\mathbf{x}^{\alpha_j}$ con $j = \{1,\dots,s\}$ que divide a $\mathbf{x}^{\beta_i}$, lo que contradice la definición de base de Gröbner reducida. □

Recordemos que llamamos *forma normal* de un monomio $f \in \mathbb{K}[\mathbf{x}]$ respecto de la base de Gröbner $G_{\mathbf{c}}$ del ideal I_A utilizando el orden monomial $<_{\mathbf{c}}$, que denotamos por $\mathrm{nf}_{G_{\mathbf{c}}}(f)$, al resto de la división de f por todos los elementos de $G_{\mathbf{c}}$.
Sea $G_{\mathbf{c}}$ una base de Gröbner reducida del ideal I_A respecto a un orden de eliminación para las variables $\mathbf{y}$ y que sea compatible con el vector peso $\mathbf{c}$, entonces, por el Lema 2.8, $G'_{\mathbf{c}} = G_{\mathbf{c}} \cap \mathbb{K}[\mathbf{x}]$ es una base de Gröbner reducida del ideal de eliminación $I_A \cap \mathbb{K}[\mathbf{x}]$. Por lo tanto se tiene que:

$$f \in \mathbb{K}[N] \Leftrightarrow \mathrm{nf}_{G_{\mathbf{c}}}(f) \in \mathbb{K}[\mathbf{x}].$$

Corolario 3.1. *Fijado un orden monomial* $<$, *la forma normal de un monomio* $f \in \mathbb{K}[\mathbf{x}]$ *respecto de la base de Gröbner* $G_{\mathbf{c}}$ *del ideal* I_A *es un monomio.*

Demostración. Todo elemento g de la base de Gröbner reducida del ideal monomial I_A es un binomio y el resto de un monomio por un binomio es un monomio. Luego $\mathrm{nf}_{G_{\mathbf{c}}}(f)$ es un monomio. □

Es decir, la forma normal del monomio $\mathbf{y}^{\mathbf{b}}$ respecto de la base de Gröbner $G_{\mathbf{c}}$ del ideal I_A es de nuevo un monomio $\beta = \mathrm{nf}_{G_{\mathbf{c}}}(\mathbf{y}^{\mathbf{b}})$.
Si el monomio β contiene alguna variable y_i, con $i = 1, \ldots, m$, entonces se tiene que $\mathbf{y}^{\mathbf{b}} \notin \mathbb{K}[N]$, lo que implica que el problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$ no tiene solución. En caso contrario, si $\beta = \prod_{j=1}^{n} x_j^{\gamma_j} \in \mathbb{K}[\mathbf{x}]$, entonces el exponente $(\gamma_1, \ldots, \gamma_n)$ es solución óptima del problema entero considerado.

Teorema 3.1. *Fijado $<$ un orden monomial en $\mathbb{K}[\mathbf{x}]$. Sean $A \in \mathbb{Z}^{m\times n}$ una matriz de coeficientes, $\mathbf{b} \in \mathbb{Z}^m$ un vector arbitrario y $\mathbf{c} \in \mathbb{R}^n_{\geq 0}$ un vector de peso. Calculamos $G_{\mathbf{c}}$ una base de Gröbner del ideal I_A respecto del orden de peso $<_{\mathbf{c}}$ y consideramos $\mathbf{u}$ una solución factible del problema lineal entero $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$.*
Entonces el exponente del monomio $\mathrm{nf}_{G_{\mathbf{c}}}(\mathbf{x}^{\mathbf{u}}) \in \mathbb{K}[\mathbf{x}]$ es una solución óptima del problema lineal entero $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$.

Demostración. Por el Corolario 3.1 sabemos que la forma normal de un monomio $\mathbf{y}^{\mathbf{u}}$ respecto a la base de Gröbner $G_{\mathbf{c}}$ de I_A es un monomio, por lo tanto existe $\mathbf{u}_1 \in \mathbb{N}^n$ tal que $\mathrm{nf}_{G_{\mathbf{c}}}(\mathbf{x}^{\mathbf{u}}) = \mathbf{x}^{\mathbf{u}_1}$.

- Veamos en primer lugar que $\mathbf{u}_1$ es una solución factible del problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$.

  Sea $G_{\mathbf{c}} = \{g_1, \ldots, g_s\}$ la base de Gröbner reducida del ideal I_A respecto del orden monomial $<_{\mathbf{c}}$, entonces por hipótesis se tiene que:

  $$\mathbf{x}^{\mathbf{u}} = \sum_{i=1}^{s} \lambda_i g_i + \mathbf{x}^{\mathbf{u}_1} \Rightarrow \mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{u}_1} \in I_A.$$

  Luego por definición $A\mathbf{u}_1^t = A\mathbf{u}^t = \mathbf{b}$

- Veamos ahora que $\mathbf{u}_1$ es la solución óptima del problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$.

  Procedamos por reducción al absurdo suponiendo que existe $\mathbf{v} \in \mathbb{N}^n$ tal que $\mathbf{v}$ es solución óptima del problema $IP_{A,\mathbf{c}}(\mathbf{b})$, entonces se tiene por una parte que $f = \mathbf{x}^{\mathbf{u}_1} - \mathbf{x}^{\mathbf{v}} \in I_A$ y que, como $\mathbf{v} <_{\mathbf{c}} \mathbf{u}$, $\mathrm{LT}_{<_{\mathbf{c}}}(f) = \mathbf{x}^{\mathbf{u}_1}$. Por lo tanto $\exists g_i = \mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i} \in G_{\mathbf{c}}$ tal que $\mathbf{x}^{\alpha_i}$ divide a $\mathbf{x}^{\mathbf{u}_1}$, lo que contradice la hipótesis de que $\mathbf{x}^{\mathbf{u}_1}$ sea la forma normal de $\mathbf{x}^{\mathbf{u}}$.

□

Para aplicar este algoritmo a problemas $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$ donde tanto los coeficientes de la matriz A como los del vector $\mathbf{b}$ pueden tomar valores enteros negativos, se utiliza el ideal

$$J_A := \langle \{x_j \mathbf{t}^{\mathbf{a}_j^+} - \mathbf{t}^{\mathbf{a}_j^-}\}_{j=1}^{n} \ , \ t_0 \cdot t_1 \cdots t_d - 1 \rangle,$$

descrito en el anillo de polinomios $\mathbb{K}[t_0, \ldots, t_d, x_1, \ldots, x_n]$ y donde $\mathbf{a}_j = \mathbf{a}_j^+ - \mathbf{a}_j^-$ representa la columna j de la matriz A con $\mathbf{a}_j^+,\ \mathbf{a}_j^- \in \mathbb{N}^m$.
El ideal tórico I_A del caso anterior se puede definir como el ideal de eliminación $J_A \cap \mathbb{K}[\mathbf{x}]$. Por lo tanto, sea $\mathcal{G}_\mathbf{c}$ una base de Gröbner reducida del ideal J_A respecto a un orden de eliminación para las variables $\mathbf{y}$ y $\mathbf{t}$ y que sea compatible con el vector peso $\mathbf{c}$, por el Lema 2.8, $G_\mathbf{c} = \mathcal{G}_\mathbf{c} \cup \mathbb{K}[\mathbf{x}]$ es una base de Gröbner reducida del ideal I_A.
Observamos que $\mathbb{K}[\mathbf{y}] \cong \mathbb{K}[\mathbf{y}, \mathbf{t}, \mathbf{x}]/J_A$, por lo tanto podemos proceder como en el caso anterior.
Sin embargo, nuestro método no funciona si el vector peso $\mathbf{c}$ tiene alguna componente negativa, por eso necesitamos la siguiente definición:

Definición 3.3. *Sea $>$ un orden monomial en $\mathbb{K}[\mathbf{x}]$ y $\mathbf{c} \in \mathbb{R}^n$ un vector de peso. Se dice que el orden monomial $>_\mathbf{c}$ de $\mathbb{K}[\mathbf{x}]$ es un orden compatible con la función peso definida por el vector $\mathbf{c}$ si para cualquier vector $\mathbf{u},\ \mathbf{v} \in \mathbb{Z}^n$ tales que $A\mathbf{u}^t = A\mathbf{v}^t$, si $\mathbf{c} \cdot \mathbf{u} < \mathbf{c} \cdot \mathbf{v}$, entonces $\mathbf{x}^\mathbf{u} <_\mathbf{c} \mathbf{x}^\mathbf{v}$.*

Lema 3.2. *Sea $\mathbf{c} \in \mathbb{R}^n$ y consideremos $V \subseteq \mathbb{R}^n$ el subespacio de soluciones reales al sistema $A\mathbf{x}^t = 0$ y $C \subseteq \mathbb{R}^n_{\geq 0}$ de vectores $\mathbf{x}$ sin componentes negativas tales que $\mathbf{c} \cdot \mathbf{x} < 0$. Entonces si $C \cap V \neq \emptyset$, entonces no existe una solución óptima del problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$.*

Demostración. Véase [22], Sección 3, Lema 1. □

Teorema 3.2. *Si existe una solución óptima del problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{b})$, entonces existe un orden monomial compatible con la función peso del problema.*

Demostración. Véase [22], Sección 3, Teorema 1. □

El Teorema 3.1 sugiere el Algoritmo 8 para resolver problemas de programación lineal entera utilizando bases de Gröbner expuesto por Conti y Traverso en [22].

Algoritmo 8 Algoritmo de Conti-Traverso

INPUT: $A \in \mathbb{Z}^{m \times n}$, $\mathbf{b} \in \mathbb{Z}^m$, $\mathbf{w} \in \mathbb{R}^n$
OUTPUT: Una solución óptima del problema $\mathrm{IP}_{A,\mathbf{c}}(\mathbf{c})$.
Calcular una base de Gröbner $G_\mathbf{c}$ del ideal I_A respecto del orden $<_\mathbf{c}$.
Calcular la forma normal del monomio $\mathbf{y}^\mathbf{b}$ respecto de $G_\mathbf{c}$.
Devolver el exponente del vector de la forma normal obtenida.

3.1.2. Test sets y bases de Graver

Sea $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$ el problema de programación lineal entera asociado a la matriz de coeficientes $A \in \mathbb{Z}^{m\times n}$, al vector de peso $\mathbf{w} \in \mathbb{R}^n$ y al vector $\mathbf{b} \in \mathbb{Z}^m$. Y sea $P^I_{\mathbf{b}}$ la envolvente convexa de la región factible asociada a dicho problema.

Definición 3.4 (Test Set)**.** *Al conjunto de vectores* $t \in \ker_{\mathbb{Z}} A$, $t >_{\mathbf{w}} 0$ *tales que para una solución no óptima* $\mathbf{u}$ *del problema* $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$ *se tiene que* $\mathbf{u} - t \in P^I_{\mathbf{b}}$ *se denomina* Test Set *de la familia de problemas* $\mathrm{IP}_{A,\mathbf{w}}$ *y se denota por* $\mathcal{T}_{<_{\mathbf{w}}}$.

Si existe un Test Set finito para $\mathrm{IP}_{A,\mathbf{w}}$ entonces tenemos un método trivial para obtener soluciones óptimas para todos los problemas de la familia $\mathrm{IP}_{A,\mathbf{w}}$. En efecto, comenzamos por una solución arbitraria $\mathbf{u}$ del problema $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$ y, restando elementos apropiados del conjunto $\mathcal{T}_{<_{\mathbf{w}}}$, vamos mejorando la solución. En un número finito de pasos obtenemos una solución óptima del problema. Sabremos que el vector $\mathbf{u}'$ es la solución óptima del problema si no existe ningún vector $t \in \mathcal{T}_{<_{\mathbf{w}}}$ tal que $\mathbf{u}' - t \in P^I_{\mathbf{b}}$.
En la sección anterior habíamos definido:

- La aplicación $\mathbb{Z}$-lineal Π como:

$$\begin{array}{rccc} \Pi: & \mathbb{Z}^n & \longrightarrow & \mathbb{Z}^m \\ & \mathbf{u} & \longmapsto & \Pi(\mathbf{u}) = A\mathbf{u}^t \end{array}$$

- El homomorfismo de anillos Θ como:

$$\begin{array}{rccc} \Theta: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{K}[\mathbf{x}] \\ & \mathbf{x}^{\mathbf{u}} & \longmapsto & \theta(\mathbf{x}^{\mathbf{u}}) = \mathbf{y}^{A\mathbf{u}^t} \end{array}$$

- El ideal tórico I_A asociado a la matriz de coeficientes $A \in \mathbb{Z}^{m\times n}$ como el núcleo del homomorfismo Θ y como ya hemos visto está generado como $\mathbb{K}$-espacio vectorial por el conjunto de binomios:

$$I_A = \langle \{\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \mid \mathbf{u} \in \ker(\Pi)\}\rangle.$$

- Y el orden con peso $<_{\mathbf{w}}$ asociado al vector peso $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ y a un orden monomial $<$ fijado de $\mathbb{K}[\mathbf{x}]$.

Proposición 3.1. *La base de Gröbner reducida* G *del ideal* I_A *respecto del orden monomial de peso* $<_{\mathbf{w}}$ *es un Test Set para la familia de problemas* $\mathrm{IP}_{A,\mathbf{w}}$.

Demostración. Sea $\mathbf{u}'$ una solución óptima del problema $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$. Si $\mathbf{u} \neq \mathbf{u}'$ una solución factible del mismo problema sabemos que el binomio $\mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{u}'}$ pertenece al ideal I_A, siendo $\mathbf{x}^{\mathbf{u}}$ el término líder de dicho binomio.
Por lo tanto, existe un elemento $\mathbf{x}^{\mathbf{v}_i^+} - \mathbf{x}^{\mathbf{v}_i^-} \in G$ tal que $\mathbf{x}^{\mathbf{v}_i^+}$ divide a $\mathbf{x}^{\mathbf{u}}$ dando como resto el monomio $\mathbf{x}^{\mathbf{u}-\mathbf{v}_i^+ + \mathbf{v}_i^-} = \mathbf{x}^{\mathbf{u}-\mathbf{v}}$.
Observamos que $\mathbf{u} >_{\mathbf{w}} \mathbf{u} - \mathbf{v}$ y que, como $x^{\mathbf{v}_i^+} - x^{\mathbf{v}_i^-}$ pertence a $G \subseteq I_A$, entonces $\Pi(\mathbf{u}) = \Pi(\mathbf{u} - \mathbf{v})$. Por lo tanto, $\mathbf{u} - \mathbf{v}$ es una solución del problema $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$ que mejora la solución $\mathbf{u}$.
Además observamos que $x^{\mathbf{u}'}$ no puede ser dividido por el término líder de ningún elemento de G, pues en caso contrario obtendríamos una solución del problema $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$ más pequeña que la solución $\mathbf{u}'$, lo que contradice el hecho de que $\mathbf{u}'$ sea una solución óptima de dicho problema.
Por lo tanto hemos probado que la base de Gröbner reducida G es un Test Set para el problema $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$. □

Recordemos que definimos ideal inicial de I_A y lo denotamos por $\mathrm{LT}(I_A)$ al conjunto de formas iniciales de los elementos de I_A.
Debemos tener en cuenta que todo vector $\mathbf{u} \in \mathbb{N}^n$ es solución del problema $\mathrm{IP}_{A,\mathbf{w}}(A\mathbf{u}^t)$ y no es solución de ningún otro problema de la familia $\mathrm{IP}_{A,\mathbf{w}}$.
Los siguientes Corolarios dan una caracterización para conocer si el vector $\mathbf{u} \in \mathbb{N}^n$ es solución óptima o no del problema $\mathrm{IP}_{A,\mathbf{w}}(A\mathbf{u}^t)$.

Corolario 3.2. *Un vector $\mathbf{u} \in \mathbb{N}^n$ no es una solución óptima del problema $\mathrm{IP}_{A,\mathbf{w}}(A\mathbf{u}^t)$ si y sólo si $\mathbf{x}^{\mathbf{u}} \in \mathrm{LT}(I_A)$.*

Corolario 3.3. *Para cada elemento $\mathbf{x}^{\mathbf{v}_i^+} - \mathbf{x}^{\mathbf{v}_i^-} \in G$ se tiene que $\mathbf{x}^{\mathbf{v}_i^+}$ es un generador minimal del ideal monomial $\mathrm{LT}(I_A)$ y $\mathbf{v}_i^-$ es la única solución óptima del problema $\mathrm{IP}_{A,\mathbf{w}}(A\mathbf{v}_i^+)$.*

Corolario 3.4. *La base de Gröbner reducida G del ideal I_A respecto del orden monomial de peso $<_{\mathbf{w}}$ es el único Test Set definido para la familia de problemas $\mathrm{IP}_{A,\mathbf{w}}$.*

Definición 3.5 (Test Set Universal)**.** *Al conjunto de Test Set de la familia de problemas $\mathrm{IP}_{A,\mathbf{w}}$ para todo vector peso $\mathbf{w} \in \mathbb{R}^n$ se denomina* Test Set universal *y lo denotamos por $\mathcal{U}_A$.*

Proposición 3.2. *La base de Graver asociada a A es un test set universal para la familia de problemas IP_A.*

3.2. Programación lineal modular

Como hemos visto en la sección anterior, Conti y Traverso presentaron en 1991 en [22] un algoritmo efectivo para resolver problemas de programación entera utilizando bases de Gröbner. En 2002, Ikegami y Kaji extendieron este algoritmo para poder resolver problemas de programación entera modular (ver [37]).
A partir de ahora vamos a utilizar las siguientes aplicaciones:

$$\blacktriangledown : \ \mathbb{Z}^s \longrightarrow \mathbb{Z}_q^s \qquad \text{and} \qquad \blacktriangle : \ \mathbb{Z}_q^s \longrightarrow \mathbb{Z}^s$$

donde el entero s queda determinado por el contexto, siendo válido también el espacio de las matrices. Ambas aplicaciones actúan coordenada a coordenada y pueden ser utilizadas con vectores y matrices. La aplicación $\blacktriangledown$ consiste en realizar la reducción módulo q, mientras que la aplicación $\blacktriangle$ reemplaza la clase $0, \ldots, q-1$ por el mismo símbolo entendido como un entero.
Un problema de *programación lineal entera modular* es un problema de programación lineal en aritmética modular. La formulación general de un problema de programación lineal entero modular sobre el entero $q \geq 2$, que denotaremos por, $IP_{A,\mathbf{c},q}(\mathbf{b})$, es la siguiente:

$$\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b}) = \begin{cases} \text{Minimizar:} & z = \mathbf{w} \cdot \blacktriangle \mathbf{x} \in \mathbb{K} \\ \text{Sujeto a:} & \begin{cases} A\mathbf{x}^t \equiv \mathbf{b} \mod q \\ \mathbf{x} \in \mathbb{Z}_q^n \end{cases} \end{cases}$$

Donde $A \in \mathbb{Z}_q^{d \times n}$, $\mathbf{w} \in \mathbb{R}^n$ y $\mathbf{b} \in \mathbb{Z}_q^d$.
Observamos que las restricciones del problema están descritas en aritmética modular mientras que la optimización de la función objetivo se realiza sobre los números reales.
Definimos la región factible del problema $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ al conjunto discreto de puntos $\{\mathbf{x} \in \mathbb{Z}_q^n \mid A\mathbf{x}^t \equiv \mathbf{b} \mod q\}$. Diremos que la solución $\mathbf{u} \in \mathbb{Z}_q^n$ del problema $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ es óptima si el vector $\blacktriangle \mathbf{u} \in \mathbb{Z}^n$ minimiza la función objetivo.

3.2.1. Algoritmo Ikegami-Kaji

La diferencia principal con el algoritmo extendido de Conti y Traverso es traducir los teoremas expuestos en la sección anterior para trabajar módulo q. Para ello necesitamos añadir los binomios $\{y_j^q - 1 \ \text{ con } \ j \in \{1, \ldots, m\}\}$ al ideal que se utiliza en el algoritmo de Conti-Traverso y considerar la elevación de Lawrence de la matriz de coeficientes.

En primer lugar vamos a considerar el caso en que el vector peso $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ no contiene componentes negativas.
Supongamos que $\mathbf{x}$ denota n variables $x_1, \ \dots \ , x_n$ y $\mathbf{y}$ denota m variables $y_1, \ \dots \ , y_m$. Consideramos el homomorfismo de anillos Θ definido por:

$$\begin{array}{rccc} \Theta: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{K}[\mathbf{y}] \\ & \mathbf{x^u} & \longmapsto & \Theta(\mathbf{x^u}) = \mathbf{y}^{(\blacktriangle A)\mathbf{u}^t} \end{array}$$

Definimos el ideal J_q como el ideal binomial definido por $J_q = \langle \{y_i^q - 1\}_{i=1}^m \rangle$ en el anillo de polinomios $\mathbb{K}[\mathbf{y}]$.
En el caso no modular es fácil ver que

$$(\blacktriangle A)(\blacktriangle \mathbf{u})^t = (\blacktriangle \mathbf{b}) \Leftrightarrow \Theta(\mathbf{x}^{\blacktriangle \mathbf{u}}) = \mathbf{y}^{\blacktriangle \mathbf{b}},$$

donde $A \in \mathbb{Z}_q^{m \times n}$, $\mathbf{b} \in \mathbb{Z}_q^n$ y $\mathbf{u} \in \mathbb{Z}_q^n$.
Con el siguiente lema quedaría probado la afirmación anterior módulo q.

Lema 3.3. *$A\mathbf{u}^t \equiv \mathbf{b} \mod q$ si y sólo si $\Theta(\mathbf{x}^{\blacktriangle \mathbf{u}}) \equiv \mathbf{y}^{\blacktriangle \mathbf{b}} \mod J_q$.*

Demostración. Véase [37], Lema 1. □

En [37] se prueba que el ideal binomial asociado a los vectores del $\mathbb{Z}_q$-núcleo de la matriz de coeficientes A, que coincide con el núcleo del homomorfismo Θ, viene dado por el ideal de eliminación $I = I_A \cap \mathbb{K}[\mathbf{x}]$ donde:

$$I_A = \langle \{\phi_i - x_i\}_{i=1}^n, \{y_j^q - 1\}_{j=1}^m \rangle \subseteq \mathbb{K}[\mathbf{x}, \mathbf{y}], \text{ y } \phi_i = \Theta(x_i) = \prod_{j=1}^m y_j^{a_{i,j}},$$

donde $a_{i,j}$, con $i \in \{1, \dots, n\}$ y $j \in \{1, \dots, m\}$, representa los elementos de la matriz $\blacktriangle A$.
En otras palabras, podemos ver el ideal relacionado con el $\mathbb{Z}_q$-núcleo de la matriz de coeficientes $A \in \mathbb{Z}_q^{m \times n}$ como el ideal de eliminación relacionado con el $\mathbb{Z}$-núcleo de la matriz $(\blacktriangle A, qI_m) \in \mathbb{Z}^{m \times (m+n)}$, donde $I_m \in \mathbb{Z}^{m \times m}$ denota la matriz identidad de tamaño m.
El siguiente lema nos aporta una caracterización de los polinomios que pertenecen al ideal $I = I_A \cap \mathbb{K}[\mathbf{x}]$. La condición necesaria ya está probada en el Lema 7 de [37].

Lema 3.4. *$f \in I_A \cap \mathbb{K}[\mathbf{x}]$ si y sólo si $f \in \mathbb{K}[\mathbf{x}]$ y $\Theta(f) \equiv 0 \mod J_q$.*

Demostración. Consideramos $f \in I_A \cap \mathbb{K}[\mathbf{x}]$, sabemos que f se puede representar como combinación lineal de los generadores del ideal I_A, es decir:

$$f(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \lambda_i(\phi_i - x_i) + \sum_{j=1}^{m} \beta_j(y_j^q - 1),$$

donde $\lambda_1, \ldots, \lambda_n, \beta_1, \ldots, \beta_m \in \mathbb{K}[\mathbf{x},\mathbf{y}]$.
De esta forma:

$$\begin{aligned} \Theta(f) &= f(\Theta(x_1), \ldots, \Theta(x_n), y_1, \ldots, y_m) \\ &= \sum_{i=1}^{n} \Theta(\lambda_i)(\phi_i - \Theta(x_i)) + \sum_{j=1}^{m} \Theta(\beta_j)(y_j^q - 1) \\ &= \sum_{j=1}^{m} \Theta(\beta_j)(y_j^q - 1) \equiv 0 \mod J_q. \end{aligned}$$

Para probar el inverso, en primer lugar observemos que dado un vector $\mathbf{u} \in \mathbb{Z}_q^n$, existen polinomios $B_1, \ldots, B_n \in \mathbb{K}[\mathbf{x},\mathbf{y}]$ tales que el monomio $\mathbf{x}^{\blacktriangle \mathbf{u}} = x_1^{u_1} \cdots x_n^{u_n}$ puede expresarse como:

$$\begin{aligned} x_1^{u_1} \cdots x_n^{u_n} &= (\phi_1 + (x_1 - \phi_1))^{u_1} \cdots (\phi_n + (x_n - \phi_n))^{u_n} \\ &= \phi_1^{u_1} \cdots \phi_n^{u_n} + B_1(x_1 - \phi_1) + \ \ldots \ + B_n(x_n - \phi_n). \end{aligned}$$

Por lo tanto existen polinomios $C_1, \ldots, C_n \in \mathbb{K}[\mathbf{x},\mathbf{y}]$, tales que todo polinomio $f \in \mathbb{K}[\mathbf{x}]$ se puede expresar como:

$$f(\mathbf{x}) = f(\phi_1, \ \ldots \ , \phi_n) + \sum_{i=1}^{n} C_i(x_i - \phi_i).$$

Teniendo en cuenta la hipótesis inicial se tiene que

$$\Theta(f) = f(\Theta(x_1), \ \ldots \ , \Theta(x_n)) = f(\phi_1, \ \ldots \ , \phi_n) \equiv 0 \mod J_q.$$

De donde se deduce que:

$$f(x_1, \ \ldots \ , x_n) = \underbrace{f(\phi_1, \ \ldots \ , \phi_n)}_{J_q \subseteq I_A} + \underbrace{\sum_{i=1}^{n} C_i(x_i - \phi_i)}_{I_A} \in I_A.$$

□

Definición 3.6. *Sea $\succ$ un orden monomial en $\mathbb{K}[\mathbf{x},\mathbf{y}]$ y $\mathbf{w}\in\mathbb{R}^n_{\geq 0}$ un vector peso. Se dice que el orden monomial $\succ_{\mathbf{w}}$ de $\mathbb{K}[\mathbf{x},\mathbf{y}]$ es un* orden monomial adaptado al problema entero $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ *si verifica las siguientes dos condiciones:*

1. *El orden $\succ_{\mathbf{w}}$ es un orden de eliminación para las variables $\mathbf{y}$.*
2. *El orden $\succ_{\mathbf{w}}$ es un orden monomial compatible con el vector peso $\mathbf{w}\in\mathbb{R}^n_{\geq 0}$, es decir, para cualquier vector $\mathbf{u},\ \mathbf{v}\in\mathbb{Z}^n_q$ tales que $\Theta(\mathbf{x}^{\blacktriangle\mathbf{u}})\equiv\Theta(\mathbf{x}^{\blacktriangle\mathbf{v}})\mod J_q$, es decir, $\mathbf{y}^{\blacktriangle(A\mathbf{u}^t)}\equiv\mathbf{y}^{\blacktriangle(A\mathbf{v}^t)}\mod J_q$, si $\mathbf{w}\cdot\mathbf{u}\succ\mathbf{w}\cdot\mathbf{v}$, entonces $\mathbf{x}^{\mathbf{u}}\succ_{\mathbf{w}}\mathbf{x}^{\mathbf{v}}$.*

Sea $G_{\succ_{\mathbf{w}}}$ una base de Gröbner del ideal I_A respecto del orden monomial $\succ_{\mathbf{w}}$ adaptado al problema entero modular $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$. El siguiente teorema extiende el algoritmo de Conti-Traverso al caso modular.

Teorema 3.3. *Dado el monomio $\mathbf{x}^{\blacktriangle\mathbf{b}}$ y sea $\mathrm{nf}_{G_{\succ_{\mathbf{w}}}}(\mathbf{y}^{\blacktriangle\mathbf{b}})=\mathbf{x}^{\mathbf{u}'}\in\mathbb{K}[\mathbf{x}]$ la forma normal de dicho polinomio respecto de $G_{\succ_{\mathbf{w}}}$, entonces $\blacktriangledown\mathbf{u}'$ es la solución óptima del problema $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$.*

Demostración. Véase [37], Teorema 6. □

En el caso de un vector $\mathbf{w}\in\mathbb{R}^n$ arbitrario que puede contener alguna componente con valor negativo, no podemos aplicar el método anterior ya que no podemos definir el orden monomial $\succ_{\mathbf{w}}$ adaptado al problema $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$. Sin embargo, podemos transformar el problema original $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ en un problema equivalente $\mathrm{IP}_{A',\mathbf{w}',q}(\mathbf{b}')$ donde:

- Dada la matriz $A\in\mathbb{Z}_q^{m\times n}$ consideramos A' su elevación de Lawrence. Es decir,
$$A'=\begin{pmatrix} A & 0\\ I_n & I_n\end{pmatrix}\in\mathbb{Z}^{(m+n)\times 2n},$$
Donde $I_n\in\mathbb{Z}^{n\times n}$ es la matriz identidad y $0\in\mathbb{Z}^{m\times n}$ es la matriz nula.
- Dado el vector $\mathbf{b}\in\mathbb{Z}^m$, definimos $\mathbf{b}'=(\mathbf{b},\mathbf{c})\in\mathbb{Z}_q^{m+n}$. Con
$$\mathbf{c}=(q-1,\ \ldots\ ,q-1)\in\mathbb{Z}_q^n.$$
- Dado el vector $\mathbf{w}\in\mathbb{R}^n$, definimos $\mathbf{w}'=(\mathbf{v}_1,\mathbf{v}_2)\in\mathbb{R}^{2n}_{\geq 0}$.
Con $\begin{cases}\nu=\max\{\{|w_i|:w_i<0,\ \forall i=1,\ \ldots\ ,n\}\ ,\ \{0\}\},\\ \mathbf{v}_1=(w_1+\nu,\ \ldots\ ,w_n+\nu)\in\mathbb{R}^n_{\geq 0},\ \text{ y }\ \mathbf{v}_2=(\nu,\ \ldots,\ \nu)\in\mathbb{R}^n_{\geq 0}.\end{cases}$

El siguiente teorema demuestra que resolver el problema $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ es equivalente a resolver el problema $\mathrm{IP}_{A',\mathbf{w}',q}(\mathbf{b}')$.

Teorema 3.4. *Sean* $\mathbf{u}_1,\ \mathbf{u}_2 \in \mathbb{Z}_q^n$. *Entonces,* $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{Z}_q^{2n}$ *es solución óptima del problema* $\mathrm{IP}_{A',\mathbf{w}',q}(\mathbf{b}')$ *si y sólo si* $\mathbf{u}_1 \in \mathbb{Z}_q^n$ *es solución óptima del problema* $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$.

Demostración. Sea $\mathbf{u} \in \mathbb{Z}_q^{2n}$ una solución óptima del problema $\mathrm{IP}_{A',\mathbf{w}',q}(\mathbf{b}')$. Por hipótesis $A'\mathbf{u}^t \equiv b' \mod q$, en otras palabras:

$$\begin{pmatrix} A & 0 \\ I_n & I_n \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} \equiv \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix} \mod q \Rightarrow \begin{cases} A\mathbf{u}_1^t \equiv \mathbf{b} \mod q \\ \mathbf{u}_1 + \mathbf{u}_2 \equiv \mathbf{c} \mod q \end{cases}$$

Si utilizamos la siguiente notación $\mathbf{u}_1 = (u^1, \ldots, u^n)$ y $\mathbf{u}_2 = (u^{n+1}, \ldots, u^{2n})$, entonces la condición $\mathbf{u}_1 + \mathbf{u}_2 \equiv \mathbf{c} \mod q$ equivale a decir que

$$u^i + u^{n+i} \equiv q - 1 \mod q \text{ para todo } i \in \{1, \ \ldots \ , n\}.$$

Entonces tenemos que

$$\begin{aligned} \mathbf{w}' \cdot \blacktriangle\mathbf{u} &= \sum_{i=1}^{n} \mathbf{v}_1^i \cdot \blacktriangle\mathbf{u}_1^i + \sum_{i=1}^{n} \mathbf{v}_2^i \cdot \blacktriangle\mathbf{u}_2^i \\ &= \sum_{i=1}^{n} (w_i + \nu) \cdot \blacktriangle u^i + \sum_{i=1}^{n} \nu \cdot \blacktriangle u^{n+i} \\ &= \sum_{i=1}^{n} w_i \cdot \blacktriangle u^i + \nu \sum_{i=1}^{2n} (\blacktriangle u^i + \blacktriangle u^{n+i}) \\ &= \mathbf{w} \cdot \blacktriangle\mathbf{u}_1 + \nu \cdot n \cdot (q-1) \end{aligned}$$

Como $\nu \cdot n \cdot (q-1)$ es un valor constante independiente del valor $\mathbf{u}$, se tiene que $\mathbf{u}$ minimiza el producto $\mathbf{w}' \cdot \blacktriangle\mathbf{u}$ si y sólo si $\mathbf{u}_1$ minimiza $\mathbf{w} \cdot \blacktriangle\mathbf{u}_1$. □

El Teorema 3.3 sugiere el Algoritmo 9 para obtener una solución óptima del problema $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$.

3.2.2. Reducción del número de variables

El algoritmo de Ikegami-Kaji utiliza $m \times n$ variables para describir el ideal de eliminación $I_A \cap \mathbb{K}[\mathbf{x}]$. Sin embargo, podemos utilizar la filosofía expuesta por Di Biase y Urbanke en [26] para reducir el número de variables definiendo un ideal directamente en $\mathbb{K}[\mathbf{x}]$, lo que disminuiría el número de variables a n.

Algoritmo 9 Algoritmo de Conti-Traverso extendido

INPUT: $A \in \mathbb{Z}_q^{m\times n}$, $\mathbf{b} \in \mathbb{Z}_q^m$, $\mathbf{w} \in \mathbb{R}^n$ con $q \geq 2$.
OUTPUT: Una solución óptima del problema $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$.
Calcular una base de Gröbner $G_{\succ_\mathbf{w}}$ del ideal I_A respecto del orden monomial adaptado $\succ_\mathbf{w}$.
Calcular la forma normal del monomio $\mathbf{y}^{\blacktriangle\mathbf{b}}$ respecto de la base de Gröbner $G_{\succ_\mathbf{w}}$, $\mathrm{nf}_{G_{\succ_\mathbf{w}}}(\mathbf{y}^{\blacktriangle\mathbf{b}}) = \mathbf{x}^{\mathbf{u}'}$.
Devolver $\blacktriangledown\mathbf{u}' \in \mathbb{Z}_q^n$.

En el artículo [26] Di Biase y Urbanke trabajan en el espacio $\mathbb{K}[\mathbf{x}]$ al calcular el $\mathbb{Z}$-núcleo relacionado con el problema lineal entero $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$, reduciendo así el número de variables. A lo largo de esta sección mostraremos cómo calcular el $\mathbb{Z}_q$-núcleo de la matriz A relacionado con el problema entero modular $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ realizando sólo operaciones en $\mathbb{K}[\mathbf{x}]$.
En realidad esta sección se trata de una generalización al caso modular no binario del artículo [15].
Dada la matriz de coeficientes $A \in \mathbb{Z}_q^{m\times n}$, definimos el siguiente ideal:

$$I(A) = \langle\{\mathbf{x}^\mathbf{a} - \mathbf{x}^\mathbf{b} \mid A\blacktriangledown(\mathbf{a}-\mathbf{b})^t \equiv 0 \mod q\}\rangle.$$

Es decir el ideal $I(A)$ está formado por todos los vectores

$$\blacktriangledown(\mathbf{a}-\mathbf{b}) \in \ker_{\mathbb{Z}_q}(A).$$

Definimos la matriz $A^\perp$ como la matriz cuyas filas generan el siguiente subespacio vectorial:

$$\{\mathbf{u} \in \mathbb{Z}_q^n \mid \mathbf{u}\cdot a \equiv 0 \mod q \ , \forall a \text{ fila de la matriz } A\}.$$

La siguiente proposición es una generalización para el caso no modular de la Proposición 1 de [15].

Proposición 3.3. *Sean $\{\mathbf{w}_1,\ldots,\mathbf{w}_k\} \subseteq \mathbb{Z}_q^n$ un conjunto de generadores del espacio vectorial generado por las filas de la matriz $A \in \mathbb{Z}_q^{m\times n}$ y consideremos los vectores $\mathbf{a},\ \mathbf{b} \in \mathbb{Z}^n$. Entonces las siguientes condiciones son equivalentes:*

1. $A^\perp\blacktriangledown\mathbf{a}^t \equiv A^\perp\blacktriangledown\mathbf{b}^t \mod q$.
2. $\mathbf{x}^\mathbf{a} - \mathbf{x}^\mathbf{b} \in I(A^\perp)$.

3. $\exists \mathbf{t}_1, \mathbf{t}_2 \in \mathbb{K}[\mathbf{x}]$ *y* $\lambda_1, \ldots, \lambda_k \in \mathbb{Z}^n$ *tales que:*

$$\mathbf{x}^{\mathbf{a}+(q-1)\mathbf{b}}\mathbf{t}_1^q = t_2^q \prod_{j=1}^{s} \mathbf{x}^{\lambda_i \blacktriangle \mathbf{w}_j}.$$

Demostración. En primer lugar veremos la equivalencia entre las condiciones 1 y 2. Por el Lema 3.4 y siguiendo con la notación de la sección anterior, sabemos que $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}}$ pertenece al ideal $I(A^\perp)$ si y sólo si $\Theta(\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}}) \equiv 0$ mód J_q. Es decir si y sólo si $\mathbf{y}^{(\blacktriangle A)^\perp \mathbf{a}^t} \equiv \mathbf{y}^{(\blacktriangle A)^\perp \mathbf{b}^t} \mod J_q$ que, por el Lema 3.3, es equivalente a que $A^\perp \blacktriangledown \mathbf{a}^t \equiv A^\perp \blacktriangledown \mathbf{b}^t \mod q$.
Ahora definimos el homomorfismo Φ desde $\mathbb{K}[\mathbf{x}]$ a $\mathbb{Z}_q^n$ como:

$$\begin{array}{rccc} \Phi: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{Z}_q^n \\ & \mathbf{x}^{\mathbf{a}} & \longmapsto & \Phi(\mathbf{x}^{\mathbf{a}}) = ((\blacktriangledown a_1), \ldots, (\blacktriangledown a_n)). \end{array}$$

Es fácil ver que se verifica:

i. $\Phi(\mathbf{x}^{\mathbf{a}}) = \Phi(\mathbf{x}^{\mathbf{b}}) \iff \exists \mathbf{t}_1,\ \mathbf{t}_2 \in \mathbb{K}[\mathbf{x}]$ tal que $\mathbf{t}_1^q \mathbf{x}^{\mathbf{a}} = \mathbf{t}_2^q \mathbf{x}^{\mathbf{b}}$.

ii. $\Phi(\mathbf{x}^{\mathbf{a}}) - \Phi(\mathbf{x}^{\mathbf{b}}) = \Phi(\mathbf{x}^{\mathbf{a}}) + (q-1)\Phi(\mathbf{x}^{\mathbf{b}}) = \Phi(\mathbf{x}^{\mathbf{a}+(q-1)\mathbf{b}})$.

iii. Además $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}} \in I(A^\perp)$ si y sólo si $\Phi(\mathbf{x}^{\mathbf{a}}) - \Phi(\mathbf{x}^{\mathbf{b}})$ pertenece al subespacio vectorial generado por el conjunto de vectores $\{\mathbf{w}_1, \ \ldots\ , \mathbf{w}_k\}$, ya que:

$$\begin{array}{rcl} \mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}} \in I(A^\perp) & \Leftrightarrow & A^\perp \blacktriangledown (\mathbf{a}-\mathbf{b})^t \equiv 0 \mod q \\ & \Leftrightarrow & \blacktriangledown(\mathbf{a}-\mathbf{b}) \in \langle \{\mathbf{w}_1, \ \ldots\ , \mathbf{w}_k\} \rangle. \end{array}$$

$2 \Rightarrow 3$ Si $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}} \in I(A^\perp)$, por la propiedad *iii* que acabamos de ver, sabemos que existen $\lambda_{q,1}, \ldots, \lambda_{q,k} \in \mathbb{Z}_q^n$ tales que:

$$\Phi(\mathbf{x}^{\mathbf{a}}) - \Phi(\mathbf{x}^{\mathbf{b}}) = \Phi(\mathbf{x}^{\mathbf{a}+(q-1)\mathbf{b}}) = \sum_{i=1}^{k} \lambda_{i,q} \mathbf{w}_i = \Phi(\prod_{i=1}^{k} \mathbf{x}^{\blacktriangle \lambda_{q,i} \blacktriangle \mathbf{w}_i}).$$

Consideramos $\lambda_{q,i} = \lambda_i$ para todo $i = 1, \ \ldots\ , k$, por la propiedad *i* sabemos que existen $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{K}[\mathbf{x}]$ tales que:

$$\mathbf{x}^{\mathbf{a}+(q-1)\mathbf{b}}\mathbf{t}_1^q = t_2^q \prod_{j=1}^{s} \mathbf{x}^{\lambda_i \blacktriangle \mathbf{w}_j}.$$

$2 \Leftarrow 3$ Supongamos que existen $\mathbf{t}_1,\ \mathbf{t}_2 \in \mathbb{K}[\mathbf{x}]$ y $\lambda_1, \ldots, \lambda_k \in \mathbb{Z}^n$ tales que

$$\mathbf{x}^{\mathbf{a}+(q-1)\mathbf{b}}\mathbf{t}_1^q = t_2^q \prod_{j=1}^{s} \mathbf{x}^{\lambda_i \blacktriangle \mathbf{w}_j}.$$

Entonces tenemos que:

$$\Phi(\mathbf{x}^{\mathbf{a}}) - \Phi(\mathbf{x}^{\mathbf{b}}) = \Phi(\mathbf{x}^{\mathbf{a}+(q-1)\mathbf{b}}) = \Phi(\prod_{i=1}^{k} \mathbf{x}^{\lambda_i \blacktriangle \mathbf{w}_i}) = \sum_{i=1}^{k} \blacktriangledown \lambda_i \mathbf{w}_i.$$

De donde podemos concluir que el vector $\Phi(\mathbf{x}^{\mathbf{a}}) - \Phi(\mathbf{x}^{\mathbf{b}})$ es combinación lineal del conjunto $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$, es decir, $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}} \in I(A^{\perp})$.

□

Sean $\{\mathbf{w}_1,\ \ldots\ , \mathbf{w}_k\} \subseteq \mathbb{Z}_q^n$ un conjunto de generadores del espacio vectorial generado por las filas de la matriz $A \in \mathbb{Z}_q^{m \times n}$, definimos el siguiente ideal:

$$\blacktriangle I = \langle \{\mathbf{x}^{\blacktriangle \mathbf{w}_1} - 1,\ \ldots\ , \mathbf{x}^{\blacktriangle \mathbf{w}_k} - 1\} \cup \{x_i^q - 1\}_{i=1}^n \rangle \subseteq \mathbb{K}[\mathbf{x}].$$

Teorema 3.5. $\blacktriangle I = I(A^{\perp}) = I_A \cap \mathbb{K}[\mathbf{x}]$.

Demostración. Es claro que $\blacktriangle I \subseteq I(A^{\perp})$ pues todos los binomios de conjunto de generadores del ideal $\blacktriangle I$ pertenecen al ideal $I(A^{\perp})$.
Por lo tanto para probar la igualdad $\blacktriangle I = I(A^{\perp})$ bastaría con probar que todo binomio $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}} \in I(A^{\perp})$ pertenece también al ideal $\blacktriangle I$. La Proposición 3.3 nos indica la existencia de $\mathbf{t}_1,\ \mathbf{t}_2 \in \mathbb{K}[\mathbf{x}]$ y de $\lambda_1, \ldots, \lambda_k \in \mathbb{Z}^n$ tales que:

$$\mathbf{x}^{\mathbf{a}+(q-1)\mathbf{b}}\mathbf{t}_1^q = \mathbf{t}_2^q \prod_{j=1}^{s} \mathbf{x}^{\lambda_j \blacktriangle \mathbf{w}_j}.$$

Además observamos que si $\mathbf{z}_1 - 1,\ \mathbf{z}_2 - 1 \in \blacktriangle I$, se tiene que

$$\mathbf{z}_1\mathbf{z}_2 - 1 = (\mathbf{z}_1 - 1)\mathbf{z}_2 + (\mathbf{z}_2 - 1) \in \blacktriangle I.$$

Como $\mathbf{x}^{\blacktriangle \mathbf{w}_j} - 1 \in \blacktriangle I$ para todo $j = \{1,\ \ldots\ , k\}$, entonces para un subconjunto finito $\Lambda \subseteq \{1,\ \ldots\ , k\}$ se tiene que $\prod_{j \in \Lambda} \mathbf{x}^{\blacktriangle \mathbf{w}_j} - 1 \in \blacktriangle I$.
Sean $\lambda_1, \ldots, \lambda_k \in \mathbb{Z}^n$, entonces

$$\prod_{j=1}^{k} \mathbf{x}^{\lambda_j \blacktriangle \mathbf{w}_j} - 1 = (\prod_{j=1}^{k} \mathbf{x}^{\blacktriangle \mathbf{w}_j} - 1) \prod_{j|\lambda_j > 0} \mathbf{x}^{(\lambda_j - 1)\blacktriangle \mathbf{w}_j} + (\prod_{j|\lambda_j > 0} \mathbf{x}^{(\lambda_j - 1)\blacktriangle \mathbf{w}_j} - 1)$$

$$= \ \ldots\ = (\prod_{j=1}^{k} \mathbf{x}^{\blacktriangle \mathbf{w}_j} - 1) \prod_{j|\lambda_j > 0} \mathbf{x}^{(\lambda_j - 1)\blacktriangle \mathbf{w}_j} + \ \ldots\ + (\prod_{j|\lambda_j > r} \mathbf{x}^{\blacktriangle \mathbf{w}_j} - 1) \in \blacktriangle I,$$

donde $r = \max_{i\in\{1,\ \ldots\ ,k\}} |\lambda_i|$.
Y por lo tanto se puede deducir que:

$$\mathbf{t}_1^q \mathbf{x}^{\mathbf{a}+(q-1)\mathbf{b}} - 1 = \mathbf{t}_2^q \prod_{j=1}^{k} \mathbf{x}^{\lambda_j \blacktriangle \mathbf{w}_j} - 1 \in \blacktriangle I.$$

Luego

$$\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}} = \mathbf{x}^{\mathbf{b}}(\mathbf{x}^{\mathbf{a}+(q-1)\mathbf{b}-1}) - \mathbf{x}^{\mathbf{a}}(\mathbf{x}^{q\mathbf{b}} - 1) \in \blacktriangle I.$$

De donde podemos concluir que $I(A^\perp) = \blacktriangle I$.
Veamos ahora que $\blacktriangle I = I_A \cap \mathbb{K}[\mathbf{x}]$.
En primer lugar observamos que $\Theta(\mathbf{x}^{\blacktriangle w_i} - 1) = 0$, pues $AA^\perp = 0$ y que

$$\begin{aligned} \Theta(x_i^q - 1) = \Theta(\mathbf{x}^{qe_i} - 1) &= \mathbf{y}^{(\blacktriangle A^\perp)(qe_i)^t} - 1 \\ &\equiv \sum_{j | a_{ij}^\perp \neq 0} B_j(y_j^q - 1) \equiv 0 \mod J_q \end{aligned}$$

Donde $(\blacktriangle A^\perp) = (a_{ij}^\perp)_{\substack{j=1,\ldots,n-k \\ i=1,\ldots,n}} \in \mathbb{Z}^{(n-k)\times n}$, $B_j \in \mathbb{K}[\mathbf{y}]$ y e_i representa al vector i-ésimo de la base estándar de $\mathbb{Z}^n$.
Por lo tanto, por el Lema 3.4, se tiene que todo binomio del conjunto de generadores del ideal $\blacktriangle I$ pertenece al ideal $I_A \cap \mathbb{K}[\mathbf{x}]$.
Recíprocamente, como el ideal de eliminación $I_A \cap \mathbb{K}[\mathbf{x}]$ es un ideal binomial, entonces está generado por binomios. Sea $f = \mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} \in I_A \cap \mathbb{K}[\mathbf{x}]$ con $\mathbf{u},\ \mathbf{v} \in \mathbb{Z}_{\geq 0}^n$. Por el Lema 3.4 sabemos que

$$\Theta(f) = \mathbf{y}^{(\blacktriangle A^\perp)\mathbf{u}^t} - \mathbf{y}^{(\blacktriangle A^\perp)\mathbf{v}^t} \equiv 0 \mod J_q.$$

Lo que implica, por el Lema 3.3, que $A^\perp \blacktriangledown \mathbf{u}^t \equiv A^\perp \blacktriangledown \mathbf{v}^t \mod q$. Teniendo en cuenta la proposición 3.3 podemos concluir que $f \in I(A^\perp) = \blacktriangle I$.

□

Observemos que la matriz $A^\perp$ coincide con la matriz no negativa que buscan Di Biase y Urbanke en [26]. Por lo tanto este teorema puede ser visto como una generalización de lo establecido en dicho artículo para eliminar las variables $\mathbf{y}$ del ideal de eliminación que definen Ikegami y Kaji.

3.2.3. Utilización de técnicas FGLM

Relación con la eliminación Gaussiana

Consideramos el siguiente sistema de ecuaciones lineales:

$$\begin{pmatrix} 2 & -3 & 4 & 0 \\ 1 & 4 & 0 & 3 \\ -1 & -15 & 4 & -9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.1}$$

Podemos definir a partir de este sistema de ecuaciones el ideal I definido por:

$$I = \langle 3x - 6y - 2z, 2x - 4y + 4w, x - 2y - z - w \rangle \subseteq \mathbb{K}[x, y, z, w].$$

Aplicamos la eliminación gaussiana al sistema 3.1 obteniendo el siguiente resultado:

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

Fijados como orden monomial el orden lexicográfico y como orden de las variables $x > y > z > w$, si calculamos una base de Gröbner reducida G del ideal I, obtenemos el mismo sistema de generadores que el que nos aporta la eliminación gaussiana.

$$I = \langle x - 2y - z - w, z + 3w \rangle.$$

Con este sencillo ejemplo se pueden comprobar las semejanzas entre el cálculo de bases de Gröbner utilizando el orden lexicográfico y la eliminación gaussiana.

Técnicas FGLM

En esta sección presentamos un algoritmo para calcular la base de Gröbner reducida del ideal asociado a un problema de programación lineal entera utilizando técnicas algebraicas. Este algoritmo fue presentado para el caso binario en [15], pero su extensión al caso general módulo q es directa. Para estudiar en profundidad estas técnicas dirigimos al lector a [28] y [29].
El concepto de módulo sobre un anillo R es una generalización de la noción de espacio vectorial sobre cuerpos.

Definición 3.7. *Sea $(R, \oplus, \otimes)$ un anillo conmutativo. Un módulo M sobre un anillo R, que escribimos como un R-módulo, consiste en un grupo abeliano $(M, +)$ y una multiplicación escalar $\cdot : R \times M \longrightarrow M$ satisfaciendo las siguientes propiedades:*

1. $\forall \mathbf{r},\ \mathbf{s} \in R,\ \forall \mathbf{x} \in M \mid \mathbf{r} \cdot (\mathbf{s} \cdot \mathbf{x}) = (\mathbf{r} \otimes \mathbf{s}) \cdot \mathbf{x}$.
2. $\forall \mathbf{r},\ \mathbf{s} \in R,\ \forall \mathbf{x} \in M \mid (\mathbf{r} \oplus \mathbf{s}) \cdot \mathbf{x} = \mathbf{r} \cdot \mathbf{x} + \mathbf{s} \cdot \mathbf{x}$.
3. $\forall \mathbf{r} \in R,\ \forall \mathbf{x},\ \mathbf{y} \in M \mid \mathbf{r} \cdot (\mathbf{x} + \mathbf{y}) = \mathbf{r} \cdot \mathbf{x} + \mathbf{r} \cdot \mathbf{y}$.
4. $\forall \mathbf{x} \in M \mid 1_R \cdot \mathbf{x} = \mathbf{x}$, *donde 1_R denota al elemento neutro del anillo R.*

Si R es un anillo arbitrario y $n \in \mathbb{N}$, entonces el producto cartesiano R^n es un módulo sobre R utilizando las operaciones componente a componente. Otros ejemplos de R-módulos se pueden obtener considerando submódulos de R^n, es decir subconjuntos de R^n cerrados para la suma y la multiplicación escalar por elementos de R.
De esta forma si consideramos el anillo de polinomios $\mathbb{K}[\mathbf{x}]$ y un subconjunto finito de vectores $\{f_1, \ldots, f_s\} \subseteq \mathbb{K}[\mathbf{x}]^m$ entonces:

$$\langle f_1,\ \ldots\ , f_s \rangle = \langle \{\lambda_1 \cdot f_1 + \ \ldots\ + \lambda_s \cdot f_s \in \mathbb{K}[\mathbf{x}]^m,\ \text{ con }\ \lambda_1, \ldots, \lambda_s \in \mathbb{K}[\mathbf{x}]\} \rangle$$

es un $\mathbb{K}[\mathbf{x}]$-módulo.

Proposición 3.4. *Sea M un módulo sobre un anillo R. Diremos que un subconjunto $F \subseteq M$ es una base libre del módulo M si todo elemento f de M se puede escribir de forma única como una combinación lineal de los elementos de F.*

Demostración. La demostración es análoga a la que se realiza para demostrar la proposición equivalente de bases de espacios vectoriales. □

Definición 3.8. *Sea M un módulo sobre un anillo R, diremos que M es un módulo libre si M tiene una base libre, es decir, existe un conjunto generador R-linealmente independiente de M.*

El R-módulo $M = R^m$ es un módulo libre. Los elementos:

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad , \ \ldots\ , \quad e_m = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

forman una base libre de M como R-módulo que denominamos *base estándar* de M.

Proposición 3.5. *Sea $(f_1, \ldots, f_s)$ una s-upla ordenada de elementos de M. El conjunto de todas las s-uplas $(\lambda_1, \ldots, \lambda_s)^t \in R^s$ tales que*

$$\lambda_1 \cdot f_1 + \ \ldots \ + \lambda_s \cdot f_s = 0$$

es un R-módulo de R^s llamado módulo de sizigias de $(f_1, \ldots, f_s)$ y denotado por $\mathrm{Syz}(f_1, \ldots, f_s)$.

Demostración. Sean $(\alpha_1, \ldots, \alpha_s)^t$, $(\beta_1, \ldots, \beta_s)^t \in \mathrm{Syz}(f_1, \ldots, f_s)$ y sea $\gamma \in R$, entonces:

$$\begin{aligned} \alpha_1 \cdot f_1 + \ldots + \alpha_s \cdot f_s = 0 \\ \beta_1 \cdot f_1 + \ldots + \beta_s \cdot f_s = 0 \end{aligned}$$

Utilizando las propiedades distributivas de módulo, se obtiene que

$$(\gamma \cdot \alpha_1 + \beta_1) \cdot f_1 + \ldots + (\gamma \cdot \alpha_s + \beta_s) \cdot f_s = 0.$$

Lo que prueba que $(\gamma \cdot \alpha_1 + \beta_1, \ldots, \gamma \cdot \alpha_s + \beta_s)^t \in \mathrm{Syz}(f_1, \ldots, f_s)$. Luego tenemos que $\mathrm{Syz}(f_1, \ldots, f_s)$ es un submódulo de R^s. □

Ejemplo 3.1. *Sea $F = (x, x^2 + z, y + z) \subseteq \mathbb{K}[x, y, z]^3$ y fijamos como orden monomial el orden lexicográfico y como orden de las variables $x > y > z$. Observamos que $S = (-x + y, 1, -x) \in \mathbb{K}[\mathbf{x}]^3$ es una sizigia en los términos líderes de F, pues:*

$$(-x+y)\cdot \mathrm{LT}_{\succ_{lex}}(x)+1\cdot \mathrm{LT}_{\succ_{lex}}(x^2+z)+(-x)\cdot \mathrm{LT}_{\succ_{lex}}(y+z) = x^2+yx+x^2-xy = 0.$$

Definición 3.9. *Sea M un R-módulo generado por el conjunto $\{f_1, \ldots, f_s\}$, definimos matriz representativa de M como una matriz cuyas columnas generan* $\mathrm{Syz}(f_1, \ldots, f_s) \subseteq R^s$.

A partir de ahora sea $\mathbb{K}$ un cuerpo arbitrario finito, $\mathbb{K}[x_1, \ldots, x_n] := \mathbb{K}[\mathbf{x}]$ el anillo de polinomios en n variables. Denotamos por $\mathbb{T}_1$ como el conjunto de monomios en $\mathbb{K}[\mathbf{x}]$. Recordemos que un monomio en $\mathbb{K}[\mathbf{x}]$ es un producto de la forma $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ que identificamos con los vectores $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$.
Un monomio $\mathbf{m}$ en el $\mathbb{K}[\mathbf{x}]$-módulo libre $\mathbb{K}[\mathbf{x}]^r$ es un elemento de la forma $\mathbf{x}^\alpha \cdot e_i$ para algún $i \in \{1, \ldots, r\}$, donde e_i representa un vector de la base estándar de $\mathbb{K}[\mathbf{x}]^r$. Denotaremos por $\mathbb{T}_r$ como el conjunto de monomios en $\mathbb{K}[\mathbf{x}]^r$.

Siguiendo esta notación, todo elemento $f \in \mathbb{K}[\mathbf{x}]^r$ se puede escribir de forma única como una combinación lineal de monomios $\mathbf{m}_i \in \mathbb{T}_r$:

$$f = \sum_{i=1}^{r} c_i \cdot \mathbf{m}_i \quad \text{con} \quad c_i \in \mathbb{K}[\mathbf{x}] \quad \text{y} \quad \mathbf{m}_i \in \mathbb{T}_r.$$

Sean $\mathbf{m} = \mathbf{x}^\alpha \cdot e_i$, $\mathbf{n} = \mathbf{x}^\beta \cdot e_j$ monomios en $\mathbb{K}[\mathbf{x}]^r$. Diremos que $\mathbf{n}$ divide a $\mathbf{m}$ si y sólo si $i = j$ y $\mathbf{x}^\beta$ divide a $\mathbf{x}^\alpha$.
Además, si $i = j$ definimos máximo común divisor entre $\mathbf{m}$ y $\mathbf{n}$ como el máximo común divisor entre $\mathbf{x}^\alpha$ y $\mathbf{x}^\beta$ multiplicado por e_i. En caso contrario, si $i \neq j$ entonces el máximo común divisor es cero.

Definición 3.10. *Diremos que un submódulo $M \subseteq \mathbb{K}[\mathbf{x}]^r$ es un submódulo monomial si M está generado por un conjunto de monomios.*

Los submódulos monomiales tienen propiedades análogas a las de los ideales monomiales. Por ejemplo, al igual que ocurría en los ideales monomiales, sea $M = \langle \mathbf{m}_1, \ldots, \mathbf{m}_t \rangle$ con $\mathbf{m}_i \in \mathbb{T}_r$ un submódulo monomial y f un elemento arbitrario de $\mathbb{K}[\mathbf{x}]^r$, entonces, $f \in M$ si y sólo si todo término de f es divisible por algún $\mathbf{m}_i$.
Para extender la teoría de bases de Gröbner sobre anillos a los módulos nos falta definir un orden en los monomios de $\mathbb{K}[\mathbf{x}]^r$, construir un algoritmo de la división en los elementos de $\mathbb{K}[\mathbf{x}]^r$ y extender el algoritmo de Buchberger.

Definición 3.11. *Un orden monomial sobre $\mathbb{K}[\mathbf{x}]^r$ es una relación $\succ$ sobre el conjunto de monomios $\mathbb{T}_r$ que verifica:*

1. *$\succ$ es un orden total.*
2. *Si $\mathbf{m} \succ \mathbf{n}$, entonces $\mathbf{x}^\alpha \cdot \mathbf{m} \succ \mathbf{x}^\alpha \cdot \mathbf{n}$, para todo $\mathbf{m}$, $\mathbf{n} \in \mathbb{T}_r$ y todo $\mathbf{x}^\alpha \in \mathbb{T}_1$.*
3. *$\succ$ es un buen orden, es decir, $\mathbf{x}^\alpha \cdot \mathbf{m} \succ \mathbf{m}$, para todo $\mathbf{m} \in \mathbb{T}_r$ y todo $\mathbf{x}^\alpha \in \mathbb{T}_1 \setminus \{1\}$.*

La mayoría de los órdenes monomiales de $\mathbb{K}[\mathbf{x}]^r$ se trata de extensiones de órdenes monomiales sobre $\mathbb{K}[\mathbf{x}]$. Tenemos dos opciones para obtener estos órdenes, dar prioridad a los coeficientes, o bien dar prioridad a la posición del vector. Es decir, si fijamos $\succ$ un orden en el conjunto de monomios $\{\mathbf{x}^\alpha : \alpha \in \mathbb{N}^n\}$ y $<$ un orden en los enteros, podemos definir los siguientes órdenes sobre los monomios de $K[\mathbf{x}]^r$:

- Orden TOP (*Term Over Position*), que denota el orden monomial sobre $\mathbb{K}[\mathbf{x}]^r$ que da prioridad a los coeficientes, es decir:

$$\mathbf{x}^\alpha \cdot e_i \succ_{TOP} \mathbf{x}^\beta \cdot e_j \Leftrightarrow \mathbf{x}^\alpha \succ \mathbf{x}^\beta \text{ ó } \mathbf{x}^\alpha = \mathbf{x}^\beta \text{ y } i < j.$$

- Orden POT (*Position Over Term*), que denota el orden monomial sobre $\mathbb{K}[\mathbf{x}]^r$ que da prioridad a la posición del vector, es decir:

$$\mathbf{x}^\alpha \cdot e_i \succ_{POT} \mathbf{x}^\beta \cdot e_j \Leftrightarrow i < j \text{ ó } i = j \text{ y } \mathbf{x}^\alpha \succ \mathbf{x}^\beta.$$

Definición 3.12. *Fijado un orden monomial $\succ$ en $\mathbb{K}[\mathbf{x}]$ y dada una r-upla de monomios $\mathbf{w} = (w_1, \ldots, w_r) \subseteq \mathbb{K}[\mathbf{x}]^r$ con $w_i \in \mathbb{T}_1$. Definimos un orden POT en $\mathbb{K}[\mathbf{x}]^r$ inducido por $\succ$ y por $\mathbf{w}$ y lo denotamos por $\succ_{\mathbf{w}}$ como:*

$$\mathbf{x}^\alpha \cdot e_i \succ_{\mathbf{w}} \mathbf{x}^\beta \cdot e_j \Leftrightarrow w_i \cdot \mathbf{x}^\alpha \succ w_j \cdot \mathbf{x}^\beta \quad ó \quad w_i \cdot \mathbf{x}^\alpha = w_j \cdot \mathbf{x}^\beta \quad y \quad i < j.$$

El siguiente teorema nos proporciona un resultado que extiende la división euclídea de polinomios en una sola variable a s-uplas de polinomios en varias variables.

Teorema 3.6 (Algoritmo de la división)**.**
Fijado un orden monomial $\succ$ en $\mathbb{K}[\mathbf{x}]^r$ y dada una s-upla ordenada $F = (f_1, \ldots, f_s)$ de elementos de $\mathbb{K}[\mathbf{x}]^r$. Todo polinomio $f \in \mathbb{K}[\mathbf{x}]^r$ se puede escribir de forma única como:

$$f = \lambda_1 \cdot f_1 + \cdots + \lambda_s \cdot f_s + \mathbf{r}$$

donde $\lambda_i \in \mathbb{K}[\mathbf{x}]$, $\mathbf{r} \in \mathbb{K}[\mathbf{x}]^r$ y $\mathrm{LT}(\lambda_i \cdot f_i) \succ \mathrm{LT}(f)$ para todo $i \in \{1, \ldots, s\}$ y si $\mathbf{r} \neq 0$, entonces $\mathbf{r}$ es una combinación lineal de monomios no divisibles por los monomios $\{\mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s)\}$.
Al elemento $\mathbf{r} \in \mathbb{K}[\mathbf{x}]^r$ se le denomina resto de la división de f por F.

Demostración. Véase [25]. Capítulo 5.2, Teorema 2.5. □

Definición 3.13. *Sea M un submódulo de $\mathbb{K}[\mathbf{x}]^r$ y fijamos un orden monomial $\succ$ en $\mathbb{K}[\mathbf{x}]^r$ denotamos por $\langle \mathrm{LT}(M) \rangle$ al submódulo generado por el conjunto de monomios $\{\mathrm{LT}(f) \mid f \in M\}$.*
A una colección finita de elementos de M, $G = \{g_1, \ldots, g_s\} \subseteq M$ se denomina base de Gröbner de M si $\langle \mathrm{LT}(M) \rangle = \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) \rangle$.

La siguiente proposición y el siguiente teorema nos aportan un criterio para determinar si un elemento de $\mathbb{K}[\mathbf{x}]^r$ pertenece a un submódulo $M \subseteq \mathbb{K}[\mathbf{x}]^r$. En ambos supuestos la demostración es análoga al caso de ideales sobre un anillo de polinomios.

Proposición 3.6. *Sea G una base de Gröbner de un submódulo $M \subseteq \mathbb{K}[\mathbf{x}]^r$ y sea $f \in \mathbb{K}[\mathbf{x}]^r$.*

1. *$f \in M$ si y sólo si el resto de la división de f por G es cero.*
2. *G genera a M como módulo, es decir, $M = \langle G \rangle$.*

Definición 3.14. *Fijamos un orden monomial $\succ$ en $\mathbb{K}[\mathbf{x}]^r$ y sean $f,\ g \in \mathbb{K}[\mathbf{x}]^r$. Si denotamos por $\mathbf{m}$ el mínimo común múltiplo de $\mathrm{LT}(f)$ y $\mathrm{LT}(g)$, definimos la r-upla de polinomios S de f y g como:*

$$S(f,g) = \frac{\mathbf{m}}{\mathrm{LT}(f)} \cdot f - \frac{\mathbf{m}}{\mathrm{LT}(g)} \cdot g.$$

Teorema 3.7 (Criterio de Buchberger)**.**
Sea M un submódulo de $\mathbb{K}[\mathbf{x}]^r$. Un sistema de generadores $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{K}[\mathbf{x}]^r$ del módulo M es una base de Gröbner de M si y sólo si para cualesquiera $i,\ j \in \{a, \ldots, s\}$ se verifica que el resto de la división del elemento $S(g_i, g_j)$ por G es cero.

Este Teorema sugiere el Algoritmo 10 para calcular una base de Gröbner de un submódulo $M \subseteq \mathbb{K}[\mathbf{x}]^r$ a partir de un sistema de generadores de M en un número finito de etapas.

Algoritmo 10 Algoritmo de Buchberger para submódulos

INPUT: $F = \{f_1,\ \ldots\ , f_s\} \subseteq \mathbb{K}[\mathbf{x}]^r$ y un orden monomial $\succ$ en $\mathbb{K}[\mathbf{x}]^r$.
OUTPUT: Una base de Gröbner $G = \{g_1,\ \ldots\ , g_t\}$ del submódulo $M = \langle F \rangle \subseteq \mathbb{K}[\mathbf{x}]^r$ respecto de $\succ$.

```
G := F
𝒢 := {{f_i, f_j}, f_i ≠ f_j ∈ G}
while 𝒢 ≠ ∅ do
  Elegir {f, g} ∈ 𝒢
  𝒢 := 𝒢 \ {{f, g}}
  Sea h el resto de la división entre S(f, g) y G.
  if h ≠ 0 then
    𝒢 := 𝒢 ∩ {{u, h}, ∀u ∈ G}
    G := G ∩ {h}
  end if
end while
Devolver G.
```

A continuación exponemos la generalización para submódulos del algoritmo FGLM descrita en [28].

Consideramos el submódulo $N \subseteq \mathbb{K}[\mathbf{x}]^r$. Sea G_1 una base de Gröbner de N respecto del orden $\succ_1$ y sea $\succ_2$ otro orden en $\mathbb{K}[\mathbf{x}]^r$. El algoritmo que vamos a explicar construye una nueva base G_2 de N respecto del orden $\succ_2$.
Durante el Algoritmo 11 utilizaremos las siguientes funciones:

- La función $\mathrm{nf}_{G_1}(s)$ que nos devuelve la forma normal de un elemento $s \in \mathbb{K}[\mathbf{x}]^r$ respecto de la base de Gröbner G_1.
- La función `Next`(S) que en primer lugar nos devuelve el elemento inicial de la lista S y luego borra dicho elemento de la lista indicada.
- La función `Insert`(S, t) que añade a la lista S los términos

$$\{x_i \cdot t\}_{i=1,\ldots,n}.$$

- La función `Order`(S) que ordena los elementos de la lista S respecto del orden $\succ_2$.

Además definimos las siguientes listas de elementos en $\mathbb{K}[\mathbf{x}]^r$:

- La lista `NewBasis` en la que guardamos la nueva base de Gröbner construida respecto del orden $\succ_2$.
- La lista `NextTerms` que contiene los términos que hay que considerar en cada iteración.
- La lista `LeadTerms` que contiene los términos líderes de la lista `NewBasis` ordenados de forma creciente por el orden $\succ_2$.
- La lista `RedTerms` que contiene elementos del anillo $\mathbb{K}[\mathbf{x}]^r/G_1$

En primer lugar inicializamos la lista `RedTerms` con el término más pequeño respecto del orden $\succ_2$ que necesariamente es de la forma $\{1 \cdot e_k \mid 1 \leq k \leq r\}$, ya que si fuese de la forma $\mathbf{x}^\alpha \cdot e_k$ con $\alpha \in \mathbb{Z}^n$ y $k \in \{1, \ldots, r\}$, entonces se tendría que $\mathbf{x}^\alpha \cdot e_k \succ_2 1 \cdot e_k$, lo que contradice la primera propiedad de la definición de orden en $\mathbb{K}[\mathbf{x}]^r$. Además inicializamos la lista `NextTerms` con los elementos $\{1 \cdot e_k \mid 1 \leq k \leq r\}$ ordenados respecto del orden $\succ_2$.
En cada iteración escogemos el elemento $t :=$ `Next(NextTerms)`, calculamos su forma normal respecto de la base de Gröbner inicial, G_1. En el caso de que la forma normal obtenida se pueda escribir como combinación lineal de las formas normales de los elementos de `RedTerms`, entonces añadimos t a la nueva base de Gröbner G_2. En caso contrario, se añade t a la lista `RedTerms`.
El algoritmo propuesto en [15] calcula una base de Gröbner respecto de $\succ$ del ideal $I = \langle f_1, \ldots, f_s \rangle$ a partir de una base de Gröbner del módulo de

Algoritmo 11 Algoritmo FGLM generalizado

INPUT: Una base de Gröbner G_1 del submódulo $N \subseteq \mathbb{K}[\mathbf{x}]^r$ respecto del orden $\succ_1$ y otro orden $\succ_2$ en $\mathbb{K}[\mathbf{x}]^r$.

OUTPUT: Una Base de Gröbner reducida G_2 de N respecto del orden $\succ_2$

Inicializamos t con el término más pequeño en $\mathcal{T}_r$ respecto de $\succ_2$.
`RedTerms` $:= [t]$;
`NextTerms` $:=$ `Order`$([1 \cdot e_k \mid 1 \cdot e_k \neq t \text{ y } 1 \leq k \leq r])$;
`NewBasis` $:= []$;
`LeadTerms` $:= []$;
`Insert`$(\texttt{NextTerms}, t)$;
`Order`$(\texttt{NextTerms})$;
while `NextTerms` $\neq []$ **do**
  $t :=$ `Next`$(\texttt{NextTerms})$;
  if t no es un múltiplo de un elemento de la lista `LeadTerms` **then**
    if existe una combinación lineal de la forma:

$$\mathrm{nf}_{G_1}(t) = \sum_{s \in \texttt{RedTerms}} \alpha_s \cdot \mathrm{nf}_{G_1}(s) \text{ con } \alpha_s \in \mathbb{K}$$

    then
      `NewBasis` $:=$ `NewBasis` $\cup \{t - \sum_{s \in \texttt{RedTerms}} \alpha_s \cdot s\}$;
      `LeadTerms` $:=$ `LeadTerms` $\cup \{t\}$;
    else
      `RedTerms` $:=$ `RedTerms` $\cup \{t\}$;
      `Insert`$(\texttt{NextTerms}, t)$;
      `Order`$(\texttt{NextTerms})$;
    end if
  end if
end while

sizigias $\mathrm{Syz}(f_1,\ldots,f_s)$. El algoritmo se presenta en [15] para el caso binario, pero la extensión al caso general es directa.
Para ello trabajamos en el módulo libre $\mathbb{K}[\mathbf{x}]^{s+1}$ con base estándar $e_0,\ldots,e_s$. Luego consideramos el submódulo $M \subseteq \mathbb{K}[\mathbf{x}]^{s+1}$ generado por:

$$m_i = f_i \cdot e_0 + e_i = (f_i, 0, \ \ldots\ , 0, 1, 0, \ \ldots\ , 0), \quad i = 1,\ldots,s$$

Es decir m_i tiene f_i en la componente cero, un 1 en la componente i y 0 en el resto de componentes. Observamos que el submódulo M coincide con el conjunto de $s+1$-uplas $(\lambda_0,\ldots,\lambda_s) \in \mathbb{K}[\mathbf{x}]^{s+1}$ tales que

$$\lambda_0 = \sum_{i=1}^{s} \lambda_i \cdot f_i.$$

Es decir con el submódulo de sizigias $\mathrm{Syz}(-1, f_1,\ldots,f_s)$, observamos que la primera componente de cada sizigia de M se corresponde con un elemento del ideal I.
El planteamiento es el siguiente:

1. En primer lugar observamos que el conjunto:

$$\begin{aligned} f_1' &= (f_1, 1, 0, \ldots, 0) \\ f_2' &= (f_2, 0, 1, \ldots, 0) \\ \vdots &= \vdots \\ f_s' &= (f_s, 0, 0, \ldots, 1) \end{aligned}$$

   forman una base de $\mathrm{Syz}(-1, f_1,\ldots,f_s)$.

2. Es fácil probar que se trata de una base de Gröbner del módulo M respecto del orden POT inducido por $\succ$ en $\mathbb{K}[\mathbf{x}]^{s+1}$ que coincide con el orden inducido por $\succ$ y por el elemento $\mathbf{w} = (1, \mathrm{LT}_\succ(f_1),\ldots,\mathrm{LT}_\succ(f_s))$ en $\mathbb{K}[\mathbf{x}]$.

3. Utilizamos el algoritmo FGLM adaptado a submódulos recorriendo los términos de $\mathbb{K}[\mathbf{x}]^{s+1}$ para obtener una base de M respecto del orden TOP en $\mathbb{K}[\mathbf{x}]^{s+1}$ determinado por el orden en los enteros $<$ (i. e. $e_i < e_j$ si $i < j$).

4. La primera componente de la nueva base de Gröbner de M respecto del nuevo orden revela una base de Gröbner del ideal I.

Ilustramos este algoritmo en el siguiente ejemplo.

Ejemplo 3.2. *Consideramos el ideal*

$$I = \langle f_1 = x^2 + x + 1, f_2 = xy + x + 1 \rangle \subseteq \mathbb{F}_2[x, y],$$

fijamos como orden monomial el orden lexicográfico y como orden de las variables $x < y$, denotamos a dicho orden $\succ_{lex}$
Definimos el submódulo $M \subseteq \mathbb{K}[x, y]^3$ como el conjunto $\mathrm{Syz}(-1, f_1, f_2)$. *Es fácil comprobar que los elementos $\{m_1,\ m_2\}$ definidos como:*

$$m_1 = (f_1, 1, 0), \quad m_2 = (f_2, 0, 1)$$

forman una base de Gröbner del módulo M respecto del orden POT en $\mathbb{K}[x, y]^3$ inducido por el orden $\succ_{lex}$. Este orden es equivalente al orden inducido por $\succ_{lex}$ y por el elemento $\mathbf{w} = (1, \mathrm{LT}_{\succ_{lex}}(f_1), \mathrm{LT}_{\succ_{lex}}(f_2))$. Vamos a denotar a esta base de Gröbner $G_1(M)$.
Ahora aplicamos el algoritmo FGLM recorriendo los términos de $\mathbb{K}[x, y]^3$ para obtener una base de M respecto del orden TOP determinado por el orden en los enteros $<$ (i.e. $e_i < e_j$, si $i < j$). Denotamos a la base que queremos obtener como $G_2(M)$.
Cada etapa del algoritmo FGLM se puede ver como un proceso de eliminación gaussiana si trabajamos con las tablas definidas a continuación, donde:

- *En la primera fila colocamos los monomios $\mathbb{T}_1$ de $\mathbb{K}[\mathbf{x}]$ ordenados de forma creciente respecto del orden $\succ_{lex}$.*
- *En la primera columna escribimos los términos de la lista* `NextTerms` *del algoritmo FGLM adaptado a submódulos.*
- *Y en el resto de las columnas escribimos la forma normal del elemento de la lista* `NextTerms` *de la fila correspondiente respecto de la base $G_1(M)$ descrito como combinación lineal del correspondiente elemento en el conjunto* $\mathrm{Syz}(-1, f_1, f_2)$. *Es decir el elemento $(\lambda_1, \lambda_2, \lambda_3)$ se corresponde con el elemento $\lambda_1 \cdot (-1) + \lambda_2 \cdot f_1 + \lambda_3 \cdot f_2$.*

Siguiendo el algoritmo FGLM adaptado a submódulos inicializamos la lista `NextTerms` *con los elementos $(1, 0, 0)$, $(0, 1, 0)$ y $(0, 0, 1)$ cuyas formas normales respecto de la base de Gröbner $G_1(M)$ son:*

$$\begin{aligned}
\mathrm{nf}_{G_1(M)}((1,0,0)) &= (1,0,0) \equiv 1 \cdot (-1) \equiv 1 \mod 2 \\
\mathrm{nf}_{G_1(M)}((0,1,0)) &= (0,1,0) \equiv 1 \cdot f_1 = x^2 + x + 1 \\
\mathrm{nf}_{G_1(M)}((0,0,1)) &= (0,0,1) \equiv 1 \cdot f_2 = xy + x + 1
\end{aligned}$$

Es decir tenemos la siguiente tabla:

	1	x	y	x^2	xy
$(1,0,0)$	1				
$(0,1,0)$	1	1		1	
$(0,0,1)$	1	1			1

Aplicamos la eliminación de Gauss a la tabla anterior, obteniendo:

Reducción	1	x	y	x^2	xy
$(1,0,0)$	1				
$(1,1,0)$		1		1	
$(0,1,1)$				1	1

Observamos que las formas normales respecto de la base $G_1(M)$ de los tres elementos considerados son linealmente independientes, así que añadimos estos elementos a la lista `RedTerms`*. Luego incluimos en la lista* `NextTerms` *los términos* $\{x \cdot \mathbf{t} \mid \mathbf{t} \in \texttt{NextTerms}\}$.
Dicho de otra forma, si hubiésemos obtenido alguna fila con todos los elementos ceros, entonces tendríamos una sizigia en el módulo $M \subseteq \mathbb{K}[x,y]^3$ de la forma $\lambda_1 \cdot (-1) + \lambda_2 \cdot f_1 + \lambda_3 \cdot f_2 = 0$ con $\lambda_i \in \mathbb{K}$. Es decir, el orden de los polinomios que forman la sizigia sería de grado cero. Como no es el caso, existe al menos un elemento en todas las sizigias de M que tiene grado mayor que uno. Por eso introducimos la variable x.

Introducimos x	1	x	y	x^2	xy	y^2	x^3	x^2y
$(x,0,0)$		1						
$(x,x,0)$			1			1		
$(0,x,x)$					1		1	

Aplicamos la eliminación de Gauss teniendo en cuenta la tabla formada por los elementos de la lista `RedTerms` *y los nuevos elementos obtenidos.*

Reducción	1	x	y	x^2	xy	y^2	x^3	x^2y
$(x+1,0,0)$					1			
$(1,x+1,0)$							1	
$(1,1,x)$								1

Observamos que las formas normales de los elementos considerados respecto de la base $G_1(M)$ son linealmente independientes con las formas normales de los monomios de la lista `RedTerms`*, así que añadimos los nuevos elementos a dicha lista e introducimos la variable y.*

Introducimos y	1	x	y	x^2	xy	y^2	x^3	x^2y	xy^2
$(y,0,0)$			1						
$(y,y,0)$					1			1	
$(0,y,y)$								1	1

Aplicamos la eliminación de Gauss teniendo en cuenta la tabla formada por los elementos de la lista `RedTerms` *y los nuevos elementos obtenidos.*

Reducción	1	x	y	x^2	xy	y^2	x^3	x^2y	xy^2
$(y,0,0)$			1						
$(y+x,y+1,x+1)$									
$(0,1+y,x+y)$									1

Observamos que la forma normal respecto de la base $G_1(M)$ *del elemento* $(y,y,0)$ *es combinación lineal de las formas normales respecto de la base* $G_1(M)$ *de los elementos*

$$\{(0,1,0),\ (0,0,1),\ (x,x,0),\ (0,x,x)\} \in \texttt{RedTerms}.$$

Por lo tanto el elemento:

$$(y,y,0)+(0,1,0)+(0,0,1)+(x,x,0)+(0,x,x) = (y+x,y+1,x+1) \in M$$

forma un nuevo elemento de la base $G_2(M)$*. De lo que se deduce que la primera componente es un elemento de la base de Gröbner del ideal* I*.*
Seguimos con el algoritmo para obtener el resto de elementos de la base de Gröbner del ideal, así que introducimos x^2*.*

Introducimos x^2	1	x	y	x^2	xy	y^2	x^3	x^2y	xy^2	y^3	x^4	x^3y
$(x^2+x,0,x)$							1					
$(x,x^2+x,0)$											1	
(x,x,x^2)												1

Aplicamos la eliminación de Gauss teniendo en cuenta la tabla formada por los elementos de la lista `RedTerms` *y los nuevos elementos obtenidos.*

Reducción	1	x	y	x^2	xy	y^2	x^3	x^2y	xy^2	y^3	x^4	x^3y
$(x^2+x+1,1,0)$												
$(x,x^2+x,0)$											1	
(x,x,x^2)												1

Observamos que x^2+x+1 es el siguiente elemento de la base de Gröbner del ideal I.
Como no se puede obtener más sizigias con primera componente distinta de cero, hemos terminado el algoritmo presentando como base de Gröbner del ideal I respecto del orden $\succ_{lex}$ el conjunto $G=\{x^2+x+1, y+x\}$.

Ejemplo 3.3. *Consideramos un código binario $\mathcal{C}$ de parámetros $[3,2]$ y con matriz generatriz*

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{2\times 3}.$$

Queremos calcular una base de Gröbner del ideal de $\mathbb{F}_2[x_1,x_2,x_3]$ asociado al código:

$$\blacktriangle I = \langle f_1 = x_1x_3-1, f_2 = x_2x_3-1, f_3 = x_1^2-1, f_4 = x_2^2-1, f_5 = x_3^2-1\rangle$$

respecto del orden lexicográfico con orden de las variables $x_1<x_2<x_3$ que vamos a denotar por $\succ_{lex}$.
Definimos el submódulo $M\subseteq \mathbb{K}[\mathbf{x}]^{n+k+1}$ generado por el conjunto de sizigias de los polinomios $\{-1, f_1, f_2, f_3, f_4, f_5\}$.
Es fácil comprobar que los elementos $\{m_i\}_{i=1}^5$ definidos como:

$$m_1=(f_1,1,0,0,0,0),\quad m_2=(f_2,0,1,0,0,0),$$
$$m_3=(f_3,0,0,1,0,0),\quad m_4=(f_4,0,0,0,1,0),\ m_5=(f_5,0,0,0,0,1)$$

forman una base de Gröbner del módulo M respecto del orden POT en $\mathbb{K}[x_1,x_2,x_3]^6$ inducido por $\succ_{lex}$ que coincide con el orden en $\mathbb{K}[x_1,x_2,x_3]$ inducido por $\succ_{lex}$ y por $\mathbf{w}=(1,\mathrm{LT}_{\succ_{lex}}(f_1),\ldots,\mathrm{LT}_{\succ_{lex}}(f_5))$, que vamos a denotar $G_1(M)$.
Aplicamos el algoritmo FGLM recorriendo los términos de $\mathbb{K}[x_1,x_2,x_3]^6$ para obtener una base de M respecto del orden TOP determinado por el orden en los enteros $<$ ($e_i<e_j$ si $i<j$), que vamos a denominar $G_2(M)$.
Como hemos visto en el ejemplo anterior, cada etapa del algoritmo FGLM se puede ver como una eliminación gaussiana. Los binomios x_i^2-1 estarán incluidos en el cálculo aplicando la relación $x_i^2\equiv 1 \mod \blacktriangle I$.
Inicializamos el algoritmo con la siguiente tabla:

	-1	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
$(1,0,0,0,0,0)$	1							
$(0,1,0,0,0,0)$	1					1		
$(0,0,1,0,0,0)$	1						1	

Aplicamos la eliminación de Gauss a la tabla anterior.

Reducción	-1	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
$(1,0,0,0,0,0)$	1							
$(1,1,0,0,0,0)$						1		
$(1,0,1,0,0,0)$							1	

Observamos que las formas normales respecto de la base de Gröbner $G_1(M)$ de los tres primeros elementos considerados son linealmente independientes, así que los añadimos a la lista `RedTerms` *y luego introducimos los términos* $\{x_1 \cdot \mathbf{t} \;\; y \;\; x_2 \cdot \mathbf{t} \mid \mathbf{t} \in \texttt{NextTerms}\}$ *a la lista* `NextTerms`.

Introducimos x_1	-1	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
$(x_1,0,0,0,0,0)$		1						
$(x_1,x_1,0,0,0,0)$				1				
$(x_1,0,x_1,0,0,0)$								1

Introducimos x_2	-1	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
$(x_2,0,0,0,0,0)$			1					
$(x_2,x_2,0,0,0,0)$								1
$(x_2,0,x_2,0,0,0)$				1				

Aplicamos la eliminación de Gauss teniendo en cuenta la tabla formada por los elementos de la lista `RedTerms` *y los nuevos elementos obtenidos.*

Reducción	-1	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
$(x_2,0,0,0,0,0)$			1					
$(x_2-x_1,x_2,x_1,0,0,0)$								
$(x_2,0,x_2,0,0,0)$				1				

Observamos que la forma normal del elemento $(x_2,x_2,0,0,0)$ respecto de la base $G_1(M)$ es combinación lineal de las formas normales respecto de la base $G_1(M)$ de elementos de la lista `RedTerms`.
Por lo tanto el elemento $(x_2-x_1,x_2,x_1,0,0,0)$ forma un nuevo elemento de la base $G_2(M)$. De lo que se deduce que la primera componente es un elemento de la base de Gröbner del ideal I.
Como el resto de sizigias con primera componente distinta de cero no puede ser múltiplo de la variable x_2, podemos omitir de los cálculos todos los múltiplos del elemento $x_2 \cdot (1,1,0,0,0,0,0)$.
Introducimos los términos $\{x_3 \cdot \mathbf{t} \mid \mathbf{t} \in \texttt{NextTerms}\}$ *a la lista* `NextTerms`.

Introducimos x_3	-1	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
$(x_3, 0, 0, 0, 0, 0)$				1				
$(x_3, x_3, 0, 0, 0, 0)$		1						
$(x_3, 0, x_3, 0, 0, 0)$			1					

Aplicamos la eliminación de Gauss teniendo en cuenta la tabla formada por los elementos de la lista `RedTerms` *y los nuevos elementos obtenidos.*

Reducción	-1	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
$(x_3, 0, 0, 0, 0, 0)$				1				
$(x_3 - x_1, x_3, 0, 0, 0, x_1)$								
$(x_3 - x_2, 0, x_3, 0, 0, x_2)$								

Observamos que el elemento $x_3 - x_1$ *es un nuevo elemento de la base de Gröbner del ideal* I*. Nos falta un elemento más para completar la base de Gröbner del ideal que no puede ser múltiplo de los elementos*

$$x_3 \cdot (1, 1, 0, 0, 0, 0, 0), \quad x_2 \cdot (1, 1, 0, 0, 0, 0, 0)$$

Luego el último elemento de la base de Gröbner es $x_1^2 - 1$*.*
Por lo tanto obtenemos $G = \{x_2 - x_1, x_3 - x_1, x_1^2 - 1\}$ *como base de Gröbner del ideal* I *respecto del orden* $\succ_{lex}$*.*

El proceso que hemos presentado es un algoritmo general que tiene las siguientes ventajas computacionales al ser aplicado sobre un ideal asociado a un código:

- Sabemos cuándo podemos parar, ya que la dimensión de la base de Gröbner del ideal que queremos obtener está acotada por la dimensión del código que estamos considerando.
- Cada una de las etapas tiene el coste de aplicar el algoritmo de eliminación de Gauss sobre matrices con pocos 1 (lo que se conoce como matrices *sparce*).
- Los principales problemas del algoritmo de Buchberger, que son el crecimiento doble exponencial del grado y el crecimiento de los coeficientes, no tienen que ser tenidos en cuenta en este caso, ya que en $\mathbb{F}_q$ todos los polinomios que tenemos que considerar son de grado menor que $n \times (q-1)$ y, como la información se encuentra en los exponentes de los binomios, siempre podemos utilizar como cuerpo arbitrario $\mathbb{K} = \mathbb{F}_2$.

- La complejidad del algoritmo que acabamos de describir depende del número de variables n y del número de elementos irreducibles del espacio cociente $\mathbb{K}[\mathbf{x}]/\mathcal{R}_{\mathcal{C}}$, que coincide con el número de síndromes distintos del código $\mathcal{C}$, es decir 2^{n-k}. Por lo tanto, la complejidad del algoritmo viene determinada por la fórmula: $O(n^2 2^{n-k})$. Recordemos que la complejidad del algoritmo de Buchberger viene dada por la fórmula $O(2^{2^{\frac{n}{10}}})$ por lo tanto, aunque se reduce la complejidad con las técnicas FGLM, el problema del cálculo de una base de Gröbner reducida sigue siendo NP-completo.

Capítulo 4

Aplicaciones a la Teoría de Códigos y la Criptografía

Consideremos un código $\mathcal{C} \subseteq \mathbb{F}_q^n$ lineal de parámetros $[n, k, d]$ y sea $H_{\mathcal{C}}$ su matriz de paridad, es decir, podemos expresar el código como el conjunto:

$$\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n \mid H_{\mathcal{C}}\ \mathbf{c}^t \equiv 0 \mod q\}.$$

El problema de descodificación completa (PDC) consiste en, recibido un vector $\mathbf{y} \in \mathbb{F}_q^n$, calcular la palabra perteneciente al código $\mathbf{c} \in \mathcal{C} \subseteq \mathbb{F}_q^n$ que maximice la probabilidad de recibir $\mathbf{y}$ siendo enviada $\mathbf{c}$, es decir maximizar:

$$\mathbb{P}\left(\mathbf{y} \text{ recibida } / \mathbf{c} \text{ enviada}\right).$$

Por el Teorema de Bayes tenemos que:

$$\begin{aligned}\mathbb{P}\left(\mathbf{y} \text{ recibida}/\mathbf{c} \text{ enviada}\right) &= \frac{\mathbb{P}\left(\mathbf{y} \text{ recibida}, \mathbf{c} \text{ enviada}\right)}{\mathbb{P}\left(\mathbf{y} \text{ enviada}\right)} \\ &= \mathbb{P}\left(\mathbf{c} \text{ enviada}/\mathbf{y} \text{ recibida}\right) \cdot \frac{\mathbb{P}\left(\mathbf{y} \text{ recibida}\right)}{\mathbb{P}\left(\mathbf{c} \text{ enviada}\right)}.\end{aligned}$$

En el caso de utilizar canales simétricos, como todas las palabras del código tienen la misma probabilidad de ser enviadas y la probabilidad de recibir un vector $\mathbf{y} \in \mathbb{F}_q^n$ es fijo una vez que recibimos dicho vector, entonces $\mathbb{P}\left(\mathbf{y} \text{ recibida}/\mathbf{c} \text{ enviada}\right)$ toma su valor máximo cuando

$$\mathbb{P}\left(\mathbf{c} \text{ enviada}/\mathbf{y} \text{ recibida}\right)$$

alcanza el máximo.

Si la *probabilidad de error del canal de transmisión* p es estrictamente menor a $\frac{1}{2}$, entonces el problema de descodificación completa es equivalente al problema de descodificación por mínima distancia que consiste en, recibido un vector $\mathbf{y} \in \mathbb{F}_q^n$, calcular la palabra del código $\mathbf{c} \in \mathcal{C} \subseteq \mathbb{F}_q^n$ que difiera en menos posiciones del vector recibido. Es decir, calcular la palabra del código que tenga menor distancia hamming con la palabra recibida.
En efecto, si denotamos por $d_H(\mathbf{c}, \mathbf{y}) = \sharp\{i \mid 1 \leq i \leq n, c_i \neq y_i\} = d$ a la distancia de Hamming entre la palabra recibida $\mathbf{y} \in \mathbb{F}_q^n$ y la palabra enviada $\mathbf{c} \in \mathcal{C}$, se tiene que:

$$\mathbb{P}\left(\mathbf{y} \text{ recibida}/\mathbf{c} \text{ enviada}\right) = (1-p)^{n-d} \cdot p^d = (1-p)^n \cdot \left(\frac{p}{1-p}\right)^d.$$

Como $p < \frac{1}{2}$, la fórmula anterior se maximiza minimizando el parámetro d.
Si fijamos una cota t para los errores que podemos corregir el problema se transforma en determinar un elemento del código $\mathcal{C}$ (si existe) tal que difiera en menos de t posiciones de la palabra recibida.
Este método de descodificación se puede efectuar cuando el medio de transmisión es un canal simétrico.
Los métodos de descodificación propuestos son NP-completos (véanse [7, 10]), incluso si se permite realizar preproceso en los datos véase [20].
Consideramos en el espacio vectorial $\mathbb{F}_q^n$ la relación de equivalencia $\mathcal{R}_\mathcal{C}$ definida para todo $\mathbf{u},\ \mathbf{v} \in \mathbb{F}_q^n$ como $(\mathbf{u}, \mathbf{v}) \in \mathcal{R}_\mathcal{C}$ si y sólo si $\mathbf{u} - \mathbf{v} \in \mathcal{C}$, es decir si $H_\mathcal{C}\ \mathbf{u}^t \equiv H_\mathcal{C}\ \mathbf{v}^t \mod q$.
Dado un vector $\mathbf{u} \in \mathbb{F}_q^n$ definimos clase de equivalencia de dicho vector respecto de $\mathcal{R}_\mathcal{C}$ como el conjunto formado por todos los elementos relacionados con $\mathbf{u}$ y lo denotamos por $\overline{\mathbf{u}} = \{\mathbf{v} \in \mathbb{F}_q^n \mid (\mathbf{u}, \mathbf{v}) \in \mathcal{R}_\mathcal{C}\}$.
El conjunto formado por todas las clases de equivalencia de los elementos de $\mathbf{u} \in \mathbb{F}_q^n$ es el conjunto cociente $\mathbb{F}_q^n/\mathcal{R}_\mathcal{C}$. Este conjunto constituye una partición disjunta del conjunto $\mathbb{F}_q^n$, ya que las clases de equivalencia no son vacías; si dos clases de equivalencia son distintas son necesariamente disjuntas y todo elemento de $\mathbb{F}_q^n$ está en alguna de las clases de equivalencia definidas.
Es decir, teniendo en cuenta que el cardinal de cada clase de equivalencia coincide con el cardinal del código $\mathcal{C}$, tenemos que:

$$\mathbb{F}_q^n = \bigcup_{\mathbf{u} \in \mathbb{F}_q^n} \overline{\mathbf{u}} = \mathbb{F}_q^n/\mathcal{R}_\mathcal{C} \quad \Leftrightarrow \quad \sharp\overline{\mathbf{u}} \cdot \sharp\left(\mathbb{F}_q^n/\mathcal{R}_\mathcal{C}\right) = \sharp\mathbb{F}_q^n$$

$$\Leftrightarrow \quad \sharp\left(\mathbb{F}_q^n/\mathcal{R}_\mathcal{C}\right) = \frac{\sharp\mathbb{F}_q^n}{\sharp\overline{\mathbf{u}}} = \frac{q^n}{q^k} = q^{n-k}.$$

De donde se deduce que la dimensión del espacio cociente $\mathbb{F}_q^n/\mathcal{R}_\mathcal{C}$ es $n-k$.
De forma natural a cada vector $\mathbf{a} \in \mathbb{Z}_q^n$ podemos asociarle un monomio en las variables $x_1, \ldots, x_n$ como el producto $\mathbf{x}^{\blacktriangle \mathbf{a}} = x_1^{\blacktriangle a_1} \cdots x_n^{\blacktriangle a_n}$.
Inversamente, podemos definir el homomorfismo Φ que asocia a cada monomio $\mathbf{x}^\alpha$ con $\alpha \in \mathbb{N}^n$ un vector en $\mathbb{Z}_q^n$ como:

$$\begin{array}{rccc} \Phi: & \mathbb{K}[\mathbf{x}] & \longrightarrow & \mathbb{Z}_q^n \\ & \mathbf{x}^\alpha & \longmapsto & \Phi(\mathbf{x}^\alpha) = (\blacktriangledown\alpha_1, \ldots, \blacktriangledown\alpha_n) \end{array}$$

De esta forma podemos definir el ideal asociado al código $\mathcal{C}$ como:

$$I(H_\mathcal{C}) = \langle \mathbf{x}^\mathbf{a} - \mathbf{x}^\mathbf{b} \mid \Big(\Phi(\mathbf{x}^\mathbf{a}), \Phi(\mathbf{x}^\mathbf{b})\Big) \in \mathcal{R}_\mathcal{C} \rangle.$$

Sea $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\} \subseteq \mathbb{Z}_q^n$ un conjunto de generadores del código $\mathcal{C}$ (recordemos que las filas de la matriz generatriz del código $\mathcal{C}$ forman un conjunto de generadores de dicho código), entonces por el Teorema 3.5 sabemos que el ideal $I(H_\mathcal{C})$ es equivalente al ideal:

$$\blacktriangle I = \langle \{\mathbf{x}^{\blacktriangle \mathbf{w}_1} - 1, \ldots, \mathbf{x}^{\blacktriangle \mathbf{w}_k} - 1\} \cup \{x_i^q - 1\}_{i=1}^n \rangle \subseteq \mathbb{K}[\mathbf{x}]$$

De esta forma podemos definir el espacio cociente $\mathbb{K}[\mathbf{x}]/\blacktriangle I$ formado por el conjunto de clases de equivalencia:

$$[\mathbf{x}^\alpha] = \{\mathbf{x}^\beta \in \mathbb{T}_1 \mid \mathbf{x}^\alpha - \mathbf{x}^\beta \in \blacktriangle I\},$$

Donde $\mathbb{T}_1$ denota el conjunto de monomios del anillo $\mathbb{K}[\mathbf{x}]$.

4.1. Descodificación por síndrome

Expondremos en esta sección un método general de descodifición para códigos lineales. Sea $\mathcal{C} \subseteq \mathbb{F}_q^n$ un código lineal de parámetros $[n, k, d]$. Supongamos que se envía la palabra $\mathbf{c} \in \mathcal{C}$ y que recibimos el vector $\mathbf{y} \in \mathbb{F}_q^n$. La estrategia que seguiremos para descodificar el vector $\mathbf{y}$ es calcular la distancia de $\mathbf{y}$ a todas las palabras del código $\mathcal{C}$ y descodificarla por la más próxima.
Sabemos que $\mathbf{e} = \mathbf{y} - \mathbf{c}$ es el error cometido durante la transmisicón y que el código $\mathcal{C}$ considerado tiene una capacidad correctora de $t = \left\lfloor \frac{d-1}{2} \right\rfloor$, por lo tanto:

- Si durante la transmisión se han cometido a lo sumo t errores, esto es:

  $$t \geq w_H(\mathbf{e}) = w_H(\mathbf{y} - \mathbf{c}) = d_H(\mathbf{y}, \mathbf{c}),$$

  entonces $\mathbf{c} \in \mathcal{C}$ es la única palabra del código con tal propiedad y por tanto la descodificación es correcta.

- Si $t < w_H(\mathbf{e}) < d$ podremos detectar que se han producido errores ya que $\mathbf{y} \in \mathcal{C}$, pero en general no podremos corregirlos.
- Si $w_H(\mathbf{e}) \geq d$, entonces la descodificación fallará.

Definición 4.1. *Sea $H_{\mathcal{C}}$ la matriz de control del código $\mathcal{C} \subseteq \mathbb{F}_q^n$, llamaremos síndrome del vector $\mathbf{y} \in \mathbb{F}_q^n$ al vector:*

$$S(\mathbf{y}) = H_{\mathcal{C}}\ \mathbf{y}^t \in \mathbb{F}_q^{n-k}.$$

Observemos que si el vector $\mathbf{y} \in \mathcal{C}$, entonces $S(\mathbf{y}) = 0$. Es más, el hecho de encontrarnos con un vector $\mathbf{y} \in \mathbb{F}_q^n$ tal que $S(\mathbf{y}) \neq 0$ nos indica que $\mathbf{y}$ no es una palabra del código $\mathcal{C}$.
Por otra parte, como el síndrome es una aplicación lineal, se tiene que:

$$\begin{aligned} S(\mathbf{y}) = S(\mathbf{c}+\mathbf{e}) &= H_{\mathcal{C}}\ (\mathbf{c}+\mathbf{e})^t = H_{\mathcal{C}}\ \mathbf{c}^t + H_{\mathcal{C}}\ \mathbf{e}^t \\ &= S(\mathbf{c}) + S(\mathbf{e}) = 0 + S(\mathbf{e}) = S(\mathbf{e}). \end{aligned}$$

Es decir, la imagen de $\mathbf{y}$ por la aplicación síndrome sólo depende del error cometido.

Proposición 4.1. *El síndrome del vector recibido $\mathbf{y} \in \mathbb{F}_q^n$ es combinación lineal de las columnas de la matriz de paridad $H_{\mathcal{C}}$ correspondientes a las posiciones del error producido durante la transmisión.*

Demostración. Denotemos por $h^{(i)}$ a la columna i-ésima de la matriz de paridad $H_{\mathcal{C}}$ del código $\mathcal{C} \subseteq \mathbb{F}_q^n$. De esta forma podemos escribir $H_{\mathcal{C}} = (h^{(1)}, \ldots, h^{(n)})$. Sea $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$ el vector recibido, entonces se tiene:

$$S(\mathbf{y}) = H_{\mathcal{C}}\ \mathbf{y}^t = y_1 h^{(1)} + y_2 h^{(2)} + \ldots + y_n h^{(n)} = S(\mathbf{e}).$$

Supongamos que durante la transmisión se han producido $k \leq \frac{d(\mathcal{C})-1}{2}$ errores, es decir: $w_H(\mathbf{e}) = k$, entonces $\mathbf{e}$ tiene k componentes distintas de cero y

$$S(\mathbf{y}) = S(\mathbf{e}) = e_{i_1} h^{(i_1)} + e_{i_2} h^{(i_2)} + \ldots + e_{i_k} h^{(i_k)},$$

de donde se concluye la demostración. □

Teniendo en cuenta la definición de la relación de equivalencia $\mathcal{R}_{\mathcal{C}}$, es fácil ver que $(\mathbf{u}, \mathbf{v}) \in \mathcal{R}_{\mathcal{C}}$ si y sólo si $S(\mathbf{u}) = S(\mathbf{v})$. Luego, como el síndrome de la palabra recibida, $\mathbf{y}$, coincide con el síndrome del error de transmisión, $\mathbf{e}$, es decir, $S(\mathbf{y}) = S(\mathbf{e})$, entonces el error es uno de los vectores del conjunto:

$$S^{-1}\left(S(\mathbf{y})\right) = \{\mathbf{z} \in \mathbb{F}_q^n \mid S(\mathbf{z}) = S(\mathbf{y})\}.$$

Definición 4.2. *Fijado un vector* $\mathbf{y} \in \mathbb{F}_q^{n-k}$*, si en la clase de equivalencia* $[\mathbf{y}]$ *existe un único elemento de peso mínimo, este vector recibe el nombre de líder de la clase de equivalencia* $[\mathbf{y}]$*.*

Algunos autores generalizan este concepto y denotan *líder de una clase de equivalencia* a todo vector con peso mínimo en dicha clase de equivalencia. En general no existe un único vector de peso mínimo en cada clase, pero si una clase contiene un elemento de peso menor o igual que t, siendo t la capacidad correctora del código, entonces dicho elemento es el líder de la clase de equivalencia correspondiente.

Proposición 4.2. *Cada clase de equivalencia de* $\mathbb{F}_q^n/\mathcal{R}_\mathcal{C}$ *posee a lo sumo un elemento de peso menor o igual que* t*, siendo* t *la capacidad correctora del código* $\mathcal{C}$*.*

Demostración. Supongamos que tenemos dos elementos líder $\mathbf{u},\ \mathbf{v} \in \mathbb{F}_q^n/\mathcal{R}_\mathcal{C}$ con peso menor o igual que t en una misma clase de equivalencia del espacio cociente $\mathbb{F}_q^n/\mathcal{R}_\mathcal{C}$.
Al pertenecer ambos elementos a la misma clase, se tiene que $\mathbf{u} - \mathbf{v} \in \mathcal{C}$. Por lo tanto: $w_H(\mathbf{u}-\mathbf{v}) \leq w_H(\mathbf{u}) + w_H(\mathbf{v}) \leq 2t < d(\mathcal{C})$, lo que contradice el carácter minimal de la distancia mínima de un código. Deducimos entonces que $\mathbf{u}$ y $\mathbf{v}$ son iguales. □

Recibido un vector $\mathbf{y} \in \mathbb{F}_q^n$, buscamos asociar a $\mathbf{y}$ una palabra del código $\mathbf{x} \in \mathcal{C}$ que tenga el mayor parecido posible con la palabra enviada. En términos de distancia mínima esto se traduce en encontrar una palabra $\mathbf{x} \in \mathcal{C}$ que minimice $d_H(\mathbf{y}, \mathbf{x}) = w_H(\mathbf{y} - \mathbf{x})$. Como para cada $\mathbf{x} \in \mathcal{C}$ los vectores $\mathbf{y} - \mathbf{x}$ están en la misma clase de equivalencia que se corresponde con la clase $[\mathbf{y}]$ de $\mathbb{F}_q^n/\mathcal{R}_\mathcal{C}$, entonces la palabra $\mathbf{x} \in \mathcal{C}$ buscada es tal que el error producido durante la transmisión $\mathbf{e} = \mathbf{y} - \mathbf{x}$ constituya el líder de la clase de equivalencia $[\mathbf{y}]$. Por lo tanto, si el peso del error cometido no supera la capacidad correctora del código $\mathcal{C}$, entonces la descodificación es correcta.
Para llevar a cabo el proceso de descodificación construimos una tabla con dos columnas. En la primera columna escribimos el síndrome de un elemento cualquiera de cada clase y en la otra columna el líder de la clase correspondiente (si existe).

Definición 4.3. *En un tablero de síndromes completo aparecen listados, si existen, los síndromes de los vectores con peso mínimo de cada clase de equivalencia de* $\mathbb{F}_q^n/\mathcal{R}_\mathcal{C}$*. Por lo tanto esta tabla tendrá tantas filas como número de clases de equivalencia tiene* $\mathbb{F}_q^n/\mathcal{R}_\mathcal{C}$*, es decir* q^{n-k}*.*

Los tableros completos presentan varios inconvenientes principalmente relacionados con el almacenamiento de una tabla bastante grande. Además, en este caso es necesario listar una a una todas las clases, lo que no es tarea sencilla.

Definición 4.4. *En un tablero de síndromes incompleto sólo se listan las clases que poseen un líder (es decir un elemento con peso menor o igual a* $t = \left\lfloor \frac{d(\mathcal{C})-1}{2} \right\rfloor$*). En este caso la tabla tiene:*

$$M = \begin{pmatrix} n \\ 0 \end{pmatrix} + (q-1)\begin{pmatrix} n \\ 1 \end{pmatrix} + \ldots + (q-1)^t \begin{pmatrix} n \\ t \end{pmatrix} \text{ filas.}$$

El mayor inconveniente de los tableros incompletos es que sigue siendo necesario almacenar una tabla bastante amplia. A veces se alivia dicha dificultad utilizando códigos que transmiten simetría al tablero, como los códigos cíclicos, aunque es insuficiente a efectos prácticos.
Una vez hayamos creado la tabla de descodificación (que se construye sólo una vez), recibido el vector $\mathbf{y} \in \mathbb{F}_q^n$ procedemos con el algoritmo de descodificación (Algoritmo 12) utilizando el líder de la clase.

Algoritmo 12 Algoritmo de descodificación utilizando el líder de la clase

INPUT: El vector recibido $\mathbf{y} \in \mathbb{F}_q^n$ y una tabla de descodificación del código $\mathcal{C} \subseteq \mathbb{F}_q^n$.

OUTPUT: La palabra $\mathbf{x} \in \mathcal{C}$, si existe, que minimice el conjunto

$$\{d_H(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathcal{C}\}.$$

Calculamos el síndrome de la palabra recibida $S(\mathbf{y}) = H_{\mathcal{C}}\ \mathbf{y}^t$.
Buscamos el síndrome obtenido en la columna de síndromes de la tabla de descodificación.
if la clase correspondiente posee líder **then**
  Se considera que dicho líder e se corresponde con el error que se ha producido durante la transmisión, $\mathbf{e}$.
else
  La descodificación falla.
end if
La palabra descodificada es $\mathbf{x} = \mathbf{y} - \mathbf{e} \in \mathcal{C}$.

Ejemplo 4.1. *Mostramos un ejemplo de tabla incompleta.*

Consideremos el código lineal binario $\mathcal{C} \subseteq \mathbb{F}_2^4$ *con matriz generatriz:*

$$G_{\mathcal{C}} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \in \mathbb{F}_2^{2\times 4}.$$

Es fácil comprobar que la matriz de paridad $H_{\mathcal{C}}$ *de dicho código se corresponde con su matriz generatriz, es decir* $G_{\mathcal{C}} = H_{\mathcal{C}}$*. Se trata por tanto de un código de dimensión* $k = 2$*, longitud* $n = 4$ *y que tiene distancia mínima* $d = 2$*. Recordemos que para calcular la distancia mínima basta con calcular el rango de la matriz de paridad* $H_{\mathcal{C}}$ *del correspondiente código* $\mathcal{C}$*. Recibido cualquier vector* $\mathbf{y} \in \mathbb{F}_2^4$ *se tiene que* $S(\mathbf{y}) \in \mathbb{F}_2$*. Por lo tanto:*

$$S(\mathbf{y}) \in \{00, 01, 10, 11\}.$$

Para crear una tabla de síndromes incompleta, en lugar de listar todas las clases de equivalencia, escogeremos una clase de equivalencia con síndrome entre los posibles listados anteriormente.
Para ello observamos que

$$\mathcal{C} = \{0000, 1010, 0101, 1111\}.$$

De manera que:

- $0000 + \mathcal{C} = \{0000, 1000, 0100, 1100\}$ *con síndrome* 00*.*
- $1000 + \mathcal{C} = \{1000, 0010, 1101, 0111\}$ *con síndrome* 10*.*
- $0100 + \mathcal{C} = \{0100, 1110, 0001, 1011\}$ *con síndrome* 01*.*
- $1100 + \mathcal{C} = \{1100, 0110, 1001, 0011\}$ *con síndrome* 11*.*

Por lo tanto la tabla de síndromes incompleta del código $\mathcal{C}$ *es:*

error	*síndrome*
0000	*00*
1000	*10*
0100	*01*
1100	*11*

De esta forma, si recibimos el mensaje 0001*, a su síndrome* 01 *le corresponde, según la tabla, el líder* 0100*. Luego, el mensaje descodificado es:* $0001 + 0100 = 0101$*.*

Ejemplo 4.2. *Mostramos a continuación un ejemplo de tabla completa. Dado el código binario lineal* $\mathcal{C} \subseteq \mathbb{F}_2^8$ *con matriz generatriz:*

$$G_{\mathcal{C}} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{2\times 8}.$$

Es fácil comprobar que una matriz de control de dicho código es:

$$H_{\mathcal{C}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{F}_2^{6\times 8}.$$

Se trata por tanto de un código lineal con parámetros $[n = 8, k = 2, d = 5]$*, luego corrige dos errores.*
De esta forma:

- *Si recibimos el mensaje* $\mathbf{y} = 11011011$*, con síndrome* $S(\mathbf{y}) = 111000^t$ *que, según la tabla, le corresponde el líder* 10000100*.*

  El mensaje codificado es, por tanto,

  $$11011011 + 10000100 = 01011111 \in \mathcal{C}.$$

  Como el líder tiene peso 2*, podemos asegurar que la descodificación es correcta si no han ocurrido más de dos errores durante la transmisión.*

- *Si recibimos el mensaje* $\mathbf{y} = 01110010$*, con síndrome* $S(\mathbf{y}) = 101101^t$ *que, según la tabla, le corresponde el líder* 10010001*.*

  Aunque el líder tiene peso mayor que la capacidad teórica de corrección del código, es posible descodificar el mensaje recibido por

  $$01110010 + 10010001 = 11100011.$$

  Sabemos que se han producido al menos tres errores durante la transmisión y que nuestra descodificación será correcta si y sólo si se producen exactamente tres errores.

- *Si recibimos el mensaje* $\mathbf{y} = 01011000$*, con síndrome* $S(\mathbf{y}) = 000111$*.*

  Según la tabla, la clase de equivalencia correspondiente no tiene elemento líder, por lo tanto no es posible la descodificación. Podemos decir que se han producido 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

Cuadro 4.1: Tabla de descodificación completa asociada al código del ejemplo 4.2

Definición 4.5. *Un código lineal $\mathcal{C}$ es perfecto si el tablero de descodificación completo e incompleto coinciden. Es decir, si se verifica que:*

$$\sum_{i=0}^{t}(q-1)^i \left(\begin{array}{c} n \\ i \end{array} \right) = q^{n-k}.$$

La condición anterior es muy restrictiva, de hecho, los únicos códigos binarios no triviales que son perfectos son los códigos de Hamming y el código Golay binario G_{23}.

4.2. Descodificación completa en el caso binario

En esta sección vamos a considerar $\mathcal{C} \subseteq \mathbb{F}_2^n$ un código binario lineal de parámetros $[n, k, d]$, donde $H_{\mathcal{C}}$ denota su matriz de paridad, es decir, podemos expresar las palabras del código $\mathcal{C}$ de la siguiente forma:

$$\mathcal{C} = \{\mathbf{c} \in \mathbb{Z}_2^n \mid H_{\mathcal{C}}\ \mathbf{u}^t \equiv 0 \mod 2\}.$$

4.2.1. Relación con la programación lineal entera modular

Sea $\mathbf{c} \in \mathcal{C}$ la palabra del código transmitida, $\overline{e} \in \mathbb{Z}_2^n$ el error cometido durante la transmisión y $\mathbf{r} \equiv \mathbf{c} + \overline{e} \mod 2$ la palabra recibida. Entonces el problema de descodificación completa es equivalente a encontrar un vector error $\mathbf{e} \in \mathbb{Z}_2^n$ con menor peso hamming entre todos los vectores peso que tienen el mismo síndrome que la palabra recibida, es decir que verifican que

$$H_{\mathcal{C}}\ \mathbf{e}^t \equiv H_{\mathcal{C}}\ \mathbf{r}^t \mod 2.$$

Observamos que, considerando como vector peso $\mathbf{w} = (1, \ \ldots \ , 1)$, calcular el peso de Hamming del vector $\mathbf{e} \in \mathbb{Z}_2^n$ es equivalente al producto escalar entre el vector $\mathbf{w}$ y el vector error $\mathbf{e}$, es decir, $w_H(\mathbf{e}) = \mathbf{w} \cdot \mathbf{e}$.
Por lo tanto resolver el problema lineal entero modular:

$$\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b}) = \left\{ \begin{array}{l} \text{Minimizar: } \ w_H(\mathbf{e}) = \mathbf{w} \cdot \mathbf{e} \\ \text{Sujeto a: } \left\{ \begin{array}{l} H_{\mathcal{C}}\ \mathbf{e}^t = H_{\mathcal{C}}\ \mathbf{r}^t \\ \mathbf{e} \in \mathbb{Z}_q^n \end{array} \right. \end{array} \right.$$

es equivalente a la descodificación completa de la palabra $\mathbf{r}$.
Este método es equivalente al que proponen Ikegami y Kaji en [37], que hemos probado con el Teorema 3.5 que es equivalente al método propuesto en [15]. Por lo tanto la base de Gröbner reducida asociada al problema lineal entero modular anterior nos aporta un test set minimal cuyos elementos son los binomios asociados a las palabras de soporte mínimo del código.

4.2.2. Descodificación por gradiente

En esta sección vamos a describir dos algoritmos de descodificación por gradiente. El primer algoritmo que vamos a estudiar se puede consultar en [42] para obtener una descripción más detallada.
Denotemos por $\overline{\mathbf{y}}$ a la clase de equivalencia de $\mathbb{F}_2^n/\mathcal{R}_\mathcal{C}$ que contiene al vector $\mathbf{y}$ y por $w_H(\overline{\mathbf{y}})$ el peso hamming de cualquiera de los líderes de dicha clase de equivalencia. Siguiendo esta notación, Liebler define en [42] el *algoritmo de descodificación por gradiente líder* como el algoritmo que, recibido un vector $\mathbf{y} \in \mathbb{F}_2^n$, reemplaza $\mathbf{y}$ por un vector vecino $\mathbf{y}' \in \mathbb{F}_2^n$ que esté a distancia hamming 1 del vector recibido y tal que $w_H(\overline{\mathbf{y}}) \geq w_H(\overline{\mathbf{y}'})$. Luego repite el proceso con el vector $\mathbf{y}'$ hasta encontrar un vecino que se encuentre en la clase del cero, es decir que pertenezca al código.

Algoritmo 13 Algoritmo de descodificación por gradiente líder

INPUT: El vector recibido $\mathbf{y} \in \mathbb{F}_2^n$.
OUTPUT: La palabra del código $\mathbf{c} \in \mathcal{C}$ más próxima al vector $\mathbf{y}$.
  while $w_H(\overline{\mathbf{y}}) \neq 0$ **do**
    Calcular $\mathbf{y}' \in \mathbb{F}_2^n$ tal que $d_H(\mathbf{y}, \mathbf{y}') = 1$ y $w_H(\overline{\mathbf{y}}) \geq w_H(\overline{\mathbf{y}'})$.
    $\mathbf{y} := \mathbf{y}'$.
  end while
  Devolver $\mathbf{c} = \mathbf{y}$.

Este algoritmo es esencialmente el algoritmo de descodificación por síndrome dividido en etapas más pequeñas.
En el artículo [42] se presenta la primera construcción de función gradiente $\gamma : \mathbb{F}_2^n/\mathcal{R}_\mathcal{C} \longrightarrow \mathbb{Z}$ que es una función estrictamente decreciente respecto del peso hamming para poder ejecutar el Algoritmo 13 de una manera efectiva.
El siguiente algoritmo de descodificación por gradiente que vamos a estudiar fue presentado por Ashikhmin y Barg en [6]. En este algoritmo, a diferencia del anterior, en cada paso el representante elegido permanece en la misma clase de equivalencia hasta que se alcanza el líder de dicha clase.
Para entender este algoritmo necesitamos recordar la definición de palabras de soporte mínimo y de Test Set asociado a un código.
Definimos soporte de una palabra del código $\mathbf{c} \in \mathcal{C}$ como el conjunto de coordenadas distintas de cero del vector $\mathbf{c}$. Es decir: $\text{supp}(\mathbf{c}) = \{i \mid c_i \neq 0\}$.

Definición 4.6. *Diremos que una palabra del código $\mathbf{c} \in \mathcal{C}$ es una palabra de soporte mínimo en $\mathcal{C}$ si se trata de una palabra distinta del cero y su soporte no está contenido en ninguna otra palabra del código $\mathcal{C}$.*

Es decir, si no existe otra palabra $\mathbf{c}' \in \mathcal{C}$ *tal que* $\text{supp}(\mathbf{c}') \subseteq \text{supp}(\mathbf{c})$.

Denotamos por $\mathcal{M}_\mathcal{C}$ al conjunto de palabras de soporte mínimo del código $\mathcal{C}$.

Definición 4.7. *Al subconjunto de vectores* $\mathcal{T} \subseteq \mathcal{C}$ *tal que para cada vector* $\mathbf{y} \in \mathbb{F}_2^n$ *se tenga que* $\mathbf{y} \in \mathcal{C}$*, o bien existe* $\mathbf{z} \in \mathcal{T}$ *tal que* $w_H(\mathbf{y} - \mathbf{z}) < w_H(\mathbf{y})$ *se denomina Test Set asociado al código* $\mathcal{C}$*.*

La existencia de un *Test Set* del código $\mathcal{C}$ nos aporta el *algoritmo de descodificación por gradiente* propuesto en [7] y que describimos en el Algoritmo 14. La idea principal del algoritmo es, recibido un vector $\mathbf{y} \in \mathbb{F}_2^n$, restar los vectores apropiados del Test Set $\mathcal{T}$ hasta obtener una palabra del código.
En [7] se señala que si consideramos $\mathcal{T} = \mathcal{M}_\mathcal{C}$, entonces el algoritmo de descodificación por gradiente es equivalente a la descodificación por mínima distancia.

Algoritmo 14 Algoritmo de descodificación por gradiente utilizando el Test Set asociado al código

INPUT: El vector recibido $\mathbf{y} \in \mathbb{F}_2^n$ y un Test Set $\mathcal{T} \subseteq \mathcal{C}$.
OUTPUT: La palabra del código $\mathbf{c} \in \mathcal{C}$ más próxima al vector $\mathbf{y}$.

```
c := 0.
while exista z ∈ T tal que w_H(y − z) < w_H(y) do
    c := c + z.
    y := y − z.
end while
Devolver c ∈ C.
```

Liebler menciona en [42] que los dos algoritmos expuestos en esta sección, a pesar de compartir la filosofía de la descodificación por gradiente, son diferentes.
Nosotros vamos a demostrar que estos dos algoritmos son duales en el sentido de que son dos formas de entender la representación de Gröbner de un código.
Denotemos por $e_i \in \mathbb{F}_2^n$ al vector con todos sus elementos ceros excepto un 1 en el elemento que ocupa la posición i.

Definición 4.8. *Una representación de Gröbner de* $\mathbb{F}_2^n/\mathcal{R}_\mathcal{C}$ *es un par* (N, ϕ) *donde:*

- N *está formado por el conjunto de representantes de las clases de equivalencia de* $\mathbb{F}_2^n/\mathcal{R}_\mathcal{C}$ *tal que* $0 \in N$ *y para cada* $\mathbf{n} \in N \setminus \{0\}$ *existe* e_i *con* $i \in \{1, \ldots, n\}$ *tal que* $\mathbf{n} = \mathbf{n}' + e_i$ *con* $\mathbf{n}' \in N$*.*

- *La aplicación* $\phi: \ N \times \{e_i\}_{i=1}^n \longrightarrow N$ *es una función que envía cada par* $(\mathbf{n}, e_i)$ *al elemento de* N *que representa la clase de equivalencia del elemento* $\mathbf{n} + e_i$.

El nombre de *representación de Gröbner* no es una casualidad, ya que si definimos el ideal

$$I(\mathcal{C}) = \langle \{\mathbf{x}^{\blacktriangle \mathbf{w}_1} - \mathbf{x}^{\blacktriangle \mathbf{w}_2} \mid \mathbf{w}_1 - \mathbf{w}_2 \in \mathcal{C}\} \rangle \subseteq \mathbb{K}[\mathbf{x}],$$

fijamos un orden $\succ$ compatible con el grado y calculamos una base de Gröbner $G_\succ$ del ideal $I(\mathcal{C})$ respecto de $\succ$, entonces la forma normal de cualquier monomio

$$\mathbf{x}^{\blacktriangle \mathbf{w}} = \prod_{i=1}^{n} x_i^{w_i}$$

se corresponde con el síndrome del vector $\mathbf{w} = (\blacktriangledown w_1, \ldots, \blacktriangledown w_n) \in \mathbb{F}_2^n$.
De esta forma podemos considerar el conjunto N de la Definición 4.8 como el conjunto de vectores $(\blacktriangledown w_1, \ldots, \blacktriangledown w_n) \in \mathbb{F}_2^n$ cuyo monomio asociado $\prod_{i=1}^n x_i^{w_i}$ es una forma normal respecto de la base de Gröbner $G_\succ$. Es decir, como el orden fijado es compatible con el grado, el conjunto N estaría formado por los representantes de peso mínimo de las clases de equivalencia de $\mathbb{F}_2^n/\mathcal{R}_\mathcal{C}$.
Y podemos considerar la aplicación ϕ como una *tabla de multiplicación* de los elementos de N y las variables x_i.
Con esta idea, la representación de Gröbner de un código puede ser calculada mediante una modificación del algoritmo FGLM (véase [18]). Una implementación de este algoritmo en GAP [30] se puede encontrar en [19].
Como hemos visto en las secciones anteriores, el ideal binomial $I(\mathcal{C})$ es equivalente al núcleo de un problema de programación lineal modular.
Asociado a una representación de Gröbner podemos definir el borde de un código (Para estudiarlo con más detalle véase [14]).

Definición 4.9. *Sea* $\mathcal{C} \subseteq \mathbb{F}_2^n$ *un código lineal binario,* $H_\mathcal{C}$ *su matriz de paridad y* (N, ϕ) *la representación de Gröbner de* $\mathbb{F}_2^n/\mathcal{R}_\mathcal{C}$. *Definimos el borde del código* $\mathcal{C}$ *respecto de* (N, ϕ) *como el conjunto:*

$$\begin{aligned} \mathcal{B}(\mathcal{C}) \ = \ & \{(\mathbf{n}_1 + e_i, \mathbf{n}_2) \mid i \in \{1, \ldots, n\}, \ \mathbf{n}_1 + e_i \neq \mathbf{n}_2, \ \mathbf{n}_1, \ \mathbf{n}_2 \in N \\ & y \ \ H_\mathcal{C} \ (\mathbf{n}_1 + e_i)^t \equiv H_\mathcal{C} \ \mathbf{n}_2^t \mod 2\} \end{aligned}$$

Observemos que las dos componentes de todo elemento del conjunto $\mathcal{B}(\mathcal{C})$ se encuentran en la misma clase de equivalencia de $\mathbb{F}_2^n/\mathcal{R}_\mathcal{C}$, por lo tanto la suma de ambas componentes forma una palabra del código $\mathcal{C}$.

En términos de la función ϕ podemos expresar el conjunto $\mathcal{B}(\mathcal{C})$ como:

$$\mathcal{B}(\mathcal{C}) = \{(\mathbf{n} + e_i, \phi(\mathbf{n}, e_i)) \mid i \in \{1, \ldots, n\},\ \mathbf{n} \in N \ \text{ y } \ \phi(\mathbf{n}, e_i) \neq \mathbf{n} + e_i\}.$$

Dado un código $\mathcal{C}$ y su correspondiente representación de Gröbner (N, ϕ), vamos a estudiar dos tipos de reducción asociadas a los dos algoritmos descritos en esta sección.
Definimos la reducción de un elemento $\mathbf{n} \in N$ respecto al vector $e_i \in \mathbb{F}_2^n$ como el elemento $\mathbf{n}' = \phi(\mathbf{n}, e_i) \in N$ y lo denotaremos por $\mathbf{n} \to_i \mathbf{n}'$.
Como toda palabra $\mathbf{y} \in \mathbb{F}_2^n$ se puede expresar como

$$\mathbf{y} = 0 + \sum_j e_{i_j} \ \text{ con } \ i_j \in \{1, \ldots, n\},$$

entonces al realizar un número finito de reducciones respecto de los vectores e_{i_j} con $i_j \in \{1, \ldots, n\}$, obtenemos el líder de la clase de equivalencia de $\mathbb{F}_2^n / \mathcal{R}_\mathcal{C}$ que contiene a $\mathbf{y}$. Este método nos permite definir el Algoritmo 15 de descodificación por gradiente.

Algoritmo 15 Algoritmo de reducción 1

INPUT: El vector recibido $\mathbf{y} \in \mathbb{F}_2^n$
OUTPUT: La palabra del código $\mathbf{c} \in \mathcal{C}$ más próxima al vector $\mathbf{y}$.
PASO ADELANTE:
Tenemos $\mathbf{y} = 0 + \sum_{j=1}^{s} e_{i_j}$.
Queremos calcular el elemento $\mathbf{n} \in N$ que se corresponde con el líder de la clase de equivalencia de $\mathbf{y}$.
$\mathbf{n} := 0$
for $j = 1, \ldots, s$ **do**
  $\mathbf{n}' = \phi(\mathbf{n}, e_i)$, es decir $\mathbf{n} \to_i \mathbf{n}'$.
  $\mathbf{n} := \mathbf{n}'$.
end for
PASO ATRÁS:
while $\mathbf{n} \neq 0$ **do**
  Calcular $\mathbf{y}'$ tal que $\mathbf{y}' = \mathbf{y} + e_{i_j}$ y $w_H(\mathbf{n}) \geq w_H(\phi(\mathbf{n}, e_{i_j}))$.
  $\mathbf{y} := \mathbf{y}'$
  $\mathbf{n} := \phi(\mathbf{n}, e_{i_j})$
end while
Devolver $\mathbf{c} = \mathbf{y} \in \mathcal{C}$

Observamos que el Algoritmo 15 muestra información redundante ya que una vez que conocemos el líder de la clase de equivalencia de $\mathbf{y}$, al final de la etapa **PASO ADELANTE** podemos descodificar $\mathbf{y}$.

Pero al escribirlo completo podemos comprobar más fácilmente el parecido con el Algoritmo 13, ya que la etapa **PASO ATRÁS** es exactamente el algoritmo de descodificación por gradiente propuesto por Liebler en [42].

Definición 4.10. *Sea (N, ϕ) la representación de Gröbner de $\mathbb{F}_2^n/\mathcal{R}_\mathcal{C}$ y fijemos un orden en los elementos del conjunto N dado por $\{\mathbf{n}_i\}_{i=1}^{2^{n-k}}$ con $\mathbf{n}_1 = 0$. Entonces definimos el par (N^*, ϕ^*) como:*

- $N^* = \{(i, \mathbf{w}_i) \in \mathbb{Z}_{\geq 0}^2 \mid \mathbf{w}_i = w_H(\mathbf{n}_i) \ \ con \ \ i \in \{1, \ldots, 2^{n-k}\}\}$.
- $$\begin{array}{rccc} \phi^* : & N^* \times \{e_i\}_{i=1}^n & \longrightarrow & N^* \\ & ((i, \mathbf{w}_i), e_j) & \longmapsto & \phi^*\left((i, \mathbf{w}_i), e_j\right) = (i_j, \mathbf{w}_{i_j}) \end{array}$$
  con $\mathbf{n}_{i_j} = \phi(\mathbf{n}_i, e_j)$ *y* $\mathbf{w}_{i_j} = w_H(\mathbf{n}_{i_j})$.

Es decir, fijamos un orden en los elementos del conjunto N y sustituimos los elementos de N por vectores del tipo $(i, \mathbf{w}_i)$ donde $\mathbf{w}_i$ es el peso del líder de la clase del elemento que ocupa la posición i-ésima respecto del orden definido en N.
Si descodificamos utilizando el par (N^*, ϕ^*) reducimos considerablemente el tamaño de memoria necesaria del Algoritmo 15, ya que basta con guardar un orden de las formas normales y el peso de uno cualquiera de los líderes de las clases de equivalencia correspondientes.
Si calculamos la representación de Gröbner mediante técnicas FGLM (véase [18]) obtenemos el par (N^*, ϕ^*), donde N^* contiene los elementos del conjunto N ordenados de forma creciente respecto del peso de sus líderes, es decir:

$$(i, \mathbf{w}_i), \ (j, \mathbf{w}_j) \in N^* \ \ \text{y} \ \ i < j \Rightarrow \mathbf{w}_i \leq \mathbf{w}_j.$$

Tras la etapa **PASO ADELANTE** conocemos el líder de la clase de equivalencia de la palabra recibida y su peso $\mathbf{w}_l$. Si denotamos por $t = \lfloor \frac{d-1}{2} \rfloor$ la capacidad correctora del código $\mathcal{C}$, tenemos que:

- Si $\mathbf{w}_l \leq t$ entonces la descodificación es correcta.
- Si $\mathbf{w}_l > t$ entonces la descodificación falla y como en este caso el líder de la clase de equivalencia de $\mathbb{F}_2^n/\mathcal{R}_\mathcal{C}$ que contiene a la palabra recibida no es único, la etapa **PASO ATRÁS** aporta distintas soluciones dependiendo de la elección del líder realizada.

Ahora vamos a considerar el borde del código $\mathcal{C}$, $\mathcal{B}(\mathcal{C})$, veremos que la información que nos aporta este conjunto nos permite realizar otra reducción sumando palabras adecuadas del código. Esta idea subyace en la descodificación por conjuntos de comprobación propuesta por Ashikhmin y A. Barg en [6].

Algoritmo 16 Algoritmo de reducción utilizando el par (N^*, ϕ^*)

INPUT: El vector recibido $\mathbf{y} \in \mathbb{F}_2^n$

OUTPUT: La palabra del código $\mathbf{c} \in \mathcal{C}$ más próxima al vector $\mathbf{y}$.

PASO ADELANTE:

Tenemos $\mathbf{y} = 0 + \sum_{j=1}^{s} e_{i_j}$.

Queremos calcular el índice $l \in \{1, \ldots, 2^{n-k}\}$ correspondiente con la posición que ocupa el representante de la clase de equivalencia de $\mathbf{y}$ en el conjunto N respecto del orden fijado.

$i := 1,\ \mathbf{w}_1 = 0$.

for $j = 1, \ldots, s$ **do**

$\phi^* ((i, \mathbf{w}_i), e_i) = (i', \mathbf{w}_{i'}$.

$(i, \mathbf{w}_i) := (i', \mathbf{w}_{i'})$.

end for

Devolver $l := i$

PASO ATRÁS:

while $i \neq 1$ **do**

Calcular $\mathbf{y}'$ tal que $\mathbf{y}' = \mathbf{y} + e_{i_j}$ y $\mathbf{w}_i \geq \mathbf{w}_{i'}$ donde $\mathbf{w}_{i'}$ representa a la segunda componente del elemento $\phi^* \left((i, \mathbf{w}_i), e_{i_j}\right)$

$\mathbf{y} := \mathbf{y}'$

$(i, \mathbf{w}_i) := \phi^* \left((i, \mathbf{w}_i), e_{i_j}\right)$

end while

Devolver $\mathbf{c} = \mathbf{y} \in \mathcal{C}$

Definición 4.11. *Sea* $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2) \in \mathcal{B}(\mathcal{C})$ *definimos* `head` *y* `tail` *del elemento* $\mathbf{b}$ *como:*

$$\texttt{head}(\mathbf{b}) = \mathbf{b}_1 \in \mathbb{F}_2^n \qquad \texttt{tail}(\mathbf{b}) = \mathbf{b}_2 \in \mathbb{F}_2^n$$

Observemos que para todo $\mathbf{b} \in \mathcal{B}(\mathcal{C})$, $\texttt{head}(\mathbf{b}) + \texttt{tail}(\mathbf{b})$ es una palabra del código $\mathcal{C}$. Ya que como las dos componentes de un elemento de $\mathcal{B}(\mathcal{C})$ pertenecen a la misma clase de equivalencia de $\mathbb{F}_2^n/\mathcal{R}_\mathcal{C}$, su suma pertenece al código $\mathcal{C}$.
La información del conjunto $\mathcal{B}(\mathcal{C})$ es redundante, así que podemos encontrar una subestructura llamada *borde reducido* (cuyo concepto es análogo al de base de Gröbner reducida) que nos permite realizar la misma reducción.

Definición 4.12. *Diremos que un conjunto* $\mathcal{R}(\mathcal{C})$ *es el borde reducido del código* $\mathcal{C}$ *si* $\mathcal{R}(\mathcal{C}) \subseteq \mathcal{B}(\mathcal{C})$ *y además se verifican las siguientes propiedades:*

1. *Para todo par* $(\mathbf{n}, e_i)$ *tal que* $\mathbf{n} + e_i$ *es* `head` *de un elemento de* $\mathcal{B}(\mathcal{C})$*, existe un elemento* $\mathbf{h} \in \mathcal{R}(\mathcal{C})$ *tal que*

$$\operatorname{supp}(\texttt{head}(\mathbf{h})) \subseteq \operatorname{supp}(\mathbf{n} + e_i).$$

2. *Dados dos elementos distintos* $\mathbf{b}_1,\ \mathbf{b}_2 \in \mathcal{R}(\mathcal{C})$ *se tiene que:*

$$\begin{array}{l} \operatorname{supp}(\texttt{head}(\mathbf{b}_1)) \not\subseteq \operatorname{supp}(\texttt{head}(\mathbf{b}_2)) \quad y \\ \operatorname{supp}(\texttt{head}(\mathbf{b}_2)) \not\subseteq \operatorname{supp}(\texttt{head}(\mathbf{b}_1)) \end{array}$$

Con la siguiente proposición queda probado que las palabras del código que se derivan del borde reducido de un código $\mathcal{C}$, $\mathcal{R}(\mathcal{C})$, forman un Test Set asociado dicho código.

Proposición 4.3. *Definimos el siguiente subconjunto de palabras del código* $\mathcal{C}$*:*

$$\operatorname{Min}_{\text{red}}(\mathcal{C}) = \{\texttt{head}(\mathbf{b}) + \texttt{tail}(\mathbf{b}) \mid \mathbf{b} \in \mathcal{R}(\mathcal{C})\} \subseteq \mathcal{C}.$$

Entonces $\operatorname{Min}_{red}(\mathcal{C}) = \mathcal{M}_\mathcal{C}$*, donde* $\mathcal{M}_\mathcal{C}$ *denota al conjunto de palabras de soporte mínimo de* $\mathcal{C}$*.*

Demostración. Sea $\mathbf{b} = (\texttt{head}(\mathbf{b}), \texttt{tail}(\mathbf{b})) \in \mathcal{R}(\mathcal{C})$, definimos

$$\mathbf{c} = \texttt{head}(\mathbf{b}) + \texttt{tail}(\mathbf{b}) \in \mathcal{C}.$$

Procedamos por reducción al absurdo suponiendo que $\mathbf{c} \notin \mathcal{M}_\mathcal{C}$. Entonces existe $\mathbf{c}' \in \mathcal{C} \setminus \{\mathbf{0}\}$ tal que $\operatorname{supp}(\mathbf{c}') \subset \operatorname{supp}(\mathbf{c})$.

Definimos el vector $\mathbf{c}_1 \in \mathbb{F}_2^n$ como el vector que verifica:

$$\mathrm{supp}(\mathbf{c}_1) = \mathrm{supp}(\mathbf{c}') \cap \mathrm{supp}(\mathtt{head}(\mathbf{b})).$$

Observemos que el vector $\mathbf{c}_1$ verifica que $\mathrm{supp}(\mathbf{c}_1) \subseteq \mathrm{supp}(\mathtt{head}(\mathbf{b}))$.
Entonces el vector $\mathbf{c}_2 = \mathbf{c}' + \mathbf{c}_1 \in \mathbb{F}_2^n$ verifica que:

$$\mathrm{supp}(\mathbf{c}_2) \subseteq \mathrm{supp}(\mathtt{tail}(\mathbf{b})).$$

En efecto, supongamos que $i \in \mathrm{supp}(\mathbf{c}_2)$, entonces tenemos dos posibilidades:

- $i \in \mathrm{supp}(\mathbf{c}')$ y $i \notin \mathrm{supp}(\mathbf{c}_1)$. De nuevo, como $i \in \mathrm{supp}(\mathbf{c}')$, tenemos dos posibilidades:
  - $i \in \mathrm{supp}(\mathtt{head}(\mathbf{b}))$ y $i \notin \mathrm{supp}(\mathtt{tail}(\mathbf{b}))$, es decir $i \in \mathrm{supp}(\mathbf{c}_1)$, lo que contradice la hipótesis realizada.
  - $i \notin \mathrm{supp}(\mathtt{head}(\mathbf{b}))$ y $i \in \mathrm{supp}(\mathtt{tail}(\mathbf{b}))$ lo que verifica nuestra suposición.
- $i \notin \mathrm{supp}(\mathbf{c}')$ y $i \in \mathrm{supp}(\mathbf{c}_1)$. Podemos proceder de forma análoga al caso anterior con $i \notin \mathrm{supp}(\mathbf{c}')$ obteniendo el mismo resultado.

Definimos el vector $\mathbf{m} = (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{F}_2^{2n}$ como el vector tal que $\mathtt{head}(\mathbf{m}) = \mathbf{c}_1$ y $\mathtt{tail}(\mathbf{m}) = \mathbf{c}_2$. Es fácil comprobar que $\mathbf{m} \in \mathcal{B}(\mathcal{C})$ ya que se verifica que:

- $\mathtt{head}(\mathbf{m}) \neq \mathtt{tail}(\mathbf{m})$, en caso contrario se tendría que $\mathbf{c}_1 = \mathbf{c}_2$ lo que supondría que $\mathbf{c}' = \mathbf{c}_1 + \mathbf{c}_2 = \mathbf{0}$.
- $\mathtt{head}(\mathbf{m})$ y $\mathtt{tail}(\mathbf{m})$ tienen el mismo síndrome pues se tiene que:

$$0 = H_{\mathcal{C}}\ \mathbf{c}'^t = H_{\mathcal{C}}(\mathbf{c}_1 + \mathbf{c}_2)^t.$$

Por lo tanto tenemos un elemento $\mathbf{m} \in \mathcal{B}(\mathcal{C})$ tal que

$$\mathrm{supp}(\mathtt{head}(\mathbf{m})) \subset \mathrm{supp}(\mathtt{head}(\mathbf{b})),$$

lo que contradice el hecho de que $\mathcal{R}(\mathcal{C})$ sea un borde reducido de $\mathcal{C}$.

□

Por lo tanto $\mathrm{Min}_{\mathrm{red}}(\mathcal{C})$ es un Test Set minimal que permite efectuar el Algoritmo 14.
Observemos también que este Test Set se corresponde con el Test Set dado por el problema de programación linear modular asociado al código $\mathcal{C}$.
Por lo tanto, queda demostrado que los dos algoritmos de descodificación por gradiente presentados son duales en el sentido de que ambos pueden derivarse de la representación de Gröbner de un código binario.
La generalización a códigos no binarios no es directa y será una de las líneas de investigación que se seguirá en el futuro.

4.3. Conjunto de palabras de soporte mínimo

Consideramos un código lineal $\mathcal{C}$ definido sobre el alfabeto $\mathbb{Z}_q$ con longitud n y dimensión k, es decir, se trata de un subespacio lineal $\mathcal{C} \subseteq \mathbb{Z}_q^n$. A los elementos $\mathbf{c} \in \mathcal{C}$ se les denomina palabras del código y definimos soporte de la palabra $\mathbf{c} = (c_1, \ \dots \ , c_n)$ como su soporte como vector de $\mathbb{Z}_q^n$, es decir, $\text{supp}(\mathbf{c}) = \{i \mid c_i \neq 0\}$.
Diremos que una palabra $\mathbf{c} \in \mathcal{C}$ es una palabra de soporte mínimo si se trata de una palabra distinta del cero y su soporte no está contenido en ninguna otra palabra del código $\mathcal{C}$.

Definición 4.13. *Dado un código lineal $\mathcal{C} \subseteq \mathbb{Z}_q^n$, una palabra $\mathbf{c} \in \mathcal{C}$ es mínima si para toda palabra $\mathbf{c}' \in \mathcal{C}$ tal que $\text{supp}(\mathbf{c}') \subseteq \text{supp}(\mathbf{c})$, existe una constante $\lambda \in \mathbb{Z}$ tal que $\mathbf{c}' = \lambda\mathbf{c}$*

Lema 4.1. *Dos palabras de soporte mínimo del código $\mathcal{C} \subseteq \mathbb{Z}_q^n$ con el mismo soporte tienen que ser una múltiplo de la otra.*

Demostración. Consideramos $\mathbf{m} = (m_1, \dots, m_n) \in \mathcal{C}$ una palabra de soporte mínimo.

- Supongamos que existe un divisor de cero entre los elementos del soporte de la palabra $\mathbf{m}$, es decir existe $i \in \{1, \dots, n\}$ y $\alpha \in \mathbb{Z}_q \setminus \{0\}$ tal que $m_i \neq 0$ y $\alpha m_i = 0$. Entonces se tendría que $\alpha\mathbf{m} \in \mathcal{C}$ y que $\text{supp}(\alpha\mathbf{m}) \subseteq \text{supp}(\mathbf{m})$, lo que contradice la minimalidad de $\mathbf{m}$, excepto en el caso de que $\alpha\mathbf{m} = 0$.

- Supongamos que existe más de un divisor de cero entre los elementos del soporte de la palabra $\mathbf{m}$, veamos que en tal caso dichos elementos tienen que ser iguales. Procedamos por reducción al absurdo suponiendo que existen dos elementos divisores de cero diferentes en el soporte de $\mathbf{m}$, es decir $0 \neq m_{i_1} \neq m_{i_2} \neq 0$. Podemos suponer, sin pérdida de generalidad, que $\blacktriangle m_{i_1} < \blacktriangle m_{i_2}$. Recordemos que los divisores de cero en $\mathbb{Z}_q$ son todos los divisores de q, por lo tanto podemos definir $\beta = \frac{q}{\blacktriangle m_{i_1}}$. Es fácil comprobar que $\blacktriangledown\beta\mathbf{m} \neq 0$ y sin embargo $\text{supp}(\blacktriangledown\beta\mathbf{m}) \subset \text{supp}(\mathbf{m})$, lo que contradice la minimalidad de $\mathbf{m}$.

- Supongamos que no existe ningún divisor de cero entre los elementos del soporte de la palabra de $\mathbf{m}$, por lo tanto todos los elementos del soporte de $\mathbf{m}$ son unidades de $\mathbb{Z}_q$. Es decir que para todo m_i con $i \in \{1, \dots, n\}$ existe $m_i^{-1} \in \mathbb{Z}_q$ tal que $m_i m_i^{-1} = 1$.

Sea $\mathbf{m}' = (m'_1, \ldots, m'_n) \in \mathcal{C}$ otra palabra de soporte mínimo del código con $\mathrm{supp}(\mathbf{m}) = \mathrm{supp}(\mathbf{m}')$ y sean $m_i,\ m'_i$ dos elementos del soporte de $\mathbf{m}$ y $\mathbf{m}'$ respectivamente. En este caso podemos definir la palabra

$$\mathbf{c} = \mathbf{m} - \frac{m_i}{m'_i}\mathbf{m}' \in \mathcal{C}$$

que verifica que $\mathrm{supp}(\mathbf{c}) \subset \mathrm{supp}(\mathbf{m})$, lo que contradice la minimalidad de $\mathbf{m}$. Luego $\mathbf{c} = \mathbf{m} - \frac{m_i}{m'_i}\mathbf{m}' = 0$ de donde se deduce el resultado.

□

El siguiente teorema nos sugiere un algoritmo para calcular el conjunto de palabras minimales de un código lineal arbitrario definido sobre $\mathbb{Z}_q^n$.

Teorema 4.1. *La base de Graver asociada a la matriz de paridad del código $\mathcal{C} \subseteq \mathbb{Z}_q^n$, que denotaremos por $\mathrm{Gr}_{H_\mathcal{C}}$, se corresponde con el conjunto de palabras de soporte mínimo del código $\mathcal{C} \subseteq \mathbb{Z}_q^n$.*

Demostración.

1. Veamos que todas las palabras de soporte mínimo del código $\mathcal{C}$ están en la base de Graver asociada al código $\mathcal{C}$, o bien existe en la base de Graver otra palabra del código con el mismo soporte.

   Sea $\mathbf{m} \in \mathcal{C}$ una palabra de soporte mínimo. Sabemos, por el Lema 4.1, que si existe otra palabra con el mismo soporte en el código entonces una tiene que ser múltiplo de la otra, en cuyo caso elegiremos uno o varios representantes del conjunto de palabras que tienen el mismo soporte que la palabra $\mathbf{m}$ cuyos vectores asociados en $\mathbb{Z}^n$ sean mínimos respecto al orden $\sqsubset$.

   Procedamos por reducción al absurdo suponiendo que $\blacktriangle\mathbf{m} \notin \mathrm{Gr}_{H_\mathcal{C}}$, es decir, que el binomio asociado a dicha palabra, $\mathbf{x}^{(\blacktriangle\mathbf{m})^+} - \mathbf{x}^{(\blacktriangle\mathbf{m})^-}$ no es primitivo.

   Entonces existe un binomio $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} \in \blacktriangle I$ tal que $\mathbf{u} \sqsubset \blacktriangle\mathbf{m}$ de forma estricta. Además por definición del ideal $\blacktriangle I$ tenemos que $\blacktriangledown\mathbf{u} \in \mathcal{C}$, lo que contradice la minimalidad de la palabra $\mathbf{m}$.

2. Veamos que en la base de Graver $Gr_{H_\mathcal{C}}$ sólo hay palabras de soporte mínimo del código $\mathcal{C}$.

   Procedamos por reducción al absurdo suponiendo que existe una palabra del código $\mathbf{m}$ que no tiene soporte mínimo pero, sin embargo, su binomio asociado $\mathbf{x}^{\blacktriangle\mathbf{m}^+} - \mathbf{x}^{\blacktriangle\mathbf{m}^-} \in \mathrm{Gr}_{H_\mathcal{C}}$.

   Es decir, existe una palabra $\mathbf{m}' \in \mathcal{C}$ tal que: $\mathrm{supp}(\mathbf{m}') \subseteq \mathrm{supp}(\mathbf{m})$.

- En el caso de que $\blacktriangle\mathbf{m}$ y $\blacktriangle\mathbf{m}'$ sean comparables respecto del orden $\sqsubset$, entonces tendríamos que $\blacktriangle\mathbf{m}' \sqsubset \blacktriangle\mathbf{m}$, por lo tanto el binomio asociado al vector $\blacktriangle\mathbf{m}$ no sería primitivo, lo que contradice el hecho de pertenezca a $\mathrm{Gr}_{H_\mathcal{C}}$.
- En el caso de que $\blacktriangle\mathbf{m}$ y $\blacktriangle\mathbf{m}'$ no sean comparables respecto del orden $\sqsubset$, entonces definimos la siguiente aplicación:

$$\begin{array}{rccc} \blacksquare_{\mathbf{m}}: & \mathbb{Z}_q^n & \longrightarrow & \mathbb{Z}_q^n \\ & \mathbf{a}=(a_1,\ldots,a_n) & \longmapsto & \blacksquare_{\mathbf{m}}(\mathbf{a})=\mathbf{b}=(b_1,\ldots,b_n) \end{array}$$

Donde $b_i = \begin{cases} (q-1)a_i & \text{Si } m_i a_i \geq 0 \\ (-1)a_i & \text{En caso contrario.} \end{cases}$

Es fácil ver que si $\mathbf{m}' \in \mathcal{C}$, entonces $\blacksquare_{\mathbf{m}}(\mathbf{m}') \in \mathcal{C}$. Además se tiene que: $\mathrm{supp}(\mathbf{m}') = \mathrm{supp}(\blacksquare_{\mathbf{m}}(\mathbf{m}'))$.

Luego hemos encontrado un vector, $\blacktriangle\blacksquare_m(\mathbf{m}')$, cuyo vector asociado pertenece al ideal $\blacktriangle I$ y además verifica que $\blacktriangle\blacksquare_{\mathbf{m}}(\mathbf{m}') \sqsubset \blacktriangle\mathbf{m}$, lo que contradice el hecho de que el binomio asociado a la palabra $\mathbf{m}$ pertenezca a $\mathrm{Gr}_{H_\mathcal{C}}$.

□

En particular, el resultado anterior nos aporta un algoritmo para calcular el conjunto de palabras minimales de un código lineal arbitrario definido sobre $\mathbb{F}_q$ con q un número primo. Pero no funciona para el caso $q = p^r$, ya que $\mathbb{F}_{p^r} \not\equiv \mathbb{Z}_{p^r}$.

Corolario 4.1. *Dado un código lineal $\mathcal{C} \subseteq \mathbb{Z}_q^n$ y sea $H_\mathcal{C}$ una matriz de paridad de dicho código. El conjunto de palabras minimales del código $\mathcal{C}$ está formado por la proyección de las primeras n coordenadas del conjunto:*

$$\{\mathbf{w} \in \mathbb{Z}^{2n} \mid \mathbf{w} = (\mathbf{v}, (q-1)\mathbf{v}) \in \mathcal{T}\},$$

donde $\mathcal{T}$ es un Test Set obtenido a partir de la base de Gröbner reducida del problema lineal entero modular asociado a la matriz:

$$\begin{pmatrix} H_\mathcal{C} & 0 \\ I_n & I_n \end{pmatrix}$$

Corolario 4.2. *El conjunto de palabras minimales del código lineal $\mathcal{C} \subseteq \mathbb{Z}_q^n$ se puede calcular a partir del ideal*

$$\left\langle \{\mathbf{x}^{\blacktriangle\mathbf{w}_1}\mathbf{z}^{\blacktriangle\mathbf{w}_1(q-1)}, \ \ldots\ , \mathbf{x}^{\blacktriangle\mathbf{w}_k}\mathbf{z}^{\blacktriangle\mathbf{w}_k(q-1)}\} \cup \{x_i^q - 1\}_{i=1}^n \cup \{z_i^q - 1\}_{i=1}^n \right\rangle,$$

donde $\{\mathbf{w}_1, \ \ldots\ , \mathbf{w}_k\} \subseteq \mathbb{Z}_q^n$ representa las filas de una matriz generatriz del código $\mathcal{C}$.

Ejemplo 4.3. *Consideremos el código binario de Hamming* $\mathcal{C} \subseteq \mathbb{F}_2^7$ *de parámetros* $[7, 4, 3]$.
Los códigos de Hamming fueron propuestos por Richard Hamming en 1950 *y están basados en la construcción de códigos lineales con distancia mínima fija. Es decir, dado un entero* $m \in \mathbb{Z}$ *podemos construir un código de Hamming sobre el cuerpo* $\mathbb{F}_2$ *de longitud* $2^m - 1$, *que contiene* m *símbolos redundantes y* $2^m - 1 - m$ *símbolos de información. Si se enumeran los símbolos desde* 1 *hasta* $2^m - 1$, *entonces aquellos símbolos que se encuentran en la posición* 2^k *con* $k \in \{0, \ldots, m-1\}$ *son símbolos redundantes y el resto son símbolos de información. El valor de cada símbolo redundante se escoge como el total de* 1 *que hay en un conjunto específico de símbolos de información, y el conjunto de símbolos de información que se forman se escoge de tal manera que cubran todas las combinaciones de* m *símbolos de información distintas posibles. Un mismo símbolo de información puede estar afectando a dos símbolos redundantes distintos, pero nunca puede haber un símbolo de información que no afecte a ningún símbolo redundante, lo que proporciona al código una gran capacidad de corrección.*
De esta forma el código de Hamming propuesto, que presenta distancia mínima 3 *y longitud* $7 = 2^3 - 1$, *sabemos que tiene* 3 *símbolos redundantes y* 4 *símbolos de información. Dada una palabra* $\mathbf{c} = (c_1, \ldots, c_7) \in \mathcal{C}$, *los símbolos* c_1, c_2 *y* c_4 *son símbolos redundantes que verifican las siguientes condiciones:*

$$\begin{aligned} c_1 &\equiv c_3 + c_5 + c_7 \mod 2 \\ c_2 &\equiv c_3 + c_6 + c_7 \mod 2 \\ c_4 &\equiv c_5 + c_6 + c_7 \mod 2 \end{aligned}$$

Es fácil comprobar que este código permite detectar hasta dos errores que se produzcan en la transmisión, pero sólo permite corregir un error.
La matriz generatriz $G_{\mathcal{C}} \in \mathbb{F}_2^{4\times 7}$ *y de paridad* $H_{\mathcal{C}} \in \mathbb{F}_2^{3\times 7}$ *de este código son:*

$$G_{\mathcal{C}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \quad ; \quad H_{\mathcal{C}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Este código tiene 16 *palabras de peso* 0, 3, 4 *ó* 7*; Además todas las palabras del código son de soporte mínimo excepto las palabras correspondientes a* 0 *y* 1.

- *En primer lugar calculamos una base de Gröbner reducida del ideal asociado al código $\mathcal{C}$. Observamos que obtenemos la siguiente base de Gröbner reducida:*

$$\{x_3x_7+x_1, x_1x_7+x_3, x_5x_6+x_1, x_4x_6+x_3, x_3x_6+x_4, x_2x_6+x_7, x_1x_6+x_5, x_4x_5+x_7, x_3x_5+x_2, x_2x_5+x_3, x_1x_5+x_6, x_3x_4+x_6, x_2x_4+x_1, x_1x_4+x_2, x_2x_3+x_5, x_1x_3+x_7, x_1x_2+x_4\} \cup \{x_i^2-1\}_{i=1}^7$$

Es decir obtenemos 7 palabras del código de peso 3:

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

- *Al calcular una base de Graver de la elevación de Lawrence de la matriz de paridad obtenemos 155 binomios que representan las 7 palabras que ya obtuvimos en la base de Gröbner reducida y además 7 palabras de peso 4. Es decir, obtenemos:*

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

Exponemos a continuación el código utilizado en el programa `Sage` *(véase [54]) que nos permitió efectuar los cálculos anteriores.*

- ***Código para obtener una base de Gröbner reducida del ideal asociado al código de Hamming de parámetros*** $[7,4,3]$**.**

```
M = MatrixSpace(GF(2),4,7)

G=M([[1,0,0,0,1,1,0],[0,1,0,0,0,1,1],[0,0,1,0,1,1,1],
[0,0,0,1,1,0,1]])

C = LinearCode(G); C
    Linear code of length 7, dimension 4 over Finite
    Field of size 2

H = C.check_mat(); H
    [1 0 0 1 0 1 1]
    [0 1 0 1 1 1 0]
    [0 0 1 0 1 1 1]
```

```
n = C.length(); n
    7

k = C.dimension(); k
    4

L1=['x%s' %p for p in range(1,n+1)]; L1
    ['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7']

R=PolynomialRing(QQ,L1); R
    Multivariate Polynomial Ring in x1, x2, x3, x4,
    x5, x6, x7 over Rational Field

R.inject_variables()
    Defining x1, x2, x3, x4, x5, x6, x7

p=[1 for i in range(0,k)]; p
    [1, 1, 1,1]

for j in range(0,k):
    for i in range(1,n+1):
        q1 = 'x%s^%s' %(i,G[j][i-1])
        f=sage_eval(q1, locals={'x1':x1, 'x2':x2, 'x3':x3,
        'x4':x4, 'x5':x5, 'x6':x6, 'x7':x7})
        p[j]=p[j]*f
    p[j]=p[j]-1

p
    [x1*x2*x3 - 1, x1*x4*x5 - 1, x2*x4*x6 - 1,
    x1*x2*x4*x7 - 1]

L2=['x%s^2-1' %j for j in range(1, n+1)];
L3=[p[j] for j in range(0,k)];
L3.extend(L2);

I = R.ideal(L3);
sI=singular(I); sI
    x1*x5*x6-1,
    x2*x6*x7-1,
    x3*x5*x6*x7-1,
```

```
    x4*x5*x7-1,
    x1^2-1,
    x2^2-1,
    x3^2-1,
    x4^2-1,
    x5^2-1,
    x6^2-1,
    x7^2-1

B=sI.groebner(); B
    x7^2-1,
    x6*x7-x2,
    x5*x7-x4,
    x4*x7-x5,
    x3*x7-x1,
    x2*x7-x6,
    x1*x7-x3,
    x6^2-1,
    x5*x6-x1,
    x4*x6-x3,
    x3*x6-x4,
    x2*x6-x7,
    x1*x6-x5,
    x5^2-1,
    x4*x5-x7,
    x3*x5-x2,
    x2*x5-x3,
    x1*x5-x6,
    x4^2-1,
    x3*x4-x6,
    x2*x4-x1,
    x1*x4-x2,
    x3^2-1,
    x2*x3-x5,
    x1*x3-x7,
    x2^2-1,
    x1*x2-x4,
    x1^2-1
```

- ***Código para obtener una base de Graver asociada a la elevación de Lawrence de la matriz de paridad de código de Hamming de parámetros*** $[7, 4, 3]$***.***

```
M = MatrixSpace(GF(2),4,7)

G=M([[1,0,0,0,1,1,0],[0,1,0,0,0,1,1],[0,0,1,0,1,1,1],
[0,0,0,1,1,0,1]])

C = LinearCode(G); C
    Linear code of length 7, dimension 4 over Finite
    Field of size 2

H = C.check_mat(); H
    [1 0 0 1 0 1 1]
    [0 1 0 1 1 1 0]
    [0 0 1 0 1 1 1]

n = C.length(); n
    7

k = C.dimension(); k
    4

L1=['x%s' %p for p in range(1,n+1)]; L1
    ['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7']
L2=['y%s'%p for p in range (1,n+1)]; L2
    ['y1', 'y2', 'y3', 'y4', 'y5', 'y6', 'y7']
L1.extend(L2);

R=PolynomialRing(QQ,L1); R
    Multivariate Polynomial Ring in x1, x2,
    x3, x4, x5, x6, x7, y1, y2, y3,y4, y5, y6,
    y7 over Rational Field

R.inject_variables()
    Defining x1, x2, x3, x4, x5, x6, x7,
    y1, y2, y3, y4, y5, y6, y7

p=[1 for i in range(0,k)]; p
```

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

for j in range(0,k):
    for i in range(1,n+1):
        q1 = 'x%s^%s' %(i,G[j][i-1])
        f=sage_eval(q1, locals={'x1':x1, 'y1':y1,
        'x2':x2, 'y2':y2,'x3':x3, 'y3':y3,'x4':x4, 'y4':y4,
        'x5':x5, 'y5':y5,'x6':x6, 'y6':y6,'x7':x7, 'y7':y7})
        q2 = 'y%s^%s' %(i,G[j][i-1])
        g=sage_eval(q2, locals={'x1':x1, 'y1':y1,
        'x2':x2, 'y2':y2,'x3':x3, 'y3':y3,'x4':x4, 'y4':y4,
        'x5':x5, 'y5':y5,'x6':x6, 'y6':y6,'x7':x7, 'y7':y7})
        p[j]=p[j]*f*g
    p[j]=p[j]-1

p
    [x1*x5*x6*y1*y5*y6 - 1, x2*x6*x7*y2*y6*y7 - 1,
    x3*x5*x6*x7*y3*y5*y6*y7 -1, x4*x5*x7*y4*y5*y7 - 1]

L3=['x%s^2-1' %j for j in range(1, n+1)]; L3
L5=['y%s^2-1' %j for j in range(1, n+1)]; L5
L4=[p[j] for j in range(0,k)]; L4
L4.extend(L3); L4
L4.extend(L5); L4

I = R.ideal(L4); I
sI=singular(I); sI
    x1*x5*x6*y1*y5*y6-1,
    x2*x6*x7*y2*y6*y7-1,
    x3*x5*x6*x7*y3*y5*y6*y7-1,
    x4*x5*x7*y4*y5*y7-1,
    x1^2-1,
    x2^2-1,
    x3^2-1,
    x4^2-1,
    x5^2-1,
    x6^2-1,
    x7^2-1,
    y1^2-1,
    y2^2-1,
```

```
        y3^2-1,
        y4^2-1,
        y5^2-1,
        y6^2-1,
        y7^2-1

    B=sI.groebner(); B
        WARNING: Output truncated!
        full_output.txt
        y7^2-1,
        y6^2-1,
        y5^2-1,
        y4^2-1,
        y3^2-1,
        y2^2-1,
        y1^2-1,
        x7^2-1,
        x6^2-1,
        x5^2-1,
        x4^2-1,
        x3^2-1,
        x2^2-1,
        x1^2-1,
        x7*y2*y6-x2*x6*y7,
        x6*y2*y6-x2*x7*y7,
        x2*y2*y6-x6*x7*y7,
        x6*x7*y6-x2*y2*y7,
        x2*x7*y6-x6*y2*y7,
        x2*x6*y6-x7*y2*y7,
        x7*y4*y5-x4*x5*y7,
        x5*y4*y5-x4*x7*y7,
        x4*y4*y5-x5*x7*y7,
        x6*y1*y5-x1*x5*y6,
        x5*y1*y5-x1*x6*y6,
        x1*y1*y5-x5*x6*y6,
        ...
        x2*x3*x6*y1-x1*y2*y3*y6,
        x3*x4*x5*y1-x1*y3*y4*y5,
        x3*x5*x6*x7-y3*y5*y6*y7,
        x1*x4*x6*x7-y1*y4*y6*y7,
```

```
x1*x2*x5*x7-y1*y2*y5*y7,
x2*x3*x4*x7-y2*y3*y4*y7,
x2*x4*x5*x6-y2*y4*y5*y6,
x1*x2*x3*x6-y1*y2*y3*y6,
x1*x3*x4*x5-y1*y3*y4*y5
```

Estudiamos en el siguiente ejemplo un código no binario.

Ejemplo 4.4. *Consideremos un código* $\mathcal{C} \subseteq \mathbb{F}_3^4$ *con matriz generatriz* $G_{\mathcal{C}}$*, que además coincide con su matriz de paridad,* $H_{\mathcal{C}}$*:*

$$G_{\mathcal{C}} = H_{\mathcal{C}} = \begin{pmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

El código $\mathcal{C}$ *tiene* $3^2 = 9$ *palabras que listamos a continuación; además observamos que todas sus palabras, excepto la palabra* 0*, son palabras de soporte mínimo:*

$$\mathcal{C} = \{(0,0,0,0), (1,0,2,1), (2,0,1,2), (0,1,1,1), (1,1,0,2), (2,1,2,0), (0,2,2,2), (1,2,1,0), (2,2,0,1)\}$$

- *Si Calculamos una base de Gröbner del ideal asociado al código* $\mathcal{C}$*, obtenemos el siguiente conjunto de binomios:*

$$\{x_1x_4 - x_3, x_3^2 - x_2x_4, x_2x_3 - x_4^2, x_1x_3 - x_2, x_2^2 - x_3x_4, x_1x_2 - x_4, x_3x_4^2 - x_1, x_2x_4^2 - x_1^2, x4^3 - 1, x1^3 - 1\}$$

  que se corresponden con las palabras del código:

$$\{(1,0,2,1), (2,0,1,2), (0,1,1,1), (1,1,0,2), (0,2,2,2), (1,2,1,0)\}$$

- *Si ahora calculamos una base de Graver asociada a la elevación de Lawrence de la matriz de paridad del código, obtenemos* 99 *binomios que se corresponden con las* 6 *palabras que ya nos habían aparecido en el apartado anterior y, además, nos aparecen representadas las otras* 2 *palabras de soporte mínimo del código que faltaban* $(2,1,2,0)$ *y* $(2,2,0,1)$*.*

4.4. Esquemas para compartir secretos

El problema del *Reparto de Secretos* plantea la forma de distribuir una información secreta entre un conjunto de participantes, de modo que sólo los

subconjuntos del colectivo de participantes autorizados puedan reconstruir el secreto a partir de sus fragmentos.
Vamos a analizar brevemente algunos de los esquemas que han sido utilizados para resolver el problema del Reparto de secretos, centrándonos en su formulación en términos de la Teoría de Códigos.
Formalmente el problema que se plantea es dividir un conjunto de datos secretos $K = \{k_1, \ldots, k_s\}$ entre los distintos participantes $P = \{p_1, \ldots, p_l\}$, de manera que sólo los subconjuntos de P autorizados puedan reconstruir el valor secreto, mientras que los participantes de un subconjunto no autorizado no puedan obtener ninguna información sobre el secreto a partir del valor de sus fragmentos.
Para describir el problema vamos a fijar la siguiente notación:

- $P = \{p_1, \ldots, p_l\}$ denota al conjunto de personas que participan en el esquema.
- $D \notin P$ representa a un participante especial, llamado el *distribuidor*, que se encarga de administrar los distintos fragmentos del secreto al conjunto de participantes y de mantener oculto dicho secreto.
- $\mathcal{K}$ engloba al conjunto de secretos que hay que repartir.
- $\Gamma \subseteq 2^P$ define la familia de subconjuntos autorizados y se le denomina *estructura de acceso* del esquema. Además se dice que Γ es monótona ya que:
$$\text{Si} \;\; \gamma_1 \in \Gamma \;\; \text{y} \;\; \gamma_1 \subseteq \gamma_2 \Rightarrow \gamma_2 \in \Gamma.$$
La estructura de acceso queda determinada por el conjunto de sus agrupaciones autorizadas minimales, que se denomina *base de la estructura de acceso*, y se denota por Γ_0. Es decir, se trata del conjunto más pequeño que pertenece a Γ y que verifica la propiedad:
$$\forall \gamma_1, \;\; \gamma_2 \in \Gamma_0, \quad \text{ni} \;\; \gamma_1 \not\subset \gamma_2 \;\; \text{ni} \;\; \gamma_2 \not\subset \gamma_1.$$
- S es el conjunto de fragmentos en los que podemos dividir el secreto entre los distintos participantes.

Definición 4.14. *Un esquema de compartir secretos se representa como un conjunto de* reglas de distribución *que nos permite asignar fragmentos del secreto a los distintos participantes del esquema. Es decir, está formado por el conjunto aplicaciones definidas de la siguiente forma:*

$$\begin{array}{rccc} f: & P \cup \{D\} & \longrightarrow & S \cup \mathcal{K} \\ & p_i & \longmapsto & f(p_i) = s_i \in S \\ & D & \longmapsto & f(D) = K \in \mathcal{K} \end{array}$$

Al conjunto de reglas de distribución, que es de conocimiento público, se denota por $\mathcal{J}$. Si todas las funciones de $\mathcal{J}$ son lineales entonces diremos que se trata de un esquema lineal.
Para cada secreto $K \in \mathcal{K}$ consideramos el conjunto de reglas

$$\mathcal{J}_K = \{f \in \mathcal{J} \mid f(D) = K\}.$$

Definición 4.15. *Diremos que un esquema definido por un conjunto de reglas $\mathcal{J}$ es* perfecto *para realizar una estructura de acceso Γ, si satisface las siguientes condiciones:*

1. *Todo subconjunto autorizado puede determinar el valor secreto. Es decir, los fragmentos asignados a un subconjunto autorizado determinan de forma única el valor secreto. De una manera más formal, para cualquier $\gamma \in \Gamma$ y para toda función $f_1,\ f_2 \in \mathcal{J}$ que verifican que $f_1(p_i) = f_2(p_i)$ para todo $p_i \in \gamma$, entonces se tiene que $f_1(D) = f_2(D)$.*

2. *Si un subconjunto no está autorizado no puede obtener ninguna información sobre el secreto. Dicho de otra forma, si $\delta \notin \Gamma$ y $f \in \mathcal{J}$, entonces existe $\lambda(f, \delta) \in \mathbb{Z}_{\geq 0}$ tal que para cada $K \in \mathcal{K}$ se tiene que:*

   $$|\{g \in \mathcal{J}_K \mid g(p_i) = f(p_i)\ \forall p_i \in \delta\}| = \lambda(f, \delta)$$

   Es decir, existen $\lambda(f, \delta)$ posibles reglas de distribución compatibles con la distribución fijada $f \in \mathcal{J}$ para cada posible valor secreto.

Definición 4.16. *Diremos que un esquema con estructura de acceso Γ es un* (t, l)-esquema umbral *si el conjunto de participantes P del esquema tiene cardinal l y si todo subconjunto que contenga al menos t participantes es un conjunto autorizado, es decir, para todo $B \subseteq 2^P$, $B \in \Gamma$ si y sólo si $|B| \geq t$.*

La *eficiencia* de un esquema se mide a través de la *Tasa de información*, que definimos como la relación que existe entre la longitud de la representación binaria del conjunto $\mathcal{K}$ y del conjunto S. Es decir, podemos realizar la siguiente definición:

Definición 4.17. *Denotamos por tasa de información de cada participante $p_i \in P$ como el cociente: $\rho_i = \frac{\log_2 |\mathcal{K}|}{\log_2 |s_i|}$ y como tasa de información del esquema a $\rho = \min\{\rho_i \mid i \in \{1, \ldots, l\}\}$*

Definición 4.18. *Diremos que un esquema es* ideal *si se verifica que $\rho = 1$.*

Proposición 4.4. *Todo esquema perfecto verifica que $\rho \leq 1$*

Demostración. Véase [2], Lema 2. □

4.4.1. Esquema de Shamir

Este esquema definido en [56] fue uno de los primeros esquemas expuestos para compartir secretos. El problema que pretende resolver es un (t, l)-esquema umbral, es decir trata de repartir un secreto entre un conjunto de l participantes de manera que todo subconjunto de t o más participantes pueda recuperar el secreto, mientras que el resto de subconjuntos de participantes no obtiene ninguna información.
Si consideramos los enteros positivos l, s y $t \leq l$, un conjunto de participantes $P = \{p_1, \ldots, p_l\}$ y un secreto $K \in \{0, \ldots, s-1\}$, entonces la idea general de este esquema es la siguiente:

1. Definimos p como el menor número primo que sea mayor a los enteros s y $l+1$.

2. Escogemos de forma aleatoria y diferentes dos a dos los elementos

$$a_1, \ldots, a_{t-1} \in \mathbb{Z}_p.$$

3. Construimos el polinomio $q(x) = K + \sum_{i=1}^{t-1} a_i x^i$ de grado $t-1$.

4. Asignamos a cada participante el fragmento del secreto determinado por $s_j = q(j) \in \mathbb{Z}_p$ con $j \in \{1, \ldots, l\}$.

Por el Teorema de Interpolación de Lagrange sabemos que t personas del conjunto de participantes pueden recuperar el polinomio $q(x)$, de forma que podrían recuperar el secreto ya que $q(0) = K$, mientras que $t-1$ personas no obtendrían ninguna información sobre el secreto.

4.4.2. Esquema de Blakley

Blakely en [11] fue el precursor del tipo de construcciones de esquemas para compartir secretos de tipo geométrico.
La idea general de esta construcción es considerar una recta, que es de dominio público y que pasa por el secreto, en el espacio afín y un hiperplano no paralelo a dicha recta, tal que la intersección entre ambos sea el secreto. Los fragmentos que se reparten del secreto entre los participantes son puntos del hiperplano, de forma que los conjuntos autorizados describan una variedad afín contenida en el hiperplano, tal que al hacer la intersección con la recta pública recuperen el secreto.

4.4.3. Esquema basado en códigos lineales

Consideremos un código lineal $\mathcal{C} \subseteq \mathbb{F}_q$ de parámetros $[n, k, d]$ con matriz generatriz $G_{\mathcal{C}} \in \mathbb{F}_q^{m \times n}$. La matriz generatriz del código se hace pública y el distribuidor se encarga de asociar ciertos conjuntos de índices correspondientes a las columnas de $G_{\mathcal{C}}$ a los distintos participantes. Se puede asociar varias columnas a cada participante e, incluso, una misma columna a varios, pero con la condición de que las columnas adjudicadas al distribuidor sean linealmente dependientes del conjunto de columnas asociadas a los grupos autorizados e independientes del conjunto de columnas asociadas a los conjuntos no autorizados.
Sea $J \subseteq \{1, \dots n\}$ un conjunto de índices, denotamos por $G_{\mathcal{C}}(J)$ al conjunto de columnas de la matriz $G_{\mathcal{C}}$ indexadas por los elementos de J y por $\mathbf{c}(J)$ al conjunto de componentes de la palabra $\mathbf{c} \in \mathcal{C}$ indexados por el conjunto J. Denotamos por J_i el conjunto de índices de las columnas de $G_{\mathcal{C}}$ asociadas al participante i del esquema, y por J_D el conjunto de índices asociados al distribuidor del esquema. Para cada conjunto $A \subseteq 2^P$ denotamos

$$J_A = \bigcup_{p_i \in A} J_i.$$

La idea general del esquema es la siguiente:

1. Se considera un secreto $K \in \mathbb{F}_q^m$ con $m \leq k$ coordenadas y lo ampliamos hasta obtener un vector $(K, a) \in \mathbb{F}_q^k$ de longitud k.

2. Se codifica el vector mediante la aplicación:

$$\begin{array}{rccc} f: & \mathbb{F}_q^k & \longrightarrow & \mathbb{F}_q^n \\ & (K,a) & \longmapsto & (K,a)G_{\mathcal{C}} = \mathbf{c} = (\underbrace{c_1, \dots, c_m}_{J_D}, \underbrace{c_{m+1}, \dots, c_n}_{J_i}) \in \mathcal{C}. \end{array}$$

   Las primeras m componentes de la palabra $\mathbf{c}$ se asignan al distribuidor, y al resto de participantes p_i del esquema se le reparten las componentes de la palabra indexada por el correspondiente conjunto J_i.

3. Para recuperar el secreto, cada conjunto autorizado $\gamma \in \Gamma$ construye el código con matriz generatriz dada por las columnas de $G_{\mathcal{C}}$ correspondientes a los índices del conjunto $J_D \cup J_\gamma$, y descodifica la palabra construida con las componentes que tienen asignadas, considerando las componentes que corresponden al distribuidor (que son desconocidas) como un error que debe corregir el código.

Así la estructura de acceso minimal del esquema para compartir secretos basado en un código lineal está relacionada con el conjunto de palabras minimales de un código.

Bibliografía

[1] P. Abascal. *Compartir secretos mediante esquemas basados en códigos correctores*. PhD Thesis, Universidad de Oviedo, 1999.

[2] P. Abascal. *Algunos aspectos matemáticos del problema del reparto de secretos*. Bol. Soc. Esp. Mat. Apl., volume 21, 51-63, 2002.

[3] Vincenzo Acciaro. *On the complexity of computing Gröbner bases in characteristic* 2. Information Processing Letters, volume 51, Issue 6, 321-323, 1994.

[4] William W. Adams and Philippe Loustaunau. *An introduction to Gröbner bases*. Volume 3 of Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 1994.

[5] Elizabeth A. Arnold. *Modular algorithms for computing Gröbner bases*. J. Symbolic Comput. 35, volume 4, 403-419, 2003.

[6] A. Ashikhmin and A. Barg. *Minimal vectors in linear codes*. IEEE Trans. Inform.Theory, volume 44, 2010-2017, 1998.

[7] A. Barg. *Complexity issues in coding theory*. In Handbook of Coding Theory, Esevier Scienc., volume 1, 649-754, 1998.

[8] E. M. L. Beale *Survey of integer programming*. Operational Research Quarterly, volume 16, 219-228, 1965.

[9] Thomas Becker and Volker Weispfenning. *Gröbner bases*. Volume 141 of Graduate Texts in Mathematics. Springer-Verlag, New York, 1993. A computational approach to commutative algebra, In cooperation with Heinz Kredel.

[10] Elwyn R. Berlekamp, Robert J. McEliece and Henk C. A. van Tilborg. *On the inherent intractability of certain coding problems.* Institute of Electrical and Electronics Engineers. Transactions on Information Theory, volume 24, no.3, 384-386, 1978.

[11] G. R. Blakley. *Safeguarding cryptographic keys.* AFIPS Conference Proceedings, volume 48, 313-317, 1979.

[12] M. Borges-Quintana, M. A. Borges-Trenard, I. Márquez-Corbella and E. Martínez-Moro. *An Algebraic View to Gradient Descent Decoding.* Acepted to IEEE Information Theory Workshop, 2010.

[13] M. Borges-Quintana, M. A. Borges-Trenard, I. Márquez-Corbella and E. Martínez-Moro. *Descodificación por gradiente como reducción.* XII Encuentro de Álgebra Computacional y Aplicaciones, (Santiago de Compostela, 19-21 de julio de 2010).

[14] M. Borges-Quintana, M. A. Borges-Trenard, I. Márquez-Corbella and E. Martínez-Moro. *On the Border of Binary Code.* Submitted to Jour. Comp. Applied Maths., 2009.

[15] M. Borges-Quintana, M. A. Borges-Trenard, P. Fitzpatrick and E. Martínez-Moro. *Gröbner bases and combinatorics for binary codes.* Appl. Algebra Engrg. Comm. Comput., volume 19, no.5, 393-411, 2008.

[16] M. Borges-Quintana, M. A. Borges-Trenard and E. Martínez-Moro. *A Gröbner basis structure associated to linear codes.* J. Discrete Mathc. Sci. Cryptogr., volume 10, no.2, 151-191, 2007.

[17] M. Borges-Quintana, M. A. Borges-Trenard and E. Martínez-Moro. *A Gröbner representation for linear codes.* Advances in coding theory and cryptography. World Sci. Publ., Hackensack, NJ, volume 3, 17-32, 2007.

[18] M. Borges-Quintana, M. A. Borges-Trenard and E. Martínez-Moro. *A general framework for applying FGLM techniques to linear codes.* Lectures Notes in Comput. Sci., AAECC 16, volume 3857, 76-86, 2006.

[19] M. Borges-Quintana, M. A. Borges-Trenard and E. Martínez-Moro. *GBLA-LC: Gröbner bases by Linear Algebra and Linear Codes.* ICM 2006. Mathematical Software, EMS, 604-605, 2006.

[20] J. Bruck and M. Naor. *The Hardness of Decoding Linear Codes with Preprocessing*. IEEE Trans. Inform. Theory, volume 36, no.2, 1990.

[21] Xuemin Chen and I. S Reed and T. Helleseth and T. K Truong. *Use of Gröbner bases to decode binary cyclic codes up to the true minimum distance. Institute of Electical and Electronics Engineers, Transactions on Information Theory*, volume 40, no.5, 1654-1661, 1994.

[22] P. Conti and C. Traverso. *Buchberger algorithm and integer programming*. Applied algebra, algebraic algorithms and error-correcting codes (New Orleans, LA, 1991). Lectures Notes in Comput. Sci., volume 539, 130-139, 1991.

[23] P. Conti. *Basi di Gröbner e sistemi lineari diofantei*. Tech. Rep. Dip. Mat. Univ. Pisa, 1990.

[24] David A. Cox, John Little and Donal O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, second edition. An introduction to computational algebraic geometry and commutative algebra, 1997.

[25] David A. Cox, John Little and Donal O'Shea. *Using algebraic geometry*, volume 185 of Graduate Texts in Mathematics. Springer, New York, second edition, 2005.

[26] F. Di Biase and R. Urbanke. *An algorithm to calculate the kernel of certain polynomial ring homomorphisms*. Experiment. Math., volume 4, no.3, 227-234, 1995.

[27] J. C. Faugère, P. Gianini, D. Lazard and T. Mora. *Efficient computation of zero-dimensional Gröbner bases by change of ordering*. J.Symbolic Comput., volume 16, 329-344, 1993.

[28] P. Fitzpatrick. *Solving a multivariable congruence by change of therm order*. J. Symbolic Comput., volume 24(5), 575-138, 1997.

[29] P. Fitzpatrick and J. Flynn. *A Gröbner basis technique for Padé approximation*. J. Symbolic Comput., volume 13, 133-138, 1992.

[30] The GAP Group, GAP – Groups, Algorithms, and Programming. Version 4.12, 2009. `http://www.gap-system.org`.

[31] M. Giusti. *Some effectivity problems in polynomial ideal theory.* In Proc. Int. Symp. on Symbolic. and Algebraic Computation EUROSAM 84, Cambridge (England), volume 174 of LNCS, 159-171. Springer, 1994.

[32] R. E. Gomory. *An algorithm for integer solution to linear programming.* In R.L. Graves and P. Wolfe, editors Recent Advances in Mathematical Programming, 269-302, New York, McGraw-Hill, 1963.

[33] R. E. Gomory. *An algorithm for the mixed integer problem.* Technical Report, volume RM-2597, The RAND Cooperation, 1960.

[34] R. E. Gomory. *Outline of an algorithm for integer solutions to linear programs.* Bulletin of the American Society, volume 64, 275-278, 1958.

[35] Gert-Martin Greuel and Gerhard Pfister. *A Singular introduction to commutative algebra.* Springer-Verlag, Berlin, 2002. With contributions by Olaf Bachmann, Christoph Lossen and Hans Schnemann, With 1 CD-ROM (Windows, Macintosh, and UNIX).

[36] W. Cary Huffman and Vera Pless. *Fundamentals of Error Correcting Codes.* Cambridge University Press, 2003.

[37] D. Ikegami and Y. Kaji. *Maximum likelihood decoding for linear block codes using Gröbner bases.* IEICE Trans. Fund. Electron. Commun. Comput. Sci. E86-A, volume 3, 643-651, 2003.

[38] J. Justensen and T. Hoholdt. *A course in error-correcting codes.* European Mathematical Society (EMS), 2004.

[39] Y. N. Lakshman. *A single exponential bound on the complexity of computing Gröbner bases of zero-dimensional ideals.* Effective methods in algebraic geometry (Castiglioncello, 1990), 227-234, Progr. Math., 94, Birkhäuser Boston, Boston, MA, 1991.

[40] D. Lazard. *Elimination and Resolution of Systems of Algebraic Equations.* In Proc. EUROCAL 83, volume 162 of Lect. Notes in Comp. Sci, 146-157, 1983.

[41] A. H. Land and A. G. Doig. *An automatic method of solving discrete programming problems.* Econometrica, volume 28, 497-520, 1960.

[42] R. Liebler. *Implementing gradient descent decoding.* Michigan Math. J., volume 58, Issue 1, 285-291, 2009.

[43] S. Ling, C. Xing. Coding theory, a first course. Cambridge University Press, Cambridge, 2004.

[44] S. Ling. An Introduction to Error Correcting Codes. Prentice-Hall, 1970.

[45] J. Massey. *Minimal codewords and secret sharing.* Proc. Sixth Joint Swedish-Russian Workshop Inf. Theory, Mölle, Sweden, 246-249, 1993.

[46] I. Márquez-Corbella and E. Martínez-Moro. *Combinatorics of minimal codewords of some linear codes.* Submitted to Advances in Mathematics of Communications, 2010.

[47] I. Márquez-Corbella and E. Martínez-Moro. *Programación lineal modular y bases de Graver: Cálculo de soportes minimales de códigos lineales.* VII Jornadas de Matemática Discreta y Algorítmica, (Castro Urdiales, 7-9 de julio de 2010), 451-458, 2010.

[48] E. Martínez-Moro, C. Munuera-Gómez and D. Ruano Benito. Bases de Gröbner: Aplicaciones a la codificación algebraica. XX Escuela Venezolana de matemáticas, Caracas, Venezuela, 2007.

[49] Teo Mora. *Solving polynomial equation systems. II*, volume 99 of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 2005. Macaulay's paradigm and Gröbner technology.

[50] H. Niederreiter. *Coding Theory and Cryptology.* University Press Singapore. World Scientific Publishing Co., Inc., RiverEdge, NJ, USA, 2002.

[51] F. Ollivier. *Canonical Bases: Relations with Standard Bases, Finiteness Conditions and Applications to Tame Automorphisms.* Mega-90, proceedings. Progress in Mathemathics, Birkhauser, 379-400, 1991.

[52] W. W. Peterson and E. J. Jr. Weldon. *Error-Correcting Codes.* MIT Press, (1972).

[53] L. Pottier. *Minimal solution of linear diophantine systems: bounds and algorithms.* Proceedings RTA'91. LNCS, volume 488, Springer Verlag, 1991.

[54] The Sage Group, open-source math software. Version 4.4.2, 2010. `http://www.sagemath.org/.`

[55] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience, 1996.

[56] A. Shamir. *How to share a secret.* ACM Commun, volume 22, 612-619, 1979.

[57] D. Shannon and M. Sweedler. *Using Groebner bases to determine algebra membership, split surjective algebra homomorphisms and determine birational equivalence.* J. Symb., volume 6, 267-273, 1988.

[58] E. C. E. Shannon. *A Mathematical Theory of Communication.* Reprinted with correction from The Bell System Technical Journal, volume 27, 379-423, 623-656, July, October, 1948.

[59] D. Spear. *A constructive approach to commutative ring theory.* Proc. 1977 Macsyma User's conference, 369-376, 1977.

[60] Bernd Sturmfels. *Gröbner bases and convex polytopes.* Volume 8 of University Lecture Series. American Mathematical Society, Providence, RI, 1996.

[61] Rekha R. Thomas. *Applications to integer programming.* In Applications of computational algebraic geometry. San Diego, CA, 1997, volume 53 of Proc. Sympos. Appl. Math., 119-141. Amer. Math. Soc., Providence, RI, 1998.

[62] F. Winkler. *Polynomial algorithms in computer algebra.* Texts and Monographs in Symbolic Computation. Springer-Verlag, Vienna, 1996.