TEMat

Códigos de Reed-Muller: las matemáticas detrás de las primeras fotografías del planeta rojo

Andoni De Arriba De La Hera[a]
Instituto de Ciencias Matemáticas (ICMAT)
andoni.dearriba@icmat.es

Resumen: Este artículo tiene como objetivo presentar y estudiar los llamados códigos de Reed-Muller binarios. Estos son un tipo muy especial de códigos que, además, han jugado un papel fundamental en nuestra historia, puesto que fueron los responsables de que se obtuvieran las primeras fotografías en blanco y negro de la superficie marciana.

El artículo comienza con una breve introducción que tiene como objetivo situar este en contexto, así como fijar algunas de las notaciones básicas. Después, se hace un rápido repaso al mundo de los códigos desde un punto de vista matemático, estudiando todas las nociones básicas necesarias para la correcta comprensión del artículo. Con esto se pretende que cualquier lector mínimamente familiarizado con las matemáticas pueda disfrutar de la lectura. Para terminar, a modo de aplicación práctica, aparecen enlaces a programas diseñados en Mathematica que permiten interactuar con la familia de códigos estudiada.

Abstract: This paper aims to present and study the so-called binary Reed-Muller codes. These are a very special type of codes that have played a fundamental role in our history, since they were responsible for obtaining the first black and white photographies of the Martian surface.

The paper begins with a brief introduction that aims to place the work in context, as well as to fix some of the basic notations. Later on, we quickly review the world of codes from a mathematical point of view, studying all the basic notions that will be necessary for the correct understanding of the paper. With this, we hope that any reader minimally familiarized with mathematics will be able to enjoy reading the paper. To finish, as a practical application, we include links to algorithms designed in Mathematica that allow us to work with the studied code family.

Palabras clave: transmisión de información, códigos detectores y correctores de errores, códigos lineales, alfabeto, letras, palabras, codificar, decodificar.

MSC2010: 94B05.

Recibido: 10 de septiembre de 2018.
Aceptado: 24 de febrero de 2019.

Agradecimientos: Quiero agradecer a la ANEM la oportunidad que ofrece a los jóvenes investigadores con la creación de esta revista, y, en especial, a los editores por su dedicación a la misma y, más concretamente, por su insistencia en que escribiera este artículo. Quisiera también dar las gracias a los revisores encargados para este por el arduo trabajo llevado a cabo en la revisión. Finalmente, no puedo dejar sin mencionar a quien fue mi directora de TFG, M.ª Asunción García Sánchez, por toda la ayuda que me brindó para la correcta realización del mismo, ya que este artículo está basado en dicho trabajo.

Referencia: De Arriba De La Hera, Andoni. «Códigos de Reed-Muller: las matemáticas detrás de las primeras fotografías del planeta rojo». En: *TEMat*, 3 (2019), págs. 45-61. ISSN: 2530-9633. URL: https://temat.es/articulo/2019-p45.

[a] El autor estaba afiliado a la Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU) durante el desarrollo del trabajo del que parte este artículo.

1. Introducción

Este artículo tiene como objetivo estudiar los *códigos de Reed-Muller*. Esta es una de las familias más antiguas y mejor conocidas entre los *códigos lineales*. En concreto, son un tipo muy especial de códigos *detectores y correctores de errores*, con ricas propiedades algebraicas, que se utilizan habitualmente en la *transmisión de información*. Los estudios que se van a tratar en el artículo se situan en una de las aplicaciones más actuales del álgebra: la *teoría de la información*, cuyas bases fueron establecidas por Claude Elwood Shannon[1] [5], quien, a día de hoy, es considerado el padre de toda esta teoría. Hoy día, la teoría de la información es la rama de las matemáticas y la computación que se ocupa del estudio de la información y de todo lo relacionado con ella.

Supongamos que un emisor desea enviar un mensaje $\boldsymbol{x}$ a través de un canal para que lo reciba un receptor (proceso que se conoce por *transmisión de información*). A lo largo de este proceso, el mensaje $\boldsymbol{x}$ suele verse alterado debido al «ruido» del canal, de manera que el mensaje recibido por el receptor pasa a ser $\boldsymbol{x}'$, donde, en general, se tiene que $\boldsymbol{x}' \neq \boldsymbol{x}$. Aquí es donde entran los llamados códigos detectores y correctores de errores. La idea consiste en que, antes de enviar el mensaje $\boldsymbol{x}$, el emisor lo *codifica* como $\boldsymbol{c}$, añadiéndole información redundante. De esta manera, si en el canal se produce un *error* $\boldsymbol{e}$ debido al cual se recibe el mensaje alterado $\boldsymbol{c}' = \boldsymbol{c} + \boldsymbol{e}$, tras *decodificar* este, el receptor debería ser capaz de recuperar $\boldsymbol{c}$ y, de ahí, deducir $\boldsymbol{x}$. El objetivo que se busca con todo esto es el de lograr que este proceso tenga éxito de la manera lo más eficiente posible (tanto en tiempo como en memoria).

Salvo que se diga lo contrario, nuestros mensajes serán vectores (que llamaremos **palabras**) del $\mathbb{F}_q$-espacio vectorial finito $\mathbb{F}_q^n$, siendo $q = p^t$ con p primo ($\mathbb{F}_q$ es lo que llamaremos **alfabeto**, mientras que a sus elementos los denominaremos **letras**). Por simplicidad, se denota a las palabras de nuestro alfabeto por

$$\boldsymbol{x} = (x_1, x_2, \dots, x_n) \underset{\text{not.}}{\equiv} x_1 x_2 \dots x_n.$$

De esta manera, podemos definir matemáticamente un **código** como un subconjunto no vacío de palabras (a las cuales nos referiremos de manera natural por **palabras código**) de un mismo alfabeto.

La idea es hacer un estudio completo de los *códigos de Reed-Muller* en el *caso binario*. Concretamente, hablar de las tres construcciones conocidas para los mismos, así como del método de decodificación propio por el cual son tan interesantes. En primer lugar, se incluye un rápido repaso en teoría de códigos lineales, haciendo especial hincapié en aquello que hará falta para la correcta comprensión del artículo. Hecho esto, entramos en materia con el objeto de estudio y, una vez terminado, aparecen algunos de los programas diseñados en Mathematica para el Trabajo de Fin de Grado de De Arriba De La Hera [1]. Salvo que se diga lo contrario, todas las demostraciones de los resultados que aquí se utilizan pueden encontrarse en el mismo. Además, se da por hecho que el lector está familiarizado con los conceptos y resultados básicos del álgebra y la geometría, sobre todo en los casos finitos. También conviene tener un mínimo de conocimiento de combinatoria, dado que a lo largo del artículo aparecen en repetidas ocasiones resultados en los que es necesario hacer uso de coeficientes binomiales, así como algunas de las relaciones más importantes que se conocen entre los mismos.

2. Repaso a la teoría de códigos lineales

A primera vista, parece muy complicado trabajar con la definición de código dada desde un punto de vista matemático. Es por esta razón que resulta natural restringirse a familias de códigos más manejables y fáciles de implementar. En concreto, nos centramos en los llamados códigos lineales.

2.1. Nociones básicas

Definición 1. Dados $s \leq n$ números naturales, llamamos **código lineal** de longitud n y dimensión s sobre $\mathbb{F}_q$ a un $\mathbb{F}_q$-subespacio vectorial de $\mathbb{F}_q^n$ de dimensión s. Nos referimos a estos por códigos lineales q-arios de longitud n y dimensión s. ◂

[1] Matemático, ingeniero eléctrico y criptógrafo americano; 30 abril, 1916 - 24 febrero, 2001.

Sea $\mathcal{C}$ un código lineal q-ario de longitud n y dimensión s. La principal ventaja que tiene el uso de códigos lineales es que, por tratarse de subespacios vectoriales, admiten bases de la forma $\mathcal{B} = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_s\}$, de modo que toda palabra código de $\mathcal{C}$ puede expresarse de manera única como combinación lineal de estas palabras básicas. Así, escribiendo $\boldsymbol{c}_i = c_{i1} \dots c_{in}$ para todo $i \in \{1, \dots, s\}$, podemos construir la conocida como **matriz generadora** del código lineal, la cual se corresponde con

$$G = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{s1} & c_{s2} & \cdots & c_{sn} \end{pmatrix} \in \mathrm{Mat}_{s\times n}(\mathbb{F}_q).$$

Es obvio que para toda palabra código $\boldsymbol{c} \in \mathcal{C}$ que tomemos existen únicos escalares $\alpha_1, \dots, \alpha_s \in \mathbb{F}_q$ tales que $\boldsymbol{c} = (\alpha_1 \ \cdots \ \alpha_s)G$. Por tanto, para dar un código lineal, basta dar una matriz generadora del mismo.

Se define por **distancia de Hamming** entre dos palabras $\boldsymbol{x}$ e $\boldsymbol{y}$ de igual longitud al entero no negativo

$$d(\boldsymbol{x}, \boldsymbol{y}) \coloneqq |\{i \in \{1, \dots, n\} \mid x_i \neq y_i\}|. \tag{1}$$

Por otra parte, se define como **peso** de una palabra $\boldsymbol{x}$ al entero no negativo

$$\omega(\boldsymbol{x}) \coloneqq |\{i \in \{1, \dots, n\} \mid x_i \neq 0\}|. \tag{2}$$

Dado un código $\mathcal{C}$ arbitrario (no necesariamente lineal), a partir de (1) y (2) podemos introducir

$$d \underset{\text{not.}}{\equiv} d(\mathcal{C}) \coloneqq \min\{d(\boldsymbol{c}, \boldsymbol{c}') \mid \boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}, \boldsymbol{c} \neq \boldsymbol{c}'\}$$

y

$$\omega \underset{\text{not.}}{\equiv} \omega(\mathcal{C}) \coloneqq \min\{\omega(\boldsymbol{c}) \mid \boldsymbol{c} \in \mathcal{C}, \boldsymbol{c} \neq \boldsymbol{0}\},$$

donde d se conoce como **distancia mínima** del código, mientras que ω es el llamado **peso mínimo** del mismo. Si $\mathcal{C}$ es un código lineal, se comprueba fácilmente que $d = \omega$.

Un código lineal $\mathcal{C}$ *detecta hasta t errores* si, recibida $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ (siendo $\boldsymbol{c}$ la palabra código enviada y $\boldsymbol{e}$ el error dado en la transmisión, el cual se representa también como una palabra), con $0 < \omega(\boldsymbol{e}) \leq t$, entonces podemos asegurar que $\boldsymbol{y} \notin \mathcal{C}$. A su vez, un código lineal $\mathcal{C}$ *corrige hasta t errores* si, recibida $\boldsymbol{y}$, existe a lo más una palabra código $\boldsymbol{c} \in \mathcal{C}$ satisfaciendo que $d(\boldsymbol{y}, \boldsymbol{c}) \leq t$. No resulta difícil percatarse de que cualquier código $\mathcal{C}$ detecta hasta $d - 1$ errores, mientras que corrige hasta $\left\lfloor \frac{d-1}{2} \right\rfloor$.

Finalmente, se define como **código dual** de un código lineal $\mathcal{C} \subseteq \mathbb{F}_q^n$ de dimensión s al conjunto

$$\mathcal{C}^\perp \coloneqq \left\{\boldsymbol{x} \in \mathbb{F}_q^n \mid \langle \boldsymbol{x}, \boldsymbol{c} \rangle = 0 \ \forall \boldsymbol{c} \in \mathcal{C}\right\},$$

donde $\langle \cdot, \cdot \rangle$ denota el producto escalar estándar en $\mathbb{F}_q^n$. Este vuelve a ser un código lineal q-ario de longitud n, pero de dimensión $n - s$ en este caso. Por tanto, se cumple que $\dim(\mathcal{C}) + \dim(\mathcal{C}^\perp) = n$. Más aún, este hecho nos permite afirmar que $\mathcal{C}^\perp$ admite una matriz generadora $H \in \mathrm{Mat}_{(n-s)\times n}(\mathbb{F}_q)$. A esta se la conoce como **matriz de control** del código lineal $\mathcal{C}$ inicial. Entre las propiedades más importantes de H destaca que podemos definir $\mathcal{C}$ a partir de esta. En efecto, no resulta complicado comprobar que $\mathcal{C} = \{\boldsymbol{x} \in \mathbb{F}_q^n \mid \boldsymbol{x}H^\top = \boldsymbol{0}\}$. Otra propiedad a tener en cuenta es que $(\mathcal{C}^\perp)^\perp = \mathcal{C}$. Esto se debe a que toda matriz generadora G y de control H para el código lineal $\mathcal{C}$ están relacionadas mediante la igualdad $GH^\top = 0$ (o, equivalentemente, $HG^\top = 0$).

Ejemplo 2. El ejemplo típico son los **códigos de Hamming**. Pese a que estos pueden construirse sobre cualquier alfabeto, su construcción binaria es muy sencilla: dados s un natural arbitrario y $n = 2^s - 1$, el *código de Hamming binario de orden s* (longitud n y dimensión s) tiene por matriz de control aquella cuyas columnas son las n palabras no nulas de $\mathbb{F}_2^s$ escritas en forma ascendente (es decir, la representación binaria ordenada de los números del 1 al n). En todos los casos la distancia mínima es 3. ◂

Ejemplo 3. Otro ejemplo interesante son los **códigos de Hadamard**. Dado s un natural, este es el código lineal binario de longitud $n = 2^s$ y dimensión s cuya matriz generadora se construye por columnas como sigue: para cada $i \leq n$ natural, la i-ésima columna se corresponde con los bits de la representación binaria del número entero i. Este es, como ya veremos, un caso especial de código de Reed-Muller binario. ◂

2.2. Procesos de codificación y decodificación

Ya hemos comentado al comienzo que un *proceso de codificación* no es más que aquel a través del cual añadimos información redundante a nuestras palabras con el fin de que, al emplear el correspondiente *proceso de decodificación*, podamos recuperar estas si se produce algún error durante la transmisión de información. Matemáticamente, esto significa que transformamos cada palabra que queremos transmitir en palabras código. Esto no resulta una tarea sencilla en general (pues no parece existir un procedimiento estándar a través del cual se asocia a cada una de estas palabras una palabra código). Sin embargo, cuando tenemos códigos lineales entre manos, sí que se tiene un método bastante general gracias a que en estos casos se tiene una matriz generadora del código. En efecto, basta multiplicar esta con cada palabra a transmitir para obtener palabras código con las que trabajar en cada caso.

2.2.1. Codificación por matrices generadoras dadas en forma estándar

Ya hemos dicho que para dar un código lineal $\mathcal{C}$ es suficiente dar una matriz generadora. Nos preguntamos ahora: ¿se puede obtener una matriz generadora de expresión lo más sencilla posible? La respuesta a esta pregunta nos la da el resultado del álgebra lineal que nos dice que toda matriz $G \in \mathrm{Mat}_{s\times n}(\mathbb{F}_q)$ (con $s \leq n$) de rango máximo s puede llevarse, realizando operaciones elementales en filas y columnas, a una matriz de la forma $(I_s \mid B)$ con $B \in \mathrm{Mat}_{s\times(n-s)}(\mathbb{F}_q)$. A esta se la conoce por *forma estándar* de G. Sin embargo, en general, para que el código lineal que tenga a esta por matriz generadora coincida con $\mathcal{C}$, deben realizarse estas transformaciones elementales solo por filas. Luego no todo código lineal admite una matriz generadora de este tipo. La importancia de las matrices generadoras dadas en forma estándar radica en lo fácil que resulta codificar con ellas ya que, dada una palabra $\boldsymbol{x} \equiv x_1 \dots x_s \in \mathbb{F}_q^s$ arbitraria, esta se codifica como

$$\boldsymbol{x}\,(I_s \mid B) = (x_1 \dots x_s \ \underbrace{c_{s+1} \dots c_n}_{\text{redundancias}}) \in \mathcal{C},$$

donde es evidente que esta se corresponde con una palabra código en la que las s primeras letras son, precisamente, las de la palabra original $\boldsymbol{x}$. Si $G = (I_s \mid B) \in \mathrm{Mat}_{s\times n}(\mathbb{F}_q)$ es una matriz generadora para un cierto código lineal, entonces $H = (-B^\top \mid I_{n-s}) \in \mathrm{Mat}_{(n-s)\times n}(\mathbb{F}_q)$ es una matriz de control para el mismo.

2.2.2. Métodos generales de decodificación

Recordemos que el principal objetivo de la decodificación es recuperar las palabras enviadas durante la transmisión de información que pueden haberse visto alteradas a lo largo de este proceso. Para ello, el *método de decodificación* debe ser capaz de detectar y, si es posible, corregir los errores que puedan haberse dado. Matemáticamente, recibida una palabra donde se ha dado un error de peso no nulo, hemos de ser capaces de detectar que esta no es una palabra código y hallar «aquellas más próximas» para corregir este. Cuando solo existe una palabra código en dichas condiciones, la *decodificación* es *única*. Existen dos métodos de decodificación generales válidos para todo código lineal $\mathcal{C} \subseteq \mathbb{F}_q^n$ con distancia mínima d.

Primero se tiene el llamado *método de decodificación basado en líderes*. Este se basa en la relación de equivalencia sobre $\mathbb{F}_q^n$ siguiente:

$$\forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n,\ \boldsymbol{x}\mathfrak{R}\boldsymbol{y} \iff \boldsymbol{x} - \boldsymbol{y} \in \mathcal{C}.$$

Supongamos que se desea decodificar $\boldsymbol{z} \in \mathbb{F}_q^n$. Se buscan en $[\boldsymbol{z}]$ (clase representada por $\boldsymbol{z}$ en la relación de equivalencia anterior) las palabras de menor peso posible (los *líderes* de $[\boldsymbol{z}]$). Sea una de estas $\boldsymbol{e}_{\boldsymbol{z}}$. Entonces, se decodifica $\boldsymbol{z}$ por $\boldsymbol{z} - \boldsymbol{e}_{\boldsymbol{z}}$, que es una palabra código (se toma como error al líder elegido). Esta palabra $\boldsymbol{z}$ admite decodificación única en caso de que $\omega(\boldsymbol{e}_{\boldsymbol{z}}) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$. Este método es útil cuando resulta sencillo enumerar explícitamente las palabras del código.

Otro método de decodificación alternativo válido para cuando conocemos la matriz de control H para $\mathcal{C}$ es el llamado *método de decodificación basado en síndromes*. Este, definiendo $S(\boldsymbol{x}) \coloneqq \boldsymbol{x}H^\top$ como el *síndrome* de cada palabra $\boldsymbol{x} \in \mathbb{F}_q^n$ respecto de H, se basa en la relación de equivalencia sobre $\mathbb{F}_q^n$ siguiente:

$$\forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n,\ \boldsymbol{x} \sim \boldsymbol{y} \iff S(\boldsymbol{x}) = S(\boldsymbol{y}).$$

Supongamos que se desea decodificar $\boldsymbol{z} \in \mathbb{F}_q^n$. Si $S(\boldsymbol{z}) = \boldsymbol{0}$, se tiene que $\boldsymbol{z}$ es una palabra código y la decodificamos tal cual. En caso contrario, hay que buscar en $\bar{\boldsymbol{z}}$ (clase representada por $\boldsymbol{z}$ en la relación de equivalencia anterior) una palabra $\boldsymbol{e_z}$ de peso lo mínimo posible, para decodificar $\boldsymbol{z}$ como $\boldsymbol{z} - \boldsymbol{e_z}$, que es claramente una palabra código. Para obtener $\boldsymbol{e_z}$ en la práctica, construimos una tabla con los síndromes de las palabras del espacio vectorial total, ordenadas por pesos de menor a mayor. Habitualmente, se construye una tabla con los síndromes de las palabras con peso hasta $\left\lfloor \frac{d-1}{2} \right\rfloor$. A esta se la conoce por *tabla de síndromes*. Si el síndrome de nuestra palabra coincide con alguno de la tabla, podemos asegurar que la decodificación será única. Sin embargo, esto no es así cuando el peso de la palabra líder es mayor que $\left\lfloor \frac{d-1}{2} \right\rfloor$. Este método es útil cuando es fácil calcular una matriz de control del código.

Existen otros métodos de decodificación propios para ciertos tipos de códigos lineales más eficaces que estos dos, como puede ser el *método de decodificación cíclica*, válido para los llamados códigos cíclicos. Más adelante aparecerá otro método de decodificación, que tiene especial importancia cuando se trabaja con códigos de Reed-Muller binarios. Este es el llamado *método de decodificación por mayoría*.

Ejemplo 4. Consideremos el código de Hamming binario de orden 3. Una matriz de control para este código es

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

Para encontrar las palabras del código, basta con resolver las ecuaciones determinadas por H siguientes:

$$\begin{cases} & & & + x_4 & + x_5 & + x_6 & + x_7 = 0; \\ & + x_2 & + x_3 & & & + x_6 & + x_7 = 0; \\ + x_1 & & + x_3 & & + x_5 & & + x_7 = 0. \end{cases}$$

Podemos obtener una base tomando x_3, x_5, x_6 y x_7 como variables libres. Dando a tres de estas el valor 0 y 1 a la restante, obtenemos $\mathcal{B} = \{1110000, 1001100, 0101010, 1101001\}$. Denotando al código por $\mathcal{H}(3)$, tenemos que

$$\mathcal{H}(3) = \left\{ \begin{array}{cccccccc} 0000000 & 1110000 & 1001100 & 0101010 & 1101001 & 0111100 & 1011010 & 0011001 \\ 1100110 & 0100101 & 1000011 & 0010110 & 1010101 & 0110011 & 0001111 & 1111111 \end{array} \right\}.$$

Considerando la palabra 1010101, que pertenece al código, vamos a reemplazar el último 1 por un 0. Ahora, empleando los dos métodos que se acaban de explicar, vamos a recuperar la palabra original.

- Aplicando, por un lado, el método de decodificación basado en líderes, tenemos que calcular la clase de equivalencia $[1010100] \equiv \{\boldsymbol{x} \mid \boldsymbol{x} - 1010100 \in \mathcal{H}(3)\}$. Esta es

  $$\left\{ \begin{array}{cccccccc} 1010100 & 0100100 & 0011000 & 1111110 & 0111101 & 1101000 & 0001110 & 1001101 \\ 0110010 & 1110001 & 0010111 & 1000010 & \mathbf{0000001} & 1100111 & 1011011 & 0101011 \end{array} \right\}.$$

  Como hay una única palabra de peso 1, la decodificación es única, y es 1010101.

- Si, por el contrario, aplicamos el método de decodificación basado en síndromes, calculamos tanto $S(1010100) = (1010100)H^\top = (111)$ como los síndromes de las palabras con peso hasta 1, para ver cuáles coinciden con este. Omitiendo la palabra nula, tenemos la siguiente tabla de síndromes:

palabras	1000000	0100000	0010000	0001000	0000100	0000010	0000001
síndromes	001	010	011	100	101	110	**111**

  Como hay una única palabra de mismo síndrome, la decodificación es única, y es 1010101.

En resumen, la decodificación ha sido la que cabía esperar empleando ambos métodos. ◂

2.3. Algunas construcciones de códigos lineales

Se incluyen a continuación tres construcciones interesantes de códigos lineales. Resulta un buen ejercicio de repaso comprobar que, efectivamente, lo son, así como que se cumplen todas las propiedades que se van a enunciar para las mismas. Todas estas comprobaciones pueden encontrarse en el trabajo de De Arriba De La Hera [1, capítulo 1, Problemas Resueltos].

Ejemplo 5 (código suma). Sean $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$ dos códigos lineales con distancia mínima d_i, dimensión k_i y matriz generadora $G_i \in \mathrm{Mat}_{k_i \times n}(\mathbb{F}_q)$, respectivamente, siendo $i \in \{1, 2\}$. Se demuestra que la **suma** de $\mathcal{C}_1$ y $\mathcal{C}_2$, dada por

$$\mathcal{C}_1 + \mathcal{C}_2 \coloneqq \{\boldsymbol{c}_1 + \boldsymbol{c}_2 \mid \boldsymbol{c}_i \in \mathcal{C}_i \text{ con } i \in \{1, 2\}\},$$

es un código lineal q-ario con distancia mínima $d \leq \text{mín}\{d_1, d_2\}$. Se puede probar, además, que, si $\mathcal{C}_1 \cap \mathcal{C}_2 = \{\mathbf{0}\}$, entonces $\dim(\mathcal{C}_1 + \mathcal{C}_2) = k_1 + k_2$ y una matriz generadora de $\mathcal{C}_1 + \mathcal{C}_2$ viene dada por

$$G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}. \qquad \blacktriangleleft$$

Ejemplo 6 (código concatenación). Sean $\mathcal{C}_i \subseteq \mathbb{F}_q^{n_i}$ códigos lineales de longitud n_i, distancia mínima d_i, dimensión k_i, matriz generadora $G_i \in \mathrm{Mat}_{k_i \times n_i}(\mathbb{F}_q)$ y matriz de control $H_i \in \mathrm{Mat}_{(n-k_i)\times n}(\mathbb{F}_q)$, respectivamente, siendo $i \in \{1, 2\}$. Se demuestra que la **concatenación** de $\mathcal{C}_1$ con $\mathcal{C}_2$, dada por

$$\mathcal{C}_1 * \mathcal{C}_2 \coloneqq \{\boldsymbol{c}_1 * \boldsymbol{c}_2 = c_{11} \dots c_{1n_1} c_{21} \dots c_{2n_2} \mid \boldsymbol{c}_i \in C_i \text{ con } i \in \{1, 2\}\},$$

es un código lineal q-ario de longitud $n_1 + n_2$, dimensión $k_1 + k_2$, distancia mínima $d = \text{mín}\{d_1, d_2\}$ y matrices generadora y de control

$$G = \begin{pmatrix} G_1 & 0 \\ 0 & G_2 \end{pmatrix} \qquad \text{y} \qquad H = \begin{pmatrix} H_1 & 0 \\ 0 & H_2 \end{pmatrix}. \qquad \blacktriangleleft$$

Ejemplo 7 (construcción de Plotkin). Sean $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$ dos códigos lineales de distancia mínima d_i, dimensión k_i, matriz generadora $G_i \in \mathrm{Mat}_{k_i \times n}(\mathbb{F}_q)$ y matriz de control $H_i \in \mathrm{Mat}_{(n-k_i)\times n}(\mathbb{F}_q)$, respectivamente, siendo $i \in \{1, 2\}$. Se demuestra que el conjunto definido por

$$\mathcal{C}_1 \circledast \mathcal{C}_2 \coloneqq \{(\boldsymbol{c}_1 \mid \boldsymbol{c}_1 + \boldsymbol{c}_2) \mid \boldsymbol{c}_i \in \mathcal{C}_i \text{ con } i \in \{1, 2\}\},$$

donde $(\boldsymbol{c}_1 \mid \boldsymbol{c}_1 + \boldsymbol{c}_2) \coloneqq \boldsymbol{c}_1 * (\boldsymbol{c}_1 + \boldsymbol{c}_2) = \boldsymbol{c}_1 * \boldsymbol{c}_1 + \mathbf{0} * \boldsymbol{c}_2$ para todo $\boldsymbol{c}_1 \in \mathcal{C}_1, \boldsymbol{c}_2 \in \mathcal{C}_2$, es un código lineal q-ario de longitud $2n$, dimensión $k_1 + k_2$, distancia mínima $d = \text{mín}\{2d_1, d_2\}$ y matrices generadora y de control

$$G = \begin{pmatrix} G_1 & G_1 \\ 0 & G_2 \end{pmatrix} \qquad \text{y} \qquad H = \begin{pmatrix} H_1 & 0 \\ -H_2 & H_2 \end{pmatrix}. \qquad \blacktriangleleft$$

Nota 8. El código lineal presentado en el ejemplo 7 jugará un papel muy importante cuando tratemos la segunda de las construcciones para los códigos de Reed-Muller binarios. ◀

3. Códigos de Reed-Muller binarios

3.1. Aspectos históricos

Los códigos de Reed-Muller son una familia infinita de códigos lineales, que toman su nombre de los dos matemáticos que los propusieron en el año 1954, prácticamente al mismo tiempo, en dos trabajos independientes: Irving Stoy Reed[2] y David Eugene Muller[3]. Ambos se ocuparon de introducir estos en el caso binario. Hoy se sabe que el primero en realizar la primera de las construcciones para estos códigos en su forma binaria fue Muller [3], mientras que su estudio en detalle y la sencilla decodificación por la que son tan conocidos e importantes en este caso binario es obra de Reed [4].

Los códigos de Reed-Muller tienen una gran importancia en la historia. Su estudio en la década de los años 50 fue fundamental para que en los años posteriores se hiciesen grandes avances en la exploración espacial. Así, desde 1969 hasta 1977, todas las naves espaciales de la NASA iban equipadas con un código de Reed-Muller binario de longitud 32, dimensión 6 y distancia mínima 16. Se escogió dicho código dado que el cociente entre la dimensión del mismo y su longitud es (relativamente) pequeño para la amplia distancia mínima que posee. Por esta razón podemos decir que nos encontramos ante un código de *bajo coste* y buenas capacidades para *corregir errores*.

[2] Matemático e ingeniero estadounidense; 12 noviembre, 1923 - 11 septiembre, 2012.
[3] Matemático e informático teórico estadounidense; 2 noviembre, 1924 - 27 abril, 2008.

Una de las misiones más destacadas que se llevó a cabo con el uso de estos códigos, y que pasó a la historia por su gran impacto, fue la realizada en los años 70 por la sonda Mariner 9, ya que esta fue la primera que permitió la observación fotográfica de la superficie marciana. Esta sonda fue lanzada el 30 de mayo del 1971, llegando a su destino el 13 de noviembre del mismo año, convirtiéndose así en la primera nave espacial en orbitar un planeta distinto al nuestro. Científicamente, esta misión, que constituyó una continuación de las observaciones adquiridas por las sondas Mariner 6 y 7, tenía como objetivo mostrar las primeras fotografías de Marte. En un principio la misión se complicó debido a las grandes tormentas de arena que se dieron sobre todo el conjunto de la superficie del planeta. Sin embargo, en 1972, cuando por fin amainaron dichas tormentas, se obtuvieron estas primeras fotografías en blanco y negro, las cuales cambiaron completamente la visión que se tenía hasta entonces del planeta rojo (figura 1). La sonda tomó fotografías en blanco y negro de $600 \times 600 = 360\,000$ píxeles, donde a cada píxel se le asignó una 6-tupla para representar el brillo. Cada píxel era codificado como una palabra de longitud 32 (se emplearon 26 bits de redundancia). Fue necesario usar un código con palabras de gran longitud, pues los errores de transmisión debían minimizarse al máximo dada la enorme distancia que los mensajes recorrían desde Marte a la Tierra. Era, además, imprescindible, dado el tiempo requerido por cada transmisión, que la decodificación fuese posible en la mayor parte de los casos y con garantías de unicidad.

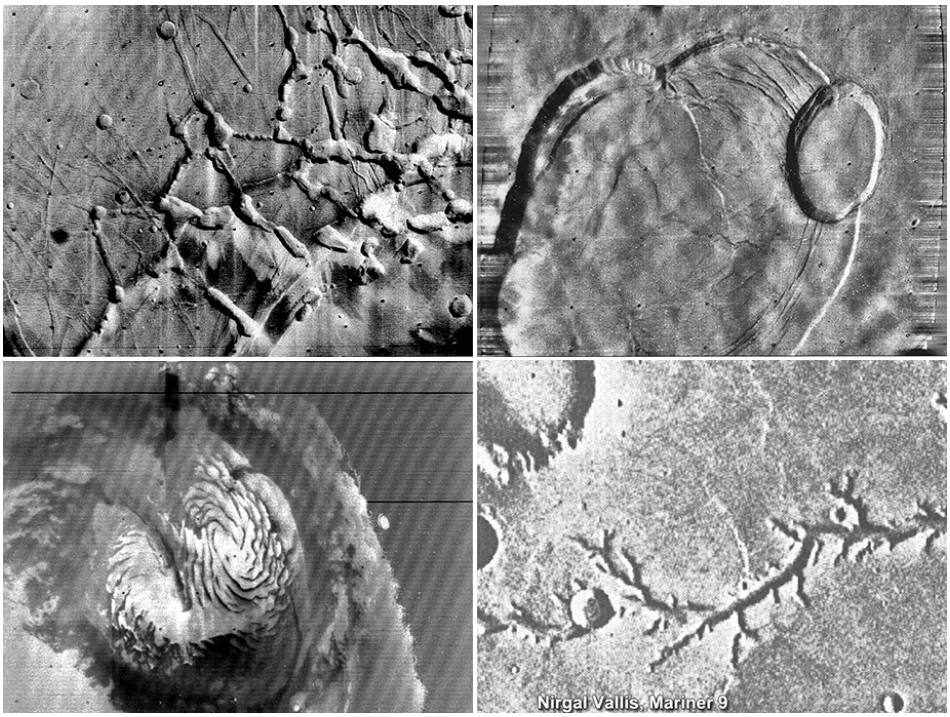

Figura 1: Imágenes, facilitadas por la NASA bajo dominio público, captadas por la sonda Mariner 9.

3.2. Construcciones y propiedades principales

Como ya se ha adelantado, existen tres formas de introducir los códigos de Reed-Muller binarios. Cada una de ellas resulta importante por motivos distintos, como ya iremos comprobando a lo largo del artículo.

3.2.1. Construcción original de Muller

Esta es la primera construcción conocida, debida a D. E. Muller. Además de la referencia principal [1], ha sido necesario también consultar las notas de Iranzo Aznar y Pérez Monasor [2, Lección 7] para la redacción de esta parte por motivos que se entenderán más adelante (véase la nota 19). Para introducir esta construcción en su versión original, es necesaria la conocida como *teoría de Boole*, la cual se pretende estudiar brevemente a continuación.

Definición 9. Sea m un número natural. Una aplicación $f\colon \mathbb{F}_2^m \to \mathbb{F}_2$ es una **función booleana** de m variables. Se denota al conjunto de todas las funciones booleanas de m variables por $\mathfrak{B}_m$. ◂

Observación 10. Podemos dotar a $\mathfrak{B}_m$ con una estructura de anillo conmutativo y unitario. Más aún, se tiene que este conjunto también posee estructura de $\mathbb{F}_2$-espacio vectorial. En definitiva, nos encontramos ante una $\mathbb{F}_2$-álgebra conmutativa y unitaria, la cual se conoce usualmente por **álgebra de Boole**. ◂

A partir de este momento, y salvo que se diga lo contrario, vamos a trabajar con el álgebra $\mathbb{F}_2^m$ dado m un entero no negativo, y tendremos siempre que $n = 2^m$.

Observación 11. Si se fija un orden en los n elementos de $\mathbb{F}_2^m$, es posible describir toda función booleana $f\colon \mathbb{F}_2^m \to \mathbb{F}_2$ de manera unívoca a través de una tabla con todos los elementos de $\mathbb{F}_2^m$ y los respectivos valores que toma f en cada uno de estos. A esta se la conoce por *tabla de verdad* asociada a f. Además, fijado un orden $\mathbb{F}_2^m = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$ y dada $f \in \mathfrak{B}_m$ arbitraria, inducimos la siguiente notación:

$$f_i \underset{\text{not.}}{\equiv} f(\boldsymbol{v}_i), \quad \forall\, i \in \{1, \dots, n\}.$$

Llamaremos a este valor *coordenada i-ésima* de f bajo el orden establecido en $\mathbb{F}_2^m$. En estas condiciones, toda aplicación $f \in \mathfrak{B}_m$ puede identificarse de manera unívoca empleando tablas de verdad con la correspondiente palabra $\boldsymbol{f} \equiv f_1 \dots f_n \in \mathbb{F}_2^n$. Así, lo natural es trabajar con el conjunto $\mathbb{F}_2^m = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$ ordenado cuando se tengan funciones booleanas. A $\boldsymbol{f} \in \mathbb{F}_2^n$ se la conoce por *palabra característica* de f bajo el orden establecido. En particular, se deduce de este hecho que $|\mathfrak{B}_m| = 2^n < \infty$. ◂

Definición 12. Definimos el *anillo de los polinomios booleanos* de m indeterminadas como el cociente

$$\mathcal{P}_m \underset{\text{not.}}{\equiv} \frac{\mathbb{F}_2[x_1, \dots, x_m]}{(x_1^2 - x_1, \dots, x_m^2 - x_m)}.$$

Cada clase de equivalencia $\overline{F} = F + (x_1^2 - x_1, \dots, x_m^2 - x_m) \in \mathcal{P}_m$ posee un único representante

$$F^* \underset{\text{not.}}{\equiv} \sum a_{i_1 \dots i_m} x_1^{i_1} \cdots x_m^{i_m} \in \mathbb{F}_2[x_1, \dots, x_m] \quad \left(a_{i_1 \dots i_m} \in \mathbb{F}_2\right),$$

verificando que $i_1, \dots, i_m \in \{0, 1\}$. Esta se conoce por *forma reducida* del polinomio F. Además, es evidente que esta está unívocamente determinada aplicando a cada monomio de F las reglas

$$\underbrace{x_i x_j = x_j x_i}_{\text{conmutatividad en } \mathbb{F}_2[x_1, \dots, x_m]} \quad \text{y} \quad \underbrace{x_i^2 = x_i,}_{\text{pequeño teorema de Fermat en } \mathbb{F}_2}$$

para cualesquiera $i, j \in \{1, \dots, m\}$ diferentes, hasta que los factores del polinomio resultante sean todos distintos. Resulta natural definir en estas condiciones el *grado* de un polinomio booleano como el grado, entendido en el sentido usual, del correspondiente polinomio en su forma reducida. ◂

Observación 13. No es complicado percatarse de que $\mathcal{P}_m$ tiene también estructura de espacio vectorial sobre $\mathbb{F}_2$. De hecho, el conjunto $\mathcal{B} = \{x_1^{r_1} \cdots x_m^{r_m} \mid r_i \in \{0, 1\}\ \forall\, i \in \{1, \dots, m\}\}$ es una base de este (en particular, $\mathcal{P}_m$ es finito como espacio vectorial). Así, dado que el número de monomios booleanos de m indeterminadas y grado k es $\binom{m}{k}$ trivialmente, no resulta complicado comprobar que $|\mathcal{P}_m| = 2^m$. ◂

De ahora en adelante, y salvo que se diga lo contrario, nos restringimos al uso de polinomios en forma reducida cuando tratemos con los elementos de $\mathcal{P}_m$ y vamos a suponer que tenemos fijado un orden $\mathbb{F}_2^m = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$. En particular, utilizar palabras en $\mathbb{F}_2^n$ y funciones booleanas de m variables resulta equivalente. Razonando por inducción sobre m, el número de indeterminadas, obtenemos el siguiente resultado, necesario para probar el que sigue y que resulta clave para llevar a cabo esta construcción.

Lema 14. *Sea $F \in \mathbb{F}_2[x_1, \dots, x_m]$ un polinomio arbitrario no necesariamente dado en forma reducida. Si se cumple que $F(u_1, \dots, u_m) = 0$ para todo $u_1, \dots, u_m \in \mathbb{F}_2$, entonces $F^* = 0$.*

Teorema 15. *Los conjuntos $\mathfrak{B}_m$ y $\mathcal{P}_m$ son isomorfos como $\mathbb{F}_2$-álgebras conmutativas y unitarias, a través de la aplicación que asocia a cada polinomio booleano F la única función booleana f tal que*

$$F(u_1, u_2, \dots, u_m) = f(\boldsymbol{u}), \quad \forall\, \boldsymbol{u} = (u_1, u_2, \dots, u_m) \in \mathbb{F}_2^m.$$

En consecuencia, cada polinomio en forma reducida tiene asociada una, y solo una, palabra de $\mathbb{F}_2^n$.

Demostración. Veamos en primer lugar que la aplicación

$$\begin{array}{rccc} \psi_m: & \mathbb{F}_2[x_1,\dots,x_m] & \longrightarrow & \mathbb{F}_2^n \\ & F & \longmapsto & \psi_m(F) := \boldsymbol{f} \equiv f_1 \dots f_n \end{array} \tag{3}$$

es un epimorfismo de álgebras sobre $\mathbb{F}_2$. El único paso no trivial es comprobar la sobreyectividad. Para ello, dado $\boldsymbol{x} \in \mathbb{F}_2^n$ arbitrario, suponiendo que $\boldsymbol{v}_i = (\alpha_1^i, \dots, \alpha_m^i) \in \mathbb{F}_2^m$ para todo $i \in \{1, \dots, n\}$, definimos

$$F_{\boldsymbol{v}_i}(x_1, \dots, x_m) := \prod_{j=1}^{m} \left(1 - \left(x_j - \alpha_j^i\right)\right) \in \mathbb{F}_2[x_1, \dots, x_m], \quad \forall\, i \in \{1, \dots, n\}.$$

Por el pequeño teorema de Fermat, se tiene que, para todo $\boldsymbol{w} = (w_1, \dots, w_m) \in \mathbb{F}_2^m$ y todo $i \in \{1, \dots, n\}$,

$$F_{\boldsymbol{v}_i}(w_1, \dots, w_m) = \begin{cases} 1, & \text{si } \boldsymbol{w} = \boldsymbol{v}_i; \\ 0, & \text{si } \boldsymbol{w} \neq \boldsymbol{v}_i. \end{cases}$$

De esta manera, resulta inmediato comprobar, por cómo se define la aplicación (3), si escribimos las imágenes explícitamente, que el conjunto $\mathcal{B} = \{\psi_m(F_{\boldsymbol{v}_i}) \mid i \in \{1, \dots, n\}\}$ se corresponde con la base canónica de $\mathbb{F}_2^n$. En consecuencia, existen escalares $k_i \in \mathbb{F}_2$ para cada $i \in \{1, \dots, n\}$ tales que

$$\boldsymbol{x} = \sum_{i=1}^{n} k_i \psi_m(F_{\boldsymbol{v}_i}) = \psi_m\left(\sum_{i=1}^{n} k_i F_{\boldsymbol{v}_i}\right).$$

Así, tomando $F = \sum_{i=1}^{n} k_i F_{\boldsymbol{v}_i}$, es evidente que $\psi_m(F) = \boldsymbol{x}$. Queda así probada la sobreyectividad de (3). Hecho esto, aplicando el primer teorema de isomorfía para álgebras, se sigue que

$$\frac{\mathbb{F}_2[x_1, \dots, x_m]}{\operatorname{Ker}(\psi_m)} \cong \operatorname{Im}(\psi_m) \equiv \mathbb{F}_2^n.$$

Basta probar, por tanto, que $\operatorname{Ker}(\psi_m) = (x_1^2 - x_1, \dots, x_m^2 - x_m)$ y habremos terminado. El contenido no trivial es $\operatorname{Ker}(\psi_m) \subseteq (x_1^2 - x_1, \dots, x_m^2 - x_m)$ y se argumenta por reducción al absurdo: dado $F \in \operatorname{Ker}(\psi_m)$ arbitrario, vamos a suponer que $F \notin (x_1^2 - x_1, \dots, x_m^2 - x_m)$ para llegar a una contradicción. En estas condiciones, suponiendo que R es la forma reducida de F, se tiene que la clase de equivalencia para F puede escribirse como $\overline{F} = R + (x_1^2 - x_1, \dots, x_m^2 - x_m)$, donde $R \neq 0$ necesariamente. Pero, como $\psi_m(F) = \mathbf{0}$ por hipótesis, estamos ante las condiciones del lema 14, luego $F^* = R = 0$, en contra de la suposición hecha. ■

Sea r un número natural tal que $r \leq m$. Denotaremos por $\mathcal{P}_m(r)$ al conjunto de los polinomios booleanos de m indeterminadas en $\mathcal{P}_m$ con grado menor o igual que r. Este es trivialmente un $\mathbb{F}_2$-subespacio vectorial de $\mathcal{P}_m$ de dimensión finita. Empleando entonces el teorema 15 que acabamos de probar, estamos en condiciones de dar la siguiente definición importante.

Definición 16. Se define por **código de Reed-Muller** binario $\mathcal{RM}(r, m)$ de orden r y longitud n a la imagen directa de $\mathcal{P}_m(r)$ a través de la aplicación dada en (3). Dicho de otra manera, este es el conjunto de todas las palabras binarias de longitud n asociadas a todos los polinomios booleanos de m indeterminadas mediante el isomorfismo dado en el teorema 15 con grado menor o igual que r. ◂

Nota 17. Por convenio, escribiremos que $\mathcal{RM}(\ell, m) = \{0 \dots 0\}$ para todo entero $\ell < 0$. Además, por la definición, es inmediato que $\mathcal{RM}(m, m) = \mathbb{F}_2^n$ y $\mathcal{RM}(0, m) = \{0 \dots 0, 1 \dots 1\}$. ◂

Observación 18. Esta definición es independiente del orden fijado sobre $\mathbb{F}_2^m$. En otras palabras, dado un código de Reed-Muller binario construido bajo un cierto orden en $\mathbb{F}_2^m$ fijado, al cambiar dicho orden, obtenemos otro código de Reed-Muller binario, con los mismos parámetros que el inicial (estos solo se diferencian en que se han permutado entre sí las letras de un número finito de posiciones fijadas en todas las palabras del código). Cuando dos códigos están relacionados tal y como se acaba de explicar, se dice que son *códigos equivalentes por permutación*. ◂

Nota 19. Esta construcción fue generalizada a cualquier cuerpo finito $\mathbb{F}_q$ en 1968. El lector interesado en ello puede consultar el trabajo de De Arriba De La Hera [1, capítulo 2] y las referencias ahí recogidas para comprender esta generalización. ◂

Proposición 20. *El código de Reed-Muller binario $\mathcal{RM}(r,m)$ es un código lineal de longitud n sobre $\mathbb{F}_2$. Manteniendo las notaciones introducidas en el teorema 15, una matriz generadora para este es*

$$G(r,m) \underset{not.}{\equiv} \begin{pmatrix} \psi_m(1) \\ \psi_m(x_1) \\ \vdots \\ \psi_m(x_m) \\ \psi_m(x_1x_2) \\ \vdots \\ \psi_m(x_{m-r+1}\dots x_m) \end{pmatrix}. \tag{4}$$

En consecuencia, la dimensión como $\mathbb{F}_2$-subespacio vectorial de $\mathbb{F}_2^n$ viene dada por la fórmula

$$\dim(\mathcal{RM}(r,m)) = \sum_{k=0}^{r} \binom{m}{k}. \tag{5}$$

Además, no resulta complicado comprobar que $\mathcal{RM}(r,m)^\perp = \mathcal{RM}(m-r-1,m)$.

Demostración. Basta probar que todas las palabras de $\mathbb{F}_2^n$ asociadas a los monomios reducidos de $\mathcal{P}_m(r)$ constituyen una base de $\mathcal{RM}(r,m)$. Nos es suficiente ver que estas generan un sistema linealmente independiente argumentando por reducción al absurdo y haciendo uso del lema 14. Una vez se tiene esto, con un simple argumento combinatorio, obtenemos la fórmula dada para la dimensión. Finalmente, la igualdad entre códigos lineales se deduce de la inclusión $\mathcal{RM}(m-r-1,m) \subseteq \mathcal{RM}(r,m)^\perp$ dado que la suma de las dimensiones de $\mathcal{RM}(m-r-1,m)$ y $\mathcal{RM}(r,m)$ es 2^m. ■

Ejemplo 21. Vamos a construir empleando la definición dada el código $\mathcal{RM}(1,3)$. Para ello, vamos a fijar el orden $\mathbb{F}_2^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Hecho esto, tenemos que obtener $\mathcal{P}_3(1)$, esto es, los polinomios booleanos de 3 indeterminadas que tengan grado menor o igual que 1. Estos son los siguientes:

$$\begin{array}{cccccccc} 0, & x_1, & x_2, & x_3, & x_1+x_2, & x_1+x_3, & x_2+x_3, & x_1+x_2+x_3, \\ 1, & 1+x_1, & 1+x_2, & 1+x_3, & 1+x_1+x_2, & 1+x_1+x_3, & 1+x_2+x_3, & 1+x_1+x_2+x_3. \end{array}$$

Así, ya podemos enumerar las palabras del código de Reed-Muller deseado, y tenemos que estas son

$$\left\{ \begin{array}{cccccccc} 00000000 & 00001111 & 00110011 & 01010101 & 00111100 & 01011010 & 01100110 & 01101001 \\ 11111111 & 11110000 & 11001100 & 10101010 & 11000011 & 10100101 & 10011001 & 10010110 \end{array} \right\}.$$

A partir de esto, podemos obtener una matriz generadora para el código inmediatamente, y tenemos por la fórmula (5) que la dimensión de este es 4. ◂

3.2.2. Construcción recursiva de Plotkin

Nos basamos en la construcción de Plotkin recogida en el ejemplo 7 para obtener una construcción recursiva de los códigos de Reed-Muller binarios, solo válida para un cierto orden en los elementos de $\mathbb{F}_2^m$.

Definición 22. Dada la expansión binaria $i_0 + 2i_1 + 2^2 i_2 + \cdots + 2^{m-2} i_{m-2} + 2^{m-1} i_{m-1}$ de un cierto entero i, con $i_0, i_1, \dots, i_{m-2}, i_{m-1} \in \{0,1\}$, asociamos a $i+1$ el elemento $(i_{m-1}, i_{m-2}, \dots, i_2, i_1, i_0) \in \mathbb{F}_2^m$. Esto es,

$$\begin{array}{ccc} 1 & \longrightarrow & (0,0,\dots,0,0,0); \\ 2 & \longrightarrow & (0,0,\dots,0,0,1); \\ 3 & \longrightarrow & (0,0,\dots,0,1,0); \\ 4 & \longrightarrow & (0,0,\dots,0,1,1); \\ 5 & \longrightarrow & (0,0,\dots,1,0,0); \\ \vdots & \vdots & \vdots \\ 2^m-2 & \longrightarrow & (1,1,\dots,1,0,1); \\ 2^m-1 & \longrightarrow & (1,1,\dots,1,1,0); \\ 2^m & \longrightarrow & (1,1,\dots,1,1,1). \end{array}$$

Al orden inducido por esta asociación sobre $\mathbb{F}_2^m$ lo llamaremos *orden canónico* de $\mathbb{F}_2^m$. ◂

Supondremos a lo largo de todo este apartado que tenemos fijado el orden canónico para los elementos de $\mathbb{F}_2^m$. Además, mantendremos la notación ψ_m introducida en el teorema 15 para la aplicación (3). Nuestro primer objetivo pasa por estudiar las propiedades fundamentales que se tienen bajo este orden, las cuales se recogen en los resultados siguientes, y nos permiten escribir $\mathcal{RM}(r,m)$ como construcción de Plotkin entre $\mathcal{RM}(r,m-1)$ y $\mathcal{RM}(r-1,m-1)$. Hecho esto, podremos obtener expresiones dependientes de r y m tanto para la distancia mínima como para una matriz generadora de todos los códigos de Reed-Muller binarios (esto último solo para cuando se tiene el orden canónico) de manera recursiva.

Lema 23. *Se tiene la siguiente tabla de verdad:*

$$\begin{array}{ccc}
\psi_m(x_m) & = & 0101010101010101\ldots010101; \\
\psi_m(x_{m-1}) & = & 001100110011001\ldots0110011; \\
\psi_m(x_{m-2}) & = & 00001111000011\ldots00001111; \\
\vdots & \vdots & \vdots \\
\psi_m(x_2) & = & 00\ldots011\ldots100\ldots011\ldots11; \\
\psi_m(x_1) & = & \underbrace{00000\ldots0000}_{2^{m-1}}\underbrace{11111\ldots1111}_{2^{m-1}}.
\end{array}$$

Lema 24. *Sea F un polinomio booleano en cuya expresión no aparece la indeterminada x_1 y que, por tanto, podemos ver como polinomio de $m-1$ indeterminadas. Si la palabra binaria asociada en este caso es $\boldsymbol{f} = f_1\ldots f_{2^{m-1}}$, entonces se tiene que la palabra binaria asociada a F como polinomio booleano de $\mathcal{P}_m$ es la concatenación $\boldsymbol{f} * \boldsymbol{f} = f_1\ldots f_{2^{m-1}}f_1\ldots f_{2^{m-1}}$.*

Estos dos resultados son fundamentales para probar el que sigue a continuación, que es característico de esta construcción. Las demostraciones para ambos son inmediatas: para el primero, basta construir las tablas de verdad para el orden canónico de $\mathbb{F}_2^m$ asociadas a los monomios booleanos x_i como elementos de $\mathcal{P}_m$, mientras que el segundo se sigue de construir las tablas de verdad para el orden canónico de $\mathbb{F}_2^m$ asociadas al polinomio dado en el enunciado, visto como elemento tanto de $\mathcal{P}_m$ como de $\mathcal{P}_{m-1}$, y, hecho esto, comparar estas dos palabras para ver que la segunda concatenada consigo misma es la primera.

Teorema 25. *Dado un número natural r tal que $0 < r < m$, se cumple que*

$$\mathcal{RM}(r,m) = \mathcal{RM}(r,m-1) \circledast \mathcal{RM}(r-1,m-1). \tag{6}$$

Demostración. Veamos primero que $\mathcal{RM}(r,m) \subseteq \mathcal{RM}(r,m-1) \circledast \mathcal{RM}(r-1,m-1)$. Sea $\boldsymbol{x} \in \mathcal{RM}(r,m)$ una palabra código, asociada al polinomio booleano F de m indeterminadas y grado menor o igual que r. Podemos expresar F en función de dos polinomios booleanos de $m-1$ indeterminadas G y H por

$$F(x_1,\ldots,x_m) = G(x_2,\ldots,x_m) + x_1H(x_2,\ldots,x_m), \tag{7}$$

donde G tiene grado menor o igual que r y H, grado menor o igual que $r-1$. Supongamos que $\boldsymbol{x}_G$ y $\boldsymbol{x}_H$ son las palabras binarias asociadas a estos polinomios, respectivamente, los cuales estamos viendo como si fuesen polinomios booleanos de $m-1$ indeterminadas. Es evidente, por definición, que $\boldsymbol{x}_G \in \mathcal{RM}(r,m-1)$ y $\boldsymbol{x}_H \in \mathcal{RM}(r-1,m-1)$. Ahora, por el lema 24, se tiene que $\boldsymbol{x}_G * \boldsymbol{x}_G$ y $\boldsymbol{x}_H * \boldsymbol{x}_H$ son las palabras binarias asociadas a nuestros polinomios booleanos, respectivamente, vistos como si tuvieran m indeterminadas. Aplicando entonces ψ_m a (7), por el lema 23 se tiene, por ser este un homomorfismo de álgebras, que

$$\boldsymbol{x} = \psi_m(F) = \psi_m(G) + \psi_m(x_1)\psi_m(H) = \boldsymbol{x}_G * \boldsymbol{x}_G + \boldsymbol{0} * \boldsymbol{x}_H \in \mathcal{RM}(r,m-1) \circledast \mathcal{RM}(r-1,m-1).$$

Para obtener la igualdad basta ver que ambos códigos tienen la misma dimensión. En efecto, por (5) y la fórmula de Pascal, por como viene dada la fórmula correspondiente a la dimensión en la construcción de Plotkin recogida en el ejemplo 7,

$$\begin{aligned}
\dim(\mathcal{RM}(r,m)) &= \sum_{k=0}^{r}\binom{m}{k} = \binom{m}{0} + \sum_{k=1}^{r}\binom{m}{k} = \binom{m-1}{0} + \sum_{k=1}^{r}\binom{m-1}{k} + \sum_{k=1}^{r}\binom{m-1}{k-1} \\
&= \sum_{k=0}^{r}\binom{m-1}{k} + \sum_{k=0}^{r-1}\binom{m-1}{k} = \dim(\mathcal{RM}(r,m-1) \circledast \mathcal{RM}(r-1,m-1))
\end{aligned}$$

por las propiedades de los coeficientes binomiales y haciendo el correspondiente cambio de variable. ■

Definición 26. Dado r un entero tal que $0 \leq r \leq m$, se define recursivamente $\mathcal{RM}(r,m)$ para el orden canónico de $\mathbb{F}_2^m$ teniendo en cuenta el caso base de la nota 17 y empleando la fórmula (6). ◂

Proposición 27. *Dado r natural tal que $0 \leq r \leq m$, la distancia mínima de $\mathcal{RM}(r,m)$ es 2^{m-r}.*

Demostración. Consecuencia del ejemplo 7, calculando primero esta a mano en el caso base. ■

Lema 28. *Se tiene que $G(0,m) = (1\ 1\ \cdots\ 1)$ y $G(m,m) = \begin{pmatrix} G(m-1,m) \\ 0\ \cdots\ 0\ 1 \end{pmatrix}$.*

Demostración. Ambas igualdades son consecuencia inmediata de cómo viene dada la matriz generadora correspondiente en (4). La primera se debe a que la aplicación ψ_m verifica que $\psi_m(1) = \mathbf{1}$. La segunda se sigue de que ψ_m es un homomorfismo de álgebras y el lema 23, pues tenemos de esta manera que $\psi_m(x_1 \cdots x_m) = \psi_m(x_1) \cdots \psi_m(x_m) = 0 \ldots 01$ y el resto de polinomios que quedan tienen por palabras características a las del código $\mathcal{RM}(m-1,m)$, cuya matriz generadora es $G(m-1,m)$. ■

Proposición 29. *Dado r natural tal que $0 \leq r \leq m$, se tiene que la matriz generadora de $\mathcal{RM}(r,m)$ viene dada de manera recursiva a partir de las matrices generadoras de $\mathcal{RM}(r,m-1)$ y $\mathcal{RM}(r-1,m-1)$ como sigue:*

$$G(r,m) = \begin{pmatrix} G(r,m-1) & G(r,m-1) \\ 0 & G(r-1,m-1) \end{pmatrix}. \tag{8}$$

Demostración. Consecuencia del ejemplo 7, teniendo en cuenta que tenemos por caso base los dos casos recogidos en el lema 28. ■

Ejemplo 30. Usando la proposición 29 que acabamos de demostrar, vamos a construir $G(1,3)$. Esto es, vamos a dar la matriz generadora de $\mathcal{RM}(1,3)$ para el orden canónico de $\mathbb{F}_2^m$. De esta manera, podemos enumerar las palabras de $\mathcal{RM}(1,3)$ para este orden fácilmente. En virtud de lo que hemos visto, por inducción, se tiene que

$$\begin{aligned} G(1,3) &= \begin{pmatrix} G(1,2) & G(1,2) \\ 0 & G(0,2) \end{pmatrix} = \begin{pmatrix} G(1,1) & G(1,1) & G(1,1) & G(1,1) \\ 0\ \ 0 & G(0,1) & 0\ \ 0 & G(0,1) \\ 0\ \ 0 & 0\ \ 0 & 1\ \ 1 & 1\ \ 1 \end{pmatrix} \\ &= \begin{pmatrix} G(0,1) & G(0,1) & G(0,1) & G(0,1) \\ 0\ \ 1 & 0\ \ 1 & 0\ \ 1 & 0\ \ 1 \\ 0\ \ 0 & 1\ \ 1 & 0\ \ 0 & 1\ \ 1 \\ 0\ \ 0 & 0\ \ 0 & 1\ \ 1 & 1\ \ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \end{aligned}$$

Además, por la fórmula dada en la proposición 27, se tiene que la distancia mínima es $2^{3-1} = 2^2 = 4$. Así, hemos concluido que este es un código de Reed-Muller binario cuya longitud y dimensión vienen a ser 8 y 4, respectivamente, con distancia mínima 4 también. Por tanto, pese a las buenas capacidades para corregir errores, este código no resulta ser de bajo coste. Sin embargo, es posible dar con otro código que es lo bastante eficaz en estos dos aspectos, como se va a comprobar al final del artículo. ◂

Nota 31. Cabe resaltar que el orden dado en la definición 22 no es único. A saber, en las mismas condiciones enunciadas, también podemos tomar el orden inducido por la asignación

$$\begin{array}{ccc} 1 & \longrightarrow & (0,0,0,\ldots,0,0); \\ 2 & \longrightarrow & (1,0,0,\ldots,0,0); \\ 3 & \longrightarrow & (0,1,0,\ldots,0,0); \\ 4 & \longrightarrow & (1,1,0,\ldots,0,0); \\ 5 & \longrightarrow & (0,0,1,\ldots,0,0); \\ \vdots & \vdots & \vdots \\ 2^m-2 & \longrightarrow & (1,0,1,\ldots,1,1); \\ 2^m-1 & \longrightarrow & (0,1,1,\ldots,1,1); \\ 2^m & \longrightarrow & (1,1,1,\ldots,1,1). \end{array}$$

Este orden parece más «coherente» y los resultados se cumplen exactamente igual, cambiando algunos detalles. Sin embargo, se ha tomado el otro debido al funcionamiento interno del comando `Tuples` en Mathematica, ya que este genera automáticamente el orden dado en la definición 22. ◂

3.2.3. Construcción geométrica

Para terminar, vamos a obtener información adicional acerca de los códigos de Reed-Muller binarios desde un punto de vista geométrico. Para ello, se emplea el $\mathbb{F}_2$-espacio vectorial $\mathbb{F}_2^m$ como geometría finita, que se denota por $EG(m,2)$. Obtenemos así una caracterización geométrica de $\mathcal{RM}(r,m)$.

Definición 32. Dados un subespacio vectorial V de $\mathbb{F}_2^m$ y un punto $\boldsymbol{a} \in EG(m,2)$, una **variedad afín** que pasa por $\boldsymbol{a}$ y tiene dirección V es la clase de equivalencia $\boldsymbol{a}+V = \{\boldsymbol{a}+\boldsymbol{v} \mid \boldsymbol{v} \in V\}$. Llamaremos **dimensión** de la variedad $\boldsymbol{a}+V$ a la del subespacio vectorial V. Si esta es k, diremos que $\boldsymbol{a}+V$ es una k**-variedad**. ◂

Observación 33. Un subconjunto de $EG(m,2)$ es una k-variedad si y solo si este es el conjunto de soluciones para un sistema de $m-k$ ecuaciones lineales sobre $\mathbb{F}_2$ en m variables con rango $m-k$. Esto se observa mediante equivalencias teniendo en cuenta que, dado V un $\mathbb{F}_2$-subespacio vectorial de $\mathbb{F}_2^m$ de dimensión k (esto es, lo que hemos definido por código lineal binario de longitud m y dimensión k) con H matriz de control, entonces $\boldsymbol{x}$ es palabra código si y solo si se tiene que $\boldsymbol{x}H^\top = 0$. ◂

Sea F un polinomio booleano de m indeterminadas. Fijado un orden $EG(m,2) = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$ en nuestra geometría finita, asociamos a este, además de la correspondiente palabra binaria $\boldsymbol{f}$, el subconjunto

$$S_F \underset{\text{not.}}{\equiv} \{\boldsymbol{v} \in \mathbb{F}_2^m \mid f(\boldsymbol{v}) = 1\} = \{\boldsymbol{v}_i \mid f_i = 1\} \subseteq EG(m,2).$$

Mediante este proceso se obtienen todos los subconjuntos de la geometría finita $EG(m,2)$. Así, diremos que F es el polinomio booleano asociado a S_F y $\boldsymbol{f}$ es la palabra característica asociada a S_F. Nuestro objetivo es describir los códigos de Reed-Muller binarios en términos de las variedades afines de $EG(m,2)$.

Proposición 34. *Si $S \subseteq EG(m,2)$ es una k-variedad, el correspondiente polinomio booleano asociado a esta tiene grado $m-k$. Aunque el recíproco no es cierto en general, sí que lo es para monomios booleanos.*

Demostración. La k-variedad S es el conjunto de soluciones para un sistema de $m-k$ ecuaciones lineales en m variables con rango $m-k$ por la observación 33, en las cuales podemos suponer sin pérdida de generalidad que en la parte derecha tenemos un 1 en todos los casos. Una solución de este sistema lo es también de la ecuación que resulta de multiplicar todos los polinomios de la parte izquierda e igualarlos a 1. El polinomio resultante tiene grado $m-k$. En conclusión, como S es el conjunto de soluciones para la ecuación que se sigue de igualar este a 1, el polinomio booleano asociado a S tiene grado $m-k$. El recíproco es trivialmente cierto si se tienen monomios booleanos. Para cualquier otro caso, basta dar un contraejemplo: existen polinomios booleanos F de grado $k \geq 2$ tales que el subconjunto $S_F \subseteq EG(m,2)$ no es una $(m-k)$-variedad. Sea $F = x_1x_2 + x_3$ un polinomio booleano de 3 variables y grado 2. Por reducción al absurdo, si S_F es una variedad afín, como F es de grado 2, esta es una recta afín. Pero

$$S_F = \{(x_1,x_2,x_3) \in \mathbb{F}_2^3 \mid x_1x_2 + x_3 = 1\} = \{(1,1,0),(0,0,1),(1,0,1),(0,1,1)\},$$

lo cual es absurdo, ya que las rectas afines tienen exactamente dos puntos. ■

Teorema 35. *El código de Reed-Muller binario $\mathcal{RM}(r,m)$ es el subespacio vectorial generado por todas las palabras características asociadas a las variedades afines de $EG(m,2)$ con dimensión al menos $m-r$.*

Demostración. Sea $\boldsymbol{x}$ la palabra característica asociada a una variedad afín $S_{\boldsymbol{x}}$ de dimensión al menos $m-r$. Supongamos que $F_{\boldsymbol{x}}$ es el polinomio booleano que tiene asociada esta palabra característica. En virtud de la proposición 34, se tiene que este polinomio booleano tiene grado menor o igual que r. Así, por como se definen los códigos de Reed-Muller binarios, se tiene que $\boldsymbol{x} \in \mathcal{RM}(r,m)$. Recíprocamente, sea F el polinomio booleano asociado a una palabra $\boldsymbol{x}_F \in \mathcal{RM}(r,m)$. Sabemos que este tiene grado $s \leq r$. Supongamos que $F = \sum_{i=1}^{l} P_i$, siendo P_i con $i \in \{1,\dots,l\}$ los monomios booleanos en los que se descompone F. Obsérvese que estos tienen grado $\deg(P_i) \leq s$ para todo $i \in \{1,\dots,l\}$. Por la linealidad de nuestra aplicación ψ_m se tiene que $\boldsymbol{x}_F = \sum_{i=1}^{l} \boldsymbol{x}_{P_i}$ es la palabra característica asociada a F. Ahora bien, por la segunda parte de la proposición 34, cada $\boldsymbol{x}_{P_i}$ es la palabra característica asociada a una variedad afín de dimensión $m - \deg(P_i) \geq m-s$. Por tanto, la palabra $\boldsymbol{x}_F$ es suma de palabras características de variedades afines con dimensión al menos $m-s \geq m-r$. ■

Definición 36. Dado r un entero tal que $0 \leq r \leq m$, se define geométricamente $\mathcal{RM}(r,m)$ como el subespacio vectorial generado por todas las palabras características asociadas a las variedades afines de $EG(m,2)$ con dimensión al menos $m-r$. ◂

La siguiente es una consecuencia importante del teorema 35, empleando la parte final de la proposición 20, necesaria para entender la decodificación en los códigos de Reed-Muller binarios.

Corolario 37. *Todas las palabras características asociadas a conjuntos que sean $(r+1)$-variedades de $EG(m,2)$ son elementos de $\mathcal{RM}(r,m)^{\perp}$.*

3.3. Métodos de codificación y decodificación en códigos de Reed-Muller binarios

Para terminar, vamos a estudiar los métodos de codificación y decodificación propios para códigos de Reed-Muller binarios. Recogemos en una tabla los dos algoritmos que describen estos dos procesos.

Empezamos estableciendo un procedimiento de codificación basado en el resultado general siguiente que permite obtener un orden para los elementos de $\mathbb{F}_2^m$ tal que $\mathcal{RM}(r,m)$ admite una matriz generadora dada en forma estándar.

Proposición 38. *Toda matriz binaria $G \in Mat_{s\times n}(\mathbb{F}_2)$ de rango máximo $s \leq n$ puede llevarse, realizando operaciones elementales por filas y permutaciones en las columnas, a una matriz dada en forma estándar.*

Demostración. Sea $G = (g_{ij})_{(i,j)\in(\{1,\dots,s\}\times\{1,\dots,n\})} \in \mathrm{Mat}_{s\times n}(\mathbb{F}_2)$. Como el rango de G coincide con el número de filas s, necesariamente en cada una de estas ha de existir al menos un elemento no nulo.

1. Si $g_{11} = 1$, para cada $i \in \{2,\dots,s\}$, sustituimos cada fila i-ésima de G por la fila i-ésima de G menos su primera fila. De esta forma, a través de operaciones elementales por filas, transformamos G en

$$\left(\begin{array}{c|ccc} 1 & * & \cdots & * \\ \hline 0 & & & \\ \vdots & & B & \\ 0 & & & \end{array}\right), \tag{9}$$

   que sigue teniendo rango s, siendo $B \in \mathrm{Mat}_{(s-1)\times(n-1)}(\mathbb{F}_2)$.
2. Si $g_{11} = 0$, ha de existir una columna j de G tal que $g_{1j} = 1$. Permutamos las columnas 1 y j de G entre sí. La matriz que así obtenemos está en las condiciones descritas en el paso 1 anterior.

En cualquier caso, obtenemos una matriz del tipo (9). Estudiamos ahora la posición $(2,2)$ de esta nueva matriz. Sin pérdida de generalidad, podemos suponer que $g_{22} = 1$. Así, realizando una vez más operaciones elementales por filas, podemos llevar esta matriz a una de la forma

$$\left(\begin{array}{cc|ccc} 1 & 0 & * & \cdots & * \\ 0 & 1 & * & \cdots & * \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & D & \\ 0 & 0 & & & \end{array}\right),$$

que vuelve a ser una matriz de rango s, donde $D \in \mathrm{Mat}_{(s-2)\times(n-2)}(\mathbb{F}_2)$. Reiterando este proceso un total de s veces, obtenemos finalmente una matriz dada en forma estándar. ■

Corolario 39. *Dado el código de Reed-Muller binario $\mathcal{RM}(r,m)$, existe un orden para los elementos de $\mathbb{F}_2^m$ tal que $\mathcal{RM}(r,m)$ admite una matriz generadora dada en forma estándar.*

Demostración. Basta hacer uso de la proposición 38 con una matriz generadora, teniendo en cuenta que permutar dos columnas de esta entre sí equivale a intercambiar la posición de dos letras en todas las palabras del código $\mathcal{RM}(r,m)$. Por la observación 18, esto no cambia el código de Reed-Muller binario. ■

Definición 40. Resulta natural referirnos al orden de $\mathbb{F}_2^m$ obtenido por aplicación del corolario 39 como *orden estándar* de $\mathbb{F}_2^m$ respecto de $\mathcal{RM}(r,m)$. Este dista mucho de ser único en general. ◂

La mayor ventaja que tienen los códigos de Reed-Muller binarios es su fácil decodificación por el llamado *algoritmo de Reed*. Este toma como base un método muy práctico y eficiente de decodificación para cierto tipo de códigos lineales, del cual hemos comentado algo al comienzo. Su principal característica es que no emplea síndromes, pues detecta directamente las posiciones donde se han producido los errores.

3.3.1. Algoritmo de Reed

Hay muchas formas de presentar el algoritmo de Reed, pero la mejor manera de hacerlo es en términos de la geometría finita $EG(m,2)$. Supongamos que, fijado un orden $EG(m,2) = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$, se ha enviado una palabra código $\boldsymbol{c} \in \mathcal{RM}(r,m)$, a partir de la cual recibimos $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$. Asumiendo que $\omega(\boldsymbol{e}) \leq 2^{m-r-1} - 1$, este tiene que ser capaz de determinar las posiciones $i \in \{1, 2, \dots, n\}$ en las que se hayan cometido errores durante la transmisión (puesto que, conocidas estas, dado que estamos en el caso binario, la decodificación será trivial). Para ello, reformularemos el problema empleando las 0-variedades de $EG(m,2)$.

Definición 41. Sea S una k-variedad con palabra característica asociada $\boldsymbol{x}_S$. Recibida $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ tal que $\boldsymbol{c} \in \mathcal{RM}(r,m)$, con $\omega(\boldsymbol{e}) \leq 2^{m-r-1} - 1$, la *paridad* de S respecto de $\boldsymbol{y}$ no es más que la paridad, en el sentido binario usual (donde 0 representa par, mientras que 1, impar), de $\langle \boldsymbol{x}_S, \boldsymbol{e} \rangle \equiv \omega(\boldsymbol{x}_S \boldsymbol{e}) \in \mathbb{F}_2$. ◂

En estas circunstancias, por como vienen dadas las palabras características de las 0-variedades, determinar si la i-ésima coordenada de $\boldsymbol{y}$ es correcta o no para cada $i \in \{1, 2, \dots, n\}$ equivale a calcular la paridad correspondiente a la 0-variedad $S = \{\boldsymbol{v}_i\}$. Desafortunadamente, esta no puede ser evaluada directamente. Afortunadamente, tenemos el siguiente resultado, que es consecuencia inmediata del corolario 37.

Proposición 42. *Bajo las condiciones enunciadas, si S es una $(r+1)$-variedad con palabra característica asociada $\boldsymbol{x}_S$, se tiene que la paridad de S respecto de $\boldsymbol{y}$ coincide con la de $\omega(\boldsymbol{x}_S \boldsymbol{y})$.*

La idea consiste en utilizar el conocimiento de las paridades respecto de $\boldsymbol{y}$ con todas las $(r+1)$-variedades para determinar la paridad respecto de $\boldsymbol{y}$ del resto de k-variedades, con $k \leq r$. Para ello, se procede por «lógica mayoritaria». Esto es, dada una k-variedad S para la que conocemos todas las paridades respecto de $\boldsymbol{y}$ en las $(k+1)$-variedades que la contienen, diremos que su paridad respecto de $\boldsymbol{y}$ coincide con aquella que tienen la mayoría de estas variedades afines. El resultado que demuestra la veracidad de este mecanismo es la segunda clave del algoritmo de Reed y requiere de otros dos resultados técnicos.

Lema 43. *Para cada k-variedad $S = \boldsymbol{a} + V$ de $EG(m,2)$ y cada punto $\boldsymbol{b} \in EG(m,2) - S$ que consideremos, existe una única variedad afín de dimensión $k+1$ que contiene tanto a S como a $\boldsymbol{b}$.*

Lema 44. *Cada k-variedad de $EG(m,2)$, con $k < m$, está contenida exactamente en $2^{m-k} - 1$ variedades afines de dimensión $k+1$.*

El primero de estos dos es un análogo al quinto postulado de Euclídes para nuestra geometría finita. Es, por tanto, un resultado de existencia y unicidad, cuya prueba es similar a la que se da para este en un curso de geometría elemental. El segundo, por otro lado, se sigue del anterior haciendo un rápido argumento de combinatoria entre variedades afines.

Teorema 45 (criterio de la lógica mayoritaria, CLM). *Bajo estas condiciones, dada una k-variedad S, con $k \leq r$, se tiene que la paridad de S respecto de $\boldsymbol{y}$ coincide con la que tienen la mayoría de las $(k+1)$-variedades que contienen a S.*

Demostración. Por el lema 44, tenemos que S está contenida en $2^{m-k} - 1$ variedades afines de dimensión $k+1$, donde cada una de estas viene determinada de forma unívoca dando un punto exterior a S en virtud del lema 43. Por hipótesis, dado que el número de errores en $\boldsymbol{y}$ no supera los $2^{m-r-1} - 1$, existen a lo más $2^{m-r-1} - 1$ variedades afines de dimensión $k+1$ que contienen a S determinadas por los puntos exteriores correspondientes a una coordenada incorrecta de $\boldsymbol{y}$. El resto de las $(k+1)$-variedades tienen la propiedad de que no contienen puntos exteriores a S correspondientes a coordenadas erróneas de $\boldsymbol{y}$ por la unicidad probada en el lema 43. En efecto, si alguna de estas variedades afines contiene algún punto exterior a S correspondiente a una coordenada errónea de $\boldsymbol{y}$, necesariamente debería ser una de las anteriores debido a esta unicidad. Así, por la definición 41, estas tienen la misma paridad respecto de $\boldsymbol{y}$ que S. En resumen, por todo lo mencionado, el número de $(k+1)$-variedades con la misma paridad respecto de $\boldsymbol{y}$ que S es al menos de $(2^{m-k} - 1) - (2^{m-r-1} - 1)$. El resultado a partir de aquí se debe a que trivialmente se cumple la desigualdad $2^{m-k} - 2^{m-r-1} \geq 2^{m-r-1}$, pues $k \leq r$. ■

Ejemplo 46. Supongamos recibida la palabra 01100001, codificada mediante el orden canónico de $\mathbb{F}_2^3$ en $\mathcal{RM}(1,3)$, donde se ha producido un error. Vamos a decodificarla mediante el algoritmo de Reed. Primero calculamos la paridad de los planos afines de $EG(3,2) = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5, \boldsymbol{v}_6, \boldsymbol{v}_7, \boldsymbol{v}_8\}$ empleando la proposición 42. Se tiene la tabla siguiente:

plano	palabra	paridad	plano	palabra	paridad
$\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4\}$	11110000	par	$\{\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_5, \boldsymbol{v}_8\}$	01101001	impar
$\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_5, \boldsymbol{v}_6\}$	11001100	impar	$\{\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_6, \boldsymbol{v}_7\}$	01100110	par
$\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_7, \boldsymbol{v}_8\}$	11000011	par	$\{\boldsymbol{v}_2, \boldsymbol{v}_4, \boldsymbol{v}_5, \boldsymbol{v}_7\}$	01011010	impar
$\{\boldsymbol{v}_1, \boldsymbol{v}_3, \boldsymbol{v}_5, \boldsymbol{v}_7\}$	10101010	impar	$\{\boldsymbol{v}_2, \boldsymbol{v}_4, \boldsymbol{v}_6, \boldsymbol{v}_8\}$	01010101	par
$\{\boldsymbol{v}_1, \boldsymbol{v}_3, \boldsymbol{v}_6, \boldsymbol{v}_8\}$	10100101	par	$\{\boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5, \boldsymbol{v}_6\}$	00111100	impar
$\{\boldsymbol{v}_1, \boldsymbol{v}_4, \boldsymbol{v}_5, \boldsymbol{v}_8\}$	10011001	impar	$\{\boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_7, \boldsymbol{v}_8\}$	00110011	par
$\{\boldsymbol{v}_1, \boldsymbol{v}_4, \boldsymbol{v}_6, \boldsymbol{v}_7\}$	10010110	par	$\{\boldsymbol{v}_5, \boldsymbol{v}_6, \boldsymbol{v}_7, \boldsymbol{v}_8\}$	00001111	impar

Ahora, por el criterio de la lógica mayoritaria, se calculan las paridades correspondientes a las rectas afines. Estas son las siguientes:

recta	paridad	recta	paridad	recta	paridad	recta	paridad
$\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$	par	$\{\boldsymbol{v}_2, \boldsymbol{v}_3\}$	par	$\{\boldsymbol{v}_3, \boldsymbol{v}_5\}$	impar	$\{\boldsymbol{v}_4, \boldsymbol{v}_8\}$	par
$\{\boldsymbol{v}_1, \boldsymbol{v}_3\}$	par	$\{\boldsymbol{v}_2, \boldsymbol{v}_4\}$	par	$\{\boldsymbol{v}_3, \boldsymbol{v}_6\}$	par	$\{\boldsymbol{v}_5, \boldsymbol{v}_6\}$	impar
$\{\boldsymbol{v}_1, \boldsymbol{v}_4\}$	par	$\{\boldsymbol{v}_2, \boldsymbol{v}_5\}$	impar	$\{\boldsymbol{v}_3, \boldsymbol{v}_7\}$	par	$\{\boldsymbol{v}_5, \boldsymbol{v}_7\}$	impar
$\{\boldsymbol{v}_1, \boldsymbol{v}_5\}$	impar	$\{\boldsymbol{v}_2, \boldsymbol{v}_6\}$	par	$\{\boldsymbol{v}_3, \boldsymbol{v}_8\}$	par	$\{\boldsymbol{v}_5, \boldsymbol{v}_8\}$	impar
$\{\boldsymbol{v}_1, \boldsymbol{v}_6\}$	par	$\{\boldsymbol{v}_2, \boldsymbol{v}_7\}$	par	$\{\boldsymbol{v}_4, \boldsymbol{v}_5\}$	impar	$\{\boldsymbol{v}_6, \boldsymbol{v}_7\}$	par
$\{\boldsymbol{v}_1, \boldsymbol{v}_7\}$	par	$\{\boldsymbol{v}_2, \boldsymbol{v}_8\}$	par	$\{\boldsymbol{v}_4, \boldsymbol{v}_6\}$	par	$\{\boldsymbol{v}_6, \boldsymbol{v}_8\}$	par
$\{\boldsymbol{v}_1, \boldsymbol{v}_8\}$	par	$\{\boldsymbol{v}_3, \boldsymbol{v}_4\}$	par	$\{\boldsymbol{v}_4, \boldsymbol{v}_7\}$	par	$\{\boldsymbol{v}_7, \boldsymbol{v}_8\}$	par

De la misma forma, si obtenemos la paridad de cada punto, se observa que el único impar es el $\boldsymbol{v}_5$. En definitiva, concluimos que el único error dado durante la transmisión de información se encuentra en la quinta posición. Por tanto, decodificamos la palabra recibida como 01101001. ◂

Dados los parámetros r y m del código de Reed-Muller binario, calculamos la matriz generadora dada en forma estándar con la que vamos a trabajar. Hecho esto, para cada palabra $\boldsymbol{x}$ a codificar, procedemos a su codificación tal y como ya se explicó en su momento al comienzo del artículo.

Algoritmo 1 (Cálculo de la matriz estándar).

```
subrutina MATRIZ ESTÁNDAR(r,m)
    calcular matriz generadora usando la proposición 29
    si la matriz ya está en forma estándar, entonces
        no hacer nada
    en caso contrario
        aplicar procedimiento de la proposición 38
    fin si
    devolver la matriz G dada en forma estándar
fin subrutina
```

Algoritmo 2 (Codificación).

```
subrutina CODIFICAR(G,x)
    devolver el producto xG
fin subrutina
```

Dado el código de Reed-Muller binario $\mathcal{RM}(r, m)$, recibida una palabra $\boldsymbol{y}$ arbitraria, se describen los pasos a seguir para su decodificación mediante el algoritmo de Reed.

Algoritmo 3 (Decodificación).

```
subrutina DECODIFICAR(y,r,m)
    si la palabra y tiene más de 2^(m−r−1) − 1 errores, entonces
        no se puede decodificar de manera única
    en caso contrario
        calcular la paridad respecto de y para las (r + 1)-variedades de EG(m, 2) por la proposición 42
        para k desde r hasta 0 hacer
            calcular la paridad respecto de y para todas las k-variedades de EG(m, 2) usando el CLM
        fin para
        corregir las coordenadas de y correspondientes a todas las 0-variedades impares respecto de y
        devolver la palabra corregida
    fin si
fin subrutina
```

Referencias

[1] De Arriba De La Hera, Andoni. *Códigos de Reed-Muller*. Trabajo de Fin de Grado. Zientzia eta Teknologia Fakultatea-Facultad de Ciencia y Tecnología (ZTF-FCT), Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU), 2016. URL: http://hdl.handle.net/10810/20121.

[2] Iranzo Aznar, María Jesús y Pérez Monasor, Francisco. *LECCIONES de Elementos de Álgebra. Aplicaciones*. Facultad de Matemáticas, Universidad de Valencia. URL: http://www.uv.es/iranzo/lecciones_de_codigos.pdf.

[3] Muller, David Eugene. «Application of Boolean algebra to switching circuit design and to error detection». En: *Transactions of the IRE Professional Group on Electronic Computers* EC-3.3 (sep. de 1954), págs. 6-12. ISSN: 2168-1740. https://doi.org/10.1109/IREPGELC.1954.6499441.

[4] Reed, Irving Stoy. «A class of multiple-error-correcting codes and the decoding scheme». En: *Transactions of the IRE Professional Group on Information Theory* 4.4 (1954), págs. 38-49. ISSN: 2168-2690. https://doi.org/10.1109/TIT.1954.1057465.

[5] Shannon, Claude Elwood. «A mathematical theory of communication». En: *The Bell System Technical Journal* 27 (1948), págs. 379-423, 623-656. ISSN: 0005-8580. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

A. Anexo: algunos programas en Mathematica

Para terminar, se incluye el archivo con los programas en Mathematica (junto con un PDF generado a partir de este) que son útiles para llevar a cabo la codificación y decodificación en códigos de Reed-Muller binarios. En particular, estamos interesados en el código $\mathcal{RM}(1,5)$ utilizado por la NASA cuando se llevo a cabo la misión del Mariner 9 (véase el ejemplo 47). Todos estos programas tienen como objetivo realizar los procedimientos de codificación y decodificación de manera eficaz para dicho código en especial, aunque resulten válidos para todos los de la forma $\mathcal{RM}(1,m)$ con m número natural arbitrario. Ambos archivos se pueden encontrar aquí:

Programa en Mathematica: https://temat.es/articulo/2019-p45/a-anexocodrm-nb (anexocodrm.nb)

Código en PDF: https://temat.es/articulo/2019-p45/a-anexocodrm-pdf (anexocodrm.pdf)

Ejemplo 47. Vamos a aplicar estos algoritmos para comprobar que $\mathcal{RM}(1,5)$ es el código de Reed-Muller binario que empleó la sonda Mariner 9 para obtener las primeras fotografías de la superficie marciana. Lo primero es observar que este es un código lineal de longitud 32. Calculamos una matriz generadora escribiendo en Mathematica la instrucción correspondiente, obteniendo así

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

que tiene rango máximo 6 (coincide con el número de filas). Por consiguiente, esta es también la dimensión del código. La distancia mínima se obtiene a partir de la fórmula dada en la proposición 27, y es 16. En definitiva, este es el código de Reed-Muller binario mencionado al comienzo del artículo. Vamos a hacer un ejemplo de decodificación. Supongamos que se está enviando información codificada en $\mathcal{RM}(1,5)$ mediante el orden estándar de $\mathbb{F}_2^5$ y que recibimos la palabra 10011101010001101110010010001001 que no tiene más de siete errores. Vamos a decodificar esta mediante el algoritmo de Reed. Escribiendo la instrucción correspondiente en Mathematica, obtenemos la palabra 10010001111001100110011010011001 (obsérvese que hemos sido capaces de corregir los siete errores que se habían dado durante la transmisión de información). Así, como se sabe que esta ha sido codificada mediante una matriz dada en forma estándar y nuestro código es de dimensión 6, como ya hemos comprobado, la palabra original enviada es 100100. ◂