Códigos de Reed-Muller

Tobias Briones

Abril 2021

Capítulo 1

Marco Teórico

1.0.1. Antecedentes

En el momento en el que un transmisor emite un mensaje mediante el medio hacia un receptor, a causa del medio (aire o cable) por el que se transmiten los paquetes el receptor puede recibir y recibirá más de alguna vez mensajes no íntegros distintos a los emitidos originalmente por el transmisor. En esta etapa entra en acción la teoría de códigos para la detección y corrección de errores. Ejemplos de estos errores son por ejemplo cambiar el valor de un bit, de manera que si se transmite la cadena 10011101, se puede recibir al otro lado un 10011100 el cual cambio el ultimo bit del mensaje. Podemos detectar que hubo un error porque la *n-tupla* recibida no pertenece a un cierto *código* y corregir el error en caso de ser corregible y de estar suficientemente cerca de su valor original para poder inferir este.

La corrección de errores se puede llevar a cabo en una gran lista de dispositivos tales como RAM, ROM, discos de almacenamiento, DVDs, memorias USB y en comunicaciones inalámbricas por medio de antena celular, satélite y otros. Los codigos de Reed-Muller son utilizados principalmente en aplicaciones inalámbricas como el $5G$ y en las misiones de la NASA (AMS, 2021):

> La misión de la nave espacial Mariner tomó fotografías (sin color) de Marte con éxito en 1965. Las imágenes eran de $200x200$ y a cada píxel se le asignaba uno de los 64 niveles de brillo (seis bits). Dado que los datos se transmitieron a aproximadamente 8 bits por segundo (!), La transmisión de una sola imagen tomó aproximadamente 8 horas. Cuando el Mariner 9 entró en órbita alrededor de Marte en 1972, se estaban obteniendo imágenes mucho mejores. Esto se debió a que la nave espacial usó un código Reed-Muller que tenía $6bits$ de información y $26bits$ adicionales para proporcionar corrección de errores (las palabras del código tenían $32bits$ de longitud). Aunque la velocidad de transmisión ahora era de unos 16,000 bits por segundo, las imágenes individuales eran más grandes, por lo que las cámaras estaban adquiriendo unos 100,000 bits por segundo. Esto significó que las imágenes se almacenaron para su transmisión. Cuando Viking aterrizó en Marte en 1976, la tecnología había mejorado, de modo que se estaban obteniendo imágenes en color. El enfoque inicial para hacer esto fue tomar imágenes separadas de la misma ubicación usando filtros de tres colores diferentes. Las imágenes en blanco y negro separadas obtenidas a través de cada filtro se transmitieron y luego se reconstruyó una imagen en color a partir de la información de las tres imágenes en blanco y negro.
>
> La NASA ha utilizado muchos códigos de corrección de errores diferentes. Para las misiones entre 1969 y 1977, la nave espacial Mariner utilizó un código Reed-Muller. El ruido al que estaban sujetas estas naves espaciales estaba bien aproximado por una çurva de campana"(distribución normal), por lo que los códigos Reed-Muller se adaptaban bien a la situación.

La idea de corregir errores mediante códigos ya existía tal como lo había probado Claude Shannon. Para poder implementar esta idea prácticamente luego tomaron turno otros científicos de la computación como Richard Hamming (1915-1998) y Marcel J. E. Golay (1902-1989) para poder hacer las construcciones de estos sistemas de códigos.

Los tipos de errores en el canal de comunicación pueden ser:

- Ruido o distorsión eléctrica: Existen muchos factores que alteran la señal enviada, incluso el mismo ruido electromagnético que los mismos cables generan (crosstalk o alien crosstalk). Así también muchos otros factores como ondas de sonido, ruido como electricidad de motores, interruptores de potencia, etc.
- Errores de ráfaga: Dos o más (muchos o miles) de bits han sido cambiados.
- Errores de bit aleatorios: Los bits se han reorganizado en otro orden.
- Cross talk y eco: Como se menciono en el primer elemento de esta lista, el crosstalk se produce cuando el cable esta rodeado de otros cables. El eco es parecido al cross talk, pero ocurre en una misma linea de transmisión.

Sobre los códigos de Reed-Muller

Los códigos de Reed-Muller son una familia infinita de códigos, que toman su nombre de los dos matemáticos que los propusieron en el año 1954 al mismo tiempo, en trabajos independientes: I. S. Reed (Irving Stoy Reed, matemático e ingeniero estadounidense, 1923 - 2012) y D. E. Muller (David Eugene Muller, matemático e informático teórico estadounidense, 1924 - 2008).

Ambos centraron su estudio en los códigos de Reed-Muller binarios. Hoy se sabe que el primero en realizar su construcción fue Muller, mientras que su estudio en detalle y la sencilla decodificación por la que son tan conocidos e importantes es obra de Reed. Posteriormente, estos fueron generalizados a cualquier cuerpo finito en 1968. Estos estudios están situados dentro en la Teoría de la Información, cuyas bases fueron establecidas por Shannon (Claude Elwood Shannon, matemático; ingeniero electrónico y criptógrafo estadounidense, 1916 - 2001) a través de un artículo publicado en el Bell System Technical Journal en la década de los años 40 titulado "A Mathematical Theory of Communication". Hoy en día, la Teoría de la Información es una rama de las matemáticas y de la computación que se ocupa del estudio de la información y de todo lo relacionado con ella.

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 trataba por tanto de un código lineal de bajo coste debido a su pequeña dimensión en comparación a su longitud, y con buenas capacidades de corrección de errores por su elevada distancia mínima.

Como se había mencionado en una sección anterior, los códigos de Reed-Muller han sido utilizados en la NASA por sus ventajas de ser códigos lineales, eficientes y fáciles de modelar, esto los hace posibles de implementar a diferencia de códigos no-lineales. Una de las misiones más destacadas que se llevo a cabo con el uso de estos códigos fue la de la sonda Mariner 9, que fue la primera que permitió la observación fotográfica de la superficie del planeta Marte. La sonda Mariner 9 fue lanzada hacia su destino el 30 de mayo de 1971, llegando a Marte 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 de Marte adquiridas por las sondas Mariner 6 y 7, tenía como objetivo mostrar las primeras fotografías de la superficie marciana. 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. Finalmente, en 1972, cuando por fin amainaron las tormentas, se obtuvieron las primeras fotografías claras del planeta que cambiaron completamente la visión que se tenía hasta entonces del planeta rojo. La sonda tomó fotografías en blanco y negro de $600 \times 600 = 3600$ píxeles, donde a cada píxel se le asignó una $6-tupla$ para representar el brillo. De esta manera, cada píxel era codificado como una palabra de longitud 32, esto es, se emplearon 26 bits de redundancia.

Polinomios mínimos

Campo de Galois

Los elementos del campo de Galois $gf(p^n)$ son definidos como

$$GF(p^n) = (0, 1, 2, ..., p-1) \cup (p, p+1, p+2, ..., p+p-1)$$
$$\cup ... \cup (p^{n-1}, p^{n-1}+1, p^{n-1}+2, ..., p^{n-1}+p-1$$

Donde $p \in \mathbb{P}$ y $n \in \mathbb{Z}^+$. El orden del campo está dado por p^n mientras que p es llamado la característica del campo.

Por ejemplo

$$GF(2^3) = (0, 1, 2, 2+1, 2^2, 2^2+1, 2^2+2, 2^2+2+1) = (1, 2, 3, 4, 5, 6, 7)$$

consiste de $2^3 = 8$ elementos donde cada elemento es un polinomio de grado menor o igual a 2 evaluado en 2.

Recordar que las operaciones en $GF(2)$ (0s y 1s) se definen como XOR para la suma y AND para el producto. Estas serán las operaciones que se computarán por ejemplo, cuando se trabaje con $\mathbb{F}_2^n$.

Definición 1.0.1 (Polinomio mínimo)**.** Sea $\alpha \in GF(q^m)$. El **polinomio mínimo** de α con respecto al subcampo $GF(q)$ es el polinomio mónico $p(x) \in GF(q)[x]$ de grado mínimo tal que $p(\alpha) = 0$.

Proposición 1.0.1. *El polinomio mínimo mónico de un elemento α con respecto al subcampo $GF(q)$ es único.*

Prueba:

Suponer que $f(x)$ y $g(x)$ son polinomios mínimos mónicos distintos de $\alpha \in GF(q^m)$ con respecto a $GF(q)$. Ya que por hipótesis $f(x) \leq g(x)$, entonces existe un polinomio $r(x) \neq \mathbf{0}$ tal que $f(x) = g(x) + r(x)$, por lo que $r(\alpha) = 0$ y $deg(r(x)) < deg(f(x))$. Esto es una contradicción de la hipótisis de que $f(x)$ es un polinomio mínimo. Por tanto se ha probado la proposición requerida.

Definición 1.0.2 (Código lineal)**.** Un **código lineal C** de longitud n y dimensión s sobre $\mathbb{F}_{\shortparallel}$ es un F_q-subespacio vectorial de $\mathbb{F}_{\shortparallel}^{\ltimes}$ de dimensión $s \leq n$.

Definición 1.0.3 (Distancia de Hamming)**.**

$$d(C) = min_{u,v \in C \wedge u \neq v}\{d(u, v)\}$$

$$d(u, v) = \#\{i : 1 \leq i \leq n \wedge u_i \neq v_i\}$$

Definición 1.0.4 (Peso de una palabra código)**.** El peso de una palabra código $\mathbf{x} \in C$ es:

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

También considerar la definición de peso de un código lineal $wt(C) = min\{wt(\mathbf{c}) | \mathbf{c} \in C \wedge \mathbf{c} \neq 0\}$.

Definición 1.0.5 (Código Dual)**.** El **código dual** $C^{\perp}$ de un código lineal $C \subseteq F_q^n$ de dimensión s se define como:

$$C^{\perp} = \{\mathbf{x} \in \mathbb{F}_{\shortparallel}^{\ltimes} | < \mathbf{x}, \mathbf{c} >= 0, \forall \mathbf{c} \in C$$

Definición 1.0.6 (Código Autodual)**.** Se dice que un código lineal C es autodual si $C = C^{\perp}$.

Definición 1.0.7 (Matriz de Control)**.** La **matriz de control** del código lineal C es la matriz generadora de $C^{\perp}$ dada como $H \in \mathbb{M}_{(m-s)xn}(\mathbb{F}_{\shortparallel})$ tal que

$$C = \{\mathbf{x} \in \mathbb{F}_{\shortparallel}^{\ltimes} | \mathbf{x}H^T = \mathbf{0}\}$$

Luego de esto, notar que el dual $C^{\perp}$ de C con longitud n y dimensión s es también un código lineal de longitud n pero de dimensión $n - s$ (esto se puede notar ya que su matriz generadora tiene menos filas que la matriz generadora de C). Por tanto, se cumple que $dim(C) + dim(C^{\perp}) = n$.

Proposición 1.0.2. *Sea C un código lineal, entonces $(C^{\perp})^{\perp} = C$.*

1.0.2. Construcción General de los Códigos de Reed-Muller

Los códigos de Reed-Muller son un conjunto infinito de códigos y además forman parte de los códigos de evaluación. A continuación se da una construcción general como referencia.

Definición 1.0.8 (Función evaluación)**.** Sean $P = \{\mathbf{p}_1, ..., \mathbf{p}_n | n \in \mathbb{N}\}$ un subconjunto finito de un objeto geométrico X y V un $\mathscr{F}_q$-espacio vectorial de aplicaciones $f : X \to \mathscr{F}_q$. Se llama **evaluación** en P de las aplicaciones de V a la aplicación:

$$ev_P : V \to \mathbb{F}_{\shortparallel}^{\ltimes}$$

$$f \to ev_P(f) = f(\mathbf{p}_1)...f(\mathbf{p}_n).$$

Como V es un $\mathbb{F}_q$-espacio vectorial, si ev_P resulta ser una aplicación lineal, entonces su imagen directa $ev_P(V)$ será un F_q-subespacio vectorial de F_q^n, es decir, un código lineal de longitud n sobre F_q cuyas palabras son las imágenes de las aplicaciones de V a través de ev_P. Bajo todas estas circunstancias, se conoce al código $ev_P(V)$ así obtenido por código de evaluación en P de las aplicaciones de V. Los parámetros que definen al código pueden deducirse de las propiedades de V como F_q-espacio vectorial. Los códigos de Reed-Muller son un caso particular de códigos de evaluación, donde $P = X = F_q^m$ (que en este caso es un anillo conmutativo y unitario con estructura de F_q-espacio vectorial) y $V = F_q[x1, ..., xm]$ (que, además de ser un F_q-espacio vectorial, también tiene estructura de anillo conmutativo y unitario).

Definición 1.0.9. Una aplicación $q - aria$ de m variables es una aplicación $f : \mathbb{F}_q^m \to \mathbb{F}_q$.

Definición 1.0.10. Sea $F \in F_q[x_1, ..., x_m]$ un polinomio. Si $n = q^m$, se conoce por aplicación polinómica asociada a F a la aplicación $q - aria$ $f : \mathbb{F}_q^m \to \mathbb{F}_q$ tal que, para cualquier $m - tupla$ $(a_1, ..., a_m) \in \mathbb{F}_q^m$, verifica que $f(a_1, ..., a_m) = F(a_1, ..., a_m)$.

Proposición 1.0.3. *Sea $F \in \mathbb{F}_q[x_1, ..., x_m]$. Si $F(\boldsymbol{v}) = 0$ para todo $\boldsymbol{v} \in F_q^m$ (en particular $F^*(\boldsymbol{v}) = 0$ para todo $\boldsymbol{v} \in \mathbb{F}_q^m$, entonces $F^* = 0$.*

Proposición 1.0.4. *Los conjuntos $\mathbb{F}_q^n$ y P_m^q son isomorfos como F_q-álgebras. En consecuencia, cada polinomio de una misma clase de equivalencia determina la misma aplicación polinómica.*

Proposición 1.0.5. *La aplicación evaluación en $\mathbb{F}_q^m$ de los polinomios de $\mathbb{F}_q[x_1, ..., x_m]$ en $\mathbb{F}_q^n$ dada por:*

$$ev_{\mathbb{F}_q^m} : \mathbb{F}_q[x_1, ..., x_m] \to \mathbb{F}_q^n$$
$$F \to ev_{\mathbb{F}_q^m}(F) = \boldsymbol{F} = F_1, ..., F_n$$

es un epimorfismo de anillos, que además es lineal.

Definición 1.0.11 (Aplicación evaluación Isomorfismo)**.**

$$\bar{ev}_{\mathbb{F}_q^m} : P_m^q \to \mathbb{F}_q^n$$

$$\bar{F} \to \bar{ev}_{\mathbb{F}_q^m}(\bar{F}) = ev_{\mathbb{F}_q^m}(F).$$

Definición 1.0.12 (Código de Reed-Muller (general))**.** Sean $p \in \mathbb{N}$ primo y $q = p^s$. Sean m, $r \in \mathbb{N}$ tales que $r \leq m(q-1)$. Un **código de Reed-Muller q-ario** $\mathscr{RM}_q(r, m)$ de orden r y longitud q^m a la imagen directa de $\mathscr{P}_m^p(r)$ a través de la aplicación - Definición (Aplicación evaluación Isomorfismo) -. Además, se define $\mathscr{RM}_q(l, m) = \{0..,0\}, \forall l < 0$.

Ejemplo: Sean $m, s, p \in \mathbb{N}$, p primo, $q = p^s$ y $n = q^m$. Entonces,

$$\mathscr{RM}_q(0,m) = \{0..,0,1..,1,..,1,q-1...q-1\}$$

donde ... y $q-1...q-1$ son n veces.

Se verifican las siguientes propiedades:

- Todo código de Reed-Muller $\mathscr{RM}_q(r,m)$ es equivalente a otro código de Reed-Muller con los mismos parámetros si solo se realizan permutaciones en las letras de dos posiciones fijadas en todas las palabras del código. Por tanto, dado un código de Reed-Muller $\mathscr{RM}_q(r,m)$ bajo un cierto orden en los elementos de $\mathbb{F}_q^m$, al cambiar dicho orden, obtenemos otro código de Reed-Muller, con los mismos parámetros, equivalente al anterior.
- Para todo par de enteros $i \leq j$ (con $i,j \in \{0,1,...,m(q-1)\}$), se tiene que $\mathscr{RM}_q(i,m) \subseteq \mathscr{RM}_q(j,m)$.

Proposición 1.0.6. *La matriz que tiene como filas las palabras características de los monomios reducidos de $P_m^q(r)$ es una matriz generadora de $\mathscr{RM}_q(r,m)$.*

Prueba:

Basta probar que las palabras características de todos los monomios reducidos de $P_m^q(r)$ constituyen una base de $\mathscr{RM}_q(r,m)$. Se sabe que una base de $P_m^q(r)$ está formada por los monomios que contiene. Por tanto, las palabras características de todos los monomios reducidos de este espacio constituyen un sistema generador de $\bar{ev}_{\mathbb{F}_q^m}(P_m^q(r)) = \mathscr{RM}_q(r,m)$ por la linealidad de $\bar{ev}_{\mathbb{F}_q^m}$. Notar que estas también forman un conjunto libre. Sean $F_1,...F_t \in P_m^q(r)$ monomios reducidos distintos. Supongamos mediante un argumento de contradicción que existen $k_1,...,k_t$ escalares, no todos nulos, tales que

$$k_1\bar{ev}_{\mathbb{F}_q^m}(F_1) + ... + k_t\bar{ev}_{\mathbb{F}_q^m}(F_t) = \mathbf{0}$$

Esto es, que existe un conjunto de palabras características asociadas a algunos monomios reducidos de $P_m^q(r)$ linealmente dependientes. Entonces, por la linealidad de $\bar{ev}_{\mathbb{F}_q^m}$,

$$\bar{ev}_{\mathbb{F}_q^m}(k_1F_1 + ... + k_tF_t) = k_1\bar{ev}_{\mathbb{F}_q^m}(F_1) + ... + k_t\bar{ev}_{\mathbb{F}_q^m}(F_t) = \mathbf{0}$$

Como la suma de monomios reducidos es un polinomio reducido, por la proposición 1,0,3 se tiene que necesariamente $k_1F_1 + ... + k_tF_t$ se corresponde con el polinomio idénticamente nulo. Así, dado que todos los monomios $F_1,...,F_t$ son distintos, necesariamente $k_1 = ... = k_t = 0$, contradiciendo que estos escalares no pueden ser todos nulos bajos la hipótesis establecida. En consecuencia, todo conjunto de palabras características asociadas a monomios reducidos de $P_m^q(r)$ es linealmente independiente, por lo que, en particular, las palabras características de todos los monomios reducidos son linealmente independientes.

Proposición 1.0.7. *La dimensión de $\mathscr{RM}_q(r,m)$ como $\mathbb{F}_q$-subespacio vectorial es el número de soluciones de las ecuaciones $i_1 + ... + i_m = t$ tales que $0 \leq i_1,...,i_m \leq q-1$, para todo $t \in \{0,1,...,r\}$. Esto es,*

$$dim(\mathscr{RM}(r,m)) = \sum_{t=0}^{r} |\{(i_1,...,i_m) | i_1 + ... + i_m = t, 0 \leq i_j \leq q-1\}|$$

Prueba:

Por la proposición 1,0,6, basta probar que el número de monomios reducidos de $P_m^q(r)$ coincide con el número de soluciones de las ecuaciones $i_1 + ... + i_m - t$ tales que $0 \leq i_1,...,i_m \leq q-1$ para todo $t \in \{0,1,...,r\}$. Para ello, basta traducir el cálculo del número de monomios reducidos de grado t que se encuentran en $\mathbb{F}_q[x_1,...,x_m]$ a términos combinatorios. Ahora, fijando $t \in \{0,1,...,r\}$, se observa que el número de monomios reducidos de $\mathbb{F}_q[x_1,...,x_m]$ de grado exactamente t coincide con el número de m-tuplas $(i_1,...,i_m)$ tales que $0 \leq i_1,...,i_m \leq q-1$ con $j \in \{1,...,m\}$, siendo $i_1 + ... + i_m = t$. Ahora, este corresponde con el número de soluciones de la ecuación $i_1 + ... + i_m = t$, donde $0 \leq i_1,...,i_m \leq q-1$. Por tanto, con el principio aditivo se obtiene el resultado.

1.0.3. Definición recursiva

Para cada entero positivo m y cada entero r tal que $0 \leq r \leq m$, existe un código de Reed-Muller de orden r $\mathscr{RM}(r, m)$.

Definición 1.0.13 (primer orden)**.** El código de Reed-Muller de primer orden $\mathscr{RM}(1, m)$ son códigos binarios definidos por todos los enteros $m \geq 1$ recursivamente por:

- $\mathscr{RM}(1, 1) = \{00, 01, 10, 11\} = \mathbb{Z}_2^2$
- para $m > 1$, $\mathscr{RM}(1, m) = \{(\mathbf{u}, \mathbf{u}), (\mathbf{u}, \mathbf{u+1}) | \mathbf{u} \in \mathscr{RM}(1, m-1)\}$

Es decir, $\mathscr{RM}(1, 1)$ es el código que consiste en todas las tuplas binarias. $\mathscr{RM}(1, 2)$ es una copia de $\mathscr{RM}(1, 1)$ (duplicando cada tupla, por ejemplo, pasar de Z_2^2 a Z_2^4) y además volver a duplicar dichas tuplas pero con el lado derecho sumándole **1**. Y así recursivamente para definir las siguientes $n - tuplas$ de los siguientes $\mathscr{RM}(1, m)$.

Ejemplo

$$\begin{aligned}
\mathscr{RM}(1,2) = \{&0000, \quad 0011,\\
&0101, \quad 0110,\\
&1010, \quad 1001,\\
&1111, \quad 1100\}
\end{aligned}$$

$$\begin{aligned}
\mathscr{RM}(1,3) = \{&00000000, \quad 00001111,\\
&01010101, \quad 01011010,\\
&10101010, \quad 10100101,\\
&11111111, \quad 11110000,\\
&00110011, \quad 00111100,\\
&01100110, \quad 01101001,\\
&10011001, \quad 10010110,\\
&11001100, \quad 11000011\}
\end{aligned}$$

1.0.4. Construcción (u, u+v)

Proposición 1.0.8 (construcción (u, u +v))**.** *Sea C_i un código lineal q-ario $[n, k_i, d_i]$. Entonces el código C dado por*

$$C = \{(\boldsymbol{u}, \boldsymbol{u+v}) | \boldsymbol{u} \in C_1, \boldsymbol{v} \in C_2\}$$

es un código lineal q-ario $[2n, k_1 + k_2, min(2d_1, d_2)]$.

Prueba:

Tenemos que C es un código lineal q-ario $[2n, k, d]$ para algún k y d. Sea el mapeo $C_1 \times C_2 \to C$ dado por $(c_1, c_2) \to (c_1, c_1 + c_2)$. Esta definición es una biyección, entonces el tamaño de C es igual al tamaño de $C_1 \times C_2$ el cual es igual a $q^{k_1} q^{k_2} = q^{k_1+k_2}$. Por consiguiente, la dimensión de C es $k_1 + k_2$. Considerar una palabra código distinta de cero $(c_1, c_1 + c_2)$ de C. Si $c_2 = 0$ entonces $c_1 \neq 0$ y

$$wt((c_1, c_1 + c_2)) = wt((c_1, c_2)) = 2wt(c_1) \geq 2d_1 \geq min(2d_1, d_2)$$

En otro caso, si $c_2 \neq 0$, entonces

$$wt((c_1, c_1 + c_2)) = wt(c_1) + wt(c_1 + c_2) \geq wr(c_1) + (wt(c_2) - wt(c_1)) \geq wt(c_2) \geq min(2d_1, d_2)$$

Por tanto, $d \geq min(2d_1, d_2)$.

Si $x \in C_1$ tiene wtd_1 y $y \in C_2$ tiene wtd_2, entonces $(x, x) \in C$ tiene $2d_1$ y $(0, y) \in C$ tiene wtd_2 así que $d \leq min(2d_1, d_2)$. Por tanto, $d = min(2d_1, d_2)$.

1.0.5. Dimensión

Proposición 1.0.9. *Para $m > 0$, el código de Reed-Muller $\mathscr{RM}(1, m)$ es un código lineal binario $[2^m, m+1, 2^{m+1}]$ en donde toda palabra código excepto **0** y **1** tiene peso 2^{m-1}.*

Prueba:

$\mathscr{RM}(1,1)$ es claramente un código $[2, 2, 1]$ por su definición. Haciendo prueba por inducción, suponer que $\mathscr{RM}(1, m-1)$ es un código $[2^{m-1}, m, 2^{m-2}$. Por la hipótesis inductiva, $\mathscr{RM}(1, m)$ es construido usando la construcción $(u, u+v)$ con $C_1 = \mathscr{RM}(1, m-1)$ y $C_2 = \{\mathbf{0}, \mathbf{1}\}$. Entonces, $\mathscr{RM}(1, m)$ es un código $[2(2^{m-1}), m+1, min(2(2^{m-2}), 2^{m-1})]$ o lo mismo $[2^m, m+1, 2^{m-1}]$. Como los códigos en $\mathscr{RM}(1, m)$ son ya sea de la forma (u, u) o $(u, u+1)$ con $u \in \mathscr{RM}(1, m)$, si la palabra es de la forma (u, u) entonces $\mathbf{u}$ no puede ser $\mathbf{0}$ o $\mathbf{1}$ en $\mathscr{RM}(1, m-1)$ ya que sino la palabra sería $\mathbf{0}$ o $\mathbf{1}$ en $\mathscr{RM}(1, m)$. Luego, por hipótesis de inducción, $\mathbf{u}$ tiene peso 2^{m-2} y también (u, u) tiene peso $2(2^{m-2}) = 2^{m-1}$.

Asumir ahora que la palabra es de la forma $(u, u + \mathbf{1})$.

Si $u = \mathbf{0}$, entonces $u + \mathbf{1}$ es $\mathbf{1}$ y $wt(u, u + \mathbf{1}) = 2^{m-1}$.

Si $u = \mathbf{1}$, entonces $u + \mathbf{1}$ es $\mathbf{0}$ y $wt(u, u + 1) = 2^{m-1}$.

Para todo $u \in \mathscr{RM}(1, m-1)$ restante, $wt(u) = 2^{m-2}$, esto es, la mitad de las coordinadas de u son 1. Entonces, en $u + \mathbf{1}$ también se tiene la mitad de las coordinadas igual a 1, por tanto en $wt(u, u + \mathbf{1})$ también tenemos la mitad de la coordenadas siendo 1, por tanto $wt(u, u + \mathbf{1}) = 2(2^{m-2}) = 2^{m-1}$.

Proposición 1.0.10. *Si $G_{r,m}$ es una matriz generadora de $\mathscr{RM}(r, m)$ entonces una matriz generadora para $\mathscr{RM}(r+1, m+1)$ es dada por:*

$$G_{r+1,m+1} = \begin{pmatrix} G_{r+1,m} & G_{r+1,m} \\ 0 & G_{r,m} \end{pmatrix}$$

La dimensión de $\mathscr{RM}(r, m)$ es:

$$1 + \binom{m}{1} + \binom{m}{2} + ... + \binom{m}{r}$$

El peso mínimo es igual a la distancia mínima de $\mathscr{RM}(r, m)$ igual a 2^{m-r}.

Proposición 1.0.11. *El conjunto de todos los posibles productos externos de hasta m de v_i forma una base para $\mathbb{F}_2^n$.*

Prueba:

Existen $\sum_{i=0}^{m} \binom{m}{s} = 2^m = n$ vectores que satisfacen esa condición. $\mathbb{F}_2^n$ tiene dimensión n por lo que basta verificar que los n vectores son generados, o también, que $\mathscr{RM}(m, m) = \mathbb{F}_2^n$.

Sea x un vector binario de longitud m, un elemento de X. Sea $(x)_i$ el i-ésimo elemento de x. Definir

$$\begin{cases} v_i, & \text{si } (x)_i = 0 \\ v_0 + v_i, & \text{si } (x)_i = 1 \end{cases}$$

donde $1 \leq i \leq m$.

Entonces $\mathbb{I}_x = y_1 \wedge ... \wedge y_m$. Con la expansión mediante la propiedad distributiva del producto externo nos da $\mathbb{I}_x \in \mathscr{RM}(m, m)$. Entonces ya que los vectores $\{\mathbb{I}_x | \mathbf{x} \in X\}$ generan $\mathbb{F}_2^n$ tenemos que $\mathscr{RM}(m, n) = \mathbb{F}_2^n$.

Proposición 1.0.12. *El código $\mathscr{RM}(r,m)$ tiene dimensión*

$$\sum_{i=0}^{r} \binom{m}{s}$$

Prueba:
Con la proposición anterior, todos los productos externos deben de ser linealmente independientes, así que la dimensión de $\mathscr{RM}(r,m)$ debe de ser la cantidad de estos vectores.

Definición 1.0.14 (Código Dual)**.** El **código dual** $C^{\perp}$ de un código lineal $C \subseteq \mathscr{F}_q^n$ de dimensión s se define como:

$$C^{\perp} = \{\mathbf{x} \in \mathscr{F}_q^n | < \mathbf{x}, \mathbf{c} >= 0, \forall \mathbf{c} \in C\}$$

Definición 1.0.15 (Código Autodual)**.** Si $C = C^{\perp}$, entonces C es un código autodual.

Definición 1.0.16 (Matriz de control)**.** Sea $C[n,k]$ un código lineal sobre un campo $\mathscr{F}$ y con matriz generatriz G, esto es, $C = \{mG | m \in \mathscr{F}\}$. Sea H una matriz tal que:

$$C = \{\mathbf{x} \in \mathscr{F}^n | H\mathbf{x}^T = 0\}$$

entonces, **H** se le llama matriz de control de C.

1.0.6. Peso de una palabra

Proposición 1.0.13. *$\mathscr{RM}(m-1,m)$ consiste en todas las palabras binarias de longitud 2^m de peso par. Por tanto, si $r < m$, $\mathscr{RM}(r,m)$ solo contiene palabras código de peso par.*

Prueba:

$\mathscr{RM}(0,1) = \{00, 11\}$ claramente consiste en todas las palabras binarias de longitud 2. Suponer que esto se cumple para $\mathscr{RM}(m-2,m-1)$. Una palabra código de $\mathscr{RM}(m-1,m)$ tiene la forma $(u, u+v) = (u,u) + (0,v)$ con $u \in \mathscr{RM}(m-1,m-1)$ y $v \in \mathscr{RM}(m-2,m-1)$.

Ahora, (u,u) tiene peso par para cualquier u, y $(0,v)$ tiene peso para por la hipótesis de inducción. Entonces:

$$wt((u,u+v)) = wt((u,u)) + wt((0,v)) - 2(u,u)\cdot(0,v)$$

El cual es también par. Entonces $\mathscr{RM}(m-1,m)$ contiene solo vectores pares. ya que $dim(\mathscr{RM}(m-1,m)) = 2^m - 1$, contiene a todos los vectores con peso par. Por último, si $r < m$, entonces $\mathscr{RM}(r,m) \subseteq \mathscr{RM}(m-1,m)$

Proposición 1.0.14. *Toda palabra código de un código lineal binario auto-ortogonal tiene peso par.*

Prueba

Tenemos que $C \subseteq C^{\perp}$ por hipótesis. Para una palabra código $w = (a_1, a_2, ..., a_n)$, $w \cdot w = 0$. Pero $w \cdot w = a_1^2 + a_2^2 + ... + a_n^2$ lo cual es igual al peso de w, $wt(w)$. Entonces $w \cdot w = 0$ en $\mathbb{F}_2$ sii w tiene peso par.

Proposición 1.0.15. *$\mathscr{RM}(m-r-1,m)$ y $\mathscr{RM}(r,m)$ son códigos duales.*

Prueba:

Para $m = 2$, ambos códigos existen solo si $r = 0$ o 1.

Tenemos que $\mathscr{RM}(0,2) = \{0000, 1111\}$ y $\mathscr{RM}(1,2) = \{0000, 0101, 1010, 1111, 0011, 0110, 1001, 1100\}$.

Ya que todos los vectores en $\mathscr{RM}(1,2)$ tiene peso par, entonces todo vector de $\mathscr{RM}(0,2)$ es ortogonal a todo vector de $\mathscr{RM}(1,2)$. (1)

1.0.7. Código Dual de RM(r,m)

Suponer que (1) es cierto para $m-1$. Dadas las matrices generatrices de ambos códigos:

$$G_{r,m} = \begin{pmatrix} G_{r,m-1} & G_{r,m-1} \\ 0 & G_{r-1,m-1} \end{pmatrix}$$

$$G_{m-r-1,m} = \begin{pmatrix} G_{m-r-1,m-1} & G_{m-r-1,m-1} \\ 0 & G_{m-r-2,m-1} \end{pmatrix}$$

Las filas de la forma (a,a) son ortogonales a las de la forma (b,b).

Las filas de la forma (a,a) son ortogonales a las de la forma $(0,d)$ por la hipótesis inductiva. Por la misma razón, las filas de la forma $(0,c)$ son ortogonales a las de forma (b,b).

Ya que $\mathscr{RM}(m-r-2,m-1) \subset \mathscr{RM}(m-r-1,m-1)$, las filas de la forma $(0,c)$ son ortogonales a las de forma $(0,d)$.

Por tanto, $\mathscr{RM}(m-r-1,m) \subset \mathscr{RM}(r,m)^{\perp}$. Ahora:

$$\begin{aligned} dim(\mathscr{RM}(r,m)^{\perp}) &= 2^m - [1 + \binom{m}{1} + ... + \binom{m}{r}] \\ &= \binom{m}{r+1} + \binom{m}{r+2} + ... + \binom{m}{m} \\ &= \binom{m}{m-r-1} + \binom{m}{m-r-2} + ... + 1 \\ &= dim(\mathscr{RM}(m-r-1,m)) \end{aligned}$$

Por tanto, $\mathscr{RM}(m-r-1,m) = \mathscr{RM}(r,m)^{\perp}$.

1.0.8. Construcción mediante funciones Booleanas

En esta sección se detallará una construcción mediante funciones Booleanas la cual es una particularización de la construcción general presentada al inicio.

Proposición 1.0.16. *Dado un orden en $\mathbb{F}_2^{2^m}$, se tiene que el conjunto de los polinomios Booleanos P_m es isomorfo, como $\mathbb{F}_2$-álgebra, a $\mathbb{F}_2^{2^m}$ a través de la aplicación:*

$$\phi_m = e\bar{v}_{\mathbb{F}_2^m} : P_m \to \mathbb{F}_2^{2^m}$$
$$F \to e\bar{v}_{\mathbb{F}_2^m}(F) = \boldsymbol{F} = F_1...F_{2^m}$$

De esto, cada polinomio Booleano tiene asociada una, y solo una, palabra binaria $\mathbf{x} \in \mathbb{F}_2^{x^m}$.

Definición 1.0.17. Sean m y r dos números naturales, donde $0 \leq r \leq m$. Se define por código de Reed-Muller binario de longitud 2^m y orden r, lo cual denotaremos por simplicidad como $\mathscr{RM}(r,m)$, al conjunto de todas las palabras binarias de longitud 2^m asociadas a todos los polinomios Booleanos de m indeterminadas y grado menor o igual que r.

Proposición 1.0.17. *El código de Reed-Muller binario $\mathscr{RM}(r,m)$ es un código lineal de longitud 2^m sobre $\mathbb{F}_2$, y su dimensión como $\mathbb{F}_2$-subespacio vectorial viene dada por:*

$$dim(\mathscr{RM}(r,m)) = \sum_{k=0}^{r} \binom{m}{k}$$

Además, la matriz

$$G(r,m) = \begin{pmatrix} \phi_m(1) \\ \phi_m(x_1) \\ . \\ . \\ . \\ \phi_m(x_m) \\ \phi_m(x_1x_2) \\ . \\ . \\ . \\ \phi_m(x_{m-r+1}...x_m) \end{pmatrix}$$

donde ϕ_m (aplicación que se utilizó en la proposición anterior) es una matriz generadora de $\mathscr{RM}(r,m)$.

Prueba:

Tenemos que $\mathscr{RM}(r,m)$ es un código lineal por las propiedades establecidas de la construcción general. Luego, a fin de obtener la fórmula de la dimensión, por la proposición 1,0,7 se debe de hallar el número de soluciones de las ecuaciones $i_1 + ... + i_m = t$ con $t \in \{0,1,...,r\}$, tales que $0 \leq i_1, ..., i_m \leq q-1$. Sin embargo, fijado t, este valor coincide con el número de subconjuntos de t elementos distintos elegidos de un conjunto de cardinal m sin importar el orden. Pero esta es la definición de las combinaciones de m elementos tomadas de t en t. El resultado a partir de aquí vuelve a ser inmediato por el Principio Aditivo. La fórmula se obtiene por la proposición 1,0,6.

1.0.9. Construcción recursiva de Plotkin

Esta construcción recursiva es válida solo cuando se tiene fijado un orden para los elementos de $\mathbb{F}_2^m$. Dada la expansión binaria $i_0 + 2i_1 + 2^2i_2 + ... + 2^{m-2}i_{m-2} + 2^{m-1}i_{m-1}$ con $i_0, i_1, ..., i_{m-2}, i_{m-1} \in \{0,1\}$, de un cierto entero i, asociamos a $i+1$ el elemento $(i_0, i_1, ..., i_{m-2}, i_{m-1}) \in \mathbb{F}_2^m$, esto es:

$$\begin{aligned} 1 \to &\quad (0,0,0,...,0,0), \\ 2 \to &\quad (1,0,0,...,0,0), \\ 3 \to &\quad (0,1,0,...,0,0), \\ 4 \to &\quad (1,1,0,...,0,0), \\ 5 \to &\quad (0,0,1,...,0,0), \\ .. &\quad . \\ .. &\quad . \\ .. &\quad . \\ 2^m-2 \to &\quad (1,0,1,...,1,1), \\ 2^m-1 \to &\quad (0,1,1,...,1,1), \\ 2^m \to &\quad (1,1,1,...,1,1), \end{aligned}$$

Este se denomina orden canónico de $\mathbb{F}_2^m$. El objetivo es obtener expresiones dependientes de r y m tanto para la distancia mínima de $\mathscr{RM}(r,m)$ como para una matriz generadora de $\mathscr{RM}(r,m)$.

Tener en cuenta los siguientes resultados:

Proposición 1.0.18. *La palabra característica asociada al polinomio Booleano x_i, como elemento de P_m, es aquella que tiene en su k-ésima posición un 1 cuando la expansión binaria de k tiene un 1 en la i-ésima posición, para $k \in \{1, ..., 2^m\}$. En términos de tablas de verdad, esto es:*

$$\begin{aligned} \phi(x_1) = &\quad 01010101..,0101, \\ \phi(x_2) = &\quad 00110011..,0011, \\ .. &\quad . \\ .. &\quad . \\ .. &\quad . \\ \phi(x_m) = &\quad 000..,0011..,11 \end{aligned}$$

(2^{m-1} veces para cada una de las dos partes del último término)

Proposición 1.0.19. *Sea $F(x_1, ..., x_{m-1}$ un polinomio Booleano de $m-1$ indeterminadas que tiene como palabra binaria asociada a $\boldsymbol{x} = x_1...x_{2^{m-1}}$. Entonces, la palabra binaria asociada a F como polinomio Booleano de P_m es $\boldsymbol{x} * \boldsymbol{x} = x_1...x_{2^{m-1}}x_1...x_{2^{m-1}}$.*

Proposición 1.0.20. *Dados dos números naturales r y m tales que $0 < r < m$, se cumple que*

$$\mathbb{RM}(r, m) = \mathscr{RM}(r, m-1) \circledast \mathscr{RM}(r-1, m-1)$$

Es decir, el código de Reed-Muller $\mathscr{RM}(r, m)$ es la construcción de Plotkin de los códigos de Reed-Muller $\mathscr{RM}(r, m-1)$ y $\mathscr{RM}(r-1, m-1)$.

Definición 1.0.18. Dados r y m dos números naturales tales que $0 < r < m$, se define de manera recursiva el código de Reed-Muller binario de longitud 2^m y orden r $\mathscr{RM}(r, m)$ a través de kas reglas siguientes:

$$1)\, \mathscr{RM}(0, n) = \{0 \cdots 0, 1 \cdots 1\} \forall n \in \{1, ..., m\} (2^n \textit{ para } \cdots)$$

$$2)\, \mathscr{RM}(n, n) = \mathbb{F}_2^{2^n}, \forall n \in \{1, ..., m\}$$

$$3)\, \mathscr{RM}(s, n) = \mathscr{RM}(s, n-1) \circledast \mathscr{RM}(s-1, n-1), \forall s \in \{1, ..., r\}, \forall n \in \{s, ..., m\}$$

Proposición 1.0.21. *Dados r y m naturales tales que $0 \leq r \leq m$, se tiene que la distancia mínima de $\mathscr{RM}(r, m)$ es 2^{m-r}.*

Prueba:

Con la definición recursiva de arriba, tenemos lo siguiente. Hay que tener en cuenta que, pese a que se tenga fijado el orden canónico de $\mathbb{F}_2^m$, podemos calcular la distancia mínima de nuestro código de Reed-Muller a partir de la construcción de Plotkin, puesto que cambiar el orden en $\mathbb{F}_2^m$ solo hace que se obtenga un código equivalente, que sigue siendo un código de Reed-Muller con los mismos parámetros.

Por inducción sobre m, para $m = 1$ se tienen dos posibles códigos de Reed-Muller binarios. Como $RM(0, 1) = \{00, 11\}$ y $\mathscr{RM}(1, 1) = \{00, 11, 01, 10\}$, es evidente que $d(\mathscr{RM}(0, 1)) = 2 = 2^{1-0}$ y $d(\mathscr{RM}(1, 1)) = 1 = 2^{1-1}$. Supongamos cierto el resultado para $m-1$ y probémoslo para m. Por hipótesis inductiva, se tiene que

$$d(\mathscr{RM}(r, m)) = min\{2d(\mathscr{RM}(r, m-1)), d(\mathscr{RM}(r-1, m-1))\}$$

$$= min\{22^{m-r-1}, 2^{m-r}\} = 2^{m-r}$$

Código Reed-Muller de la NASA

El código de Reed-Muller que llevaban equipado las naves espaciales de la NASA en la década de los años 50 se presenta a continuación:

Describir el código de Reed-Muller $\mathscr{RM}(1,5)$: calcular una matriz generadora, su dimensión y su distancia mínima.

Solución:
Utilizando el programa proveído por esta investigación, se obtiene la matriz siguiente:

```
>> genmat(1,5)
ans =

 Columns 1 through 19:

   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
   0   0   1   1   0   0   1   1   0   0   1   1   0   0   1   1   0   0   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   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   0   0   0   1   1   1

 Columns 20 through 32:

   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
   1   0   0   1   1   0   0   1   1   0   0   1   1
   0   1   1   1   1   0   0   0   0   1   1   1   1
   0   0   0   0   0   1   1   1   1   1   1   1   1
   1   1   1   1   1   1   1   1   1   1   1   1   1

>>
```

Figura 1.1: Matriz generadora

Esta matriz generadora tiene rango máximo 6 (coincide con el número de filas), la dimensión del código $\mathscr{RM}(1,5)$ es 6 precisamente. La distancia mínima se obtiene a partir de la fórmula dada en la proposición 1,0,21, por lo que la distancia mínima es 16.

Proposición 1.0.22. *Las palabras características de todo monomio Booleano de grado exactamente $t \geq 0$ tienen peso 2^{m-t}. De esto se deduce que todas las palabras de $\mathscr{RM}(r,m)$ con $r < m$ tienen peso par. Además, $\mathscr{RM}(m-1,m)$ se corresponde con el conjunto de todas las palabras binarias de $\mathbb{F}_2^{2^m}$ de peso par.*

Prueba:

Sea $F \in \mathbb{F}_2[x_1,...,x_m]$ un monomio de grado t, y vamos a suponer que se tiene fijado un orden $\mathbb{F}_2^m = \{\mathbf{v}_1,...,\mathbf{v}_{2^m}\}$. Por hipótesis, como F es de grado t, se tiene que $F = x_{i_1}\cdots x_{i_t}$, con $i_1,\cdots,i_t \in \{1,...,m\}$. Queremos probar que $\mathbf{F}$ tiene peso 2^{m-t}. Como estamos en el caso binario, sabemos que $\omega(\mathbf{F}) = \sum_{i=1}^{2^m} F_i$, pues $F_i \in \{0,1\} \forall i \in \{1,...,2^m\}$. Además, dado el vector $\mathbf{v}_j$ que ocupa la posición j-ésima en el orden establecido, tenemos la siguiente caracterización para las letras de $\mathbb{F}$:

$$f_j = 1 \iff v_{ji_1} = \cdots = v_{ji_t} = 1$$

Si se prueba que hay 2^{m-t} letras F_j que verifican esta equivalencia, se probará la primera parte de la proposición. Esto es inmediato, pues fijadas estas t coordenadas $\{i_1,...,i_t\}$ en cada vector $\mathbf{v}_j \in \mathbb{F}_2^m$, quedan libres las $m-t$ restantes, que pueden tomar 2 valores distintos. A partir de aquí, con un sencillo argumento combinatorio, se concluye el resultado.

De acuerdo a la proposición 1,0,17 que dice que una matriz generadora de este código tiene por filas las palabras características de algunos monomios Booleanos, que acabamos de ver que tienen en particular peso par cuando su grado es menor estrictamente que m, se sigue de lo mostrado en la sección de la construcción mediante funciones Booleanas que toda palabra de $\mathscr{RM}(r,m)$ para $r < m$ tiene también peso par. Además, como $\mathscr{RM}(m-1,m)$ está formado por la mitad de las palabras de $\mathbb{F}_2^{2^m}$, pues

$$dim(\mathscr{RM}(m-1,m)) = \sum_{k=0}^{m-1} \binom{m}{k} = 2^m - 1$$

por lo que acabamos de probar, necesariamente $\mathscr{RM}(m-1,m)$ tiene que ser el conjunto de todas las palabras de $\mathbb{F}_2^{2^m}$ con peso par.

1.0.10. Codificación/Decodificación

Suponer que el proceso de codificación para un código Reed-Muller de r-ésimo orden es el siguiente.

Sean $\sum_{i=0}^{r}\binom{m}{i}$ los bits de datos dados como

$$\begin{array}{l} d_0, \\ d_1, \quad d_2, \quad \ldots, \quad d_m, \\ d_{1,2}, d_{1,3}, \ldots, d_{1,m}, \quad d_{2,3}, d_{2,4}, \ldots, d_{2,m}, \quad \ldots, \quad d_{m-1,m} \\ \vdots \qquad \ddots \\ d_{1,2,\ldots,r}, \quad \ldots, \quad d_{m-r+1,m-r+2,\ldots,m} \end{array}$$

donde la fila i-ésima de arriba enumera $\binom{m}{i}$ bits de datos.

Definiendo el polinomio

$$\begin{array}{rcl} d(x_1,x_2,\ldots,x_m) = & & d_0 \\ & \oplus & d_1x_1 \oplus d_2x_2 \oplus \cdots \oplus d_mx_m \\ & \oplus & d_{1,2}x_1x_2 \oplus d_{1,3}x_1x_3 \oplus \cdots d_{m-1,m}x_{m-1}x_m \\ & \oplus & \cdots \\ & \oplus & d_{1,2,\ldots,r}x_1x_2\cdots x_r \oplus \cdots d_{m-r+1,m-r+2,\ldots,m}x_{m-r+1}x_{m-r+2}\cdots x_m \end{array}$$

de grado r en m variables binarias $x_1, x_2, \ldots, x_m$. Notar que la fila i-ésima tiene todos los términos $\binom{m}{i}$ del grado i.

Registrando todos los 2^m valores de esta función booleana como la palabra código:

$$\mathbf{d} = (d(0,0,\ldots,0), d(0,0,\ldots,0,1), d(0,0,\ldots,1,0), \ldots, d(1,1,\ldots,1))$$

Se tiene la tabla de verdad para esta función booleana.

El peso de Hamming de $d(x_1,x_2,\ldots,x_m)$ es el número de entradas en $\mathbf{d}$ que tienen valor 1. En lo que sigue, necesitaremos la siguiente propiedad: El peso de Hamming de cualquier polinomio es impar si y solo si el polinomio es de grado m, es decir, $d_{1,2,\ldots,m} = 1$ y por lo tanto el término $x_1x_2\cdots x_m$ está incluido en el polinomio.

Los códigos $RM(r,m)$ se pueden decodificar utilizando decodificación lógica mayoritaria. La idea básica de la decodificación lógica mayoritaria es construir varias sumas de verificación (checksum) para cada elemento de palabra de código recibido. Dado que cada una de las diferentes sumas de verificación deben tener el mismo valor (es decir, el valor del peso del elemento de la palabra del mensaje), podemos usar una decodificación lógica mayoritaria para descifrar el valor del elemento de la palabra del mensaje. Una vez que se decodifica cada orden del polinomio, la palabra recibida se modifica en consecuencia eliminando las palabras de código correspondientes ponderadas por las contribuciones del mensaje decodificado, hasta la etapa actual. Entonces, para un código RM de orden r, tenemos que decodificar iterativamente $r+1$, veces antes de llegar a la palabra de código final recibida. Además, los valores de los bits del mensaje se calculan mediante este esquema; finalmente podemos calcular la palabra código multiplicando la palabra del mensaje (recién decodificada) con la matriz generadora del código.

Dada la palabra de código (o tabla de verdad) $\mathbf{d}$, a fin de determinar el valor del bit de datos $d_{1,2,\ldots,r}$ que es el coeficiente del término $x_1x_2\cdots x_r$ en $d(x_1,x_2,\ldots,x_m)$ consideremos el polinomio:

$$g(x_1,x_2,\ldots,x_m) = x_{r+1}x_{r+2}\cdots x_m d(x_1,x_2,\ldots,x_m)$$

donde supongamos que para todo $i, 1 \le i \le m$, todas las apariciones de x_i^2 en el lado derecho de la ecuación anterior han sido reemplazadas por x_i ya que x_i toma solo los valores de 0 y 1. Dado que $d(x_1,\ldots,x_m)$ es de grado r

de la forma que se muestra arriba, concluimos que $x_{r+1}x_{r+2}\cdots x_m d(x_1, x_2, \ldots, x_m)$ incluye el término $x_1x_2\cdots x_m$ si y solo si $d_{1,2,\ldots,r} = 1$, es decir, $d(x_1, \ldots, x_m)$ contiene el término $x_1x_2\cdots x_r$. De ello se deduce que la tabla de verdad de $x_{r+1}x_{r+2}\cdots x_m d(x_1, x_2, \ldots, x_m)$ contiene un número impar de 1 entradas si y solo si $d_{1,2,\ldots,r} = 1$. Ahora, la tabla de verdad de $x_{r+1}x_{r+2}\cdots x_m d(x_1, x_2, \ldots, x_m)$ tiene 0s siempre que al menos uno de $x_{r+1}, x_{r+2}, \ldots, x_m$ tiene valor 0, y las entradas distintas de cero, si las hay, deben estar en las posiciones 2^r donde $x_{r+1} = x_{r+2} = \cdots = x_m = 1$ son fijos y solo $x_1, x_2, \ldots, x_r$ varían. Además, las entradas en estas posiciones 2^r deben ser las mismas que las entradas en $\mathbf{d}$, lo cual se sabe ya. Para verificar si hay un número impar de entradas de 1 en $\mathbf{d}$ en estas posiciones de 2^r podemos calcular la paridad (también conocida como suma XOR) de estos 2^r bits en $\mathbf{d}$. Formalmente, podemos escribir

$$\begin{aligned} d_{1,2,\ldots,r} = \quad & d(0,0,\ldots,0,0,1,1,\ldots,1) \\ & \oplus d(0,0,\ldots,0,1,1,1,\ldots,1) \\ & \oplus d(0,0,\ldots,1,0,1,1,\ldots,1) \\ & \oplus d(0,0,\ldots,1,1,1,1,\ldots,1) \\ & \oplus \cdots \\ & \oplus d(1,1,\ldots,1,1,1,1,\ldots,1). \end{aligned}$$

Ahora nótese que se pueden establecer sumas de paridad similares para los coeficientes de todos los demás términos del grado r. Se debe de crear $g(x_1, x_2, \ldots, x_m)$ multiplicando $d(x_1, x_2, \ldots, x_m)$ por todas las variables que faltan y luego calcular la paridad de los 2^r bits en $\mathbf{d}$.

Luego, al haber determinado los coeficientes de los términos de grado r, es necesario modificar $\mathbf{d}$ para que sea la tabla de verdad no de los $d(x_1, x_2, \ldots, x_m)$ con los que comenzamos pero en lugar de un polinomio $\hat{d}(x_1, x_2, \ldots, x_m)$ de grado $r-1$. Este nuevo polinomio es solo los términos de grado $r-1$ o menos en $d(x_1, x_2, \ldots, x_m)$, y la modificación es simplemente restar de la tabla de verdad del polinomio

$$d_{1,2,\ldots,r}x_1x_2\cdots x_r \oplus \cdots d_{m-r+1,m-r+2,\ldots,m}x_{m-r+1}x_{m-r+2}\cdots x_m$$

de $\mathbf{d}$ (lo mismo que usar el XOR en la tabla de verdad en $\mathbf{d}$). Si esto no se hace, entonces multiplicar por las variables faltantes $x_r, x_{r+1}, \ldots x_m$ al intentar calcular $d_{1,2,\ldots,r-1}$ no funcionará porque el término $d_{1,2,\ldots,r}x_1x_2\cdots x_r$ creará $x_1x_2\cdots x_m$ términos adicionales que interfieren en el cálculo de $d_{1,2,\ldots,r-1}$.

Por lo tanto, la decodificación funciona en etapas: primero determine los coeficientes de los términos grado- r y réstelos, luego determine los coeficientes de los términos grado - $(r-1)$ y réstelos, y continúe hasta que todos se han calculado los plazos.

Todo lo anterior funciona si la palabra de código transmitida (o grabada) $\mathbf{d}$ se recibe (o lee) sin errores. Pero cuando algunos de los bits en $\mathbf{d}$ son incorrectos, no podemos estar seguros de que el cálculo de paridad

$$\begin{aligned} d_{1,2,\ldots,r} = \quad & d(0,0,\ldots,0,0,1,1,\ldots,1) \\ & \oplus d(0,0,\ldots,0,1,1,1,\ldots,1) \\ & \oplus d(0,0,\ldots,1,0,1,1,\ldots,1) \\ & \oplus d(0,0,\ldots,1,1,1,1,\ldots,1) \\ & \oplus \cdots \\ & \oplus d(1,1,\ldots,1,1,1,1,\ldots,1). \end{aligned}$$

dé el valor correcto de $d_{1,2,\ldots,r}$. Después de todo, un número impar de bits que participan en el cálculo de paridad puede ser incorrecto. Por lo tanto, el cálculo de paridad anterior debe tratarse como una estimación del valor de $d_{1,2,\ldots,r}$. Afortunadamente, podemos obtener $2^{m-r}-1$ otras estimaciones de $d_{1,2,\ldots,r}$ de otros bits en $\mathbf{d}$. Por ejemplo, el polinomio

$$\begin{aligned} g(x_1, x_2, \ldots, x_m) &= \bar{x}_{r+1}x_{r+2}\cdots x_m d(x_1, x_2, \ldots, x_m) \\ &= (x_{r+1} \oplus 1)x_{r+2}\cdots x_m d(x_1, x_2, \ldots, x_m) \\ &= x_{r+1}x_{r+2}\cdots x_m d(x_1, x_2, \ldots, x_m) \oplus x_{r+2}\cdots x_m d(x_1, x_2, \ldots, x_m) \end{aligned}$$

tiene grado m si y solo si $d_{1,2,\ldots,r} = 1$ pero su tabla de verdad tiene entradas distintas de cero en 2^r otras ubicaciones, a saber. donde $x_{r+1} = 0, x_{r+2} = x_{r+3} = \cdots = x_m = 1$. De esta manera, hacemos 2^{mr} cálculos de paridad diferentes para $d_{1,2,\ldots,r}$ usando todos los 2^m bits en $\mathbf{d}$, y luego obtenemos un voto mayoritario de las diferentes estimaciones para estimar el valor de $d_{1,2,\ldots,r}$. Siempre que no más de $2^{mr-1} - 1$ bits en el $\mathbf{d}$ recibido sean incorrectos, el voto de la mayoría da como resultado una estimación correcta, es decir, hasta $2^{mr-1}-$ los errores de 1 bit se pueden corregir: un voto de empate indica que se ha producido un patrón de error detectable pero incorregible.

El resto de la decodificación procede como se describió anteriormente con los cambios obvios de hacer múltiples cálculos de paridad al estimar cada bit de datos, y lo que se resta entre las etapas son los términos estimados de ese grado. Tenga en cuenta que se pueden realizar más cálculos de paridad en etapas posteriores, lo que puede garantizar una doble seguridad si las etapas anteriores dieron como resultado una decodificación correcta, pero no proporcionan protección adicional contra errores si la decodificación incorrecta ocurrió en una etapa anterior. Una pista para saber si la decodificación tuvo éxito, es tener una palabra recibida modificada con valor cero al final de la decodificación de la etapa $r + 1$ a través de la decodificación lógica mayoritaria. Esta técnica fue propuesta por Irving S. Reed, y es más general cuando se aplica a otros códigos de geometría finita.

1.0.11. Decodificación por fuerza bruta

En esta sección se presenta un análisis más avanzado sobre las posibilidades de decodificación en los códigos de Reed-Muller. Consideremos cómo la decodificación de máxima probabilidad de un código $\mathscr{R}.\mathscr{M}(1, n)$ mediante la comparación de $\mathbf{r}$ con cada una de las palabras de código 2^{n+1} se compara con un decodificador canónico de Reed-Muller.

Decodificador de comparación: Tenemos que determinar la distancia de Hamming entre $\mathbf{r}$ y cada palabra código. Una forma de hacer esto es calcular $\mathbf{r} \oplus \mathbf{c}$ (2^n operaciones XOR para cada $\mathbf{c}$) y luego contar los UNOS en la suma. Ignorando la complejidad de este recuento, observamos que se requiere un total de operaciones 2^{2n+1} XOR. Luego, cuando tenemos las distancias de 2^{n+1} calculadas, tenemos que encontrar la palabra de código $\hat{\mathbf{c}}$ que sea más cercana a $\mathbf{r}$. Por lo tanto, se necesitan más comparaciones, etc. Pero todavía no hemos terminado: tenemos que determinar los bits de información de $\hat{\mathbf{c}}$ y se necesitan más cálculos.

Decodificador canónico: Las 2^{n-1} verificaciones en cada uno de los n bits de información de "grado-1"se calculan usando un XOR cada uno para un total de $2^{n-1} \times n$ operaciones XOR. Necesitamos contar cuántas de las 2^{n-1} verificaciones son UNOS, pero eso es solo la mitad del trabajo que contar el número de UNOS en 2^n bits, y debe hacerse solo n veces en lugar de 2^{n+1} veces. Habiendo determinado los bits de información de n "grado-1", tenemos que encontrar la "palabra de códigoçorrespondiente y restarla de $\mathbf{r}$. Podría parecer que el cálculo de la palabra de código necesitaría $(n-1) \times 2^n$ XOR sumas, pero si calculamos las sumas en el orden del código Gray en lugar del orden de conteo binario natural, podemos administrarlo con solo 2^n XORs. En cualquier caso, dado que el método çomparar con todas las palabras códigorequiere el almacenamiento de todas las palabras de código de 2^{n+1}, podríamos considerar almacenar las palabras de código de 2^n y usar el "grado-1"de n bits de información como una dirección que se utilizará en una tabla de búsqueda. Finalmente, tenemos que contar el número de UNOS en lo que queda para determinar el bit de información de "grado 0".

Dicho esto, existe un método de decodificación para los códigos $RM(1, n)$ que implementa efectivamente el método çomparar con todas las palabras de código", y es bastante eficiente. Si $\mathbf{r} \in \mathbb{Z}_2^{2^n}$ es el vector recibido, cree un vector $\mathbf{x} \in \{+1, -1\}^{2^n}$ haciendo $x_i = (-1)^{r_i}$. Sea H_n la matriz $2^n \times 2^n$ Hadamard en la forma de Sylvester, es decir,

$$H_n = \begin{bmatrix} H_{n-1} & H_{n-1} & H_{n-1} & \end{bmatrix}; \qquad H_1 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}.$$

Tenga en cuenta que las filas de H_n son el "grado-1"palabras de código en el código $RM(1, n)$ traducidas del alfabeto $\{0, 1\}$ al alfabeto $\{+1, -1\}$. Entonces, $\mathbf{y} = \mathbf{x}H$ es un vector cuya k -ésima entrada tiene valor $2^n - d_k$ donde d_k es la distancia de Hamming entre $\mathbf{r}$ y k -ésima palabra de código, $0 \leq k \leq 2^n - 1$. El algoritmo de decodificación entonces es calcular $\mathbf{y}$ y determinar

$$D = \text{argmax}\,|y_k|.$$

Si la representación binaria estándar de D es

$$D = \sum_{i=1}^{n} D_i 2^{I-1},$$

Luego $(D_n, D_{n-1}, \cdots, D_1)$ son los bits de información n "grado-1"mientras que $D_0 = \ frac1 - y_D2$ es el bit de información de grado 0. Todo esto está bien y es excelente, pero el punto real es que existe un algoritmo de transformación rápida de Hadamard (muy similar al algoritmo de transformada rápida de Fourier radix-2) que reduce el esfuerzo computacional de encontrar $\mathbf{y}$ a partir de $(2^n)^2$ multiplicaciones y adiciones a $n2^n$ operaciones, y este algoritmo de transformación rápida de Hadamard hace que este decodificador sea eminentemente práctico y eficiente. Se implementó hace más de 45 años en las misiones Mariner a Marte.

1.1. Bibliografía

- AMS, American Mathematical Society. (n.d.). Error correction technologies. AMS. Retrieved April 2, 2021, from http://www.ams.org/publicoutreach/feature-column/fcarc-errors6
- http://www-math.ucdenver.edu/~wcherowi/courses/m7823/reedmuller.pdf
- https://web.mat.upc.edu/sebastia.xambo/ICT/ICT12-06-LinearCodes-ControlMatrix.pdf
- https://addi.ehu.es/bitstream/handle/10810/20121/TFG_DeArribaDeLaHeraAndoni.pdf;jsessionid=B1A0849B5D5BE3CAF3F3DBB4E2013A49?sequence=2
- https://arxiv.org/pdf/2002.03317.pdf
- https://www.researchgate.net/publication/337689159_Polynomials_in_Error_Detection_and_Correction_in_Data_Communication_System
- http://www.ams.org/publicoutreach/feature-column/fcarc-errors6
- https://sites.math.washington.edu/~morrow/336_12/papers/juan.pdf