# Tarea No. 4: Solución

2. **Representación Sistemática de Códigos Lineales:**
Considere el código generado por la matriz binaria

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

(a) Determine una forma sistemática para la matriz generadora $G$.

**Solución**

$$\begin{aligned} G &= \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} (1) \\ (2) \\ (3) \end{matrix} \\ &= \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} (1) \\ (1)+(2) \\ (3) \end{matrix} \\ &= \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{matrix} (1) \\ (2') \\ (2')+(3) \end{matrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{matrix} (1)+(3') \\ (2') \\ (3') \end{matrix} \\ &= [I|P] \end{aligned}$$

(b) Encuentre una matriz de chequeo de paridad $H$.

$$H = [-P^T|I] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(c) Cuál es la distancia mínima del código y la capacidad correctora del código?

Notamos que $H$ tiene dos columnas linealmente dependientes (2 y 7). Por lo tanto el peso mínimo del código es a lo más 2.

Existen palabras de peso 1? No, porque la matriz generadora no tiene filas de peso 1. En consecuencia el peso mínimo de código es 2 y la distancia mínima es 2.

Cuál es la capacidad correctora del código? Dado que

$$2 = d_{\min} \geq 2t + 1$$
$$\Rightarrow t \leq \frac{1}{2}.$$

Este código no puede corregir errores.

(d) Escriba el arreglo estándar del código (puede hacerlo utilizando algún software como MATLAB, para ayudarse).

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0000000 | 0010110 | 0100001 | 0110111 | 1001010 | 1011100 | 1101011 | 1111101 |
| 2 | 0000001 | 0010111 | 0100000 | 0110110 | 1001011 | 1011101 | 1101010 | 1111100 |
| 3 | 0000010 | 0010100 | 0100011 | 0110101 | 1001000 | 1011110 | 1101001 | 1111111 |
| 4 | 0000100 | 0010010 | 0100101 | 0110011 | 1001110 | 1011000 | 1101111 | 1111001 |
| 5 | 0001000 | 0011110 | 0101001 | 0111111 | 1000010 | 1010100 | 1100011 | 1110101 |
| 6 | 0010000 | 0000110 | 0110001 | 0100111 | 1011010 | 1001100 | 1111011 | 1101101 |
| 7 | 1000000 | 1010110 | 1100001 | 1110111 | 0001010 | 0011100 | 0101011 | 0111101 |
| 8 | 0000011 | 0010101 | 0100010 | 0110100 | 1001001 | 1011111 | 1101000 | 1111110 |
| 9 | 0000101 | 0010011 | 0100100 | 0110010 | 1001111 | 1011001 | 1101110 | 1111000 |
| 10 | 0001001 | 0011111 | 0101000 | 0111110 | 1000011 | 1010101 | 1100010 | 1110100 |
| 11 | 0010001 | 0000111 | 0110000 | 0100110 | 1011011 | 1001101 | 1111010 | 1101100 |
| 12 | 1000001 | 1010111 | 1100000 | 1110110 | 0001011 | 0011101 | 0101010 | 0111100 |
| 13 | 1000100 | 1010010 | 1100101 | 1110011 | 0001110 | 0011000 | 0101111 | 0111001 |
| 14 | 1010000 | 1000110 | 1110001 | 1100111 | 0011010 | 0001100 | 0111011 | 0101101 |
| 15 | 0000111 | 0010001 | 0100110 | 0110000 | 1001101 | 1011011 | 1101100 | 1111010 |
| 16 | 0111000 | 0101110 | 0011001 | 0001111 | 1110010 | 1100100 | 1010011 | 1000101 |

3. **Dual de un Código:**
Considere el código dual $\mathcal{C}^\perp$ de un código lineal $\mathcal{C}(n,k,d)$. Sea $H$ y $G$ las matrices de chequeo de paridad y generadora del código $\mathcal{C}$, respectivamente. Demuestre:

(a) $G$ es la matriz de chequeo de paridad de $\mathcal{C}^\perp$.

Por definición

$$\mathcal{C}^\perp = \{\mathbf{v} \in \mathbb{F}_q^n : \ \mathbf{c}\mathbf{v}^T = \sum_{i=1}^{n} c_i v_i = 0 \ \forall \mathbf{c} \in \mathcal{C}\}.$$

Consideremos las filas de la matriz generadora de $\mathcal{C}$:

$$G = \begin{bmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_k \end{bmatrix}$$

Entonces, $\mathbf{v} \in \mathcal{C}^\perp \Leftrightarrow \mathbf{g}_i \mathbf{v} c^T = 0 \ \forall \ i = 1 \dots k$. En otras palabras

$$\mathcal{C}^\perp = \{\mathbf{v} \in \mathbb{F}_q^n : \ \mathbf{g}_i \mathbf{v}^T = 0 \ i = 1 \dots k\}.$$

Esto no es más que otra forma de escribir

$$\mathcal{C}^\perp = \{\mathbf{v} \in \mathbb{F}_q^n : \ G\mathbf{v}^T = \mathbf{0} \ i = 1 \dots k\}.$$

Por lo tanto, $G$ es la matriz de paridad de $\mathcal{C}^\perp$.

(b) $H$ es la matriz generadora de $\mathcal{C}^\perp$.

Sea

$$H = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{n-k} \end{bmatrix}$$

donde $\mathbf{h}_1, \dots, \mathbf{h}_{n-k}$ son las filas de $H$.

Por definición de $\mathcal{C}$ tenemos que

$$\forall \mathbf{c} \in \mathcal{C} \ \mathbf{h}_i \mathbf{c}^T = 0 \ \ \forall i = 1, \dots, n-k.$$

Ello implica que $\mathbf{h}_1, \dots, \mathbf{h}_{n-k} \in \mathcal{C}^\perp$.

Como $H$ es de rango completo, las columnas de la matriz son linealmente independientes, y dado que $\dim(\mathcal{C}^\perp) = n-k$, forman una base de $\mathcal{C}^\perp$ y pueden escribirse en forma matricial como $G^\perp = H$.

(c) $\dim(\mathcal{C}^\perp) = n - k$.

Usando la parte (a):

$$\dim(\mathcal{C}^\perp) = n - \dim(\mathcal{C}) = n - k.$$

(d) $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.

De (a) y (b) tenemos que la matriz de chequeo de paridad de $(\mathcal{C}^\perp)^\perp$ es la generadora de $\mathcal{C}^\perp$, la que a su vez es la matriz de chequeo de $\mathcal{C}$.

4. **Función Enumeradora de Peso:**
Considere el código de Golay $\mathcal{C}(23, 12, 7)$, y defina el código binario de Golay extendido $\mathcal{C}^*(24, 12, 8)$. Sean $A_i$ y $A_i^*$ el número de palabras de código de peso $i$ en $\mathcal{C}$ y $\mathcal{C}^*$, respectivamente.

(a) Encuentre una recurrencia para determinar $A_i$.

**Solución**

El código de Golay $\mathcal{C}(23, 12, 7)$ es un código lineal de distancia mínima 7, lo que implica que puede corregir un máximo de 3 errores. Además, es perfecto lo que significa que las esferas de radio 3 alrededor de cada una de las $2^{12}$ palabras cubren por completo el espacio $\mathbb{F}_2^{23}$.

El procedimiento que vamos a seguir para determinar la recurrencia consiste en contar todas las secuencias $\mathbf{x} \in \mathbb{F}_2^{23}$ de un determinado peso $i$. Estas secuencias pertenecen a una de las siguientes cuatro categorías:

1. La secuencia es una palabra de código, esto es, $d(\mathbf{x}, C) = 0$,
2. La secuencia difiere en un bit de una palabra de código, esto es, $d(\mathbf{x}, C) = 1$,
3. La secuencia difiere en dos bit de una palabra de código, esto es, $d(\mathbf{x}, C) = 2$,
4. La secuencia difiere en tres bit de una palabra de código, esto es, $d(\mathbf{x}, C) = 3$.

Si sumamos el número total de estas secuencias debe coincidir con el total de secuencias de peso $i$ en $\mathbb{F}_2^{23}$, que es $\binom{23}{i}$.

En lo que sigue $A_i$ denota el número de palabras de código de peso $i$.

- El número de secuencias $\mathbf{x} \in \mathbb{F}_2^{23}$ de peso $i$ tal que $d(\mathbf{x}, C) = 0$ es exactamente $A_i$.
- Las secuencias $\mathbf{x} \in \mathbb{F}_2^{23}$ de peso $i$ tal que $d(\mathbf{x}, C) = 1$ se dividen en dos grupos: las que tienen un 1 más que una palabra de código de peso $i - 1$, y las que tiene un 1 de menos que una palabra de código de peso $i + 1$. Cuántas hay del primer grupo?

  $$\binom{n - i + 1}{1} A_{i-1},$$

  y del segundo grupo

  $$\binom{i + 1}{1} A_{i+1}.$$

- Las secuencias $\mathbf{x} \in \mathbb{F}_2^{23}$ de peso $i$ tal que $d(\mathbf{x}, C) = 2$ se dividen en tres grupos: las que tienen dos 1's más que una palabra de código de peso $i - 2$, las que tiene dos 1's menos que una palabra de código de peso $i + 2$, y las que tienen un 1 en la posición de un 0 en la palabra de código y un 0 en la posición de un 1 en la palabra de código.

  Cuántas hay en total de cada tipo? Del primero hay

  $$\binom{n - i + 2}{2} A_{i-2},$$

del segundo tipo hay

$$\binom{i+2}{2}A_{i+2},$$

y del tercer grupo hay

$$\binom{i}{1}\binom{n-i}{1}A_i.$$

- Finalmente, las secuencias $\mathbf{x} \in \mathbb{F}_2^{23}$ de peso $i$ tal que $d(\mathbf{x}, C) = 3$ se dividen en cuatro grupos: las que tienen tres 1's más que una palabra de código de peso $i-3$, las que tiene tres 1's menos que una palabra de código de peso $i+3$, las que tienen dos 1's en las posiciones de dos 0's en la palabra de código y un 0 en la posición de un 1 en la palabra de código, y las que tienen un 1 en las posición de un 0 en la palabra de código y dos 0's en las posiciones de dos 1's en la palabra de código.

  Cuántas hay en total de cada tipo? Del primero hay

$$\binom{n-i+3}{3}A_{i-3},$$

  del segundo tipo hay

$$\binom{i+3}{3}A_{i+3},$$

  del tercer grupo hay

$$\binom{i+1}{2}\binom{n-i-1}{1}A_{i+1},$$

  y del cuarto grupo

$$\binom{i-1}{1}\binom{n-i+1}{1}A_{i-1}.$$

Sumando todos los términos obtenemos

$$\begin{aligned}\binom{n}{i} =& A_i + \binom{n-i+1}{1}A_{i-1} + \binom{i+1}{1}A_{i+1} + \binom{n-i+2}{2}A_{i-2} + \binom{i+2}{2}A_{i+2} + \binom{i}{1}\binom{n-i}{1}A_i \\ &+ \binom{n-i+3}{3}A_{i-3} + \binom{i+3}{3}A_{i+3} + \binom{i+1}{2}\binom{n-i-1}{1}A_{i+1} + \binom{i-1}{1}\binom{n-i+1}{1}A_{i-1}.\end{aligned}$$

Reordenando términos

$$\begin{aligned}A_{i+3} = \Bigg[\binom{n}{i} - \Bigg(\binom{n-i+3}{3}A_{i-3} + \binom{n-i+2}{2}A_{i-2} + \Big[\binom{n-i+1}{1} + \binom{i+1}{2}\binom{n-i-1}{1}\Big]A_{i-1} \\ + \Big[1 + \binom{i}{1}\binom{n-i}{1}\Big]A_i + \Big[\binom{i+1}{1} + \binom{i+1}{2}\binom{n-i-1}{1}\Big]A_{i+1} + \binom{i+2}{2}A_{i+2}\Bigg] \Big/ \binom{i+3}{3}\end{aligned}$$

Tenemos los siguientes valores:

- $A_0 = 1$
- $A_1 = \ldots = A_6 = 0$.

- Si $i = 4$ podemos obtener $A_7$:

$$\binom{23}{4} = \binom{7}{3} A_7 \Rightarrow A_7 = 253.$$

El resto de los valores los podemos obtener programando la recursión anterior. Obtenemos en definitiva

| $i$ | 0 | 7 | 8 | 11 | 12 | 15 | 16 | 23 |
|---|---|---|---|---|---|---|---|---|
| $A_i$ | 1 | 253 | 506 | 1288 | 1288 | 506 | 253 | 1 |

El resto de los coeficientes son 0.

(b) Exprese $A_i^*$ en función de $A_i$ y $A_{i-1}$.

Al extender el código, agregamos un símbolo de paridad a cada palabra de código. De la distribución de pesos del código original notamos que si la palabra es de peso impar (7, 11 y 15) el bit de paridad que se agrega es 1 por lo que el peso de la correspondiente palabra extendida es par.

Si el peso de la palabra es par (8, 12 y 16) el bit de paridad es cero, y el peso de la palabra sigue siendo par.

Por lo tanto, el código extendido $\mathcal{C}^*(24, 12, 8)$ no tiene secuencias de peso par, es decir, $A_i^* = 0$ para $i$ impar. En consecuencia, podemos afirmar que si $i$ es par, entonces

$$A_i^* = A_i + A_{i-1}.$$

(c) Utilice ambas expresiones para determinar la expresión de la función enumeradora de peso de cada código. Utilizando el resultado de la parte anterior tenemos

| $i$ | 0 | 8 | 12 | 16 | 24 |
|---|---|---|---|---|---|
| $A_i^*$ | 1 | 759 | 2576 | 759 | 1 |

5. **Decodificador Lineal :**
Recuerde que el síndrome $\mathbf{s}$ es una función lineal del patrón de error $\mathbf{e}$. Un *decodificador lineal* produce una estimación $\hat{\mathbf{e}}$ del patrón de error como función lineal del síndrome $\mathbf{s}$. Esto es,

$$\hat{\mathbf{e}} = L(\mathbf{s}),$$

donde $L(\mathbf{s} + \mathbf{s}') = L(\mathbf{s}) + L(\mathbf{s}')$.
Demuestre que un decodificador para un codificador lineal binario puede corregir hasta un máximo de $n - k$ de un total posible de $n$ errores. Concluya que la labor de decodificación es inherentemente no lineal (esto es un hecho extremadamente importante).
**Demostración**

El conjunto de todos los síndromes posibles $\mathbf{s}$ es igual a $\mathbb{F}_2^{n-k}$, el que es un espacio vectorial de dimensión $n - k$. La imagen de $\mathbb{F}_2^{n-k}$ a través de cualquier transformación lineal $L(\mathbf{s})$ no puede tener una dimensión mayor que $n - k$.
Por otro lado, $\mathbb{F}_2^{n-k}$ es un subespacio de $\mathbb{F}_2^n$, cuya base canónica son las secuencias con un sólo 1 en alguna de las $n$ posiciones posibles. Esta base corresponde a todos los patrones posibles de un solo error.
Como $\dim(\{L(\mathbf{s}) : \mathbf{s}\} \leq n-k$, como máximo $n-k$ de los $n$ posibles errores unitarios pueden pertenecer al conjunto, lo que demuestra la proposición.