Sección de Matemáticas
Universidad de La Laguna

Alba Santos Rodríguez

Códigos Producto de Matrices

Matrix-Product Codes

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Julio de 2021

DIRIGIDO POR
Irene Márquez Corbella

Irene Márquez Corbella
Matemáticas, Estadística e I.O.
Universidad de La Laguna
38200 La Laguna, Tenerife

Agradecimientos

A mi tutora Irene, por toda la ayuda que me aportado en la realización de este trabajo, y a mi familia y amigos, por su inestimable apoyo.

Alba Santos Rodríguez
La Laguna, 6 de julio de 2021

Resumen · Abstract

Resumen

La Teoría de códigos es una rama de las matemáticas que lidia con el problema de la transmisión eficiente de un mensaje. En este trabajo, hablaremos de teoría de Códigos Correctores. Estos códigos son capaces de detectar y corregir errores que se hayan producido en la transmisión de un mensaje a través de un canal con ruido.
En particular, nos centraremos en Códigos Lineales, una familia de códigos que tienen una codificación eficiente, definida por una aplicación lineal. Estudiaremos sus propiedades, para después definir una serie de construcciones de códigos interesantes. Posteriormente, generalizaremos estas construcciones previas definiendo el Código Producto de Matrices. Esta construcción nos permite crear un nuevo código de mayor longitud, a partir de códigos lineales más pequeños. Los parámetros de estos códigos vendrán definidos por los parámetros de los códigos pequeños, y la matriz que consideremos. Finalmente, describiremos un algoritmo de decodificación eficiente para este tipo de códigos, que corrige el número máximo posible de errores.

Palabras clave: *Códigos Lineales – Código Producto de Matrices – Algoritmo de Decodificación*

Abstract

Coding theory is a branch of mathematics that deals with the problem of a message's efficient transmission. In this work, we will talk about Error-correcting codes. These codes are capable of detecting and correcting errors that may have occurred during the transmission of the message through a noisy channel.
Particularly, we will focus on Linear Codes, a family of codes which have an efficient encoding, defined by a linear map. We will study their properties, and later on, we will define some interesting code constructions. Hereafter, we will generalize the previous ones, defining the Matrix Product Codes. This construction allows us to create a new, longer code from tinier linear codes. These codes parameters will be defined by the parameters of the smaller codes and the matrix. Finally, we will describe an efficient decoding algorithm for this type of codes that corrects the maximum possible number of errors.

Keywords: *Linear Codes - Matrix-Product Codes - Decoding Algorithm*

Contenido

Introducción

Todo lo que nos rodea es información. A diario nos vemos inmersos en cientos de procesos de comunicación. La vida, y más ahora en el siglo XXI, está marcada por el constante intercambio de información. A lo largo de la historia, con el desarrollo de la tecnología, los procesos comunicativos se han ido sofisticando más y más, hasta el punto de ser capaces de almacenar cientos de GBs de información en un pequeño pendrive, o poder recuperar imagenes enviadas desde un satélite en el espacio. Estos avances se lo debemos en gran parte al auge de las comunicaciones a mediados del siglo XX.

Durante esta época, el matemático e ingeniero eléctrico Claude Shannon publicó *A Mathematical Theory of Communication (1948)*, obra que puso los cimientos de la **Teoría de la Información** y de la **Teoría de Códigos**. En esta publicación, Shannon desarrolla una serie de leyes matemáticas que se ocupan de la medición y representación de la información, y que rigen su transmisión y procesamiento, así como la capacidad de los canales sobre los que se transmite la información. Shannon además propone un modelo esquemático del proceso de comunicación: una *fuente de información* entrega un *mensaje*, que es codificado por un *transmisor* en una *señal* transmitida. La señal recibida es la suma de señal transmitida y un inevitable *ruido*. Shannon asume que las interferencias en cualquier canal de comunicación son inevitables y propone una manera revolucionaria de combatirlo: añadir redundancia al mensaje que se envía, codificándolo. Así, a pesar de las interferencias, siempre se pueden corregir los errores y recuperar la información original. Pese a la innovación que supuso este resultado, presentaba un inconveniente: su prueba era no constructiva, dejando por tanto abierto el problema de diseñar medios de codificación y decodificación.

Las ideas expuestas en esta publicación pronto fueron acogidas con gran entusiasmo por ingenieros y matemáticos alrededor del mundo; en particular, por otro matemático, llamado Richard Hamming, que trabajó junto a Shannon en los Laboratorios Bell. Hamming, que se encontraba exasperado con el alto número de errores que cometían los equipos con los que trabajaba al leer tarjetas perforadas, estaba decidido a solucionar el problema. Dos años después de la publicación de Shannon, publicó en el *Bell System Technical Journal* el artículo fundamental *Error detecting and error correcting codes (1950)*, en donde introdujo el concepto de distancia de Hamming, y definió los códigos de Hamming.

Estas publicaciones sentaron las bases de lo que hoy conocemos como **Teoría de Códigos Correctores**, una rama de la matemática que surge con el objetivo de transferir un mensaje de forma eficiente (usando el menor número de recursos en el menor tiempo posible) y fiable (el receptor debe ser capaz de recuperar el mensaje original). En ella, se emplea el álgebra para proteger la transmisión de datos de los errores (ya sea este error humano, ruido del canal, interferencias, etc.).

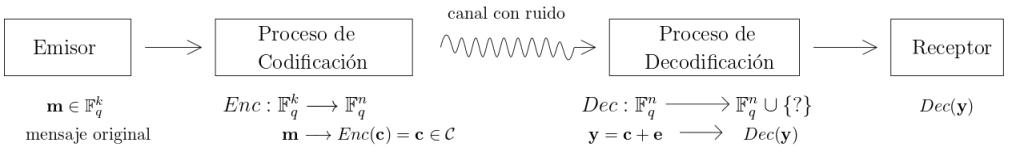
A la hora de establecer una comunicación con códigos, debemos elegir un alfabeto $\mathcal{A}$. En este trabajo, usaremos como alfabeto el cuerpo finito de q elementos, $\mathcal{A} = \mathbb{F}_q$. El mensaje $\mathbf{m}$ que queremos enviar será una k-tupla de elementos del alfabeto, $\mathbf{m} = (m_1, \ldots, m_k) \in \mathcal{A}^k = \mathbb{F}_q^k$. Para evitar los posibles errores que se produzcan al enviarlo por un canal con ruido, este mensaje $\mathbf{m}$ pasará por un proceso de codificación, que viene definido por la aplicación inyectiva

$$\begin{aligned} Enc : \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ \mathbf{m} &\longrightarrow \mathbf{c} \qquad \text{con } n > k. \end{aligned}$$

Llamaremos **código** $\mathcal{C} \subseteq \mathbb{F}_q^n$ a la imagen de esta aplicación, y llamaremos **palabra codificada**, o simplemente palabra del código, a todo vector $\mathbf{c} \in \mathcal{C}$. La codificación es una transformación que añade información redundante; tenemos k dígitos de información y añadimos $n - k$ bits redundantes. Esta información extra nos ayudará luego a recuperar el mensaje original.

Una vez definida la codificación, podemos hablar del proceso de comunicación con códigos. Este se divide en cuatro fases:

1. Nuestro mensaje $\mathbf{m} \in \mathbb{F}_q^k$ se codifica hasta convertirlo en una palabra $\mathbf{c}$ del código $\mathcal{C} \subseteq \mathbb{F}_q^n$.
2. Esta palabra se envía a través de un canal donde se pueden producir errores en el mensaje.
3. Recibimos un vector $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_q^n$, donde $\mathbf{e} \in \mathbb{F}_q^n$ representa los errores que se han producido.
4. Por último, realizamos el proceso más delicado, la decodificación. Buscamos recuperar la palabra original, sabiendo que, si $\mathbf{y} \notin \mathcal{C}$, se han producido errores, pero no pudiendo asegurar nada si $\mathbf{y} \in \mathcal{C}$.



El canal que consideraremos en este trabajo tiene ciertas particularidades. Utilizaremos un canal **discreto** y **sin memoria**. Esto quiere decir que:

- El conjunto de símbolos de entrada y salida es el alfabeto $\mathcal{A} = \mathbb{F}_q$; siempre que enviamos una letra de $\mathbb{F}_q$, recibimos una letra de $\mathbb{F}_q$.
- La transmisión de un símbolo no está influenciada por la transmisión de símbolos anteriores; transmitir un símbolo a través del canal significa asociar a dicho símbo-

lo, aleatoriamente, un símbolo según una distribución de probabilidades asociada al canal.

Por lo tanto, en nuestro trabajo no están permitidos errores como deleción de letras del mensaje o rotación de las letras de un mensaje.

Centrándonos en la construcción de códigos, nos interesa que:

- Tengan una alta capacidad de corrección. Esto lo conseguiremos con una distancia mínima entre palabras del código lo mayor posible.
- Sean eficientes; es decir, que tengan el menor número de bits de información redundante posible.
- Posean algoritmos de codificación y decodificación rápidos. A priori, esta última condición no es fácil de conseguir.

En cuanto a sus aplicaciones, la Teoría de Códigos está presente en infinidad de ejemplos de la vida real. Puede servirnos simplemente para detectar errores, como en el DNI o el código ISBN de los libros, donde tenemos unos dígitos de control que nos permiten saber si el resto de información es correcta; como también para detectar errores y corregirlos, como es el caso de los codigos Reed-Solomon, que se emplean en la comunicación por satélite, o en los CDs.

Este trabajo estará centrado en un tipo de códigos, llamados **Códigos Producto de Matrices**, de los que estudiaremos sus propiedades y daremos un algoritmo de decodificación eficiente. El contenido de este trabajo estará dividido en 3 capítulos y un apéndice.

En el primer capítulo fijaremos desde el principio el alfabeto que utilizaremos, el cuerpo finito $\mathbb{F}_q$, con $q = p^r$, donde p es primo. Este cuerpo finito tiene una representación principal, dada por el anillo cociente del dominio de polinomios en una variable con coeficientes en $\mathbb{F}_p$, módulo un polinomio en $\mathbb{F}_p[x]$ irreducible de grado r. Para probar la existencia de este polinomio, recurriremos a la Fórmula de Gauss para polinomios irreducibles sobre $\mathbb{F}_q$. Posteriormente, daremos unas nociones básicas sobre **Teoría de Códigos**, para luego adentrarnos en el estudio de una familia de códigos en particular, llamada **Códigos Lineales**. La codificación de estos códigos viene definida por una aplicación lineal que nos permite una codificación rápida. A continuación, hablaremos del problema de decodificación en estos códigos e introduciremos un par de definiciones y resultados necesarios. Esta decodificación no será un proceso fácil, ya que en general no se conocen algoritmos rápidos de decodificación que funcionen para todos los códigos lineales.

En el segundo capítulo nos centraremos en los **Códigos Producto de Matrices**. En primer lugar introduciremos algunas construcciones de códigos , que serán casos particulares de un código producto de matrices, para después dar su definición en general. Estos códigos presentan una gran ventaja: nos permiten definir nuevos códigos lineales de mayor longitud a partir de códigos lineales $\mathcal{C}_i$ más pequeños que ya conozcamos. Probaremos que, eligiendo una matriz con ciertas características,

y conociendo los parámetros de los códigos lineales que consideremos, podremos obtener fácilmente los parámetros del código producto de matrices.

En el tercer capítulo hablaremos de la decodificación del código producto de matrices, para el caso particular en el que los códigos lineales $\mathcal{C}_i$ están anidados. En este caso, tendremos un algoritmo de decodificación eficiente para estos códigos, del que demostraremos que corrige el máximo número de errores posible. Finalmente, veremos dos ejemplos de decodificación, construyendo nuestro código producto de matrices a partir de **Códigos de Reed-Solomon**. En estos ejemplos veremos cómo cambia el proceso de decodificación dependiendo del tipo de decodificadores DC_i para los códigos $\mathcal{C}_i$ de Reed-Solomon que consideremos.

Los códigos de Reed-Solomon serán trabajados en el apéndice, donde hablaremos de sus propiedades y daremos un algoritmo de decodificación eficiente para ellos, que también corrige el máximo número de errores posible.

1

Teoría de Códigos Correctores

Este capítulo nos servirá de introducción a la Teoría de Códigos lineales. En él, estableceremos nuestro alfabeto, el cuerpo finito de q elementos $\mathbb{F}_q$, y sentaremos las bases de teoría de códigos necesarias para los siguientes capítulos, donde estudiaremos ciertos códigos lineales en particular.

1.1. Cuerpos Finitos

Antes de comenzar con Teoría de Códigos, es conveniente recordar ciertos resultados sobre cuerpos finitos. Estos son útiles en numerosas ramas de la matemática; en nuestro caso particular, utilizaremos la teoría de cuerpos finitos para desarrollar la **Teoría de Códigos Lineales**.

1.1.1. Definiciones básicas

Primero, definiremos una serie de conceptos que serán utilizados en las proposiciones posteriores.

Definición 1.1. *Diremos que $\mathbb{F}$ es un **cuerpo** si $\mathbb{F}$ es un anillo conmutativo y unitario, en el que todo elemento distinto de cero es invertible respecto del producto.*
*Diremos que $\mathbb{F}$ es un **cuerpo finito**, si, como bien indica su nombre, es un cuerpo con un número finito de elementos.*

Definición 1.2. *Sean $\mathbb{L}$ y $\mathbb{K}$ dos cuerpos, tales que $\mathbb{K} \subseteq \mathbb{L}$. Diremos que $\mathbb{K}$ es subcuerpo **maximal** de $\mathbb{L}$ si, para cualquier cuerpo $\mathbb{F}$ tal que $\mathbb{K} \subseteq \mathbb{F} \subseteq \mathbb{L}$, se tiene que $\mathbb{F} = \mathbb{K}$ o bien $\mathbb{F} = \mathbb{L}$.*

Definición 1.3. *Sean $\mathbb{K}$ y $\mathbb{L}$ dos cuerpos tales que $\mathbb{K} \subset \mathbb{L}$, y el polinomio $f(x) \in \mathbb{K}[x]$. Si $f(x) = f_1(x) \cdots f_r(x)$, con $f_i(x) \in \mathbb{L}[x]$ irreducibles de grado 1, entonces $f(x)$ se descompone en $\mathbb{L}[x]$ en factores simples. Al menor cuerpo que verifica esta propiedad se le llama **cuerpo de descomposición de** $f(x)$.*

Definición 1.4. *Sea A un anillo unitario, donde denotamos el neutro para el producto como 1_A. El homomorfismo característico de A es el único homomorfismo de anillos de $\mathbb{Z}$ en A que se define de la siguiente manera:*

$$\begin{aligned} \varphi_A : \mathbb{Z} &\longrightarrow A \\ 1 &\longrightarrow 1_A \\ n &\longrightarrow 1_A + \overset{n \text{ veces}}{\cdots} + 1_A, \text{ con } n > 0 \end{aligned}$$

*Como $\mathbb{Z}$ es un dominio de ideales principales, existe $b \in \mathbb{N}$ tal que $ker(\varphi_A) =< b >$. Al entero b se le llama **característica** de A y se denota por $car(A)$.*

Proposición 1.5. *Sea A un dominio de integridad, entonces $car(A) = 0$ o $car(A) = p$ con p primo.*

Demostración. En el caso en que el homomorfismo ϕ_A es inyectivo, diremos que $car(A) = 0$.
Veamos ahora el caso en el que el homomorfismo ϕ_A no es inyectivo, es decir, $car(A) \neq 0$. Supongamos que $car(A)$ no es primo, es decir, podemos descomponer la característica en $car(A) = s \cdot r$, con $r, s < n$. Se tiene entonces que $0 = car(A) \cdot 1 = (s \cdot r) \cdot 1 = s \cdot r$. Como A es dominio de integridad, tenemos que $s = 0$ o $r = 0$. Con lo cual llegamos a una contradicción, ya que $car(A)$ es el entero más pequeño tal que $car(A) \cdot 1 = 0$. Por tanto, $car(A)$ debe ser primo.

□

Definición 1.6. *Sea $\alpha \in \mathbb{L}$ un elemento algebraico*[1] *sobre $\mathbb{K}$ donde $\mathbb{L}$ es una extensión* [2]

*de $\mathbb{K}$. Al polinomio de menor grado mónico que anula a α con coeficientes en $\mathbb{K}$ se le llama **polinomio mínimo de α sobre** $\mathbb{K}$ y se denota $m_{\alpha,\mathbb{K}} \in \mathbb{K}[x]$.*

1.1.2. Algunos resultados sobre Cuerpos Finitos

A continuación, probaremos una serie de resultados que nos serán útiles en la sección.

Teorema 1.7. *Un cuerpo finito de q elementos existe si y solo si q es potencia de un primo. Además, dicho cuerpo es único salvo isomorfismo, y se denota como $\mathbb{F}_q$.*

Para probar este Teorema, demostraremos antes un par de proposiciones necesarias.

Proposición 1.8. *Sea $\mathbb{F}_q$ un cuerpo finito de q elementos, con $q = p^r$, con p primo. Consideremos dos elementos cualesquiera α_i y $\alpha_j \in \mathbb{F}_q$. Entonces, se tiene que:*

1. $p \cdot \alpha_i = 0$, *es decir, la característica de $\mathbb{F}_q$ es p.*
2. $(\alpha_i + \alpha_j)^q = \alpha_i^q + \alpha_j^q$.
3. $\alpha_i^q = \alpha_i$.

Demostración.

[1] Un elemento $\alpha \in \mathbb{L}$ se dice algebraico sobre $\mathbb{K}$ si es cero de algún polinomio en $\mathbb{K}[x]$. En caso contrario diremos que α es trascendental sobre $\mathbb{K}$.
[2] Sean $\mathbb{L}$ y $\mathbb{K}$ cuerpos. $\mathbb{L}$ es una extensión de $\mathbb{K}$ si $\mathbb{K} \subset \mathbb{L}$, y $\mathbb{K}$ es un subanillo de $\mathbb{L}$. La denotaremos por $\mathbb{L}/\mathbb{K}$. Diremos que la extensión $\mathbb{L}/\mathbb{K}$ es algebraica si cada elemento de $\mathbb{L}$ es algebraico sobre $\mathbb{K}$.

1. Como $\mathbb{F}_q$ es finito, la aplicación φ_A no es inyectiva, y por la Proposición 1.5, se tiene que $car(\mathbb{F}_q) = p$, con p primo.
2. Sea $q = p^r$. Procedemos por inducción sobre r. Sea $r = 1$. Entonces, por el binomio de Newton tenemos que
$$(a+b)^p = \binom{p}{0}a^p + \binom{p}{1}a^{p-1}b + \cdots + \binom{p}{p-1}ab^{p-1} + \binom{p}{p}b^p.$$
Como $\binom{p}{i}$ es múltiplo de p, $\forall i = 1, \ldots, p-1$, se tiene que
$$(a+b)^p = a^p + b^p.$$
Supongamos cierto el enunciado cierto para $r' < r$. Es decir, $(a+b)^{p^{r'}} = a^{p^{r'}} + b^{p^{r'}}$. Luego, para r, aplicando nuestra hipótesis de inducción, se tiene que:
$$(a+b)^q = (a+b)^{p^r} = \left((a+b)^{p^{r-1}}\right)^p = \left(a^{p^{r-1}} + b^{p^{r-1}}\right)^p = a^{p^r} + b^{p^r} = a^q + b^q.$$
3. Observemos que, si $\alpha_i = 0$, entonces se tiene la igualdad. Supongamos ahora que $\alpha_i \in \mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$. Como $\mathbb{F}_q^*$ es un grupo multiplicativo de orden $q-1$, se tiene que $\alpha_i^{q-1} = 1$, $\forall \alpha_i \in \mathbb{F}_q$. Por tanto concluimos que $\alpha_i^q = \alpha_i$.

□

Proposición 1.9. *Sea $\mathbb{F}_q$ un cuerpo finito de q elementos. $\mathbb{F}_q$ existe si q es potencia de un primo.*

Demostración. Veamos que:

1. Aplicando el primer teorema de isomorfía a φ_A, se tiene que $\mathbb{Z}_p = \mathbb{F}_p \cong Im(\varphi_A) \subseteq \mathbb{F}_q$. Luego $\mathbb{F}_q$ contiene al subcuerpo $\mathbb{F}_p$.
2. $\mathbb{F}_q$ es cuerpo, luego $(\mathbb{F}_q, +)$ es grupo. Por tanto, podemos definir el producto escalar $\lambda \cdot x \in \mathbb{F}_q$, $\forall \lambda \in \mathbb{F}_p$, $\forall x \in \mathbb{F}_q$ (ya que $\mathbb{F}_p \subseteq \mathbb{F}_q$). Luego $\mathbb{F}_q$ es $\mathbb{F}_p$-espacio vectorial. Además, como $\mathbb{F}_q$ es finito, se tiene entonces que $dim_{\mathbb{F}_p}(\mathbb{F}_q) = n < \infty$.
3. El número de elementos es $q = p^r$, con p primo. Si consideramos $\alpha_1, \ldots, \alpha_r$ los elementos de una base de $\mathbb{F}_q$ sobre $\mathbb{F}_p$, tendremos que cualquier elemento de $\mathbb{F}_q$ se puede escribir de la forma $\lambda_1\alpha_1 + \cdots + \lambda_r\alpha_r$, con $\lambda_i \in \mathbb{F}_p$. Por lo tanto, el cardinal de $\mathbb{F}_q$ es p^r.

□

Una vez probados estos resultados, nos encontramos en condiciones de probar el Teorema 1.7.

Demostración del Teorema 1.7. Ya sabemos, por la Proposición 1.9, que si tenemos un cuerpo finito de q elementos, entonces $q = p^r$ con p primo. Veamos ahora que, para cualquier potencia de un primo, existe un cuerpo finito con ese cardinal.
Consideremos el polinomio $P(x) = x^q - x \in \mathbb{F}_p[x]$ y sea R el conjunto de raíces de $P(x)$, en la clausura algebraica[3] de $\mathbb{F}_p$. Denotamos como $P'(x)$ a la primera derivada

[3] Diremos que la extensión $\mathbb{L}/\mathbb{K}$ es una **clausura algebraica** si la extensión es algebraica y todo polinomio irreducible en $\mathbb{K}[x]$ factoriza sobre $\mathbb{L}$.

de $P(x)$. Como $mcd(P(x), P'(x)) = 1$, tenemos que $P(x)$ no tiene raíces múltiples. Es decir, el cardinal de R es q. Además, R cumple que:

- Los elementos neutros pertenecen a R, ya que 0 y 1 son raíz de $F(x)$.
- La suma y multiplicación de elementos de R está en R, debido a que $(\alpha_i + \alpha_j)^q = \alpha_i^q + \alpha_j^q$, y a que $\alpha_i^q = \alpha_i$ en $\mathbb{F}_q$ (por la Proposición 1.8).
- El inverso multiplicativo está en R, ya que $(\alpha_i^{-1})^q = (\alpha_i^q)^{-1} = \alpha_i^{-1}$.
- El inverso aditivo está en R, puesto que:
  - Si q es impar, $(-\alpha_i)^q = -(\alpha_i^q) = -\alpha_i$. Luego $P(-\alpha) = 0$.
  - Si q es par, $q = 2^r$. Por lo tanto, la característica de R es 2. Luego, $P(-\alpha) = 2 \cdot \alpha = 0$.

Luego R es cuerpo con q elementos. Además, este cuerpo es único salvo isomorfismo; ya que, si consideramos otro cuerpo K de q elementos (que serán las raíces del polinomio $x^q - x$), podremos identificar cada elemento de R con uno de K.

□

Por el resultado anterior, sabemos que para todo $q = p^r$, con p primo, existe un cuerpo $\mathbb{F}_q$. Veamos ahora cómo podemos construir este cuerpo.
Consideremos el anillo cociente del dominio de polinomios en una variable con coeficientes en $\mathbb{F}_p$, módulo un polinomio irreducible de grado r. Tendremos que $\mathbb{F}_q \cong \frac{\mathbb{F}_p[x]}{(f(x))}$, donde $f(x) \in \mathbb{F}_p[x]$ es un polinomio irreducible de grado r y $(f(x))$ denota el ideal generado por $f(x)$. Abusando de notación, denotaremos por x a x módulo $f(x)$. Entonces, los monomios $1, x, \ldots, x^{r-1}$ forman una base de $\mathbb{F}_q$ como $\mathbb{F}_p$-espacio vectorial. Por lo tanto, cualquier elemento en este cuerpo se representa de forma única por un polinomio $g(x) \in \mathbb{F}_p[x]$ de grado menor que r. A esta representación de elementos de $\mathbb{F}_q$ se denomina **representación principal** de $\mathbb{F}_q$.

Ejemplo 1.10. Supongamos que queremos construir un cuerpo de $3^2 = 9$ elementos. Tenemos que $f(x) = x^2 + 1 \in \mathbb{F}_3[x]$, es irreducible (es de grado 2 y no tiene raíces en $\mathbb{F}_3$). Luego se sigue que:

$$\mathbb{F}_9 \cong \frac{\mathbb{F}_3[x]}{(x^2+1)} = \{f(x) \in \mathbb{F}_3[x] \mid deg(f) < 2\} = \{0, 1, 2, x, 2x, x+1, x+2, 2x+1, 2x+2\}$$

La pregunta que nos hacemos a continuación es: dado $q = p^r$, ¿existe siempre un polinomio irreducible $f(x) \in \mathbb{F}_p[x]$ con $deg(f) = r$? Para comprobar la existencia de este polinomio, utilizaremos la **Fórmula de Gauss**, que nos da el número de polinomios irreducibles sobre $\mathbb{F}_q$.
Las siguientes proposiciones son resultados que utilizaremos para demostrar dicha fórmula (Teorema 1.18).

Proposición 1.11. *Las raíces de un polinomio irreducible sobre $\mathbb{F}_q$ son siempre distintas.*

Demostración. Consideremos un elemento $\alpha \in \mathbb{L}$, con $\mathbb{L}$ cuerpo tal que $\mathbb{F}_q \subseteq \mathbb{L}$, que cumple que $f(\alpha) = 0$ para algún polinomio $f(x) \in \mathbb{F}_q[x]$ irreducible. Luego, por los resultados de la Proposición 1.8, tenemos que:

$$0 = 0^q = (f(\alpha))^q = \left[\sum_{i=0}^{n} a_i \cdot \alpha^i\right]^q = \sum_{i=0}^{n} a_i^q \cdot (\alpha^i)^q = \sum_{i=0}^{n} a_i \cdot (\alpha^q)^i = f(\alpha^q)$$

Por tanto, $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$ son raíces de $f(x)$. Veamos que son distintas.
Supongamos que $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{m-1}}$ son raíces distintas, con $m < n$, pero que $\alpha^{q^m} = \alpha^{q^i}$ para algún $i \in \{0, \ldots, m-1\}$. Si $i > 0$, tenemos que $\left(\alpha^{q^{i-1}}\right)^q = \alpha^{q^i} = \alpha^{q^m} = \left(\alpha^{q^{m-1}}\right)^q$. Luego $\left(\alpha^{q^{i-1}}\right)^q - \left(\alpha^{q^{m-1}}\right)^q = \left(\alpha^{q^{i-1}} - \alpha^{q^{m-1}}\right)^q = 0$, lo que implica que $\alpha^{q^{i-1}} = \alpha^{q^{m-1}}$, que contradice la hipótesis de que $\alpha, \ldots, \alpha^{q^{m-1}}$ sean distintas. Con lo cual, tenemos que $\alpha^{q^m} = \alpha$.
Ahora definimos un polinomio que tiene como raíces a $\alpha, \ldots, \alpha^{q^{m-1}}$.

$$g(x) = \prod_{i=0}^{m-1} (x - \alpha^{q^i}) = \sum_{j=0}^{m} b_j \cdot x^j.$$

Entonces:

$$(g(x))^q = \prod_{i=0}^{m-1} \left(x - \alpha^{q^i}\right)^q = \prod_{i=0}^{m-1} \left(x^q - \alpha^{q^{i+1}}\right) = \prod_{k=1}^{m} \left(x^q - \alpha^{q^k}\right) =$$
$$= \prod_{i=0}^{m-1} \left(x^q - \alpha^{q^i}\right) = g(x^q).$$

Luego tenemos que

$$(g(x))^q = \sum_{j=j_0}^{m} \left(b_j \cdot x^j\right)^q = \sum_{j=j_0}^{m} b_j^q \cdot \left(x^j\right)^q = \sum_{j_0}^{m} b_j \cdot (x^q)^j = g(x^q).$$

Es decir, $b_j^q = b_j \; \forall \, 0 \leq j \leq m$, por lo que deducimos que $g(x) \in \mathbb{F}_q[x]$.
Hemos visto que $g(x)$ es divisor de $f(x)$ que, por hipótesis, es irreducible en $\mathbb{F}_q[x]$.Esto implica que $m = n$ y $f(x) = g(x) \cdot \lambda$, siendo λ un escalar perteneciente a $\mathbb{F}_q$. Luego $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$ son las n raíces distintas del polinomio $f(x) \in \mathbb{F}_q[x]$.

□

Proposición 1.12. *$\mathbb{F}_{q^n}$ es el cuerpo de descomposición de cualquier polinomio irreducible $p(x)$ de grado n sobre $\mathbb{F}_q$. Es decir, $p(x)$ se descompone en factores lineales sobre $\mathbb{F}_{q^n}$, pero no sobre ningún subcuerpo de $\mathbb{F}_{q^n}$ más pequeño.*

Demostración. Sea $p(x) = a_0 + \cdots + a_n \cdot x^n$ con $a_i \in \mathbb{F}_q$, irreducible. El cuerpo de descomposición de $p(x)$ será $\mathbb{F}_q(\alpha_1, \ldots, \alpha_n)$, donde α_i son las raíces de $p(x)$.
Por la Proposición 1.11 , sabemos que todas las raíces son diferentes entre sí, luego $\mathbb{F}_q(\alpha_1, \ldots, \alpha_n)$ tendrá q^n elementos. Y por el Teorema 1.7 , sabemos que los cuerpos finitos son únicos salvo isomorfismo. Por lo tanto, $\mathbb{F}_q(\alpha_1, \ldots, \alpha_n) = \mathbb{F}_{q^n}$

□

Proposición 1.13. *Dos polinomios irreducibles sobre $\mathbb{F}_q$ no pueden tener una raíz en común.*

Demostración. Supongamos que $f(x)$ y $g(x) \in \mathbb{F}_q[x]$ son dos polinomios irreducibles, de grado n y m respectivamente ($n \leq m$). Si tienen en común una raíz α, tendrán en común las raíces $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{m-1}}$. Entonces, por la Proposición 1.11, tenemos que $f(x) = g(x) \cdot p(x)$, con $p(x)$ el polinomio que tiene las raíces $\alpha^{q^m}, \ldots, \alpha^{q^{n-1}}$. Con lo cual llegamos a un absurdo, ya que habíamos supuesto que $f(x)$ era irreducible.

□

Proposición 1.14. *$\mathbb{F}_{q^a} \subseteq \mathbb{F}_{q^b}$ si y solo si a divide a b.*

Demostración. Consideremos la torre de cuerpos $\mathbb{F}_q \hookrightarrow \mathbb{F}_{q^a} \hookrightarrow \mathbb{F}_{q^b}$. Entonces tenemos que $\left[\mathbb{F}_{q^b} : \mathbb{F}_q\right] = b = \left[\mathbb{F}_{q^b} : \mathbb{F}_{q^a}\right] \cdot \left[\mathbb{F}_{q^a} : \mathbb{F}_q\right] = s \cdot a$. Por lo tanto, a divide a b.
Recíprocamente, si a divide a b, tenemos que $b = s \cdot a$ para algún $s \in \mathbb{Z}$.
Sabemos además que los elementos de un cuerpo $\mathbb{F}_{q^n}$ son las raíces del polinomio $x^{q^n} - x \in \mathbb{F}_q[x]$. Por tanto, $\mathbb{F}_{q^b}$ serán las raíces de

$$f(x) = x^{q^b} - x = x^{q^{s \cdot a}} - x.$$

De donde se concluye que $\mathbb{F}_{q^a} \subseteq \mathbb{F}_{q^b}$.

□

Proposición 1.15. *Sea $n \in \mathbb{N}$ un número cuya factorización en primos es $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$. Los subcuerpos maximales de $\mathbb{F}_{q^n}$ son de la forma $\mathbb{F}_{q^{\frac{n}{p_i}}}$ con $i \in \{1, \ldots, r\}$.*

Demostración. Sea $L \equiv \mathbb{F}_{q^s}$ un subcuerpo maximal de $\mathbb{F}_q^n$. Tenemos que $\mathbb{F}_{q^s} \subsetneq \mathbb{F}_{q^n}$, por tanto $s = p_1^{\beta_1} \cdots p_r^{\beta_r}$ divide a $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$, lo que significa que existe al menos un β_i tal que $\beta_i < \alpha_i$ para cierto $i \in \{1, \ldots, r\}$.
Supongamos que existe más de un índice β_j que cumple la condición anterior. Sin pérdida de generalidad, suponemos que existen dos elementos $\beta_1, \beta_2 \in \mathbb{N}$ tal que $\beta_1 < \alpha_1$ y $\beta_2 < \alpha_2$. Luego tendríamos por la Proposición 1.14 que existe un subcuerpo $\mathbb{F}_{q^{s'}}$ tal que $\mathbb{F}_{q^s} \subsetneq \mathbb{F}_{q^{s'}} \subsetneq \mathbb{F}_{q^n}$, con $s = p_1^{\beta_1} p_2^{\beta_2} \cdots p_r^{\alpha_r}$ y $s' = p_1^{\beta_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$. Por tanto llegamos a un absurdo, pues habíamos supuesto que $\mathbb{F}_{q^s}$ era maximal.

Entonces, supongamos que solo cambia un exponente respecto de la descomposición en primos de n. Sin perdida de generalidad, suponemos que este exponente es β_1. Consideremos ahora otro exponente β_1' tal que $\beta_1 < \beta_1' < \alpha_1$, con $\beta_1, \beta_1', \alpha_1 \in \mathbb{N}$. Luego, para $s = p_1^{\beta_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$ y $s' = p_1^{\beta_1'} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$, tendremos que $\mathbb{F}_{q^s} \subsetneq \mathbb{F}_{q^{s'}} \subsetneq \mathbb{F}_{q^n}$, que de nuevo contradice la hipótesis de que L sea maximal. Por tanto, $\beta_1 = \alpha_1 - 1$ y $\beta_j = \alpha_j, \forall j = 2, \ldots, r$.

Veamos ahora que el recíproco es cierto. Sea el cuerpo $\mathbb{F}_{q^s}$, con $s = \frac{n}{p_i}$ para cierto índice $i \in \{1, \ldots, r\}$. Sin pérdida de generalidad, consideremos que $s = \frac{n}{p_1}$ y supongamos que existe otro cuerpo $\mathbb{F}_{q^m}$ tal que $\mathbb{F}_{q^s} \subsetneq \mathbb{F}_{q^m} \subsetneq \mathbb{F}_{q^n}$. De nuevo por la Proposición 1.14, m divide a n y s divide a m. Entonces, como $s = \frac{n}{p_i}$, se tiene que $m = n$ o $m = s$. Es decir, $\mathbb{F}_{q^s}$ es un subcuerpo propio maximal de F_{q^n}.

□

Proposición 1.16. *Sean $\mathbb{F}_{q^\alpha}$ y $\mathbb{F}_{q^\beta}$ dos cuerpos finitos. Entonces, $\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} = \mathbb{F}_{q^{\text{m.c.d.}(\alpha,\beta)}}$.*

Demostración. Primero, es fácil comprobar que la intersección de cuerpos es cuerpo. Además, al estar en el caso de cuerpos finitos, tenemos que la intersección también será un cuerpo finito. Por tanto, $\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} = \mathbb{F}_{q^s}$. Luego, $\mathbb{F}_{q^s} \subseteq \mathbb{F}_{q^\alpha}$ y $\mathbb{F}_{q^s} \subseteq \mathbb{F}_{q^\beta}$. Entonces, por la Proposición 1.14, se tiene que s divide tanto a α como a β. Es decir, s divide a m.c.d.(α, β), lo que implica que $\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} \subseteq \mathbb{F}_{q^{\text{m.c.d.}(\alpha,\beta)}}$.

Además, sabemos que m.c.d.(α, β) divide tanto a α como a β. Entonces, por la Proposición 1.14 tenemos que $\mathbb{F}_{q^{\text{m.c.d.}(\alpha,\beta)}} \subseteq \mathbb{F}_{q^\alpha}$ y $\mathbb{F}_{q^{\text{m.c.d.}(\alpha,\beta)}} \subseteq \mathbb{F}_{q^\beta}$. Por tanto, $\mathbb{F}_{q^{\text{m.c.d.}(\alpha,\beta)}} \subseteq \mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta}$.

□

Lema 1.17. *Sea α un elemento algebraico sobre $\mathbb{F}_q$. Las siguientes definiciones son equivalentes:*

1. $\left[\mathbb{F}_q(\alpha) : \mathbb{F}_q\right] = n$.
2. *α no está contenido en ningún subcuerpo propio de $\mathbb{F}_{q^n}$.*
3. *α no está contenido en ningún subcuerpo maximal de $\mathbb{F}_{q^n}$.*

Demostración. Como todo subcuerpo maximal es propio, se tiene que 2 implica 3. Además, como todo subcuerpo propio está contenido en un maximal, se tiene que 2 y 3 son equivalentes.
Por otra parte, tenemos que $\mathbb{F}_q(\alpha)$ es el menor subcuerpo que contiene a $\mathbb{F}_q$ y a α. Además, como $[\mathbb{F}_q(\alpha) : \mathbb{F}_q] = n$, el polinomio mínimo $m_{\alpha,\mathbb{F}_q}(x)$ es de grado n. Entonces, por la Proposición 1.12 se tiene que $\mathbb{F}_{q^n}$ es el cuerpo de descomposición de $m_{\alpha,\mathbb{F}_q}(x)$. Luego, $\mathbb{F}_q(\alpha) \subseteq \mathbb{F}_{q^n}$. Y como $[\mathbb{F}_{q^n} : \mathbb{F}_q] = n = [\mathbb{F}_q(\alpha) : \mathbb{F}_q]$, se concluye que $\mathbb{F}_{q^n} = \mathbb{F}_q(\alpha)$. Por tanto 1 es equivalente a 2.

□

1.1.3. Fórmula de Gauss para polinomios irreducibles sobre $\mathbb{F}_q$

En esta subsección, probaremos la fórmula de Gauss utilizando los resultados vistos en la sección anterior, y el Principio de Inclusión-Exclusión.

Este resultado se ha extraído de [4].

Teorema 1.18. ***Fórmula de Gauss.***
El número de polinomios mónicos irreducibles de grado n que existen sobre un cuerpo finito $\mathbb{F}_q$ viene dado por

$$\frac{1}{n}\sum_{d|n}\mu\left(\frac{n}{d}\right)\cdot q^d$$

donde d son todos los divisores positivos de n, y $\mu(r)$ es la función de Möbius.

Observación 1.19. La función de Möbius se define como

$$\mu(n) = \begin{cases} 1 & \text{si } n \text{ es libre de cuadrados y tiene un número par de factores primos distintos.} \\ -1 & \text{si } n \text{ es libre de cuadrados y tiene un número impar de factores primos distintos.} \\ 0 & \text{si } n \text{ es divisible por algún cuadrado.} \end{cases}$$

Demostración. El caso $n = 1$ es trivial, dado que cualquier polinomio mónico de grado 1 es irreducible. Por tanto, habrán q polinomios mónicos irreducibles de grado 1, que es justo lo que obtenemos en la fórmula de Gauss al sustituir $n = 1$. Para $n > 1$, consideramos los conjuntos

$$\mathcal{P}_n = \{\alpha \in \mathbb{F}_q^n[x] \mid p(x) \text{ es mónico irreducible de grado n}\}$$
$$\mathcal{R}_n = \bigcup_{p(x)\in\mathcal{P}_n} \{\alpha \mid \alpha \text{ raíz de } p(x)\}.$$

Por las Proposiciones 1.11 y 1.13, tenemos que cada $p(x) \in \mathcal{P}_n$ aporta n elementos diferentes a $\mathcal{R}_n$. Luego, $|\mathcal{R}_n| = n \cdot |\mathcal{P}_n|$.

Sea $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ la descomposición en factores primos de n. Entonces, por la Proposición 1.15, tenemos que los subcuerpos maximales de $\mathbb{F}_{q^n}$ son de la forma $\mathbb{F}_{q^{\frac{n}{p_i}}}$.

Además, por el Lema 1.17, tenemos que

$$\mathcal{R}_n = \{\alpha \in \mathbb{F}_{q^n} \mid \left[\mathbb{F}_q(\alpha) : \mathbb{F}_q\right] = n\} =$$
$$= \{\alpha \in \mathbb{F}_{q^n} \mid \alpha \text{ no está contenido en ningún subcuerpo propio de } \mathbb{F}_{q^n}\} =$$
$$= \{\alpha \in \mathbb{F}_{q^n} \mid \alpha \text{ no está contenido en ningún subcuerpo maximal de } \mathbb{F}_{q^n}\}.$$

Entonces, empleando la definición del apartado 3 del Lema 1.17, tenemos que $|\mathcal{R}_n| = \left| \left(\bigcup_{i=1}^{r} \mathbb{F}_{q^{\frac{n}{p_i}}} \right)^c \right|$, donde A^c denota el complementario del conjunto A en $\mathbb{F}_{q^n}$.

Por la Proposición 1.16, sabemos que la intersección de maximales será de la forma $\bigcap_{i=1}^{s} \mathbb{F}_{q^{\frac{n}{p_i}}} = \mathbb{F}_{q^{\frac{n}{p_1 \cdots p_s}}}$. Entonces, aplicando el Principio de Inclusión-Exclusión[4] obtenemos:

$$|\mathcal{R}_n| = q^n - q^{\frac{n}{p_1}} - \cdots - q^{\frac{n}{p_r}} + q^{\frac{n}{p_1 \cdot p_2}} + q^{\frac{n}{p_1 \cdot p_3}} + \cdots\cdots + (-1)^r q^{\frac{n}{p_1 \cdots p_r}} =$$
$$= \sum_{d|n} \mu\left(\frac{n}{d}\right) \cdot q^d.$$

Por tanto, finalmente tenemos que $|\mathcal{P}_n| = \dfrac{|\mathcal{R}_n|}{n} = \dfrac{1}{n} \displaystyle\sum_{d|n} \mu\left(\frac{n}{d}\right) \cdot q^d$.

□

1.2. Teoría de Códigos

A continuación, introduciremos algunas definiciones y resultados básicos de Teoría de Códigos.

[4] Principio de Inclusión-Exclusión:

$$\left|\bigcup_{i=1}^{n} A_i\right| = \left(\sum_{i=1}^{n} |A_i|\right) - \left(\sum_{\leq i<j\leq n} |A_i \cap A_j|\right) + \cdots + (-1)^{n-1}\left(|A_1 \cap A_2 \cap \ldots \cap A_{n-1} \cap A_n|\right).$$

1.2.1. Nociones básicas

Definición 1.20. ***(Código bloque)*** *Sea $\mathcal{C} \subseteq \mathbb{F}_q^n$ un código de M elementos ($\#\mathcal{C} = M$). Diremos que $\mathcal{C}$ es un $[n, M]_q$-código.*

Definición 1.21. *Sean $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$. Definimos la **distancia de Hamming entre** $\mathbf{x}$ **e** $\mathbf{y}$ como $d_H(\mathbf{x}, \mathbf{y}) = \#\{i \mid x_i \neq y_i\}$, donde $\#A$ denota el cardinal del conjunto A.*

Definición 1.22. *Sea $\mathbf{x} \in \mathbb{F}_q^n$. Definimos el **soporte** de $\mathbf{x}$ como $supp(\mathbf{x}) = \#\{i \mid x_i \neq 0\}$.*

Proposición 1.23. *La distancia de Hamming es una métrica.*

Demostración. Las propiedades de ser no negativa y simétrica se deducen de la definición.
Veamos que se cumple la desigualdad triangular, empleando ciertos resultados de Teoría de Conjuntos.[5]
Consideremos $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_q^n$, y definimos el conjunto $A = \{i \mid x_i \neq y_i\}$. Es decir, $\#A = d_H(\mathbf{x}, \mathbf{y})$. Tenemos pues que

$$A^c = \{i \mid x_i = y_i\} = \{i \mid x_i = y_i = z_i\} \bigcup \{i \mid x_i = y_i \neq z_i\}.$$

Luego, el cardinal de este conjunto es

$$\#A^c \geq \#\{i \mid x_i = y_i = z_i\} = \#\left(\{i \mid x_i = z_i\} \bigcap \{i \mid v_i = z_i\}\right).$$

Entonces, tomando el complementario tenemos que

$$\begin{gathered}\#A \leq \#\left(\{i \mid x_i = z_i\}^c \bigcup \{i \mid v_i = z_i\}^c\right) = \#\left(\{i \mid x_i \neq z_i\} \bigcup \{i \mid v_i \neq z_i\}\right) \leq \\ \leq \#\{i \mid x_i \neq z_i\} + \#\{i \mid v_i \neq z_i\}.\end{gathered}$$

Es decir, $d_H(\mathbf{x}, \mathbf{y}) \leq d_H(\mathbf{x}, \mathbf{z}) + d_H(\mathbf{z}, \mathbf{y})$.

□

Definición 1.24. *Sea $x \in \mathbb{F}_q^n$. Definimos el **peso de Hamming** de x como*

$$w_H(x) = \#\{i \mid x_i \neq 0\}.$$

Lema 1.25. *Sean $x, y \in \mathbb{F}_q^n$. Se tiene que $d_H(x, y) = w_H(x - y)$.*

Demostración. Sea $d_H(x, y) = \#\{i \mid x_i \neq y_i\} = s$, luego, el vector $x - y$ tendrá s elementos distintos de cero, es decir, $w_H(x - y) = s$.

□

[5] Sean A, B conjuntos. Entonces se tiene:

- $\#\left(A \cup B\right) \leq \#A + \#B$.
- $\#A \leq \#\left(A \cup B\right)$.
- Si $\#A \leq \#B$ entonces $\#A^c \geq \#B^c$.

Definición 1.26. *Sea $\mathcal{C} \subseteq \mathbb{F}_q^n$ un código. Definimos la **distancia mínima de** $\mathcal{C}$ como*

$$d_H(\mathcal{C}) = \min_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{C} \\ \mathbf{x}\neq\mathbf{y}}} d_H(\mathbf{x},\mathbf{y})$$

*y el **peso mínimo de** $\mathcal{C}$ como*

$$w_H(\mathcal{C}) = \min_{\substack{\mathbf{c}\in\mathcal{C} \\ c\neq 0}} w_H(c).$$

1.2.2. Códigos Lineales

Si el alfabeto $\mathcal{A}$ es un cuerpo finito $\mathbb{F}_q$, entonces $\mathbb{F}_q^n$ es un espacio vectorial. Esto nos permite trabajar con códigos $\mathcal{C} \subseteq \mathbb{F}_q^n$ subespacios vectoriales, que tienen estructura algebraica, y que, por tanto, tienen más propiedades que los códigos arbitrarios. Este tipo de códigos, llamados **códigos lineales**, son los más utilizados en la vida cotidiana. En esta sección estudiaremos alguna de sus propiedades.

Definición 1.27. *Un **código lineal** $\mathcal{C} \subseteq \mathbb{F}_q^n$ es un subespacio vectorial de $\mathbb{F}_q^n$. Diremos que la dimensión del código lineal es la dimensión que tiene como subespacio vectorial de $\mathbb{F}_q^n$.*
Si $\mathcal{C} \subseteq \mathbb{F}_q^n$ es un código lineal con $dim(\mathcal{C}) = k$, diremos que $\mathcal{C}$ es un $[n,k]_q$-código. Si además conocemos su distancia mínima $d_H(\mathcal{C}) = d$, diremos que $\mathcal{C}$ es un $[n,k,d]_q$-código.

Observación 1.28. De ahora en adelante, utilizaremos la siguiente notación. Sea $\mathcal{C}$ un $[n,k,d]_q$-código. Entonces $n(\mathcal{C}) = n$, $k(\mathcal{C}) = k$ y $d(\mathcal{C}) = d$.

Observación 1.29. Dado que $\mathcal{C}$ es un subespacio vectorial de $\mathbb{F}_q^n$ de dimensión k, existe una base $\mathcal{B} = \{g_1, \ldots, g_k\}$ de $\mathcal{C}$ con $g_i = (g_{i1}, \ldots, g_{in}) \in \mathbb{F}_q^n$.
Podemos definir la siguiente matriz:

$$G = \begin{pmatrix} g_1 \\ \vdots \\ g_k \end{pmatrix} = \begin{pmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{k1} & \cdots & g_{kn} \end{pmatrix} \in \mathbb{F}_{q^{k\times n}}.$$

Tenemos que cualquier palabra $\mathbf{c} \in \mathcal{C}$ puede ser expresada como
$c = \lambda_1 g_1 + \cdots + \lambda_k g_k = (\lambda_1, \ldots, \lambda_k) \cdot G$, con $\lambda_i \in \mathbb{F}_q$.
Esto nos permite definir un proceso de codificación a partir de una aplicación lineal:

$$\begin{aligned} Enc : \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ \mathbf{m} &\longrightarrow \mathbf{m} \cdot G = \mathbf{c} \in \mathcal{C}. \end{aligned}$$

Definición 1.30. *Sea $\mathcal{C}$ un código lineal. Diremos que una matriz $G \in \mathbb{F}_q^{k\times n}$ es una **matriz generatriz de** $\mathcal{C}$ si las filas de G forman una base de $\mathcal{C}$.*

Observación 1.31. Es importante apreciar que la matriz generatriz no es única, dado que el concepto de base de un espacio vectorial no es único.
Si las k primeras columnas de G son independientes, podemos aplicar operaciones

elementales por filas hasta conseguir $G = (I_k \mid A)$, donde I_k denota la matriz identidad de tamaño k. A toda matriz generatriz con esta forma se la denota por **matriz generatriz en forma sistémica**.

Existen dos formas de describir un subespacio vectorial: explícitamente, dando una base; o implícitamente, como el conjunto de soluciones de un sistema homogéneo. Por tanto, habrán dos maneras de describir un código lineal. La forma explícita viene dada por la matriz generatriz G, y la forma implícita motiva la siguiente definición.

Definición 1.32. *Sea $H \in \mathbb{F}_q^{(n-k)\times n}$. Diremos que H es una **matriz de paridad** de un $[n,k]_q$-código $\mathcal{C}$ si $\mathcal{C} = \{\mathbf{x} \in \mathbb{F}_q^n : H\mathbf{x}^T = 0\}$.*

Observación 1.33. La matriz de paridad nos permitirá comprobar si una palabra pertenece al código $\mathcal{C}$, dado que, si $\mathbf{c} \in \mathcal{C}$, entonces $H\mathbf{c}^T = 0$. En caso contrario $H\mathbf{c}^T \neq 0$.

Proposición 1.34. *Sea $\mathcal{C}$ un $[n,k]_q$-código. Sea $G \in \mathbb{F}_q^{k\times n}$ y $H \in \mathbb{F}_q^{(n-k)\times n}$, con $rang(G) = k$ y $rang(H) = n-k$. Entonces, G es una matriz generatriz de $\mathcal{C}$ y H es una matriz de paridad de $\mathcal{C}$ si y solo si $GH^T = 0$.*

Demostración. Sabemos que, para cualquier $\mathbf{c} \in \mathcal{C}$, se tiene que $H\mathbf{c}^T = 0$, con $\mathbf{c} = \mathbf{m}G$. Luego, $H(\mathbf{m}G)^T = HG^T\mathbf{m}^T = 0$, $\forall \mathbf{m} \in \mathbb{F}_q^k$. De donde se deduce que $HG^T = 0$.
Ahora, supongamos que G es una matriz generatriz de un código $\mathcal{C}_1$ con $dim(\mathcal{C}_1) = k$ y H es una matriz de paridad de un código $\mathcal{C}_2$ con $rang(H) = n-k$. Por hipótesis tenemos que $GH^T = 0$; es decir, para cualquier $\mathbf{c} = \mathbf{m}G \in \mathcal{C}_1$, se tiene que $H\mathbf{c}^T = H(\mathbf{m}G)^T = HG^T\mathbf{m}^T = 0$. Luego $\mathcal{C}_1 \subseteq \mathcal{C}_2$; y como $dim(\mathcal{C}_1) = dim(\mathcal{C}_2)$, se deduce que $\mathcal{C}_1 = \mathcal{C}_2$.

□

Proposición 1.35. *$(I_k \mid P) \in \mathbb{F}_q^{k\times n}$ es una matriz generatriz de $\mathcal{C}$ si y solo si $\left(-P^T \mid I_{n-k}\right) \in \mathbb{F}_q^{(n-k)\times n}$ es una matriz de paridad de $\mathcal{C}$.*

Demostración. Suponiendo que $G = (I_k \mid P)$ es una matriz generatriz de $\mathcal{C}$, es fácil ver que $G \cdot \left(-P^T \mid I_{n-k}\right)^T = 0$, con $H = \left(-P^T \mid I_{n-k}\right) \in \mathbb{F}_q^{(n-k)\times n}$. Luego H es una matriz de paridad de $\mathcal{C}$ por la Proposición 1.34. Se procede de igual forma para demostrar la otra implicación.

□

Proposición 1.36. *Sea $\mathcal{C}$ un código lineal. Entonces, $w_H(\mathcal{C}) = d(\mathcal{C})$.*

Demostración. Tenemos que $d(\mathcal{C}) = \min\limits_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{C}\\ \mathbf{x}\neq\mathbf{y}}} d_H(\mathbf{x},\mathbf{y})$. Es decir, existen $\mathbf{x},\mathbf{y} \in \mathcal{C}$ tales que $d(\mathcal{C}) = d_H(\mathbf{x},\mathbf{y}) = w_H(\mathbf{x}-\mathbf{y}) \geq w_H(\mathcal{C}) = \min\limits_{\substack{\mathbf{c}\in\mathcal{C}\\ \mathbf{c}\neq 0}} w_H(\mathbf{c})$, donde la segunda igualdad se debe al Lema 1.25, y a que $\mathbf{x}-\mathbf{y} \in \mathcal{C}$ por ser $\mathcal{C}$ un subespacio vectorial. Además, tenemos que existe $\mathbf{c} \in \mathcal{C} \setminus \{0\}$ tal que $w_H(\mathcal{C}) = w_H(\mathbf{c}) = d_H(\mathbf{c},0) \geq d(\mathcal{C})$. Por tanto, $d(\mathcal{C}) = w_H(\mathcal{C})$.

□

Proposición 1.37. *Sea $H \in \mathbb{F}_q^{(n-k)\times n}$ una matriz de paridad de $\mathcal{C}$. Entonces $d(\mathcal{C}) = d$, donde d es el menor entero tal que d columnas de H son linealmente dependientes.*

Demostración. Sea $d = d(\mathcal{C})$ y r el menor entero tal que r columnas de H son linealmente dependientes. Sea $H = (h_1, \ldots, h_n)$, donde $h_i \in \mathbb{F}_q^{n-k}$ denota las columnas de H, con $i = 1, \ldots, n$.
Sea $\mathbf{c} \in \mathcal{C}$ tal que $w_H(\mathbf{c}) = d(\mathcal{C}) = d$. Por tanto, $supp(\mathbf{c}) = \{j_1, \ldots, j_d\}$. Luego, $Hc^T = h_{j_1}c_{j_1} + \ldots + h_{j_d}c_{j_d} = 0$, lo que quiere decir que las columnas $\{h_{j_1}, \ldots, h_{j_d}\}$ son linealmente dependientes. Luego $d = d(\mathcal{C}) \geq r$.
Consideremos ahora las columnas $\{h_{j_1}, \ldots, h_{j_r}\}$ linealmente dependientes de H, donde r es el menor número para el que se cumple la dependencia. Entonces, existen $a_1, \ldots, a_r \in \mathbb{F}_q$ (con algún $a_i \neq 0$), tales que $a_1h_{j_1} + \ldots + a_rh_{j_r} = 0$
Luego, si tomamos $\mathbf{c} \in \mathbb{F}_q^n$ tal que

$$c_i = \begin{cases} a_i & \text{para todo índice } i \in \{j_1, \ldots, j_r\} \\ 0 & \text{en caso contrario} \end{cases}$$

tendremos que $H\mathbf{c}^T = 0$. Luego $\mathbf{c} \in \mathcal{C}$. Por tanto, $w_H(\mathbf{c}) = r \geq d(\mathcal{C})$.
De las dos desigualdades concluimos que $r = d = d(\mathcal{C})$.

□

Corolario 1.38. ***Cota de Singleton.*** *Sea $\mathcal{C}$ es un $[n, k, d]_q$-código. Entonces $d - 1 \leq n - k$.*

Demostración. Sea H la matriz de paridad de $\mathcal{C}$. Tenemos que d es el menor número de columnas de H que son linealmente dependendientes. Por tanto, cualquieras $d - 1$ columnas de H son linealmente independientes. Como el rango de H, por ser matriz de paridad, es $n - k$, tendremos que necesariamente $n - k \geq d - 1$.

□

Definición 1.39. *Sea $\mathcal{C}$ un $[n, k, d]_q$-código. Diremos que $\mathcal{C}$ es MDS (en inglés, maximum distance separable) si alcanza la cota de Singleton; es decir, si $d(\mathcal{C}) = n - k + 1$.*

Proposición 1.40. *Sea $\mathcal{C}$ un $[n, k, d]_q$-código, y sea G una matriz generatriz de $\mathcal{C}$. Entonces, se tiene que $\mathcal{C}$ es MDS si y sólo si cualesquiera k columnas de G son linealmente independientes.*

Demostración. Supongamos que $\mathcal{C}$ es MDS. Sea G la matriz generatriz del código $\mathcal{C}$, y sea G' una matriz cuadrada formada por cualesquiera k columnas elegidas de la matriz G.
Consideremos ahora $\mathbf{x} \in \mathbb{F}_q^k$ tal que

$$\mathbf{x}G' = (x_1g_{11} + \cdots + x_kg_{k1}, \ldots, x_1g_{1k} + \cdots + x_kg_{kk}) = 0.$$

Entonces, la palabra del código $\mathbf{c} = \mathbf{x}G = (0, \ldots, 0, c_{k+1}, \ldots, c_n) \in \mathcal{C}$ tendrá a lo sumo $n - k$ posiciones distintas de 0. Luego $w_H(\mathbf{c}) \leq n - k$. Pero, como $\mathcal{C}$ es MDS, deducimos que $\mathbf{c} = 0$. Luego, dado que el rango de G es k, por ser matriz generatriz del código $\mathcal{C}$, tenemos que $\mathbf{x} = 0$. Por tanto las k columnas de G' son linealmente independientes.

Recíprocamente, asumamos ahora que cualesquiera k columnas de G son linealmente independientes. Sea $\mathbf{c} = (c_1, \ldots, c_n)$ una palabra de $\mathcal{C}$ tal que $c_i = 0$ en, como mínimo, k posiciones. Sea $\mathbf{c} = \mathbf{x}G$ para algún $\mathbf{x} \in \mathbb{F}_q^k$. Sea G' la matriz cuadrada formada por las k columnas de G que corresponden a las k posiciones donde $\mathbf{c}$ es igual a 0. Entonces $\mathbf{x}G' = 0$. Luego $\mathbf{x} = 0$, puesto que las k columnas de G' son linealmente independientes por hipótesis. Por tanto $\mathbf{c} = \mathbf{x}G = 0$. Esto implica que la distancia mínima de $\mathcal{C}$ es al menos $n - (k-1) = n - k + 1$. Y sabiendo que la cota de Singleton nos dice que $d(\mathcal{C}) \leq n - k + 1$, obtenemos que $\mathcal{C}$ es MDS.

□

Definición 1.41. *Diremos que una aplicación $\varphi : \mathbb{F}_{q^n} \longrightarrow \mathbb{F}_{q^n}$ es una **isometría** si deja la distancia de Hamming invariante, es decir, si*

$$d_H(\mathbf{x}, \mathbf{y}) = d_H(\varphi(\mathbf{x}), \varphi(\mathbf{y})) \; \forall \mathbf{x}, \mathbf{y} \in \mathbb{F}_{q^n}.$$

Definición 1.42. *Sean $\mathcal{C}$ y $\mathcal{D}$ dos códigos en $\mathbb{F}_{q^n}$. Diremos que $\mathcal{C}$ y $\mathcal{D}$ son **equivalentes** si existe una isometría φ sobre $\mathbb{F}_{q^n}$ tal que $\varphi(\mathcal{C}) = \mathcal{D}$.*

1.2.3. Decodificación en Códigos Lineales

El objetivo de Teoría de Códigos es construir códigos con una alta capacidad de corrección, eficientes, y con algoritmos de codificación y decodificación rápidos. En códigos lineales los algoritmos de codificación son aplicaciones lineales; luego consisten en multiplicar un vector por una matriz, algo que no requiere muchas operaciones. Sin embargo, no se conocen algoritmos de decodificación rápidos que funcionen para cualquier código lineal.
En esta sección introduciremos nociones básicas sobre la decodificación de códigos lineales.

Capacidad correctora de un Código

Definición 1.43. *Diremos que $\mathcal{C} \subset \mathbb{F}_q^n$ es t-corrector si, para cualesquiera $\mathbf{c_1}, \mathbf{c_2} \in \mathcal{C}$ palabras del código, y cualesquiera $\mathbf{e_1}, \mathbf{e_2} \in \mathbb{F}_q^n$ vectores de errores, con $w_H(\mathbf{e}_1) \leq t$ y $w_H(\mathbf{e}_2) \leq t$, se verifica que $\mathbf{c_1} + \mathbf{e_1} \neq \mathbf{c_2} + \mathbf{e_2}$.*
*Es decir, si el número de errores es a lo sumo t, entonces existe una **única** palabra del código $\mathbf{c} \in \mathcal{C}$ a distancia de Hamming menor o igual que t de la palabra recibida $\mathbf{y} \in \mathbb{F}_q^n$.*

Proposición 1.44. *Sea $\mathcal{C}$ un $[n, k, d]_q$-código. Entonces, se tiene que $\mathcal{C}$ es t-corrector si y solo si $t \leq \lfloor \frac{d-1}{2} \rfloor$.*

Demostración. Primero, partamos de la hipótesis de que $\mathcal{C}$ es t-corrector, y supongamos por reducción al absurdo que $d < 2t$.
Consideremos dos palabras $\mathbf{c_1}, \mathbf{c_2} \in \mathcal{C}$ tales que su distancia sea mínima, es decir,

$$d = d_H(\mathbf{c_1}, \mathbf{c_2}) = \#\{i : c_{1i} \neq c_{2i}\}.$$

Dividimos estos índices en dos conjuntos disjuntos:

$$d = \#\{i_1, \ldots, i_{N_1}\} \cup \{j_1, \ldots, j_{N_2}\}$$
$$\text{con } N_1, N_2 \leq t, N_1 + N_2 = d.$$

Definimos ahora un vector $\mathbf{z} \in \mathbb{F}_q^n$, que sea igual a $\mathbf{c}_1$ en las posiciones $i_1, \ldots, i_{N_1}$, igual a $\mathbf{c}_2$ en las posiciones $j_1, \ldots, j_{N_2}$, e igual a ambos en las posiciones donde $\mathbf{c_1} = \mathbf{c_2}$. Es decir,

$$z_i = \begin{cases} c_{1i}, & \forall i \notin \{j_1, \ldots, j_{N_2}\} \\ c_{2i}, & \forall i \in \{j_1, \ldots, j_{N_2}\} \end{cases}.$$

De esta forma, si definimos $\mathbf{e_1} = \mathbf{z} - \mathbf{c_1}$ y $\mathbf{e_2} = \mathbf{z} - \mathbf{c_2}$, tendríamos que se cumple $w_H(\mathbf{e_1}) = d_H(\mathbf{c_1}, \mathbf{z}) = N_2 \leq t$ y $w_H(\mathbf{e_2}) = d_H(\mathbf{c_2}, \mathbf{z}) = N_1 \leq t$, y sin embargo, $\mathbf{c_1} + \mathbf{e_1} = \mathbf{z} = \mathbf{c_2} + \mathbf{e_2}$. Luego, se contradice nuestra hipótesis de que $\mathcal{C}$ sea t-corrector.

Recíprocamente, partamos de la hipótesis que $t \leq \lfloor \frac{d-1}{2} \rfloor$. Supongamos, por reducción al absurdo que $\mathbf{c_1} + \mathbf{e_1} = \mathbf{c_2} + \mathbf{e_2}$ con $\mathbf{c_1}, \mathbf{c_2} \in \mathcal{C}, \mathbf{e_1}, \mathbf{e_2} \in \mathbb{F}_q^n$, donde $w_H(\mathbf{e_1}), w_H(\mathbf{e_2}) \leq t$.
Tendremos entonces que la palabra del código $\mathbf{c} = \mathbf{c_1} - \mathbf{c_2} = \mathbf{e_1} - \mathbf{e_2}$ tendrá peso $w_H(\mathbf{c}) = w_H(\mathbf{c_1} - \mathbf{c_2}) = w_H(\mathbf{e_1} - \mathbf{e_2}) \leq 2t < d - 1$. Luego, llegamos a un absurdo, ya que $d_H(\mathcal{C}) = d$. Luego, $\mathcal{C}$ es t-corrector.

□

Definición 1.45. *Diremos que* $t_{\mathcal{C}} = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$ *es la* ***capacidad correctora*** *del código* $\mathcal{C}$.

El problema de Decodificación

Sea $\mathcal{C}$ un $[n, k, d]_q$-código. Si $\mathbf{c} \in \mathcal{C}$ es la palabra del código que es enviada, e $\mathbf{y}$ es el vector recibido, entonces podemos definir

$$E = \{i \mid y_i \neq c_i\} \text{ como el conjunto de posiciones de error.}$$

Definiremos el vector error como $\mathbf{e} = \mathbf{y} - \mathbf{c}$. Se tiene entonces que $supp(\mathbf{e}) = E$, y $w_H(\mathbf{e})$ será el número de errores producidos al enviar la palabra $\mathbf{c} \in \mathcal{C}$.

Definición 1.46. *Sea* $\mathbf{c} \in \mathcal{C}$ *la palabra del código enviada en la comunicación, y sea* $\mathbf{y} = \mathbf{c} + \mathbf{e}$ *el vector recibido. Un* ***decodificador por mínimas distancias*** $\mathcal{D}$ *del código* $\mathcal{C}$*, que corrige* s *errores, se define como la aplicación:*

$$\mathcal{D} : \mathbb{F}_q^n \longrightarrow \mathcal{C} \cup \{?\}$$

donde $\mathcal{D}(\mathbf{y}) = \mathbf{c}$ *si existe una* ***única*** *palabra* $\mathbf{c} \in \mathcal{C}$ *tal que* $d_H(\mathbf{y}, \mathcal{C}) = \text{mín}\{d_H(\mathbf{y}, \mathbf{c}) \mid \mathbf{c} \in \mathcal{C}\} \leq s$.

En caso contrario, la respuesta del decodificador $\mathcal{D}$ dependerá del tipo que estemos usando:

(I) **Decodificador Tipo 1**:
Si $d_H(\mathbf{y}, \mathcal{C}) > s$, o bien si existen dos palabras, $\mathbf{c_1}, \mathbf{c_2} \in \mathcal{C}$ tales que $d_H(\mathbf{y}, \mathcal{C}) = d_H(\mathbf{y}, \mathbf{c_1}) = d_H(\mathbf{y}, \mathbf{c_2})$, el decodificador nos devuelve $\mathcal{D}(\mathbf{y}) = \{?\}$. En este caso es fácil detectar que se han producido errores.

(II) **Decodificador Tipo 2**:
Si existen dos palabras, $\mathbf{c_1}, \mathbf{c_2} \in \mathcal{C}$ que verifican que $d_H(\mathbf{y}, \mathcal{C}) = d_H(\mathbf{y}, \mathbf{c_1}) = d_H(\mathbf{y}, \mathbf{c_2})$, el decodificador nos devuelve una de ellas, $\mathcal{D}(\mathbf{y}) = \mathbf{c_i}$. Esta palabra no tiene que ser necesariamente la correcta. Este error no siempre se detecta.
Si $d_H(\mathbf{y}, \mathcal{C}) > s$, nos devuelve $\mathcal{D}(\mathbf{y}) = \{?\}$.

Observación 1.47. Si $s \leq t_{\mathcal{C}}$ entonces solo tendremos decodificadores de Tipo 1.

Proposición 1.48. *Todo código lineal $\mathcal{C}$ tiene un algoritmo de decodificación por mínimas distancias que corrige $t_{\mathcal{C}}$ errores.*

Demostración. Sea $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_q^n$ el vector recibido. Procedemos con el siguiente algoritmo:

1. Calculamos $d_H(\mathbf{y}, \mathbf{c})$, para todo $\mathbf{c} \in \mathcal{C}$
2. Devolvemos $\mathbf{c} \in \mathcal{C}$ que verifica que $d_H(\mathbf{y}, \mathbf{c}) = min\{d_H(\mathbf{y}, \mathbf{c}') \mid c' \in \mathcal{C}\}$

Por la Proposición 1.44, tenemos que $\mathcal{C}$ es $t_{\mathcal{C}}$-corrector. Por tanto, la palabra $\mathbf{c}$ que devuelve el algoritmo es **única**.

□

Observación 1.49. Hay que tener en cuenta que este algoritmo, que se conoce como **Búsqueda Exhaustiva**, es **teórico**. En la práctica, desde que el número de palabras sea elevado, el algoritmo no será eficiente. Esto se debe a que es un algoritmo de complejidad $\mathcal{O}(nq^k)$, porque debemos comparar un vector $\mathbf{y} \in \mathbb{F}_q^n$ de longitud n con las q^k palabras de $\mathcal{C}$.

En Teoría de Códigos, uno de los problemas que atrae el interés de investigadores es la búsqueda de algoritmos de decodificación eficientes para cualquier código lineal.
Sin embargo, hasta el momento todos los algoritmos que se han presentado y funcionan para cualquier código lineal tienen un coste computacional exponencial; sólo se han conseguido pequeñas mejoras del algoritmo de búsqueda exhaustiva. Es más, en [1] se demuestra que el problema de decodificación de códigos lineales está catalogado como un problema NP (se prueba que este problema es equivalente al problema de emparejamiento 3-dimensional, que es un problema NP).

Es por ello que los investigadores no tienen esperanzas de encontrar algoritmos de decodificación eficientes que funcionen para cualquier código lineal. En su lugar, se buscan familias de códigos lineales con algoritmos de decodificación rápidos.
En particular, en este trabajo hablaremos de dos de ellas: los Códigos Reed-Solomon y los Códigos Producto de Matrices.

2

Códigos Producto de Matrices

En este segundo capítulo vamos a construir códigos lineales más grandes a partir de códigos más pequeños. Inicialmente estudiaremos algunas construcciones particulares, que son muy utilizadas, y posteriormente estudiaremos una generalización de estos, la estructura **código producto de matrices**. Esta estructura es interesante ya que, si elegimos de cierta manera la matriz y los códigos lineales que la forman, tendremos una cota inferior a la distancia mínima del código obtenido.

2.1. Algunas Construcciones de códigos

En esta sección veremos algunos métodos clásicos para construir códigos a partir de otros ya conocidos.

2.1.1. Código Suma Directa

Definición 2.1. *Sean $\mathcal{C}_1$ un $[n_1, k_1]_q$-código, y $\mathcal{C}_2$ un $[n_2, k_2]_q$-código. Definimos el código* ***suma directa*** *$\mathcal{C}_1 \oplus \mathcal{C}_2$, también llamado construcción $(\mathbf{u} \mid \mathbf{v})$, como*

$$\mathcal{C}_1 \oplus \mathcal{C}_2 = \{(\mathbf{u} \mid \mathbf{v}) \mid \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}$$

donde $(\mathbf{u} \mid \mathbf{v})$ denota la palabra $(u_1, \ldots, u_{n_1}, v_1, \ldots, v_{n_2})$, con $\mathbf{u} = (u_1, \ldots, u_{n_1}) \in \mathcal{C}_1$ y $\mathbf{v} = (v_1, \ldots, v_{n_2}) \in \mathcal{C}_2$.

Proposición 2.2. *Sea $\mathcal{C}_i$ un $[n_i, k_i, d_i]$-código, con matriz generatriz $G_i \in \mathbb{F}_q^{k_i \times n_i}$ y matriz de paridad $H_i \in \mathbb{F}_q^{(n_i - k_i) \times n_i}$, para $i \in \{1, 2\}$. Entonces, $\mathcal{C}_1 \oplus \mathcal{C}_2$ es un $[n_1 + n_2, k_1 + k_2, min\{d_1, d_2\}]_q$-código, con las siguientes matrices generatriz y de paridad, respectivamente*

$$G = \left(\begin{array}{c|c} G_1 & 0 \\ \hline 0 & G_2 \end{array}\right) \in \mathbb{F}_q^{(k_1+k_2)\times(n_1+n_2)} \quad H = \left(\begin{array}{c|c} H_1 & 0 \\ \hline 0 & H_2 \end{array}\right) \in \mathbb{F}_q^{(n_1+n_2-k_1-k_2)\times(n_1+n_2)}$$

Demostración. Sea $\{x_1, \ldots, x_{k_1}\}, \{y_1, \ldots, y_{k_2}\}$ las bases de los códigos $\mathcal{C}_1$ y $\mathcal{C}_2$, respectivamente. Por tanto, $\{(x_1 \mid 0), \ldots, (x_{k_1} \mid 0), (0 \mid y_1), \ldots, (0 \mid y_{k_2})\}$ es base del código $\mathcal{C}_1 \oplus \mathcal{C}_2$. Por tanto, $\mathcal{C}_1 \oplus \mathcal{C}_2$ es un $[n_1 + n_2, k_1 + k_2]_q$-código con G como una matriz generatriz. Además, como $G \cdot H^T = 0$, se tiene por la Proposición 1.34 que H es una matriz de paridad de $\mathcal{C}_1 \oplus \mathcal{C}_2$.
Veamos ahora cuál es la distancia mínima del código. Consideremos $\mathbf{u}, \mathbf{u}' \in \mathcal{C}_1$ y

$\mathbf{v}, \mathbf{v}' \in \mathcal{C}_2$, tales que $d_H(\mathbf{u}, \mathbf{u}') = d_1$ y $d_H(\mathbf{v}, \mathbf{v}') = d_2$. Tendremos entonces que $d_H((\mathbf{u} \mid 0), (\mathbf{u}' \mid 0)) = d_1$ y $d_H((0 \mid \mathbf{v}), (0 \mid \mathbf{v}')) = d_2$. Por tanto $d(\mathcal{C}_1 \oplus \mathcal{C}_2) = \min\{d_1, d_2\}$ será la distancia mínima de $\mathcal{C}_1 \oplus \mathcal{C}_2$.

□

2.1.2. Construcción de Plotkin

Definición 2.3. *Sea $\mathcal{C}_i$ un $[n, k_i, d_i]_q$-código, con matriz generatriz $G_i \in \mathbb{F}_q^{k_i \times n}$ y matriz de paridad $H_i \in \mathbb{F}_q^{(n-k_i)\times n}$, para $i \in \{1, 2\}$. Definimos la **construcción de Plotkin** como el código*

$$\mathcal{C} = \{(\mathbf{u} \mid \mathbf{u} + \mathbf{v}) \mid \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}$$

Teorema 2.4. *La construcción de Plotkin de $\mathcal{C}_1$ y $\mathcal{C}_2$ es un $[2n, k_1 + k_2, d = \min\{2d_1, d_2\}]_q$-código con las siguientes matrices generatriz y de paridad, respectivamente*

$$G = \left(\begin{array}{c|c} G_1 & G_1 \\ \hline 0 & G_2 \end{array}\right) \in \mathbb{F}_q^{(k_1+k_2)\times 2n} \quad H = \left(\begin{array}{c|c} H_1 & -H_2 \\ \hline 0 & H_2 \end{array}\right) \in \mathbb{F}_q^{(2n-k_1-k_2)\times 2n}$$

Demostración. Sean $\{x_1, \ldots, x_{k_1}\}, \{y_1, \ldots, y_{k_2}\}$ bases de los códigos $\mathcal{C}_1$ y $\mathcal{C}_2$, respectivamente. Entonces, cualquier palabra de la forma $(u \mid u + v)$ es combinación lineal de $\{(x_1 \mid x_1), \ldots, (x_{k_1} \mid x_{k_1}), (0 \mid y_1), \ldots, (0 \mid y_{k_2})\}$, que al ser linealmente independientes, son una base del código de Plotkin. Por tanto, el código de Plotkin $\mathcal{C}$ es un $[2n, k_1 + k_2]_q$-código con una matriz generatriz G y una matriz de paridad H (por la Proposición 1.34, ya que $GH^T = 0$). Veamos ahora cuál es la distancia mínima d de $\mathcal{C}$. Para cualquier palabra del código $(\mathbf{x} \mid \mathbf{x} + \mathbf{y})$ tenemos que $w_H((\mathbf{x} \mid \mathbf{x} + \mathbf{y})) = w_H(\mathbf{x}) + w_H(\mathbf{x} + \mathbf{y})$. Si $\mathbf{y} = 0$, entonces $w_H(\mathbf{x} \mid \mathbf{x} + \mathbf{y}) = 2w_H(\mathbf{x}) \geq 2d_1$. Supongamos ahora que estamos en el caso $\mathbf{y} \neq 0$. Sabemos que

$$\begin{gathered} w_H(\mathbf{x} + \mathbf{y}) = w_H(\mathbf{x}) + w_H(\mathbf{y}) - \{j \in supp(\mathbf{y}) \cap supp(\mathbf{x}) \mid x_j + y_j = 0\} \geq \\ \geq w_H(\mathbf{x}) + w_H(\mathbf{y}) - w_H(\mathbf{x}) = w_H(\mathbf{y}) \end{gathered}$$

Tenemos por tanto que

$$w_H((\mathbf{x} \mid \mathbf{x} + \mathbf{y})) = w_H(\mathbf{x}) + w_H(\mathbf{x} + \mathbf{y}) \geq w_H(\mathbf{x}) + w_H(\mathbf{y}) \geq w_H(\mathbf{y}) \geq d_2$$

Por tanto, $d \geq \min\{2d_1, d_2\}$.
Ahora, consideremos $\mathbf{x_0} \in \mathcal{C}_1, \mathbf{y_0} \in \mathcal{C}_2$ tal que $w_H(\mathbf{x_0}) = d_1$, y $w_H(\mathbf{y_0}) = d_2$. Tenemos que $(\mathbf{x_0} \mid \mathbf{x_0})$ y $(0 \mid \mathbf{y}_0)$ serán palabras del código Plotkin $\mathcal{C}$ tales que $w_H((\mathbf{x_0} \mid \mathbf{x_0})) = 2d_1$ y $w_H((0 \mid \mathbf{y}_0)) = d_2$. Luego $d = \min\{2d_1, d_2\}$ es la distancia mínima del código de Plotkin $\mathcal{C}$.

□

2.1.3. Código $(u + v \mid u - v)$

Definición 2.5. *Sea $\mathcal{C}_i$ un $[n, k_i, d_i]_q$-código, con matriz generatriz $G_i \in \mathbb{F}_q^{k_i \times n}$ y matriz de paridad $H_i \in \mathbb{F}_q^{(n-k_i)\times n}$ para $i \in \{1, 2\}$. Definimos la **construcción** $(\mathbf{u} + \mathbf{v} \mid \mathbf{u} - \mathbf{v})$ como el código*

$$\mathcal{C} = \{(\mathbf{u} + \mathbf{v} \mid \mathbf{u} - \mathbf{v}) \mid \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}$$

Observación 2.6. Cuando consideramos esta construcción, asumiremos que q es impar, dado que si q es par, $\mathbf{u}+\mathbf{v} = \mathbf{u}-\mathbf{v}$.

Proposición 2.7. *Para q impar, sea $\mathcal{C}$ la construcción $(\mathbf{u}+\mathbf{v} \mid \mathbf{u}-\mathbf{v})$ que se obtiene de los códigos $\mathcal{C}_1$ y $\mathcal{C}_2$. Se tiene que $\mathcal{C}$ es un $[2n, k_1+k_2, d]_q$-código con distancia mínima $d \geq \{2d_1, 2d_2, \text{máx}\{d_1, d_2\}\}$, cuyas matrices generatriz y de paridad son las siguientes*

$$G = \left(\begin{array}{c|c} G_1 & G_1 \\ \hline G_2 & -G_2 \end{array}\right) \in \mathbb{F}_q^{(k_1+k_2)\times 2n} \quad H = \left(\begin{array}{c|c} H_1 & H_2 \\ \hline H_1 & -H_2 \end{array}\right) \in \mathbb{F}_q^{(2n-k_1-k_2)\times 2n}$$

Demostración. Supongamos que $\{x_1, \ldots, x_{k_1}\}, \{y_1, \ldots, y_{k_2}\}$ son bases de los códigos $\mathcal{C}_1$ y $\mathcal{C}_2$, respectivamente. Por tanto, cualquier palabra del código $\mathcal{C}$, que será de la forma $(\mathbf{u}+\mathbf{v} \mid \mathbf{u}-\mathbf{v})$, se puede escribir como $(\mathbf{u} \mid \mathbf{u}) + (\mathbf{v} \mid -\mathbf{v})$, donde $(\mathbf{u} \mid \mathbf{u})$ es una combinación lineal de $\{(x_1 \mid x_1), \ldots, (x_{k_1} \mid x_{k_1})\}$, y $(\mathbf{v} \mid -\mathbf{v})$ es una combinación lineal de $\{(y_1 \mid -y_1), \ldots, (y_{k_2} \mid -y_{k_2})\}$. Por tanto, debemos probar ahora que el conjunto de vectores $\{(x_i \mid x_i), (y_j \mid -y_j)\}_{\substack{i\in\{1,\ldots,k_1\}\\ j\in\{1,\ldots,k_2\}}}$ son linealmente independientes. Supongamos que existen λ_i, μ_j tal que

$$\sum_i \lambda_i (x_i \mid x_i) + \sum_j \mu_j (y_j \mid -y_j) = 0$$

Entonces se tiene que $\sum_i \lambda_i x_i + \sum_j \mu_j y_j = 0$ y $\sum_i \lambda_i x_i - \sum_j \mu_j y_j = 0$. Sumando estas dos ecuaciones obtenemos que $\sum_i \lambda_i x_i = 0$. Como $\{x_1, \ldots, x_{k_1}\}$ es base de $\mathcal{C}_1$, se deduce que $\lambda_i = 0, \forall i \in \{1, \ldots, k_1\}$. Análogamente, restando las ecuaciones obtenemos que $\sum_j \mu_j y_j = 0$. Como $\{y_1, \ldots, y_{k_2}\}$ es base de $\mathcal{C}_2$, se deduce que $\mu_j = 0, \forall j \in \{1, \ldots, k_2\}$.
Por tanto, G es una matriz generatriz del código $\mathcal{C}$. Además, por la Proposición 1.34, que H es una matriz de paridad, pues cumple $GH^T = 0$.
Veamos ahora cual es la distancia mínima d del código $\mathcal{C}$. Sea $(\mathbf{u}+\mathbf{v} \mid \mathbf{u}-\mathbf{v}) \neq 0$ una palabra del código $\mathcal{C}$. Si $\mathbf{v} = 0$, tendremos que $w_H((\mathbf{u}+\mathbf{v} \mid \mathbf{u}-\mathbf{v})) = w_H((\mathbf{u} \mid \mathbf{u})) = 2d_1$; y si $\mathbf{u} = 0$, tendremos que $w_H((\mathbf{u}+\mathbf{v} \mid \mathbf{u}-\mathbf{v})) = w_H((\mathbf{v} \mid -\mathbf{v})) = 2d_2$. Supongamos entonces que $\mathbf{u} \neq 0$ y $\mathbf{v} \neq 0$. Entonces,

$$\begin{gathered} w_H(\mathbf{u}-\mathbf{v}) = \#\{i \mid u_i - v_i \neq 0\} = \#\{i \mid u_i \neq v_i\} \geq \\ \geq \#\{i \mid u_i \neq 0\} - \#\{i \mid u_i = v_i\} = w_H(u) - r \end{gathered}$$

donde r es el número de posiciones i donde $u_i = v_i \neq 0$.
Si $u_i = v_i \neq 0$, entonces $u_i + v_i = 2u_i \neq 0$ (porque supusimos que q era impar). Luego, tendremos que $w_H(\mathbf{u}+\mathbf{v}) \geq r$. Por tanto,

$$w_H((\mathbf{u}+\mathbf{v} \mid \mathbf{u}-\mathbf{v})) = w_H((\mathbf{u}+\mathbf{v})) + w_H((\mathbf{u}-\mathbf{v})) \geq r + (w_H(\mathbf{u}) - r) = w_H(\mathbf{u}) \geq d_1$$

De la misma forma, tenemos que

$$w_H(\mathbf{u}-\mathbf{v}) \geq \#\{i \mid v_i \neq 0\} - \#\{i \mid u_i = v_i\} = w_H(\mathbf{v}) - r.$$

Luego, $w_H((\mathbf{u}+\mathbf{v} \mid \mathbf{u}-\mathbf{v})) = w_H((\mathbf{u}+\mathbf{v})) + w_H((\mathbf{u}-\mathbf{v})) \geq r + (w_H(\mathbf{v}) - r) = w_H(\mathbf{v}) \geq d_2$.
Concluimos pues que la distancia mínima del código $\mathcal{C}$ es $d \geq \{2d_1, 2d_2, \text{máx}\{d_1, d_2\}\}$.

□

2.2. Código Producto de Matrices

Definición 2.8. *Sea $A = (a_{ij}) \in \mathbb{F}_q^{M\times N}$ una matriz y sean $\mathcal{C}_1, \ldots, \mathcal{C}_M$ códigos de longitud n sobre $\mathbb{F}_q$. Definimos el **código producto de matrices** $\mathcal{C} = [\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A$ como el conjunto de todas las palabras formadas como los productos de matrices $\mathbf{c} = [\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A$, donde $\mathbf{c_i} \in \mathcal{C}_i$ es una palabra representada como un vector columna $n \times 1$.*
Las palabras de este código serán por tanto matrices $n \times N$ en $\mathbb{F}_q$:

$$\mathbf{c} = [\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A = \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1M} \\ \vdots & \ddots & & \vdots \\ c_{n1} & c_{n2} & \ldots & c_{nM} \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ \vdots & \ddots & & \vdots \\ a_{M1} & a_{M2} & \ldots & a_{MN} \end{pmatrix} =$$

$$= \begin{pmatrix} c_{11}a_{11} + c_{12}a_{21} + \ldots + c_{1M}a_{M1} & \ldots & c_{11}a_{1N} + \ldots + c_{1M}a_{MN} \\ \vdots & \ddots & \vdots \\ c_{n1}a_{11} + c_{n2}a_{21} + \ldots + c_{nM}a_{M1} & \ldots & c_{n1}a_{1N} + \ldots + c_{nM}a_{MN} \end{pmatrix}$$

Observación 2.9. Tenemos que la j-ésima columna de una palabra del código se puede escribir como $\sum_{i=1}^{M} c_{ji}a_{ij}$. Por tanto, tenemos que las palabras del código se pueden expresar como vectores de la siguiente forma:

$$\mathbf{c} = \left(\sum_{i=1}^{M} \mathbf{c_i} a_{i1}, \ldots, \sum_{i=1}^{M} \mathbf{c_i} a_{iN} \right) \in \mathbb{F}_q^{nN}$$

donde $\mathbf{c_i} = (c_{1i}, \ldots, c_{ni})^T \in \mathcal{C}_i$.
Por abuso de notación, vamos a mezclar ambas notaciones. Según nos convenga, utilizaremos la notación en forma de matriz, o en forma de vectores.

Ejemplo 2.10. El código producto de matrices utilizando la matriz $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ nos da la construcción de Plotkin (Definición 2.3), y utilizando $A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ nos da la construcción $(\mathbf{u}+\mathbf{v}|\mathbf{u}-\mathbf{v})$ (Definición 2.5).

Definición 2.11. *Sea $G_i \in \mathbb{F}_q^{k_i \times n}$ una matriz generatriz del código $\mathcal{C}_i$, con $i \in \{1, \ldots, M\}$. Entonces, una matriz generatriz del código producto de matrices $\mathcal{C} = [\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A$ es:*

$$G = \begin{pmatrix} G_1 a_{11} & \ldots & G_1 a_{1N} \\ \vdots & \ddots & \vdots \\ G_M a_{M1} & \ldots & G_M a_{MN} \end{pmatrix} \in \mathbb{F}_q^{(k_1+\cdots+k_M)\times nN}.$$

Observación 2.12. Tenemos por tanto que $dim([\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A) \leq k_1 + \ldots + k_M$, donde $k_i = dim(\mathcal{C}_i)$ para $i \in \{1, \ldots, M\}$. La igualdad entre dimensiones solo se dará si A tiene rango máximo.

Ejemplo 2.13. En $\mathbb{F}_2$, consideremos los siguientes códigos lineales de longitud $n = 3$.

- Sea $\mathcal{C}_1$ el código generado por el vector $(1,1,1)$. Luego $G_1 = (1,1,1)$ y $dim(\mathcal{C}_1) = 1$.
  $\mathcal{C}_1 = < (1,1,1) > = \{(0,0,0), (1,1,1)\}$.
- Sea $\mathcal{C}_2$ el código generado por el vector $(1,0,0)$. Luego $G_2 = (1,0,0)$ y $dim(\mathcal{C}_2) = 1$.
  $\mathcal{C}_2 = < (1,0,0) > = \{(0,0,0), (1,0,0)\}$.
- Sea $\mathcal{C}_3$ el código generado por el vector $(1,0,1)$ y $(0,1,1)$. Luego $G_3 = \begin{pmatrix} 1\,0\,1 \\ 0\,1\,1 \end{pmatrix}$ y $dim(\mathcal{C}_3) = 2$.
  $\mathcal{C}_2 = < (1,0,1), (0,1,1) > = \{(0,0,0), (1,0,1), (0,1,1), (1,1,0)\}$.

Y consideramos la matriz $A = \begin{pmatrix} 1\,1\,1 \\ 0\,1\,1 \\ 0\,0\,1 \end{pmatrix} \in \mathbb{F}_2^{3\times 3}$

Entonces tenemos que nuestro código producto de matrices $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3] \cdot A$ tendrá 16 palabras, de la forma $\mathbf{c} = [\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}] \cdot A$ (donde $\mathbf{c_1} \in \mathcal{C}_1$, $\mathbf{c_2} \in \mathcal{C}_2$, y $\mathbf{c_3} \in \mathcal{C}_3$). Estas palabras serán de longitud $n \times N = 3 \times 3 = 9$. Calculemos algunas palabras de $\mathcal{C}$.

- Si $\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}$ son las palabras nulas, obtenemos la palabra nula de $\mathcal{C}$,
$$\mathbf{0} = \begin{pmatrix} 0\,0\,0 \\ 0\,0\,0 \\ 0\,0\,0 \end{pmatrix} \cdot A = (0,0,0|0,0,0|0,0,0).$$
- Si $\mathbf{c_1} = (0,0,0), \mathbf{c_2} = (1,0,0), \mathbf{c_3} = (1,0,1)$, obtenemos la palabra
$$\mathbf{c} = \begin{pmatrix} 0\,1\,1 \\ 0\,0\,0 \\ 0\,0\,1 \end{pmatrix} \cdot A = (0,0,0|1,0,0|0,0,1).$$

Y tenemos, por la Definición 2.11, que una matriz generatriz del código producto es:

$$G = \left(\begin{array}{c|c|c} 1\,1\,1 & 1\,1\,1 & 1\,1\,1 \\ \hline 0\,0\,0 & 1\,0\,0 & 1\,0\,0 \\ \hline 0\,0\,0 & 0\,0\,0 & 1\,0\,1 \\ 0\,0\,0 & 0\,0\,0 & 0\,1\,1 \end{array}\right).$$

Por la observación 2.12, como A es de rango máximo, se tiene que

$$dim(\mathcal{C}) = dim([\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3] \cdot A) = dim(\mathcal{C}_1) + dim(\mathcal{C}_2) + dim(\mathcal{C}_3) = 4.$$

Proposición 2.14. *Sean los códigos lineales $\mathcal{C}_1, \ldots, \mathcal{C}_M \subseteq \mathbb{F}_q^n$ y una matriz $A \in \mathbb{F}_q^{M\times N}$. Entonces, se tiene que:*

(1) Si A_Π es un matriz obtenida al realizar una permutación Π de las filas de la matriz A, entonces:

$$[\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A = \left[\mathcal{C}_{\Pi(1)}, \ldots, \mathcal{C}_{\Pi(M)}\right] \cdot A_\Pi.$$

(II) *Si A_Π es un matriz obtenida al realizar una permutación Π de las columnas de la matriz A, entonces:*

$$[\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A \text{ es un código equivalente a } [\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A_\Pi.$$

Demostración. (I) Sea $A = \left(a_{ij}\right) \in \mathbb{F}_q^{M\times N}$. Entonces la matriz A permutada por filas será $A_\Pi = (a_{\Pi(i)j})$. Luego, una palabra del código $\left[\mathcal{C}_{\Pi(1)}, \ldots, \mathcal{C}_{\Pi(M)}\right] \cdot A_\Pi$ será de la forma:

$$\left[\mathbf{c_{\Pi(1)}}, \ldots, \mathbf{c_{\Pi(M)}}\right] \cdot A_\Pi =$$

$$= \begin{pmatrix} c_{1\Pi(1)}a_{\Pi(1)1} + \ldots + c_{1\Pi(M)}a_{\Pi(M)1} & \cdots & c_{1\Pi(1)}a_{\Pi(1)N} + \ldots + c_{1\Pi(M)}a_{\Pi(M)N} \\ \vdots & \ddots & \vdots \\ c_{n\Pi(1)}a_{\Pi(1)1} + \ldots + c_{n\Pi(M)}a_{\Pi(M)1} & \cdots & c_{n\Pi(1)}a_{\Pi(1)N} + \ldots + c_{n\Pi(M)}a_{\Pi(M)N} \end{pmatrix} =$$

$$= \begin{pmatrix} c_{11}a_{11} + \ldots + c_{1M}a_{M1} & \cdots & c_{11}a_{1N} + \ldots + c_{1M}a_{MN} \\ \vdots & \ddots & \vdots \\ c_{n1}a_{11} + \ldots + c_{nM}a_{M1} & \cdots & c_{n1}a_{1N} + \ldots + c_{nM}a_{MN} \end{pmatrix} =$$

$$= [\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A$$

Por tanto, tenemos que $\left[\mathcal{C}_{\Pi(1)}, \ldots, \mathcal{C}_{\Pi(M)}\right] \cdot A_\Pi = [\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A$.

(II) Sea $A = \left(a_{ij}\right) \in \mathbb{F}_q^{M\times N}$. Entonces A permutada por columnas será $A_\Pi = (a_{i\Pi(j)})$. Luego, una palabra del código $[\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A_\Pi$ será de la forma:

$$[\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A_\Pi =$$

$$= \begin{pmatrix} c_{11}a_{1\Pi(1)} + \ldots + c_{1M}a_{M\Pi(1)} & \cdots & c_{11}a_{1\Pi(N)} + \ldots + c_{1M}a_{M\Pi(N)} \\ \vdots & \ddots & \vdots \\ c_{n1}a_{1\Pi(1)} + \ldots + c_{nM}a_{M\Pi(1)} & \cdots & c_{n1}a_{1\Pi(N)} + \ldots + c_{nM}a_{M\Pi(N)} \end{pmatrix} =$$

$$= \left(\sum_{i=1}^{M} \mathbf{c_i}a_{i\Pi(1)}, \ldots, \sum_{i=1}^{M} \mathbf{c_i}a_{i\Pi(N)}\right) \text{ con } \mathbf{c_i} = (c_{1i}, \ldots, c_{ni})^T \in \mathcal{C}_i.$$

Por tanto, las palabras del código $[\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A_\Pi$ tienen exactamente los mismos elementos que las del código $[\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A$, la única diferencia es ordenación por columnas. Es decir, existe una aplicación $\varphi : \mathbb{F}_q^{n\times N} \longrightarrow \mathbb{F}_q^{n\times N}$ tal que:

$$\varphi\left(\sum_{i=1}^{M} \mathbf{c_i}a_{i1}, \ldots, \sum_{i=1}^{M} \mathbf{c_i}a_{iN}\right) = \left(\sum_{i=1}^{M} \mathbf{c_i}a_{i\Pi(1)}, \ldots, \sum_{i=1}^{M} \mathbf{c_i}a_{i\Pi(N)}\right)$$

donde la reordenación viene dada por la aplicación

$$\Pi : \{1, \ldots, M\} \longrightarrow \{1, \ldots, M\}.$$

Por tanto, tenemos que para $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^{n\times N}$, que sean palabras del código $[\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A$, se cumple que $d_H(\varphi(\mathbf{x}), \varphi(\mathbf{y})) = d_H(\mathbf{x}, \mathbf{y})$, dado que:

$$d_H(\mathbf{x},\mathbf{y}) = w(\mathbf{x}-\mathbf{y}) =$$
$$w\left(\left(\sum_{i=1}^{M}\mathbf{c_i}a_{i1},\ldots,\sum_{i=1}^{M}\mathbf{c_i}a_{iN}\right)-\left(\sum_{i=1}^{M}\mathbf{c'_i}a_{i1},\ldots,\sum_{i=1}^{M}\mathbf{c'_i}a_{iN}\right)\right) =$$
$$= w\left(\left(\sum_{i=1}^{M}\mathbf{c_i}a_{i\Pi(1)},\ldots,\sum_{i=1}^{M}\mathbf{c_i}a_{i\Pi(N)}\right)-\left(\sum_{i=1}^{M}\mathbf{c'_i}a_{i\Pi(1)},\ldots,\sum_{i=1}^{M}\mathbf{c'_i}a_{i\Pi(N)}\right)\right) =$$
$$= w_H(\varphi(\mathbf{x})-\varphi(\mathbf{y})) = d_H(\varphi(\mathbf{x}),\varphi(\mathbf{y})).$$

Por tanto, siguiendo la Definición 1.42, tenemos que $[\mathcal{C}_1,\ldots,\mathcal{C}_M]\cdot A$ es un código equivalente a $[\mathcal{C}_1,\ldots,\mathcal{C}_M]\cdot A_\Pi$.

□

Definición 2.15. *Dada la matriz $A \in \mathbb{F}_q^{M\times N}$, una inversa a la derecha de A será una matriz A^{-1} tal que $A\cdot A^{-1} = I_M$ (Para esto es necesario que $M \leq N$). Diremos entonces que A es* ***no singular****.*

Definición 2.16. *Denotamos por $A(j_1,\ldots,j_t)$ a la matriz cuadrada de tamaño t formada a partir de las primeras t filas de A y las columnas $j_1,\ldots,j_t$ con $1 \leq j_1 < \ldots < j_t \leq N$ y $1 \leq t \leq M$.*

Observación 2.17. Si $A(1,\ldots,M) \in \mathbb{F}_q^{M\times M}$ es no singular entonces $\left(A(1,\ldots,M)^{-1} \mid 0\right)$ es la matriz inversa de A por la derecha.

Proposición 2.18. *Sean $A \in \mathbb{F}_q^{M\times N}$ y $\mathcal{C}_1,\ldots,\mathcal{C}_M$ códigos lineales de longitud n en $\mathbb{F}_q$. Definimos el código producto de matrices $\mathcal{C} = [\mathcal{C}_1,\ldots,\mathcal{C}_M]\cdot A$. Entonces, si A tiene una submatriz formada por M columnas que es no singular, tenemos*

$$\#\mathcal{C} = \#[\mathcal{C}_1,\ldots,\mathcal{C}_M]\cdot A = \#\mathcal{C}_1\cdots\#\mathcal{C}_M$$

Demostración. Para ver que los cardinales son iguales, probaremos que existe una biyección entre $\mathcal{C}_1\times\ldots\times\mathcal{C}_M$ y $[\mathcal{C}_1,\ldots,\mathcal{C}_M]\cdot A$. Consideremos la aplicación:

$$\begin{aligned}\varphi : \mathcal{C}_1\times\ldots\times\mathcal{C}_M &\longrightarrow [\mathcal{C}_1,\ldots,\mathcal{C}_M]\cdot A\\ (\mathbf{c_1},\ldots,\mathbf{c_M}) &\longrightarrow [\mathbf{c_1},\ldots,\mathbf{c_M}]\cdot A\end{aligned}$$

Por cómo está definida la aplicación, y por la definición de $\mathcal{C}$, se tiene que φ es sobreyectiva. Veamos ahora que φ es inyectiva.
Por hipótesis, sabemos que A tiene M columnas linealmente independientes, que no tienen que ser necesariamente las primeras. Sin embargo, por la Proposición 2.14 sabemos que, si consideramos una permutación que coloque las M columnas linealmente independientes las primeras, obtendremos un código equivalente a $\mathcal{C}$. Luego, sin pérdida de generalidad, podemos considerar que $A(1,\ldots,M)$ es no singular. Por tanto, la inversa de A por la derecha será $A^{-1} = \left(A(1,\ldots,M)^{-1} \mid 0\right)$.
Entonces, si tenemos dos elementos $\mathbf{c},\mathbf{c'} \in [\mathcal{C}_1,\ldots,\mathcal{C}_M]\cdot A$ iguales, tendremos que:

$$\begin{aligned}\mathbf{c} = [\mathbf{c_1},\ldots,\mathbf{c_M}]\cdot A &= [\mathbf{c'_1},\ldots,\mathbf{c'_M}]\cdot A = \mathbf{c'}\\ [\mathbf{c_1},\ldots,\mathbf{c_M}]\cdot A\cdot A^{-1} &= [\mathbf{c'_1},\ldots,\mathbf{c'_M}]\cdot A\cdot A^{-1}\\ [\mathbf{c_1},\ldots,\mathbf{c_M}] &= [\mathbf{c'_1},\ldots,\mathbf{c'_M}]\end{aligned}$$

Luego φ es inyectiva, y por tanto, biyectiva.

□

Definición 2.19. *Diremos que una matriz A es **no singular por columnas (NSC)** si $A(j_1, \ldots, j_t)$ es una matriz no singular para cada $1 \leq t \leq M$ y $1 \leq j_1 < \ldots < j_t \leq N$.*

Observación 2.20. Toda matriz NSC es también no singular, pero el recíproco no es cierto (por ejemplo, $A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ es no singular pero $A(1,2)$ es singular).

Observación 2.21. No existe ninguna matriz cuadrada de tamaño 3 en $\mathbb{F}_2$ que sea NSC.

Definición 2.22. *Diremos que un código producto de matrices es NSC si la matriz asociada al código es NSC.*

Ejemplo 2.23. Sea $\mathbb{F}_q = \{\alpha_1, \ldots, \alpha_q\}$. Para $1 \leq M \leq q$ definimos la matriz de Vandermonde:

$$V_M = \begin{pmatrix} 1 & \ldots & 1 \\ \alpha_1 & \ldots & \alpha_q \\ \vdots & \ddots & \vdots \\ \alpha_1^{M-1} & \ldots & \alpha_q^{M-1} \end{pmatrix}$$

V_M es una matriz NSC, ya que $V_M(j_1, \ldots, j_N$ es no singular para cualquier $1 \leq M \leq N \leq q$ y $1 \leq j_1 < \ldots < j_N \leq q$.

Por tanto, con este ejemplo hemos visto que existen matrices NSC de tamaño $M \times N$ sobre $\mathbb{F}_q$ para todo $1 \leq M \leq N \leq q$.

Observación 2.24. Es obvio que para $M = 1$, existe una matriz $A \in \mathbb{F}_q^{M \times N}$ que sea NSC para cualquier N (por ejemplo, el vector de tamaño $1 \times N$ donde todos sus elementos son 1). Veamos qué ocurre con las matrices NSC para $M \geq 2$, con $M \leq N$.

Proposición 2.25. *Para $M \geq 2$ existe una matriz NSC de tamaño $M \times N$ sobre $\mathbb{F}_q$ si y solo si $M \leq N \leq q$.*

Demostración. Por el Ejemplo 2.23 tenemos que si $1 \leq M \leq N \leq q$, existe una matriz NSC, la matriz de Vandermonde. Veamos ahora la otra implicación, por reducción al absurdo.
Sea A una matriz NSC de tamaño $2 \times N$ y supongamos que $N \geq q + 1$. Sabemos que la primera fila de A no tendrá ningún cero, y la segunda fila tendrá, a lo sumo, un cero (dado que si tuviera más, no sería NSC). Sin pérdida de generalidad, asumimos que $a_{22}, \ldots, a_{2N}$ son distintos de cero. Tenemos entonces que, para $2 \leq j \leq N$, $a_{1j} = \alpha^{r_j}$ y $a_{2j} = \alpha^{s_j}$, donde α es el elemento primitivo de $\mathbb{F}_q$.
Por hipótesis, tenemos que $\begin{vmatrix} a_{1j} & a_{1k} \\ a_{2j} & a_{2k} \end{vmatrix}$ será distinto de cero para $2 \leq j < k \leq N$, es decir;

$$r_j + s_k \not\equiv r_k + s_j \bmod (q-1) \text{ , para } 2 \leq j < k \leq N \qquad (*)$$

En particular, $r_2 \not\equiv r_k - s_k + s_2$ para cualquier k tal que $3 \leq k \leq N$. Por tanto, $r_k - s_k + s_2$ toma a lo sumo $q-2$ valores diferentes, módulo $(q-1)$ (ya que k toma al menos $q-1$ valores desde 3 hasta $N \geq q+1$).
Luego, existirán k_1 y k_2 (con $3 \leq k_1 < k_2 \leq N$) tal que

$$r_{k_1} - s_{k_1} + s_2 \equiv r_{k_2} - s_{k_2} + s_2 \bmod (q-1)$$

Es decir, $r_{k_1} + s_{k_2} \equiv r_{k_2} + s_{k_1} \bmod (q-1)$, lo que contradiría (*).

□

Definición 2.26. *Diremos que A es una matriz triangular superior si $a_{ij} = 0$ para todo $i > j$. Diremos que A es una matriz **triangular** si es una permutación por columnas de una matriz triangular superior.*

Proposición 2.27. *Una matriz **triangular NSC** tiene exactamente $i-1$ ceros en la fila i-ésima, con $1 \leq i \leq M$, y ninguna matriz NSC puede tener más ceros.*

Demostración. Si la fila i-ésima de A tiene más de $i-1$ ceros, entonces A no será NSC, ya que, si $a_{ij_1}, \ldots, a_{ij_i}$ son ceros, entonces el determinante de $A(j_1, \ldots, j_i)$ será cero.

□

Teorema 2.28. *Si A es NSC, y $\mathcal{C} = [\mathcal{C}_1 \cdots \mathcal{C}_M] \cdot A$ entonces:*

(I) $\#\mathcal{C} = \#\mathcal{C}_1 \cdots \#\mathcal{C}_M$

(II) $d(\mathcal{C}) \geq d^* = min\{Nd_1, (N-1)d_2, \ldots, (N-M+1)d_M\}$

(III) Si A es además triangular, entonces $d(\mathcal{C}) = d^*$

Demostración. (I) Como A es NSC, tenemos que A es singular; luego por la Proposición 2.18 se sigue el resultado.

(II) Para códigos de una sola palabra, diremos que la distancia mínima será ∞.
Tenemos entonces que la distancia mínima de $\mathcal{C}$ será ∞ si y sólo si cada $\mathcal{C}_1, \ldots, \mathcal{C}_M$ tienen distancia mínima ∞. (Es decir, $\mathcal{C}$ tendrá una sola palabra si a su vez $\mathcal{C}_1, \ldots, \mathcal{C}_M$ sólo tienen una palabra).
Estudiemos ahora la distancia mínima para el caso $\#\mathcal{C} > 1$. Probemos que para $\mathbf{c} = [\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A \neq 0 \in \mathcal{C}$, se tiene que $w_H(\mathbf{c}) \geq d^*$. Es decir, probemos que $\mathbf{c}$ es distinto de 0 en al menos d^* posiciones:

$$c_{h1}a_{1j} + \cdots + c_{hM}a_{Mj} \neq 0, \text{ para al menos } d^* \text{ valores de } h \text{ y } j. \qquad (*)$$

- Vamos a calcular para cuántos valores de j la ecuación (*) es distinta de 0.
  Como $\mathbf{c} = \mathbf{c} = [\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A \neq 0$, existirá alguna palabra $\mathbf{c_i} \in \mathcal{C}_i$ que sea distinta de 0. Consideremos el índice r, para el que $\mathbf{c_r}$ será la última palabra distinta de 0. Es decir, $r = \text{máx}\{i : \mathbf{c_i} \neq 0\}$. Luego, (*) será cierto si

$$c_{h1}a_{1j} + \cdots + c_{hr}a_{rj} \neq 0.$$

Probemos pues que, si la palabra $\mathbf{c_r}$ en la posición h es distinta de 0 ($c_{hr} \neq 0$), entonces se tiene $c_{h1}a_{1j} + \cdots + c_{hr}a_{rj} \neq 0$ para al menos $(N - r + 1)$ valores de j. Procedemos por reducción al absurdo.
Tomemos un h tal que $c_{hr} \neq 0$ (podemos hacer esto ya que $\mathbf{c_r} \neq 0$). Vamos a asumir que $c_{h1}a_{1j} + \cdots + c_{hr}a_{rj} = 0$ para al menos r valores de j. Es decir, existen $j_1, \ldots, j_r$ (con $1 \leq j_1 < \ldots < j_r \leq N$) tales que:

$$\begin{gathered} c_{h1}a_{1j_1} + \cdots + c_{hr}a_{rj_1} = 0 \\ \vdots \\ c_{h1}a_{1j_r} + \cdots + c_{hr}a_{rj_r} = 0 \end{gathered}$$

Por tanto, el sistema lineal

$$[c_{h1}, \ldots, c_{hr}] \cdot \begin{bmatrix} a_{1j_1} & \cdots & a_{1j_r} \\ \vdots & \ddots & \vdots \\ a_{rj_1} & \cdots & a_{rj_r} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

tiene solución distinta de la trivial. Es decir, la matriz $A(j_1, \ldots, j_r) \in \mathbb{F}_q^{r \times r}$ es singular. Con lo cual, hemos llegado a una contradicción, dado que por hipótesis A es NSC. Por tanto la ecuación (*) es distinta de 0 en $N - r + 1$ valores de j.

- Veamos ahora, para cada j, cuántos valores de h hacen que la ecuación (*) sea distinta de 0.
  Como $\mathbf{c_r} \neq 0$, tenemos que $w_H(\mathbf{c_r}) \geq w_H(\mathcal{C}_r) = d_r$, donde d_r es la distancia mínima del código $\mathcal{C}_r$. Por tanto $c_{hr} \neq 0$ para d_r valores de h entre 1 y n.

Por consiguiente, tenemos que $w_H(\mathbf{c}) \geq (N - r + 1)d_r \geq d^*$.

(III) Por la Proposición 2.14 podemos considerar que A es triangular superior. Sólo debemos probar que existen palabras del código que están, a lo sumo, a una distancia d^* la una de la otra.
Sea $(N - r + 1)d_r$ el valor más pequeño de $(N - i + 1)d_i$, con $i \in \{1, \ldots, M\}$. Sea $\mathbf{c} = [\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A \in \mathcal{C}$ una palabra del código tal que

$$w_H(\mathbf{c_r}) = d_H(\mathbf{c_r}, 0) = d_r, \text{ y } \mathbf{c_i} = 0 \text{ para cualquier } i \neq r.$$

Entonces tendremos que

$$\begin{gathered} w_H(\mathbf{c}) = w_H([0, \ldots, \mathbf{c_r}, \ldots, 0] \cdot A) = \\ w_H(\mathbf{c_r} \cdot a_{rr}) + w_H(\mathbf{c_r} \cdot a_{rr+1}) + \ldots + w_H(\mathbf{c_r} \cdot a_{rN}) \leq \\ \leq (N - r + 1)w_H(\mathbf{c_r}) = (N - r + 1)d_r \end{gathered}$$

□

Definición 2.29. *Sea la matriz* $A = \begin{pmatrix} a_{11} & \ldots & a_{1N} \\ \vdots & & \vdots \\ a_{M1} & & a_{MN} \end{pmatrix}$. *Definimos* A_i *como la matriz formada por las* i *primeras filas de* A*, y definimos el código* C_{R_i} *como aquel que tiene como matriz generatriz a* A_i.

Para la siguiente Proposición, es necesario recordar la Definición de códigos MDS, dada en el Capítulo 1 (Definición 1.39).

Proposición 2.30. *A es NSC si y sólo si los códigos C_{R_i} son MDS $\forall i \in \{1,\dots,M\}$.*

Demostración. Por la propia definición de matriz NSC (Definición 2.19), tenemos que A es NSC si y solo si $A(j_1,\dots,j_t)$ son matrices no singulares para $1 \leq t \leq M$, con $1 \leq j_1 < \dots < j_t \leq M$. A su vez, esto será cierto si y sólo si cualesquiera r columnas de A_r son linealmente independientes. Y por la Proposición 1.40, tendremos que esto se cumple si y sólo si C_{R_r} es MDS para cualquier $r \in \{1,\dots,M\}$.

□

Proposición 2.31. *Sea $\mathcal{C} = [\mathcal{C}_1,\dots,\mathcal{C}_M] \cdot A$ con $A \in \mathbb{F}_q^{M\times N}$ una matriz NSC. Entonces $d(\mathcal{C}) \geq \min\{d_1D_1,\dots,d_MD_M\}$, con $d_i = d(C_i)$ y $D_i = d(C_{R_i})$.*

Demostración. Consideremos un $\mathbf{c} \in \mathcal{C} = [\mathcal{C}_1,\dots,\mathcal{C}_M] \cdot A$ cualquiera. Entonces, la palabra será de la forma $\mathbf{c} = [\mathbf{c_1},\dots,\mathbf{c_M}] \cdots A$, con $\mathbf{c_i} = \begin{bmatrix} c_{1i} \\ \vdots \\ c_{ni} \end{bmatrix} \in \mathcal{C}_i$ Supongamos, sin pérdida de generalidad, que $\mathbf{c_i} \neq 0$ para $i \in \{1,\dots,r\}$, y $\mathbf{c_i} = 0$ para los índices restantes, es decir, $i \in \{r+1,\dots,M\}$. Entonces,

$$\mathbf{c} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1r} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ c_{11} & c_{12} & \cdots & c_{1r} & 0 & \cdots & 0 \end{pmatrix} \cdot \begin{pmatrix} a_{11} & \cdots & a_{1M} \\ \vdots & & \vdots \\ a_{M1} & \cdots & a_{MN} \end{pmatrix} =$$

$$= \begin{pmatrix} c_{11}a_{11} + \cdots + c_{1r}a_{r1} & \cdots & c_{11}a_{1N} + \cdots + c_{1r}a_{rN} \\ \vdots & & \vdots \\ c_{n1}a_{11} + \cdots + c_{nr}a_{r1} & \cdots & c_{n1}a_{1N} + \cdots + c_{nr}a_{rN} \end{pmatrix} =: \begin{pmatrix} s_{11} & \cdots & s_{1N} \\ \vdots & & \vdots \\ s_{n1} & & s_{nN} \end{pmatrix}$$

Basándonos en la nueva notación que hemos introducido para $\mathbf{c}$, denominaremos a la fila i-ésima de $\mathbf{c}$ como $\mathbf{s_i} = (s_{i1},\dots,s_{iN}) = (c_{i1}\dots c_{iM}) \cdot A$.
Observemos que $s_i \in C_{R_r}$ para cualquier $i = 1,\dots,n$, dado que la fila está generada por los elementos de A_r. Es decir, los elementos de la matriz formada por las r primeras filas de A. Por hipótesis, tenemos que $\mathbf{c_r} \neq 0$. Luego se tiene que $w_H(\mathbf{c_r}) \geq d_r$. Es decir, existen ciertos $\mu_1,\dots,\mu_{d_r}$ tales que $c_{\mu_i,r} \neq 0$ para $i = 1,\dots,d_r$.
Luego, la palabra $\mathbf{s_i}$ con $i \in \{\mu_1,\dots,\mu_{d_r}\}$ es una palabra de C_{R_r} distinta de cero. En caso de ser igual a cero, A_r no tendría rango máximo.
Por tanto, tenemos que $w_H(\mathbf{s_i}) \geq D_r$, dado que el peso de una palabra del código C_{R_r} distinta de 0 será siempre mayor o igual a la distancia mínima de este, D_r. Luego tenemos entonces que $w_H(\mathbf{c}) \geq d_rD_r$.

□

Teorema 2.32. *Sea $\mathcal{C} = [\mathcal{C}_1,\dots,\mathcal{C}_M] \cdot A$ el código producto de matrices, donde $\mathcal{C}_1,\dots,\mathcal{C}_M$ son códigos lineales anidados (es decir, $\mathcal{C}_1 \supset \cdots \supset \mathcal{C}_M$) y la matriz $A \in \mathbb{F}_q^{M\times N}$ tiene rango máximo. Entonces, la distancia mínima de $\mathcal{C}$ es*

$$d(\mathcal{C}) = \min\{d_1D_1, d_2D_2, \dots, d_MD_M\}$$

donde $d_i = d(\mathcal{C}_i)$, $D_i = d(C_{R_i})$ (C_{R_i} es el dado por la Definición 2.29) .

Demostración. Ya sabemos por la Proposición 2.31 que para cualquier palabra $\mathbf{c} = [\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A \neq 0$ tiene peso mayor o igual a $d(\mathcal{C}) = \text{mín}\{d_1 D_1, \ldots, d_M D_M\}$, donde $d(\mathcal{C}) = d_r D_r$, para cierto $r \in \{1, \ldots, M\}$. Para ver la igualdad, vamos a construir una palabra con peso $d_r D_r$ para $r \in \{1, \ldots, M\}$. Escogemos palabras $\mathbf{c_1}, \ldots, \mathbf{c_r}$ tales que $\mathbf{c_1} = \ldots = \mathbf{c_r}$, con $w_H(\mathbf{c_1}) = d_r$. Esto es posible, dado que $\mathcal{C}_1 \supset \cdots \supset \mathcal{C}_M$. Para los índices restantes tendremos que $\mathbf{c_{r+1}} = \ldots = \mathbf{c_M} = 0$.
Definamos ahora una palabra del código C_{R_r} con peso D_r, $f = \sum_{i=1}^{M} f_i R_i$, con $f_i \in \mathbb{F}_q$. Entonces, si $\mathbf{c'_i} = f_i \mathbf{c_i}$, se tiene que:

$$\left(\sum_{j=1}^{M} a_{j,1} \mathbf{c'_j}, \ldots, \sum_{j=1}^{M} a_{j,N} \mathbf{c'_j} \right) = \mathbf{c_1} \left(\sum_{j=1}^{r} a_{j,1} f_j, \ldots, \sum_{j=1}^{r} a_{j,N} f_j \right) = \mathbf{c_1} f$$

que es una palabra de $\mathcal{C}$ con peso $d_r D_r$. Por tanto, se tiene la igualdad.

□

Observación 2.33. Tenemos que, si A es NSC, el resultado $II)$ del Teorema 2.28

$$d(\mathcal{C}) \geq min\{Nd_1, (N-1)d_2, \ldots, (N-M+1)d_M\}$$

se puede deducir por la Proposición 2.31. Esto se debe a que A_i define un código C_{R_i} que será MDS (porque las filas serán linealmente independientes), luego $D_i = N - i + 1$.

Observación 2.34. En el caso de que A sea NSC y triangular, se tiene la igualdad $d(\mathcal{C}) = \text{mín}\{Nd_1, (N-1)d_2, \ldots, (N-M+1)d_M\}$ (apartado $III)$ del Teorema 2.28). Observemos que este resultado también se podría obtener por el Teorema 2.32, con A NSC , si los códigos $\mathcal{C}_1, \ldots, \mathcal{C}_M$ son anidados.

Ejemplo 2.35. En $\mathbb{F}_3$, consideremos los códigos lineales $\mathcal{C}_1, \mathcal{C}_2$ y $\mathcal{C}_3$ con matrices generatrices

$$G_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{F}_3^{3\times 3}, G_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \end{pmatrix} \in \mathbb{F}_3^{3\times 2} \quad \text{y} \quad G_3 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_3^{3\times 1}$$

respectivamente. Tenemos entonces que $\mathcal{C}_1$ es un $[3,3,1]_3$-código, $\mathcal{C}_2$ es un $[3,2,2]_3$-código, y $\mathcal{C}_3$ es un $[3,1,3]_3$-código. Además, están anidados, es decir, $\mathcal{C}_1 \supset \mathcal{C}_2 \supset \mathcal{C}_3$.

Consideremos el código producto de matrices $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3] \cdot A$, donde

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{F}_3^{3\times 3}.$$

La distancia mínima de los códigos C_{R_i} que se obtienen de la matriz A (Definición 2.29) será $D_1 = 3$, $D_2 = 2$ y $D_3 = 1$. Por tanto, tenemos que $\mathcal{C}$ es un código de longitud $n = 3 \times 3 = 9$, dimensión $k = 3 + 2 + 1 = 6$ (por ser A de rango máximo), y de distancia mínima $d = \text{mín}\{d_1 D_1, d_2 D_2, d_3 D_3\} = 3$ (por el Teorema 2.32).

3

Decodificación de Código Producto de Matrices Anidado

En este capítulo estudiaremos un algoritmo de decodificación eficiente para un tipo concreto de códigos Producto de Matrices, en los que los códigos $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M$ serán códigos anidados; es decir, $\mathcal{C}_1 \supset \mathcal{C}_2 \supset \cdots \supset \mathcal{C}_M$. El decodificador que hemos obtenido lo desarrollaron F. Hernando, K. Lally y D. Ruano en 2009 [5].

Consideremos $\mathcal{C}_1, \ldots, \mathcal{C}_M \subset \mathbb{F}_q^n$ códigos lineales, y vamos a suponer que, además, están anidados; es decir, $\mathcal{C}_1 \supset \mathcal{C}_2 \supset \ldots \supset \mathcal{C}_M$. También supondremos que $A \in \mathbb{F}_q^{M \times N}$ es una matriz NSC, con $M \leq N$.
Supongamos que tenemos un algoritmo de decodificación DC_i para cada $\mathcal{C}_i$, que corrige hasta $t_i = \lfloor \frac{d_i - 1}{2} \rfloor$ errores, con $d_i = d(\mathcal{C}_i)$, $\forall i \in \{1, \ldots, M\}$. En esta seccción, nos planteamos el problema de buscar un algoritmo de decodificación para el código producto de matrices $\mathcal{C} = [\mathcal{C}_1 \cdots \mathcal{C}_M] \cdot A \subset \mathbb{F}_q^{nN}$.

Recordemos que $\mathcal{C}$ es un código de longitud $n(\mathcal{C}) = nN$ (por la Definición 2.8), dimensión $k(\mathcal{C}) = k_1 + \ldots + k_M$ (por la Observación 2.12, al ser A NSC), y con distancia mínima $d(\mathcal{C}) = \text{mín}\{Nd_1, (N-1)d_2, \ldots, (N-M+1)d_M\}$ (por el Teorema 2.32 y la Observación 2.34, ya que al ser los códigos $\mathcal{C}_i$ anidados y A NSC, se tiene la igualdad para la distancia mínima).

Sabemos que una palabra de $\mathcal{C}$ es de la forma $\mathbf{c} = (\sum_{j=1}^M a_{j,1}\mathbf{c_j}, \ldots, \sum_{j=1}^M a_{j,N}\mathbf{c_j})$, donde $\mathbf{c_j} \in \mathcal{C}_j$, para todo j. Supongamos que recibimos el vector $\mathbf{p} = \mathbf{c} + \mathbf{e} = (\sum_{j=1}^M a_{j,1}\mathbf{c_j} + e_1, \ldots, \sum_{j=1}^M a_{j,N}\mathbf{c_j} + e_N)$, donde $\mathbf{e} = (e_1, \ldots, e_N) \in \mathbb{F}_q^{nN}$.

Una primera idea como algoritmo de decodificación sería pensar en decodificar los N bloques de $\mathbf{p}$ con DC_1. Podemos hacer esto ya que, como estamos trabajando con códigos $\mathcal{C}_1, \ldots, \mathcal{C}_M$ anidados, tenemos que el bloque j-ésimo de $\mathbf{c}$, $\mathbf{c}^{(j)} = \sum_{j=1}^M a_{j,i}\mathbf{c_j}$ es una palabra de $\mathcal{C}_1$, para todo $j \in \{1, \ldots, N\}$. En este caso, seríamos capaces de corregir, como máximo, t_1 errores en cada bloque. Es decir, $t_1 N$ errores en total. Sin embargo, si los errores no están distribuidos de forma que haya máximo t_1 errores en cada bloque, la decodificación fallará. Por tanto, para garantizar que la decodificación con esta idea se realiza con éxito, el número máximo de errores que podemos corregir es t_1.

Aplicando este algoritmo, no alcanzaríamos la máxima capacidad correctora del código $\mathcal{C}$, que hemos definido como $t_{\mathcal{C}} = \lfloor \frac{d(\mathcal{C}) - 1}{2} \rfloor$ errores.

En las siguientes líneas describiremos un algoritmo donde usamos todos los decodificadores DC_i para $i \in \{1, \dots, M\}$, y con el que podemos alcanzar la capacidad máxima de corrección del código $\mathcal{C}$, $t_{\mathcal{C}}$. En este algoritmo, la distribución de errores no debe verificar ninguna condición especial para que la decodificación pueda ser realizada con éxito.

Input: Recibimos $\mathbf{y} = \mathbf{c} + \mathbf{e}$ con $\mathbf{c} \in \mathcal{C} = [\mathcal{C}_1, \dots, \mathcal{C}_M] \cdot A$, donde $\mathcal{C}_M \subset \cdots \subset \mathcal{C}_1$, y $\mathbf{e} \in \mathbb{F}_q^{nN}$ con $w_H(\mathbf{e}) \leq t_{\mathcal{C}} = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$.
Conocemos $A \in \mathbb{F}_q^{M \times N}$ y $\mathcal{C}_1, \dots, \mathcal{C}_M$, así como unos decodificadores por mínimas distancias DC_i de cada código, que corrigen t_i errores.

---

1 $\mathbf{y}' = \mathbf{y}$, $A' = A$
2 **for** $\{i_1, \dots, i_M\} \subset \{1, \dots, N\}$ **do**
3 $\mathbf{y} = \mathbf{y}'$, $A = A'$
4 **for** $j = 1, \dots M$ **do**
5 $y_{i_j} = DC_j(y_{i_j})$
6 **if** $y_{i_j} =$ *"error"* **then**
7 **break** Volvemos a la línea 2 tomando otro subconjunto de índices
8 **end**
9 **for** $k = j+1, \dots M$ **do**
10 $y_{i_k} = y_{i_k} - \frac{a_{j,i_k}}{a_{j,i_j}} y_{i_j}$
11 column $_{i_k}(A) =$ column $_{i_k}(A) - \frac{a_{j,i_k}}{a_{j,i_j}}$column$_{i_j}(A)$
12 **end**
13 **end**
14 Recuperamos $(\mathbf{c_1}, \dots, \mathbf{c_M})$
15 $\mathbf{y} = [\mathbf{c_1}, \dots, \mathbf{c_M}] \cdot A$
16 **if** $\mathbf{y} \in \mathcal{C}$ y $w_H(\mathbf{y} - \mathbf{y}') \leq \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$ **then**
17 **return y**
18 **end**
19 **else**
20 **break** Volvemos a la línea 2 tomando otro subconjunto de índices
21 *# Este caso ocurre cuando los decodificadores DC_j son de Tipo 2.*
22 **end**
23 **end**

Algoritmo 1: Decodificación de Código Producto de Matrices Anidados

Teorema 3.1. *El algoritmo 1 es un algoritmo de decodificación por mínimas distancias del código $\mathcal{C} = [\mathcal{C}_1, \dots, \mathcal{C}_M] \cdot A \in \mathbb{F}_q^{nN}$ que corrige $t \leq t_{\mathcal{C}}$ errores y devuelve Error si la palabra recibida está a una distancia de $\mathcal{C}$ mayor a t.*

Durante las siguientes secciones vamos a describir y justificar este resultado.

3.1. Descripción del algoritmo

Supongamos que la palabra $\mathbf{c} = (\sum_{j=1}^{M} a_{j,1}\mathbf{c_j}, \dots, \sum_{j=1}^{M} a_{j,N}\mathbf{c_j})$, con $\mathbf{c_j} \in \mathcal{C}_j$, $\forall j \in \{1, \dots, M\}$ es enviada, y que recibimos $\mathbf{p} = \mathbf{c} + \mathbf{e}$, donde $\mathbf{e} = (e_1, \dots, e_N) \in \mathbb{F}_q^{nN}$ es

el vector de errores (con $e_i \in \mathbb{F}_q^n$).
Sea $\{i_1, \ldots, i_M\} \subset \{1, \ldots, N\}$ un subconjunto de índices para los que el vector de errores $\mathbf{e}$ satisface que $w_H(e_{i_j}) \leq t_j, \forall j \in \{1, \ldots, M\}$.
Llamamos $p_i = \sum_{j=1}^{M} a_{j,i}\mathbf{c_j} + e_i \in \mathbb{F}_q^n$ al bloque i-ésimo de $\mathbf{p}$, con $i \in \{1, \ldots N\}$. Como ya hemos visto, cada bloque $\sum_{j=1}^{M} a_{j,i}\mathbf{c_j}$ de $\mathbf{c}$ es una palabra del código $\mathcal{C}_1$. Por tanto, como $w_H(e_{i_1}) \leq t_1$, podemos aplicar DC_1 al i_1-ésimo bloque de $\mathbf{p}$, y así recuperamos el i_1-bloque de la palabra recibida (obtenemos e_{i_1} y el bloque p_{i_1} sin errores).
Esto nos permite que, pese a no conocer la palabra $\mathbf{c_1} \in \mathcal{C}_1$ que forma parte de $\mathbf{c}$, la podamos eliminar de todos los bloques i-ésimos de $\mathbf{p}$, con $i \neq i_1$, al realizar la siguiente operación.
Consideremos un nuevo vector $\mathbf{p^{2)}} \in \mathbb{F}_q^{nN}$ que definimos como

$$p_i^{2)} = p_i - \tfrac{a_{1,i}}{a_{1,i_1}}(p_{i_1} - e_{i_1}) = \sum_{j=1}^{M} a_{j,i}\mathbf{c_j} + e_i - \frac{a_{1,i}}{a_{1,i_1}}\left(\sum_{j=1}^{M} a_{j,i}\mathbf{c_j}\right) =$$
$$= \sum_{j=1}^{M}\left(a_{j,i} - \frac{a_{1,i}}{a_{1,i_1}}a_{j,i_1}\right)\mathbf{c_j} + e_i.$$

Aquí vemos que, cuando $j = 1$, $\left(a_{1,i} - \tfrac{a_{1,i}}{a_{1,i_1}}a_{1,i_1}\right) = (a_{1,i} - a_{1,i}) = 0$. Por tanto, hemos conseguido eliminar $\mathbf{c_1}$ de p_i, con $i \neq i_1$.

Reescribamos $\mathbf{p^{2)}}$ de la siguiente manera:

$$p_i^{2)} = p_i - \tfrac{a_{1,i}}{a_{1,i_1}}(p_{i_1} - e_{i_1}) = \sum_{j=2}^{M} a_{j,i}^{2)}c_j + e_i, \text{ con } i \neq i_1$$

donde $a_{j,i}^{2)} = a_{j,i} - \tfrac{a_{1,i}}{a_{1,i_1}}a_{j,i_1}$, y para el índice i_1 definimos $p_{i_1}^{2)} = p_{i_1} - e_{i_1}$.
Tenemos que $a_{j,i}^{2)}$ está bien definido, ya que el denominador a_{1,i_1} es distinto de cero, porque, al ser A una matriz NSC, los elementos de la primera fila de A son distintos de cero (Proposición 2.27).

Ahora, como $\mathcal{C}_2 \supset \ldots \supset \mathcal{C}_M$, tenemos que cada bloque i-ésimo de $\mathbf{p^{2)}}$ es una palabra del código $\mathcal{C}_2$ más el vector error e_i, para cierto $i \in \{1, \ldots, M\} \setminus \{i_1\}$.
Entonces, como $w_H(e_{i_2}) \leq t_2$, procedemos a decodificar el i_2-ésimo bloque $p_{i_2}^{2)} = \sum_{j=2}^{M} a_{j,i_2}^{2)}\mathbf{c_j} + e_{i_2}$ de $\mathbf{p^{2)}}$ usando el decodificador DC_2. De esta forma obtenemos e_{i_2} y el i_2-ésimo bloque de $\mathbf{p}$ sin errores. De nuevo, no conocemos $\mathbf{c_2} \in \mathcal{C}_2$, que forma parte de $\mathbf{c} \in \mathcal{C}$, pero sí conocemos el error e_{i_2}. Por tanto, podemos eliminar $\mathbf{c_2}$ de cada bloque i-ésimo de $\mathbf{p}$ considerando el nuevo vector $\mathbf{p^{3)}} \in \mathbb{F}_q^{nN}$ definido de la siguiente forma:

$$p_i^{3)} = p_i^{2)} - \tfrac{a_{2,i}^{2)}}{a_{2,i_2}^{2)}}(p_{i_2}^{2)} - e_{i_2}) = \sum_{j=3}^{M} a_{j,i}^{3)}\mathbf{c_j} + e_i, \text{ con } i \neq i_1, i_2$$

donde $a^{3)}_{j,i} = a^{2)}_{j,i} - \frac{a^{2)}_{2,i}}{a^{2)}_{2,i_2}} a^{2)}_{j,i_2}$, y para los índices i_1, i_2 definimos $p^{3)}_{i_1} = p^{2)}_{i_1}$ y $p^{3)}_{i_2} = p^{2)}_{i_2} - e_{i_2}$.
Tenemos por tanto que el bloque i-ésimo de $\mathbf{p^{3)}}$ es una palabra del código $\mathcal{C}_3$ más el error e_i, para cierto $i \in \{1, \ldots, M\} \setminus \{i_1, i_2\}$. Luego, en el índice i_3, donde $w_H(e_{i_3}) < t_3$, usamos el decodificador DC_3, y obtenemos el error e_{i_3}.
Repetimos este proceso iterativamente, definiendo $\mathbf{p^{k)}}$ para los restantes $k \in \{i_3, \ldots, i_M\}$:

$$p^{k)}_i = p^{k-1)}_i - \frac{a^{k-1)}_{k-1,i}}{a^{k-1)}_{k-1,i_{k-1}}} (p^{k-1)}_{i_{k-1}} - e_{i_{k-1}}) = \sum_{j=k}^{M} a^{k)}_{j,i} \mathbf{c_j} + e_i, \text{ con } i \neq i_1, \ldots, i_{k-1}$$

cuyos bloques serán palabras de $\mathcal{C}_k$. Por tanto, podemos obtener el error e_{i_k} usando el decodificador DC_k y así sucesivamente.

Una vez usados todos los decodificadores, habremos obtenido los errores e_i de los M bloques. El vector resultante $\left(\sum_{j=1}^{M} a_{j,i_1} \mathbf{c_j}, \ldots, \sum_{j=1}^{M} a_{j,i_M} \mathbf{c_j}\right)$ en realidad es el producto $[\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A(i_1, \ldots, i_M)$, donde $A(i_1, \ldots, i_M)$ es la submatriz de tamaño $M \times M$ cuyas columnas son $i_1, \ldots, i_M$.
Como A es NSC, la matriz $A(i_1, \ldots, i_M)$ es de rango M; es decir, rango máximo. Por lo tanto, podemos calcular las palabras $\mathbf{c_1}, \ldots, \mathbf{c_M}$ fácilmente, simplemente invirtiendo la matriz $A(i_1, \ldots, i_M)$, o resolviendo el sistema lineal de ecuaciones correspondiente.
Para recuperar los $N - M$ bloques restantes, no es necesario utilizar ningún proceso de decodificación (dado que ya conocemos A y los $\mathbf{c_j}$); simplemente calculamos

$$\mathbf{c} = [\mathbf{c_1}, \ldots, \mathbf{c_M}] \cdot A = \left(\sum_{j=1}^{M} a_{j,1} \mathbf{c_j}, \ldots, \sum_{j=1}^{M} a_{j,N} \mathbf{c_j}\right).$$

Y así hemos recuperado la palabra $\mathbf{c}$ que habíamos enviado originalmente, siempre que el número de errores que se han producido durante la transmisión haya sido $< t_{\mathcal{C}}$, y siempre que encontremos un conjunto de índices $\{i_1, \ldots, i_M\}$ tal que $w(e_{i_j}) < t_j$.

3.2. Justificación del algoritmo

En el algoritmo presuponemos que si el vector error $\mathbf{e}$ cumple que $w_H(\mathbf{e}) \leq \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$, entonces siempre existe un subconjunto de índices $\{i_1, \ldots, i_M\} \subset \{1, \ldots N\}$ tal que satisfacen que $w(e_{i_j}) \leq t_j$ para todo j. Probemos que realmente se puede escoger un conjunto de índices que cumpla tal condición. Con el siguiente resultado, queda demostrado que el algoritmo acaba en un número finito de pasos.

Teorema 3.2. *Sea $\mathcal{C}$ el código generado por el producto de matrices $[\mathcal{C}_1, \ldots, \mathcal{C}_M] \cdot A$, donde $\mathcal{C}_1 \supset \cdots \supset \mathcal{C}_M$, y A es una matriz NSC. Sea $\mathbf{e} = (e_1, \ldots, e_N) \in \mathbb{F}_q^{nN}$ un vector de errores que cumple que $w_H(\mathbf{e}) \leq \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$. Entonces, existe un subconjunto ordenado de índices $\{i_1, \ldots, i_M\} \subset \{1, \ldots N\}$ que satisface que $w(e_{i_j}) \leq t_j = \lfloor \frac{d_j-1}{2} \rfloor$ para todo $j \in \{1, \ldots, M\}$.*

Demostración. Procedemos por inducción sobre M.

1. Primero debemos probar que siempre podremos elegir un primer índice con el que comenzar el proceso. Es decir, debemos probar que existe i_1 tal que $w_H(e_{i_1}) \leq t_1$.

Supongamos que no existe ningún $i_1 \in \{1,\dots,N\}$ tal que $w_H(e_{i_1}) \leq t_1$. Entonces, se tendrá que $w_H(e_i) > t_1 = \lfloor \frac{d_1-1}{2} \rfloor$ para todo $i \in \{1,\dots,N\}$. Luego, $w_H(e_i) \geq \frac{d_1}{2}$ para todo i. Entonces, usando que $d(\mathcal{C}) = \min\{Nd_1,\dots,(N-M+1)d_M\}$ (Teorema 2.32), tendremos que

$$w(\mathbf{e}) \geq \frac{Nd1}{2} > \frac{Nd_1-1}{2} \geq \lfloor \frac{Nd_1-1}{2} \rfloor \geq \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$$

con lo cual llegamos a un absurdo.

2. Asumamos, como hipótesis de inducción, que la propiedad de los pesos de los errores se cumple para un cierto subconjunto $\{i_1,\dots,i_{j-1}\} \subset \{1,\dots,N\}$ de tamaño $j-1 < M$.

3. Veamos que se sigue cumpliendo añadiendo un índice más (es decir, considerando un subconjunto de tamaño j).
Para ello, procedamos por reducción al absurdo, suponiendo que no existe ningún i_j tal que $w_H(e_{i_j}) \leq t_j$. Es decir, tendremos que $w_H(e_i) > \lfloor \frac{d_j-1}{2} \rfloor$ para todo $i \in \{1,\dots,N\} \setminus \{i_1,\dots i_{j-1}\}$. Luego, $w_H(e_i) \geq \frac{d_j}{2}$ para todo $i \in \{1,\dots,N\} \setminus \{i_1,\dots i_{j-1}\}$. Usando de nuevo el Teorema 2.32, obtenemos que

$$w_H(e) \geq \sum_{k=1}^{j-1} w_H(e_{i_k}) + \frac{(l-j+1)d_j}{2} > \sum_{k=1}^{j-1} w_H(e_{i_k}) + \frac{(l-j+1)d_j - 1}{2} \geq \lfloor \frac{(l-j+1)d_j-1}{2} \rfloor \geq \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$$

que contradice nuestra hipótesis de partida, luego el resultado queda demostrado.

□

A la hora de describir el algoritmo, definimos $\mathbf{p^{k+1)}}$ para poder eliminar la palabra c_k correspondiente y continuar con el proceso de decodificación. Sin embargo, para que $\mathbf{p^{k+1)}}$ esté bien definido es necesario que $a^{k)}_{k,i_k} \neq 0$ para cada $k = 2,\dots,M$. Probemos que esto se cumple siempre que A sea NSC.

Proposición 3.3. *Si A es NSC, se tiene que $\mathbf{p^{k+1)}}$ está bien definido, $\forall k \in \{1,\dots,M\{$.*

Demostración. Vamos a demostrarlo por inducción sobre k, con $k \in \{1,\dots,M\{$. Sea $A^{1)} = A$. Por la Proposición 2.27, tenemos que $a_{1,i_1} \neq 0$, por ser A una matriz NSC.
A la hora de calcular $\mathbf{p^{2)}}$, a cada $a_{j,i}$ de la matriz $A^{1)}$ debemos de restarle $\frac{a_{1,i}}{a_{1,i_1}} a_{j,i_1}$, excepto a los que se encuentren en la columna i_1-ésima, que permanecerán igual. Llamemos a la matriz resultante de hacer estas operaciones $A^{2)}$.

$$A^{2)} = \begin{pmatrix} a_{1,1} - \frac{a_{1,1}}{a_{1,i_1}} a_{1,i_1} & a_{1,2} - \frac{a_{1,2}}{a_{1,i_1}} a_{1,i_1} & \dots & a_{1,i_1} & \dots & a_{1,N} - \frac{a_{1,N}}{a_{1,i_1}} a_{1,i_1} \\ \vdots & & \ddots & & & \vdots \\ a_{M,1} - \frac{a_{1,1}}{a_{1,i_1}} a_{M,i_1} & a_{M,2} - \frac{a_{1,2}}{a_{1,i_1}} a_{M,i_1} & \dots & a_{M,i_1} & \dots & a_{M,N} - \frac{a_{1,N}}{a_{1,i_1}} a_{M,i_1} \end{pmatrix}$$

$$= \begin{pmatrix} 0 & \dots & a^{2)}_{1,i_1} & \dots & 0 \\ a^{2)}_{2,1} & \dots & a^{2)}_{2,i_1} & \dots & a^{2)}_{2,N} \\ \vdots & & \ddots & & \vdots \\ a^{2)}_{M,1} & \dots & a^{2)}_{M,i_1} & \dots & a^{2)}_{M,N} \end{pmatrix}.$$

Al definir $A^{2)}$ hemos obtenido $N-1$ ceros en la primera fila. Luego, podemos definir la siguiente matriz triangular:

$$A^{2)}(i_1, i_2) = \begin{pmatrix} a^{2)}_{1,i_1} & 0 \\ a^{2)}_{2,i_1} & a^{2)}_{2,i_2} \end{pmatrix}$$

siguiendo la notación de la Definición 2.16.
Tendremos que $det(A^{2)}(i_1, i_2)) = a^{2)}_{1,i_1} \cdot a^{2)}_{2,i_2}$. Como A es NSC, este determinante es necesariamente distinto de cero. Y como sabemos que $a^{2)}_{1,i_1} = a_{1,i_1} \neq 0$, por la Proposición 2.27, se tiene entonces que $a^{2)}_{2,i_2} \neq 0$.

En general, para cualquier $k \leq M$ definimos la matriz $A^{k)} = (a^{k)}_{i,j})$ de forma recursiva a partir de $A^{k-1)}$ haciendo las siguientes $N-(k-1)$ operaciones elementales por columnas:

$$\text{columna}_i(A^{k)}) = \text{columna}_i(A^{k-1)}) - \frac{a^{k-1)}_{k-1,i}}{a^{k-1)}_{k-1,i_{k-1}}} \text{columna}_{i_{k-1}}(A^{k-1)})$$

para cada $i \notin \{i_1, \dots i_{k-1}\}$. Estas operaciones convierte en ceros a $N-(k-1)$ elementos de la $(k-1)$-ésima fila de la matriz $A^{k)}$.
Luego, al considerar la matriz $A^{k)}(i_1, \dots, i_k)$, se tiene una matriz triangular, definida como:

$$A^{k)}(i_1, \dots, i_k) = \begin{pmatrix} a^{k)}_{1,i_1} & 0 & 0 & \dots & 0 & 0 \\ a^{k)}_{2,i_1} & a^{k)}_{2,i_2} & 0 & \dots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ a^{k)}_{k-1,i_1} & a^{k)}_{k-1,i_2} & a^{k)}_{k-1,i_3} & \dots & a^{k)}_{k-1,i_{k-1}} & 0 \\ a^{k)}_{k,i_1} & a^{k)}_{k,i_2} & a^{k)}_{k,i_3} & \dots & a^{k)}_{k,i_{k-1}} & a^{k)}_{k,i_k} \end{pmatrix}$$

cuyo determinante es $a^{k)}_{1,i_1} \cdots a^{k)}_{k,i_k}$. Como, por hipótesis, A es NSC, $A^{k)}(i_1, \dots, i_k)$ es no singular. Luego, el determinante $detA^{k)}(i_1, \dots, i_k)$ es distinto de cero. Por hipótesis de inducción, se tiene que $a^{k)}_{1,i_1}, \dots, a^{k)}_{k-1,i_{k-1}}$ son distintos de cero, luego se deduce que $a^{k)}_{k,i_k} \neq 0$.

□

3.3. Errores en la decodificación

Como ya hemos descrito, el algoritmo decodificará con éxito la palabra recibida cuando encuentre un conjunto de índices $\{i_1, \ldots, i_M\} \subset \{1, \ldots, N\}$ tal que cada bloque de errores e_{i_j} satisfaga que $w_H(e_{i_j}) \leq t_j = \lfloor \frac{d_j - 1}{2} \rfloor$, para todo $j \in \{1, \ldots, M\}$, siempre que $w_H(e) \leq \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$. Por el Teorema 3.2, sabemos que este conjunto de índices siempre existe.
Si comenzamos con un conjunto de índices $\{i_1, \ldots, i_M\}$, y ocurre que $w_H(e_{i_j}) > t_j$ para algún j, el proceso de decodificación fallará y habrá que que empezar con un nuevo conjunto de índices. Dependiendo del tipo de decodificador que estemos utilizando, habrán dos posibles opciones de detectar el error:

(I) **Para Decodificadores DC_j de Tipo 1**:
Si el decodificador DC_j es de Tipo 1, entonces nos devolverá como respuesta $DC_j(p_{i_j}^{j)}) = \{?\}$.
En caso de que esto ocurra, deberemos empezar de nuevo el proceso considerando un subconjunto de índices $\{i_1, \ldots, i_M\}$ diferente del anterior. Si tras probar con todos los posibles subconjuntos $\{i_1, \ldots, i_M\} \subset \{1, \ldots, N\}$ no obtenemos una palabra $\mathbf{c} \in \mathcal{C}$ que cumpla $d_H(\mathbf{c}, \mathbf{y}) \leq t_{\mathcal{C}}$, entonces se han producido más de $t_{\mathcal{C}}$ errores en la transmisión del mensaje.

(II) **Para Decodificadores DC_j de Tipo 2** :
Si el decodificador DC_j es de Tipo 2, entonces puede decodificar de forma incorrecta el bloque $p_{i_j}^{j)}$, por lo cual obtendríamos $e_k \neq e_{i_j}$. En este caso, podemos detectar el error de decodificación al final de la iteración, comprobando si la palabra final $\mathbf{c} = [c_1, \ldots, c_M] \cdot A$ es efectivamente una palabra de $\mathcal{C}$.
Si la palabra decodificada incorrectamente perteneciera a $\mathcal{C}$, entonces existirían dos palabras, una correcta y la otra incorrecta, $\mathbf{c}, \hat{\mathbf{c}} \in \mathcal{C}$ tales que $\mathbf{c} + \mathbf{e} = \hat{\mathbf{c}} + \hat{\mathbf{e}} = \mathbf{y}$, con $w_H(\mathbf{e}) = w_H(\hat{\mathbf{e}}) \leq t_{\mathcal{C}} = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$. Entonces se tendría que $w_H(\mathbf{c} - \hat{\mathbf{c}}) \leq 2w_H(\mathbf{e}) \leq d(\mathcal{C}) - 1 < d(\mathcal{C})$, lo cual es un absurdo, ya que no pueden haber dos palabras del código $\mathcal{C}$ a una distancia menor que $d(\mathcal{C})$.
Por tanto, si ha habido una decodificación incorrecta, obtendremos una palabra que no es del código $\mathcal{C}$, y tendremos que comenzar de nuevo el proceso con un nuevo subconjunto de índices.
Si tras probar con todos los posibles subconjuntos $\{i_1, \ldots, i_M\} \subset \{1, \ldots, N\}$ no obtenemos una palabra $\mathbf{c} \in \mathcal{C}$ que cumpla $d_H(\mathbf{c}, \mathbf{y}) \leq t_{\mathcal{C}}$, entonces se han producido más de $t_{\mathcal{C}}$ errores en la transmisión del mensaje.

3.4. Ejemplos

Veamos a continuación un ejemplo de decodificación.

Ejemplo 3.4. Vamos a trabajar en el cuerpo $\mathbb{F}_{11}$. Definimos

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{F}_{11}^{3\times 3}.$$

Observemos que A es triangular y NSC.
Consideremos ahora el vector $\mathbf{a} \in (\mathbb{F}_{11} \setminus \{0\})^{10}$, con $a_i \neq a_j$; es decir, $\mathbf{a} = (1, \ldots, 10)$. Definimos entonces los códigos de Reed Solomon[1] $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ (Véase Definición A.1), con $n(\mathcal{C}_i) = 10$ con matrices generatrices G_1, G_2, G_3 respectivamente.

$$G_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 4 & 9 & 5 & 3 & 3 & 5 & 9 & 4 & 1 \\ 1 & 8 & 5 & 9 & 4 & 7 & 2 & 6 & 3 & 10 \\ 1 & 5 & 4 & 3 & 9 & 9 & 3 & 4 & 5 & 1 \\ 1 & 10 & 1 & 1 & 1 & 10 & 10 & 10 & 1 & 10 \end{pmatrix} \in \mathbb{F}_{11}^{6\times 10}, \; G_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 4 & 9 & 5 & 3 & 3 & 5 & 9 & 4 & 1 \\ 1 & 8 & 5 & 9 & 4 & 7 & 2 & 6 & 3 & 10 \end{pmatrix} \in \mathbb{F}_{11}^{4\times 10}$$

y

$$G_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix} \in \mathbb{F}_{11}^{2\times 10}.$$

Por la Proposición A.3 (que se encuentra en el apéndice) tendremos que:

- $\mathcal{C}_1$ es un código de dimensión $k(\mathcal{C}_1) = 6$, distancia mínima $d(\mathcal{C}_1) = 5$, y capacidad correctora $t_1 = 2$.
- $\mathcal{C}_2$ es un código de dimensión $k(\mathcal{C}_2) = 4$, distancia mínima $d(\mathcal{C}_2) = 7$, y capacidad correctora $t_2 = 3$.
- $\mathcal{C}_3$ es un código de dimensión $k(\mathcal{C}_3) = 2$, distancia mínima $d(\mathcal{C}_1) = 9$, y capacidad correctora $t_3 = 4$.

Consideremos el código producto $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3] \cdot A$. Por la propia definición del código producto de matrices, la Observación 2.12 y el Teorema 2.32 para A NSC (explicado en la observación 2.34), tenemos que los parámetros de este código son:

- Longitud $n(\mathcal{C}) = 10 \times 3 = 30$.
- Dimensión $k(\mathcal{C}) = 6 + 4 + 2 = 12$.
- Distancia mínima $d(\mathcal{C}) = \min\{3d_1, 2d_2, 1d_3\} = \min\{3 \cdot 5, 2 \cdot 7, 1 \cdot 9\} = 9$.

Por tanto, la capacidad correctora de $\mathcal{C}$ será $t_{\mathcal{C}} = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor = \lfloor \frac{9-1}{2} \rfloor = 4$.

Sea $\mathbf{m} = (\underbrace{0,1,2,3,4,5}_{m_1}, \underbrace{6,7,8,9}_{m_2}, \underbrace{10,0}_{m_3}) \in \mathbb{F}_{11}^{12}$ nuestro mensaje.
La codificación de este mensaje es:

$$\mathbf{c} = \mathbf{m} \cdot G = (m_1 G_1, m_2 G_2, m_3 G_3) \cdot A = (c_1, c_2, c_3) \cdot A = (4,5,2,3,9,9,8,0,7,8,9,0,4,5,2,4,3,2,10,4,0,7,2,3,10,0,10,0,2,5) \in \mathbb{F}_{11}^{3\cdot 10}.$$

[1] En el Apéndice de este trabajo hablamos más extensamente de los códigos Reed-Solomon, estudiamos sus propiedades, y damos un algoritmo de decodificación eficiente para ellos.

Como la capacidad correctora de $\mathcal{C}$ es $t_{\mathcal{C}} = 4$, vamos a introducir un vector de errores cuyo peso de Hamming sea 4.

$$\mathbf{e} = (0,3,0,2,0,0,6,0,0,7,0,0,0,0,\ldots,0).$$

donde

- $e_1 = (0,3,0,2,0,0,6,0,0,7) \in \mathbb{F}_{11}^{10}$.
- $e_2 = (0,0,0,0,0,0,0,0,0,0) \in \mathbb{F}_{11}^{10}$.
- $e_3 = (0,0,0,0,0,0,0,0,0,0) \in \mathbb{F}_{11}^{10}$.

El vector recibido $\mathbf{y} = \mathbf{c} + \mathbf{e}$ es:

$$\mathbf{y} = (\underbrace{4,8,2,5,9,9,3,0,7,4}_{y_1},\underbrace{9,0,4,5,2,4,3,2,10,4}_{y_2},\underbrace{0,7,2,3,10,0,10,0,2,5}_{y_3}) \in \mathbb{F}_{11}^{3\cdot 10}.$$

Con estos parámetros de partida, veremos qué ocurre en la decodificación, dependiendo del tipo de decodificadores DC_j que estemos usando:

1. En este caso, utilizaremos decodificadores DC_j de Tipo 1. Siempre que elijamos un subconjunto de índices incorrecto, DC_j nos devolverá error, y empezaremos de nuevo. Después de elegir el subconjunto de índices correcto, lograremos decodificar correctamente.
   Como $M = N = 3$, es posible realizar $3! = 6$ ordenaciones distintas como subconjuntos de 3 elementos.

   - Elegimos $\{i_1 = 1, i_2 = 2, i_3 = 3\}$. Aplicamos DC_1 a y_1, y nos devuelve error, ya que el número de errores es mayor que t_1. Tendríamos el mismo problema si elegimos $\{i_1 = 1, i_2 = 3, i_3 = 1\}$.

   - Elegimos ahora la ordenación $\{i_1 = 2, i_2 = 1, i_3 = 3\}$.
     Para $j = 1$ aplicamos DC_1 a y_2. Como en el segundo bloque no hay ningún error, obtenemos $y_2^{2)} = y_2$.

     Calculamos los $y_{i_k}^{2)}$ para los índices restantes:

     • $i_2 = 1$

     $$y_1^{2)} = y_1 - \tfrac{a_{1,1}}{a_{1,2}} y_2^{2)} = (6,8,9,0,7,5,0,9,8,0).$$

     Calculamos la nueva matriz A, restando a la primera columna de A la segunda columna de A multiplicada por $\frac{a_{1,1}}{a_{1,2}} = 1$. Es decir,

     $$\text{columna}_1(A^{2)}) = \text{columna}_1(A^{1)}) - \tfrac{a_{1,1}}{a_{1,2}} \text{columna}_2(A^{1)})$$
     $$A^{2)} = \begin{pmatrix} 0 & 1 & 1 \\ 9 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

     • $i_3 = 3$

$$y_3^{2)} = y_3 - \frac{a_{1,3}}{a_{1,2}} y_2^{2)} = (2,7,9,9,8,7,7,9,3,1).$$

Recalculamos la matriz $A^{2)}$, restando a la tercera columna de A la segunda columna de A multiplicada por $\frac{a_{1,3}}{a_{1,2}} = 1$. Es decir,

$$\text{columna}_3(A^{2)}) = \text{columna}_3(A^{1)}) - \frac{a_{1,3}}{a_{1,2}} \text{columna}_2(A^{1)})$$

$$A^{2)} = \begin{pmatrix} 0 & 1 & 0 \\ 9 & 2 & 10 \\ 0 & 0 & 1 \end{pmatrix}.$$

Para $j = 2$, aplicamos el decodificador DC_2 a $y_1^{2)}$, y el algoritmo de nuevo nos devuelve fallo (al ser $w(e_1) > t_2$).

- Elegimos una nueva ordenación $\{i_1 = 2, i_2 = 3, i_3 = 1\}$.
Para $j = 1$ aplicamos DC_1 a y_2, y volvemos a tener

$$y_2 = y_2^{2)}.$$

Calculamos los $y_{i_k}^{2)}$ para los índices restantes:

- $i_2 = 3$

$$y_3^{2)} = y_3 - \frac{a_{1,3}}{a_{1,2}} y_2^{2)} = (2,7,9,9,8,7,7,9,3,1).$$

Calculamos la nueva matriz A, restando a la tercera columna de A la segunda columna de A multiplicada por $\frac{a_{1,3}}{a_{1,2}} = 1$. Es decir,

$$\text{columna}_3(A^{2)}) = \text{columna}_3(A^{1)}) - \frac{a_{1,3}}{a_{1,2}} \text{columna}_2(A^{1)})$$

$$A^{2)} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 10 \\ 0 & 0 & 1 \end{pmatrix}.$$

- $i_3 = 1$. Calculamos

$$y_1^{2)} = y_1 - \frac{a_{1,1}}{a_{1,2}} y_2^{2)} = (6,8,9,0,7,5,0,9,8,0).$$

Actualizamos $A^{2)}$

$$\text{columna}_1(A^{2)}) = \text{columna}_1(A^{1)}) - \frac{a_{1,1}}{a_{1,2}} \text{columna}_2(A^{1)})$$

$$A^{2)} = \begin{pmatrix} 0 & 1 & 0 \\ 9 & 2 & 10 \\ 0 & 0 & 1 \end{pmatrix}.$$

Para $j = 2$ aplicamos DC_2 al bloque $y_3^{2)}$. Como no se han producido errores en el segundo bloque tenemos que

$$y_3^{3)} = (2,7,9,9,8,7,7,9,3,1) = y_3^{2)}.$$

- $i_3 = 1$, calculamos

$$y_1^{3)} = y_1^{2)} - \frac{a_{2,1}^{2)}}{a_{2,3}^{2)}} \cdot y_3^{3)} = y_1^{2)} - \frac{9}{10} \cdot y_3^{3)} = (2,5,2,4,2,2,8,2,2,9).$$

Calculamos la matriz $A^{3)}$:

$$\text{columna}_1(A^{3)}) = \text{columna}_1(A^{2)}) - \frac{a^{2)}_{2,1}}{a^{2)}_{2,3}}\text{columna}_3(A^{1)})$$

$$A^{3)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 10 \\ 9 & 0 & 1 \end{pmatrix}.$$

Para $j = 3$, aplicamos DC_3 sobre $y_1^{3)}$ y obtenemos

$$y_1^{4)} = (2,2,2,2,2,2,2,2,2,2).$$

Ahora recuperamos el vector error:

$$e_1 = y_1^{3)} - y_1^{4)} = (0,3,0,2,0,0,6,0,0,7)$$
$$e_2 = y_2 - y_2^{2)} = (0,0,0,0,0,0,0,0,0,0)$$
$$e_3 = y_3^{2)} - y_3^{3)} = (0,0,0,0,0,0,0,0,0,0)$$

Hemos así recuperado la palabra originalmente enviada.

2. Veamos ahora este mismo ejemplo, trabajando con Decodificadores DC_j de Tipo 2. En este caso, veremos que si escogemos un subconjunto de índices que no es el adecuado, lo detectaremos solo al final, por haber obtenido una palabra que no pertenece al código $\mathcal{C}$.

   Vamos a probar el conjunto de índices $\{i_1 = 1, i_2 = 2, i_3 = 3\}$. Sabemos que este conjunto de índices no nos permitirá decodificar correctamente, ya que $w_H(e_1) > t_1$. Pero, como en este caso, DC_1 es un decodificador de tipo 2, nos devolverá una palabra que no es la originalmente enviada. Solo podremos detectar este error al final del proceso.
   En este caso, hemos implementado el algoritmo utilizando SageMath, que contiene una librería de Teoría de Códigos. En particular, tiene implementados algoritmos de decodificación de códigos RS de tipo 2.

   Definimos A y la longitud de nuestro código.

[1]:
```
A=matrix(GF(11),[[1,1,1],[0,2,1],[0,0,1]])
n=10
```

   Definimos $\mathcal{C}_1$ y su matriz generatriz G_1.

[3]:
```
k1=6
C1 = codes.GeneralizedReedSolomonCode(GF(11).list()[1:n+1], k1)
```

[3]:
```
[10, 6, 5] Reed-Solomon Code over GF(11)
```

[4]:
```
G1=C1.generator_matrix()
```

   Definimos $\mathcal{C}_2$ y su matriz generatriz G_2.

[5]:
```
k2=4
C2 = codes.GeneralizedReedSolomonCode(GF(11).list()[1:n+1], k2)
```

[5]:
```
[10, 4, 7] Reed-Solomon Code over GF(11)
```

[6]:
```
G2=C2.generator_matrix()
```

Definimos $\mathcal{C}_3$ y su matriz generatriz G_3.

[7]:
```
k3=2
C3 = codes.GeneralizedReedSolomonCode(GF(11).list()[1:n+1], k3)
```

[7]:
```
[10, 2, 9] Reed-Solomon Code over GF(11)
```

[8]:
```
G3=C3.generator_matrix()
```

Codificamos $\mathbf{m}$ y obtenemos nuestra palabra $\mathbf{c} \in \mathcal{C}$.

[9]:
```
m1=vector(GF(11),[0,1,2,3,4,5])
m2=vector(GF(11),[6,7,8,9])
m3=vector(GF(11),[10,0])
```

[10]:
```
c1=m1*G1
c2=m2*G2
c3=m3*G3
```

[11]:
```
c=(matrix(GF(11),[c1,c2,c3]).transpose()*A).transpose(); c
```

Definimos nuestro vector recibido $\mathbf{y} = \mathbf{c} + \mathbf{e}$.

[12]:
```
y1=c[0]+vector(GF(11),[0,3,0,2,0,0,6,0,0,7])
y2=c[1]+vector(GF(11),[0,0,0,0,0,0,0,0,0,0])
y3=c[2]+vector(GF(11),[0,0,0,0,0,0,0,0,0,0])
```

Y definimos nuestros decodificadores DC_i de Tipo 2.

[14]:
```
D1 = codes.decoders.LinearCodeNearestNeighborDecoder(C1)
D2 = codes.decoders.LinearCodeNearestNeighborDecoder(C2)
D3 = codes.decoders.LinearCodeNearestNeighborDecoder(C3)
```

Comenzamos decodificando y_1 con el Decodificador DC_1. Obtenemos la palabra decodificada $DC_1(y_1) \in \mathcal{C}_1$, y comprobamos que se han producido errores.

[15]:
```
D1.decode_to_code(y1), D1.decode_to_code(y1)==y1
```

[15]:
```
((4, 4, 2, 5, 9, 8, 3, 0, 8, 4), False)
```

Entonces, una vez decodificado y_1, tenemos que $y_1^{2)} = DC_1(y_1) = y_1 - \hat{e_1}$.

[16]:
```
y12=vector(GF(11),[4, 4, 2, 5, 9, 8, 3, 0, 8, 4])
```

Calculamos ahora $y_2^{2)} = y_2 - \frac{a_{12}}{a_{11}} y_1^{2)}$ y $A^{2)}$.

[19]:
```
y22=y2-A[0][1]/A[0][0]*y12; y22
```

[20]:
```
A2=A.transpose(); A2[1]=A2[1]-(A[0][1]/A[0][0])*A2[0]; A2=A2.
↪transpose(); A2
```

Por último, calculamos $y_3^{2)} = y_3 - \frac{a_{13}}{a_{11}} y_1^{2)}$ y actualizamos $A^{2)}$.

[21]:
```
y32=y3-(A[0][2]/A[0][0])*y12; y32
```

[22]:
```
A2=A2.transpose(); A2[2]=A2[2]-(A[0][2]/A[0][0])*A2[0]; A2=A2.
↪transpose(); A2
```

Continuamos ahora con $i_2 = 2$. Decodificamos $y_2^{2)}$ con el decodificador DC_2 y comprobamos que se han cometido errores.

[23]:
```
D2.decode_to_code(y22), D2.decode_to_code(y22)==y22
```

[23]:
```
((5, 7, 2, 1, 4, 0, 0, 4, 1, 2), False)
```

Entonces, tenemos que $y_2^{3)} = DC_2(y_2^{2)}) = y_2^{2)} - \hat{e}_2 \in \mathcal{C}_2$.

[24]:
```
y23=vector(GF(11),[5, 7, 2, 1, 4, 0, 0, 4, 1, 2])
```

Calculamos $y_3^{3)} = y_3^{3)} - \frac{a_{23}^{2)}}{a_{22}^{2)}} y_2^{3)}$ y $A^{3)}$.

[25]:
```
y33=y32-(A2[1][2]/A2[2][2])*y23; y33
```

[26]:
```
A3=A2.transpose(); A3[2]=A3[2]-(A2[1][2]/A2[2][2])*A3[1]; A3=A3.
↪transpose(); A3
```

Por último, para el índice $i_3 = 3$, decodificamos $y_3^{3)}$ con DC_3.

[27]:
```
D3.decode_to_code(y33)
```

[27]:
```
(2, 9, 5, 1, 8, 4, 0, 7, 3, 10)
```

Entonces, tenemos que $y_3^{4)} = DC_3(y_3^3) = y_3^3 - \hat{e}_3 \in \mathcal{C}_3$.

[28]:
```
y34=vector(GF(11),[2, 9, 5, 1, 8, 4, 0, 7, 3, 10])
```

Por tanto, obtenemos $y^{4)} = (y_1^{4)}, y_2^{4)}, y_3^{4)}) = (y_1^{2)}, y_2^{3)}, y_3^{4)})$.

[29]:
```
(y12,y23,y34)
```

[29]:
```
((4, 4, 2, 5, 9, 8, 3, 0, 8, 4),
 (5, 7, 2, 1, 4, 0, 0, 4, 1, 2),
 (2, 9, 5, 1, 8, 4, 0, 7, 3, 10))
```

Comprobemos ahora que la decodificación ha sido incorrecta. Si la decodificación hubiera sido correcta, $y^{4)} = [\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}] \cdot A(i_1, i_2, i_3) = [\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}] \cdot A$, con $\mathbf{c_1} \in \mathcal{C}_1, \mathbf{c_2} \in \mathcal{C}_2$ y $\mathbf{c_3} \in \mathcal{C}_3$. Como A es NSC, se tiene que $[\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}] = y^{4)} \cdot A^{-1}$.
Entonces, calculemos estas palabras y veamos si efectivamente pertenecen a sus respectivos códigos.

[36]:
```
B=Y.transpose()*A.inverse()
```

[42]:
```
[B[0] in C1, B[1] in C2, B[2] in C3]
```

[42]:
```
[False, False, False]
```

Y hemos obtenido, como era de esperar dado que el subconjunto de índices que elegimos no era el correcto, que la palabra decodificada $y^{4)}$ no pertenece a $\mathcal{C}$ (tal y como justificamos en la sección 3.3, apartado (II)).
Si eligieramos el conjunto de índices $\{i_1 = 2, i_2 = 3, i_3 = 1\}$, tal y como vimos en el ejemplo con decodificadores de tipo 1, la decodificación sería correcta.

A

Apéndice

A.1. Códigos de Reed-Solomon

Los códigos de Reed-Solomon fueron presentados por I.S. Reed y G. Solomon en 1960 [8]. Son una familia de códigos interesantes, relacionados con la evaluación de polinomios en una variable. Estos códigos se utilizan en la actualidad en dispositivos de almacenaje de datos (en CDs, DVDs, Blu-ray, DSL, WIMAX o RAID), como también en la comunicación por satélite.
Estos códigos son MDS; es decir, son códigos óptimos que, fijados $k(\mathcal{C})$ y $n(\mathcal{C})$, alcanzan la máxima $d(\mathcal{C})$ posible. Además, muy pronto se descubrieron algoritmos de decodificación eficientes que corrigen hasta la capacidad de corrección.
Debido a esto, son códigos ideales para construir un código producto de matrices a partir de ellos, dado que conocemos todos sus parámetros y tenemos un algoritmo eficiente de decodificación.

Definimos $\mathcal{L}_k$ como el conjunto de polinomios en $\mathbb{F}_q[x]$ de grado menor que k. Es decir:

$$\mathcal{L}_k = \{f \in \mathbb{F}_q[x] \mid deg(f) < k\}.$$

Es fácil comprobar que $\mathcal{L}_k$ es un espacio vectorial de dimensión k y que una base estándar de este espacio son los monomios $\{1, x, \ldots, x^{k-1}\}$. Consideremos ahora un vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}_q^n$, cuyas componentes son distintas entre ellas ($a_i \neq a_j$). Por cómo hemos definido este vector, tendremos que $n \leq q$.
Y definamos a continuación la siguiente aplicación de evaluación:

$$\begin{aligned} ev_{\mathbf{a}} : \mathcal{L}_k &\longrightarrow \mathbb{F}_q^n \\ f &\longrightarrow (f(a_1), \ldots, f(a_n)) \end{aligned}$$

que evalúa los polinomios de $\mathcal{L}_k$ en cada una de las coordenadas del vector $\mathbf{a} \in \mathbb{F}_q^n$.

Definición A.1. *Se define el **código de Reed-Solomon (RS)** de dimensión k y longitud n, asociado al vector $\mathbf{a} \in \mathbb{F}_q^n$ como la imagen del espacio $\mathcal{L}_k$ utilizando la aplicación de evaluación definida anteriormente. Es decir:*

$$RS_{q,\mathbf{a}}(k, n) = \{ev_{\mathbf{a}}(f) \mid f \in \mathcal{L}_k\}.$$

Cuando $n = q$ diremos que el código es RS en el sentido estricto.

Observación A.2. El grupo de las unidades de un cuerpo finito, $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$, es cíclico. Es decir, $\mathbb{F}_q^*$ está generado por un elemento $\alpha \in \mathbb{F}_q^*$, que denominamos elemento primitivo. Sea α un elemento primitivo de $\mathbb{F}_q$. Vamos a definir $\mathbf{a} \in \mathbb{F}_q^k$ como el vector cuyas componentes serán $a_0 = 0$ y $a_i = \alpha^i$, para $i = 1, \ldots, q-1$. Con esta definición el código $RS_{q,\mathbf{a}}(k,n)$ es RS en el sentido estricto.
Cuando utilicemos esta configuración del vector $\mathbf{a} \in \mathbb{F}_q^n$, por abuso del lenguaje, no especificaremos el vector en la aplicación evaluación, y la notación que usaremos para referirnos a los códigos RS con esta construcción será

$$RS_q(k) = \{ev(f) \mid f \in \mathcal{L}_k\} = Im(ev).$$

En la siguiente proposición, estudiaremos los parámetros del código $\mathcal{C} = RS_q(k)$, que resulta ser un código MDS (veáse Definición 1.39).

Proposición A.3. *Sea $\mathcal{C} = RS_q(k)$. Entonces, $\mathcal{C}$ es un $[n,k]$-código. Además, se tiene que $\mathcal{C}$ es MDS; es decir, la distancia mínima de $\mathcal{C}$ es $n-k+1$.*

Demostración. Por definición, la longitud de $RS_q(k)$ es n, que es la longitud del vector $\mathbf{a} \in \mathbb{F}_q^n$. Como mencionamos anteriormente, $\mathcal{L}_k$ es un espacio vectorial de dimensión k. Luego, para ver que $\mathcal{C}$ es de dimensión k nos basta con comprobar que la aplicación $ev : \mathcal{L}_k \longrightarrow \mathbb{F}_q^n$ es inyectiva.
Supongamos que existen dos polinomios distintos $f, g \in \mathcal{L}_k$, tales que $f(a) = g(a)$, $\forall a \in \mathbb{F}_q$. Entonces, tendremos que $p = f - g \in \mathcal{L}_k$ será un polinomio de grado $< k$ con n raíces. Con lo cual, por el Teorema Fundamental del Álgebra [1], tenemos que $p = 0$. Luego $f = g$; por tanto la aplicación ev es inyectiva.
Veamos ahora que $d_H(\mathcal{C}) = n-k+1$. Por la cota de Singleton (Corolario 1.38) tenemos que $d \geq n-k+1$. Además, una palabra del código $c \in RS_q(k)$ cumple que $w_H(c) \geq n-k+1$. Esto se debe a que un polinomio de $\mathcal{L}_k$ tendrá a lo sumo $k-1$ raíces. Por tanto se tiene que $d = n-k+1$.

□

Observación A.4. Si $\mathbf{a} = (x_1, \ldots, x_n)$, la siguiente matriz de Vandermonde

$$G = \begin{pmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_1^{k-1} & \cdots & x_n^{k-1} \end{pmatrix} \in \mathbb{F}_q^{k \times n}$$

es una matriz generatriz del código $RS_{q,\mathbf{a}}(k,n)$.
Si $\mathbf{a} = (0, 1, \alpha, \ldots, \alpha^{q-2})$, una matriz generatriz del código $RS_q(k)$ es de la forma

$$G = \begin{pmatrix} ev(1) \\ ev(x) \\ ev(x^2) \\ \vdots \\ ev(x^{k-1}) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & \alpha & \cdots & \alpha^{q-2} \\ 0 & 1^2 & \alpha^2 & \cdots & \alpha^{(q-2)2} \\ \vdots & & & \ddots & \vdots \\ 0 & 1^{k-1} & \alpha^{k-1} & \cdots & \alpha^{(q-2)(k-1)} \end{pmatrix} \in \mathbb{F}_q^{k \times n}.$$

[1] Teorema Fundamental del Álgebra: Todo polinomio de grado n tiene, contando multiplicidades, exactamente n raíces complejas.

A.2. Algoritmo eficiente de decodificación por Mínima Distancia para Códigos Reed-Solomon

El objetivo de esta sección es el de proporcionar un algoritmo eficiente para la decodificación de códigos Reed-Solomon. En esta memoria adaptaremos el algoritmo de decodificación presentado por Berlekamp y Welch en 1968 [2], basado en la idea de interpolación de Lagrange.
Consideremos el código $\mathcal{C} = RS_q(k)$. Supongamos que enviamos una palabra del código $\mathbf{c} = ev(f) \in \mathcal{C}$, con $f \in \mathcal{L}_k$, y que recibimos el vector $\mathbf{y} = \mathbf{c} + \mathbf{e}$, donde $\mathbf{e} \in \mathbb{F}_q^n$ es el vector de errores producidos durante la transmisión. Vamos a suponer que el número de errores es menor que la capacidad de corrección de $\mathcal{C}$, que por la Proposición 1.44, sabemos que es $\lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$. Es decir, $w_H(\mathbf{e}) \leq t_{RS} = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor = \lfloor \frac{n-k}{2} \rfloor$.
Vamos a describir un algoritmo que corrige t_{RS} errores. El algoritmo considera el vector $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$, y nos devuelve un polinomio $f \in \mathbb{F}_q[x]$ con $deg(f) < k$ tal que $d_H(\mathbf{y}, ev(f)) = d_H(\mathbf{y}, RS_q(k))$; o bien nos devuelve un error de decodificación si el número de errores que se ha producido durante la comunicación son $> t_{RS}$.
Por tanto, tenemos un algoritmo de decodificación que corrige t_{RS} errores.

Input: Recibimos $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_q^n$. Suponemos $d_H(\mathbf{y}, \mathcal{C}) \leq t$, con $\mathcal{C} = RS_q(k)$.
Output: $\mathbf{c} \in \mathcal{C}$ tal que $d_H(\mathbf{y}, \mathbf{c}) = d_H(\mathbf{y}, \mathcal{C})$.

Sea $\mathbf{a} = (a_1, \ldots, a_n) = (0, 1, \ldots, \alpha^{q-2})$ con $\mathbb{F}_q^* = < \alpha >$
Consideramos los polinomios:

$$\begin{cases} E(x) = x^t + \sum_{i=0}^{t-1} A_i x^i \in \mathbb{F}_q[A_0, \ldots, A_{t-1}][x] \\ P(x) = \sum_{i=0}^{t+k-1} B_i x^i \in \mathbb{F}_q[B_0, \ldots, B_{t+k-1}][x] \end{cases} \tag{A.1}$$

Resolvemos el siguiente sistema de ecuaciones lineales en las variables A_i, B_j, para $j \in \{0, \ldots, t+k-1\}$, $i \in \{0, \ldots, t-1\}$.

$$P(a_l) - E(a_l) y_l = 0, \quad \text{con } l \in \{1, \ldots, n\}. \tag{A.2}$$

if (A.2) *es un Sistema Incompatible* **then**
  return *ERROR DE DECODIFICACIÓN*
end
Sea $A_i = \lambda_i$ y $B_j = \beta_j$ una solución del sistema (A.2), con $j \in \{0, \ldots, t+k-1\}$, $i \in \{0, \ldots, t-1\}$
else
  $f(x) = \frac{P(\beta_0, \ldots, \beta_{t+k-1})(x)}{E(\lambda_0, \ldots, \lambda_{t-1})(x)}$, y $\mathbf{c} = ev(f)$
  if $d(\mathbf{y}, \mathbf{c}) > t$. **then**
    return *ERROR DE DECODIFICACIÓN*
  end
  else
    return $\mathbf{c} = ev(f) \in \mathcal{C}$
  end
end

Algoritmo 2: Decodificación de Código Reed-Solomon Dec_{RS}

Primero veamos un resultado necesario para verificar el correcto funcionamiento del algoritmo descrito.

Proposición A.5. *Sea $E(x) \in \mathcal{L}_t$ un polinomio mónico, y sea $P(x) \in \mathcal{L}_{t+k-1}$. Entonces, si $d_H(\mathbf{y}, RS_q(k)) \leq t$, el sistema de ecuaciones lineales (A.2) es un sistema compatible.*

Demostración. Suponemos por hipótesis que $d_H(\mathbf{y}, RS_q(k)) \leq t$. Entonces, existe un polinomio $g \in \mathcal{L}_k$ tal que $d_H(\mathbf{y}, ev(g)) \leq t$. Sea I el conjunto de índices donde $ev(g)$ e $\mathbf{y}$ no coinciden. Es decir, $I = \{i_1, \ldots, i_s\} = \{i \mid y_i \neq g(a_i)\}$.
Definimos $E(x) = \prod_{j=1}^{s}(x - a_i)x^{t-s}$ y $P(x) = E(x)g(x)$. Observamos que $E(x) \in \mathcal{L}_t$ y es mónico, y $P(x) \in \mathcal{L}_{t+k-1}$, pues $g(x) \in \mathcal{L}_k$. Además $P(a_i) = E(a_i)y_i \ \forall i = 1, \ldots, n$ ya que:

- si $i \in I$, entonces $E(a_i) = 0$
- si $i \notin I$, entonces $g(a_i) = y_i$. Por tanto $E(a_i)g(a_i) = E(a_i)y_i$

Luego hemos encontrado $E(x), P(x)$ que satisfacen las condiciones del sistema.

□

Observación A.6. Por hipótesis, hemos impuesto que $E(x)$ sea mónico de grado t.
Esto se debe a que, si no forzaramos esta condición, el coeficiente que acompaña a x^t podría ser 0. Si esto ocurriera, como $E(x)g(x) = E(x)y_i$, tendríamos como solución $g(x) \in \mathbb{F}_q[x]$ con $deg(g) > k$, con lo cual, tendríamos una solución que no es la buscada, pues $g \notin \mathcal{L}_k$, o lo que es lo mismo, $ev(g) \notin RS_q(k)$.

Probemos a continuación que efectivamente el Algoritmo 2 es correcto.

Proposición A.7. *El algoritmo Dec_{RS} (Algoritmo 2) es un algoritmo de decodificación para el código $RS_q(k)$ que corrige $t = t_{RS} = \lfloor \frac{n-k}{2} \rfloor$ errores.*

Demostración. Sean los polinomios $E(x)$ y $P(x)$ definidos en (A.1) una solución no nula del sistema (A.2).
Sea $\mathbf{y} \in \mathbb{F}_q^n$ el vector recibido. Buscamos $f \in \mathcal{L}_k$ tal que $d_H(\mathbf{y}, ev(f)) = d_H(\mathbf{y}, RS_q(k))$.
Consideremos el conjunto de índices $I = \{i \in \{1, \ldots, n\} \mid y_i \neq f(a_i)\}$. Para cualquier $i \in \{1, \ldots, n\} \setminus I$, se tiene que

$$P(a_i) - E(a_i)f(a_i) = P(a_i) - E(a_i)y_i = 0.$$

Luego, puesto que, $ev(f) \neq \mathbf{y}$ en, a lo sumo, t posiciones por hipótesis, tenemos que $P(x) - E(x)f(x)$ tiene al menos $n - t$ ceros. Y además, $deg(P(x)) - E(x)f(x) \leq t + n - 1$.
Por otra parte, por la Proposición 1.44, tenemos que $t = \lfloor \frac{n-k}{2} \rfloor < \frac{n-k+1}{2}$. Luego, $t + k - 1 < n - t$.

Luego, $P(x) - E(x)f(x) = 0, \forall x \in \mathbb{F}_q$ y además $deg(P(x)) = deg(E(x)f(x)) = t + k - 1 < q$. De donde se deduce que $f(x) = \frac{P(x)}{E(x)}$ es la solución buscada.

□

Bibliografía

[1] E. Berlekamp, R.J. McEliece, H.C.A van Tilborg. *On the inherent intractability of certain coding problems. IEEE Transactions on Information*. Theory IT. Vol 24, pp. 384-386, 1978.

[2] E. Berlekamp, L. Welch. *Error Correction of Algebraic Block Codes*. US Patent 4.633.470, 1968.

[3] T. Blackmore, G.H. Norton. *Matrix-Product Codes over* $\mathbb{F}_q$. Applicable Algebra in Engineering, Communication and Computing. Vol 12 (6), pp. 477-500, 2001.

[4] S. K. Chebolu, J. Mináč. *Counting irreducible polynomials over finite fields using the Inclusion-Exclusion Principle*. Mathematics magazine. Vol 84 (5), pp. 369-370, 2011.

[5] F. Hernando, K. Lally, D. Ruano. *Construction and decoding of matrix-product codes from nested codes*. Applicable Algebra in Engineering, Communication and Computing. Vol 20 (497), 2009.

[6] J. Justesen, T. Høholdt. *A Course In Error-Correcting Codes*. European Mathematical Society Publishing House, 2004.

[7] R. Pellikaan, X. Wu, S. Bulygin, R. Jurrius. *Codes, Cryptology and Curves with Computer Algebra*. Cambridge University Press, 2018.

[8] I.S. Reed, G. Solomon. *Polynomial Codes Over Certain Finite Fields.* Journal of Society for Industrial and Applied Mathematics, Vol. 8 (2) ,pp. 300-304, 1960.

[9] *SageMath*. https://www.sagemath.org/. The Sage Mathematics Software System (Version 9.1).

Matrix-Product Codes

Sección de Matemáticas
Universidad de La Laguna

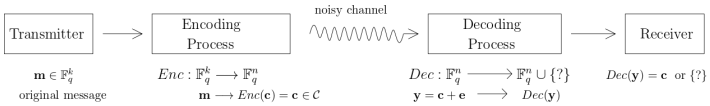
Alba Santos Rodríguez
Facultad de Ciencias · Sección de Matemáticas
Universidad de La Laguna
alu0100953657@ull.edu.es

Abstract

Coding theory is a branch of mathematics that deals with the problem of a message's efficient transmission. In this work, we will talk about Error-correcting codes. These codes that are capable of detecting and correcting errors that may have occurred during the transmission of the message through a noisy channel. Particularly, we will focus on Linear Codes, a family of codes which have an efficient encoding, defined by a linear map. We will study their properties, and later on, we will define some interesting code constructions. Hereafter, we will generalize the previous ones, defining the Matrix Product Codes. This construction allows us to create a new, longer code from tinier linear codes. These codes' parameters will be defined by the parameters of the smaller codes and the matrix. Finally, we will describe an efficient decoding algorithm for these type of codes that corrects the maximum possible number of errors.

Introduction

The rise of interest in communication technologies during the 20th century brought two mathematicians, **Shannon** and **Hamming**, to lay the foundations of Coding Theory. They defined a communication process with an alphabet $\mathcal{A} = \mathbb{F}_q$, where $\mathbb{F}_q$ stands for the finite field with q elements.



In this process, we want to send a message $\mathbf{m} \in \mathbb{F}_q^k$ through a noisy channel, and be able to correct the errors that may happen during the communication. To do so, we **encode** our message $\mathbf{m} \in \mathbb{F}_q^k$ through an injective map, and we obtain a vector $\mathbf{c} \in \mathbb{F}_q^n$. The image of this injective map is the **code** $\mathcal{C}$ and its elements $\mathbf{c}$ are called **codewords**. After that, we send the message. The receiver will get a vector $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_q^n$, where $\mathbf{e} \in \mathbb{F}_q^n$ stands for the errors that have happened during the transmission. To correct them, we have a **decoding process**. Coding theory is used in multiple situations: from detecting a mistake in your ID number, or allowing you to read a CD correctly, to spatial communications with satellites.

1. Coding Theory

Definition 1. *For $\mathbf{x} = (x_1, \dots, x_n), \mathbf{y} = (y_1, \dots, y_n) \in \mathbb{F}_q^n$, the **Hamming distance** is defined as $d_H(\mathbf{x}, \mathbf{y}) = \#\{i \mid x_i \neq y_i\}$.*

Definition 2. *A linear code $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$. The parameters of the code will be length $n(\mathcal{C}) = n$, dimension $k(\mathcal{C}) = dim(\mathcal{C})$, and minimum distance $d(\mathcal{C}) = \min\{d_H(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in \mathcal{C}\}$.*
We will refer to a linear code $\mathcal{C}$ with the above parameters as a $[n, k, d]_q$-code.

Definition 3. *A matrix $G \in \mathbb{F}_q^{k \times n}$ is called a **generator matrix** of a $[n, k, d]_q$-code $\mathcal{C}$ if the rows of G are a basis of $\mathcal{C}$.*

Definition 4. *This matrix allows us to define an **encoding process** for linear codes, through the following linear map:*

$$\begin{aligned} Enc : \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ m &\longrightarrow m \cdot G = c \in \mathcal{C} \end{aligned}$$

Definition 5. *Let $\mathbf{c} \in \mathcal{C}$ be the sent codeword, and $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_q^n$ the received vector. A **minimum distance decoder** $\mathcal{D}$ for the code $\mathcal{C}$, that corrects s errors, is defined as the following map:*

$$\mathcal{D} : \mathbb{F}_q^n \longrightarrow \mathcal{C} \cup \{?\}$$

*where $\mathcal{D}(\mathbf{y}) = \mathbf{c}$ if there's a **unique** codeword $\mathbf{c} \in \mathcal{C}$ that satisfies $d_H(\mathbf{y}, \mathcal{C}) = \min\{d_H(\mathbf{y}, \mathbf{c}) \mid \mathbf{c} \in \mathcal{C}\} \leq s$.*

If $d_H(\mathbf{y}, \mathcal{C}) > s$, or if there are two codewords $\mathbf{c_1}, \mathbf{c_2} \in \mathcal{C}$ that satisfy $d_H(\mathbf{y}, \mathcal{C}) = d_H(\mathbf{y}, \mathbf{c_1}) = d_H(\mathbf{y}, \mathbf{c_2})$, the decoder will return $\mathcal{D}(\mathbf{y}) = \{?\}$ as a response.

2. Matrix-Product Codes

In order to find bigger linear codes for which we have a lower bound for the minimum distance $d(\mathcal{C})$, we will define a new code $\mathcal{C}$ from tinier linear codes $\mathcal{C}_1, \dots, \mathcal{C}_M \subset \mathbb{F}_q^n$ and a matrix $A = (a_{ij}) \in \mathbb{F}_q^{M \times N}$. This new code $\mathcal{C}$ will be called **matrix-product code**. Its parameters will be determined by the ones of the $\mathcal{C}_i$ codes, and the characteristics of the matrix A.

Definition 6. *The **matrix-product code** $\mathcal{C} = [\mathcal{C}_1 \cdots \mathcal{C}_M] \cdot A$ is the set of all matrix-products $[c_1 \cdots c_M] \cdot A$, where $c_i = (c_{1,i}, \dots, c_{n,i}) \in \mathcal{C}_i$. Therefore, a typical codeword $\mathbf{c} \in \mathcal{C}$ is*

$$\mathbf{c} = \begin{pmatrix} c_{11}a_{11} + c_{12}a_{21} + \dots + c_{1M}a_{M1} & \dots & c_{11}a_{1N} + \dots + c_{1M}a_{MN} \\ \vdots & \ddots & \vdots \\ c_{n1}a_{11} + c_{n2}a_{21} + \dots + c_{nM}a_{M1} & \dots & c_{n1}a_{1N} + \dots + c_{nM}a_{MN} \end{pmatrix}.$$

Example 7. *Let $\mathcal{C}_i$ be a $[n, k_i, d_i]$-code, for $i = 1, 2$. The **Plotkin construction** is defined as $\mathcal{C} = \{(u \mid u + v) \mid u \in \mathcal{C}_1, v \in \mathcal{C}_2\}$. This construction could be seen as a matrix-product code*

$$\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2] \cdot A, \text{ where } A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Definition 8. *We call A **non singular by columns (NSC)** if $A(j_1, \dots, j_t)$ is non singular for each $1 \leq t \leq M$ and $1 \leq j_1 < \dots < j_t \leq N$.*

Theorem 9. *If A is NSC and $\mathcal{C} = [\mathcal{C}_1, \cdots, \mathcal{C}_M] \cdot A$ then*

1. *$\#\mathcal{C} = \#\mathcal{C}_1 \cdots \#\mathcal{C}_M$.*
2. *$d(\mathcal{C}) \geq d^* = \min\{Nd_1, (N-1)d_2, \dots, (N-M+1)d_M\}$.*
3. *If A is also triangular then $d(\mathcal{C}) = d^*$.*

Theorem 10. *Let $\mathcal{C}_1, \dots, \mathcal{C}_M$ be nested codes, i.e., $\mathcal{C}_1 \supseteq \dots \supseteq \mathcal{C}_M$, and $\mathcal{C} = [\mathcal{C}_1 \cdots \mathcal{C}_M] \cdot A$ the matrix-product code. Then $d(\mathcal{C}) = \min\{d_1 D_1, \dots, d_M D_M\}$, where $d_i = d(\mathcal{C}_i)$, and $D_i = d(C_{R_i})$.*

Remark. *C_{R_i} stands for the code with generator matrix A_i, where A_i is compounded by the i first rows of A.*

3. Decoding Matrix-Product Codes from nested codes

Input: We receive $\mathbf{y} = \mathbf{c} + \mathbf{e}$, where $\mathbf{c} \in \mathcal{C}$ and $\mathbf{e} \in \mathbb{F}_q^{nN}$, with $w(\mathbf{e}) \leq t_{\mathcal{C}}$.
Let $A \in \mathbb{F}_q^{M \times N}$ be a NSC matrix and $\mathcal{C}_M \subset \cdots \subset \mathcal{C}_1$ be nested linear codes.
Let $DC_1, \dots, DC_M$ be the decoders.

1 $\mathbf{y}' = \mathbf{y}$, $A' = A$
2 **for** $\{i_1, \dots, i_M\} \subset \{1, \dots, N\}$ **do**
3 $\quad \mathbf{y} = \mathbf{y}'$, $A = A'$
4 $\quad$ **for** $j = 1, \dots M$ **do**
5 $\quad\quad y_{i_j} = DC_j(y_{i_j})$
6 $\quad\quad$ **if** $y_{i_j} =$ *"failure"* **then**
7 $\quad\quad\quad$ **break** Consider another $i_1, \dots, i_M$ in line 2
8 $\quad\quad$ **end**
9 $\quad\quad$ **for** $k = j+1, \dots M$ **do**
10 $\quad\quad\quad y_{i_k} = y_{i_k} - \frac{a_{j,i_k}}{a_{j,i_j}} y_{i_j}$
11 $\quad\quad\quad$ column $_{i_k}(A) =$ column $_{i_k}(A) - \frac{a_{j,i_k}}{a_{j,i_j}}$column$_{i_j}(A)$
12 $\quad\quad$ **end**
13 $\quad$ **end**
14 $\quad$ Obtain $(c_1, \dots, c_M)$
15 $\quad \mathbf{y} = [c_1, \dots, c_M] \cdot A$
16 $\quad$ **if** $\mathbf{y} \in \mathcal{C}$ *and* $w(\mathbf{y} - \mathbf{y}') \leq \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$ **then**
17 $\quad\quad$ **return** $\mathbf{y}$
18 $\quad$ **end**
19 $\quad$ **else**
20 $\quad\quad$ **break** Consider another $i_1, \dots, i_M$ in line 2
21 $\quad$ **end**
22 **end**

Decoding algorithm for Matrix-Product Codes from Nested Codes

Theorem 11. *This algorithm is a minimum distance decoding algorithm for a matrix-product code $\mathcal{C} = [\mathcal{C}_1 \cdots \mathcal{C}_M] \cdot A$, that corrects up to $t_{\mathcal{C}} = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$ errors.*

References

[1] T. Blackmore, G.H. Norton. *Matrix-Product Codes over $\mathbb{F}_q$*. Applicable Algebra in Engineering, Communication and Computing. Vol 12 (6), pp. 477-500, 2001.

[2] F. Hernando, K. Lally, D. Ruano. *Construction and decoding of matrix-product codes from nested codes*. Applicable Algebra in Engineering, Communication and Computing. Vol 20 (497), 2009.