



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

UNIDAD-IZTAPALAPA

# Algunos aspectos algebraicos de códigos de convolución

T E S I S

QUE PARA OBTENER EL GRADO ACADÉMICO DE

MAESTRA EN CIENCIAS

(MATEMÁTICAS APLICADAS E INDUSTRIALES)

PRESENTA:

Perla Araceli Maldonado Cortez

DIRECTOR DE TESIS: Dr. Horacio Tapia Recillas

CIUDAD DE MÉXICO, FEBRERO DE 2017

# Índice general

<b>1. Nociones algebraicas básicas asociadas</b>	<b>6</b>
1.1. Grupos . . . . .	6
1.2. Anillos . . . . .	15
1.3. Campos finitos . . . . .	21
1.3.1. Espacio vectorial . . . . .	26
1.4. Forma normal de Smith . . . . .	27
1.5. Espacio de sucesiones . . . . .	33
<b>2. Ecuaciones de paridad y polinomios generadores</b>	<b>36</b>
2.1. La operación de convolución . . . . .	36
2.2. Códigos de convolución . . . . .	39
2.3. Conceptos preliminares de códigos de convolución . . . . .	40
2.4. Polinomios generadores . . . . .	43
2.5. Ecuaciones de paridad . . . . .	46
2.6. Polinomios generadores obtenidos de ecuaciones de paridad . . . . .	56
2.6.1. Ejemplos . . . . .	59
<b>3. Matrices generadoras y forma normal de Smith</b>	<b>63</b>
3.1. Matrices generadoras . . . . .	63
3.2. Ejemplo de forma normal de Smith . . . . .	69
3.3. Matrices catastróficas . . . . .	75
3.4. Códigos sistemáticos . . . . .	79
3.5. Matrices generadoras formadas a partir de sucesiones generadoras. . . . .	81
<b>4. Decodificación</b>	<b>85</b>
4.1. Diagramas de trellis . . . . .	85
4.2. Distancia libre $d_{free}$ . . . . .	96

4.3. Algoritmo de Viterbi . . . . .	99
4.4. Códigos de convolución MDS . . . . .	109

## Introducción

Al transmitir información de una fuente emisora a otra receptora, es importante que el receptor reciba la misma que el emisor envió; si no sucede así, que por lo menos sea capaz de detectar los errores que ocurrieron en la transmisión. Aunque se puede ir más allá de la detección, se tiene la capacidad de corregir y recuperar la información original, una forma de hacerlo es por medio de los códigos detectores-correctores de errores, [1].

Debido a que somos una *sociedad de información* y necesitamos métodos que garanticen la fiabilidad de esta, el estudio de los códigos detectores-correctores adquiere gran importancia. C. Shannon (1948) y R. Hamming (1950) fueron precursores en el estudio de estos códigos, [26]. Existen dos grandes grupos de códigos detectores-correctores de errores; los códigos de bloque y los de cascada. Algunos ejemplos importantes de los de bloque son los *códigos de Hamming* (R.W. Hamming 1950, [28]) y los *códigos Reed-Solomon* (I. Reed y G. Solomon 1960, [29]), mientras que los *códigos de convolución* (P. Elias 1955, [9]) son un ejemplo, también importante, de códigos de cascada.

P. Elias en su artículo “Coding for Noisy Channels” 1955, [9], demostró que los códigos lineales en forma de cascada en los canales binarios simétricos son suficientes para explotar al máximo dicho canal. En este mismo trabajo se introducen los “Códigos de convolución”, con los cuales es posible el uso de estos códigos con longitud de restricción infinita para transmitir información con una tasa igual a la capacidad del canal con probabilidad cercana a uno de que ningún símbolo decodificado sea erróneo. Los códigos de convolución fueron introducidos por Elias como una alternativa a los códigos de bloque, debido a su simplicidad de generación y a su pequeño registro de desplazamiento [4], [30].

Los códigos de convolución heredan su nombre de la operación de *convolución* discreta, debido a que se generan a través de esta. Son códigos lineales que tienen varias similitudes con los códigos de bloque, aunque difieren en varios aspectos. A diferencia de los códigos de bloque, en los códigos de convolución la codificación depende tanto de la información de entrada como de información almacenada, que se denomina *memoria*, [3]. En principio, cada secuencia de longitud  $b$  el codificador los transforma en secuencias de longi-

tud  $c$ ,  $b \leq c$ , que determina la *tasa* de transmisión,  $R = b/c$ . En la práctica, estas secuencia de información están formadas de bits, es decir, elementos del campo binario  $\mathbb{F}_2 = \{0, 1\}$ .

En los códigos de bloque la distancia mínima (de Hamming) proporciona información sobre el número de errores que el código puede detectar. En el caso de códigos de convolución la distancia libre,  $d_{free}$ , juega un papel similar: mide la capacidad para decodificar una secuencia de información recibida, la cual puede ser de longitud infinita [3]. En este caso se dirá que un código de convolución tiene parámetros  $(c, b, m, d_{free})$ .

Por sus características, los códigos de convolución resultan muy útiles en áreas de telecomunicaciones, electrónica, acústica, y sistemas de comunicación digital [6], [13]; por ejemplo en comunicaciones inalámbricas (IMT-2000, GSM, IS-95), móviles, satélital y espacial, dispositivos de almacenamiento y televisión de alta definición, entre otros [7], [14], [24].

Los códigos de convolución, como muchos otros elementos que tienen aplicación práctica, están sustentados en conceptos matemáticos. El propósito de este trabajo es destacar varios de esos conceptos, particularmente algebraicos, los cuales incluyen la operación de convolución, matrices sobre un anillo, especialmente el anillo de polinomios con coeficientes en un campo finito, de funciones racionales o bien de series de Laurent. El hecho que el anillo de polinomios sobre un campo finito es euclideo es de gran relevancia, ya que entre otras cosas, la forma Normal de Smith de una matriz sobre este anillo tiene una representación muy particular y práctica.

Al igual que en los códigos de bloque, en los de convolución la decodificación es de gran importancia. Existen varios métodos, pero en este trabajo solo mostramos a grandes rasgos el algoritmo de Viterbi (Andrew Viterbi, 1967), [12], que es el método de decodificación más usado y hasta ahora más eficiente en estos códigos. Este algoritmo propone un esquema de máxima verosimilitud de probabilidad basado en el diagrama de trellis, el algoritmo de Viterbi encuentra las sucesiones de estados más probables.

Este trabajo se divide en 4 capítulos, distribuidos de la siguiente manera:

En el primer capítulo se darán algunos de los antecedentes algebraicos

básicos asociados a los códigos de convolución.

En el Capítulo 2 se define la operación de convolución discreta, se abordan los conceptos básicos de los códigos de convolución, además se introducirán los polinomios generadores, ecuaciones de paridad, y algunas de sus propiedades.

Las matrices generadoras, la forma normal de Smith y cómo esta ayuda a reducir el proceso de codificación son tratadas en el Capítulo 3, donde además se abordan las matrices catastróficas y por qué deben ser evitadas. Incluyendo cómo pasar de una matriz con entradas en un anillo polinomial a una matriz sistemática, que es una matriz con entradas racionales.

Además de su descripción por medio de matrices, los códigos de convolución tienen otras representaciones. El capítulo 4 está dedicado a la construcción de varios de estas representaciones por medio de diagramas (de estados). Los diagramas de trellis también se abordarán, los cuales son de gran importancia en el algoritmo de decodificación de Viterbi, con el cual se finalizará.

Por último, se agrega un apéndice en el cual se hacen algunos ejemplos de codificación y decodificación de códigos de convolución usando MATLAB.

Queda mucho por hacer en el estudio de los fundamentos algebraicos de los códigos de convolución, estudio que es relevante, ya que el desarrollo de la información digital y en comunicaciones tiene aplicaciones a necesidades de la vida actual.

# Capítulo 1

## Nociones algebraicas básicas asociadas

En este capítulo recordaremos algunas de las nociones algebraicas básicas asociadas a los códigos de convolución, debido a que se convierten en herramientas fundamentales para el mejor entendimiento de estos códigos. Entre estas nociones se encuentran: grupos, anillos, campos, espacios vectoriales y matrices, además de la operación de convolución.

### 1.1. Grupos

En esta sección se recordarán algunos conceptos básicos de grupos que incluyen, definición, homomorfismos, isomorfismos, grupos cocientes, y se darán algunos ejemplos para ilustrar los conceptos. Para mayores detalles se puede consultar, por ejemplo, [7], [10], [16].

Un **grupo** es un conjunto de elementos,  $G$ , en el cual se define una operación " $*$ ",

$$\begin{aligned} * : G \times G &\longrightarrow G \\ (g_1, g_2) &\longrightarrow g_1 * g_2, \end{aligned}$$

tal que se satisfacen las siguientes propiedades:

(i) *Asociativa*:  $(g_1 * g_2) * g_3 = g_1 * (g_2 * g_3)$ , para todo  $g_1, g_2, g_3 \in G$ .

(ii) *Neutro*: existe  $e \in G$  tal que para todo  $g \in G$ ,

$$e * g = g = g * e.$$

Al cual se le llama *elemento neutro* y es único.

(iii) *Inverso*: para todo  $g \in G$  existe  $g^{-1} \in G$  tal que:

$$g * g^{-1} = e = g^{-1} * g.$$

Al elemento  $g^{-1}$  se le llama el *inverso* de  $g$  y es único.

Si para todo elemento  $g, h \in G$  se satisface que:

$$g * h = h * g,$$

se dice que el grupo  $G$  es *abeliano* o *conmutativo*.

Si la operación del grupo se denota por “+”,  $(G, +)$  es llamado *grupo aditivo*, en el caso de ser la multiplicación es un *grupo multiplicativo*.

El orden,  $ord(G)$ , de un grupo  $(G, *)$  es la cardinalidad del conjunto  $G$  la cual puede ser finita o infinita.

A continuación se mencionan algunos ejemplos de grupos. Para el propósito de este trabajo los siguientes son importantes. Sea

$$\mathbb{F}_2 := \{0, 1\},$$

en este conjunto se puede definir la operación suma y multiplicación como en las siguientes tablas:

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

En disciplinas como telecomunicaciones, computación, ingeniería, entre otras los elementos de  $\mathbb{F}_2$  reciben el nombre de *bits*.



1)  $(\mathbb{F}_2, +)$ : la suma de bits.

2)  $(\mathbb{F}_2^*, *)$ , donde  $\mathbb{F}_2^* = \mathbb{F}_2 - \{0\}$ .

3)  $(\mathbb{F}_2^n, +)$ , donde  $\mathbb{F}_2^n = \{a = (a_1, \dots, a_n) : a_i \in \mathbb{F}_2, i = 1, \dots, n\}$ : las  $n$ -adas de bits con la operación suma, coordenada a coordenada: si  $a, b \in \mathbb{F}_2^n$ ,

$$a + b = (a_1, \dots, a_n) + (b_1, \dots, b_n) = (c_1, \dots, c_n), c_i = (a_i + b_i) \in \mathbb{F}_2.$$

4)  $(\mathbb{F}_2[x], +)$ , donde  $\mathbb{F}_2[x] = \{\sum_{i=0}^n f_i x^i : f_i \in \mathbb{F}_2\}$ : el conjunto de polinomios con coeficientes binarios con la operación usual de suma de polinomios:

$$\sum_{i=0}^n f_i x^i + \sum_{i=0}^m g_i x^i = \sum_{i=0}^{s=\max\{n,m\}} h_i x^i : f_i, g_i \in \mathbb{F}_2, h_i = f_i + g_i \in \mathbb{F}_2.$$

5)  $(\mathbb{Z}, +)$ : el grupo de los números enteros con la suma usual.

Los ejemplos anteriores son grupos abelianos.

6) El conjunto de matrices  $n \times n$  con entradas, por ejemplo en  $\mathbb{F}_2$  o en  $\mathbb{Z}$ , con la operación de multiplicación de matrices, es un grupo no-conmutativo.

Un ejemplo importante de grupo es el **cíclico**. Un grupo  $(G, *)$  es cíclico si existe un elemento  $g \in G$  tal que cualquier elemento de  $G$  es una potencia (múltiplo) de ese elemento:

$$G = \langle g \rangle = \{g^n : n \in \mathbb{Z}\},$$

al elemento  $g$  con esta propiedad se le llama un generador del grupo cíclico.

**Un homomorfismo** entre grupos  $(A, +)$  y  $(B, *)$ , es una función

$$f : (A, +) \longrightarrow (B, *),$$

que preserva las operaciones de éstos grupos, es decir,

$$f(a + b) = f(a) * f(b).$$

Un par de ejemplos de homomorfismo entre grupos son los siguientes:

1)

$$f : (\mathbb{R}, +) \longrightarrow (\mathbb{R}^*, *)$$

$$f(x) = e^x,$$

donde  $(\mathbb{R}, +)$  es el grupo aditivo de los números reales y  $(\mathbb{R}^*, *)$  es el grupo multiplicativo, donde  $\mathbb{R}^* = \mathbb{R} - \{0\}$ . Es inmediato ver que  $f$  es un homomorfismo:  $f(x + y) = e^{x+y} = e^x e^y = f(x)f(y)$ .

2)

$$f : (\mathbb{F}_2^n, +) \longrightarrow (\mathbb{F}_2[x], +)$$

$$f(a_0, a_1, \dots, a_{n-1}) \longrightarrow a_0 + a_1x + \dots + a_{n-1}x^{n-1},$$

si  $a, b \in \mathbb{F}_2^n$ ,

$$f(a + b) = f(a_0 + b_0, \dots, a_{n-1} + b_{n-1}) = (a_0 + b_0) + \dots + (a_{n-1} + b_{n-1})x^{n-1} = (a_0 + \dots + a_{n-1}x^{n-1}) + (b_0 + \dots + b_{n-1}x^{n-1}) = f(a) + f(b).$$

Obsérvese que en este caso  $f$  no es biyectiva. En general cuando la función  $f$  que define el homomorfismo entre grupos es biyectiva, se dice que es un **isomorfismo**, lo que significa que ambos grupos tienen la misma estructura algebraica, es decir, son isomorfos.

Un par de ejemplos de isomorfismos entre grupos son los siguientes:

1)

$$f : G \longrightarrow G$$

$$x \longrightarrow x^{-1},$$

$G$  un grupo abeliano. Sean  $x, y \in G$  tal que con la multiplicación:

$$f(xy) = (xy)^{-1} = y^{-1}x^{-1} = x^{-1}y^{-1} = f(x)f(y),$$

y como  $f$  es biyectiva, entonces es un isomorfismo.

2)

Si  $A$  es un grupo abeliano finito de orden  $n$ , y  $m.c.d(n, k) = 1$ ,

$$f : A \longrightarrow A$$

$$f(a) \longrightarrow a^k,$$

ya que

$$f(a^d) = (a^d)^k = a^{kd} a^{nc} = a^{kd+nc} = a^1 = a,$$

ya que  $(a^c)^n = e$ , donde  $e$  es elemento neutro de  $A$ , y  $kd + nc = 1$ .

Un subconjunto,  $H$  de un grupo  $G$ , se dice que es un **subgrupo** de  $G$  si con la misma operación que  $G$ ,  $H$  es un grupo. Se puede probar que esto equivale a que:

$$e_G \in H$$

y

$$ab^{-1} \in H,$$

para todo  $a, b \in H$ .

Un ejemplo general de subgrupo. Si  $k$  es un entero fijo, entonces:

$$k\mathbb{Z} = \{kn : n \in \mathbb{Z} = \dots, -2k, -k, 0, k, 2k, \dots\}$$

es un subgrupo de los enteros bajo la adición. Si  $k = 0$ , entonces obtenemos el subgrupo trivial  $\{0\}$ , mientras que si  $k = 1$  obtenemos el grupo  $\mathbb{Z}$ , por lo cual con  $k = 1$ ,  $k\mathbb{Z}$  es un subgrupo propio de  $\mathbb{Z}$ .

Recordemos que una relación de *equivalencia*,  $R$ , en un conjunto  $X$  es una relación entre los elementos de  $X$  que satisface las siguientes propiedades:

(i) Reflexividad.  $xRx$ .

(ii) Simetría. Si  $xRy$ , entonces  $yRx$ .

(iii) Transitividad. Si  $xRy$  y  $yRz$ , entonces  $xRz$ , para todo elemento  $x, y, z \in X$ .

Si  $x \in X$ , a todos los elementos de  $X$  relacionados con  $x$ , denotado por  $[x]$ , y se le llama la clase de equivalencia de  $x$ :

$$[x] = \{u \in X : uRx\}.$$

Cada elemento de este subconjunto es un representante de la clase de equivalencia. Es fácil ver que las clases de equivalencia de  $X$  forman una partición, es decir:

- Son ajenas dos a dos:  $[x] \cap [y] = \emptyset$ , si  $x$  no esta relacionado con  $y$ .
- $X = \bigcup_{x \in X} [x]$ .

A continuación se dará un ejemplo de relación de equivalencia en un grupo que nos será de mucha utilidad.

Sean  $(G, *)$  un grupo abeliano y  $H$  un subgrupo de  $G$ .

Decimos que  $a, b \in G$  están relacionados  $aR_H b$ , si  $a * b^{-1} \in H$ .

Es fácil ver que esta es una relación de equivalencia. A la clase de equivalencia  $[a]$  se acostumbra denotarla también por  $a * H$ , y se le llama la clase lateral derecha de  $a$ . De manera similar se define la clase lateral izquierda,  $H * a$ , y como el grupo  $G$  es abeliano,  $a * H = H * a$ . Al conjunto formado por las clases de equivalencia se denota por:

$$G/H = \{g * H : g \in G\}.$$

Veamos un ejemplo que ilustre lo anterior. Sea  $(\mathbb{Z}, +)$  el grupo aditivo de los enteros. Para  $n \in \mathbb{Z}$ , definimos:

$$n\mathbb{Z} = \{x = nz, z \in \mathbb{Z}\}$$

Es fácil ver que  $(n\mathbb{Z}, +)$ , los múltiplos de  $n$ , es un subgrupo de  $(\mathbb{Z}, +)$ . Entonces,  $x, y \in \mathbb{Z}$  están relacionados si  $x \in (y + n\mathbb{Z})$ , es decir,  $x$  está relacionado con  $y$  siempre y cuando  $x - y$  sea un múltiplo de  $n$ . Si  $m \in \mathbb{Z}$  su clase de equivalencia es:

$$[m] = \{k \in \mathbb{Z} : k - m \in (n + \mathbb{Z})\}$$

y se tiene que

$$\mathbb{Z}/n\mathbb{Z} = \{[m]; m \in \mathbb{Z}\}.$$

Dado un entero  $m$  es costumbre denotar por  $\bar{m}$  a la clase de equivalencia  $[m]$ .

Sea  $(G, *)$  un grupo. Se dice que un subgrupo  $H$  de  $G$  es *normal* en  $G$  si

$$g * H = H * g$$

Se puede ver que esta condición es equivalente a:

$$g^{-1} * H * g = H,$$

para toda  $g \in G$ .

En el caso en que el grupo  $G$  sea conmutativo, esta condición se cumple trivialmente y por lo tanto todo subgrupo es normal.

Dado que a partir de un grupo abeliano  $(G, *)$  y un subgrupo  $H$  de este, se obtiene el conjunto de clases de equivalencia (laterales), es natural preguntarse si este conjunto de clases de equivalencia  $G/H$  es también un grupo. La respuesta es afirmativa y la operación entre los elementos de  $G/H$  está definida de la siguiente manera:

$$(gH) \cdot (kH) = (g * k)H$$

es decir, el producto de dos clases de equivalencia  $gH$  y  $kH$  es otra clase de equivalencia la cual es la clase del elemento  $g * k$ , es decir,  $(g * k)H$ .

Es fácil ver que esta operación está bien definida, ya que no depende del representante de la clase, y la pareja  $(G/H, \cdot)$  es un grupo (conmutativo si  $G$  lo es), llamado el *grupo cociente* de  $G$  módulo  $H$ .

El neutro de este grupo es la clase del neutro  $e$  de  $G$ :  $eH$ . El inverso de  $gH$  es  $g^{-1}H$ .

Para ilustrar el concepto de grupo cociente veamos un ejemplo. Sea  $(\mathbb{Z}, +)$ , el grupo (aditivo) de los enteros y  $H = n\mathbb{Z}$ . El grupo cociente es:

$$\mathbb{Z}/n\mathbb{Z} = \{[m] = \{m + (n\mathbb{Z})\} : m \in \mathbb{Z}\}$$

donde la operación está dada por:

$$[m] + [k] = \{m + (n\mathbb{Z})\} + \{k + (n\mathbb{Z})\} = [m + k] = \{(m + k) + (n\mathbb{Z})\}.$$

Este grupo cociente es el bien conocido grupo (aditivo) de *enteros modulares*, módulo  $n$ .

Si  $n$  y  $m$  son dos enteros (se puede suponer que  $n \leq m$ ), por el algoritmo de la división se tiene que

$$m = nq + r, ; q \in \mathbb{Z}, 0 \leq r < n - 1$$

siendo  $q$  el cociente y  $r$  el residuo.

Por lo tanto,  $m - r = nq$ , es decir,  $m$  y  $r$  son equivalentes módulo  $n$  y por consiguiente  $[m] = [r]$ , es decir, son el mismo elemento en los enteros modulares, y tanto  $m$  como  $r$  son representantes de esa misma clase de

equivalencia. Así, hay tantas clases de equivalencia como residuos, los cuales dado el módulo  $n$ , los residuos son:  $r = 0, 1, \dots, n - 1$ . Por consiguiente se tiene que

$$\mathbb{Z}/n\mathbb{Z} = \{[0], [1], [2], \dots, [n - 1]\} = \{\bar{0}, \bar{1}, \dots, \overline{n - 1}\}.$$

o, haciendo abuso de la notación:

$$\mathbb{Z}/n\mathbb{Z} = \{0, 1, \dots, n - 1\}.$$

Obsérvese que  $\mathbb{Z}/2\mathbb{Z} = \{\bar{0}, \bar{1}\}$  el cual se puede identificar con los números binarios  $\mathbb{F}_2 = \{0, 1\}$  introducidos anteriormente.

Otro ejemplo de anillo cociente que es de mucha utilidad es el de los polinomios en una variable con coeficientes en los números reales  $\mathbb{R}$ , los complejos  $\mathbb{C}$ , los binarios  $\mathbb{F}_2$  o en general sobre un campo, módulo un polinomio fijo, el cual se describirá a continuación.

Sea  $K$  alguno de los campos mencionados anteriormente y

$$K[x] = \{f(x) = a_0 + a_1x + \dots + a_nx^n \mid a_i \in K, n \in \mathbb{N}\}$$

el conjunto de polinomios en la variable  $x$  con coeficientes en  $K$ , al número natural  $n$  se le llama grado del polinomio. Anteriormente se vio que este conjunto con la operación suma usual de polinomios en un grupo conmutativo (aditivo),  $(K[x], +)$ . Sea  $f(x) \in K[x]$  un polinomio fijo de grado  $n$ . Es fácil ver que

$$f(x)K[x] = \langle f(x) \rangle = \{f(x)g(x) : g(x) \in K[x]\}$$

es un subgrupo de  $(K[x], +)$ .

De manera similar al caso de los enteros  $\mathbb{Z}$  se puede definir una relación de equivalencia usando este subgrupo:

$a(x), b(x) \in K[x]$  están relacionados si y sólo si  $a(x) - b(x) \in f(x)K[x]$ , es decir, si  $a(x) - b(x)$  es un múltiplo de  $f(x)$ .

Es fácil ver que esta es una relación de equivalencia. La clase de equivalencia de  $a(x) \in K[x]$  módulo  $f(x)$ ,  $[a(x)]$ , es el subconjunto de  $K[x]$  de los elementos que son equivalentes a  $a(x)$ , es decir, aquellos  $b(x) \in K[x]$  tales que  $a(x) - b(x)$  es un múltiplo de  $f(x)$ , en otros términos,

$$[a(x)] = \{a(x) + \langle f(x) \rangle\} = \{b(x) \in K[x] : a(x) - b(x) = q(x)f(x), q(x) \in K[x]\}.$$

A la clase de equivalencia  $[a(x)]$  se acostumbra denotarla por  $\overline{a(x)}$ . El conjunto de clases de equivalencia módulo  $f(x)$ , se denota por:

$$K[x]/f(x)K[x] = K[x]/\langle f(x) \rangle = \{\overline{a(x)} : a(x) \in K[x]\}$$

Al igual que en el caso de los enteros  $\mathbb{Z}$ , a este conjunto se le puede dar estructura de grupo (aditivo) definiendo la operación de la siguiente manera: si  $\overline{a(x)}, \overline{b(x)} \in K[x]/\langle f(x) \rangle$ ,

$$\overline{a(x)} + \overline{b(x)} = \overline{a(x) + b(x)}.$$

En este grupo, al igual que en los enteros, también se tiene el algoritmo de la división: si  $f(x), g(x) \in K[x]$ , existen polinomios  $q(x), r(x) \in K[x]$  tales que:

$$g(x) = q(x)f(x) + r(x), \text{ con } 0 \leq \text{grad}(r(x)) < \text{grad}(f(x)).$$

Al polinomio  $q(x)$  se le llama cociente y a  $r(x)$  el residuo.

Como consecuencia de esto se tiene que para saber la clase residual (de equivalencia) de un polinomio  $a(x)$  módulo un polinomio  $f(x)$  basta dividir a  $a(x)$  por  $f(x)$  y ver cual es su residuo, ya que:

$$a(x) - r(x) = q(x)f(x),$$

es decir,  $a(x)$  y  $r(x)$  son equivalentes módulo  $f(x)$  y por consiguiente:  $\overline{a(x)} = \overline{r(x)}$ .

En otras palabras, se toma cualquier representante de la clase  $\overline{a(x)}$  y cualquier representante de la clase  $\overline{b(x)}$ , los cuales son elementos de  $K[x]$ , se suman como elementos de  $K[x]$  y se toma el residuo que se obtiene de dividir esta suma por  $f(x)$ : si  $a(x)$  y  $b(x)$  son los representantes de las respectivas clases,

$$a(x) + b(x) = q(x)f(x) + r(x),$$

y  $\overline{a(x) + b(x)} = \overline{r(x)}$ .

Por consiguiente si se toma a  $f(x) \in K[x]$  de grado  $n$ , los elementos de  $K[x]/f(x)K[x]$ , es decir, las clases residuales (de equivalencia) módulo  $f(x)$ , y dado que el residuo que se obtiene de dividir a un polinomio  $g(x)$  por  $f(x)$  tiene grado a lo mas  $n-1$ , y abusando del lenguaje se puede considerar que

$$K[x]/f(x)K[x] = \{a_0 + a_1x + \cdots + a_{n-1}x^{n-1}, a_i \in K\},$$

recordando que cada uno de estos elementos representa una clase de equivalencia, módulo  $f(x)$ .

**Observación.** Tanto en los números enteros como en el caso de los polinomios se ha usado el hecho que ambos son grupos aditivos, pero también poseen la operación producto, aunque el lector la conoce, no se ha introducido formalmente. Esto se hará en la siguiente subsección donde se recordará el concepto de *anillo*, algunas propiedades y ejemplos.

## 1.2. Anillos

En esta subsección se recordarán conceptos básicos de anillos, sobre todo conmutativos, se darán algunas de sus propiedades y se darán ejemplos que serán útiles a lo largo del trabajo.

La terna  $(A, +, *)$  donde  $A$  es un conjunto no vacío y “+”, “\*”, dos operaciones en  $A$ , usualmente llamadas suma y producto, es un **anillo**,  $(A, +, \cdot)$  si satisface las siguientes propiedades:

- (i)  $(A, +)$  es un grupo (aditivo) conmutativo.
- (ii) El producto es asociativo:  $a * (b * c) = (a * b) * c$ , para todo  $a, b, c \in A$ .
- (iii) El producto es distributivo respecto a la suma:  $(a + b) * c = a * c + b * c$  y  $a * (b + c) = a * b + a * c$ , para todo  $a, b, c \in A$ .

Si el producto es conmutativo, el anillo es *conmutativo*. Si existe un elemento  $e \in A$  tal que  $e * a = a * e$ , para todo  $a \in A$ , se dirá que  $A$  es un anillo con identidad.

Algunos ejemplos de anillos conmutativos son:

- 1)  $(\mathbb{Z}, +, \cdot)$ : el anillo de los números enteros con la suma y multiplicación usual.
- 2)  $(K, +, \cdot)$ : donde  $K$  puede ser los números reales  $\mathbb{R}$ , complejos  $\mathbb{C}$ , racionales  $\mathbb{Q}$  o bien los números binarios  $\mathbb{F}_2$ .



3)  $(K[x], +, \cdot)$ : el anillo de polinomios en una indeterminada con coeficientes en  $K$ , donde la operación “ $\cdot$ ” es el producto usual de polinomios. El caso  $K = \mathbb{F}_2$  será de mucha utilidad en este trabajo.

4)  $(\{a_0 + a_1x + a_2x^2 + \dots : a_i \in \mathbb{F}_2\}, +, \cdot)$  El anillo de series de potencias en una indeterminada  $x$  con coeficientes en  $\mathbb{F}_2$ , es decir, es el conjunto de series de potencias con coeficientes binarios, donde la suma y multiplicación de los coeficientes se hacen módulo 2.

Acordaremos que, a menos que se diga lo contrario, todos los anillos que se usarán en este escrito son conmutativos. Un ejemplo de anillo no conmutativo que se usará más adelante son las matrices con entradas en  $\mathbb{R}, \mathbb{Q}, \mathbb{Z}, \mathbb{F}_2$  y  $\mathbb{F}_2[x]$ , con las operaciones usuales de matrices. En particular pueden ser matrices  $(a_{ij}) \in \text{Mat}_{b \times c}(\mathbb{F}_2[x])$ , tal que  $b \leq c$ , ya que con la operación suma es conmutativo, pero no con el producto, por lo cual  $(\text{Mat}_{b \times c}(\mathbb{F}_2[x]), +, \cdot)$ , es un anillo con identidad pero no conmutativo.

Un subconjunto  $B$  de un anillo  $A$  es un **subanillo** si para todo  $a, b \in B$  se tiene que  $a - b \in B$  y  $a \cdot b \in B$ .

Un **ideal**  $I$  de un anillo  $A$  es un subconjunto  $I \subset A$  que satisface:

(i)  $(I, +)$  : es un subgrupo aditivo.

(ii) Para todo  $r \in I$  y todo  $a \in A$ ,  $ra \in I$ .

Sea  $I$  un ideal del anillo  $A$ . Si existe  $s \in I$  tal que  $\{as : a \in A\} = As = I = \langle s \rangle$ , diremos que  $I$  es un ideal principal generado por  $s$ . Algunos ejemplos son los siguientes:

1) Sean  $A = \mathbb{Z}$  e  $I = n\mathbb{Z}$ , entonces  $I$  es un ideal principal de  $A$ .

2) Un anillo  $A$  con identidad siempre contiene por lo menos dos ideales:  $\{0\}$  que consta únicamente del elemento 0 y  $A = eA$ .

3)  $I = \{2n : n \in \mathbb{Z}\}$  es un ideal de  $\mathbb{Z}$ .

4) En  $\mathbb{Z}$ ,  $\langle 2, 5 \rangle = \langle 1 \rangle = \mathbb{Z}$ , ya que  $3 \cdot 2 + (-1) \cdot 5 = 1$ .

Un **Dominio entero** es un anillo  $A$  con identidad  $1_A \neq 0$ , tal que si  $a, b \in A$  y  $a \cdot b = 0$ , entonces  $a = 0$  o  $b = 0$ . En un dominio entero  $A$ , un elemento irreducible o primo es un elemento  $p \in A$ ,  $p \neq 0, 1_A$ , tal que siempre que  $p = a \cdot b$  con  $a, b \in A$ , se tiene que  $a$  o  $b$  es la identidad.

Algunos ejemplos de dominio entero son:

1)  $(\mathbb{Z}, +, \cdot)$

2)  $(\mathbb{R}, +, \cdot)$

3)  $(\mathbb{Q}, +, \cdot)$

4)  $(\mathbb{C}, +, \cdot)$

5)  $(\mathbb{F}_2[x], +, \cdot)$

6)  $\{a + b\sqrt{2} : a, b \in \mathbb{Z}\}$ .

7)  $\mathbb{Z}_6$ , las clases de restos módulo 6 es un anillo conmutativo con unidad, pero no un dominio entero porque  $2 \cdot 3 = 0$ .

Un tipo de anillos de particular importancia para nosotros son **los anillos euclidianos**. Un anillo euclidiano  $\mathbb{E}$  es un anillo para el cual existe una función  $v$ :

$$v : \mathbb{E} - \{0\} \longrightarrow \mathbb{N} \cup \{0\},$$

llamada *grado o norma*, que satisface:

(i) Para todo  $a, b \in \mathbb{E} - \{0\}$ ,  $v(a) \leq v(a \cdot b)$ .

(ii) Dados  $a \in \mathbb{E}$  y  $b \in \mathbb{E} - \{0\}$ , existen  $q, r \in \mathbb{E}$ , tal que  $a = bq + r$ , con  $0 \leq v(r) < v(q)$ .

Los anillos euclidianos son aquellos a los que se les puede aplicar el algoritmo de la división o algoritmo de Euclides.

Un ejemplo de anillo euclidiano es el de los números enteros  $(\mathbb{Z}, +, \cdot)$ . En este caso la norma  $v$  es el valor absoluto y se tiene que para todo  $a, b \in \mathbb{Z}$  existen  $q, r \in \mathbb{Z}$  tales que:

$$a = bq + r, \quad 0 \leq r < b.$$

Al elemento  $q$  se le llama cociente,  $r$  es el residuo y la pareja  $(q, r)$  es única.

Otros ejemplos de anillos euclidianos son:

1)  $(\mathbb{F}_2[x], +, \cdot)$ : anillo de polinomios con coeficientes en  $\mathbb{F}_2$ , donde la función  $v$  es el grado de un polinomio. Si  $a(x), b(x) \in \mathbb{F}_2[x]$  y  $n = \text{grad}(b(x)) \leq \text{grad}(a(x)) = m$ ,  $b(x) \neq 0$ , existen  $r(x)$  y  $q(x)$  en  $\mathbb{F}_2[x]$ , tales que:

$$a(x) = b(x)q(x) + r(x),$$

donde  $0 \leq \text{grad}(r(x)) < \text{grad}(b(x))$ ,  $r(x)$  es llamado el residuo y  $q(x)$  el cociente.

3)  $(\mathbb{Z}[i], +, \cdot)$ : los enteros gaussianos son los números formados por el conjunto  $\{a + bi : a, b \in \mathbb{Z}\}$ , tal que  $i^2 = -1$ , cuyas operaciones son las de los números complejos  $\mathbb{C}$  y  $v(a + bi) = a^2 + b^2$ .

4) Los enteros de Eisenstein,  $\mathbb{Z}[w] = \{a + bw : a, b \in \mathbb{Z}\}$ , donde  $w$  es una raíz cúbica primitiva de 1, [10].

Un resultado importante que será de gran utilidad en este trabajo es el siguiente:

**Teorema** ([16], [10]). Todo anillo euclidiano es de ideales principales.

Es decir, en todo anillo euclidiano  $A$ , cualquier ideal  $I$  esta generado por un elemento, es decir, es de la forma  $I = xA = \langle x \rangle = \{xa; a \in A\}$ .

**Observación.** El anillo de polinomios es uno de los que serán de mucha utilidad en este trabajo. El hecho que es un anillo euclideano, y por consecuencia de ideales principales, es de vital importancia en los resultados que aquí se presenten, y en general, es de gran utilidad en otras aplicaciones.

Como en el caso de grupos, los subgrupos normales ayudan a construir el grupo cociente. En el caso de anillos se usan los ideales para construir el **anillo cociente** de la siguiente manera:

$$A/I = \{r + I : r \in A\}.$$

$I$  es un subgrupo de  $A$  y como se vio en la subsección anterior, este conjunto tiene la estructura de grupo (abeliano), con la suma definida como:  $(r + I) + (s + I) = (r + s) + I$ . El elemento neutro respecto a esta operación es  $0 + I = I$  y el inverso de  $r + I$  es  $-r + I$ . La multiplicación está definida de la siguiente forma:

$$(r + I) \cdot (s + I) = (rs) + I; r + I, s + I \in R/I.$$

Es fácil ver que esta operación no depende de la elección de los representantes  $r, s$ , es decir, si  $r + I = r' + I$  y  $s + I = s' + I$ , entonces  $(rs) + I = (r's') + I$ . Con estas operaciones  $(R/I, +, \cdot)$  un anillo (conmutativo).

A continuación se mencionan algunos ejemplos de anillo cociente, el definido sobre los enteros (rationales) y sobre el anillo de polinomios es de particular importancia en este trabajo.

Ejemplo. Sea  $n \in \mathbb{Z}$  positivo e  $I = \langle n \rangle$  el ideal generado por  $n$ . Como se mencionó anteriormente,  $\mathbb{Z}/I = \bar{0} = 0 + I, \bar{1} = 1 + I, \dots, \overline{n-1} = (n-1) + I$  es un grupo aditivo. Si  $\bar{a}$  y  $\bar{b}$  son dos elementos de este grupo,

$$\overline{ab} = \bar{a}\bar{b},$$

es decir, se toma un representante de la clase  $\bar{a}$ , digamos  $a$ , y uno de la clase  $\bar{b}$ , digamos  $b$ , los cuales son elementos de  $\mathbb{Z}$ , se multiplican como tales y se reducen módulo  $n$ , por ejemplo viendo cual es su residuo al dividirlo por el módulo  $n$ .

Como caso concreto sea  $I = 6\mathbb{Z}$ , entonces  $R/I = \mathbb{Z}/6\mathbb{Z} = \{\bar{0} = 6\mathbb{Z}, \bar{1} = 1 + 6\mathbb{Z}, \bar{2} = 2 + 6\mathbb{Z}, \bar{3} = 3 + 6\mathbb{Z}, \bar{4} = 4 + 6\mathbb{Z}, \bar{5} = 5 + 6\mathbb{Z}\}$ . Por ejemplo para obtener el producto  $\bar{4} \cdot \bar{5}$ , se multiplica  $4 \cdot 5 = 20$  y se reduce módulo 6,  $20 = (3)(6) + 2$ , resultando  $\bar{2}$ .

Un **campo**  $\mathbb{F}$ , es un anillo conmutativo  $(\mathbb{F}, +, \cdot)$ , tal que  $(\mathbb{F}, +)$  y  $(\mathbb{F}^*, +, \cdot)$ ,  $\mathbb{F}^* = \mathbb{F} - \{0\}$ , son grupos abelianos y  $(\mathbb{F}, +, \cdot)$  es un dominio entero. En este

caso, todos los elementos distintos de cero tienen inverso bajo el producto. Algunos ejemplos de campos representativos son:

- 1)  $(\mathbb{R}, +, \cdot)$ : los números reales.
- 2)  $(\mathbb{Q}, +, \cdot)$ : los números racionales.
- 3)  $(\mathbb{C}, +, \cdot)$ : los números complejos.

Los siguientes resultados son interesantes y su demostración no es difícil y el lector interesado puede consultar, por ejemplo [6], [10].

**Teorema.** a) El anillo  $\mathbb{Z}/\langle n \rangle$  es un campo con  $n$  elementos si y sólo si  $n$  es un número primo.

b) Si  $K$  es un campo, el anillo  $K[x]/\langle f(x) \rangle$  es un campo si y sólo si  $f(x)$  es irreducible.

Recordemos que un polinomio  $a(x) \in K$  es **irreducible** si para cualesquiera dos polinomios  $b(x), c(x)$  de grado mayor que 1,

$$a(x) \neq b(x)c(x).$$

Algunos ejemplos de polinomios irreducibles son:

1)  $(x^2 - 2) \in \mathbb{Z}[x]$

2)  $(x^2 + x + 1) \in \mathbb{F}_2[x]$

3)  $(x^2 + 1) \in \mathbb{R}[x]$

4)  $(x^2 - \frac{4}{9}) \in \mathbb{Q}[x]$

5) El campo de las funciones racionales,  $(\mathbb{F}_2(x), +, \cdot)$ : sea  $\mathbb{F}_2$  el campo de los números binarios,  $\mathbb{F}_2[x]$  el anillo de polinomios en una indeterminada con coeficientes binario, y sea  $\mathbb{F}_2(x) = \{ \frac{f(x)}{g(x)} = h(x) : g(x) \neq 0, f(x), g(x) \in \mathbb{F}_2[x] \}$ . Donde la suma se define como:

$$\frac{f(x)}{g(x)} + \frac{f'(x)}{g'(x)} = \frac{f(x)g'(x) + f'(x)g(x)}{g(x)g'(x)},$$

y el producto como:

$$\frac{f(x)}{g(x)} \cdot \frac{f'(x)}{g'(x)} = \frac{f(x)f'(x)}{g(x)g'(x)}.$$

El inverso de  $\frac{f(x)}{g(x)} \neq 0$  es  $\frac{g(x)}{f(x)}$ . Es fácil ver que con con estas operaciones  $\mathbb{F}_2(x)$  es un campo y se le llama el campo de funciones racionales sobre  $\mathbb{F}_2$ .

6) Sea  $\mathbb{F}_2((x)) = \{\sum_{i=r}^{\infty} f_i x^i : f_i \in \mathbb{F}_2, r \in \mathbb{Z}\}$ .  $\mathbb{F}_2((x))$  con la operación natural de suma y producto es un anillo conmutativo, y dado que todo elemento no-cero tiene inverso, resulta ser un campo, llamado de las series de Laurent.

En teoría de códigos resultan de especial interés los campos con un número finito de elementos, los cuales se llaman **campos finitos** o campos de Galois (en honor a E. Galois).

**Dominios de factorización única.** Si en un dominio entero  $A$ , todo elemento no nulo y diferente de la unidad se puede escribir en forma única, salvo unidades o el orden de los factores, como producto de irreducibles, se dice que  $A$  es un dominio de factorización única. En el anillo de enteros  $\mathbb{Z}$ , todo entero no cero ni unidad se puede factorizar, en forma única, como producto de potencias de números primos (Teorema fundamental de la aritmética), [16]. Un ejemplo de dominio de factorización única es  $K[x]$  con  $K$  campo.

### 1.3. Campos finitos

Como se mencionó anteriormente, los campos finitos son de vital importancia en el estudio de códigos. A continuación se dará una construcción de un campo finito y se verán algunas de sus propiedades.

Sea  $p$  un número primo y

$$\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{\overline{0}, \overline{1}, \dots, \overline{p-1}\},$$

el campo finito (modular) con  $p$  elementos.

Sea  $f(x) \in \mathbb{F}_p[x]$  irreducible de grado  $n$  y

$$\mathbb{F}_p[x]/\langle f(x) \rangle = \{a(x) + \langle f(x) \rangle\}, \quad (1.1)$$

donde  $\langle f(x) \rangle$  es el ideal generado por el polinomio  $f(x)$ , el anillo cociente.

Los elementos del anillo  $\mathbb{F}_p[x]/\langle f(x) \rangle$  son el conjunto que resulta de aplicar el algoritmo de la división a cualquier polinomio  $p(x) \in \mathbb{F}_p[x]$ . En general tendremos:

$$p(x) = f(x)q(x) + r(x), \quad (1.2)$$

para algún  $q(x) \in \mathbb{F}_p[x]$  y  $r(x) \in \mathbb{F}_p[x]$ , el grado de  $r(x)$  es

$$0 \leq \text{grad}(r(x)) < \text{grad}(f(x)) = n.$$

Despejando de (1.2)

$$p(x) - r(x) = f(x)q(x)$$

que es equivalente a la operación módulo, denotada como:

$$p(x) \equiv r(x) \pmod{f(x)}.$$

Los residuos serán los representantes de las clases de equivalencia de polinomios, y las clases son equivalentes si al realizar el algoritmo de la división generan el mismo residuo. Estos residuos son precisamente todos los polinomios con coeficientes en  $\mathbb{F}_p$  con grado  $\leq n - 1$ , es decir,

$$\mathbb{F}_p[x]/\langle f(x) \rangle = \{a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_{n-1}x^{n-1} : a_i \in \mathbb{F}_p\},$$

que implica que el anillo  $\mathbb{F}_p[x]/\langle f(x) \rangle$  tiene  $p^n$  elementos.

Para que ese anillo cociente sea un campo basta con que  $f(x)$  sea un polinomio irreducible, ya que si  $f(x)$  es irreducible, entonces todo polinomio  $g(x) \in \mathbb{F}_p[x]$ , a menos que sea múltiplo de éste, es primo relativo con  $f(x)$ , es decir,

$$(f(x), g(x)) = 1,$$

y mediante el algoritmo extendido de Euclides es posible encontrar una combinación lineal, tal que

$$1 = f(x)k(x) + g(x)l(x)$$

para  $k(x)$  y  $l(x)$  en  $\mathbb{F}_p[x]$ ; y como estamos trabajando módulo  $f(x)$ , entonces

$$\bar{1} = \overline{f(x)k(x) + g(x)l(x)} = \overline{f(x)} \cdot \overline{k(x)} + \overline{g(x)} \cdot \overline{l(x)}$$

que implica que:

$$\bar{1} = \overline{g(x)} \cdot \overline{l(x)} \quad (1.3)$$

ya que  $\overline{f(x)} \cdot \overline{k(x)}$  es la clase del cero, es decir,  $\overline{f(x)} = \bar{0}$ , o bien,

$$f(x) \cdot k(x) \equiv 0 \pmod{f(x)}.$$

La ecuación (1.3) indica que para  $p(x) \neq 0$  en  $\mathbb{F}_p[x]$  existe algún  $l(x) \in \mathbb{F}_p[x]$  tal que  $l(x)$  es inverso de  $p(x)$ , es decir, todo elemento en  $\mathbb{F}_p[x]$  distinto de cero tiene inverso, por lo cual el anillo cociente es un campo,

$$\bar{1} = \overline{g(x)} \cdot [\overline{g(x)}]^{-1}$$

donde  $\overline{l(x)} = [\overline{g(x)}]^{-1}$ .

A este campo se le denota por  $\mathbb{F}_{p^n}$  y tiene cardinalidad  $q = p^n$ .

A continuación se dará la construcción explícita de algunos campos finitos.

Dos propiedades importantes de los campos finitos son:

1. Módulo isomorfismo el campo con  $q = p^n$  elementos es único.
2. Si  $\mathbb{F}_q$  es un campo finito, su grupo multiplicativo  $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$  es un grupo cíclico de orden  $q - 1$ . A un generador de este grupo se le llama *elemento primitivo* del campo finito.

### Ejemplos.

1. Campo de 9 elementos. En este caso  $q = 9 = 3^2$ , por lo tanto  $p = 3$  y hay que tomar un polinomio irreducible  $f(x) \in \mathbb{F}_3[x]$  de grado 2. Por ejemplo, un polinomio irreducible de grado 2 sobre  $\mathbb{F}_3$  (para generar este campo) es  $f(x) = x^2 + 2x + 2$ . El lector puede verificarlo.
2. Campo con 8 elementos. Aquí  $p = 2$  y hay que tomar un polinomio irreducible de grado 3 sobre los números binarios. Sea  $p(x) = x^3 + x^2 + 1 \in \mathbb{F}_2$ . Es fácil ver que este polinomio es irreducible

$$\mathbb{F}_2[x]/\langle x^3 + x^2 + 1 \rangle = \mathbb{F}_{2^3}$$



$$= \mathbb{F}_8 = \{\overline{0}, \overline{1}, \overline{x}, \overline{x^2}, \overline{x^2 + 1}, \overline{x^2 + x + 1}, \overline{x + 1}, \overline{x^2 + x}\}.$$

Si  $\alpha = \overline{x}$ ,

$$a_0 + a_1\alpha + a_2\alpha^2 = a_0 + a_1\overline{x} + a_2\overline{x^2} = 0, \quad a_i \in \mathbb{F}_2,$$

y se tiene que,

$$\mathbb{F}_8 = \{0, 1, \alpha, \alpha^2, \alpha^2 + 1, \alpha^2 + \alpha + 1, \alpha + 1, \alpha^2 + \alpha\}.$$

Para generar  $\mathbb{F}_8$  usamos el polinomio  $f(x) = x^3 + x^2 + 1$ , que es irreducible en  $\mathbb{F}_2[x]$  y de grado 3, pero ¿qué pasa si usamos un  $g(x)$  irreducible, tal que  $g(x) \in \mathbb{F}_2[x]$ ,  $\text{grad}(g(x)) = 3$  y  $g(x) \neq f(x)$ ?

Tomemos  $g(x) = x^3 + x + 1$  que es un polinomio irreducible en  $\mathbb{F}_2[x]$  y hagamos lo mismo que hicimos para  $\mathbb{F}_2[x]/\langle f(x) \rangle$ .

Para  $\mathbb{F}_2[x]/\langle x^3 + x + 1 \rangle$ ,

$$\overline{x^3 + x + 1} = \overline{0}$$

es decir, los múltiplos de  $x^3 + x + 1$  están en la clase del 0.

Para este campo, cada vez que aparezca el término  $\overline{x^3}$  lo sustituiremos por  $\overline{x + 1}$ , ya que si  $\overline{x^3} + \overline{x} + 1 = 0$ , entonces

$$\overline{x^3} = -\overline{x} - 1 = \overline{x} + 1.$$

Con el ideal generado por ese polinomio tenemos:

$$\mathbb{F}_2[x]/\langle x^3 + x + 1 \rangle = \{\overline{0}, \overline{1}, \overline{x}, \overline{x^2}, \overline{x + 1}, \overline{x^2 + x}, \overline{x^2 + x + 1}, \overline{x^2 + 1}\} := \mathbb{F}'_8,$$

$\mathbb{F}'_8$  se generará de la misma manera que  $\mathbb{F}_8$  y los elementos de  $\mathbb{F}_8$  son los mismos que los de  $\mathbb{F}'_8$ , lo que los hace diferentes es su aritmética.

La aritmética de estos campos está dada por el polinomio que se elige. Para  $\mathbb{F}'_8$  cada vez que nos encontremos con  $x^3$  sustituiremos por  $x + 1$  y en  $\mathbb{F}_8$  cada vez que aparezca  $x^3$  sustituiremos por  $x^2 + 1$ . Las tablas de suma y multiplicación correspondientes son las siguientes:

Para  $\mathbb{F}_8$

.	0	1	$\alpha$	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$
0	0	0	0	0	0	0	0	0
1	0	1	$\alpha$	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$
$\alpha$	0	$\alpha$	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$	1
$\alpha^2$	0	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$	1	$\alpha$
$\alpha^2+1$	0	$\alpha^2+1$	$\alpha^2+\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$	1	$\alpha$	$\alpha^2$
$\alpha^2+\alpha+1$	0	$\alpha^2+\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$	1	$\alpha$	$\alpha^2$	$\alpha^2+1$
$\alpha+1$	0	$\alpha+1$	$\alpha^2+\alpha$	1	$\alpha$	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha+1$
$\alpha^2+\alpha$	0	$\alpha^2+\alpha$	1	$\alpha$	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha+1$	$\alpha+1$

+	0	1	$\alpha$	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$
0	0	1	$\alpha$	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$
1	1	0	$\alpha+1$	$\alpha^2+1$	$\alpha^2$	$\alpha^2+\alpha$	$\alpha$	$\alpha^2+\alpha+1$
$\alpha$	$\alpha$	$\alpha+1$	0	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+1$	1	$\alpha^2$
$\alpha^2$	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha$	0	1	$\alpha+1$	$\alpha^2+\alpha+1$	$\alpha$
$\alpha^2+1$	$\alpha^2+1$	$\alpha^2$	$\alpha^2+\alpha+1$	1	0	$\alpha$	$\alpha^2$	$\alpha+1$
$\alpha^2+\alpha+1$	$\alpha^2+\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+1$	$\alpha+1$	$\alpha$	0	$\alpha^2$	1
$\alpha+1$	$\alpha+1$	$\alpha$	1	$\alpha^2+\alpha+1$	$\alpha^2+\alpha$	$\alpha^2$	0	$\alpha^2+1$
$\alpha^2+\alpha$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2$	$\alpha$	$\alpha+1$	1	$\alpha^2+1$	0

y para  $\mathbb{F}'_8$

.	0	1	$\alpha$	$\alpha^2$	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+1$
0	0	0	0	0	0	0	0	0
1	0	1	$\alpha$	$\alpha^2$	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+1$
$\alpha$	0	$\alpha$	$\alpha^2$	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+1$	1
$\alpha^2$	0	$\alpha^2$	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+1$	1	$\alpha$
$\alpha+1$	0	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+1$	1	$\alpha$	$\alpha^2$
$\alpha^2+\alpha$	0	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+1$	1	$\alpha$	$\alpha^2$	$\alpha+1$
$\alpha+\alpha+1$	0	$\alpha^2+\alpha+1$	$\alpha^2+1$	1	$\alpha$	$\alpha^2$	$\alpha+1$	$\alpha^2+\alpha$
$\alpha^2+1$	0	$\alpha^2+1$	1	$\alpha$	$\alpha^2$	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$

+	0	1	$\alpha$	$\alpha^2$	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+1$
0	0	1	$\alpha$	$\alpha^2$	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+1$
1	1	0	$\alpha+1$	$\alpha^2+1$	$\alpha$	$\alpha^2+\alpha+1$	$\alpha^2+\alpha$	$\alpha^2$
$\alpha$	$\alpha$	$\alpha+1$	0	$\alpha^2+\alpha$	1	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha+1$
$\alpha^2$	$\alpha^2$	$\alpha^2+1$	$\alpha^2+\alpha$	0	$\alpha^2+\alpha+1$	$\alpha$	$\alpha+1$	1
$\alpha+1$	$\alpha+1$	$\alpha$	1	$\alpha^2+\alpha+1$	0	$\alpha^2+1$	$\alpha^2+1$	$\alpha^2+\alpha$
$\alpha^2+\alpha$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	$\alpha^2$	$\alpha$	$\alpha^2+1$	0	1	$\alpha+1$
$\alpha^2+\alpha+1$	$\alpha^2+\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+1$	$\alpha+1$	$\alpha^2$	1	0	$\alpha$
$\alpha^2+1$	$\alpha^2+1$	$\alpha^2$	$\alpha^2+\alpha+1$	1	$\alpha^2+\alpha$	$\alpha+1$	$\alpha$	0

Se puede ver que  $\mathbb{F}'_8 = \mathbb{F}'_8 - \{0\}$  y  $\mathbb{F}_8 = \mathbb{F}_8 - \{0\}$  son grupos cíclicos y un isomorfismo  $\varphi : \mathbb{F}'_8 \rightarrow \mathbb{F}_8$ , está determinado por:

$$\begin{aligned}\varphi(0) &= 0 \\ \varphi(1) &= 1 \\ \varphi(\alpha) &= \alpha \\ \varphi(\alpha^2) &= \alpha^2 \\ \varphi(\alpha+1) &= \alpha^2+1 \\ \varphi(\alpha^2+\alpha) &= \alpha^2+\alpha+1 \\ \varphi(\alpha^2+\alpha+1) &= \alpha+1 \\ \varphi(\alpha^2+1) &= \alpha^2+\alpha.\end{aligned}$$

### 1.3.1. Espacio vectorial

Sea  $(V, +)$  un grupo abeliano aditivo y  $\mathbb{F}$  un campo. Si el mapeo

$$\begin{aligned}\mathbb{F} \times V &\longrightarrow V \\ (a, v) &\longrightarrow av,\end{aligned}$$

al que llamaremos multiplicación por escalares, es asociativo en la multiplicación y distribuye la suma para valores del campo y del grupo abeliano  $(V, +)$ , entonces diremos que  $V$  es un espacio vectorial sobre el campo  $\mathbb{F}$  o un  $\mathbb{F}$ -espacio y los elementos  $v \in V$  son llamados vectores.

Un subconjunto,  $\{v_1, \dots, v_k\}$  de  $V$ , de vectores diferentes de cero se dice que son linealmente independientes sobre  $\mathbb{F}$ , si no es posible encontrar  $f_1, f_2, \dots, f_k \in \mathbb{F}$ , con al menos de uno de ellos diferente de cero, tales que:

$$\sum_{i=1}^k f_i v_i = 0.$$

Un subconjunto de vectores,  $\{v_1, \dots, v_k\} \subset V$ , se dice que es conjunto de **generadores** de  $V$ , si todo elemento del espacio vectorial se puede escribir como combinación lineal de estos vectores

$$v = \sum_{i=1}^k f_i v_i, f_i \in \mathbb{F}.$$

Si un subconjunto de vectores  $\{v_1, \dots, v_k\}$  del espacio  $V$  es linealmente independiente y generador, diremos que es una **base** de  $V$ . Un espacio vectorial no necesariamente tiene una única base, pero todo espacio vectorial tiene al menos una base. Si una base de un espacio vectorial es de cardinalidad finita, cualquier otra base tiene la misma cardinalidad; similarmente si una base tiene cardinalidad infinita. A la cardinalidad de cualquier base de un espacio vectorial se le llama **dimensión** del espacio vectorial.

#### Ejemplos de espacios vectoriales.

1) El conjunto de los polinomios de grado menor o igual que 2 con coeficientes reales,

$$P_2 := \{ax^2 + bx + c : a, b, c \in \mathbb{R}\}.$$

Este es un  $\mathbb{R}$ -espacio vectorial, pues podemos sumar dos elementos de  $P_2$  y obtenemos otro elemento del mismo conjunto y de igual manera con la multiplicación por un escalar real:

$$(ax^2 + bx + c) + (a'x^2 + b'x + c') = (a + a')x^2 + (b + b')x + (c + c') \in P_2,$$

$$r \in \mathbb{R}, r(ax^2 + bx + c) = (rax^2 + rbx + rc) \in P_2.$$

El cero del espacio vectorial es el polinomio cero. Una base natural es el conjunto  $\{1, x, x^2\}$ , y el espacio tiene dimensión 3.

2)  $\mathbb{R}^n = \mathbb{R} \times \cdots \times \mathbb{R}$ , con  $n \geq 1$ ,  $\mathbb{R}$  el campo de los números reales. A este espacio vectorial se le llama el producto cartesiano sobre  $\mathbb{R}$ . Una base (natural) esta dada por los vectores (canónicos)  $e_i$ ,  $i = 1, 2, \dots, n$ , donde todas las coordenadas de  $e_i$  son iguales a cero excepto en la entrada  $i$  donde la coordenada es igual a 1. Si en lugar de  $\mathbb{R}$  se tiene cualquier campo, el resultado es el mismo.

3)  $(\mathbb{F}_2[x], +, *)$ : el conjunto de polinomios en una indeterminada  $x$  sobre  $\mathbb{F}_2$ , ya que análogamente al ejemplo 1, podemos sumar dos elementos de  $\mathbb{F}_2[x]$  y obtenemos otro elemento del mismo anillo y de igual manera con la multiplicación por un escalar, cabe mencionar que este espacio vectorial no tiene dimensión finita.

La **dimensión** de  $V$  sobre  $\mathbb{F}$ ,  $\dim_{\mathbb{F}}(V) = k$ , es el número de elementos que forman cualquier base de  $V$ , y toda base tiene el mismo número de elementos. Si  $k$  no existe, diremos que  $V$  es de dimensión infinita.

## 1.4. Forma normal de Smith

La forma normal de Smith permite descomponer una matriz, de tamaño  $b \times c$ ,  $b \leq c$ , con entradas en un anillo euclidiano, particularmente con entradas polinomiales, como producto de tres matrices, cuyas características veremos a continuación. Esta descomposición, como se verá mas adelante, es importante para entender con mayor claridad un código de convolución y sus diagramas asociados. Tal descomposición se basa en el siguiente resultado cuyo fundamento algebraico es el algoritmo de Euclides [7], [25]:

**Teorema (Forma normal de Smith).** Sea  $\mathbb{E}$  un anillo euclidiano,  $G$  una matriz no cero  $b \times c$ ,  $b \leq c$  con entradas en  $\mathbb{E}$ . Entonces,

$$G = A\Gamma B,$$

donde  $A$  y  $B$  son una matriz  $b \times b$  y  $c \times c$ , respectivamente con entradas en  $\mathbb{E}$ , y  $\Gamma$  es una matriz diagonal  $b \times c$ , tal que:

$$\Gamma = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_s, 0, \dots, 0).$$

Los elementos  $\gamma_i \in \mathbb{E}$  son diferentes de cero y son tales que  $\gamma_i$  divide a  $\gamma_{i+1}$  para toda  $i$ ,  $1 \leq i \leq s$ . Estos elementos son llamados los factores invariantes de  $G$ .

Este resultado es particularmente útil en el caso del anillo de los enteros y el anillo de polinomios en una indeterminada con coeficientes en un campo, y de particular importancia en este trabajo cuando el campo es finito.

La matriz  $G$  no necesariamente es una matriz cuadrada, esta característica resulta de gran importancia en nuestro caso, debido a que la tasa de un código de convolución tiene asociada un tamaño de matriz y viceversa, por lo cual para toda tasa  $R = b/c$ ,  $b \leq c$ , de un código de convolución podemos usar la forma normal de Smith.

A continuación se indican los pasos que conducen a la descomposición en la forma normal de Smith de una matriz sobre un anillo euclidiano. Para una demostración mas detallada y formal sugerimos al lector consultar por ejemplo [7], [25].

### **Método para llevar una matriz $G$ a la forma normal de Smith**

Primero consideremos las siguientes tipos de operaciones elementales:

**Tipo 1:** intercambio de dos renglones (o columnas).

**Tipo 2:** Multiplicar un renglón (o columna) por un elemento diferente de cero, en el anillo euclidiano y sumar a otro renglón (o columna).

Sea la matriz,

$$G = \begin{pmatrix} g_{11} & g_{12} & g_{13} & \cdots & g_{1c} \\ g_{21} & g_{22} & g_{23} & \cdots & g_{2c} \\ g_{31} & g_{32} & g_{33} & \cdots & g_{3c} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ g_{b1} & g_{b2} & g_{b3} & \cdots & g_{bc} \end{pmatrix} = (g_{ij})$$

de tamaño  $b \times c$ .

Para crear la igualdad entre  $G$  y una descomposición de esta, escribimos la matriz identidad  $I_b$  (matriz identidad de tamaño  $b \times b$ ) a la izquierda de la matriz  $G$  y a la derecha la matriz  $I_c$ , la multiplicación de estas matrices no afecta a  $G$ , ya que son matrices identidad,

$$G = I_c G I_b,$$

es decir,

$$G = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} g_{11} & g_{12} & g_{13} & \cdots & g_{1c} \\ g_{21} & g_{22} & g_{23} & \cdots & g_{2c} \\ g_{31} & g_{32} & g_{33} & \cdots & g_{3c} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ g_{b1} & g_{b2} & g_{b3} & \cdots & g_{bc} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Con estas tres matrices, procedemos a realizar los siguientes pasos:

1. Sin pérdida de generalidad suponemos que  $g_{11}$  es el mínimo de los valores, dependiendo de en qué anillo euclidiano estemos trabajando es definido éste, podemos suponer esto ya que por intercambio de renglones y columnas, es posible llevar el mínimo a la entrada superior izquierda.
2. Si hay algún  $g_{1k} \neq 0$ ,  $k > 1$ , realizamos el algoritmo de la división para  $g_{1k}$  y  $g_{11}$  tal que  $g_{1k} = q_k g_{11} + r_{1k}$ , con  $0 \leq |r_{1k}| \leq |g_{11}|$ , donde  $|\cdot|$  es la función norma del anillo euclidiano  $\mathbb{E}$ .
3. Multiplicamos la primera columna de  $G$  por  $(-q_k)$  y sumamos a la co-

lumna  $k$  para así obtener:

$$G' = \begin{pmatrix} g_{11} & \cdots & g_{1k} - g_k g_{11} = r_{1k} & \cdots & g_{1c} \\ g_{21} & \cdots & g_{2k} - g_k g_{21} = r_{2k} & \cdots & g_{2c} \\ g_{31} & \cdots & g_{3k} - g_k g_{31} = r_{3k} & \cdots & g_{3c} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ g_{b1} & \cdots & g_{bk} - g_k g_{b1} = r_{bk} & \cdots & g_{bc} \end{pmatrix}$$

ésta es una matriz equivalente a  $G$ .

En general, cada vez que realicemos una operación cualquiera, suma, multiplicación, también se realizará sobre  $I_b$  o  $I_c$ , dependiendo si la operación es sobre renglón o columna. Si es sobre renglón se hará la misma operación sobre  $I_c$  y si es sobre columna lo reproduciremos sobre  $I_b$ .

Si  $r_{1k} = 0$ , repetimos este proceso con otro elemento diferente de cero de la misma columna, y si  $r_{1k} \neq 0$  regresamos al paso 1 con la matriz  $G'$ .

4. Repitiendo este proceso tantas veces como sea necesario y usando un proceso análogo sobre la primera columna obtenemos una matriz  $C$  equivalente a  $G$  y  $G'$

$$C = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & a_{2c} \\ \cdot & \cdot & \cdot & \cdots \\ 0 & a_{b2} & \cdots & a_{bc} \end{pmatrix}$$

5. Si  $a_{11}$  no divide a algún  $a_{ij}$ , sumando el renglón  $i$  al renglón 1, obtenemos

$$C' = \begin{pmatrix} a_{11} & a_{i2} & \cdots & a_{ic} \\ 0 & a_{22} & \cdots & a_{2c} \\ \cdot & \cdot & \cdot & \cdots \\ 0 & a_{b2} & \cdots & a_{bc} \end{pmatrix}$$

y regresamos nuevamente al paso 1.

6. Aplicando repetidamente los pasos del 1 al 5 obtenemos

$$D = \begin{pmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & d_{2c} \\ \cdot & \cdot & \cdot & \cdots \\ 0 & d_{b2} & \cdots & d_{bc} \end{pmatrix}$$

donde  $d_{11}$  divide a cualquier entrada  $d_{ij}$ .

7. Definimos  $e_1 = d_{11}$  y regresamos al paso 1 con la matriz

$$\begin{pmatrix} d_{22} & \cdots & d_{2c} \\ \cdot & \cdot & \cdot \\ d_{b2} & \cdots & d_{bc} \end{pmatrix}$$

Obteniendo así una matriz  $\Gamma$  equivalente a  $G$ ,  $\Gamma$  es una matriz diagonal, y como ya mencionamos, los  $\gamma_i$  son elementos diferentes de cero y son llamados los factores invariantes de  $G$  y satisfacen que  $\gamma_i$  divide a  $\gamma_{i+1}$  para toda  $i$ . Esta diagonalización se puede realizar debido a que estamos en un anillo euclidiano.

Al principio de la sección se hizo mención de las matrices identidad  $I_b$  e  $I_c$ , que preservan la invertibilidad pese a las operaciones básicas realizadas en ellas. Conjuntamente con las operaciones realizadas sobre  $G$  se realizan operaciones sobre  $I_b$  e  $I_c$ , ya que cuando realicemos una operación de columnas sobre  $G$  aplicamos la misma operación sobre  $I_b$  y cuando se haga una de renglones sobre  $G$  realizaremos la misma operación sobre  $I_c$ , esto permite que obtengamos una matriz  $A$  equivalente a  $I_b$  y una matriz  $B$  equivalente a  $I_c$ , de tal forma que se cumpla la igualdad y no sea solamente una equivalencia de matrices, es decir,

$$G = A\Gamma B.$$

A continuación realizaremos un ejemplo de la forma normal de Smith en el anillo euclidiano  $\mathbb{F}_2[x]$ . Sea la matriz  $G(x)$  de tamaño  $2 \times 3$ , con entradas en el anillo  $\mathbb{F}_2[x]$ , tal que:

$$G(x) = \begin{pmatrix} 1+x & x & 1 \\ x^2 & 1 & 1+x+x^2 \end{pmatrix},$$

necesitamos dos matrices identidad de tamaño  $b$  y  $c$ , es decir,  $I_2$  e  $I_3$ , la primera a la izquierda de  $G(x)$  y la segunda a la derecha de ésta.

La entrada más pequeña de  $G(x)$  es 1, así que llevaremos este elemento a la parte superior izquierda mediante el intercambio de columnas (la primera y la tercera),

$$\begin{pmatrix} 1+x & x & 1 \\ x^2 & 1 & 1+x+x^2 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & x & 1+x \\ 1+x+x^2 & 1 & x^2 \end{pmatrix}$$



para hacer cero la segunda entrada del primer renglón, y como  $x(1) = x$ ,

$$\begin{pmatrix} 1 & x & 1+x \\ 1+x+x^2 & 1 & x^2 \end{pmatrix} \begin{pmatrix} 1 & x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1+x \\ 1+x+x^2 & 1+x+x^2+x^3 & x^2 \end{pmatrix}$$

para hacer cero la tercera entrada del primer renglón, y como  $(1+x)(1) = 1+x$ , entonces:

$$\begin{pmatrix} 1 & 0 & 1+x \\ 1+x+x^2 & 1+x+x^2+x^3 & x^2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1+x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \\ \begin{pmatrix} 1 & 0 & 0 \\ 1+x+x^2 & 1+x+x^2+x^3 & 1+x^2+x^3 \end{pmatrix}$$

Ahora nos interesa hacer cero el primer término del segundo renglón, esto lo podemos ver como una operación sobre las columnas de la matriz. Y como  $(1+x+x^2)(1) = 1+x+x^2$ ,

$$\begin{pmatrix} 1 & 0 \\ 1+x+x^2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1+x+x^2 & 1+x+x^2+x^3 & 1+x^2+x^3 \end{pmatrix} = \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1+x+x^2+x^3 & 1+x^2+x^3 \end{pmatrix}$$

y ahora dividimos  $1+x^2+x^3$  por  $1+x+x^2+x^3$ , es decir,  $1+x^2+x^3 = (1+x+x^2+x^3)1+x$ , por lo cual

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1+x+x^2+x^3 & 1+x^2+x^3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} =1 & 0 & 0 \\ 0 & 1+x+x^2+x^3 & x \end{pmatrix}$$

cambiamos de orden la segunda y tercera columna

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1+x+x^2+x^3 & x \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & x & 1+x+x^2+x^3 \end{pmatrix}$$

mientras que  $1+x+x^2+x^3 = x(1+x+x^2) + 1$ ,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & x & 1+x+x^2+x^3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1+x+x^2 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & x & 1 \end{pmatrix}$$

nuevamente reordenando las columnas

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

finalmente:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & x \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & x \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \Gamma(x),$$

tal que:

$$G(x) = A(x)\Gamma(x)B(x),$$

que es igual a

$$\begin{pmatrix} 1 & 0 \\ 1+x+x^2 & 1 \end{pmatrix} \Gamma(x) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & x \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1+x+x^2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \\ \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1+x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

donde

$$\begin{pmatrix} 1 & 0 \\ 1+x+x^2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1+x & x & 1 \\ 1+x^2+x^3 & 1+x+x^2+x^3 & 0 \\ x+x^2 & 1+x+x^2 & 0 \end{pmatrix}.$$

## 1.5. Espacio de sucesiones

Sea  $A$  una estructura algebraica, grupo, anillo, campo, etc., y  $A^{\mathbb{Z}}$  el conjunto de funciones:

$$A^{\mathbb{Z}} = \{\psi : \mathbb{Z} \longrightarrow A; \psi(i) = a_i\}.$$

Si  $\psi \in A^{\mathbb{Z}}$  la imagen de  $i \in \mathbb{Z}$  se acostumbra denotarla por  $a_i$  y la función  $\psi$  se identifica con:

$$a = (\cdots, a_{-i}, a_{-i+1}, \cdots, a_0, a_1, \cdots, a_{i-1}, a_i, \cdots),$$

la cual se denomina *sucesión* sobre  $A$ . Es fácil ver que con la suma usual de funciones  $A^{\mathbb{Z}}$  es un grupo.

En este trabajo, particularmente nos interesan las **sucesiones binarias**, es decir, cuando  $A = \mathbb{Z}_2 = \mathbb{F}_2 = \{0, 1\}$ , las cuales pueden ser finitas o infinitas. Si son finitas las denotaremos con  $\mathbb{S}^n(\mathbb{F}_2) := \{0, 1\}^n$  o  $\mathbb{F}_2^n$ ,  $n \in \mathbb{N}$  lo cual significa que son de longitud  $n$ ,  $n \in \mathbb{N}$ . Si son infinitas las denotaremos con  $\mathbb{S}_\infty(\mathbb{F}_2) := \{0, 1\}_\infty$ :

$$\mathbb{S}^n(\mathbb{F}_2) = \{(a_0, a_1, \dots, a_{n-1}) : a_i \in \mathbb{F}_2\},$$

$$\mathbb{S}_\infty(\mathbb{F}_2) = \{(\dots, a_{-i}, a_{-i+1}, \dots, a_0, a_1, \dots, a_n, \dots) : a_i \in \mathbb{F}_2\}.$$

Obsérvese que  $\{0, 1\}^n \subset \{0, 1\}_\infty$ .

Considérese la siguiente función:

$$\phi : \mathbb{S}^n(\mathbb{F}_2) \longrightarrow \mathbb{F}_2[x]$$

$$a = (a_0, \dots, a_{n-1}) \longrightarrow a(x) = a_0 + \dots + a_{n-1}x^{n-1}, \quad a \in \mathbb{S}^n(\mathbb{F}_2).$$

Es fácil ver que  $\phi$  es un homomorfismo inyectivo de  $\mathbb{F}_2$ -espacios vectoriales, cuya imagen es el conjunto de polinomios de grado a lo mas  $n - 1$ , y por lo tanto es biyectiva en su imagen. De manera análoga se puede definir un homomorfismo

$$\phi_\infty : \mathbb{S}_\infty(\mathbb{F}_2) \longrightarrow \mathbb{F}_2[[x]]$$

$$(a_0, \dots, a_{n-1}, \dots) \longrightarrow a_0 + \dots + a_{n-1}x^{n-1} + \dots,$$

es decir, a cada sucesión infinita se le asocia una serie de potencias.

Con las funciones  $\psi$  y  $\phi$ , a cada sucesión finita de longitud  $n$  se le asocia un polinomio de grado  $\leq n - 1$ ; y a toda sucesión infinita se le asocia una serie de potencia, respectivamente. Estas funciones son de gran importancia en el desarrollo de este trabajo.

Por otro lado, es importante para este trabajo retomar algunas particularidades en ciertos anillos (varias de estas ya se mencionaron): polinomios, serie de potencias y series de Laurent, en estos definiremos el momento, el cual hace referencia al subíndice en el cual inicia la sucesión, por ejemplo, un polinomio:

$$p(x) = a_0 + a_1x + \dots + a_nx^n,$$

diremos que inicia en momento  $i = 0$  y representará a una sucesión finita  $p = (a_0, a_1, \dots, a_n)$ , mientras que una serie de potencias

$$q(x) = a_0 + a_1x + \dots + a_nx^n + \dots,$$

inicia en momento  $i = 0$ , sin embargo es infinita.

Las series de Laurent son infinitas

$$r(x) = a_{-n}x^{-n} + \dots + a_0 + a_1x + \dots + a_nx^n + \dots,$$

e inician en el momento  $i = -n, n \in \mathbb{N}$ , tal que  $r = (a_n, \dots, a_0, a_1, \dots, a_n, \dots)$ .

El anillo de polinomios  $\mathbb{F}_2[x]$  está contenido en el anillo de las series de Laurent,  $\mathbb{F}_2[x] \subset \mathbb{F}_2((x))$ .

Para fines prácticos y debido a las convenientes propiedades del anillo de polinomios  $\mathbb{F}_2[x]$ , es preferible trabajar en este que en  $\mathbb{F}_2((x))$ .

En toda serie de Laurent

$$w(x) = w_{-t}x^{-t} + \dots + w_{-1}x^{-1} + w_0 + w_1x + w_2x^2 + \dots + w_rx^r + \dots,$$

podemos renombrar las variables como

$$x^{-(t-i)} = y^i,$$

obteniendo:

$$w_{-t} + w_{-t+1}y + \dots + w_{-1}y^{t-1} + w_0y^t + w_1x^{t+1} + w_2x^{t+2} + \dots + w_rx^{t+r} + \dots,$$

de cierta manera recorreremos el índice en el cual inicia la sucesión y obtenemos una serie de potencias, las cuales son más convenientes para las aplicaciones de los códigos de convolución.

# Capítulo 2

## Ecuaciones de paridad y polinomios generadores

En el capítulo 2 se recordará la operación de convolución y se definirán los códigos de convolución, se darán sus parámetros principales y algunas de sus propiedades. Se introducirán las ecuaciones de paridad, polinomios generadores de estos códigos y la relación entre ellos.

### 2.1. La operación de convolución

Una pregunta natural que surge al hablar de los códigos de convolución es: ¿Qué relación tienen los códigos de convolución con esta operación?

Para introducir los códigos de convolución es de gran importancia recordar la operación de *convolución*, ya que precisamente de ella heredan el nombre.

Frecuentemente, en textos de disciplinas de ingeniería, física y matemáticas, podemos encontrar el término *convolución*. Es común encontrar este término ya que se relaciona con numerosas aplicaciones de estas áreas. Por ejemplo: en el estudio de imágenes digitales, la aplicación de un filtro espacial a una imagen se explica por el proceso de convolución que da como resultado la imagen filtrada; en acústica un eco es la convolución del sonido original con una función que represente los objetos variados que lo reflejan; en ingeniería eléctrica, electrónica y otras disciplinas, la salida de un sistema lineal ya sea estacionario o con tiempo-invariante es la convolución de la entrada con la

respuesta del sistema a un impulso, por mencionar algunos ejemplos, [21], [22]. En estos ejemplos se usan sucesiones (cortas o largas) y se tiene la convolución de sucesiones.

A grandes rasgos, diremos que una convolución, la cual se denota con “\*”, es un operador matemático que transforma dos funciones del mismo tipo,  $f$  y  $g$ , en una tercera función que en cierto sentido representa la magnitud en la que se superponen  $f$  y  $g$ , es decir,

$$f[t] * g[t] = h[t],$$

a la función  $f[t]$  le llamamos de entrada y a  $g[t]$  función de respuesta, mientras que a  $h[t]$ , que esta determinada por la función  $g[t]$  y la función de entrada, [24].

Dependiendo de cómo sea el dominio, es decir, qué valores puede tomar el parámetro  $t$  podemos hablar de convolución continua o discreta, para el caso de los códigos de convolución nos interesa la convolución discreta.

El dominio de una función discreta  $f$  es un conjunto numerable (discreto), es decir,

$$f : \mathbb{M} \subset \mathbb{N} \longrightarrow \mathbb{M}'.$$

Si  $\mathbb{F}$  es un conjunto de funciones discretas la operación de convolución está definida de la siguiente manera:

$$* : \mathbb{F} \times \mathbb{F} \longrightarrow \mathbb{F}$$

$$(f, g) \longrightarrow h := f * g,$$

dada por:

$$h[t] = f[t] * g[t] = \sum_n f[n]g[t - n]$$

En los códigos de convolución el mensaje de entrada es la información que deseamos codificar, mientras que la salida es la palabra codificada, por lo cual veremos a las sucesiones, polinomios o series de potencias como funciones. Recordemos que las sucesiones pueden ser de longitud finita, no necesariamente de la misma longitud,  $\mathbb{S}^n(\mathbb{F}_2) := \{0, 1\}^n$  o  $\mathbb{F}_2^n$ ,  $n \in \mathbb{N}$ , y de longitud infinita,  $\mathbb{S}_\infty(\mathbb{F}_2) := \{0, 1\}_\infty$ .

Si se está trabajando en  $\mathbb{S}_\infty(\mathbb{F}_2)$ , la operación convolución es:

$$\begin{aligned} * : \mathbb{S}_\infty(\mathbb{F}_2) \times \mathbb{S}_\infty(\mathbb{F}_2) &\longrightarrow \mathbb{S}_\infty(\mathbb{F}_2) \\ (u, g) &\longrightarrow u * g = v. \end{aligned}$$

donde  $u = u_0u_1u_2 \cdots$ ,  $g = g_0g_1g_2 \cdots$  y

$$v_i = \sum_{j=0}^{\infty} u_j g_{i-j}.$$

Si  $u, g$  son sucesiones cortas las cuales se pueden suponer de la misma longitud completando con ceros,

$$\begin{aligned} * : \mathbb{S}^n(\mathbb{F}_2) \times \mathbb{S}^n(\mathbb{F}_2) &\longrightarrow \mathbb{S}^n(\mathbb{F}_2), \\ u * g = v = v_0v_1 \cdots v_s &\in \mathbb{S}^n(\mathbb{F}_2), \\ v_i = \sum_{j=0}^i u_j g_{i-j}, & i = 1, 2, \cdots, n. \end{aligned}$$

Esta ecuación es de particular interés para nosotros, ya que es la que determina el nombre de los códigos de convolución.

En los anillos  $\mathbb{F}_2[x]$  y  $\mathbb{F}_2[[x]]$  la convolución coincide con la multiplicación de elementos del anillo. Veremos esto para el anillo  $\mathbb{F}_2[x]$ , el otro caso es análogo.

Sean  $a(x)$  y  $b(x)$  dos polinomios cualesquiera en  $\mathbb{F}_2[x]$ , de grado  $n$  y  $m$  respectivamente,  $m, n \in \mathbb{N}$ , y sin pérdida de generalidad, digamos, que  $n \leq m$ . Hagamos la operación producto, que hace a  $\mathbb{F}_2[x]$ , junto con la suma un anillo,

$$\begin{aligned} a(x) &= a_0 + a_1x + a_2x^2 + \cdots + a_mx^m, \\ b(x) &= b_0 + b_1x + b_2x^2 + \cdots + b_nx^n, \end{aligned}$$

entonces  $a(x) \cdot b(x) := c(x)$ , donde:

$$c(x) = c_0 + c_1x + c_2x^2 + \cdots + c_mx^m + c_{m+1}x^{m+1} + \cdots + c_{m+n}x^{m+n},$$

$$\text{grad}(c(x)) = n + m,$$

donde  $grad(c(x))$  es el grado del polinomio  $c(x)$ , y

$$c_k = \sum_{i=0}^k a_i b_{k-i}, \quad k = 0, 1, \dots, n+m, \quad k = 0, 1, \dots, n+m.$$

Se puede ver que con la operación de convolución en las sucesiones finitas,  $(\mathbb{S}^n, +, *)$ , y los polinomios con el producto usual de polinomios,  $(\mathbb{F}_2[x], +, \cdot)$ , identificando a una sucesión (finita) con un polinomio, ambos anillos son isomorfos. Por lo tanto, usar las sucesiones con la convolución es lo mismo que usar polinomios con su producto usual.

Mas adelante se verá porque los códigos de convolución deben su nombre a la ecuación de convolución discreta:

$$c_k = \sum_{i=0}^k a_i b_{k-i}, \quad k = 0, 1, \dots, n+m, \quad k = 0, 1, \dots, n+m,$$

## 2.2. Códigos de convolución

Sean  $\mathbb{F}_2[x]$  y  $\mathbb{F}_2[[x]]$  el anillo de polinomios y de series de Laurent, las cuales, como se mencionó anteriormente se pueden identificar con las sucesiones binarias cortas y largas respectivamente, y sea  $\mathbb{F}_2(x)$  el campo de las funciones racionales.

**Definición.** Sea  $G(x)$  una matrix  $b \times c$ ,  $b, c \in \mathbb{N}$ ,  $b \leq c$  con entradas en el campo de las funciones racionales. El código de convolución  $C$ , con matriz generadora  $G(x)$ , es la imagen de la transformación lineal

$$G(x) : \mathbb{F}_2(x)^b \longrightarrow \mathbb{F}_2(x)^c$$

$$u(x) \mapsto v(x) = u(x)G(x).$$

Es decir,

$$C = \{v(x) \in \mathbb{F}_2((x))^c : v(x) = u(x)G(x), \quad u(x) \in \mathbb{F}_2((x))^b\},$$

a  $G(x)$  se le llama matrix *generadora* del código de convolución  $C$ .

Esta definición de código de convolución generaliza a los códigos de bloque, ya que se puede tomar  $\mathbb{F}_2^b$  y  $\mathbb{F}_2^c$  y la matrix con entradas en  $\mathbb{F}_2$ .



Un ejemplo de una matriz generadora  $G(x)$ ,  $2 \times 3$  es:

$$G(x) = \begin{pmatrix} x & x^2 & 1 \\ 1+x & 1+x^3 & 1+x+x^2 \end{pmatrix},$$

donde, dado cualquier vector de información,  $1 \times 2$ ,  $u(x) = (u_1(x), u_2(x))$  por la matriz, es decir,

$$C = \{v(x) = u(x)G(x) = (v_1(x), v_2(x), v_3(x)) =$$

$$(u_1(x)x + u_2(x)(1+x), u_1(x)x^2 + u_2(x)(1+x^3), u_1(x)1 + u_2(x)(1+x+x^2))\}.$$

Cuando varia el vector de información también  $\frac{1}{2}$  en lo hace el vector codificado (de 3 componentes). Este código es de tasa  $R = 2/3$ .

### 2.3. Conceptos preliminares de códigos de convolución

En esta subsección veremos los conceptos preliminares de los códigos de convolución, los cuales se irán desarrollando y ejemplificando a lo largo de las siguientes secciones.

Mencionamos que los mensajes a codificar son polinomios, sucesiones, series de potencias o series de Lauren, en caso de no ser una sucesión, por medio de un isomorfismo los podemos ver como sucesiones, cuyas entradas son bits, bajo esta idea, cuando se codifica por cada  $b$  bits de información que entran al codificador convolutivo, el codificador produce  $c$  bits de información de salida, cuando su matriz generadora es de tamaño  $b \times c$ . Lo anterior determina lo que se llama la tasa del código,

$$R = \frac{b}{c},$$

$c, b \in \mathbb{N}$ ,  $b \leq c$ . En la implementación de los códigos de convolución generalmente se usa la desigualdad estricta  $b < c$  con  $b, c$  pequeñas y por lo regular se codifica bit a bit, es decir,

$$R = 1/c.$$

Las tasas más comunes encontradas en la práctica y en la literatura son  $R = \frac{1}{2}$ ,  $R = \frac{1}{3}$  y ocasionalmente  $R = \frac{2}{3}$ .

Se puede pensar que se tiene una “ventana” de longitud  $k$  que cubre los  $b$  bits del codificador mas otra cantidad fija de estos, diremos  $m$  de ellos, esta ventana contiene no solo la información que la tasa  $R$  proporciona al codificador, también una cantidad fija de bits necesarios para que la ventana este llena y pueda darse el proceso de codificación, lo que implica que la información de salida no sólo dependa del número de entradas por unidad de tiempo (índice que más adelante identificaremos en cada codificador), sino también del bloque de  $m$  bits anteriores, a  $m$  le llamaremos la memoria del código.

A  $k$  le llamaremos la longitud de restricción del código de convolución, y es un parámetro más, que depende directamente de  $b$ , tal que:

$$\underbrace{b + m} = k,$$

por lo cual se satisface  $k \in \mathbb{N}$  y  $m < k$ .

$m$  es un parámetro de gran importancia, ya que este permite que la información codificada en cada instante, de cierta forma “herede” características generadas por información codificada con anterioridad.

Un ejemplo de parámetros de código de convolución es  $(2, 1, 3)$ , cuya memoria es  $m = 3$  y por cada bit que entra al codificador este produce dos bits de salida, así que  $R = 1/2$ .

Casos particulares son cuando  $m = 0$ ,  $k = b$ , ya que en este caso se tiene un código de bloque, debido a que la información que se procesa no depende de información previa para ser codificada, es decir, la memoria es nula. Otro caso particular es cuando  $k = 1$ , ya que  $m = 0$  y  $b = 1$ , que indica que la codificación es bit a bit y la secuencia de información no es dividida en bloques, se codifica de una manera continua. Un caso particular más es cuando se cumple la igualdad  $b = c$ , lo que implica  $R = 1$ , es decir, que por cada  $b$  bits de información tenemos  $b$  bits de salida, por ejemplo las tasas  $R = \frac{2}{2}$  y  $R = \frac{3}{3}$  que generan códigos de bloque.

Generalmente en los códigos de convolución, con  $b \leq c$ , la longitud del mensaje codificado tienen mayor longitud que el mensaje que se codificó.

A excepción de cuando se genera un código de bloque.

En general si la secuencia binaria a codificar es de la forma  $u = u_0u_1 \cdots u_{l-1} \in \mathbb{F}_2^l$ , es importante el orden y etiquetamiento de las componentes de estas sucesiones, lo cual se hace de izquierda a derecha, el primer bit de la izquierda  $u_0$  ocupará la posición 1, el siguiente bit tiene la posición 2 y así sucesivamente hasta llegar a la posición  $l$ -ésima, que es el bit de la derecha  $u_{l-1}$ ,

$$\begin{array}{cccc} u_0 & u_1 & u_2 & \cdots u_{l-1} \\ \uparrow & \uparrow & \uparrow & \cdots \uparrow \\ 1 & 2 & 3 & \cdots l. \end{array}$$

Otra notación que es común usar es:

$$u(0)u(1)u(2) \cdots u(l-1),$$

dependiendo de lo que necesitaremos los etiquetamos de la forma más conveniente.

Ahora, supongamos que la ventana que cubre los bits es desplazante y que como ya se mencionó, cubre una cantidad fija de bits,  $k = b + m$ , y que pese a los desplazamientos esta ventana siempre admite el mismo número de bits. Sin pérdida de generalidad podemos convenir que la ventana se desplaza hacia la derecha. Esta ventana acordaremos que se desplaza  $b$  bits en cada movimiento, si  $b = 1$  se codifica bit a bit, que es lo más usual, pero no necesariamente  $b = 1$ .

Ilustraremos lo anterior con un ejemplo. Sea el mensaje  $u = 10011$  y el codificador que usamos tiene una memoria  $m = 2$  y una tasa  $R = 1/2$ , es decir, que la ventana se desplazará solamente una posición en cada movimiento. La ventana contiene los bits de la memoria y los que entran al codificador  $b + m = 3 = k$ .

En el primer desplazamiento la ventana cubrirá los bits:  $\boxed{1 \ 0 \ 0}$ , en el segundo desplazamiento  $\boxed{0 \ 0 \ 1}$ , y finalmente en el tercero  $\boxed{0 \ 1 \ 1}$ , pero de esta manera no estamos codificando el primer ni el último bit, para codificarlo agregamos bits a la derecha e izquierda del mensaje que deseamos codificar. Necesitamos bits que no alteren la información original, así que agregamos tantos ceros a la izquierda y derecha como sean necesarios para

llenar la ventana, de esta manera podemos codificar el primer y último bit. En nuestro caso tenemos que agregar 2 bits en cada extremo:

$$u = 001001100,$$

y ahora lo podemos ver como  $u \in \mathbb{F}_2^9$ . Efectivamente, los ceros agregados a la derecha e izquierda del mensaje no afectan la información, la transmisión ni la codificación de esta, solo la longitud. Por lo tanto la ventana en un inicio  $i = 0$  contendrá a 

0	0	1
---	---	---

 en el primer movimiento  $i = 1$ : 

0	1	0
---	---	---

, en el segundo  $i = 2$ : 

1	0	0
---	---	---

, después en  $i = 3$  a 

0	0	1
---	---	---

, posteriormente en  $i = 4$  a 

0	1	1
---	---	---

, el siguiente  $i = 5$ : 

1	1	0
---	---	---

, para finalizar con  $i = 6$ : 

1	0	0
---	---	---

.

Ahora supongamos que tenemos este mismo mensaje con el mismo tamaño de ventana,  $k = 3$ , pero con una tasa  $R = 2/3$ ,  $b = 2$  y  $c = 3$ , es decir, en cada desplazamiento entran 2 bits y salen 3, en el primer movimiento entran dos bits, para llenar la ventana solo tenemos que agregar un cero a la derecha del mensaje  $u$  y otro a su izquierda,  $u = 0100110$ , entonces, la ventana inicialmente  $i = 0$  contendrá 

0	1	0
---	---	---

, enseguida en  $i = 1$  a 

0	0	1
---	---	---

 y posteriormente en  $i = 2$  a 

1	1	0
---	---	---

, en este caso solo tenemos tres etapas de la ventana.

En el primer caso de las ventanas, donde  $b = 1$ , siempre hay dos bits que estuvieron en la ventana anterior, ya que  $m = 2$  es el tamaño de la memoria, en cambio, para  $b = 2$  tenemos que un bit es el que se mantiene de la ventana anterior, por lo cual  $m = 1$ . En general tendremos:

$$k = b + m,$$

o equivalentemente

$$m = k - b.$$

## 2.4. Polinomios generadores

Toda matriz generadora  $G(x)$ ,  $b \times c$ , de un código de convolución, asociada a una tasa  $R = b/c$ , en caso de tener entradas racionales puede siempre

llevarse a una matriz equivalente de entradas polinomiales, es decir,

$$G(x) = \begin{pmatrix} g_{11}(x) & g_{12}(x) & \cdots & g_{1c}(x) \\ g_{21}(x) & g_{22}(x) & \cdots & g_{2c}(x) \\ \cdot & \cdot & \cdots & \cdots \\ g_{b1}(x) & g_{b2}(x) & \cdots & g_{bc}(x) \end{pmatrix}$$

donde cada elemento  $g_{ij}(x)$  es un polinomio con coeficientes binarios.

$$g(x) = \sum_{i=0}^m g_i x^i, \quad m \in \mathbb{N}, \quad g_i \in \mathbb{F}_2.$$

En particular nos interesan las tasas  $R = 1/c$ , lo que indica matrices generadoras de tamaño  $1 \times c$ :

$$G(x) = (g_{11}(x) \quad g_{12}(x) \quad \cdots \quad g_{1c}(x)).$$

es decir,  $c$  polinomios conforman un codificador, el cual es el encargado de generar las palabras que conforman el código. Para codificar se multiplica  $u(x)$  por cada polinomio generador y posteriormente se concatenan

$$u(x)g_n(x) := v_n(x), \quad n = 1, \cdots, c,$$

los polinomios regularmente se acomodan en forma matricial, por lo que el conjunto de polinomios generadores puede generalizarse como una matriz, sin embargo es importante mencionarse como otro tipo de codificador, ya que en la literatura estos son encontrados con frecuencia.

En cada polinomio generador la memoria  $m$  coincide con el grado del polinomio. Por ejemplo, sea el polinomio generador  $1 + x^2 + x^3$  al que denotaremos con  $g_1(x)$ , la memoria de este polinomio es  $m = 3$ , y el mayor grado del sistema de polinomios es la memoria del codificador.

Se toma el mayor de los grados del conjunto de ecuaciones, ya que la ventana tiene que cubrir el mensaje multiplicado por cada uno de los elementos, incluido el término de mayor grado, pero como esta ventana es fija y la memoria es fija, entonces tiene que tomar el tamaño de  $b$  mas el grado mayor del polinomio para poder cubrir todos los elementos. Como estos elementos se llevan generalmente a forma matricial, sucede lo mismo en estas, se toma

el mayor grado de los polinomios de la matriz.

**Ejemplo.** El codificador de un código de convolución de tasa  $R = 1/3$  esta compuesto por los polinomios:

$$\begin{aligned}g_1(x) &= 1 + x, \\g_2(x) &= 1 + x + x^3, \\g_3(x) &= x + x^2,\end{aligned}$$

y se desea codificar el mensaje  $u = 101 \in \mathbb{F}_2^3$ . Para que la multiplicación del mensaje  $u$  con los polinomios generadores tenga sentido es necesario multiplicar elementos de la misma estructura sobre  $\mathbb{F}_2$ , así que le asociamos su elemento correspondiente en  $\mathbb{F}_2[x]$ , es decir,  $u(x) = 1 + x^2$ , entonces

$$\begin{aligned}v_1(x) &= u(x)g_1(x) = 1 + x + x^2 + x^3, \\v_2(x) &= u(x)g_2(x) = 1 + x + x^2 + x^5, \\v_3(x) &= u(x)g_3(x) = x + x^2 + x^3 + x^4,\end{aligned}$$

que lo podemos ver como:

$$\begin{aligned}u(x)(g_1(x), g_2(x), g_3(x)) &= (v_1(x), v_2(x), v_3(x)) \\&= (1 + x + x^2 + x^3, 1 + x + x^2 + x^5, x + x^2 + x^3 + x^4),\end{aligned}$$

al que le podemos asociar  $v = 1111, 111001, 01111$ . Como las longitudes no son las mismas en los 3 bloques se acostumbra tomar la longitud mayor de los bloques,

$$v = 111100, 111001, 011111.$$

El grado de los polinomios generadores es 1, 3 y 2 respectivamente, que implica que el codificador tiene memoria 3, ya es el mayor de los grados.

Para distinguir cada uno de los polinomios generadores, los denotaremos como:

$$g_n(x) = \sum_{i=0}^{m_n} g_i^{(n)} x^i, \quad m_n = \text{grado}(g_n(x)), \quad g_i^{(n)} \in \mathbb{F}_2, \quad n = 1, 2, \dots, c,$$

mientras que la memoria del codificador es:

$$m = \max\{m_1, m_2, \dots, m_c\}.$$

## 2.5. Ecuaciones de paridad

Dada una sucesión (corta)  $u = (u_0, u_1, \dots, u_n)$  y otra sucesión fija  $g = (g_0, g_1, \dots, g_m)$ , a las cuales le aplicamos la operación de convolución, obtenemos  $u * g$ , que generalmente se expresa como:

$$p_n(i) = \sum_{j=0}^i g_j \cdot u_{i-j} \in \mathbb{F}_2, \quad n = 1, 2, \dots, c, \quad (2.1)$$

sin embargo, en esta sección le daremos otra presentación, a la cual le llamaremos ecuación de paridad. Las ecuaciones de paridad representan de manera directa la ecuación de convolución.

En los códigos de convolución las ecuaciones de paridad, cada una de ellas genera un bit de salida por cada  $b$  bits que entran al codificador. Estas ecuaciones generalmente se usan para tasas  $R = b/c$  con  $b = 1$ , bit a bit, esto no quiere decir que para  $b \neq 1$  no se puedan usar, solamente al usarla hay que tomar ciertas consideraciones.

En general, para usar las ecuaciones de paridad, si tenemos el mensaje

$$u = (u_0, u_1, u_2, \dots, u_{l-1}),$$

con una tasa  $R = b/c$ ,  $b \leq c$ , en el primer movimiento entran los primeros  $b$  bits, pero para codificar correctamente todos los  $u_i$  agregamos a la derecha e izquierda de  $u$ ,  $k - b$  ceros, es decir,  $m$  el tamaño de la memoria, para llenar la ventana

$$u = (\underbrace{0, \dots, 0}_{k-b}, u_0, u_1, u_2, \dots, u_{l-1}, \underbrace{0, \dots, 0}_{k-b}).$$

Este cambio no altera la información, solo tendrá repercusión en la longitud del mensaje final, ya que este cambio hace que

$$u \in \mathbb{F}_2^{l+2k-2b},$$

$$\mathbb{F}_2^{l+2k-2b} = \mathbb{F}_2^{l+2m}.$$

En las ecuaciones de paridad el número de ecuaciones esta determinado por  $c$ , el mismo  $c$  de la tasa  $R = b/c$ ,  $b = 1$ , (para  $b \neq 1$  tendremos  $bc$  ecuaciones)

y como cada ecuación genera un bit de redundancia por cada  $i$ , tendremos  $c$  bits por cada unidad de tiempo  $i$ .

Haciendo un cambio de variables en (2.1), tal que  $u = (u(r), u(r+1), \dots, u(n))$  y  $g = (g(0), g(1), \dots, g(m))$ , obtenemos:

$$p_n(i) = \sum_{j=0}^{k-b} g(j) \cdot u(i-j) \in \mathbb{F}_2, \quad (2.2)$$

que es la convolución de la sucesión  $g_i = g(i)$  con la sucesión  $u_t = u(t)$ .

Desarrollando la relación anterior obtenemos la forma de las ecuaciones de paridad:

$$p_n(i) = g(0)u(i) + g(1)u(i-1) + \dots + g(k-b)u(i-k+b),$$

los  $g(j) \in \mathbb{F}_2$  son fijos y están determinados por la sucesión  $g(j) = g(0), \dots, g(k-b)$ , a la que llamamos sucesión generadora. Si  $i-j < 0$  asignamos el valor

$$u(i-j) := 0. \quad (2.3)$$

En el conjunto de polinomios que determinan un codificador de un código de convolución ( $b = 1$ ),  $R = \frac{1}{c}$ , el mayor grado de estos es la memoria, en el conjunto de ecuaciones de paridad, según la biyección la memoria es el mayor de los  $k-b$  restados en el conjunto de  $c$  ecuaciones.

A continuación realizaremos un ejemplo de un codificador, usando ecuaciones de paridad y posteriormente probaremos mas formalmente la biyección entre un polinomio generador y una ecuación de paridad.

**Ejemplo:** Queremos transmitir el mensaje

$$u = 11001,$$

para codificar este mensaje lo haremos bit a bit,  $b = 1$ , y por cada bit que entre al codificador tendremos 3 bits de salida determinados por las ecuaciones de paridad:

$$p_1(i) = u(i) + u(i-1) + u(i-2),$$

$$p_2(i) = u(i) + u(i-2) + u(i-3),$$



$$p_3(i) = u(i) + u(i - 3),$$

estas tres ecuaciones conforman el codificador del código de convolución, y a cada ecuación le podemos asociar

$$p_1(i) \longleftrightarrow (1, 1, 1, 0)$$

$$p_2(i) \longleftrightarrow (1, 0, 1, 1)$$

$$p_3(i) \longleftrightarrow (1, 0, 0, 1)$$

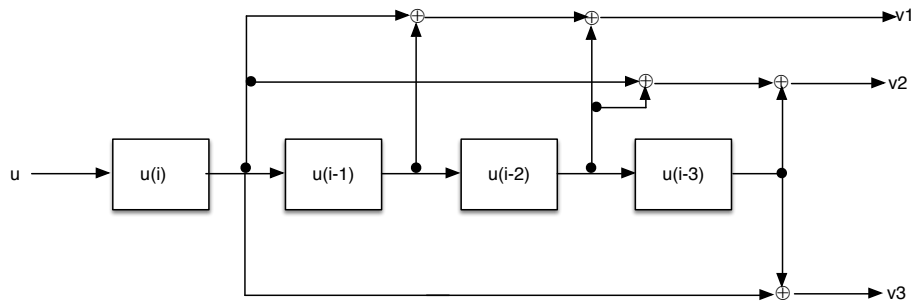


Figura 2.1: Diagrama asociado a este código.

El sistema lo forman las tres ecuaciones:  $p_1(i)$ ,  $p_2(i)$  y  $p_3(i)$ , asociado a una tasa  $R = 1/3$ . La tasa  $R = 1/3$  implica que por cada entrada al codificador obtendremos 3. En el diagrama  $u = (u(i), u(i-1), u(i-2), u(i-3))$  es la información que entra al codificador, mientras que las salidas están representadas como:  $v = (v(1), v(2), v(3))$ , y cada una esta generada por una ecuación de paridad. En el diagrama hay cuatro cajas, la primera representa a  $u(i)$ , la segunda a  $u(i-1)$ , la tercera a  $u(i-2)$  y la cuarta a  $u(i-3)$ , la salida  $v(1)$  podemos observar que esta formada por la suma de las cajas  $u(i)$ ,  $u(i-1)$  y  $u(i-2)$ , es decir, esta generada por la ecuación  $p_1(i)$ , mientras que  $v(2)$  corresponde a la suma de  $u(i)$ ,  $u(i-2)$  y  $u(i-3)$  que se asocian a la ecuación  $p_2(i)$ , y finalmente  $v(3)$  corresponde a la suma de las cajas  $u(i)$  y  $u(i-3)$  que corresponden a la ecuación  $p_3(i)$ . En este caso, cada conjunto de  $u(i)$ ,  $u(i-1)$ ,  $u(i-2)$  y  $u(i-3)$  generan 3 bits. Cada ecuación genera un bit por cada  $u$  y por cada  $i$ .

Tenemos 3 ecuaciones de paridad y  $b = 1$ , entonces la tasa asociada a este código es  $R = \frac{1}{3}$ . De las ecuaciones podemos observar que la memoria  $m = 3$  y como  $b = 1$ , entonces implica que  $m + b = 4 = k$ , ya que la ventana tiene que cubrir el bit entrante en cada movimiento más los 3 bits de la memoria.

La ventana contendrá en un inicio solo al primer bit del mensaje  $u$ , para codificar ese primer bit agregamos  $k - b = 3$  ceros a la izquierda del mensaje y tres ceros a la derecha, por lo tanto la nueva  $u$  es:

$$00011001000,$$

que tiene longitud  $11=5+3+3=l + (k - 1) + (k - 1) = l + 2k - 2$ .

La ventana en  $i = 0$  contiene los bits 0001, después se desplazara un lugar a la derecha y contendrá otros bits diferentes. La tabla siguiente muestra los bits que contendrá en el tiempo correspondiente

$i$	$u(i - 3)u(i - 2)u(i - 1)u(i)$
0	0001
1	0011
2	0110
3	1100
4	1001
5	0010
6	0100
7	1000

(2.4)

Si aplicamos las ecuaciones de paridad para  $i = 0$ , tenemos

$$p_1(0) = u(0) + u(-1) + u(-2) = 1 + 0 + 0 = 1,$$

$$p_2(0) = u(0) + u(-2) + u(-3) = 1 + 0 + 0 = 1,$$

$$p_3(0) = u(0) + u(-3) = 1 + 0 = 1,$$

para  $i = 1$ , 

0	0	1	1
---	---	---	---

$$p_1(1) = 1 + 1 + 0 = 0,$$

$$p_2(1) = 1 + 0 + 0 = 1,$$

$$p_3(1) = 1 + 0 = 1,$$

para  $i = 2$  la ventana contiene  $\boxed{0 \mid 1 \mid 1 \mid 0}$  y obtenemos 010, con  $i = 3$ ,  $\boxed{1 \mid 1 \mid 0 \mid 0}$  produce los bits 101, con  $i = 4$ :  $\boxed{1 \mid 0 \mid 0 \mid 1}$  obtenemos 100, para  $i = 5$ ,  $\boxed{0 \mid 0 \mid 1 \mid 0}$  las ecuaciones producen 100, para  $i = 6$ ,  $\boxed{0 \mid 1 \mid 0 \mid 0}$  obtenemos 110, y finalmente  $i = 7$ ,  $\boxed{1 \mid 0 \mid 0 \mid 0}$  arroja 011.

Antes de dar el mensaje codificado, veamos que existen dos formas de ordenarlo.

La primera es,

$$\overline{p_1}(i), \overline{p_2}(i), \overline{p_3}(i),$$

donde

$$(p_1(0), p_1(1), p_1(2), p_1(3), p_1(4), p_1(5), p_1(6), p_1(7)) =: \overline{p_1}(i),$$

$$(p_2(0), p_2(1), p_2(2), p_2(3), p_2(4), p_2(5), p_2(6), p_2(7)) =: \overline{p_2}(i),$$

$$(p_3(0), p_3(1), p_3(2), p_3(3), p_3(4), p_3(5), p_3(6), p_3(7)) =: \overline{p_3}(i),$$

Para dar la segunda forma definimos la operación  $\oplus$ : si  $A = (a_1, a_2, \dots, a_n)$  y  $B = (b_1, b_2, \dots, b_n)$ , entonces:

$$A \oplus B := (a_1 b_1, a_2 b_2, \dots, a_n b_n).$$

Así,

$$v = \overline{p_1}(i) \oplus \overline{p_2}(i) \oplus \overline{p_3}(i) = (p_1(0)p_2(0)p_3(0), \dots, p_1(7)p_2(7)p_3(7)),$$

que es la codificación de la palabra  $u$ . En nuestro ejemplo:

$$v = (111 \ 011 \ 010 \ 101 \ 100 \ 100 \ 110 \ 011),$$

que tiene longitud  $24 = 3 \times (5 + 3)$ . La longitud de la palabra codificada es relativamente sencillo saberla, ya que por cada bit que entra a cada ecuación de paridad genera un bit de salida, tenemos  $c = 3$  de estas, además la longitud del mensaje inicial es  $l = 5$  y la memoria es  $m = 3$ . En general la longitud del mensaje codificado es:

$$c(l + m).$$

Recapitulemos: para codificar el mensaje  $u$  de longitud  $l$ , con un código de convolución de tasa  $R = b/c$  y con un codificador formado por  $c$  ecuaciones de paridad:

$$p_1(i), \dots, p_c(i).$$

Cada ecuación de paridad genera un bit por cada  $b$  de entrada, pero hay  $c$  ecuaciones, por lo cual por cada  $b$  bits obtendremos  $c$  bits de salida, que son los bits codificados.

Ahora veremos como a partir de una matriz generadora podemos pasar a un sistema de ecuaciones de paridad, es decir, dos codificadores diferentes son equivalentes. Sea  $C$  un código de convolución de tasa  $R = b/c$  con matriz generadora  $G(x)$  de tamaño  $\frac{1}{2} b \times c$ . Por ejemplo si  $R = 1/3$  y  $G(x)$ :

$$G(x) = [1 + x^2, 1 + x, 1 + x + x^2],$$

ahora bien, a todo mensaje  $u$  de  $longitud(u) = l$ , le podemos asociar un elemento  $u(x)$  del anillo  $\mathbb{F}_2[x]$ ,

$$u(x) = \sum_{j=0}^{l-1} u_j x^j.$$

Entonces,

$$u(x)G(x) = v(x) = (v_1(x), v_2(x), v_3(x)),$$

donde  $v_1(x) = u(x)(1 + x^2)$ ,  $v_2(x) = u(x)(1 + x)$  y  $v_3(x) = u(x)(1 + x + x^2)$ .

Efectuado los productos anteriores se tiene:

$$u(x)(1 + x^2) = u(x) + u(x)x^2 = \sum_{j=0}^{l-1} u_j x^j + \sum_{j=0}^{l-1} u_j x^{j+2},$$

$$u(x)(1 + x) = u(x) + u(x)x = \sum_{j=0}^{l-1} u_j x^j + \sum_{j=0}^{l-1} u_j x^{j+1},$$

$$u(x)(1 + x + x^2) = u(x) + u(x)x + u(x)x^2 = \sum_{j=0}^{l-1} u_j x^j + \sum_{j=0}^{l-1} u_j x^{j+1} + \sum_{j=0}^{l-1} u_j x^{j+2}.$$

Si en la primera expresión,  $\sum_{j=0}^{l-1} u_j x^j + \sum_{j=0}^{l-1} u_j x^{j+2}$ , hacemos cambios de variable  $j = i$  e  $j = i - 2$  en el primer y segundo sumando, respectivamente, obtenemos:

$$\sum_{i=0}^{l-1} u_i x^i + \sum_{i=2}^{l+1} u_{i-2} x^i.$$

De forma análoga:

$$\sum_{i=0}^{l-1} u_i x^i + \sum_{i=1}^l u_{i-1} x^i, \quad \sum_{i=0}^{l-1} u_i x^i + \sum_{i=1}^l u_{i-1} x^i + \sum_{i=2}^{l+1} u_{i-2} x^i,$$

y desarrollamos la primera expresión:

$$\sum_{i=0}^{l-1} u_i x^i + \sum_{i=2}^{l+1} u_{i-2} x^i = u_0 x^0 + u_1 x^1 + \cdots + u_{l-1} x^{l-1} + u_0 x^2 + u_1 x^3 + \cdots + u_{l-1} x^{l+1},$$

Como agregar ceros anteriores a  $u_0$  y posterior a  $u_{l-1}$  no afecta la información ni la codificación, se puede agregar los términos  $u_l x^l + u_{l+1} x^{l+1}$  y  $u_{-2} x^0 + u_{-1} x^1$  y se obtiene:

$$\begin{aligned} & u_0 x^0 + \cdots + u_l x^l + u_{l+1} x^{l+1} + u_{-2} x^0 + u_{-1} x^1 + u_0 x^2 + \cdots + u_{l-1} x^{l+1} \\ &= (u_{-2} + u_0) x^0 + (u_{-1} + u_1) x^1 + \cdots + (u_{l-2} + u_l) x^l + (u_{l-1} + u_{l+1}) x^{l+1}. \end{aligned}$$

Los términos que se preservan en cada sumando son:

$$u_i + u_{i-2}, \quad i \geq 0.$$

Análogamente con las otras dos sumas:

$$\begin{aligned} & \sum_{i=0}^{l-1} u_i x^i + \sum_{i=1}^l u_{i-1} x^i \\ &= u_0 x^0 + u_1 x^1 + \cdots + u_{l-1} x^{l-1} + u_0 x^1 + u_1 x^2 + u_2 x^3 + \cdots + u_{l-1} x^l \\ &= u_0 x^0 + u_1 x^1 + \cdots + u_{l-1} x^{l-1} + u_l x^l + u_{-1} x^0 + u_2 x^3 + \cdots + u_{l-1} x^l \\ &= (u_{-1}) + u_0 x^0 + (u_0 + u_1) x^1 + \cdots + (u_{l-1} + u_l) x^l + (u_l + u_{l+1}) x^{l+1}. \end{aligned}$$

Los términos que se preservan en cada sumando son:

$$u_i + u_{i-1}, \quad i \geq 0.$$

Se procede de igual manera con  $\sum_{i=0}^{l-1} u_i x^i + \sum_{i=1}^l u_{l-1} x^i + \sum_{i=2}^{l+1} u_{l-2} x^i$ ,

$$u_i + u_{l-1} + u_{l-2}, \quad i \geq 0,$$

son los términos que se preservan.

Podemos obtener la codificación con las expresiones:

$$p_1(i) = u_i + u_{l-2}, \quad i \geq 0,$$

$$p_2(i) = u_i + u_{l-1}, \quad i \geq 0,$$

$$p_3(i) = u_i + u_{l-1} + u_{l-2}, \quad i \geq 0,$$

que con un cambio de notación, esto debido a que en la literatura generalmente se encuentran de esta forma, se obtiene:

$$p_1(i) := u(i) + u(i-2), \quad i \geq 0,$$

$$p_2(i) := u(i) + u(i-1), \quad i \geq 0,$$

$$p_3(i) := u(i) + u(i-1) + u(i-2), \quad i \geq 0.$$

que son las *ecuaciones de paridad* que generan el mismo código que la matriz  $G(x) = [1 + x^2, 1 + x, 1 + x + x^2]$ . Si  $i = 0$ :

$$u(0) + u(-2), \quad u(0) + u(-1), \quad u(0) + u(-1) + u(-2),$$

si  $i = 1$ :

$$u(1) + u(-1), \quad u(1) + u(0), \quad u(1) + u(0) + u(-1).$$

Cada  $u(i)$  es la  $i$ -ésima posición del mensaje a codificar, es decir, si tenemos el mensaje  $u = u_0 u_1 u_2 \cdots u_{l-1}$  lo veremos como  $u = u(0)u(1)u(2) \cdots u(l-1)$  y haremos la respectiva suma que la ecuación nos indica. Nótese que en la sucesión no aparecen los términos  $u(-2), u(-1), u(l)$  y algunos posteriores, en estos casos tomaremos estos bits como 0.

En la figura 2.2 se puede observar que  $u(i)$  esta representado por la primera caja, mientras que  $u(i-1)$  por la segunda y  $u(i-2)$  por la tercera caja. La salida  $v_1$  es la suma de las cajas  $u(i)$  y  $u(i-2)$ , que componen a  $p_1(i)$ , mientras que  $v_2$  es la suma de las cajas  $u(i)$  y  $u(i-1)$  que componen a  $p_2(i)$ , y por último  $v_3$  es la suma de las cajas  $u(i), u(i-1)$  y  $u(i-2)$ , es decir,  $p_3(i)$ .

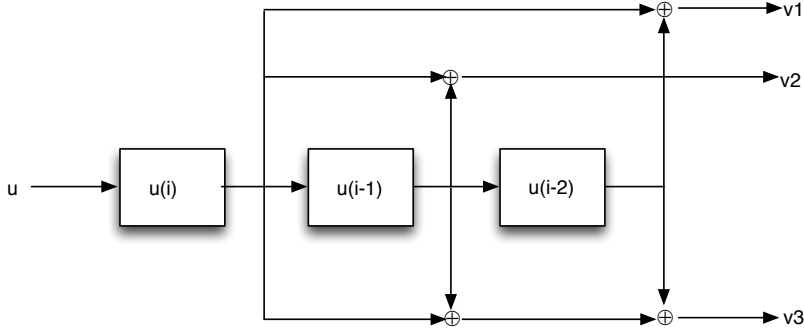


Figura 2.2: Diagrama del sistema asociado

En el ejemplo, las ecuaciones de paridad  $p_1(i) := u(i) + u(i - 2)$ ,  $p_2(i) := u(i) + u(i - 1)$ , y  $p_3(i) := u(i) + u(i - 1) + u(i - 2)$ , para  $i \geq 0$ , corresponden a los polinomios generadores que forman la matriz  $G(x) = [1 + x^2 \ 1 + x \ 1 + x + x^2]$ , es decir,  $1 + x^2$ ,  $1 + x$  y  $1 + x + x^2$ , respectivamente.

Supongamos que el mensaje a codificar es  $u = 1101$ , que lo podemos ver como 00110100 y a su vez como:

$$u(-2)u(-1)u(0)u(1)u(2)u(3)u(4)u(5).$$

Usando las tres ecuaciones anteriores, para el tiempo  $i = 0$  tendremos los bits  $001 = u_{-2}u_{-1}u_0 = u(-2)u(-1)u(0)$ , por lo cual:

$$p_1(0) = u(i) + u(i - 2) = u(0) + u(-2) = 1 + 0 = 1,$$

$$p_2(0) = u(i) + u(i - 1) = u(0) + u(-1) = 1 + 0 = 1,$$

$$p_3(0) = u(i) + u(i - 1) + u(i - 2) = u(0) + u(-1) + u(-2) = 1 + 0 + 0 = 1,$$

para  $p_j(1)$  se usan los bits  $u_{-1}u_0u_1 = 011$ , de tal forma que  $p_1(1) = 1$ ,  $p_2(1) = 0$  y  $p_3(1) = 0$ , de manera análoga se obtienen los otros bits generados por  $p_i(2), p_i(3), p_i(4)$  y  $p_i(5)$ ,  $i = 1, 2, 3$ .

El mensaje  $u = 1101$  codificado con la matriz  $G(x) = [1 + x^2 \ 1 + x \ 1 + x + x^2]$ :

$$u(x)G(x) = (1 + x + x^3)[1 + x^2 \ 1 + x \ 1 + x + x^2] =$$

$$(1 + x + x^2 + x^5 \quad 1 + x^2 + x^3 + x^5 \quad 1 + x^4 + x^5),$$

nótese que  $p_1(0)$ ,  $p_2(0)$  y  $p_3(0)$  generan los bits correspondientes a los coeficientes de  $x^0$ ,

$$(\textcircled{1}+x+x^2+0\cdot x^3+0\cdot x^4+x^5 \quad \textcircled{1}+0\cdot x+x^2+x^3+0\cdot x^4x^5 \quad \textcircled{1}+0\cdot x+0\cdot x^2+0\cdot x^3+x^4+x^5),$$

$p_1(1)$ ,  $p_2(1)$  y  $p_3(1)$  corresponden a los bits de los coeficientes de  $x$ ,

$$(1+\textcircled{1}x+x^2+0\cdot x^3+0\cdot x^4+x^5 \quad 1+\textcircled{0}\cdot x+x^2+x^3+0\cdot x^4x^5 \quad 1+\textcircled{0}\cdot x+0\cdot x^2+0\cdot x^3+x^4+x^5),$$

esto para todos los  $i$ .

En general en una matriz generadora de un código de convolución  $G(x)$ ,  $b \times c$ :

$$G(x) = \begin{pmatrix} g_{11}(x) & g_{12}(x) & \cdots & g_{1c}(x) \\ g_{21}(x) & g_{22}(x) & \cdots & g_{2c}(x) \\ \cdot & \cdot & \cdots & \cdots \\ g_{b1}(x) & g_{b2}(x) & \cdots & g_{bc}(x) \end{pmatrix}$$

supongamos que el grado de los  $g_{ij}(x)$ , para cada  $1 \leq i \leq b$ , es  $m_{ij}$ , respectivamente, y los mensajes a codificar son matrices de tamaño  $\frac{1}{2}$  o  $1 \times b$ ,  $u(x) = (s_1(x), s_2(x), \dots, s_b(x))$ , y haciendo abuso de la notación, para tratar de hacer mas claro los cambios de variables posteriores, los escribiremos como:

$$s_i(x) = \sum_{r=0}^{l_i-1} u_i(r)x^r,$$

y las  $c$  salidas serán de la forma

$$v_j = s_1(x)g_{1j}(x) + s_2(x)g_{2j}(x) + \cdots + s_b(x)g_{bj}(x),$$

$1 \leq j \leq c$ , si sustituimos cada  $s_i(x)$

$$v_j = \sum_{r=0}^{l_1-1} u_1(r)x^r g_{1j}(x) + \sum_{r=0}^{l_2-1} u_2(r)x^r g_{2j}(x) + \cdots + \sum_{r=0}^{l_b-1} u_b(r)x^r g_{bj}(x),$$

el grado de los  $g_{ij}(x)$  es  $m_{ij}$ , respectivamente, cada  $g_{ij}(x)$  tiene varios términos, que se desarrollarán de manera análoga al ejemplo, aquí solo consideraremos el término de grado mayor, por lo cual obtenemos

$$v_j = \sum_{r=0}^{l_1-1} u_1(r)x^r x^{m_{1j}} + \sum_{r=0}^{l_2-1} u_2(r)x^r x^{m_{2j}} + \cdots + \sum_{r=0}^{l_b-1} u_b(r)x^r x^{m_{bj}},$$



$$v_j = \sum_{r=0}^{l_1-1} u_1(r)x^{r+m_{1j}} + \sum_{r=0}^{l_2-1} u_2(r)x^{r+m_{2j}} + \cdots + \sum_{r=0}^{l_b-1} u_b(r)x^{r+m_{bj}},$$

haciendo el respectivo cambio de variable en cada sumando,  $r = i - m_{ij}$

$$v_j = \sum_{i=m_{1j}}^{l_1+m_{1j}-1} u_1(i-m_{1j})x^i + \sum_{i=m_{2j}}^{l_2+m_{2j}-1} u_2(i-m_{2j})x^i + \cdots + \sum_{i=m_{bj}}^{l_b+m_{bj}-1} u_b(i-m_{bj})x^i,$$

usando el mismo argumento que a las ecuaciones del ejemplo anterior, obtenemos:

$$u_1(i - m_{1j}) + u_2(i - m_{2j}) + \cdots + u_b(i - m_{bj}) := p_1(i),$$

y de igual manera son obtenidas las otras  $p_t(i)$ ,  $1 \leq t \leq c$ , en general,

$$p_t(i) = u_1(i-m_{tj}) + u_2(i-m_{tj}) + \cdots + u_b(i-m_{tj}), 1 \leq i \leq b, 1 \leq j \leq c, 1 \leq t \leq b.$$

Podemos concluir que a partir de un polinomio, varios de ellos o una matriz podemos obtener una ecuación de paridad o las respectivas ecuaciones de paridad asociadas.

## 2.6. Polinomios generadores obtenidos de ecuaciones de paridad

Los polinomios generadores están directamente relacionados con las ecuaciones de paridad, son dos expresiones equivalentes. Los coeficientes del polinomio generador están ligados con los coeficientes de la ecuación de paridad correspondiente, por lo cual no dependen del tiempo, ni de la posición, a diferencia de las ecuaciones de paridad.

De un polinomio generador podemos pasar a una ecuación de paridad y viceversa, por lo cual diremos que existe una relación biyectiva

$$p_n(i) \longleftrightarrow g_n(x).$$

Teniendo ecuaciones de paridad obtenemos los polinomios generadores, la obtención de estos prácticamente es un proceso inverso al anterior. Por ejemplo: dada la ecuación de paridad

$$p_1(i) = u(i) + u(i - 1) + u(i - 2),$$

esta solo produce un bit por cada  $i$ , y la suma de estas produce toda la codificación. Si hacemos la suma de  $i = 0$  hasta  $l + 1$ ,

$$\begin{aligned} & \sum_{i=0}^{l+1} x^i (u(i) + u(i-1) + u(i-2)) \\ &= \sum_{i=0}^{l+1} u(i)x^i + \sum_{i=0}^{l+1} u(i-1)x^i + \sum_{i=0}^{l+1} u(i-2)x^i, \end{aligned}$$

algunos de estos términos son cero, por lo cual los podemos prescindir de ellos. Quitamos los primeros ceros de cada suma:

$$= \sum_{i=0}^{l+1} u(i)x^i + \sum_{i=1}^{l+1} u(i-1)x^i + \sum_{i=2}^{l+1} u(i-2)x^i,$$

y si quitamos los últimos ceros de cada suma

$$= \sum_{i=0}^{l-1} u(i)x^i + \sum_{i=1}^l u(i-1)x^i + \sum_{i=2}^{l+1} u(i-2)x^i,$$

haciendo los respectivos cambios de variable  $j = i$ ,  $j = i - 1$  y  $j = i - 2$ :

$$\begin{aligned} &= \sum_{j=0}^{l-1} u(j)x^j + \sum_{j=0}^{l-1} u(j)x^{j+1} + \sum_{j=0}^{l-1} u(j)x^{j+2} \\ &= \sum_{j=0}^{l-1} u(j)x^j + \sum_{j=0}^{l-1} u(j)x^j x + \sum_{j=0}^{l-1} u(j)x^j x^2, \\ &= \sum_{j=0}^{l-1} u(j)x^j (1 + x + x^2), \end{aligned}$$

recordemos que  $u(x) = \sum_{i=0}^{l-1} u(j)x^j$ , por lo cual:

$$= u(x)(1 + x + x^2),$$

al polinomio  $1 + x + x^2$  le llamaremos

$$g_1(x) = 1 + x + x^2,$$

que es el polinomio generador asociado a la ecuación de paridad  $p_1(i) = u(i) + u(i-1) + u(i-2)$ . Nótese que la memoria  $m = 2$  coincide en ambas presentaciones y como  $1 + x + x^2$  esta asociado a  $u(i) + u(i-1) + u(i-2)$ , entonces corresponden término a término, es decir, 1 corresponde a  $u(i)$ ,  $x$  a  $u(i-1)$  y  $x^2$  a  $u(i-2)$ .

En general dada una ecuación de paridad

$$p_n(i) = a_0u(i) + a_1u(i-1) + a_2u(i-2) + a_3u(i-3) + \cdots + a_{k-b}u(i-k+b), \quad a_i \in \mathbb{F}_2$$

su polinomio generador asociado es

$$g_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{k-b}x^{k-b}, \quad a_i \in \mathbb{F}_2,$$

que también tiene asociada la sucesión

$$g_n = (a_0, a_1, a_2, a_3, \cdots, a_{k-b}) \in \mathbb{F}_2^{k-b+1},$$

donde

$$m = \max_{1 \leq n \leq c} \text{grad}(g_n(x)).$$

En general, el polinomio generador surge de ver a la ecuación de paridad como la suma de todos sus términos

$$a_0 \sum_{i=0}^{l-1} u(i)x^i + a_1 \sum_{i=1}^l u(i-1)x^i + a_2 \sum_{i=2}^{l+1} u(i-2)x^i + \cdots + a_{k-b} \sum_{i=k-b}^{l+k-b-1} u(i-k+b)x^i,$$

haciendo los respectivos cambios de variables  $j = i$ ,  $j = i + 1$ ,  $j = i + 2, \cdots, j = k - b$  obtenemos:

$$\begin{aligned} & a_0 \sum_{j=0}^{l-1} u(j)x^j + a_1 \sum_{j=0}^{l-1} u(j)x^{j+1} + a_2 \sum_{j=0}^{l-1} u(j)x^{j+2} + \cdots + a_{k-b} \sum_{i=0}^{l-1} u(j)x^{j+k-b} \\ &= a_0 \sum_{j=0}^{l-1} u(j)x^j + a_1 \sum_{j=0}^{l-1} u(j)x^j x + a_2 \sum_{j=0}^{l-1} u(j)x^j x^2 + \cdots + a_{k-b} \sum_{i=0}^{l-1} u(j)x^j x^{k-b}, \end{aligned}$$

factorizando:

$$\sum_{j=0}^{L-1} u(j)x^j (a_0 + a_1x + a_2x^2 + \cdots + a_{k-b}x^{k-b}),$$

donde el polinomio generador es

$$g_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{k-b}x^{k-b}.$$

Los polinomios generadores  $g_n(x)$  tienen exactamente los mismos coeficientes  $a_i \in \mathbb{F}_2$  que su respectiva ecuación de paridad  $p_n(i)$ .

En resumen, cada polinomio que es entrada de una matriz generadora de un código de convolución, determina una ecuación de paridad; y viceversa, cada ecuación de paridad determina un polinomio que es una entrada de una matriz generadora del código. Así que hablar de polinomios generadores (entradas de la matriz generadora), es lo mismo que hablar de ecuaciones de paridad.

### 2.6.1. Ejemplos

**Ejemplo 1.** Dado el mensaje  $u = 1001 \in \mathbb{F}_2^4$  y el codificador del código de convolución, de tasa  $R = 1/2$ , el cual determina el código de convolución:

$$g_1 = (1, 1, 1) \in \mathbb{F}_2^3,$$

$$g_2 = (1, 1) \in \mathbb{F}_2^2,$$

el codificador lo podemos ver como  $(g_1, g_2)$  y si realizamos la convolución de  $u$  con cada sucesión generadora, obtenemos:

$$u * g_1 = 111111 \in \mathbb{F}_2^6,$$

$$u * g_2 = 11011 \in \mathbb{F}_2^5,$$

tal que  $u * g_1 = v_1$  y  $u * g_2 = v_2$ ,

$$(v_1, v_2) = (111111, 11011),$$

que es la codificación de la palabra  $u$ .

Las sucesiones son la presentación más natural de codificadores y mensajes a codificar.

**Ejemplo 2.** A toda sucesión le podemos asociar un polinomio y viceversa. Por lo tanto, sea el mensaje que deseamos codificar,

$$u(x) = 1 + x^3,$$

y el codificador de tasa  $R = 1/2$  determinado por el par de polinomios generadores

$$\begin{aligned}g_1(x) &= 1 + x + x^2, \\g_2(x) &= 1 + x,\end{aligned}$$

realizamos el producto usual de polinomios

$$\begin{aligned}u(x)g_1(x) &= v_1(x), \\u(x)g_2(x) &= v_2(x),\end{aligned}$$

que se puede ver como:

$$\begin{aligned}u(x)(g_1(x), g_2(x)) &= (u(x)g_1(x), u(x)g_2(x)) \\&= (1 + x + x^2 + x^3 + x^4 + x^5, 1 + x + x^3 + x^4) \\&= (v_1(x), v_2(x)) \longleftrightarrow (111111, 11011),\end{aligned}$$

en caso de polinomios

$$\begin{aligned}\cdot : \mathbb{F}_2[x] \times \mathbb{F}_2[x] &\longrightarrow \mathbb{F}_2[x] \\u(x)g_n(x) &\mapsto v_n(x).\end{aligned}$$

**Ejemplo 3.** Usando la ecuación  $\sum_{i=0}^j u_i g_j^{(j)}$ , que es la ecuación de convolución discreta, tal que  $u_1 = 1001$ ,  $g_1 = g_0^{(1)}$ ,  $g_1^{(1)} g_2^{(1)} = 111$ , y  $g_2 = g_0^{(2)}$ ,  $g_1^{(2)} = 11$ , obtenemos

$$\begin{aligned}v_0^{(1)} &:= 1 = u_0 g_0^{(1)} = 1 \cdot 1, \\v_1^{(1)} &:= 1 = u_0 g_1^{(1)} + u_1 g_0^{(1)} = 1 \cdot 1 + 0 \cdot 1, \\v_2^{(1)} &:= 1 = u_0 g_2^{(1)} + u_1 g_1^{(1)} + u_2 g_0^{(1)} = 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 1, \\v_3^{(1)} &:= 1 = u_0 g_3^{(1)} + u_1 g_2^{(1)} + u_2 g_1^{(1)} + u_3 g_0^{(1)} = 1 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 1 \cdot 1, \\v_4^{(1)} &:= 1 = u_0 g_4^{(1)} + u_1 g_3^{(1)} + u_2 g_2^{(1)} + u_3 g_1^{(1)} + u_4 g_0^{(1)} = 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 + 0 \cdot 1, \\v_5^{(1)} &:= 1 = u_0 g_5^{(1)} + u_1 g_4^{(1)} + u_2 g_3^{(1)} + u_3 g_2^{(1)} + u_4 g_1^{(1)} + u_5 g_0^{(1)} = 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 1,\end{aligned}$$

cuando aparecen  $u_i, g_j^{(1)}$  tal que  $i, j > \text{grad}(u(x)), \text{grad}(g_1(x))$ , entonces  $u_i, g_j^{(1)} = 0$ .

Análogamente para  $g_2$  y así definimos

$$(v_1, v_2) := (v_0^{(1)} v_1^{(1)} v_2^{(1)} v_3^{(1)} v_4^{(1)} v_5^{(1)}, v_0^{(2)} v_1^{(2)} v_2^{(2)} v_3^{(2)} v_4^{(2)}) = (111111, 11011),$$

es decir,  $v_1 = v_0^{(1)}v_1^{(1)}v_2^{(1)}v_3^{(1)}v_4^{(1)}v_5^{(1)}$  y  $v_2 = v_0^{(2)}v_1^{(2)}v_2^{(2)}v_3^{(2)}v_4^{(2)}$ .

La longitud del mensaje codificado es diferente en ambas entradas, es importante establecer la longitud del mensaje. En general tomaremos

$$L = \max\{l(v_1), \dots, l(v_n)\},$$

y “completamos” con los ceros a la derecha de cada mensaje hasta igualar a  $L$ . En nuestro ejemplo asociaremos

$$l(111111) = 6 = l(110110),$$

$$l(v_1) = l(v_2).$$

La ecuación de convolución discreta genera término a término los coeficientes del polinomio  $v(x)$  y de igual manera las coordenadas de la convolución de sucesiones. Véanse los 3 ejemplos para comparar.

Para algunas aplicaciones de los códigos de convolución se usa lo que llamamos tiempo o momento y denotaremos con  $i$ .  $i$  es un índice entero que indica que bits o bit es el que se está codificando, podemos considerar a  $i$  positivo o negativo según en el tiempo que sea requerida iniciar la codificación. En algunas situaciones se hace necesario trabajar con subíndices negativos o agregar algunos para “completar” información. En las sucesiones identificamos el índice  $i$  con los subíndices de  $u$  y  $v$ , por ejemplo si iniciamos en el tiempo  $i = 0$  tenemos la sucesiones:

$$u = u_0u_1 \cdots u_{l-1}, \quad u \in \mathbb{F}_2^l,$$

$$v = v_0v_1 \cdots v_{l+m-1}, \quad v \in \mathbb{F}_2^{l+m}.$$

Un ejemplo es: si dada la tasa  $R = 2/3$  y queremos iniciar la codificación en el tiempo  $-2$ , ( $i = -2$ ), entonces tendremos

$$u = u_{-2}u_{-1}u_0u_1u_2 \cdots,$$

con  $u_i = u_i^{(1)}u_i^{(2)}$  para  $i \geq -2$ ,  $u_i \in \mathbb{F}_2^2$ , (hay dos coordenadas de entrada debido a que la tasa es  $R = 2/3$  y habrán 3 de salida).

La tasa  $R = 2/3$  genera la sucesión

$$v = v_{-2}v_{-1}v_0v_1 \cdots,$$

$v_j = v_j^{(1)}v_j^{(2)}v_j^{(3)}$ ,  $v_j \in \mathbb{F}_2^3$  para  $j \geq -2$ . Generalmente para la implementación se inician en un tiempo  $i = 0$ , aunque no se descartan tiempos anteriores.

Ahora, un ejemplo iniciando con subíndice cero. Dado el mensaje de entrada  $u = u_0u_1u_2 \dots$  y una tasa  $R = 1/2$ , el codificador genera dos sucesiones de salida

$$v^{(1)} = v_0^{(1)}v_1^{(1)}v_2^{(1)} \dots$$

y

$$v^{(2)} = v_0^{(2)}v_1^{(2)}v_2^{(2)} \dots$$

En general, cada una de estas  $c$  sucesiones de salida  $v$  tiene longitud  $l + m$ , estas  $c$  salidas se “conjuntan” y se tiene la longitud  $c(l + m)$  de la palabra codificada:

$$v \in \mathbb{F}_2^{c(l+m)},$$

es decir, el código de convolución  $C$  es tal que,

$$C \subset \mathbb{F}_2^{c(l+m)}.$$

## Capítulo 3

# Matrices generadoras y forma normal de Smith

Como ya se mencionó antes, las matrices generadoras de los códigos de convolución pueden tener entradas polinomiales, es decir, con elementos en el anillo  $\mathbb{F}_2[x]$ , o racionales, con entradas en el campo  $\mathbb{F}_2(x)$ . En caso de tener una matriz generadora con entradas racionales, a través de operaciones elementales se tiene una matriz equivalente con entradas polinomiales y estas matrices generan códigos equivalentes.

Por otro lado, realizar la codificación del mensaje con una matriz con entradas polinomiales resulta más costoso que hacerlo con una matriz diagonal. En este capítulo se verá que la forma Normal de Smith permite hacer una simplificación en este sentido.

### 3.1. Matrices generadoras

Como ya se mencionó, una *matriz generadora*  $G(x)$  de un código de convolución  $C$  con parámetros  $(c, b, m)$ , es una matriz de tamaño  $b \times c$ ,  $b \leq c$ , con entradas en el campo de las funciones racionales, es decir,  $G(x) \in \text{Mat}_{b \times c}(\mathbb{F}_2(x))$ . Al multiplicar el mensaje de entrada por la matriz generadora obtenemos la codificación de dicho mensaje:

$$u(x)G(x) = v(x),$$

$u(x) \in \mathbb{F}_2(x)^b$  y  $v(x) \in \mathbb{F}_2(x)^c$ . En general se desea que los renglones de la matriz  $G(x)$  sean linealmente independientes, para que de esta forma sean



una base del código  $C$  sobre el campo de funciones racionales  $\mathbb{F}_2(x)$ , es decir,

$$C = \text{Im}(G(x)),$$

y  $G(x)$  tendrá rango  $b$  sobre  $\mathbb{F}_2(x)$ .

En la aplicación regularmente se usan matrices generadoras  $G(x)$  con entradas en el anillo de polinomios  $\mathbb{F}_2[x]$ , en ocasiones surgen o son convenientes las matrices generadoras en el campo de las funciones racionales binarias, y como  $\mathbb{F}_2[x] \subset \mathbb{F}_2(x)$  podemos generalizar las matrices generadoras al campo  $\mathbb{F}_2(x)$ .

De igual forma que en los códigos de bloque (clásicos), una matriz  $G(x)$  de un código de convolución es equivalente a otra matriz  $G'(x)$  si podemos llevar  $G(x)$  a  $G'(x)$  mediante combinación de las siguientes operaciones (elementales):

- i) Multiplicar un renglón (columna) de  $G(x)$  por cualquier elemento diferente de cero de  $\mathbb{F}_2(x)$
- ii) Intercambio de dos renglones (columnas)
- iii) Suma de renglones (columnas).

Una matriz equivalente a  $G(x)$  también es una matriz generadora, en particular si en la matriz  $G(x)$  se multiplica un renglón por cualquier elemento diferente de cero en  $\mathbb{F}_2(x)$ , también es una matriz generadora para un código y se dirá que éstas generan códigos de convolución equivalentes. Por ejemplo, dada la matriz generadora

$$G(x) = \begin{pmatrix} 1+x & 1+x & x+x^2 \\ x & 1+x^2 & 0 \end{pmatrix}$$

y  $(1+x) \in \mathbb{F}_2(x)$ , entonces,

$$\begin{aligned} (1+x)G(x) &= (1+x) \begin{pmatrix} 1+x & 1+x & x+x^2 \\ x & 1+x^2 & 0 \end{pmatrix} = \\ & \begin{pmatrix} 1+x^2 & 1+x^2 & x+x^3 \\ x+x^2 & 1+x+x^2+x^3 & 0 \end{pmatrix} = G'(x), \end{aligned}$$

es decir, todo múltiplo de una matriz generadora  $G(x)$  también es una matriz generadora  $G'(x)$  de un código equivalente. Por tal motivo siempre podemos llevar una matriz generadora con entradas en el campo de las funciones racionales sobre  $\mathbb{F}_2$  a una matriz con entradas polinomiales sobre el campo  $\mathbb{F}_2$ , esto lo hacemos multiplicando la matriz  $G(x)$  por el mínimo común múltiplo de los denominadores, que también es un elemento de  $\mathbb{F}_2[x]$ , obteniendo así  $G'(x)$ ,  $G(x) \in Mat(\mathbb{F}_2(x))$  y  $G'(x) \in Mat(\mathbb{F}_2[x])$ .

Cuando las entradas de una matriz generadora son polinomiales la llamaremos matriz generadora polinomial o matriz polinomial y cuando son racionales es una matriz generadora racional o simplemente matriz racional. Generalmente evitaremos las matrices con entradas racionales, debido a que para la codificación resultan más “complejas” que las matrices polinomiales, agregando que  $\mathbb{F}_2[x]$  ofrece la generosidad de ser un anillo euclidiano, y esto nos permite aplicar el algoritmo de la división a las entradas de la matriz generadora polinomial, es decir, aplicar la forma normal de Smith.

Un ejemplo de una matriz con entradas en  $\mathbb{F}_2(x)$  es

$$G(x) = \begin{pmatrix} \frac{1}{1+x} & \frac{1+x}{1+x+x^2} & \frac{x^2}{x+x^2} \\ \frac{x}{1+x} & \frac{1+x+x^2}{1+x^2} & \frac{x^3}{1+x^2} \end{pmatrix}$$

el mínimo común múltiplo de los denominadores  $1+x$ ,  $1+x+x^2$ ,  $x+x^2$  y  $1+x^2$  es

$$x+x^2+x^4+x^5 = (1+x)(1+x)x(1+x+x^2),$$

por lo cual una matriz equivalente a  $G(x)$  es  $G'(x)$ , tal que

$$\begin{aligned} & ((1+x)(1+x)x(1+x+x^2)) \cdot G(x) = \\ G'(x) &= \begin{pmatrix} x+x^4 & x+x^2+x^3+x^4 & x^3+x^4+x^5 \\ x^2+x^3+x^4+x^5 & x+x^3+x^5 & x^4+x^5+x^6 \end{pmatrix}, \end{aligned}$$

es decir, generan códigos equivalentes, y así obtenemos una matriz cuyas entradas están en  $\mathbb{F}_2[x]$ . Diremos que como el grado mayor de los polinomios de la matriz es 6, entonces la matriz generadora  $G(x)$  tiene memoria  $m=6$ , la memoria es el máximo grado de las entradas de la matriz.

Todo código de convolución tiene asociada una matriz polinomial y una matriz racional, ya que podemos llevar una matriz generadora polinomial a una

matriz equivalente generadora racional (más adelante daremos un ejemplo de esto) y viceversa. De acuerdo al uso que se le quiera dar elegiremos esta, no podemos decir que una sea mejor que la otra, ya que depende para que sea requerida. En nuestro ejemplo la matriz racional asociada al código es:

$$\frac{1}{x + x^2 + x^4 + x^5} \begin{pmatrix} x + x^4 & x + x^2 + x^3 + x^4 & x^3 + x^4 + x^5 \\ x^2 + x^3 + x^4 + x^5 & x + x^3 + x^5 & x^4 + x^5 + x^6 \end{pmatrix} = G(x).$$

En la figura 3.1 se muestra el diagrama asociado a la matriz  $G'(x)$ . Ge-

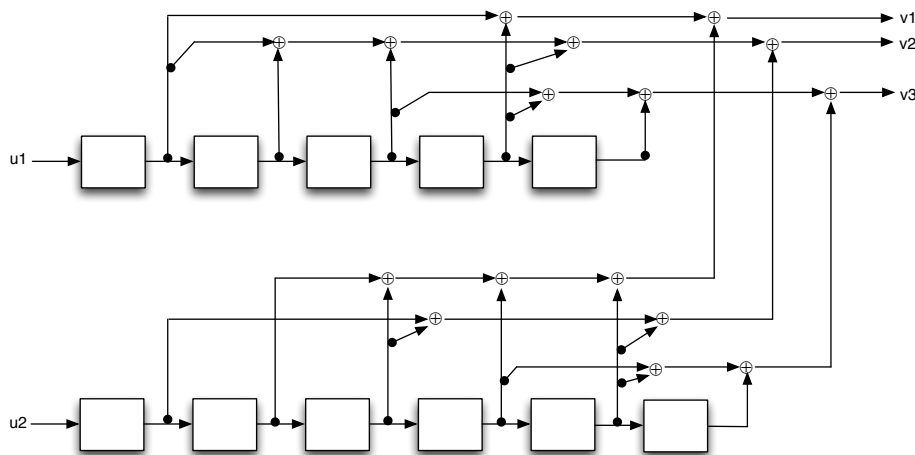


Figura 3.1: Diagrama asociado a la matriz  $G'(x)$  de tasa  $R = 2/3$ .

neralmente podremos asociar un diagrama de este tipo a cualquier matriz generadora, este diagrama ilustra que por cada dos entradas  $u_1$  y  $u_2$  (que corresponden a  $u_1$  y  $u_2$ ) que entran al codificador, este genera tres  $v_1$ ,  $v_2$  y  $v_3$  (corresponden a  $v_1$ ,  $v_2$  y  $v_3$ ), es decir, la tasa asociada es  $R = 2/3$ .

Nótese que existen dos renglones de “cajas”, el primero con 5 y el segundo con 6. La entrada  $u_1$  está ligado directamente al primer renglón de la matriz, mientras que  $u_2$  al segundo;  $v_1$ ,  $v_2$  y  $v_3$  las salidas del codificador están ligadas tanto al primer renglón como al segundo, podemos decir que son una combinación de ambos renglones. El grado máximo de primer renglón de la matriz  $G'(x)$  es 5 y del segundo es 6, y cada una de las “cajas” del diagrama representan un  $x^i$ ,  $i \geq 1$ , la “caja” más próxima al mensaje de entrada  $u_1$

representa  $x^1$ , la siguiente “caja”  $x^2$  y así hasta la última “caja” representa a  $x^5$ , esto pasa de la misma manera con el siguiente bloque, hay 6 cajas ya que el máximo grado es 6.

Veamos que en la figura la salida  $v_1$  se compone de dos partes, una que se forma por elementos del primer renglón y la otra por elementos del segundo. Del primer renglón tenemos la suma de la primera caja con la cuarta, es decir  $x + x^4$ , mientras que del segundo renglón tenemos la suma de la segunda, tercera, cuarta y quinta caja, esto es:  $x^2 + x^3 + x^4 + x^5$ . Estos dos polinomios conforman la salida  $v_1$ , las salidas  $v_2$  y  $v_3$  tienen una explicación análoga.

En general, un diagrama asociado a una matriz tendrá tantos renglones de cajas como renglones la matriz, es decir  $c$ , y cada renglón del diagrama tendrá tantas cajas como el mayor de los grados de los polinomios del renglón, la primera caja representa  $x$ , la segunda  $x^2$  y así sucesivamente (en cada renglón). Cada polinomio del renglón 1 de la matriz puede escribirse sumando las cajas correspondientes, igualmente para cada renglón. Mientras que las salidas están en términos de todos los renglones de cajas, ya que las salidas suman columnas, es decir, la salida  $v_1$  estará dada por la suma de las columnas correspondientes al primer polinomio de la columna 1, al primer del polinomio 2 y así sucesivamente, igual para las columnas  $v_i$ . Cabe señalar que en los diagramas al aparecer y referirnos a  $v_i$  e  $u_i$  nos referimos a  $v_i$  e  $u_i$ , respectivamente.

A cada matriz generadora le podemos asociar un diagrama y de cada diagrama podemos leer una matriz generadora. La tasa de una matriz generadora,  $R = b/c$ , esta ligada directamente al tamaño de la matriz, es decir,  $b \times c$ .

Si

$$G(x) = \begin{pmatrix} g_{11}(x) & g_{12}(x) & \cdots & g_{1c}(x) \\ g_{21}(x) & g_{22}(x) & \cdots & g_{2c}(x) \\ \cdot & \cdot & \cdots & \cdots \\ g_{b1}(x) & g_{b2}(x) & \cdots & g_{bc}(x) \end{pmatrix}$$

$$\begin{aligned} \max\{\text{grad}(g_{ij}(x))\} &= m, \\ \text{grad}(u(x)G(x)) &= \max\{\text{grad}(g_{ij}(x)) + \text{grad}(u_i(x))\} = \text{grad}(v_i(x)), \\ &1 \leq i \leq b, 1 \leq j \leq c. \end{aligned}$$

$g_{ij}(x)$  son las entradas de la matriz  $G(x)$  y  $u_i(x)$  es cada una de las coordenadas de  $u(x) = (u_1(x), \dots, u_b(x))$ .

La matriz generadora para un código de convolución determina la codificación, por lo cual la elección de esta es muy importante, y cabe mencionar que existen matrices más convenientes que otras.

Un ejemplo sencillo de codificación usando matriz generadora es: dada la matriz  $G(x)$ ,

$$G(x) = [1 + x^2 + x^3 \quad 1 + x],$$

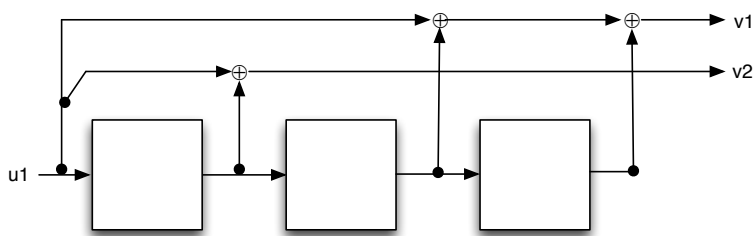


Figura 3.2: Diagrama asociado a la matriz  $G(x) = [1 + x^2 + x^3 \quad 1 + x]$  de tasa  $R = 1/2$ . Este diagrama tiene 3 cajas en un solo renglón, debido que solo hay una entrada  $u1$  y el mayor grado que aparece en la matriz es tres.

la memoria de  $G(x)$  es  $m = 3$ , y el mensaje a codificar es  $u = 11011$ , pero para codificarlo tenemos que llevarlo a su expresión polinomial, es decir,  $u(x) = (1 + x + x^3 + x^4)$ ,

$$\begin{aligned} u(x)G(x) &= (1 + x + x^3 + x^4)[1 + x^2 + x^3 \quad 1 + x] \\ &= (1 + x + x^2 + x^3 + x^5 + x^7 \quad 1 + x^2 + x^3 + x^5) = v(x), \end{aligned}$$

cuyo elemento sobre el campo binario es:

$$v = (11110101 \quad 101101),$$

$$v \in \mathbb{F}_2^{13}.$$

Las entradas del arreglo  $v$  tienen longitud 8 y 5, respectivamente. Tomaremos la entrada con mayor longitud obtenida y homogeneizaremos la otra entrada, completando el otro bloque con tantos ceros a la derecha como sean necesarios para tener bloques de la misma longitud, generalmente se procede de esta manera para hacer el proceso de decodificación. Con lo que obtenemos:

$$v = (11110101 \quad 11101000) \in \mathbb{F}_2^{16},$$

que lo veremos como

$$v = 11 \ 10 \ 11 \ 11 \ 00 \ 11 \ 00 \ 10.$$

Como se explicó al inicio de este trabajo, esta concatenación es común encontrarla en la literatura.

## 3.2. Ejemplo de forma normal de Smith

Las matrices generadoras de los códigos de convolución pueden tener entradas polinomiales,  $\mathbb{F}_2[x]$ , o racionales,  $\mathbb{F}_2(x)$ . Ya mencionamos que en caso de tener una matriz generadora con entradas racionales, a través de operaciones elementales podemos convertirla en una matriz equivalente con entradas polinomiales y estas matrices generan códigos equivalentes.

Realizar la multiplicación del mensaje por una matriz con todas las entradas polinomiales resulta más complicado que multiplicar por una matriz diagonal, y esto se traduce en un costo operacional mayor. La forma normal de Smith ayuda a hacer una simplificación.

En el capítulo 1 se definió el proceso para llevar una matriz a la forma normal de Smith, veamos un ejemplo al cual le incluimos en cada paso el diagrama asociado a la matriz.

### Ejemplo: Forma normal de Smith.

Sea la matriz generadora  $G(x)$ ,  $2 \times 3$  que corresponde a un código de tasa  $R = 2/3$ :

$$G(x) = \begin{pmatrix} x & x^2 & 1 \\ 1+x & 1+x^3 & 1+x+x^2 \end{pmatrix},$$

necesitamos dos matrices identidad, de tamaño  $b$  y  $c$ , es decir,  $I_2$  e  $I_3$  a la

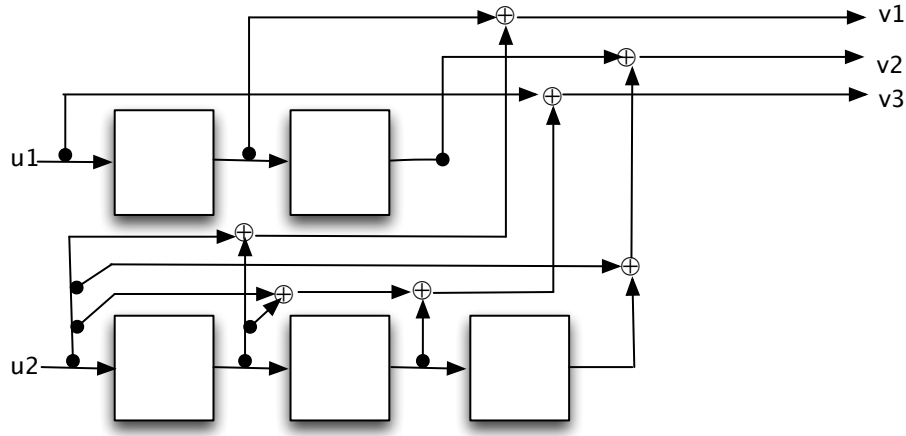


Figura 3.3: Diagrama asociado a la matriz  $G(x)$ , con tasa  $R = 2/3$ .

izquierda y derecha de  $G$ , respectivamente

$$G(x) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x & x^2 & 1 \\ 1+x & 1+x^3 & 1+x+x^2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

La entrada más pequeña de la matriz  $G(x)$  es 1, así que llevamos este elemento a la parte superior izquierda mediante el intercambio de columnas

$$\begin{pmatrix} x & x^2 & 1 \\ 1+x & 1+x^3 & 1+x+x^2 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & x^2 & x \\ 1+x+x^2 & 1+x^3 & 1+x \end{pmatrix}$$

para hacer cero a los dos términos a la derecha del 1 multiplicamos por la matriz:

$$\begin{pmatrix} 1 & x^2 & x \\ 1+x+x^2 & 1+x^3 & 1+x \end{pmatrix} \begin{pmatrix} 1 & x^2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x \\ 1+x+x^2 & 1+x^2+x^4 & 1+x \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & x \\ 1+x+x^2 & 1+x^2+x^4 & 1+x \end{pmatrix} \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

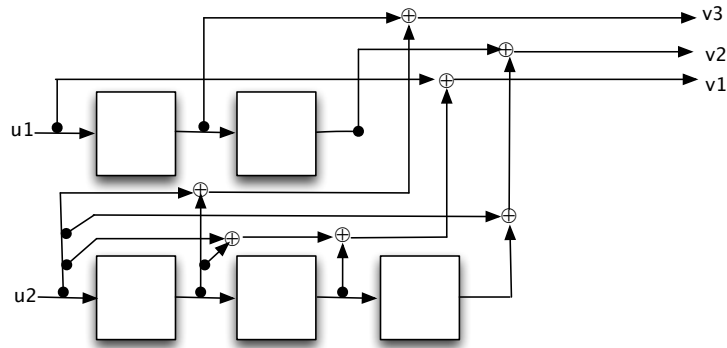


Figura 3.4:

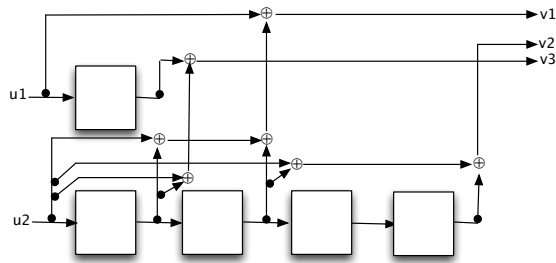


Figura 3.5:

$$= \begin{pmatrix} 1 & 0 & 0 \\ 1+x+x^2 & 1+x^2+x^4 & 1+x^2+x^3 \end{pmatrix}$$

Ahora nos interesa hacer cero el primer término del segundo renglón, esto lo podemos ver como una operación sobre las columnas

$$\begin{pmatrix} 1 & 0 \\ 1+x+x^2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1+x+x^2 & 1+x^2+x^4 & 1+x^2+x^3 \end{pmatrix} \\ = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1+x^2+x^4 & 1+x^2+x^3 \end{pmatrix}$$

por el algoritmo de la división tenemos que  $1+x^2+x^4 = (1+x^2+x^3)(1+x)+x$ ,



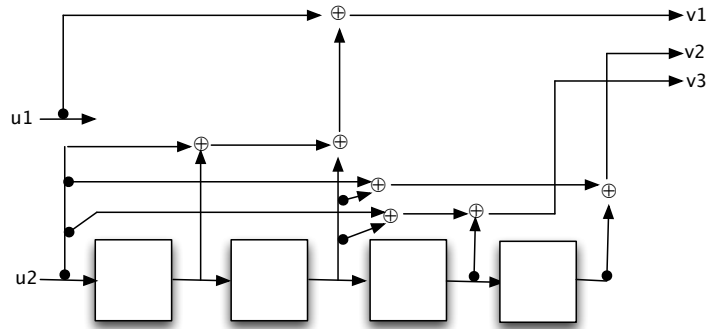


Figura 3.6:

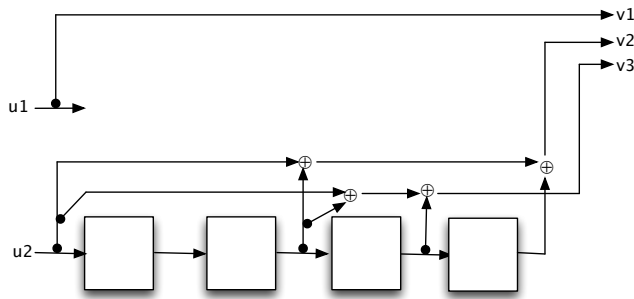


Figura 3.7:

notemos que  $\text{grad}(x) < \text{grad}(1 + x^2 + x^3)$ , por lo cual multiplicaremos la tercera columna por  $(1 + x)$  y sumamos a la segunda columna, para así obtener

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 + x^2 + x^4 & 1 + x^2 + x^3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 + x & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & x & 1 + x^2 + x^3 \end{pmatrix}$$

nuevamente por el algoritmo de la división  $1 + x^2 + x^3 = x(x + x^2) + 1$ , donde  $\text{grad}(1) < \text{grad}(x)$ , por lo cual multiplicamos a la segunda columna

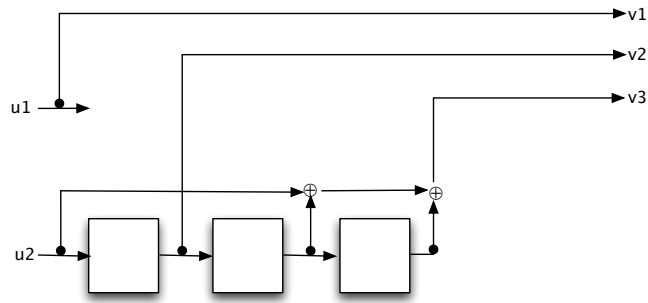


Figura 3.8:

por  $(x + x^2)$  y la sumamos a la columna 3,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & x & 1 + x^2 + x^3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & x + x^2 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & x & 1 \end{pmatrix}$$

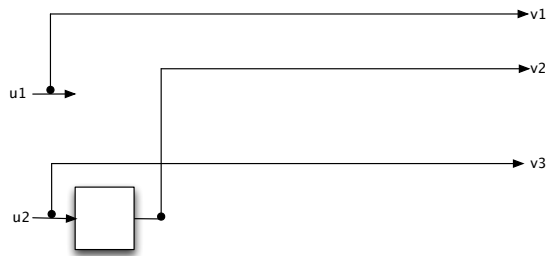


Figura 3.9:

como  $x = (x)1 + 0$  multiplicamos a la columna 3 por  $x$  y la sumamos a la columna dos, para obtener:

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

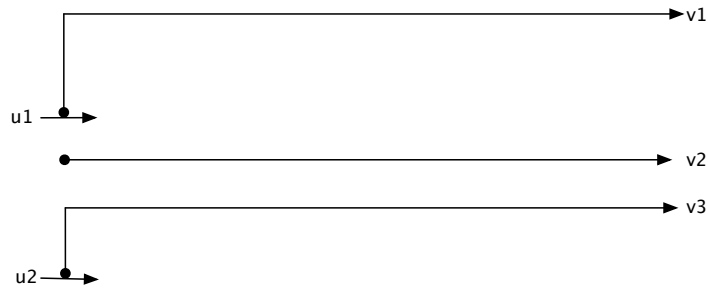


Figura 3.10:

finalmente intercambiamos la columna dos por la columna tres, y esto produce una matriz diagonal  $\Gamma$ ,

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

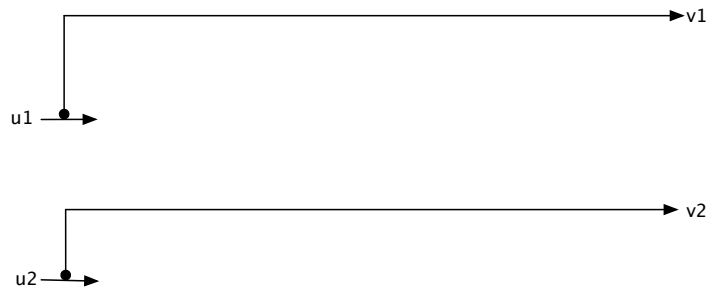


Figura 3.11: La salida 3 es la constante 0.

y la matriz  $G$  es el producto de las matrices obtenidas, en el mismo orden:

$$\begin{pmatrix} 1 & 0 \\ 1+x+x^2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & x & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & x+x^2 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1+x & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & x^2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

es decir,

$$\begin{pmatrix} 1 & 0 \\ 1+x+x^2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x & x^2 & 1 \\ 1+x^2+x^3 & 1+x^2+x^4 & 0 \\ x+x^2 & 1+x+x^3 & 0 \end{pmatrix},$$

por lo cual

$$G(x) = A(x)\Gamma(x)B(x),$$

tal que

$$u(x) \cdot G(x) = u(x)(A(x)\Gamma(x)B(x)).$$

$A(x)$  y  $B(x)$  son matrices invertibles ya que inicialmente  $I_b$  e  $I_c$  eran matrices invertibles con determinante 1, y solo se hicieron operaciones básicas sobre sus renglones y columnas, adicionalmente  $A(x)$  y  $B(x)$  tienen determinante  $x + x^2$  y 1 respectivamente.

Una codificación equivalente a  $u(x) \cdot G(x)$  es:

$$u(x) \cdot \Gamma(x).$$

En los diagramas puede verse gráficamente como el proceso de la diagonalización simplifica de gran manera el proceso de codificación, cada diagrama representa una matriz, que en todos los casos son equivalentes entre ellas. El costo de la multiplicación por esta nueva matriz se reduce de gran manera, es más sencillo multiplicar por esta nueva matriz diagonal, equivalente a la primera.

### 3.3. Matrices catastróficas

En general, al elegir una matriz como codificador para un código de convolución hay una clase de matrices generadoras que deben ser evitadas, las llamadas matrices catastróficas. En esta sección mostraremos de una forma breve el problema que surge al usarlas.

Si  $u$  es una sucesión binaria diremos que el peso  $w$  es el número de coordenadas diferentes de 0, es decir, el número de 1's en la sucesión, mientras que si  $u(x)$  es un polinomio, serie de potencias o serie de Laurent sobre  $\mathbb{F}_2$ , el peso  $w$  de  $u(x)$  son los coeficientes diferentes de 0.

Las matrices catastróficas son aquellas matrices que a través de una entrada de información de peso infinito generan una salida codificada de peso finito.

Es preciso que recordemos el concepto de matrices invertibles. Una matriz  $G(x)$  de tamaño  $b \times b$  es invertible si existe una matriz  $H(x)$ ,  $b \times b$ , tal que:

$$G(x)H(x) = I_b(x) = H(x)G(x).$$

En las matrices generadoras para códigos de convolución, debido a que no son matrices necesariamente cuadradas,  $b \times c$ ,  $b \leq c$ , no siempre tienen una matriz inversa, pero para toda matriz  $G(x)$ ,  $b \times c$  con  $b \leq c$ , existe una inversa derecha.

Una inversa derecha de  $G(x)$  es una matriz de tamaño  $c \times b$  que denotaremos con  $G'(x)$ , tal que

$$G(x)G'(x) = I_b.$$

En general,  $G(x)$  tiene inversa derecha si y sólo si tiene rango  $b$ . Si  $b < c$  y  $G(x) \in Mat(\mathbb{F}_2[x])$  hay varias posibles inversas derechas, es decir, la inversa derecha  $G'(x)$  no es única.

En general el rango de una matriz es el número de renglones linealmente independientes, en este caso como las entradas de las matrices son polinomios, con independencia lineal entenderemos que un renglón no sea producto de otro de ellos.

Si  $G(x)$  tiene rango  $b$ , entonces tiene una inversa derecha  $G'(x)$ . Por otro lado, toda sucesión  $u(x)$  es una matriz  $1 \times b$ , tal que:

$$u(x) = u(x) \cdot I_b = u(x)G(x)G'(x) = v(x)G'(x).$$

**Ejemplo.** Sea la matriz  $G(x)$ ,  $2 \times 3$  de rango 2, con entradas en  $\mathbb{F}_2[x]$ ,

$$G(x) = \begin{pmatrix} x & x^2 & 1 \\ x + x^2 & 1 & 0 \end{pmatrix},$$

una matriz que es inversa por la derecha de  $G(x)$  es

$$G'(x) = \begin{pmatrix} 1 & x \\ x + x^2 & 1 + x^2 + x^3 \\ 1 + x + x^3 + x^4 & x^4 + x^5 \end{pmatrix},$$

dado que

$$\begin{aligned} G(x)G'(x) &= \begin{pmatrix} x & x^2 & 1 \\ x + x^2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & x \\ x + x^2 & 1 + x^2 + x^3 \\ 1 + x + x^3 + x^4 & x^4 + x^5 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &= I_2, \end{aligned}$$

esta  $G'(x)$  no es la única matriz inversa derecha de  $G(x)$ , por ejemplo, otra matriz inversa derecha de  $G(x)$  es:

$$\begin{pmatrix} 1 & 1 \\ x + x^2 & 1 + x + x^2 \\ 1 + x + x^3 + x^4 & x + x^2 + x^3 + x^4 \end{pmatrix},$$

si  $u(x) = (1 \ x)$ , entonces

$$u(x)G(x) = (1 \ x) \begin{pmatrix} x & x^2 & 1 \\ x + x^2 & 1 & 0 \end{pmatrix} = (x + x^2 + x^3 \ x + x^3 \ 1) = v(x)$$

y

$$\begin{aligned} v(x)G'(x) &= (x + x^2 + x^3 \ x + x^3 \ 1) \begin{pmatrix} 1 & x \\ x + x^2 & 1 + x^2 + x^3 \\ 1 + x + x^3 + x^4 & x^4 + x^5 \end{pmatrix} \\ &= (1 \ x) = u(x). \end{aligned}$$

Como ya hemos mencionado, las matrices generadoras de un código de convolución las podemos tomar en el campo de las funciones racionales,  $\mathbb{F}_2(x)$ , aunque generalmente son tomadas en el anillo polinomial  $\mathbb{F}_2[x]$ .

Un caso particular son las funciones racionales, ya que si  $f(x) \in \mathbb{F}_2(x)$  sabemos que es un cociente  $p(x)/q(x)$ , donde  $p(x)$  y  $q(x) \in \mathbb{F}_2[x]$ ,  $f(x) = p(x)/q(x)$  tiene una expansión en serie de Laurent

$$f(x) = \sum_{i=s}^{\infty} f_i x^i, \quad s \in \mathbb{Z}, \quad f_i \in \mathbb{F}_2,$$

que son una serie de potencias,  $s = 0$  si  $q(0) = 0$ , el peso de  $f(x)$  es el número de coeficientes diferentes de 0 de la expansión de la serie de Laurent y este peso puede ser infinito.

Un ejemplo concreto es:

$$w\left(\frac{1+x}{1+x^3}\right) = w(1+x+x^3+x^4+x^6+x^7+\dots) = \infty.$$

Una matriz  $G(x)$  generadora de un código de convolución  $(c, b, m)$ , con entradas en  $\mathbb{F}_2(x)$  diremos que es catastrófica si existe un mensaje  $u(x)$  de peso infinito, tal que  $v(x) = u(x)G(x)$  tiene peso finito, en otro caso es llamada no catastrófica.

**Ejemplo.** Sea la matriz

$$G(x) = [1+x+x^2, \quad 1+x^2]$$

una inversa derecha de  $G(x)$  es

$$G'(x) = \begin{pmatrix} \frac{x}{1+x} \\ 1 \end{pmatrix},$$

y sea  $u'(x) = (1+x+x^2 \quad 1+x^2)$  cuya sucesión asociada es (111, 101), la cual tiene peso  $w(u) = 5$ , por otro lado:

$$u'(x)G'(x) = (1+x+x^2 \quad 1+x^2) \begin{pmatrix} \frac{x}{1+x} \\ 1 \end{pmatrix} = (1+x+x^2) \frac{x}{1+x} + (1+x^2)(1) =$$

$$\frac{1}{1+x} = \sum_{i=0}^{\infty} x^i = u(x),$$

cuya sucesión es 11111..., tiene peso infinito, es decir  $u(x)$  tiene peso finito, mientras que  $u'(x)$  tiene peso infinito, por lo cual  $G(x)$  es una matriz catastrófica.

No es sencillo comprobar la existencia del mensaje  $u(x)$  que indica si la matriz generadora polinomial es catastrófica, afortunadamente, tenemos un teorema

de Massey y Sain que permite decidir si una matriz generadora polinomial es catastrófica. Este resultado proporciona dos propiedades equivalentes para una matriz generadora  $G(x)$  de un código de convolución.

**Teorema (Massey-Sain)** [11]. Sea  $G(x)$  una matriz polinomial generadora para un código de convolución. La matriz  $G(x)$  es no catastrófica si y sólo si se cumple cualquiera de las dos condiciones siguientes:

(a) El máximo común divisor de los  $b \times b$  menores de  $G(x)$  es una potencia de  $x$ .

(b)  $G(x)$  tiene una inversa derecha de peso finito.

La matriz  $G(x)$  de nuestro ejemplo es catastrófica, usando el teorema de Massey-Sain lo podemos corroborar,

$$G(x) = [x + x^2 \quad 1 + x] = [x(1 + x) \quad 1 + x],$$

el máximo común divisor de los menores es  $(1 + x)$  que no es una potencia de  $x$ , por lo tanto es una matriz catastrófica.

Un ejemplo de una matriz no catastrófica es

$$\begin{pmatrix} 1 & x & 1 + x & 0 \\ 0 & 1 + x & x & 1 \end{pmatrix},$$

cuyos menores son  $1 + x$ ,  $x$  y  $1$ , el máximo común divisor es  $1$  que lo podemos ver como  $x^0 = 1$ , por lo cual esta es una matriz no catastrófica.

Las matrices catastróficas generan lo que llamaremos los códigos catastróficos, los cuales deben ser evitados ya que generan un número infinito de errores en la decodificación, a partir de un número finito de errores en la secuencia recibida por el decodificador [24].

### 3.4. Códigos sistemáticos

Como ya vimos, toda matriz racional la podemos llevar a una matriz polinomial equivalente, equivalente en el sentido que generan códigos equivalentes, esto multiplicando el mínimo común múltiplo de los denominadores



por la matriz. De igual forma podemos llevar una matriz con entradas polinomiales a una matriz con entradas racionales, esto con otro tipo más de matrices generadoras para códigos de convolución, las matrices sistemáticas, cuya principal característica es que la secuencia de entrada  $u(x)$  forma parte de la palabra codificada  $v(x)$ , esto debido a que la matriz generadora contiene una submatriz diagonal identidad de tamaño  $b \times b$ , es decir  $I_b$ .

**Ejemplo.** Sea  $G(x) \in Mat(\mathbb{F}_2[x])$ , tal que

$$G(x) = (1 + x + x^2, 1 + x^2),$$

que es una matriz polinomial, es decir,  $G(x) \in Mat(\mathbb{F}_2[x])$ . Si multiplicamos a  $G(x)$  por  $\frac{1}{1+x+x^2}$ , tal que

$$\frac{1}{1+x+x^2} \cdot G(x) = G'(x),$$

obtenemos a  $G'(x)$ , que es una matriz equivalente a  $G(x)$ , y  $G'(x) \in Mat(\mathbb{F}_2(x))$

$$G'(x) = \left(1, \frac{1+x^2}{1+x+x^2}\right).$$

En general un codificador convolutivo de tasa  $R = b/c$  representado por una matriz es llamada *matriz generadora sistemática* si cada  $b$  secuencias generan  $c$  secuencias sin cambios.

Todo código de convolución tiene una matriz generadora sistemática, ya que podemos llevar cualquier matriz generadora a una sistemática. Por medio de las operaciones básicas vistas ya en este documento podemos obtener una matriz equivalente a  $G(x)$ ,  $G'(x)$ , tal que existan  $b$  secuencias identidad y estas pueden escribirse al inicio de la matriz, es decir,

$$G(x) = (I_b \ R(x)),$$

donde  $I_b$  es una matriz identidad  $b \times b$  y  $R(x)$  una matriz de tamaño  $b \times (c-b)$  la cual tiene entradas racionales, aunque las matrices generadoras sistemáticas no generan un código propiamente, ya que existen palabras que tendrán peso infinito [7].

En conclusión, todo código convolucional tiene matrices generadoras sistemáticas y no sistemáticas.

### 3.5. Matrices generadoras formadas a partir de sucesiones generadoras.

Ahora veremos las matrices generadoras asociadas a las sucesiones generadoras, ya que a partir de ellas podemos formar una matriz que al multiplicar el mensaje de entrada por dicha matriz, se obtiene la codificación del mensaje. Por ejemplo, si tenemos los polinomios generadores

$$\begin{aligned}g_1(x) &= 1 + x + x^2, \\g_2(x) &= 1 + x^2 + x^3, \\g_3(x) &= 1 + x^3,\end{aligned}$$

o sus respectivas sucesión:

$$\begin{aligned}g_1 &= (1, 1, 1, 0) = (g_0^{(1)}, g_1^{(1)}, g_2^{(1)}, g_3^{(1)}), \\g_2 &= (1, 0, 1, 1) = (g_0^{(2)}, g_1^{(2)}, g_2^{(2)}, g_3^{(2)}), \\g_3 &= (1, 0, 0, 1) = (g_0^{(3)}, g_1^{(3)}, g_2^{(3)}, g_3^{(3)}),\end{aligned}$$

si queremos por ejemplo codificar el mensaje  $u = 11001$  que tiene longitud 5, podemos formar una matriz generadora usando las sucesiones asociadas a los polinomios generadores, para que la multiplicación tenga sentido el tamaño de esta debe ser determinado por la longitud del mensaje a codificar, por la memoria  $m$  y por el número de polinomios o sucesiones, es decir  $c$ , en este caso,  $m = 3$ ,  $c = 3$  y  $l = 5$ , y el tamaño de la matriz es:

$$l \times c(l + m).$$

En nuestro ejemplo tenemos una matriz de tamaño  $5 \times 24$  cuyas entradas están en el conjunto  $Mat(\mathbb{F}_2)$  y determina un código  $C$ , tal que

$$C \subseteq \mathbb{F}_2^{24},$$

$$\dim(C) = 5,$$

y la matriz esta determinada por:

$$G = \begin{pmatrix} g_0^{(1)} & g_0^{(2)} & g_0^{(3)} & g_1^{(1)} & g_1^{(2)} & g_1^{(3)} & g_2^{(1)} & g_2^{(2)} & \cdots & 0 \\ 0 & 0 & 0 & g_0^{(1)} & g_0^{(2)} & g_0^{(3)} & g_1^{(1)} & g_1^{(2)} & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0^{(1)} & g_0^{(2)} & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & g_3^{(3)} \end{pmatrix},$$

Si el mensaje fuese por ejemplo de longitud 6, tendríamos que agregar un renglón más y el número de columnas se modificaría a 27, formando una matriz  $6 \times 27$ , ya que  $6 \times 3(6 + 3)$ . Ahora bien, si sustituimos los valores de la matriz tenemos:

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \cdots 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \cdots 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \cdots 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \cdots 1 \end{pmatrix},$$

que es la sucesión 111100110011 recorrida 3 lugares en cada renglón. En general serán  $c$  lugares. Si realizamos la multiplicación común de matrices, entrada a entrada  $u \cdot G$ , obtenemos

$$(11001) \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \cdots 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \cdots 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \cdots 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \cdots 1 \end{pmatrix} = 111011010101100100110011,$$

que es el mismo resultado que obtenemos de realizar el mensaje  $u$  con los 3 polinomios generadores, esto se puede corroborar en la sección de polinomios generadores, que incluye el mismo mensaje asociado a los polinomios generadores.

La matriz  $G$  la podemos ver como la composición de 3 matrices:  $G_1, G_2$  y  $G_3$  que están dadas por:

$$G_1 = \begin{pmatrix} g_0^{(1)} & g_1^{(1)} & g_2^{(1)} & g_3^{(1)} & 0 & 0 & 0 & 0 \\ 0 & g_0^{(1)} & g_1^{(1)} & g_2^{(1)} & g_3^{(1)} & 0 & 0 & 0 \\ 0 & 0 & g_0^{(1)} & g_1^{(1)} & g_2^{(1)} & g_3^{(1)} & 0 & 0 \\ 0 & 0 & 0 & g_0^{(1)} & g_1^{(1)} & g_2^{(1)} & g_3^{(1)} & 0 \\ 0 & 0 & 0 & 0 & g_0^{(1)} & g_1^{(1)} & g_2^{(1)} & g_3^{(1)} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$(11001) \cdot G_1 = 10011110,$$

$$G_2 = \begin{pmatrix} g_0^{(2)} & g_1^{(2)} & g_2^{(2)} & g_3^{(2)} & 0 & 0 & 0 & 0 \\ 0 & g_0^{(2)} & g_1^{(2)} & g_2^{(2)} & g_3^{(2)} & 0 & 0 & 0 \\ 0 & 0 & g_0^{(2)} & g_1^{(2)} & g_2^{(2)} & g_3^{(2)} & 0 & 0 \\ 0 & 0 & 0 & g_0^{(2)} & g_1^{(2)} & g_2^{(2)} & g_3^{(2)} & 0 \\ 0 & 0 & 0 & 0 & g_0^{(2)} & g_1^{(2)} & g_2^{(2)} & g_3^{(2)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$(11001) \cdot G_2 = 11100011,$$

$$G_3 = \begin{pmatrix} g_0^{(3)} & g_1^{(3)} & g_2^{(3)} & g_3^{(3)} & 0 & 0 & 0 & 0 \\ 0 & g_0^{(3)} & g_1^{(3)} & g_2^{(3)} & g_3^{(3)} & 0 & 0 & 0 \\ 0 & 0 & g_0^{(3)} & g_1^{(3)} & g_2^{(3)} & g_3^{(3)} & 0 & 0 \\ 0 & 0 & 0 & g_0^{(3)} & g_1^{(3)} & g_2^{(3)} & g_3^{(3)} & 0 \\ 0 & 0 & 0 & 0 & g_0^{(3)} & g_1^{(3)} & g_2^{(3)} & g_3^{(3)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$(11001) \cdot G_3 = 11010001,$$

cada matriz  $G_i$  es una matriz circulante y  $G_i \in Mat_{l \times (l+m)}(\mathbb{F}_2)$ . Nótese que cada  $G_i$  es una matriz de tamaño  $5 \times 8$  y cada una de estas contiene a la sucesión asociada a su respectivo polinomio generador. Son  $c$  matrices de tamaño  $l \times (l+m)$ .

También podemos notar que con la concatenación de las tres matrices obtenemos una matriz  $G'$  tal que:

$$G_1, G_2, G_3 := G',$$

la concatenación la aplicamos respecto a las columnas de las matrices, como cada  $G_i$  tiene 8 columnas entonces al concatenar tenemos que  $G'$  es una matriz de tamaño  $5 \times 24$ , y precisamente

$$G' = G.$$

De igual manera, podemos considerar  $u \cdot G_i = v_i$   $i = 1, 2, 3$  y concatenar  $v_1, v_2$  y  $v_3$ ,

$$v_1, v_2, v_3 = v,$$

donde  $v = u \cdot G$ , por lo cual

$$10011110 \oplus 11100011 \oplus 11010001 = 111\ 011\ 010\ 101\ 100\ 100\ 110\ 011.$$

En general, estas matrices permiten codificar un mensaje  $u$  de longitud  $l$  (o grado  $l - 1$ ), con las sucesiones asociadas a los polinomios generadores

$$\begin{aligned} g_1 &= (g_0^{(1)}, g_1^{(1)}, g_2^{(1)}, \dots, g_m^{(1)}), \\ &\dots\dots\dots \\ g_c &= (g_0^{(c)}, g_1^{(c)}, g_2^{(c)}, \dots, g_m^{(c)}), \end{aligned}$$

donde  $g_i^{(j)} \in \mathbb{F}_2$ ,  $j = 1, 2, \dots, c$ ;  $i = 1, 2, \dots, m$ . Con estos valores podemos formar la matriz:

$$G = \begin{pmatrix} g_0^{(1)} & g_0^{(2)} & \dots & g_0^{(c)} & g_1^{(1)} & g_1^{(2)} & \dots & g_1^{(c)} & g_2^{(1)} & \dots & 0 \\ 0 & 0 & 0 & 0 & g_0^{(1)} & g_0^{(2)} & \dots & g_0^{(c)} & g_1^{(1)} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & g_m^{(c)} \end{pmatrix},$$

que es una matriz  $l \times c(l + m)$  y la podemos descomponer en  $c$  submatrices circulantes  $G_i$ ,  $i = 1, 2, \dots, c$ , tal que:

$$G_i = \begin{pmatrix} g_0^{(i)} & g_1^{(i)} & g_2^{(i)} & g_3^{(i)} & \dots & g_m^{(i)} & 0 & \dots & \dots & 0 \\ 0 & g_0^{(i)} & g_1^{(i)} & g_2^{(i)} & g_3^{(i)} & \dots & g_m^{(i)} & 0 & \dots & 0 \\ 0 & 0 & g_0^{(i)} & g_1^{(i)} & g_2^{(i)} & g_3^{(i)} & \dots & g_m^{(i)} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & g_m^{(i)} \end{pmatrix},$$

cada  $G_i$  es una matriz  $l \times (l + m)$  y la concatenación  $G_1, G_2, \dots, G_c = G$ . Estas matrices nos permite codificar de una manera equivalente a los polinomios generadores.

# Capítulo 4

## Decodificación

En este capítulo abordaremos el proceso de decodificación en los códigos de convolución, para esto primero veremos la construcción de los diagramas de trellis, en particular para códigos de tasa  $R = 1/2$ .

También definiremos el concepto de distancia libre, que es un parámetro más de los códigos de convolución, el cual mide el desempeño de estos. Finalizaremos con el algoritmo de Viterbi, que es el método más popular y eficiente para la decodificación de los códigos de convolución.

### 4.1. Diagramas de trellis

A los códigos de convolución les podemos asociar varios diagramas como son el de estados, de ciclos y de trellis, y estos dependen directamente del codificador, la tasa y la memoria.

El diagrama de trellis es un diagrama en forma de red que caracteriza a su respectivo código de convolución, ya que está diseñado para que todas las palabras del código estén representadas por un camino del diagrama.

El diagrama de trellis es la repetición de otro diagrama, llamado diagrama de ciclos, este se repite tantas veces como sea necesario, hasta completar la longitud de las palabras del código.

Los diagramas de estados están formados por  $2^m$  estados, los cuales están representados por el conjunto de los elementos del espacio vectorial  $\mathbb{F}_2^m$ , es decir, por:

$$\mathbb{E} := \mathbb{F}_2^m = \{(a_1 a_2 \cdots a_m) : a_i \in \mathbb{F}_2\},$$

siempre partiendo del estado 0 (al cual llamaremos estado inicial) y terminando en este mismo estado. Cada uno de los caminos del trellis son una palabra del código de convolución.

En el caso en que  $R = 1/c$ , de cada estado siempre parten dos líneas que determinan las ramificaciones, una para el caso de que el bit de entrada al codificador fuese 0 y otra para el caso de que fuese 1, para  $b \neq 1$  en lugar de entrar un bit al codificador entran  $b$  y se hacen las posibles  $2^b$  combinaciones, el trellis asociado tiene un mayor número de ramificaciones y el proceso de decodificación es más complejo, por la cual esta opción rara vez es adoptada.

Un diagrama asociado al diagrama de estados es el diagrama de ciclos, el cual está directamente relacionado con el primero. Las líneas en un diagrama de ciclos representan las conexiones del codificador y van de algo que llamamos ciclo  $i - 1$  hasta el siguiente ciclo  $i$  del diagrama y sobre cada una de estas líneas escribiremos las  $c$  salidas.

A cada codificador convolutivo le podemos asociar un diagrama de estados, uno de ciclos y uno de trellis. Cada tasa tiene asociado un diagrama, el cual básicamente es el mismo, las conexiones son las mismas, lo que cambia es la información sobre las líneas.

**Ejemplo.** Sean

$$p_1(i) = u(i) + u(i - 1),$$

$$p_2(i) = u(i) + u(i - 2),$$

las ecuaciones de paridad que conforman el codificador del código de convolución de memoria  $m = 2$  y tasa  $R = 1/2$ . Y sea  $u = 11011$  el mensaje que se desea codificar. Mientras que la longitud de restricción es  $k = 3 = m + b$ . Para usar las ecuaciones de paridad tenemos que agregar dos ceros a la derecha de  $u$  y dos ceros a su izquierda,  $u = 001101100$ , cuyo registro para cada tiempo lo podemos ver en la siguiente tabla:

i	Registro del ciclo	Bits de salida
0	001	11
1	011	01
2	110	11
3	101	10
4	011	01
5	110	11
6	100	01
7	000	00

(4.1)

el registro del ciclo es  $u(i-2)u(i-1)u(i)$  y los bits de salida son  $p_1(i)p_2(i)$  generados por los bits en el registro. Para realizar el diagrama de estados correspondientes este código de parámetros  $(2, 1, 2)$  lo haremos paso a paso.

Como ya mencionamos al inicio de esta sección, para este código (con esta tasa) tendremos  $2^m = 2^2$  estados, donde los estados son el conjunto de los elementos de del espacio vectorial  $\mathbb{F}_2^2$ ,

$$\mathbb{F}_2^2 = \{(0, 0), (01), (1, 0), (1, 1)\},$$

los elementos de  $\mathbb{F}_2^2$  forman una gráfica, el diagrama de estados, los cuales deben iniciar y terminar en el estado 0.

Para la construcción del diagrama de estados de tasa  $R = 1/2$  y  $m = 2$  que implica  $k = 3$ , la ventana tiene espacio para cubrir 3 bits que son los registros del ciclo, cada bit que entra al codificador produce dos bits y desplaza un bit, es decir, si en la ventana están los bits  $b_{i-1}, b_i, b_{i+1}$  y dejamos entrar el siguiente bit  $b_{i+2}$  forzosamente para que este entre tiene que salir el bit  $b_{i-1}$ , quedando la ventana ocupada por  $b_i, b_{i+1}, b_{i+2}$ , realizando este mismo procedimiento, la ventana contiene algo de la forma  $b_i00, b_i01, b_i10$  o  $b_i11$ , ya que son todas las posibles combinaciones.

Los últimos dos dígitos invertidos de la ventana determinan el estado en el cual nos encontramos y los últimos dos dígitos invertidos de cada uno de la posible nueva información en la ventana determinan a los estados con los cuales se conecta este. Por ejemplo, si los bits en la ventana son  $b_i00$  la ventana contendrá 001 o 000 que nos indicará que el estado 00 esta conectado



con el estado 10 y 00, en el caso de tener  $b_i01$  al entrar el nuevo bit podemos tener 010 o 011, lo que nos indica que el estado 10 esta conectado con 01 y 11. Mientras que  $b_i10$  tiene las posibilidades 100 y 101 que nos dice que el estado 01 se conecta con 00 y 10, las posibilidades para  $b_i11$  son 110 y 111 que nos hace conectar a 11 con 01 y 11.

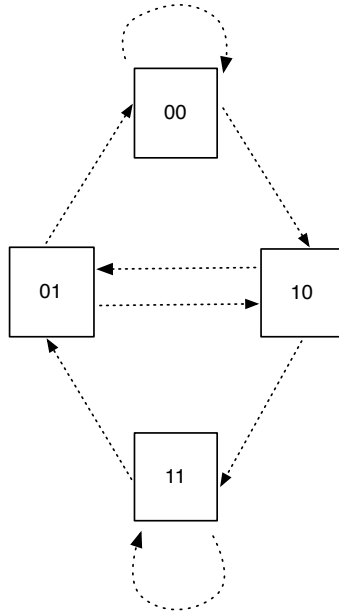


Figura 4.1: La información que se transmite en este diagrama de estados es la que varia según el codificador que usemos.

En los diagramas de estados nos interesa iniciar y finalizar en el estado 0, en la tabla (4.1) podemos ver que partimos del estado 00 y pasamos al estado 10, enseguida a 11 y pasa al 01, regresamos 10 y va nuevamente 11, vamos al 01 y terminamos en el 00.

Finalmente en el diagrama de estados agregamos los valores  $u_i/p_1(i)p_2(i)$  a cada conexión con el estado correspondiente, donde  $u_i$  es el bit perteneciente al mensaje  $u$  que entró al codificador en el ciclo  $i$ , mientras que  $p_1(i)p_2(i)$  son los bits que se producen por las dos ecuaciones de paridad en ese mismo ciclo.

Con el diagrama de estados podemos realizar el diagrama de ciclos, que

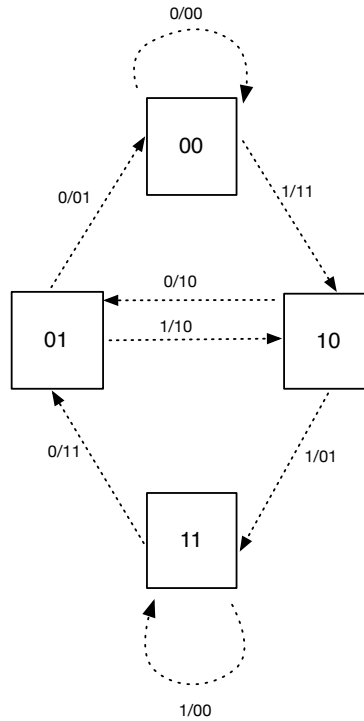


Figura 4.2: Diagrama de estados asociado al código  $(2, 1, 2)$ .

esta basado en este primero. Listando por duplicado cada estado, en dos columnas diferentes, a la columna de la izquierda le llamaremos ciclo  $i - 1$ , mientras que a la de la derecha ciclo  $i$ . En la figura 4.3 podemos observar el acomodo del diagrama de ciclos, en el cual generalmente se quedan en los extremos (arriba y abajo) los estados que tienen una conexión consigo mismos (00 y 11).

Del diagrama de estados vemos que el estado 00 se conecta con 00 y 10, el estado 01 con 10 y 00, el estado 10 con 11 y 01, mientras el estado 11 con 11 y 01, por lo cual, partimos del ciclo  $i - 1$  del estado 00 y este lo conectamos con los estados 00 y 10 del ciclo  $i$ , el estado 01 con del ciclo  $i - 1$  lo conectamos con los estados 10 y 00 del ciclo  $i$  y de igual manera los otros dos estados. Finalmente se coloca sobre las líneas la respectiva información

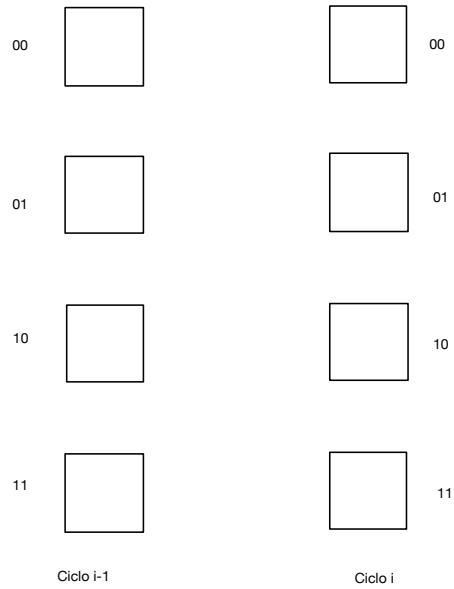


Figura 4.3:

que ya poseemos en el diagrama de estados, obteniendo así el diagrama de la figura 4.4

A su vez, con el diagrama de ciclos podemos obtener el diagrama de trellis asociado al mismo código. El diagrama de trellis consiste en repetir el diagrama de ciclos tantas veces como  $l + m$  ( $l$  longitud del mensaje), en este caso  $5 + 2$ .

Por ejemplo, una palabra del código esta representada por la trayectoria marcada, esta trayectoria muestra  $u_i/p_1(i)p_2(i)$ , al concatenar los  $u_i$  obtenemos el mensaje de entrada 1111100, y al concatenar los  $p_1(i)p_2(i)$  obtenemos la palabra del código 11 01 00 10 01 11 01.

Cuando la información es enviada por un canal ruidoso, esta posiblemente recibe una alteración debido al ruido, el diagrama de trellis ayuda a determinar el camino más cercano a la información enviada.

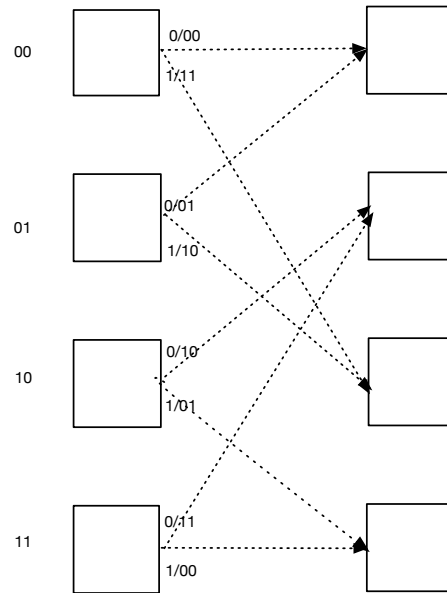


Figura 4.4: Diagrama de ciclos basado en el diagrama de estados de la figura 4.2.

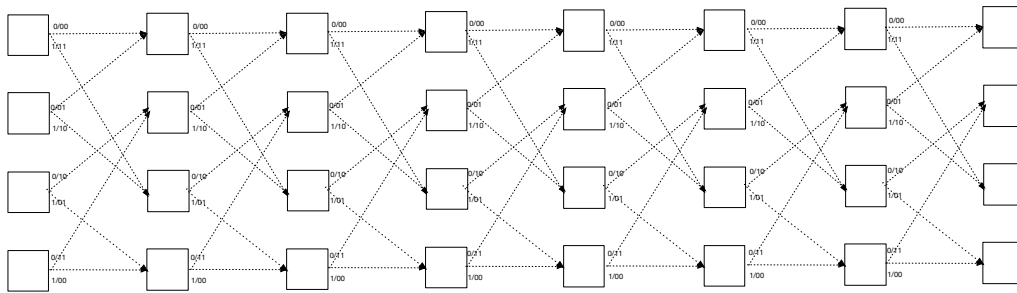


Figura 4.5: Diagrama de trellis asociado al código de convolución (2, 1, 2), generado por las ecuaciones  $p_1(i) = u(i) + u(i - 1)$ ,  $p_2(i) = u(i) + u(i - 2)$ .

Los diagramas de estados, ciclos y trellis para cualquier código (2, b, 2) son el mismo, en el sentido que las conexiones siempre son las mismas, lo que cambia es la información de las líneas, esa información es la que genera el codificador, depende directamente de este.

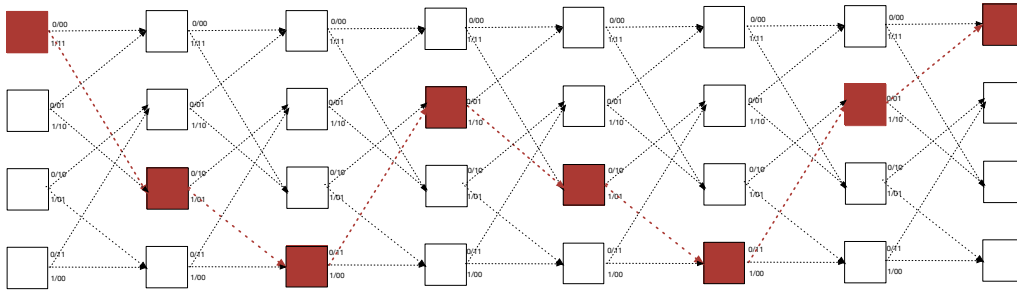


Figura 4.6: Ejemplo de una palabra del código y representada en el trellis.

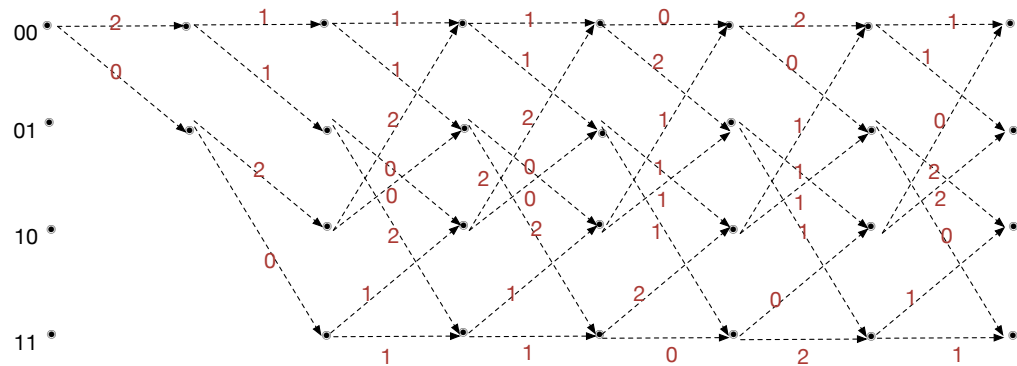


Figura 4.7: Otra versión más común del diagrama de trellis (fig 4.6), para el mismo código.

**Ejemplo.** Tenemos el mensaje  $u$ ,

$$u = 1001,$$

una tasa  $R = 1/2$  y el siguiente par de ecuaciones de paridad:

$$p_1(i) = u(i) + u(i - 1) + u(i - 2), \quad (4.2)$$

$$p_2(i) = u(i) + u(i - 1). \quad (4.3)$$

como  $m = 2$  tenemos que agregar dos ceros a la derecha de  $u$  y dos a la izquierda, esto para completar la ventana desplazante que tiene longitud 3,

$u = 00100100$ .

Usando las ecuaciones de paridad obtenemos el mensaje:  $v = 11 \ 11 \ 10 \ 11 \ 11 \ 10$ , este ejemplo fue visto en capítulos anteriores el cual para mayor detalle puede consultarse, y de forma análoga al ejemplo anterior puede comprobarse y obtenerse las siguientes conexiones, el estado 00 esta conectado con 00 y con 01, el estado 01 con 10 y 11, el estado 10 con 01 y 00, mientras que el estado 11 esta conectado con 11 y 10, que determinan los siguientes diagramas.

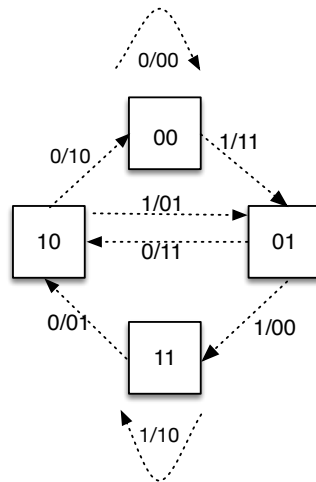


Figura 4.8: Diagrama de estados, asociado a las ecuaciones de paridad 4.2 y 4.3.

De igual manera que en el ejemplo anterior, del diagrama de estados podemos pasar al diagrama de ciclos correspondiente al código, conectando los ciclos  $i - 1$  con los respectivos ciclos de  $i$  y posteriormente colocando la información  $i \frac{1}{2}n$  correspondiente sobre las líneas del diagrama.

Del diagrama de ciclos pasamos directamente al diagrama de Trellis.

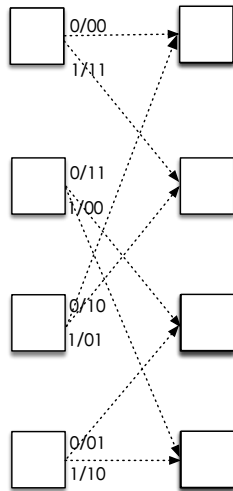


Figura 4.9: Diagrama de ciclos asociado al diagrama de estados 4.8.

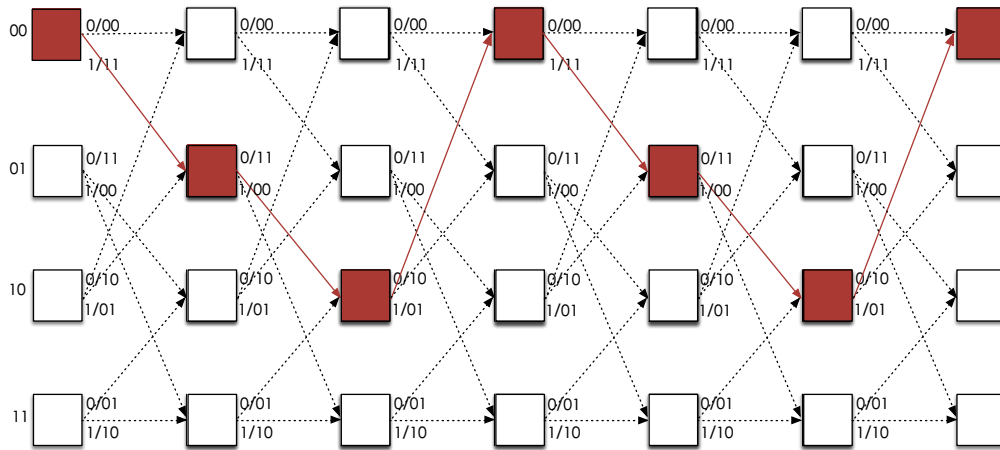


Figura 4.10: Trellis asociado a las ecuaciones de paridad  $p_1(i) = u(i) + u(i - 1) + u(i - 2)$ ,  $p_2(i) = u(i) + u(i - 1)$ , la trayectoria marcada es la que corresponde a la palabra 1001.

Aunque no se han definido los diagramas de trellis, ciclos y estados para matrices, recordemos que tenemos una biyección entre los polinomios ge-

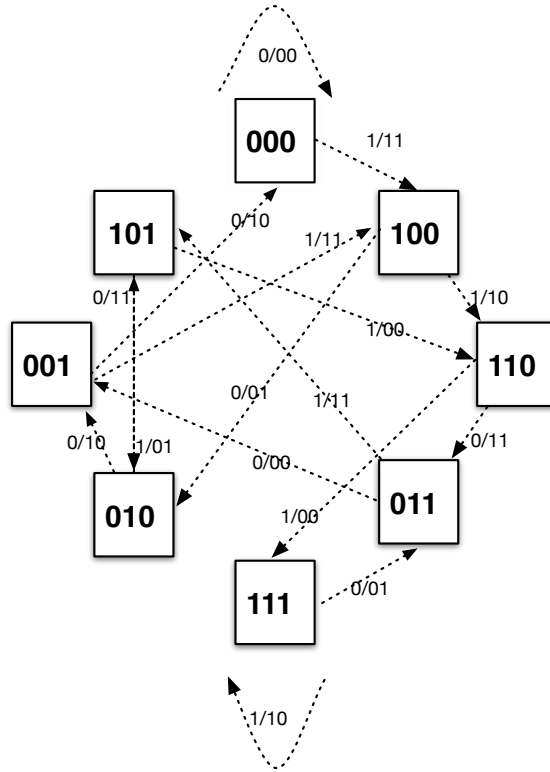


Figura 4.11: Diagrama de estados asociado a la matriz  $[1 + x^2 + x^3 \quad 1 + x]$ .

neradores y ecuaciones de paridad, las cuales se pueden arreglar en forma matricial. Por lo cual en la figura 4.11 el diagrama de estados esta definido por: codificador, tasa y memoria, esta tasa determina una matriz  $2 \times 3$ , que en el diagrama representan 2 entradas y 3 salidas. El número de estados es  $2^3$  y son el conjunto:

$$\{(a_0 a_1 a_2) : a_i \in \mathbb{F}_2, i = 0, 1, 2\} := E = \mathbb{F}_2^3,$$

$$\text{card}(E) = 2^3.$$

En la figura 4.12 se muestra el diagrama de trellis asociado a la matriz  $[1 + x^2 + x^3, 1 + x]$ , que es equivalente a las ecuaciones de paridad  $p_1(i) = u(i) + u(i - 2) + u(i - 3)$  y  $p_2(i) = u(i) + u(i - 1)$ .



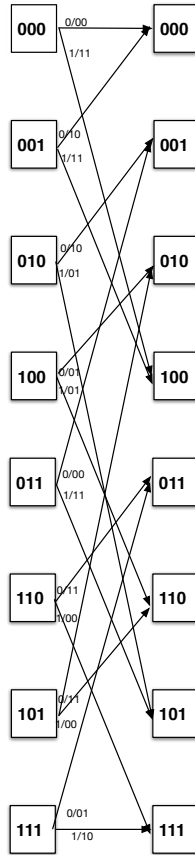


Figura 4.12: Diagrama de ciclos asociado al diagrama de estados de la figura 4.11.

## 4.2. Distancia libre $d_{free}$

En los códigos de bloque la *distancia de Hamming*  $d_H$  es una función que a cada par de palabras de la misma longitud les asocia el número de sus discordancias,

$$d_H : \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{N}$$

$$(v, v') \mapsto d_H(v, v'),$$

$$d_H(v, v') = d_H((v_1, \dots, v_n), (v'_1, \dots, v'_n)) := |\{i : v_i \neq v'_i\}|,$$

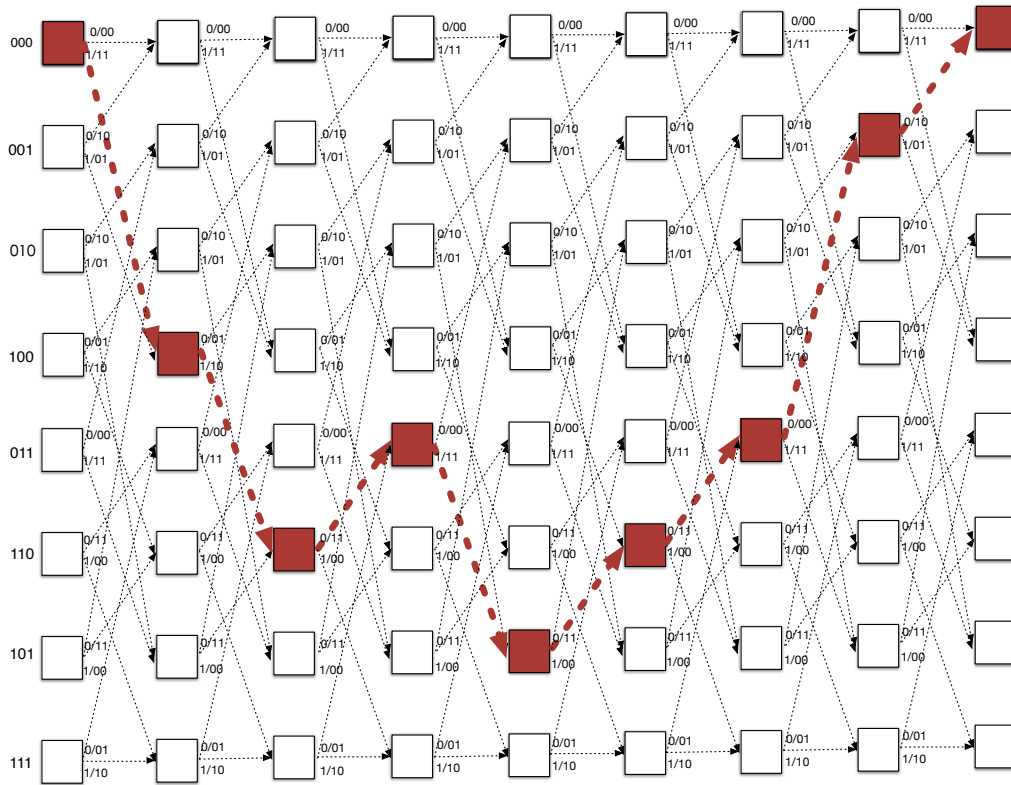


Figura 4.13: Trellis asociado a la matriz  $G(x) = [1 + x^2 + x^3 \quad 1 + x]$ . La trayectoria marcada es el mensaje 11011.

la mínima de estas distancias la llamamos *distancia mínima*,

$$d_{min} = \min\{d_H(v, v') : v, v' \in C, v \neq v'\}.$$

La distancia mínima ayuda a determinar una cota superior para la capacidad de corrección de errores,  $e$ , de dichos códigos:

$$e \leq \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor.$$

Algo similar sucede en los códigos de convolución. Como toda serie de Laurent  $f(x) \in \mathbb{F}_2((x))$ , puede expresarse como un cociente de polinomios

$$f(x) = p(x)/q(x),$$

donde  $p(x)$  y  $q(x) \in \mathbb{F}_2[x]$ ,  $q(x) \neq 0$ , y viceversa, en estos casos el peso  $w$  de  $f(x)$  lo definimos como el número de coeficientes no ceros en la expansión de la serie de Laurent  $f(x)$ . Este peso puede ser infinito [3], en particular un polinomio tendrá peso finito.

El peso de una  $n$ -tupla de un elemento de  $v(x) \in \mathbb{F}_2(x)^n$ ,  $v(x) = (v_1(x), \dots, v_n(x))$ , es la suma de los pesos de cada una de las componentes, mientras que la distancia entre dos elementos  $v(x), v'(x) \in \mathbb{F}_2(x)^n$  es definida como  $w(v(x) - v'(x))$ .

La *distancia libre*  $d_{free}$  de un código de convolución,  $C$ , es definido como la mínima de las distancias de Hamming y coincide con la definición de la distancia mínima de los códigos de bloque,

$$d_{free} = \min\{d_H(v(x), v'(x)) : v(x), v'(x) \in C, v(x) \neq v'(x)\},$$

$$d_{free} = \min\{w(v(x)) : v(x) \in C, v(x) \neq 0\}.$$

Del mismo modo que en un código de bloque, en los cuales la distancia mínima determina una cota superior de corrección de errores, en los códigos de convolución la distancia libre  $d_{free}$  es la que determina la máxima capacidad de corrección de errores. Todo código de convolución es capaz de corregir hasta:

$$\left\lfloor \frac{d_{free} - 1}{2} \right\rfloor,$$

[3], [19], [23].

Por ejemplo, supongamos que tenemos un código [2,1,3] con  $d_{free} = 5$ , este es capaz de corregir a lo más 2 errores por cada 8 bits.

La cantidad de símbolos corregidos depende directamente de la distancia libre, entre mayor sea esta mayor corrección de errores tiene el código, por tal motivo es de gran importancia encontrar códigos con la mayor distancia libre posible.

En general no es sencillo determinar la distancia libre de un código de convolución, la cual puede ser calculada por aproximaciones, aunque lo más certero podría ser calcular todas las palabras del código y ver cuales de ellas tienen la mínima distancia de Hamming o el menor peso, pero esto no es eficiente.

Aún no existe un método general algebraico para construir códigos de convolución con distancia libre máxima, pero cabe señalar que existen tablas que dan la distancia libre de varios códigos, las cuales contienen considerable cantidad de tasas con diferentes longitudes de restricción  $k$  y en varios casos proporcionan diferentes codificadores que los generan. Estas tablas han surgido de la conjunción de trabajos y resultados de Larsen, Paaske y Dault. Larsen determino para tasas  $1/2$ ,  $1/3$  y  $1/4$ , Paaske para tasas  $2/3$  y  $3/4$ , y Daut para las tasas  $2/3$ ,  $3/4$ ,  $1/5$ ,  $2/5$ ,  $3/5$ ,  $4/5$ ,  $1/6$ ,  $5/6$ ,  $1/7 - 6/7$ ,  $1/8$ ,  $3/8$ ,  $5/8$ , y  $7/8$  que son las tasas más usuales. Véase [23].

### 4.3. Algoritmo de Viterbi

El algoritmo de Viterbi lleva el nombre de su creador A. Viterbi. Este algoritmo es un método de decodificación para los códigos de convolución, quizá el más eficiente, es ampliamente usado para la corrección de errores en los códigos de convolución, ya que permite encontrar las secuencias de estados más probable en un modelo oculto de Markov, el trellis es uno de estos modelos. Debido al enfoque de esta tesis no abordamos de manera profunda este algoritmo, que por su importancia y complejidad amerita un trabajo más extenso, nosotros solo abordamos las cuestiones básicas, para mayores detalles consúltese [3], [22], [30].

Para el algoritmo de Viterbi hacemos uso del diagrama de trellis asociado al código en cuestión. El algoritmo de Viterbi permite detectar las secuencias óptimas del diagrama de trellis para detectar y así corregir los errores recibidos por los canales ruidosos empleados para la transmisión de la información.

El algoritmo de Viterbi tiene dos vertientes: la **decisión fuerte** (hard decision) y la **decisión suave** (soft decision). En cada una de estas existen dos diferentes métricas, la métrica de la rama (branch metric), BM, y la métrica de trayectoria (path metric), PM. La BM es una medida de distancia entre lo que es transmitido (esto se obtiene con las posibles conexiones que el trellis nos proporciona) y lo que es recibido y se define en cada rama del trellis, en la hard decision la BM es es la distancia de Hamming, mientras que la PM es un valor asociado con cada estado del trellis.

La decisión fuerte y decisión suave tienen el mismo algoritmo, lo que cambia son las métricas, en ambos casos trabajamos la acumulación en cada estado y finalmente se elige el camino que tenga la mínima distancia de la métrica con la cual estemos.

La decisión fuerte consiste en encontrar el camino con la mínima distancia de Hamming, esto se hace comparando la secuencia recibida con cualquiera de las secuencias posibles del diagrama de trellis. A cada rama se le asocia una etiqueta (distancia entre los bits recibidos y las posibles salidas del codificador). Hacemos esto con todas las ramas, acumulamos todas las distancias de las etiquetas de los posibles caminos y elegimos el camino de mínima distancia.

Conservamos el camino con la distancia mínima y los otros caminos los podamos, en caso de tener la misma distancia dejamos ambos y en pasos posteriores los caminos se van podando (basándonos en las distancias mínimas), hasta al final es posible tomar la decisión del camino.

**Ejemplo.** El diagrama de trellis siguiente está asociado a un código de convolución de memoria  $m = 2$  y  $R = 1/2$ . Supongamos recibimos el mensaje  $r = 11\ 10\ 10\ 11\ 10\ 10$ ,

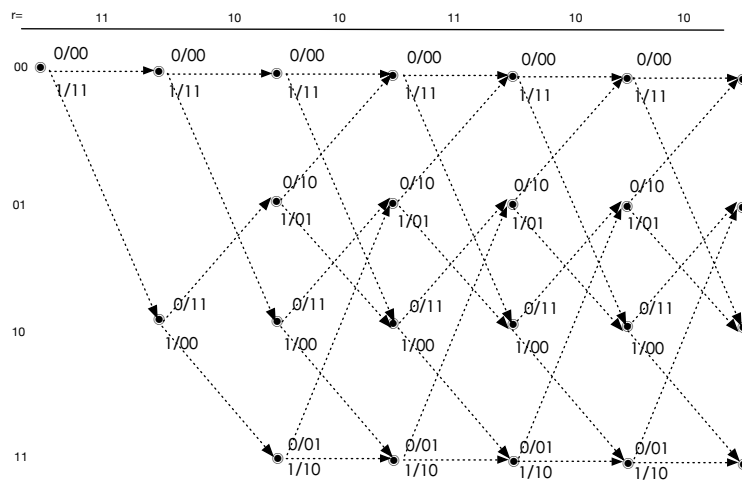


Figura 4.14: Arriba del diagrama observamos el mensaje recibido. Usaremos la hard decision

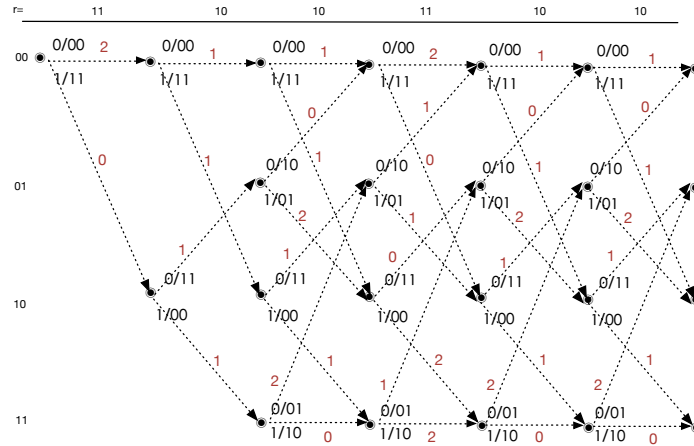


Figura 4.15: Colocamos las respectivas distancias de Hamming en las ramas (distancia de la información del diagrama con la que es recibida), de izquierda a derecha recibimos 11, la distancia con 00 es 2 y con 11 es 0, 2 y 0.

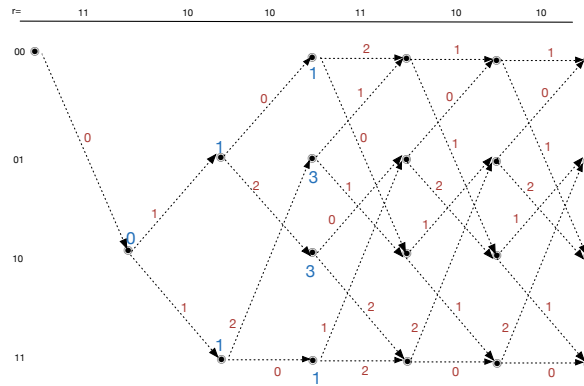


Figura 4.16: En el segundo nivel de estados tenemos distancia de Hamming 2 y 0, preservamos la mínima de estas y cortamos las ramificaciones que no tienen sentido sin la rama correspondiente a la distancia 2. En caso de que las distancias fuesen iguales dejamos ambas. Los números azules son las distancias de Hamming acumuladas.

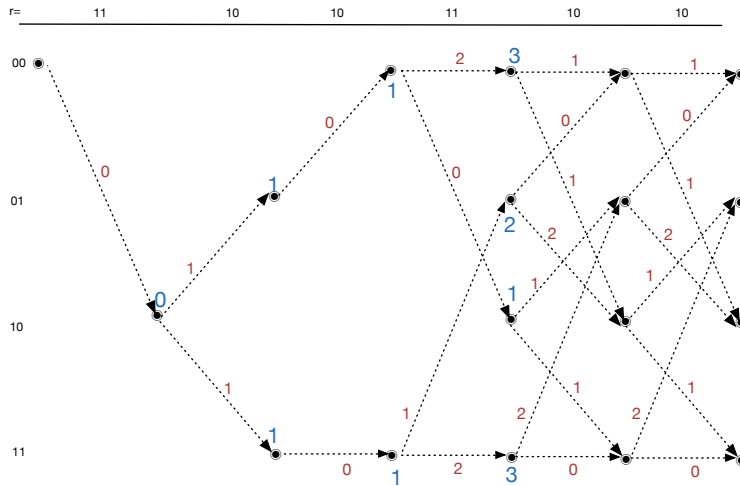


Figura 4.17: Vamos cortando los caminos con mayores distancia, aquí (en comparación con la figura anterior) podemos ver que podemos los caminos con distancias mayores.

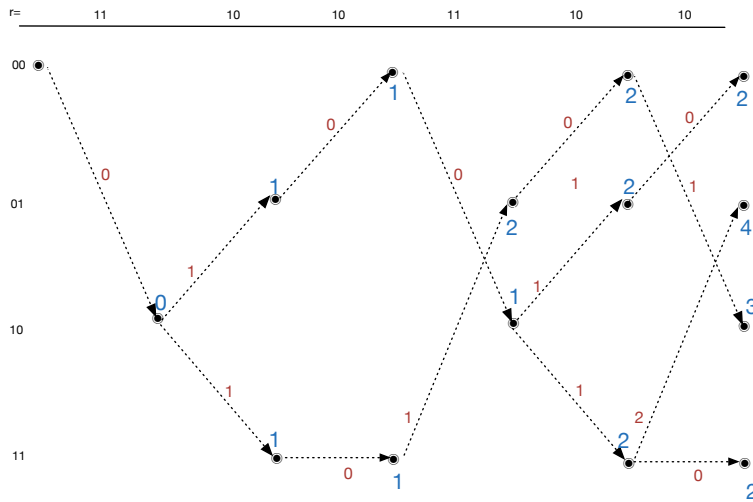


Figura 4.18:

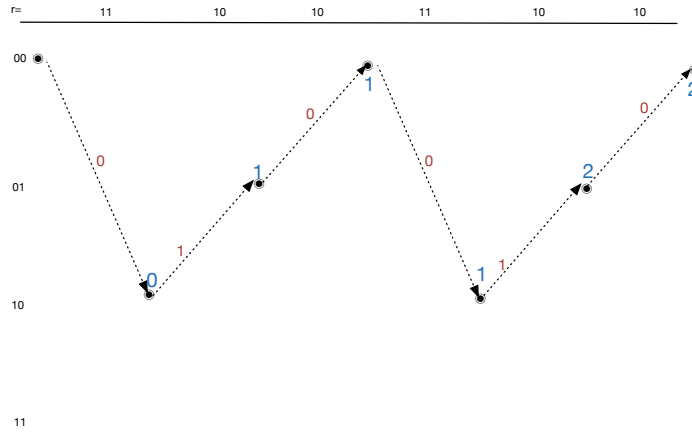


Figura 4.19: Siguiendo con el razonamiento de la distancia mínima obtenemos dos caminos con distancia mínima 2, pero necesitamos que comience y termine en el estado 00, que la cumple una sola opción.

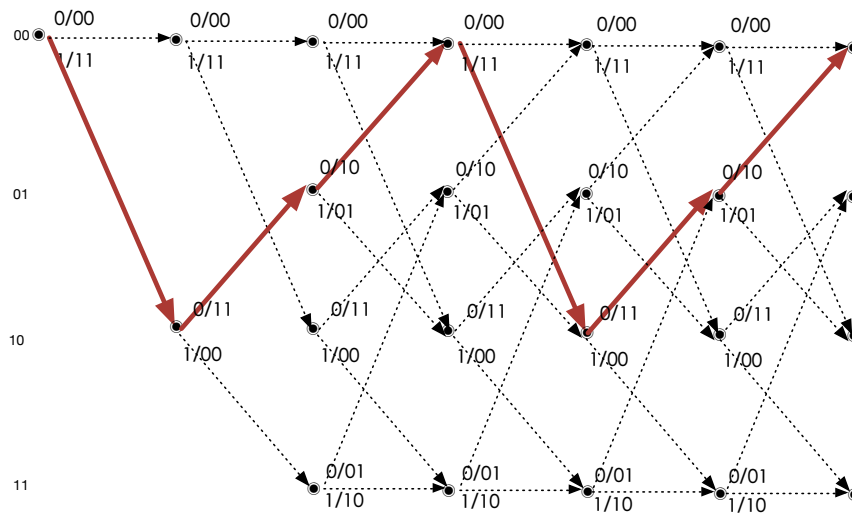


Figura 4.20: La secuencia correcta es  $v = 11\ 11\ 10\ 11\ 11\ 10$  y el mensaje original  $u = 100100$ , esta la leemos de la información que se transmite en la trayectoria correcta.



La decisión fuerte se apoya en mecanismos que se encargan de discretizar las señales recibidas, comparándolas con un umbral antes de pasarlas al decodificador, valores menores a 0.5 los convierte en 0 y los superiores o iguales los convierte en 1, por ejemplo si recibimos 0.500001 lo tomaremos como 1, la confianza es menor que si tuviésemos 0.999999, aunque ambos tomen el valor de 1, y el decodificador los trata de la misma manera a pasar de la abrupta diferencia, por lo cual solo trabajamos con 0 o 1.

La decisión suave no discretiza, más bien, utiliza una función continua de la muestra analógica como entrada al decodificador, por ejemplo, si el bit de paridad esperado es 0 la señal recibida es 0.4, nosotros usaremos 0.4 como el valor del bit o 0.4 bajo alguna función. Por razones técnicas, una atractiva métrica de la decisión suave es el cuadrado de la diferencia entre la señal recibida y la esperada. Si el codificador convolutivo produce  $c$  bits de paridad y los  $c$  bits corresponden a las muestras analógicas  $r = r_1, r_2, \dots, r_c$ , podemos construir una BM para la decisión suave de la siguiente manera (u otras de forma similar):

$$BM_{soft}[v, r] = \sum_{i=1}^c (v_i - r_i)^2,$$

donde  $v = v_1, \dots, v_c$  son los bits esperados consúltese [13].

Este algoritmo para la decisión suave es idéntico al anterior descrito por la decisión fuerte, solo que la métrica de la rama no es un entero como la distancia de Hamming, es un real positivo, y esta métrica está estrechamente relacionada con la probabilidad de que la decodificación sea correcta cuando el canal experimenta ruido gaussiano aditivo.

En la decisión suave la información es transmitida por un canal con ruido gaussiano, un canal que lo podemos describir con sus tres componentes: modulador, un canal en forma de onda y un demodulador. El modulador convierte cada bit a una onda, véanse las diferentes modulaciones en: [14], [15], [17], [18], [19]. Las modulaciones miden la fase, amplitud, frecuencia o una combinación de estas de la onda en cuestión y cada bit lo convierte a una forma de onda con duración de  $l$  segundos.

Usando la decisión suave, supongamos que se codificó un mensaje y se trans-

mitió bajo una modulación cuya secuencia recibida es

$$r = (1, -0.5), (1, 0.1), (-1, -1.2), (-0.8, -0.8), (-0.8, 0.8), (0.5, 0.5).$$

Tenemos una función  $D$ , que le llamamos demodulación, tal que

$$D : \mathbb{F}_2 \longrightarrow \{-1, 1\}$$

$$x \longmapsto (-1)^{x+1},$$

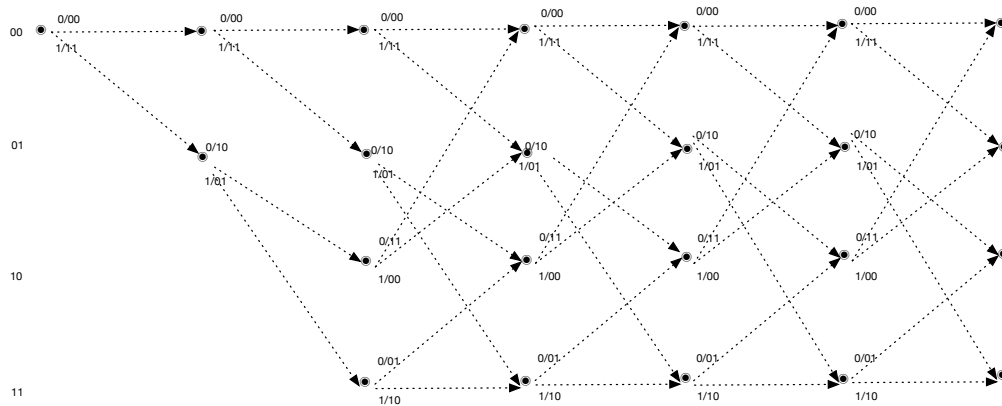
donde  $\{-1, 1\}$ , son dos raíces cuadradas de la unidad  $\mu_2 = \{-1, 1\} \subset \mathbb{C}$ , y

$$D(0) \longrightarrow (-1)^{0+1} = -1$$

$$D(1) \longrightarrow (-1)^{1+1} = 1,$$

ME PARECE QUE LA DEFINICION DE LA FUNCION  $D$  NO ES CORRECTA.

por lo cual demodulación asigna  $0 \longrightarrow -1$ , y  $1 \longrightarrow 1$ , entonces, dado el trellis:



para cada par de bits de los estados del trellis aplicamos la demodulación,

$$D(00) \longrightarrow -1, -1,$$

$$D(01) \longrightarrow -1, 1,$$

$$D(10) \longrightarrow 1, -1,$$

$$D(11) \longrightarrow 1, 1.$$

Posteriormente calculamos la distancia cuadrada euclidiana correspondiente a cada rama, por ejemplo para calcular  $d^2((1, -0.5), (0,0))$  primero aplicamos la demodulación a 00,  $D(0,0) = (-1, -1)$ , y restamos coordenada a coordenada,  $d^2((1,-0.5),(-1,-1))=(1+1)^2+(-0.5+1)^2=2^2+(-0.5)^2=4.25$ , enseguida calculamos  $d^2((1,-0.5),(1,1))=(1-1)^2+(-0.5-1)^2=0^2+(-1.5)^2=2.25$ .

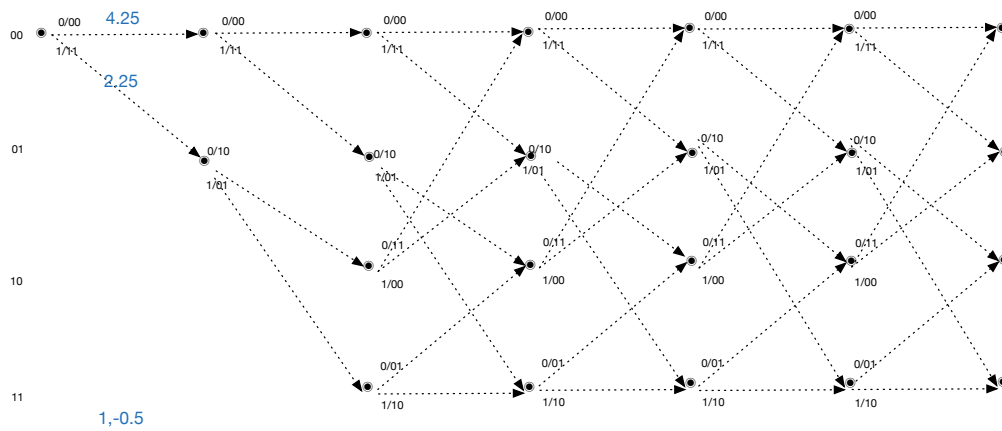


Figura 4.21: Distancia euclidiana cuadrada de  $(1, -0.5)$  a  $(0,0)$  y de  $(1, -0.5)$  a  $(1,1)$

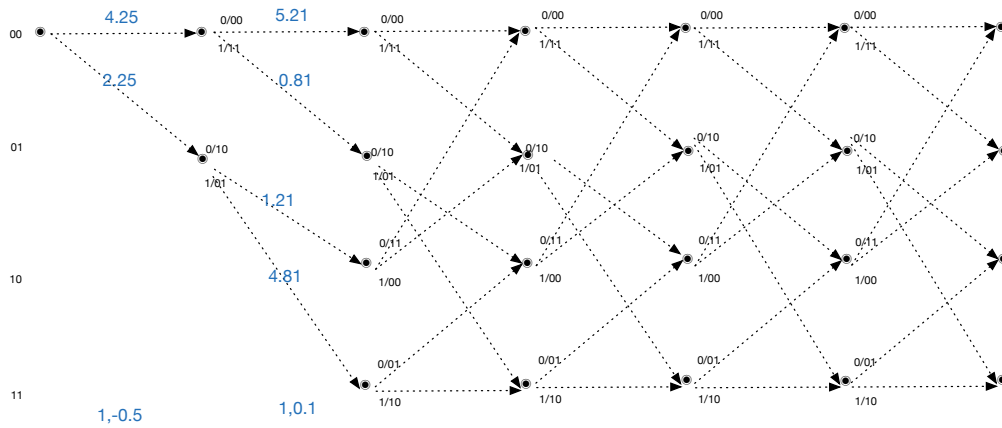


Figura 4.22: Calculamos la distancia  $d^2$  de  $(1,0,1)$  a  $(0,0)$ ,  $(1, 1)$ ,  $(1,0)$ ,  $(0,1)$ . Y de igual manera que la decisión fuerte, vamos acumulando estas distancias y vamos podando las ramas con mayor distancia, la decisión es tomada hasta el final.

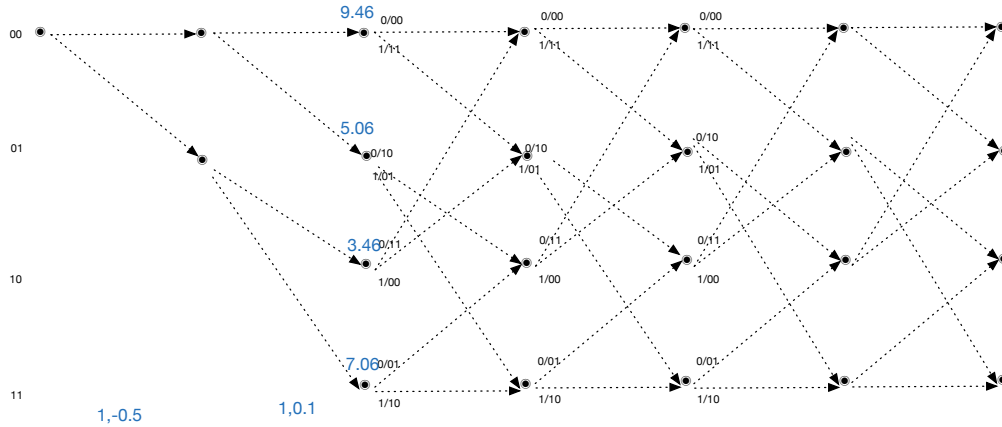


Figura 4.23: Se marcan en azul las distancias que se van acumulando y de igual manera que en la decisión fuerte, se irán podando las ramas con mayor distancia.

El mensaje enviado fue  $v = 11\ 10\ 00\ 01\ 01\ 11$  y el mensaje original

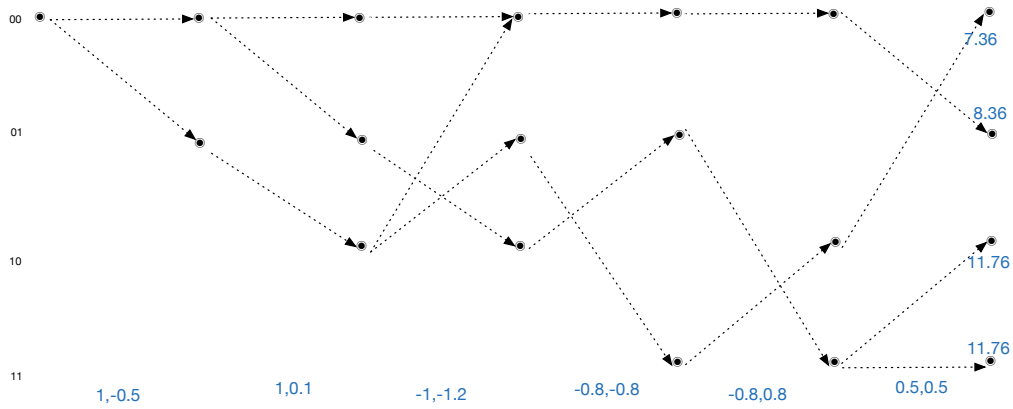


Figura 4.24: Calculando todas las distancias y podando las ramas necesarias obtenemos este diagrama, con distancia mínima 7.36.

$v = 101100$ , el mecanismo es exactamente el mismo que el de la decisión fuerte, la diferencia es la métrica usada.

La decisión suave usa la máxima verosimilitud, para esto se calcula la probabilidad condicional:

$$p(r|v),$$

donde

$$r(i) = (y_1(i), \dots, y_c(i)), \quad i = 0, 1, \dots, l + m - 1,$$

$$p(r|v) = \prod_{i=0}^{l+m-1} \prod_{j=1}^c p(r_j(i)|v_j(i)).$$

Generalmente  $y_j(i)$  se toma en el conjunto  $\{0_1, \dots, 1_4\}$ , cada uno de estos representa una cuantización binaria de un intervalo, consultar [3]. Cada elemento del conjunto  $\{0_1, \dots, 1_4\}$  tiene asignada una probabilidad y esta representa un diferente camino.

$p(r|v)$  se calcula para cada rama, dependiendo lo que se envió y lo que se recibió varía la probabilidad, únicamente son ocho diferentes probabilidades las que hay que calcular, además recordemos que es un canal simétrico el que se usa para la transmisión.

Posteriormente a las probabilidades se les aplica el logaritmo natural,  $\ln$ ,

es decir,  $\ln(p(r_i|v_i))$  y se multiplica por una constante conveniente y posteriormente se redondea. El método de acumulación y corte de ramas es exactamente el mismo que en la distancia cuadrada de la decisión fuerte. Para ver los detalles de esta probabilidad y cómo se obtiene, véase [3] y [8]. En el ejemplo anterior usamos una equivalencia a la probabilidad condicional.

El algoritmo de Viterbi está ligado estrechamente a los modelos de Markov y la decodificación de códigos de convolución con la decisión suave es basado en un modelo oculto de Markov. Los modelos de Markov gráficamente tienen asociados diagramas de trellis y de estados, como el caso de los códigos de convolución, para mayor información consúltese [3], [13], [22].

Como mencionamos al inicio de este trabajo, los códigos de convolución han sido abordados desde un punto de vista útil y práctico para su implementación y la parte de decodificación no es la excepción. El interés central de esta tesis no ha sido la decodificación, debido a esto no nos introdujimos a estas relaciones y trasfondos.

## 4.4. Códigos de convolución MDS

Existen diversos tipos de códigos, y dependiendo del tipo de información que se desea tratar se debe elegir el código adecuado. Algunos códigos sobresalen debido a sus propiedades, características y múltiples aplicaciones, por ejemplo los códigos Reed-Solomon (RS), los Reed-Muller (RM), Golay. En particular los códigos RS son códigos cíclicos y poseen varias propiedades interesantes, la sola ciclicidad los convierte en códigos atractivos y adecuados para las aplicaciones, ya que no resultan costosos operacionalmente.

Una propiedad de gran importancia de los códigos RS es que alcanzan la mayor distancia mínima posible, en general, un  $[b, c]$  código lineal  $C$  bajo un campo  $\mathbb{F}$  tiene distancia  $d \leq c - b + 1$ , cuando se cumple la igualdad diremos que son códigos de distancia máxima separable (Maximal Distance Separable, MDS), véase [20]. Los códigos MDS son códigos deseables debido a su excelente capacidad de corrección de errores.

Primero, un par de preguntas que podemos hacernos es ¿Cómo construimos un buen código de convolución?, ¿Qué características debe tener el codificador? Como en todo código lineal un buen código es el que tiene la capacidad de corregir más errores y en el caso de los códigos de convolución esta depende directamente de la distancia libre. Los códigos de convolución son códigos con muchas aplicaciones debido a sus propiedades, si estos fueran códigos MDS serían aun más útiles y llamativos, de lo que ya son, no es tan descabellado pensar en que es posible construir un código de convolución que alcance la distancia máxima.

A través de un código RS que es un código MDS podemos construir un código de convolución MDS con tasa  $R = b/c$ ,  $b \leq c$ . Véase *Maximum distance separable convolutional codes, construction and decoding* [19] que muestra como a partir de códigos Reed-Solomon que son códigos MDS, pueden generarse códigos de convolución que heredan la generosa propiedad MDS.

Aunque falta mucho por tratar sobre los códigos de convolución, existen propiedades que nos gustarian que estos tuviesen, lo que nos lleva a pensar que existen relaciones que se pueden seguir trabajando, relaciones con otro tipo de códigos, construcción de códigos de convolución a partir de estos otros códigos, incluso ver la posibilidad de hacer códigos de convolución ciclicos.

# Bibliografía

- [1] McEliece, R. J. *The Theory of Information and Coding*. Cambridge University Press; Student ed. edition. 2004.
- [2] Thamer, H. *Convolution Codes*. Information Theory 4th Class in Communications. 2008.
- [3] Huffman, W. C. and Pless, V. *Convolutional codes. Fundamentals of Error-Correcting Codes*. Cambridge University Press. 2010.
- [4] Roth, R. *Introduction to Coding Theory*. Cambridge University Press. 2006.
- [5] Lin, S. and Costello, D. J. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall. 1983.
- [6] Morelos-Zaragoza, R. H. *The Art of Error Correcting Coding* segunda edición. John Wiley and Sons. 2006.
- [7] Johannesson, R. and Zigangirov, K. Sh. *Fundamentals of Convolutional Coding*. Wiley-IEEE Press. 1999.
- [8] Willems, F. M. *Information and communication theory: Communication theory*. Spring. 2010.
- [9] Elias, P. *Coding for Noisy Channels*. Information Theory, 3rd London Symp. 1955.
- [10] Zaldívar, F. *Teoría de Galois*. Anthropos Barcelona. 1996.
- [11] Massey, J. L. and Sain, M. K. *Inverses of linear sequential circuits*. IEEE Trans. Comput. 1968.



- [12] Viterby, A. *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*. IEEE Transactions on Information Theory; vol 13. 1967.
- [13] K. Parhi, K. and Nishitami, T. *Digital Signal Processing for Multimedia Systems*. CRC Press. 1999.
- [14] Castiñeira, J. and Guy, P. *Essentials of error-control coding*. John Wiley & Sons, Ltd. 2006.
- [15] Richardson, T. *Modern coding theory*. Cambridge University Press. 2008.
- [16] Piret, F. *Convolutional Codes, an Algebraic Approach*. The MIT Press. 1988.
- [17] Moon, T. k. *Error Correction Coding: Mathematical Methods and Algorithms*. John Wiley & Sons, Inc. 2005.
- [18] Carlson, A. and Crilly, P. B. *Comunication Systems*, quinta edición. McGraw-Hill Education. 2009.
- [19] Smarandache, R. *Maximum distance separable convolutional codes, construction and decoding*. University of Notre Dame. 2000.
- [20] MacWilliams, F. J and Sloane, N. J. *The Theory of Error Correcting Codes*. North-Holland mathematical library; vol. 16. 1988.
- [21] Forero V, M. G. y Arias C, E. *Estudio del Efecto de las Máscaras de Convolución en Imágenes Mediante el Uso de la Transformada de Fourier*. Revista Ingeniería e Investigación No. 48. 2001.
- [22] Mendel, J. *Maximum-Likelihood Deconvolution: A Journey into Model-Based Signal Processing*. Springer. 1990.
- [23] Larsen, K. *Short convolutional codes with maximal free distance for rates  $1/2$ ,  $1/3$ , and  $1/4$* . IEEE Transactions on Information Theory; vol. 19. 1973. pág.
- [24] Lazcano, S. *Algoritmos iterativos de control de errores para sistemas de transmisión de información con razón señal-ruido en la región de piso de ruido*. Universidad Nacional Autónoma de México. 2011.

- [25] Bocharova, I. E. Hug, F. Johannesson, R. and Kudryashov, B.D. *Dual convolutional codes and the MacWilliams identities*. Problems of Information Transmission; vol. 48. 2012.
- [26] Tapia-Recillas, H. *Acerca de códigos y criptografía*. Universidad Autónoma Metropolitana-Iztapalapa, ContactoS 90. 2013.
- [27] Hirschma, I. I. and Widder, D. V. *The Convolution Transform*. Dover Publications Inc. 2005.
- [28] Hamming, R. W. *Error detecting and error correcting codes*. Bell System Tech Journal; vol. 29. 1950.
- [29] Reed, I. S. and Solomon, G. *Polynomial codes over certain finite fields*. Journal of the Society for Industrial and Applied Mathematics (SIAM); vol. 8. 1960.
- [30] García-Planas, M.I. y Tarragona, S. *Códigos de convolución desde el punto de vista de teoría de control. Análisis de la observabilidad*. Ciber, Revista Hispánica de Tendencias en Ciberseguridad; vol. 1. UPC. 2014.