Introducción a la Teoría de Códigos

M.A. García, L. Martínez, T. Ramírez

Facultad de Ciencia y Tecnología. UPV/EHU

Resumen Teórico
Tema 2: NOCIONES BÁSICAS DE LA TEORÍA DE CÓDIGOS

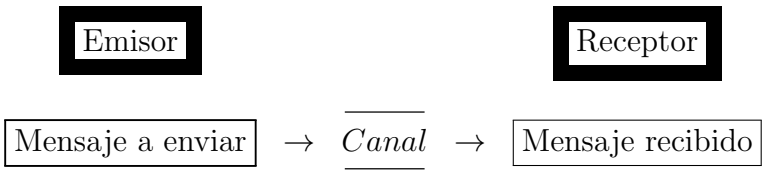
Mayo de 2017

Tema 2

Nociones básicas de la Teoría de Códigos

1 Introducción: El problema de la transmisión de la información

En la transmisión de la información nos encontramos con que un emisor manda un mensaje a través de un canal para que le llegue al receptor. Esto es, podemos representar los elementos que tenemos en el proceso de transmisión de una información mediante el siguiente esquema:



Sin embargo, durante la transmisión de la información puede haber problemas debido a interferencias que se produzcan en el canal. Estas interferencias pueden traducirse en dos situaciones diferentes:

1. Al receptor no le llega el mensaje que le ha enviado el emisor, sino que reciba un mensaje diferente del original por haber habido una mala transmisión.
2. El mensaje enviado ha sido interceptado por alguien que no es el destinatario final y este usurpador ha manipulado o ha hecho un uso indebido del mensaje captado.

Así, estas situaciones diferentes que se pueden darse sobre el canal usado nos llevan a realizar una clasificación del canal en dos tipos diferenciados:

1. Canal poco fiable: Hay "ruido" en el canal que dificulta la transmisión. Un ejemplo de este tipo de canal se tiene cuando intentamos leer un CD que presenta huellas de dedos que obstaculizan el acceso a la información grabada en él o cuando recibimos la señal de un satétite muy alejado de la Tierra, por lo que ésta llega muy debilitada, y no sabemos el valor exacto que se ha emitido.

2. Canal poco seguro: El canal que transmite la información soporta el ataque de espías que pueden manipular la información que transcurre por él o hacerse con ella sin ser el destinatario final. Por ejemplo, estaríamos ante un canal poco seguro cuando al hacer una compra por Internet con VISA, el ordenador que utilizamos para enviar la información de nuestra tarjeta tiene instalado un programa espía que captura los datos de nuestra tarjeta y se los pasa a otra persona para que haga otras compras que no hemos autorizado con nuestros datos, ó cuando el que intercepta el mensaje lo manipula antes de que continúe viajando por la red alterando el significado del mismo (p.e., hemos autorizado un pago de 30 euros a un vendedor X y el que lo altera indica al banco que autorizamos un cargo a nuestra cuenta de 300 euros, 30 para el vendedor X y 270 para abonar a Y).

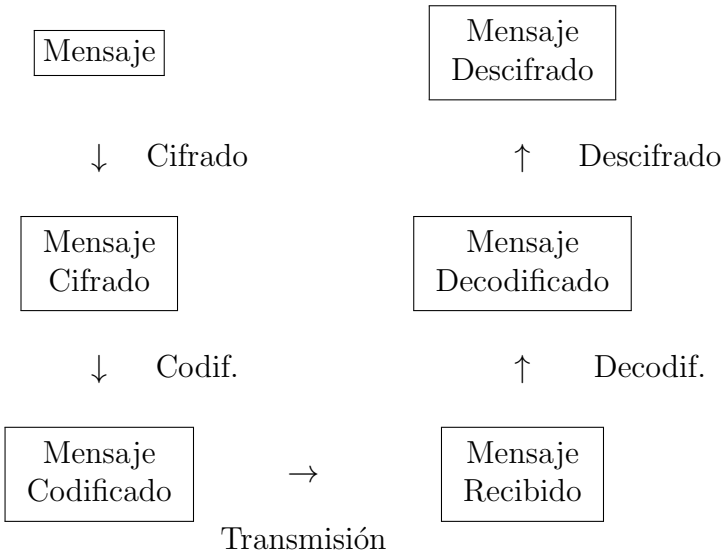
En cualquiera de los dos casos, es necesario disponer de herramientas que nos ayuden a solventar, en la medida de lo posible, este problema de interferencias en el canal. De hecho, las Matemáticas nos permiten solucionar en muchos casos esta dificultad y, tras un proceso más o menos complejo que debe aplicarse a la información que va a ser enviada por el canal, es posible "recuperar" el mensaje inicial (en el caso de canales poco fiables) o "esconder" la información que viaja por el canal, de forma que cualquiera que la intercepte no sepa de qué se trata (cuando el canal es poco seguro). Obviamente, según el tipo de interferencias que podamos sufir en el canal deberemos adoptar acciones diferentes. Así,

1. Si el canal es poco fiable, usaremos los llamados **códigos detectores y correctores de errores** que nos permiten al recibir un mensaje conocer si se ha producido algún fallo en la transmisión (esto es, detecta los errores producidos) y, en ciertos casos, incluso puede saberse cuál fue el mensaje inicial enviado. En esencia, los códigos detectores y correctores añaden información a la inicial, conociendose este proceso como codificación del mensaje, de forma que con la información extra añadida sepamos si se ha producido algún error en el mensaje recibido y, si es así, aplicarle un proceso llamado de decodificación, que permite recuperar (si es posible) el mensaje enviado.

2. Si el canal es poco seguro, la herramienta matemática de la que disponemos es el uso de los conocidos como **sistemas criptográficos**, que antes de enviar la información, la transforman en otra sin sentido para el que pretende hacerse con ella de forma ilegal y al ser recibida por el receptor la información transformada, éste revierte la transformación realizada para obtener el mensaje real que le quería mandar el emisor. El proceso de modificar la información para que carezca de sentido para el que escucha sin ser el receptor se conoce como

proceso de encriptado o cifrado y la acción de revertir el mismo, que lo hace el receptor, es el proceso de desencriptado o descifrado.

Hay veces que por las características de la información a enviar y del canal a utilizar será preciso realizar una combinación de ambas herramientas: en primer lugar se esconde la información, esto es, se cifra. Luego se le añade información extra, es decir, se codifica. A continuación se pasa por el canal y el receptor aplica primero un proceso de decodificación y al mensaje resultante uno de descifrado para llegar al mensaje original que deseaba hacerle llegar el emisor. Si queremos resumir esquemáticamente de este proceso, podemos usar el siguiente diagrama:



En cualquier caso, tanto al usar códigos detectores y correctores de errores como sistemas criptográficos, debemos ser especialmente cuidadosos en la elección del tipo de código o sistema a utilizar: debe ser acorde a las posibles interferencias que soporte el canal. Por ejemplo, si queremos mandar una información a través de un canal no muy fiable del que sabemos que cada 10 dígitos que pasamos por él se produce un error y queremos solventar este problema, de entre los códigos correctores de errores deberemos elegir uno que se ajuste a lo que nosotros necesitamos. Así, sería factible elegir uno que nos permita corregir uno o dos errores en cada bloque de 10 dígitos, pero no uno que permita corregir un error producido cada 100 datos (sería escaso) u otro que nos permita corregir uno de cada dos (podría ser demasiado costoso). Del mismo modo, cuando debamos trabajar con un canal poco seguro, deberemos valorar el tiempo y coste necesarios de los procesos de cifrado /descifrado de mensaje para adecuarlos al canal que usamos.

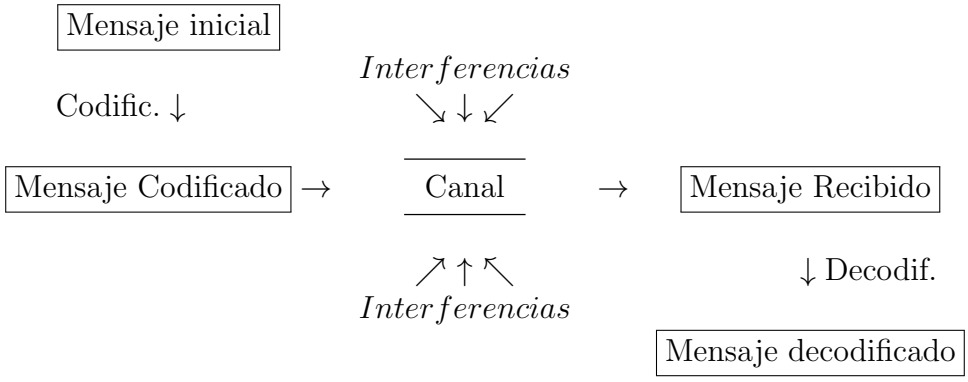
En este curso nos centraremos en estudiar algunos de los códigos detectores y correctores de errores, que son estudiados por la parte de las Matemáticas conocida como Teoría de Códigos, mientras que el estudio de los sistemas criptográficos, de los que se ocupa la Criptografía, se dejará para cursos posteriores. Asimismo, dentro del estudio de los códigos detectores y correctores de errores nos centraremos en analizar las propiedades y manejo de los llamados códigos lineales, que son los más sencillos

de trabajar. El estudio de otros tipos de códigos correctores de errores, como por ejemplo, los algebro-geométricos que usan curvas elípticas en sus definiciones, transciende el objetivo de este curso que lo que pretende es exponer de forma rigurosa, desde el punto de vista matemático, las características fundamentales de los códigos lineales, proporcionando de este modo una primera aproximación a la matemática más sencilla que está detrás de la Teoría de Códigos. No obstante, en este capítulo nos preocupamos por formalizar los conceptos más importantes sin restringirnos, en muchos del ellos, al marco exclusivo de los códigos lineales, sino que estas definiciones son válidas para los códigos detectores y correctores en general.

2 Códigos correctores y detectores de errores

Como se ha indicado en la sección anterior, el problema con el que nos enfrentamos es que deseamos enviar un mensaje a un receptor a través de un canal que puede ser poco fiable, esto es, el mensaje que le llega no tiene que ser el emitido y puede contener errores de transmisión. Lo que nos interesa es que el receptor, tras recibir el mensaje, pueda detectar si el mensaje que le ha llegado es el original y si no lo es, intentar recuperar el mensaje inicial.

De forma esquemática, resumimos las fases que componen todo el proceso como sigue:



Buscamos que el mensaje decodificado sea lo más parecido posible al mensaje codificado. Es más, nos interesa haber elegido bien el código de acuerdo al canal utilizado para poder tener una probabilidad muy alta de que el mensaje codificado y el decodificado sean iguales para que pueda el receptor deducir cuál era el mensaje inicial a partir del decodificado, sabiendo el código que se ha empleado.

En el estudio de los códigos correctores y detectores de errores usaremos algunas definiciones básicas que conviene que precisemos a qué se refieren cuando las empleemos. Por ello, a continuación vamos a formalizar matemáticamente conceptos que aparecen de forma frecuente, como los siguientes:

1. **Alfabeto**: Es un conjunto finito de elementos, llamados letras. Así, matemáticamente un alfabeto es un conjunto

$$A = \{a_1, \ldots, a_m\},$$

siendo cada $a_i \in A$ una letra del alfabeto A y $m \in \mathbb{N}$ el cardinal del alfabeto A.

2. **Palabra**: Fijado un alfabeto A, una palabra sobre el alfabeto A es un elemento formado por la concatenación de un número finito de letras de A. Salvo que se indique lo contrario, denotaremos las palabras por letras minúsculas del alfabeto latino en negrita. Por ejemplo, $\mathbf{x}= x_1 \ldots x_n$, donde $x_i \in A$ para $i = 1, \ldots, n$ es una palabra sobre A que se ha formado al poner de forma consecutiva las letras x_1,...,x_n. En muchos textos a estas palabras se les denomina "*palabras positivas*" para distinguirlas de aquellas en las que las letras x_i pueden venir afectadas del exponente -1.

3. Ω_A: Es el semigrupo libre con base A, esto es, el conjunto de todas las palabras (positivas) que se pueden formar con las letras del alfabeto A.

4. **Código**: Es un conjunto de palabras sobre un mismo alfabeto, es decir, $C \subseteq \Omega_A$. A los elementos de C se les llama palabras del código C.

5. **Longitud de una palabra** $l(\mathbf{x})$: Número de letras que tiene una palabra. Si $\mathbf{x}= x_1 \ldots x_n$, entonces $l(\mathbf{x}) = n$.

6. **Igualdad de palabras**: Dos palabras sobre el mismo alfabeto A $\mathbf{x}$ e $\mathbf{y}$ son iguales si tienen la misma longitud y coinciden los valores de cada letra en cada posición. Esto es,

$$\mathbf{x} = \mathbf{y} \Longleftrightarrow \mathbf{x} = x_1 \cdots x_n, \quad \mathbf{y} = y_1 \cdots y_n \text{ y } x_i = y_i, \forall i = 1, \ldots, n.$$

7. **Código de bloque**: Dado un código $C \subseteq \Omega_A$, se dice que C es un código de bloque si todas las palabras de C tienen la misma longitud n. También se suele llamar código de longitud n.

8. **Distancia de Hamming**: Dadas $\mathbf{x}$ e $\mathbf{y}$ dos palabras de la misma longitud se llama distancia de Hamming entre $\mathbf{x}$ e $\mathbf{y}$ al número de componentes en que difieren ambas palabras. Esto es, si $\mathbf{x}= x_1 \ldots x_n$ e $\mathbf{y}= y_1 \ldots y_n$,

$$d(\mathbf{x}, \mathbf{y}) = |\{i \in \{1, \ldots, n\} \,|\, x_i \neq y_i\}|,$$

donde $d(\mathbf{x}, \mathbf{y})$ denota la distancia de Hamming entre las palabras $\mathbf{x}$ e $\mathbf{y}$).

9. **Distancia mínima de un código de bloque** C **de longitud** n: Dado $C \subseteq \Omega_A$ tal que C es un código de bloque, se llama distancia mínima de C a la menor de las distancias entre palabras diferentes de C, esto es,

$$d(C) = \min\{d(x_1 \ldots x_n, y_1 \ldots y_n) \,| \quad x_1 \ldots x_n, y_1 \ldots y_n \in C, \; x_1 \ldots x_n \neq y_1 \ldots y_n\}$$

10. **Peso de una palabra**: Si $\mathbf{x} \in \mathbb{F}_q^n$, se llama peso de $\mathbf{x}$, y se denota por $w(\mathbf{x})$, al número de componentes no nulas de $\mathbf{x}$.

11. **Peso mínimo de un código**: Si $C \subseteq \mathbb{F}_q^n$, se llama peso mínimo de C, y se denota por $w(C)$, a

$$w(C) = \min\{w(\mathbf{x}) \,|\, \mathbf{x} \in C - \{\mathbf{0}\}\}.$$

12. **Detectar hasta t errores**: Saber que se ha producido t fallos en la transmisión, esto es, que si trabajamos con un código C que detecta t errores, entonces cualquier palabra $\mathbf{y}$ que se obtenga de una palabra de $\mathbf{c} \in C$ cambiando el valor que figura en $\mathbf{c}$ en a lo sumo t posiciones, resulta que $\mathbf{y} \notin C$.

13. **Corregir hasta t errores**: Ser capaz de recuperar la palabra original cuando se han producido t fallos en la recepción de la misma, es decir que si $\mathbf{y} \notin C$ se ha obtenido a partir de $\mathbf{c} \in C$ cambiando el valor que figura en $\mathbf{c}$ en a lo sumo t posiciones, entonces la única palabra de C que dista de $\mathbf{y}$ a lo sumo t es la propia $\mathbf{c}$. En este caso diremos que $\mathbf{c}$ es la decodificación (única) de $\mathbf{y}$.

14. **Principio de máxima verosimilitud**: En las transmisiones es más probable que se produzca siempre el menor número de fallos posible. Seguiremos este principio en todo el curso.

Ejemplos

1. El cuerpo finito $\mathbb{F}_q$, con $q = p^t$, siendo p número primo y $t \in \mathbb{N}$, es un ejemplo de alfabeto que utilizaremos en este curso. Si $q = 2$, esto es, el alfabeto es $\mathbb{F}_2 = \{0, 1\}$ y tomamos un código C de $\Omega_{\mathbb{F}_2}$, se dice que el código C es binario.

2. $\mathbf{x}$=01010 es una palabra de longitud 5 usando como alfabeto $\mathbb{F}_2$ o en general $\mathbb{F}_q$.

3. Si tomamos $A = \mathbb{F}_2 = \{0, 1\}$ y

$$C = \{\mathbf{x} \in \Omega_A \mid \mathbf{x} \text{ contiene un número par de “1”}\},$$

entonces C es un código binario. Una palabra de C es $\mathbf{x}$= 011001010 y la longitud de $\mathbf{x}$ es 9. C no es un código de bloques porque las palabras $\mathbf{y}$= 011 y $\mathbf{z}$= 1010 están en C y tienen longitud distinta. Por otro lado, la palabra $\mathbf{y}$= 011001000 $\in \Omega_A$ no pertenece a C. Otra forma de caracterizar los elementos de C es la siguiente:

$$C = \{\mathbf{x} = x_1 \dots x_m \in \Omega_A \mid m \in \mathbb{N}, \sum_{i=1}^{m} x_i \equiv 0 \bmod 2\}.$$

4. Los siguientes ejemplos son ejemplos conocidos de códigos de bloque:

- Código ASCII: Es un código que se emplea para transmitir la información desde el teclado del ordenador a la CPU. A cada letra y símbolo del teclado se le asigna un número entre 0 y 127. Este número se representa en el sistema binario como una 7-tupla, que se completa con un octavo dígito (0, ó 1) de forma que la 8-tupla resultante tenga un número par de "1". Por ejemplo a la letra A le corresponde el 65 que se representa por la 7-tupla 1000001 y se añade a esta 7-tupla un "0" de forma que la 8-tupla resultante tenga un número par de 1, dando lugar en este caso a la 8-tupla 10000010 que es la que correspondería a la letra A.
- Código ISBN: Es un código que identifica de forma única a los libros publicados. A cada libro se le asocia un número de diez cifras $a_1 \dots a_{10}$ en la que las nueve primeras, que toman valores entre 0 y 9, dan información sobre el libro (país, editorial, título,...) y la última se obtiene exigiendo que $\sum_{i=1}^{10} i a_i \equiv 0 \bmod 11$. Si a_{10} toma el valor 10, se sustituye éste por X, por lo que $a_{10} \in \{0, 1, \dots, 9, X\}$. Por ejemplo, una palabra que forma parte de este código ISBN es 0-19-853803-0.
- Código EAN: Es un código que permite identificar de forma única a los productos que se vendenen Europa. A cada producto se le asocia un número de trece cifras $a_0 \dots a_{12}$, con $a_i \in \{0, 1, \dots, 9\}$ para $i = 0, 1, \dots, 12$, donde los 3 primeros dígitos representan el código del país en donde radica la empresa que lo comercializa, los 4 o 5 siguientes son el código de empresa, que identifica al propietario de la marca y es asignado por la asociación de fabricantes y distribuidores (AECOC). Los 5 o 4 siguientes son el código de producto y el último dígito (a_{12}) es un dígito, llamado de control, que se obtiene de forma que $3\sum_{i=0}^{5} a_{2i+1} + \sum_{i=0}^{6} a_{2i} \equiv 0 \bmod 10$.

3 Distancia de Hamming

Hemos introducido la distancia de Hamming que nos mide en cuántas componentes difieren dos palabras de la misma longitud. El utilizar el nombre de "distancia" está justificado desde el punto de vista topológico, ya que es fácil demostrar que:

Proposición 3.1 *Sea $A = \{a_1, \dots, a_m\}$ un alfabeto, T_n el conjunto de las palabras sobre el alfabeto A de longitud n, esto es, $T_n = \{x_1 \dots x_n \,|\, x_i \in A, i = 1, \dots, n\}$ y $d : T_n \times T_n \to \{0, 1, 2, \dots, n\}$ la aplicación definida por*

$$d(x_1 \dots x_n, y_1 \dots y_n) = |\{i \in \{1, \dots, n\} \,|\, x_i \neq y_i\}|,$$

para todo $x_1 \dots x_n, y_1 \dots y_n \in T_n$. Entonces, (T_n, d) es un espacio métrico.

Al ser (T_n, d) un espacio métrico tiene sentido construir sus bolas cerradas. Recordemos que, dado $r \in \mathbb{N} \cup \{0\}$ y $\mathbf{x} \in T_n$, se llama **bola cerrada de centro x y radio**

r al conjunto

$$\overline{B}(\mathbf{x}, r) = \{\mathbf{y} \in A^n \mid \mathrm{d}(\mathbf{x}, \mathbf{y}) \leq r\}.$$

Obviamente, si $r \geq n$, $\overline{B}(\mathbf{x}, r) = T_n$. Más aún, es sencillo determinar el número de elementos de una bola cerrada cuando $r = 0, 1, \ldots, n$, tal y como nos lo indica el siguiente resultado:

Lema 3.2 *Sea $A = \{a_1, \ldots, a_m\}$ un alfabeto, $\boldsymbol{x} = x_1 \ldots x_n \in T_n$ y $0 \leq r \leq n$. Entonces,*

$$|\overline{B}(\boldsymbol{x}, r)| = \sum_{i=0}^{r} \binom{n}{i} (m-1)^i.$$

El Lema anterior va a ser útil para demostrar la conocida Cota de Hamming:

Proposición 3.3 (Cota de Hamming) *Sea $C \subseteq T_n$ un código de longitud n sobre el alfabeto $A = \{a_1, \ldots, a_m\}$ con distancia mínima d. Entonces,*

$$|C| \sum_{i=0}^{\left[\frac{d-1}{2}\right]} \binom{n}{i} (m-1)^i \leq m^n.$$

Para probarlo simplemente hay que darse cuenta que

$$\overline{B}(\mathbf{c}, \left[\frac{d-1}{2}\right]) \cap \overline{B}(\mathbf{c}', \left[\frac{d-1}{2}\right]) = \emptyset, \quad \forall \mathbf{c}, \mathbf{c}' \in C \text{ tales que } \mathbf{c} \neq \mathbf{c}'$$

y usar el Lema 3.2.

Relacionamos ahora el número de errores que puede detectar y corregir un código de bloque de longitud n con su distancia mínima. Se puede demostrar que:

Proposición 3.4 *Sea $C \subseteq T_n$ un código de longitud n sobre el alfabeto A con distancia mínima d. Entonces*

1. *C puede detectar hasta $d-1$ errores.*
2. *C puede corregir hasta $\left[\frac{d-1}{2}\right]$ errores.*

Ejemplo Consideramos el código $C \subseteq \mathbb{F}_2^7$ cuyas palabras son

$$\begin{aligned} C \quad = \{&0000000, 0110100, 0011010, 0001101, \\ &1000110, 1001011, 1011100, 0010111, \\ &1010001, 1110010, 0111001, 1111111, \\ &0101110, 1101000, 0100011, 1100101\}. \end{aligned}$$

Este código C tiene distancia mínima 3, ya que haciendo las distancias de pares de palabras distintas de C, se observa que el valor mínimo es este valor. Si le aplicamos ahora la Proposición 3.4, deducimos que puede detectar hasta dos errores y corregir uno. Por ejemplo, si recibimos la palabra $\mathbf{x} = 1000111$ vemos que no está en el código, por tanto se ha(n) producido error(es) en la transmisión. Pero nos fijamos que 1000110 sí pertenece al código y difiere solo en un dígito con $\mathbf{x}$, luego podemos deducir que es la que se ha emitido cuando hemos recibido $\mathbf{x}$, ya que este código corrije un error. De hecho, si calculamos la distancia que hay entre $\mathbf{x}$ y las diferentes palabras de C, vemos que la única palabra de C que dista 1 de $\mathbf{x}$ es 1000110. Más aún, C alcanza la cota de Hamming ya que

$$|C| \sum_{i=0}^{\left[\frac{d-1}{2}\right]} \binom{n}{i} (m-1)^i = 2^4 \sum_{i=0}^{1} \binom{7}{i} 1^1 = 2^4 2^3 = 2^7.$$

Estudiamos los códigos que alcanzan la Cota de Hamming en el siguiente apartado y veremos qué significa alcanzar esta cota en términos de poder determinar la palabra emitida cuando recibimos una palabra de $\mathbb{F}_2^7$.

4 Códigos perfectos

Un código $C \subseteq T_n$ con distancia mínima d que alcance la Cota de Hamming se dice que es un **código perfecto**. Esto es, si $C \subseteq T_n$ es un código de bloque sobre el alfabeto $A = \{a_1, \ldots, a_m\}$ de distancia mínima d, entonces se dice que C es perfecto, si cumple $|C| \sum_{i=0}^{\left[\frac{d-1}{2}\right]} \binom{n}{i} (m-1)^i = m^n$.

En el último ejemplo del apartado anterior, hemos comentado que el código binario

$$\begin{aligned} C \ = \{&0000000, 0110100, 0011010, 0001101, \\ &1000110, 1001011, 1011100, 0010111, \\ &1010001, 1110010, 0111001, 1111111, \\ &0101110, 1101000, 0100011, 1100101\}. \end{aligned}$$

era un código perfecto y habíamos visto que si recibiamos la palabra $\mathbf{x} = 1000111$, podíamos conocer la palabra emitida que le correspondía. En la siguiente proposición vemos que esta característica la tienen todos los códigos perfectos:

Proposición 4.1 *Sea $C \subseteq T_n$ un código de bloque de longitud n sobre un alfabeto A de distancia mínima d. Entonces, C es perfecto si y, solo si, $\dot{\bigcup}_{\boldsymbol{c} \in C} \overline{B}(\boldsymbol{c}, \left[\frac{d-1}{2}\right]) = T_n$.*

La interpretación del resultado anterior es la misma que la del último ejemplo del apartado anterior: cada elemento de T_n se encuentra en una única bola cerrada de centro $\mathbf{c}$ y radio $\left[\frac{d-1}{2}\right]$ y, como las bolas son disjuntas, esto implica que cada elemento de T_n admite decodificación única, que es precisamente el centro $\mathbf{c}$ de la bola a la que pertenece el elemento de T_n que hemos considerado.

En el siguiente resultado vemos que la distancia mínima de un código perfecto no puede tomar cualquier valor:

Proposición 4.2 *Si $C \subseteq T_n$ es un código perfecto, su distancia mínima es un número impar.*

Como consecuencia de la Proposición anterior, podemos deducir la no existencia de códigos perfectos con distancia mínima un número par.

Ejemplo Consideramos de nuevo el código binario de longitud 7 cuyas palabras son:

$$\begin{aligned} C \ = \{&0000000, 0001101, 1111111, 1110010,\\ &1101000, 1000110, 0010111, 0111001,\\ &0110100, 0100011, 1001011, 1011100,\\ &0011010, 1010001, 1100101, 0101110\} \end{aligned}$$

Como ya hemos comentado, este código tiene distancia mínima 3. Además, sabemos que si $\mathbf{c}$ y $\mathbf{c}'$ son dos palabras distintas de C, se tiene que $\overline{B}(\mathbf{c},1) \cap \overline{B}(\mathbf{c}',1) = \emptyset$ y podemos calcular para cada $\mathbf{c} \in C$ el cardinal de $\overline{B}(\mathbf{c},1)$ que es 8. Así que C es un código perfecto ya que

$$\dot{\bigcup}_{\mathbf{c}\in C} \overline{B}(\mathbf{c},1) = \mathbb{F}_2^7.$$

Esto implica que cada palabra de T_n admite, por tanto, decodificación única.

En el Tema 3 estudiaremos una familia de códigos perfectos: los códigos de Hamming, que son códigos de bloque sobre el cuerpo finito $\mathbb{F}_q$ de distancia mínima 3 y longitud $\frac{q^r-1}{q-1}$, siendo r un número natural y q potencia de un primo. De hecho, el código binario del ejemplo anterior va a ser un código de tipo Hamming.

5 Códigos equivalentes

Cuando consideramos dos códigos de bloque C_1 y C_2 con la misma longitud n sobre el mismo alfabeto A nos interesará no solo que C_1 y C_2 sean diferentes como subconjuntos de T_n, sino que también nos preocuparemos de que las capacidades correctoras de ambos u otras características intrínsecas a ellos sean diferentes. Por ello, vamos a introducir la definición de códigos equivalentes que va a recoger esta idea. Necesitamos primero introducir dos tipos de aplicaciones de T_n en sí mismo:

1. Sean $i, j \in \{1, \ldots, n\}$ con $i < j$. Se define

$$\begin{array}{rccc} \phi_{ij}: & T_n & \to & T_n \\ & x_1 \ldots x_n & \mapsto & x_1 \ldots x_{i-1} x_j x_{i+1} \ldots x_{j-1} x_i x_{j+1} \ldots x_n. \end{array}$$

   Es decir, ϕ_{ij} intercambia las letras que aparecen en las posiciones i y j de cada palabra $\mathbf{x} \in T_n$.

2. Dada una permutación $g : A \to A$, esto es, g es una aplicación biyectiva de A en sí mismo, y un índice $k \in \{1, \ldots, n\}$, se define

$$\begin{array}{rccc} \psi_{g,k}: & T_n & \to & T_n \\ & x_1 \ldots x_n & \mapsto & x_1 \ldots x_{k-1} g(x_k) x_{k+1} \ldots x_n. \end{array}$$

   Esto es, $\psi_{g,k}$ lo que hace es aplicar la permutación g a las letras que aparecen en la posición i-ésima.

Definición Sea A un alfabeto y $C_1, C_2 \subseteq T_n$. Se dice que C_2 es **equivalente** a C_1, si se puede obtener C_2 como la imagen de C_1 mediante una aplicación h, que es la composición de un número finito de aplicaciones del tipo ϕ_{ij} y $\psi_{g,k}$ para $g \in \Sigma_A$ y $i, j, k \in \{1, \ldots, n\}$.

Observamos que al ser tanto las aplicaciones del tipo ϕ_{ij} como $\psi_{g,k}$ biyectivas con inversa del mismo tipo es evidente que si C_2 es equivalente a C_1, entonces C_1 es equivalente a C_2. Por tanto, diremos simplemente que C_1 y C_2 son equivalentes. Además, si consideramos $\mathcal{T}_n$ el conjunto de los códigos de longitud n sobre el alfabeto A, se tiene que el ser equivalentes es una relación de equivalencia en $\mathcal{T}_n$, que denotamos por $\mathcal{R}$. Podemos calcular el conjunto cociente $\mathcal{T}_n/\mathcal{R}$ y la clase de un código $C \in \mathcal{T}_n$ vendrá dada por:

$$[C] = \{C_2 \mid C \mathcal{R} C_2\}.$$

Una propiedad interesante de los códigos equivalentes es la que relaciona las distancias mínimas de los códigos equivalentes y que la resumimos en la siguiente Proposición:

Proposición 5.1 *Sea $C_1 \subseteq T_n$ un código de longitud n sobre el alfabeto A y distancia mínima d. Si $C_2 \subseteq T_n$ es un código equivalente a C_1, entonces la distancia mínima de C_2 es también d.*

Para demostrar la Proposición anterior basta fijarse que si elegimos $\mathbf{x}, \mathbf{y} \in T_n$, se cumple

1. $d(\mathbf{x}, \mathbf{y}) = d(\phi_{ij}(\mathbf{x}), \phi_{ij}(\mathbf{y}))$
2. $d(\mathbf{x}, \mathbf{y}) = d(\psi_{g,k}(\mathbf{x}), \psi_{g,k}(\mathbf{y}))$

Por otro lado, si nos fijamos en la Proposición anterior, podemos deducir que una condición necesaria para que dos códigos de bloque de longitud n con alfabeto A sean equivalentes es que ambos tengan la misma distancia mínima.

Por otro lado, se puede demostrar la siguiente proposición:

Proposición 5.2 *Sea $C_1 \subseteq T_n$ un código de longitud n y $\boldsymbol{u} \in T_n$. Entonces, existe $C_2 \subseteq T_n$ equivalente a C_1 tal que $\boldsymbol{u} \in C_2$.*

Como consecuencia del resultado anterior, deducimos que

Corolario 5.3 *Sea $C_1 \subseteq \mathbb{F}_q^n$ un código de longitud n. Entonces, existe $C_2 \subseteq \mathbb{F}_q^n$ equivalente a C_1 tal que $00\ldots00 \in C_2$.*

Esto será interesante cuando deseemos construir un código de bloque de longitud n sobre $\mathbb{F}_q$ que contenga al $0\ldots0$ y sea equivalente a uno dado $C_1 \subseteq \mathbb{F}_q^n$, que no contiene al $0\ldots0$.