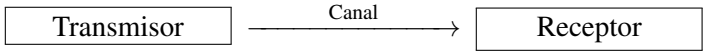
Teoría de códigos y Criptografía
Codificación de la información

Lorenzo Javier Martín García

20 de enero de 2022

1. Codificación de canal

La **Teoría de la Información** estudia el flujo de información desde un transmisor a un receptor, a través de un canal.



Como característica adicional, la información debe ser almacenada y tratada de tal manera que los papeles de receptor y emisor puedan ser intercambiados si fuera necesario.

Generalmente, la información que se transmite procede de una fuente que genera secuencias, s, de símbolos,

$$s = X_1 X_2 \cdots X_n.$$

Por ejemplo, X_n puede ser el n-ésimo símbolo de un mensaje o la salida de la n-ésima repetición de un experimento. En la práctica, esta secuencia siempre será finita, pero para propósitos teóricos, a veces es útil considerar sucesiones infinitas.

Supongamos que cada símbolo X_n es un elemento de un conjunto finito,

$$S = \{s_1, s_2, \ldots, s_q\},$$

al que llamaremos alfabeto fuente.

Por simplicidad, supondremos que la probabilidad, p_i, de que el n-ésimo símbolo de la secuencia sea s_i,

$$\Pr(X_n = s_i) = p_i,$$

permanece fija en el tiempo –es estacionaria– y depende solamente de i y no de la posición que ocupa en la cadena –no tiene memoria–. Así, diferentes símbolos pueden tener diferentes probabilidades de ocurrencia, pero no dependen de los símbolos precedentes en la secuencia. En cualquier caso, tiene que cumplirse

$$\forall i = 1, 2, \ldots, n \; p_i \geq 0 \quad \text{y} \quad \sum_{i=1}^{q} p_i = 1.$$

En términos estadísticos, S es una secuencia de n variables aleatorias independientes e idénticamente distribuidas.

En general, los símbolos fuente en su formato original no se pueden transmitir por el canal disponible, por lo que resulta necesario codificarlos.

Para codificar el alfabeto fuente se utiliza un conjunto finito,

$$T = \{t_1, t_2, \ldots, t_r\}$$

llamado alfabeto código, y que contiene r símbolos código. Evidentemente, estos símbolos dependen de la tecnología del canal de transmisión.

A los códigos que utilizan un alfabeto de codificación de r símbolos se les denomina códigos r-arios, por ejemplo, si $T = \{0,1\}$, el código es binario.

Una palabra código es una secuencia finita de elementos del alfabeto código.

Codificar consiste en asociar una palabra código a cada símbolo del alfabeto fuente para que pueda ser transmitido por un determinado canal, o por otras circunstancias.

Sabiendo cómo codificar todos los símbolos fuente, una palabra fuente se codifica como la secuencia de palabras código asociadas a los correspondientes símbolos fuente que la configuran, sin que haya separación entre ellas.

Un ejemplo de codificación es el código Morse en donde el alfabeto fuente lo constituyen las letras y los números, y el alfabeto código está formado por el punto, la raya (señal de triple duración que el punto) y la ausencia de señal (de duración un punto entre símbolos, tres puntos entre letras y tres rayas entre palabras) para separar símbolos, letras y palabras. Es conocido que la palabra SOS se codifica como ... — ...

Para simplificar la nomenclatura, y siempre que no se produzca confusión, utilizaremos 'palabra' para referirnos a las palabras código. Así una palabra w, es una secuencia finita de símbolos de T y su longitud es el número de símbolos del alfabeto código que la configuran, $\ell = |w|$.

El conjunto de todas las palabras se denota T^*, incluyendo la palabra vacía o de longitud cero. El conjunto de todas las palabras no nulas se denota T^+. Si $T^n = T \times T \times T \cdots \times T$ entonces

$$T^* = \bigcup_{n \geq 0} T^n \quad \text{y} \quad T^+ = \bigcup_{n>0} T^n.$$

Un código C es una función $C : S \longmapsto T^+$ inyectiva que a cada símbolo de S le asocia una palabra no nula formada por elementos de T.

La inyectividad asegura que cada símbolo de T se va a codificar de manera diferente a los demás, lo que puede permitir la decodificación. Si esta propiedad no se cumple, la decodificación produce ambigüedades e imposibilita la labor del receptor.

Muchas propiedades de los códigos sólo dependen de sus palabras y no de la función de codificación en sí, por lo que, en estos casos, se considera que el código es la imagen de la función C.

Si se define S^* de manera similar a T^*,

$$S^* = \bigcup_{n \geq 0} S^n \quad \text{y} \quad S^+ = \bigcup_{n>0} S^n,$$

se puede extender la función C a otra función entre S^* y T^*,

$$C^* : S^* \longmapsto T^*$$

de tal manera que cada palabra fuente se codifique como la secuencia de palabras asociadas a sus símbolos. En este esquema, aunque teóricamente no tiene que ser así, por motivos de coherencia la palabra fuente nula debe transformarse en la palabra código nula. La imagen de esta nueva función es el conjunto

$$\text{Img}(C^*) = \{w_{i_1} w_{i_2} \cdots w_{i_n} \in T^* : w_{i_j} \in C \; n \geq 0\}.$$

Si se denota por ℓ_i la longitud de las palabras w_i asociadas a cada uno de los q símbolos fuente, s_i, la longitud media del código C se define como

$$L(C) = \sum_{i=1}^{q} p_i \ell_i.$$

El objetivo de la teoría de códigos es construir códigos

- cuya decodificación sea fácil y no ambigua
- la longitud media sea lo más pequeña posible.

Ejemplo 1.1 *Por ejemplo, en las quinielas de fútbol se utilizan los símbolos* $S = \{1, X, 2\}$ *para representar la victoria del equipo local (1), el empate (X) y la victoria del equipo visitante (2).*

Si se quiere transmitir estos resultados de partidos de fútbol por un canal binario, habrá que codificar los elementos de S mediante los elementos del alfabeto $T = \{0, 1\}$.

Un posible código, $C' : S \longmapsto T^*$, *muy simbólico podría ser*

$$C'(1) = 10,\ C'(X) = 11,\ C'(2) = 01.$$

Los catorce resultados de una quiniela, $s = 1X112X112111X1$, *se codificarían como la palabra*

$$w_1 = 10111010011110100110101011 10$$

de longitud $\ell_1 = |w_1| = 28$.

Otro posible código, $C'' : S \longmapsto T^*$, *menos simbólico podría ser*

$$C''(1) = 0,\ C''(X) = 1,\ C''(2) = 00$$

en donde los resultados de la quiniela anterior se codificarían como la palabra

$$w_2 = 0100001000000010$$

de longitud $\ell_2 = |w_2| = 16$.

Un tercer código, $C''' : S \longmapsto T^*$ *podría ser*

$$C'''(1) = 0,\ C'''(X) = 10,\ C'''(2) = 11$$

en donde los resultados de la quiniela anterior se codificarían como la palabra

$$w_3 = 0100011100011000100$$

de longitud $\ell_3 = |w_3| = 19$.

Si por algún procedimiento se estableciera que la probabilidad de que el equipo local gane un partido es $p_1 = 0{,}6$, *de que empate es* $p_2 = 0{,}25$ *y de que pierda es* $p_3 = 0{,}15$, *las longitudes medias de los códigos* C', C'' *y* C''' *serían*

$$L(C') = 2 \cdot 0{,}6 + 2 \cdot 0{,}25 + 2 \cdot 0{,}15 = 2,$$
$$L(C'') = 1 \cdot 0{,}6 + 1 \cdot 0{,}25 + 2 \cdot 0{,}15 = 1{,}15,$$
$$L(C''') = 1 \cdot 0{,}6 + 2 \cdot 0{,}25 + 2 \cdot 0{,}15 = 1{,}4.$$

Cada uno de estos códigos tiene diferentes características que les convierten en apropiados o inapropiados para un determinado propósito.

2. Códigos decodificables de manera única

Un código C es decodificable de manera única si y solamente si a cada elemento de T^* le corresponde bajo C a lo sumo un elemento de S^*. Es decir, la función $C^* : S^* \longmapsto T^*$ es inyectiva, de tal manera que si $t = C^*(s) \in T^*$, entonces $s \in S^*$ es único. Para que el código sea decodificable de manera única, C^* aplicada a cada palabra fuente genera una palabra código diferente a las demás palabras generadas.

La inyectividad de la función $C : S \longmapsto T^*$ no asegura la inyectividad de la función $C^* : S^* \longmapsto T^*$ entendida como concatenación de las palabras código generadas por los símbolos fuente sin separaciones entre sí. Por ejemplo, el código C'' empleado en la codificación de resultados de los partidos de fútbol no es decodificable de manera única porque la palabra código 00 puede decodificarse como dos victorias locales o como una victoria visitante,

$$00 = C''(1)C''(1) \quad \text{o bien} \quad 00 = C''(2).$$

Es evidente que si en la cadena código se incorpora un símbolo de separación entre símbolos codificados, todas las cadenas serían decodificables de manera única, sin embargo la existencia de códigos decodificables de manera única sin utilizar símbolos de separación es una prueba de que no es necesario emplear una palabra código para especificar la separación entre cada símbolo, con la correspondiente reducción en el tamaño de la palabra código final.

Si un código es decodificable de manera única, cualquier palabra de C puede descomponerse de manera única como secuencia de palabras código. Esta característica es la base del siguiente teorema/definición.

Teorema 2.1 *Si u_i y v_j representan palabras código, $u_i = C(s)$ y $v_j = C(t)$ con $s,t \in S$, las dos condiciones siguientes son equivalentes:*

a) *El código C es decodificable de manera única.*

b) $\forall u_1u_2\cdots u_n,\ v_1v_2\cdots v_m \in C^*(S) \subset T^*\ \Big(u_1u_2\cdots u_n = v_1v_2\cdots v_m \Longleftrightarrow \big(n = m\ y\ \forall i = 1,2,\ldots,n\ v_i = u_i\big)\Big)$.

Una manera de utilizar implícitamente símbolos de separación es exigiendo que los símbolos fuente se codifiquen mediante palabras código del mismo tamaño. Así, si la longitud de cada palabra código de un símbolo fuente es n, todas las palabras código tendrán una longitud múltiplo de n y la decodificación se realizará de forma única troceando en bloques de n elementos la cadena recibida y utilizando la inyectividad de la función $C : S \longmapsto T^*$. Un código bloque es aquel cuyas palabras código tienen todas la misma longitud. Todos los códigos bloque son decodificables de manera única.

Teorema 2.2 *Si todas las palabras código de $C : S \longmapsto T^*$ tienen la misma longitud, entonces C es decodificable de manera única.*

El recíproco de este teorema no es cierto ya que hay códigos decodificables de manera única cuyas palabras no tienen la misma longitud. Las palabras del código C''' para quinielas no tienen la misma longitud y es decodificable de manera única ya que el símbolo 1 desempeña un trabajo similar al de un separador:

- Si la cadena comienza por 0, el primer resultado es victoria del equipo local y todos los ceros que le sigan se decodifican como victoria del equipo local hasta llegar al primer 1.
- Si la cadena empieza por 1 y sigue un 0, el par se decodifica como empate.
- Si la cadena empieza por 1 y sigue un 1, el par se decodifica como victoria visitante.
- Hay cadenas que no pueden ser recibidas, como la 01.

Para determinar operativamente si un código es decodificable de manera única resulta necesaria la existencia de un procedimiento operativo que proporcione información concreta al respecto. Los siguientes conjuntos sirven para este propósito.

A partir del código C[1] y de manera recursiva, se define la siguiente cadena de conjuntos:

- Si $n = 0$, entonces $C_0 = C$.
- Si $n \in \mathbb{N}$ y $n > 0$, entonces

$$C_n = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_{n-1} \text{ o bien } u \in C_{n-1} \text{ y } v \in C\}.$$

- Si $n = \infty$, $C_\infty = \bigcup_{n=1}^{\infty} C_n$.

[1] Como ya se ha advertido anteriormente, es un abuso de notación, ya que con C nos estamos refiriendo a la imagen de la función inyectiva $C : S \longmapsto T^*$.

De forma intuitiva, los elementos de C_n se determinan:

1. Calculando las palabras de C_{n-1} que tienen como prefijo una palabra del código C y las palabras del código C que tienen como prefijo una palabra del conjunto C_{n-1}.
2. Los sufijos de las palabras calculadas anteriormente son los elementos de C_n.

Como $C_0 = C$, las palabras del conjunto C_1 son los sufijos de las palabras del código C que tienen como prefijo una palabra de C :

$$C_1 = \{w \in T^+ : uw = v \text{ donde } u, v \in C\}.$$

Para el código C''' de las quinielas, $C'''_0 = \{0, 10, 11\}$, porque ninguna palabra de C''' admite como prefijo a una palabra del propio código C'''. Además, $C'''_1 = \emptyset$, $\forall i = 2, 3, \ldots$ $C'''_i = \emptyset$ y $C'''_\infty = \emptyset$.

A la vista de este ejemplo y analizando la definición de los conjuntos C_i, es evidente que si un conjunto es vacío, los siguientes también lo son.

Propiedad 2.1 *Sea C un código. Si $C_n = \emptyset$, entonces $C_{n+1} = \emptyset$.*

Puesto que la definición de cada C_n sólo depende de C y de C_{n-1}, si dos conjuntos coinciden, sus sucesores también coinciden.

Propiedad 2.2 *Sea C un código. $\forall k \in \mathbb{N}$ $\big((C_i = C_j) \Longrightarrow (C_{i+k} = C_{j+k})\big)$.*

La definición recursiva pueden hacer pensar en una cantidad infinita de distintos conjuntos C_i, sin embargo esto no es cierto, la sucesión $(C_i)_{i\in\mathbb{N}}$ es constante a partir de un determinado término o periódica, como se comprobará a continuación. De hecho, en cada iteración nunca aparecen palabras de longitud estrictamente mayor que las consideradas.

Propiedad 2.3 *Sea C un código tal que la longitud de sus palabras es $\ell_1, \ell_2, \ldots, \ell_q$. Si $w \in C_n$, entonces*

$$|w| \leq \text{máx}\{\ell_1, \ell_2, \ldots, \ell_q\}.$$

Demostración:

Por inducción sobre n.

- Si $n = 0$, el resultado es inmediato porque $C_0 = C$.
- Hipótesis de inducción: se supone que si $w \in C_{n_0}$, entonces $|w| \leq \text{máx}\{\ell_1, \ell_2, \ldots, \ell_q\}$.
- Si $w \in C_{n_0+1}$, entonces se produce alguna de las siguientes situaciones:
  - $\exists u \in C$ $uw \in C_{n_0}$.
    Si $uw \in C_{n_0}$, entonces –por la hipótesis de inducción– $|uw| \leq \text{máx}\{\ell_1, \ell_2, \ldots, \ell_q\}$ y –por la definición de longitud de una palabra– $|w| \leq |uw| \leq \text{máx}\{\ell_1, \ell_2, \ldots, \ell_q\}$.
  - $\exists u \in C_{n_0}$ $uw \in C$.
    Si $uw \in C$, entonces $|uw| \leq \text{máx}\{\ell_1, \ell_2, \ldots, \ell_q\}$ y –por la definición de longitud de una palabra– $|w| \leq |uw| \leq \text{máx}\{\ell_1, \ell_2, \ldots, \ell_q\}$.

  En cualquier caso, la desigualdad se cumple. □

Si la longitud de todas las palabras de cualquier conjunto C_i es menor que una cota independiente de i, entonces todos los C_i tienen una cantidad finita de palabras.

Corolario 2.1 *Sea C un código. $\forall i = 0, 1, 2, \ldots$ $|C_i| < \infty$.*

En particular, si $\ell = \text{máx}\{\ell_1, \ell_2, \ldots, \ell_q\}$, y el código C es r-ario, la suma de la cantidad de palabras de longitudes menores o iguales que ℓ,

$$N = r + r^2 + \cdots + r^\ell = \frac{r(r^\ell - 1)}{r-1},$$

es una cota del cardinal de C_i

$$\forall i = 0, 1, 2, \ldots \ |C_i| \leq \frac{r(r^\ell - 1)}{r-1}.$$

Corolario 2.2 *Sea C un código.* $\forall i = 0, 1, 2, \ldots \ |C_i| \leq N = \frac{r(r^\ell - 1)}{r-1}$.

Como hay 2^N subconjuntos distintos con N o menos elementos, sólo puede haber 2^N conjuntos C_i distintos, luego entre los conjuntos $C_0, C_1, \ldots, C_{2^N}$ debe haber al menos dos coincidentes.

Corolario 2.3 *Sea C un código.* $\exists i, j \in 0, 1, \ldots, 2^N, \ i < j \ C_i = C_j$.

Según el corolario 2.3 y la propiedad 2.2, la sucesión $C_0, C_1, \ldots, C_n, \ldots$ o bien se estabiliza a partir de una determinada posición o bien los conjuntos se van repitiendo periódicamente, de tal manera que si i es una posición i obtenida según el corolario 2.3, $C_\infty = \bigcup_{k=1}^{i} C_k$.

Corolario 2.4 *Sea C un código.* $\exists i \in 0, 1, \ldots, 2^N, \ C_\infty = \bigcup_{k=1}^{i} C_k$.

En general, el cálculo de los conjuntos C_i no es tan sencillo como en el caso del código C''' de las quinielas, pero tampoco suele ser un trabajo imposible. Si C es un código binario, $r = 2$ y la máxima longitud de sus palabras es 3, cada conjunto C_n tendrá como máximo $N = 2(2^3 - 1) = 14$ palabras, y en la cadena $C_1, C_2, \ldots$, los conjuntos empezarán a repetirse antes de la posición $p = 2^{14} = 16384$.

Los conjuntos C_i proporcionan un método operativo que permite determinar si un código es decodificable de manera única.

Teorema 2.3 *Sardinas-Paterson. Un código C es decodificable de manera única si y solamente si $C \cap C_\infty = \emptyset$.*

Demostración (parcial):

La demostración del Teorema de Sardinas-Paterson es relativamente engorrosa desde el punto de vista técnico porque analiza diferentes casos. A continuación se presenta un esbozo en donde aparecen las principales ideas y razonamientos que sustentan la demostración detallada.

'$\Longrightarrow$' Se razona por reducción al absurdo suponiendo que C es decodificable de manera única y $C \cap C_\infty \neq \emptyset$, existiendo una palabra código, w, que pertenece a C y a (pongamos por caso) C_2, de tal manera que existe $u \in C$ y existe $v \in C_1$ tales que $uw = v$ o $vw = u$.

Si $v \in C_1$, entonces, por definición, existen $r, s \in C$ tales que $rv = s$.

- Si $uw = v$ con $u \in C$ y $v \in C_1$, entonces la cadena $ruw = rv = s$ puede decodificarse al menos de dos maneras distintas como una palabra código, $s \in C$, o como tres palabras código, $r, u, v \in C$.
- Si $vw = u$ con $u \in C$ y $v \in C_1$, entonces la cadena $ru = rvw = sw$ puede decodificarse al menos de dos maneras distintas como las palabras código $r, u \in C$ o las palabras código, $s, w \in C$, teniendo en cuenta que $r \neq s$ y que $u \neq w$ porque v no es la palabra nula.

En cualquier caso, la suposición de que hay una palabra común a C y a C_∞ contradice que C sea decodificable de manera única.

'$\Longleftarrow$' Por reducción al absurdo, se supone que $C \cap C_\infty = \emptyset$ y que hay una cadena código $t \in T^*$ que puede decodificarse de dos maneras diferentes, $t = uv = rs$ con $u, v, r, s \in C$, $u \neq r$ y $v \neq s$.

No puede ser que $|u| = |r|$ porque entonces $u = r$. Supongamos sin pérdida de generalidad que $|u| > |r|$, entonces existe $w \in T^+$, tal que $u = rw$, resultando que $w \in C_1$ y que $s \in C_2$ porque $s = wv$ y $v \in C$, luego $s \in C \cap C_2 \subset C \cap C_\infty$, lo que contradice que C y C_∞ sean disjuntos. □

En la definición de código decodificable de manera única, solamente se exige la decodificación de manera única para las cadenas finitas de palabras código. La decodificación de manera única también puede definirse añadiendo la condición más fuerte de que todas las cadenas –finitas o infinitas– formadas por palabras código sean decodificables de manera única.

El teorema de Even-Levenshtein-Riley demuestra que un código finito o infinito es decodificable de manera única si y solamente si $C \cap C_\infty = \emptyset$ y existe algún índice n tal que $C_n = \emptyset$.

Ejercicio 2.1 *Estudiar si los siguientes códigos binarios*

a) $C = \{110, 001, 011, 101, 1111, 1100\}$.

b) $C = \{110, 001, 011, 101, 1110, 1100\}$.

c) $C = \{110, 001, 011, 100, 1111, 1100\}$.

son decodificables de manera única.

Solución:

El Teorema de Sardinas-Paterson permite determinar si un código es decodificable de manera única a partir de los conjuntos $C_n = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_{n-1} \text{ o bien } u \in C_{n-1} \text{ y } v \in C\}$.

a) Si $C = \{110, 001, 011, 101, 1111, 1100\}$, entonces

- $C_0 = C$,
- $C_1 = \{w \in T^+ : uw = v \text{ donde } u, v \in C\} = \{0\}$.
- $C_2 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_1 \text{ o bien } u \in C_1 \text{ y } v \in C\}$.
  Como la única palabra de C_1 tiene longitud 1 y todas las palabras de C tienen longitud superior a 1,
  $$C_2 = \{w \in T^+ : uw = v \text{ donde } u \in C_1 \text{ y } v \in C\} = \{01, 11\}.$$
- $C_3 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_2 \text{ o bien } u \in C_2 \text{ y } v \in C\}$.
  Como las palabras de C_2 tienen longitud 2 y todas las palabras de C tienen longitud superior a 2,
  $$C_3 = \{w \in T^+ : uw = v \text{ donde } u \in C_2 \text{ y } v \in C\} = \{1, 0, 11, 00\}.$$
- $C_4 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_3 \text{ o bien } u \in C_3 \text{ y } v \in C\}$.
  Como las palabras de C_3 tienen longitud menor o igual que 2 y todas las palabras de C tienen longitud superior a 2,
  $$C_4 = \{w \in T^+ : uw = v \text{ donde } u \in C_3 \text{ y } v \in C\} = \{10, 01, 111, 100, 0, 11, 00, 1\}.$$
- $C_5 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_4 \text{ o bien } u \in C_4 \text{ y } v \in C\}$.
  Como las palabras de C_4 tienen longitud menor o igual que 3 y todas las palabras de C tienen longitud superior o igual a 3,
  $$C_5 = \{w \in T^+ : uw = v \text{ donde } u \in C_4 \text{ y } v \in C\} = \{10, 01, 111, 100, 0, 11, 00, 1\} = C_4.$$

- Como $C_4 = C_5$, entonces $\forall n \geq 4\ C_n = \{0,1,00,01,10,11,100,111\}$ y

$$C_\infty = \bigcup_{n=1}^{\infty} C_n = \{0,1,00,01,10,11,100,111\}.$$

Al ser $C \cap C_\infty = \emptyset$, el código C es decodificable de manera única.

b) Si $C = \{110,001,011,101,1110,1100\}$, entonces

- $C_0 = C$,
- $C_1 = \{w \in T^+ : uw = v \text{ donde } u,v \in C\} = \{0\}$.
- $C_2 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_1 \text{ o bien } u \in C_1 \text{ y } v \in C\}$.
  Como la única palabra de C_1 tiene longitud 1 y todas las palabras de C tienen longitud superior a 1,

$$C_2 = \{w \in T^+ : uw = v \text{ donde } u \in C_1 \text{ y } v \in C\} = \{01,11\}.$$

- $C_3 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_2 \text{ o bien } u \in C_2 \text{ y } v \in C\}$.
  Como las palabras de C_2 tienen longitud 2 y todas las palabras de C tienen longitud superior a 2,

$$C_3 = \{w \in T^+ : uw = v \text{ donde } u \in C_2 \text{ y } v \in C\} = \{1,0,10,00\}.$$

- $C_4 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_3 \text{ o bien } u \in C_3 \text{ y } v \in C\}$.
  Como las palabras de C_3 tienen longitud menor o igual que 2 y todas las palabras de C tienen longitud superior a 2,

$$C_4 = \{w \in T^+ : uw = v \text{ donde } u \in C_3 \text{ y } v \in C\} = \{10,01,110,100,11,1\}.$$

La palabra $110 \in C_4 \subset \bigcup_{n=1}^{\infty} C_n = C_\infty$ y también pertenece al código C, por lo que

$$110 \in C \cap C_\infty \text{ y } C \cap C_\infty \neq \emptyset,$$

concluyéndose que el código C no es decodificable de manera única.

Si $C = \{110,001,011,101,1110,1100\}$, la cadena 1100011110 puede descomponerse como concatenación de palabras de C de dos maneras distintas:

$$1100|011|110 \quad \text{y} \quad 110|001|1110,$$

luego el código C no es decodificable de manera única.

c) Si $C = \{110,001,011,100,1111,1100\}$, la cadena 1100011100 puede descomponerse como concatenación de palabras de C de dos maneras distintas:

$$1100|011|100 \quad \text{y} \quad 110|001|1100,$$

luego el código C no es decodificable de manera única.

Ejercicio 2.2 *En el código ternario, C, la palabra 012120120 puede descomponerse como concatenación de palabras código de dos manera diferentes: 012120|120 y 01|212|01|20.*

Determinar una palabra w tal que $w \in C \cap C_\infty$.

Solución:

El código C está formado, al menos, por las siguientes palabras

$$C = \{012120,120,01,212,20,\ldots\}.$$

Una parte de los conjuntos asociados son

- $C_0 = C$,
- $C_1 = \{w \in T^+ : uw = v \text{ donde } u, v \in C\} = \{2120, \ldots\}$.
- $C_2 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_1 \text{ o bien } u \in C_1 \text{ y } v \in C\}$.
  2120 no es prefijo de ningún elemento de C_1, pero $212 \in C$ y $212|0 \in C_1$, luego
  $$C_2 = \{w \in T^+ : uw = v \text{ donde } u \in C_1 \text{ y } v \in C\} = \{0, \ldots\}.$$
- $C_3 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_2 \text{ o bien } u \in C_2 \text{ y } v \in C\}$.
  Como la única palabra localizada de C_2 tiene longitud 1, no hay palabras de C que sean prefijo de esa palabra de C_2, luego
  $$C_3 = \{w \in T^+ : uw = v \text{ donde } u \in C_2 \text{ y } v \in C\} = \{12120, 1, \ldots\}.$$
- $C_4 = \{w \in T^+ : uw = v \text{ donde } u \in C \text{ y } v \in C_3 \text{ o bien } u \in C_3 \text{ y } v \in C\}$.
  No hay palabras de C que sean prefijo de alguna palabra localizada de C_3, pero sí hay una palabra de C_3 que es prefijo de alguna palabra localizada de C,
  $$C_4 = \{w \in T^+ : uw = v \text{ donde } u \in C_3 \text{ y } v \in C\} = \{20, \ldots\}.$$

La palabra $w = 20 \in C \cap C_\infty$.

Es interesante observar que si el código considerado sólo tuviera las palabras localizadas, es decir $C = \{012120, 120, 01, 212, 20\}$, no sería decodificable de manera única, pero $C_5 = \emptyset$, lo que demuestra que no es suficiente que la cadena de conjuntos C_i se estabilice en el conjunto vacío para que el código sea decodificable de manera única.

Ejercicio 2.3 *Sea C un código.*

a) *Analizar las diferentes situaciones por las que una palabra $w \in C \cap C_3$.*

b) *Aplicar los resultados obtenidos para obtener una cadena con dos diferentes decodificaciones del código $C = \{000, 11, 110, 011\}$.*

Solución:

a) Sea $W \in C$.

Si $u \in C_1$, entonces $\exists w', w'' \in C \; w'u = w''$.

w pertenece a C_3 si se da alguna de las siguientes situaciones:

- $\exists w_1 \in C, v_1 \in C_2 \; w_1 w = v_1$.
  Si $v_1 \in C_2$ es porque se da alguna de las siguientes situaciones:
  - $\exists w_3 \in C, u_1 \in C_1 \; w_3 v_1 = u_1$.
    Si $u_1 \in C_1$, entonces $\exists w_5, w_6 \in C \; w_5 u_1 = w_6$.
    Sustituyendo las palabras que no pertenecen a C, en esta última igualdad se obtiene $w_5 w_3 v_1 = w_6$ y
    $$w_5 w_3 w_1 w = w_6.$$
    La palabra código w_6 puede descomponerse como concatenación de las palabras código w_5, w_3, w_1 y w.

- $\exists w_4 \in C, u_2 \in C_1$ $u_2 v_1 = w_4$.
  Si $u_2 \in C_1$, entonces $\exists w_7, w_8 \in C$ $w_7 u_2 = w_8$ por lo que $w_7 u_2 v_1 = w_7 w_4$ y $w_8 v_1 = w_7 w_4$. Sustituyendo v_1 por la concatenación de dos palabras, se obtiene

  $$w_8 w_1 w = w_7 w_4$$

  luego la concatenación de las palabras código w_8, w_1 y w produce la cadena formada por la concatenación de las palabras w_7 y w_4.

- $\exists w_2 \in C, v_2 \in C_2$ $v_2 w = w_2$.
  Si $v_2 \in C_2$, es porque se da alguna de las siguientes situaciones:
  - $\exists w_9 \in C, u_3 \in C_1$ $w_9 v_2 = u_3$.
    Si $u_3 \in C_1$, entonces $\exists w_{11}, w_{12} \in C$ $w_{11} u_3 = w_{12}$.
    Concatenando con w_9 la primera igualdad, se obtiene $w_9 v_2 w = w_9 w_2$ y $u_3 w = w_9 w_2$ y concatenando esta última igualdad con w_{11}, resulta $w_{11} u_3 w = w_{11} w_9 w_2$ o bien

    $$w_{12} w = w_{11} w_9 w_2$$

    luego la concatenación de las palabras w_{12} y w es igual a la concatenación de las palabras w_{11}, w_9 y w_2.
  - $\exists w_{10} \in C, u_4 \in C_1$ $u_4 v_2 = w_{10}$.
    Si $u_4 \in C_1$, entonces $\exists w_{13}, w_{14} \in C$ $w_{13} u_4 = w_{14}$, por lo que $w_{13} u_4 v_2 = w_{13} w_{10}$ y $w_{14} v_2 = w_{13} w_{10}$.
    Concatenando w_{14} en la igualdad $v_2 w = w_2$, se obtiene $w_{14} v_2 w = w_{14} w_2$ y

    $$w_{13} w_{10} w = w_{14} w_2$$

    luego la concatenación de las palabras w_{13}, w_{10} y w es igual a la concatenación de las palabras w_{14} y w_2.

b) Si $C = \{000, 11, 1100, 011\}$, entonces $C_0 = C$, $C_1 = \{00\}$, $C_2 = \{0\}$, $C_3 = \{00, 11\}$, $C_4 = \{0, 00\}$, $C_5 = \{0, 00, 11\}$, $C_6 = \{0, 00, 11\}$.

La palabra $11 \in C \cap C_3$.

Conocido C_2, se puede asegurar que $0|11 = 011 \in C$ con $0 \in C_2$.

Conocido C_1, $00|0 = 000 \in C$ con $00 \in C_1$ que proviene de $11|00 = 1100 \in C$ con $11 \in C$.

Concatenando $0|11 = 011$ con 00, se obtiene $00|0|11 = 00|011$ que proporciona una palabra código a partir de $0 \in C_2$ agrupando lo dos primeros elementos, $000|11 = 00|011$. Concatenando esta cadena resultante con la palabra 11, se consigue formar una palabra a partir de $00 \in C_1$, $11|000|11 = 11|00|011$ de tal modo que hay una cadenas de palabras código que puede decodificarse de dos maneras:

$$11|000|11 = 1100|011.$$

Ejercicio 2.4 *Comprobar que el código ternario $C = \{02, 12, 120, 21\}$ es decodificable de manera única pero hay cadenas infinitas que pueden decodificable de dos maneras diferentes.*

Solución:

Los conjunto asociados al código C son

$$C_0 = C,\ C_1 = \{0\},\ C_2 = \{2\},\ C_3 = \{1\},\ C_4 = \{2, 20\},\ C_5 = \{1\},\ C_6 = \{2, 20\}, \ldots \text{ y } C_\infty = \{0, 1, 2, 20\}$$

por lo que $C \cap C_\infty = \emptyset$ y, por el teorema de Sardinas-Paterson, C es decodificable de manera única.

Al no existir ningún índice a partir del cual los conjuntos asociados al códigos sean vacíos, no se cumplen las hipótesis del teorema de Even-Levenshtein-Riley y existe al menos una cadena infinita que puede decodificarse de dos maneras diferentes:

$$120212121\ldots = 120|21|21|21|\ldots = 12|02|12|12|12\ldots.$$

3. Códigos instantáneos y prefijos

Ejemplo 3.1 *El código binario* $C = \{0, 01, 011, 111\}$ *es decodificable de manera única porque*

$$C_0 = C,\ C_1 = \{1, 11\} \quad y \quad C_2 = \{11, 1\}.$$

En general, para poder decodificar un mensaje codificado con C hay que esperar hasta recibir la secuencia completa de bits, ya que una secuencia que empieza en 0 *y a continuación tiene unos,* $0111\cdots11$*, depende del número de unos que tenga para descomponerse de una u otra manera:*

- *Si hay 0 unos, la descomposición es trivial:* 0.
- *Si hay 1 uno, la descomposición es* 01.
- *Si hay 2 unos, la descomposición es* 011.
- *Si hay 3 unos, la descomposición es* $0|111$.
- *Si hay 4 unos, la descomposición es* $01|111$.
- *Si hay 5 unos, la descomposición es* $011|111$.
- *Si hay 6 unos, la descomposición es* $0|111|111$.

De tal modo que si hay k unos,

- *Si* $k = 3n$*, entonces* $01\cdots1 = 0|\underbrace{111|\cdots|111}_{\frac{k}{3}}$.
- *Si* $k = 3n+1$*, entonces* $01\cdots1 = 01|\underbrace{111|\cdots|111}_{\frac{k-1}{3}}$.
- *Si* $k = 3n+2$*, entonces* $01\cdots1 = 011|\underbrace{111|\cdots|111}_{\frac{k-2}{3}}$.

Si la cadena recibida empieza por 1*, los primeros 3 unos deben codificarse como* 111 *independientemente de la información posterior.*

Nos interesa localizar códigos decodificables de manera única cuya decodificación pueda realizarse instantáneamente sin tener que esperar a recibir todos los símbolos del mensaje.

Un código C es instantáneo, si para cualquier secuencia de palabras código, $w = w_1 w_2 \ldots$, cualquier subsecuencia que comience del mismo modo que w se decodifica de manera única, independientemente de las palabras que aparezcan posteriormente.

El código C del ejemplo 3.1 no es instantáneo. El código $C' = \{0, 10, 11\}$ es instantáneo porque sólo hay tres maneras de decodificar y no dependen de las palabras del final de la cadena:

- Si aparece un 0, la palabra recibida sólo puede decodificarse como 0.
- Si aparece un 1, la palabra tiene que tener dos bits:
  - Si es 0, la palabra recibida es 10.
  - Si es 1, la palabra recibida es 11.

En la definición de código instantáneo, el concepto subcadena incluye la cadena total, luego cualquier cadena de palabras de un código instantáneo puede decodificarse de manera única. Un código instantáneo es decodificable de manera única, pero hay códigos decodificables de manera única que no son instantáneos, como el código C del ejemplo 3.1.

Teorema 3.1 *Cualquier código instantáneo es decodificable de manera única.*

El código $C = \{1, 11\}$ tiene la propiedad de que la palabra 1 es el comienzo de la palabra 11. Un código que no cumpla esta propiedad se llama código prefijo.

Un código C es prefijo, si no hay palabras código que sean prefijo (inicio) de alguna otra palabra código.

La relación entre códigos instantáneos y prefijos es estrechísima.

Teorema 3.2 *Un código es instantáneo si y solamente si es un código prefijo.*

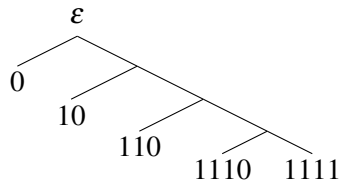
Demostración:

'$\Longrightarrow$' Se demuestra que si C no es prefijo, entonces no es instantáneo.

Si C no es un código prefijo, entonces existe una palabra, $w \in C$, que es prefijo de otra, $v = ww' \in C$. Si se recibe, la palabra w, hay que esperar a la recepción de la siguiente palabra para saber si hay que decodificar la cadena como w más otra palabra o como la palabra $v = ww'$. El código no es instantáneo.

'$\Longleftarrow$' Si C es un código prefijo, cualquier cadena de palabras puede decodificarse la primera vez que se completa una palabra, porque no hay más palabras que empiecen del mismo modo, luego el código es instantáneo: la primera palabra localizada es la que se decodifica, independientemente de los datos que vengan a continuación. □

Observando que las palabras de un código puede representarse mediante árboles en donde debajo de cada nodo están las palabras que comienzan por la etiqueta de ese nodo, véase el Cuadro 1, pueden construirse códigos instantáneos podando convenientemente el árbol generador. Así el código $C = \{0, 10, 110, 1110, 1111\}$ es un código instantáneo que se ha obtenido eliminado las ramas inferiores de las palabras seleccionadas.



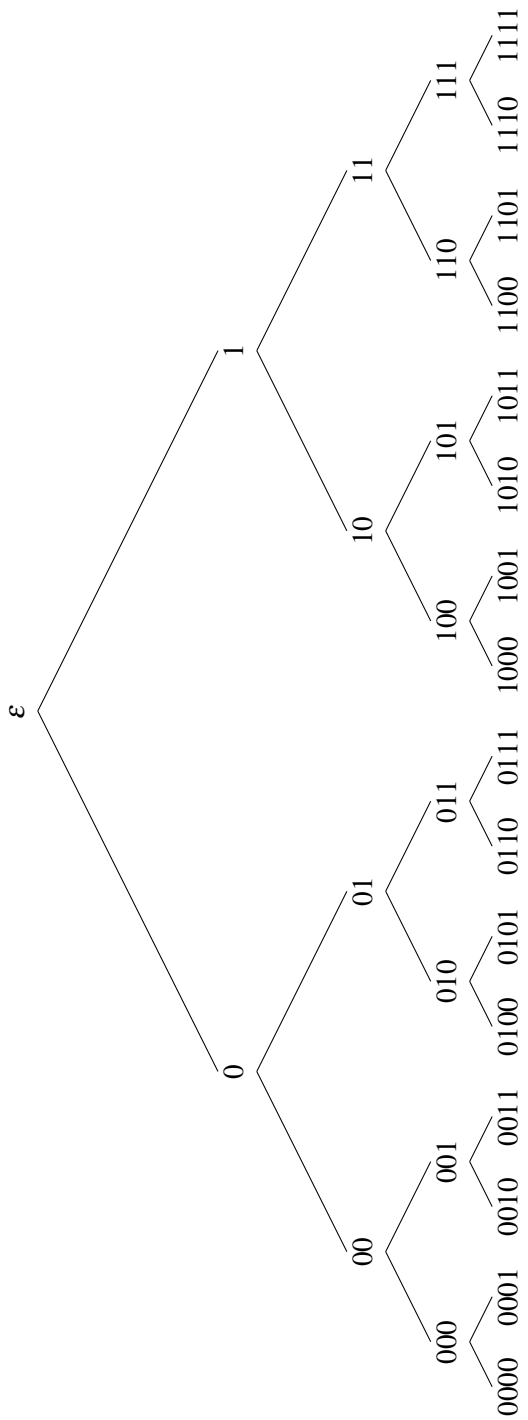
El código C está formado por una palabra de longitud 1, una de longitud 2, una de longitud 3 y dos de longitud 4. Es evidente que siguiendo una estrategia de poda, no se pueden conseguir códigos con la distribución de longitudes de palabra que uno quiera. Por ejemplo, es imposible encontrar códigos binarios instantáneos con dos palabras de longitud 1 y una de longitud 2 o con una palabra de longitud 1, otra de longitud 2, dos de longitud 3 y una de longitud 4.

En un código r-ario, la elección de una palabra de longitud n para la construcción de un código instantáneo supone que no pueden utilizarse todas las palabras que están debajo del nodo elegido, produciéndose una reducción[2] del número de palabras candidatas en $\frac{1}{r^n}$. Así, si en un código binario se elige una palabra de longitud 1, la siguiente palabra sólo puede escogerse en la mitad $(1 - \frac{1}{2})$ del árbol. Si se elige una palabra de longitud 1 y otra de longitud 2, las restantes palabras sólo podrán escogerse en la cuarta $(1 - \frac{1}{2} - \frac{1}{4} = \frac{1}{4})$ parte del árbol. En el momento en que esta cantidad sea negativa o cero, no habrá más palabras para elegir.

Para que exista un código instantáneo r-ario con longitudes de palabra $\ell_1, \ell_2, \ldots, \ell_p$, tiene que ser

$$\sum_{i=1}^{p} \frac{1}{r^{\ell_i}} \leq 1.$$

[2] r^{-n} representa el concepto impreciso de proporción de un árbol bajo un vértice etiquetado con una palabra de longitud n. No hay que olvidar que a priori la profundidad del árbol generador no está limitada.



Cuadro 1: Árbol generador de códigos binarios

Hay códigos instantáneos ternarios con cinco palabras de longitudes $1,2,3,3,4$ ya que

$$\frac{1}{3}+\frac{1}{3^2}+\frac{1}{3^3}+\frac{1}{3^3}+\frac{1}{3^4}=\frac{43}{81}\leq 1,$$

por ejemplo, $C=\{0,10,110,120,2222\}$, pero no hay códigos instantáneos binarios con cinco palabras de longitudes $1,2,3,3,4$ ya que

$$\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^4}=\frac{17}{16}>1.$$

4. Desigualdades de Kraft y de McMillan

La relación entre las longitudes de las palabras de un código y el que sea instantáneo es más fuerte que lo analizado hasta el momento. La desigualdad de Kraft proporciona una condición necesaria y suficiente para la existencia de códigos instantáneos según las longitudes de sus palabras.

Teorema 4.1 *Existe un código r-ario instantáneo con longitudes de palabra $l_1,\ldots,l_q$ si y solamente si*

$$\sum_{i=1}^{q}\frac{1}{r^{l_i}}\leq 1.$$

Demostración:

'$\Longrightarrow$' Sea C un código instantáneo o prefijo y $\ell=\max\{l_1,\ldots,l_q\}$.

En el árbol completo, los nodos del nivel ℓ pueden no proceder de ninguna palabra código, o pueden proceder solamente desde una palabra código como mucho, ya que si hubiera dos palabras código distintas que condujeran a una palabra de nivel superior, entonces una sería prefijo de la otra.

Desde cada palabra código w_i de longitud ℓ_i se puede acceder a $r^{\ell-\ell_i}$ nodos de nivel ℓ, que corresponden con todas las palabras que comienzan con w_i y que tienen longitud ℓ.

En total, desde las palabras código, se puede acceder a $\sum_{i=1}^{q} r^{\ell-\ell_i}$ palabras del nivel ℓ. Como en el nivel ℓ hay r^ℓ palabras, entonces

$$\sum_{i=1}^{q} r^{\ell-\ell_i}\leq r^\ell.$$

Dividiendo por r^ℓ, se obtiene la desigualdad buscada

$$\sum_{i=1}^{q} r^{-\ell_i}=\sum_{i=1}^{q}\frac{1}{r^{\ell_i}}\leq 1.$$

'$\Longleftarrow$' Sea C un código con q palabras de longitudes $\ell_1,\ell_2,\ldots,\ell_q$ tales que $\sum_{i=1}^{q}\frac{1}{r^{l_i}}\leq 1$.

Sin pérdida de generalidad puede suponerse que $\ell_1\leq\ell_2\leq\cdots\leq\ell_q$.

Puesto que no hay palabras de longitud mayor que ℓ_q, consideramos el árbol finito de generación de códigos que comienza en la palabra vacía y termina en el nivel ℓ_q. En total hay r^{ℓ_q} candidatas a palabra código de longitud máxima: todas las palabras de longitud ℓ_q que se pueden formar con r símbolos.

Tomamos una palabra cualquiera de longitud ℓ_1, w_1, y podamos el árbol de tal manera que se eliminan todas las ramas que conducen a palabras del nivel ℓ_q que comienzan por w_1. Se han eliminado $r^{\ell_q-\ell_1}$ palabras de nivel ℓ_q, todas las de longitud ℓ_q y que empiezan por w_1, junto con las de longitudes entre ℓ_1 y ℓ_q que empiezan por w_1.

Si $q = 1$, el código resultante es instantáneo, por lo que el resultado es inmediato.

Si $q > 1$, utilizando la hipótesis del teorema se obtiene la desigualdad

$$r^{\ell_q-\ell_1} = r^{\ell_q}\frac{1}{r^{\ell_1}} < r^{\ell_q}\sum_{i=1}^{q}\frac{1}{r^{l_i}} \leq r^{\ell_q}$$

que asegura que en el nivel ℓ_q todavía restan palabras que no tienen como prefijo a w_1.

En las ramas que no se han podado, tiene que existir una palabra, w_2, de longitud $\ell_2 \geq \ell_1$. Tomamos w_2 como la segunda palabra del código y podamos el árbol eliminando todas las ramas que conducen a palabras del nivel ℓ_q que empiezan por w_2, lo que supone $r^{\ell_q-\ell_2}$ palabras menos en el nivel ℓ_q del árbol.

En estos dos pasos se han eliminado

$$r^{\ell_q-\ell_1} + r^{\ell_q-\ell_2} = r^{\ell_q}(r^{-\ell_1} + r^{-\ell_2}) = r^{\ell_q}\sum_{i=1}^{2} r^{-\ell_i}$$

palabras de longitud ℓ_q porque al haber ordenado las longitudes, no hay ramas comunes a w_1 y w_2, ni w_2 es la forma $w_2 = w_1w'$.

Si $q = 2$, el proceso ha terminado y tenemos un código prefijo.

Si $q > 2$, todavía se puede continuar con el proceso porque, por hipótesis,

$$r^{\ell_q}\sum_{i=1}^{2} r^{-\ell_i} < r^{\ell_q}\sum_{i=1}^{q} r^{-\ell_i} \leq r^{\ell_q}$$

y todavía quedan en el nivel ℓ_q palabras que no empiezan ni por w_1 ni por w_2.

Repitiendo el proceso $q-1$ veces, se llega a la desigualdad

$$r^{\ell_q-\ell_1} + r^{\ell_q-\ell_2} + \cdots r^{\ell_q-\ell_{q-1}} = r^{\ell_q}\sum_{i=1}^{q-1} r^{-\ell_i} < r^{\ell_q}\sum_{i=1}^{q} r^{-\ell_i} \leq r^{\ell_q}$$

que asegura que todavía queda en el nivel ℓ_q una palabra que no empieza por ninguna de las $q-1$ palabras escogidas. Esta palabra junto con las anteriores forma un código prefijo e instantáneo. □

Todo código instantáneo es decodificable de manera única, pero hay códigos decodificables de manera única que no son instantáneos. Parecería razonable que una condición necesaria y suficiente para determinar la existencia de códigos decodificables de manera única en función de las longitudes de sus palabras sería más relajada que la proporcionada por la desigualdad de Kraft para códigos instantáneos. Sin embargo no es así. La desigualdad de McMillan establece la misma condición para los dos tipos de códigos.

Teorema 4.2 *Existe un código r-ario decodificable de manera única con longitudes de palabra $l_1, \ldots, l_q$ si y solamente si*

$$\sum_{i=1}^{q}\frac{1}{r^{l_i}} \leq 1.$$

La demostración se encuentra en las páginas 15 y 16 del *Information and Coding Tehory* de Gareth A. Jones and J. Mary Jones.

De las desigualdades de Kraft y McMillan se deduce el siguiente resultado:

Corolario 4.1 *Existe un código r-ario decodificable de manera única con longitud de palabras $l_1, \ldots, l_q$ si y solamente si existe un código r-ario instantáneo con longitud de palabras $l_1, \ldots, l_q$*

Estas desigualdades afirman que existen códigos con determinados parámetros que son instantáneos y decodificables de manera única y deben tenerse en cuenta las siguientes aclaraciones:

- Si un código es instantáneo ya es decodificable de manera única, luego si se construye un código instantáneo ya se tiene uno decodificable de manera única con las mismas longitudes de las palabras.
- Las desigualdades de Kraft y McMillan afirman que se pueden encontrar códigos instantáneos y decodificables únicamente si se cumplen las condiciones, pero no dicen que un código que cumpla esas condiciones sea instantáneo o decodificable de manera única. Por ejemplo, el código $C = \{0,00,000\}$ no es decodificable de manera única aunque las longitudes de sus palabras sí cumplan la desigualdad exigida.
- Los códigos decodificables de manera única no tienen que ser instantáneos, pero seguro que existe un código instantáneo que cumple las condiciones de longitudes de las palabras. Por ejemplo, el código $C_1 = \{0,01,11\}$ no es instantáneo y el código $C_2 = \{0,10,11\}$, con las mismas longitudes de palabras, sí lo es.

En la tabla siguiente

Códigos binarios con longitudes de palabra 1, 2 y 1, 2, 2							
		$\{0,00\}$	$\{1,11\}$	$\{0,00,01\}$ $\{1,11,10\}$	$\{0,00,10\}$ $\{1,11,01\}$	$\{0,00,11\}$ $\{1,11,00\}$	$\{0,01,10\}$ $\{1,10,01\}$
Decodificables de manera única	No prefijos	$\{0,01\}$	$\{1,10\}$			$\{0,01,11\}$	$\{1,10,00\}$
	Prefijos	$\{0,10\}$	$\{0,11\}$	$\{1,01\}$	$\{1,00\}$	$\{0,10,11\}$	$\{1,01,00\}$

se clasifican los 8 códigos binarios de longitudes de palabra 1, 2 y los 12 códigos binarios de longitudes de palabra 1, 2, 2, según sean decodificables de manera única o no. A su vez, los decodificables de manera única se clasifican en prefijos y no prefijos. Puede observarse que al cumplirse la condición de las desigualdades de Kraft y de McMillan, en ambos casos existen códigos prefijos y códigos decodificables de manera única que no son prefijos.

Ejercicio 4.1 *¿Cuántos códigos ternarios instantáneos hay con 9 palabras de longitudes* $1,2,2,2,2,2,3,3,3$*?*

Solución:

Observando el árbol de generación de palabras código del cuadro 2, hay 3 maneras de elegir una palabra de longitud 1.

Elegida una palabra de longitud 1, pueden elegirse hasta 6 palabras de longitud 2, lo que hace un total de $\binom{6}{5}$ opciones si se quieren elegir 5 palabras.

Elegidas 5 palabras de longitud 2, quedan únicamente 3 palabras de longitud 3 que deben escogerse para cumplir las condiciones del enunciado.

En total, hay $3\binom{6}{5} = 18$ códigos ternarios instantáneos con 9 palabras de longitudes $1,2,2,2,2,2,3,3,3$.

Ejercicio 4.2 *¿Cuántas secuencias código de longitud* ℓ, N_ℓ, *pueden formarse con las palabras del código* $C = \{0,10,11\}$*?*

Solución:

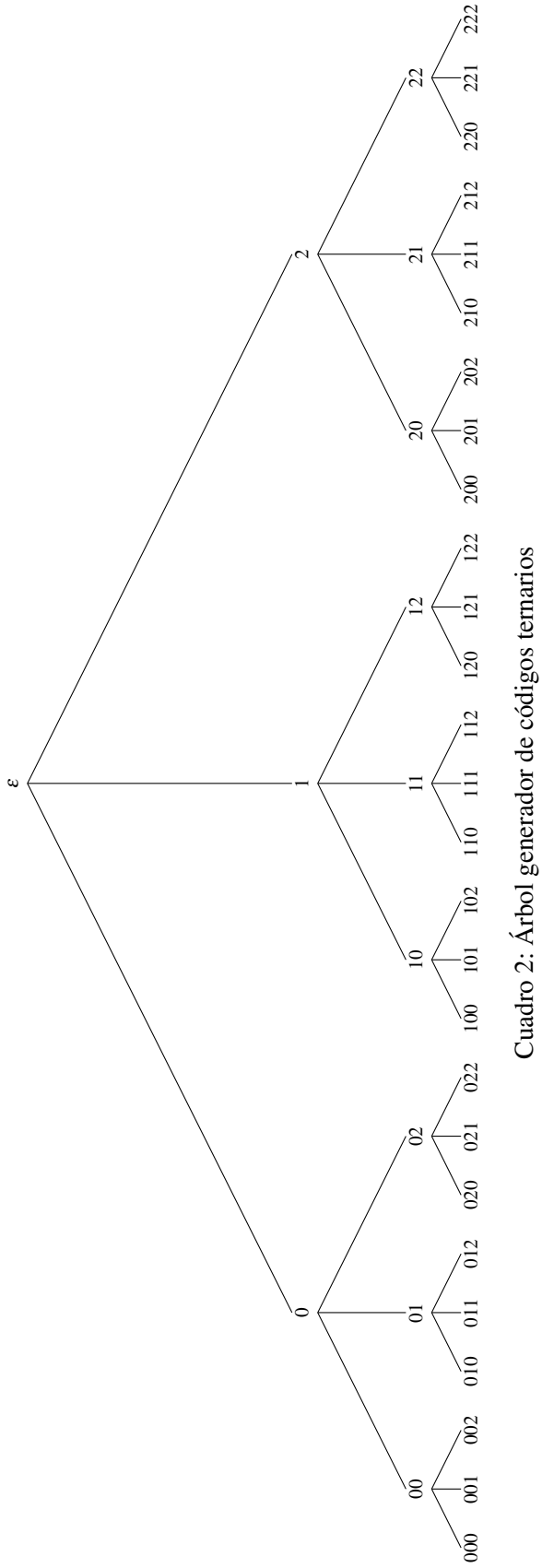
Si $\ell = 1$, sólo hay una secuencia código, porque sólo hay una palabra de longitud 1, luego $N_1 = 1$.

Si $\ell = 2$, sólo hay tres secuencias código: 00, 10 y 11, luego $N_2 = 3$.

Si $\ell = 3$, como la longitud de las palabras código es menor o igual que dos, la secuencia tiene que estar formada al menos por dos palabras, $s = \cdot w$, con $w \in C$. Con la última palabra de la secuencia $w = 0$, se pueden formar N_2 secuencias; con la última palabra $w = 10$ se pueden formar N_1 secuencias y con la última palabra $w = 11$ se pueden formar N_1 palabras, de tal modo que $N_3 = N_2 + N_1 + N_1$.

Si $\ell = 4$, o bien se ha añadido la palabra 0 al final de una secuencia de longitud 3, o bien se ha añadido una palabra de longitud 2 (10 o 11) al final de una secuencia de longitud 2, luego $N_4 = N_3 + 2N_2$.

En general, N_ℓ satisface la relación recurrente $N_\ell = N_{\ell-1} + 2N_{\ell-2}$ con $N_2 = 3$ y $N_1 = 1$.



Cuadro 2: Árbol generador de códigos ternarios

La ecuación característica de la relación recurrente es $\lambda^2 = \lambda + 2$, que tiene como soluciones $\lambda_1 = 2$ y $\lambda_1 = -1$, por lo que una solución general es

$$N_n = A2^n + B(-1)^n$$

que particularizada para $n = 2$ y $n = 1$ permite calcular los valores de A y B ya que

$$1 = 2A - B \quad \text{y} \quad 3 = 4A + B,$$

de donde $A = \frac{2}{3}$ y $B = \frac{1}{3}$, resultando

$$\forall \ell \geq 1 \; N_\ell = \frac{2^{\ell+1} + (-1)^\ell}{3}.$$

Ejercicio 4.3 *¿Cuántos códigos r-arios instantáneos de q palabras de longitudes $\ell_1 < \ell_2 < \cdots < \ell_q$ con $\sum_{i=1}^{q} r^{-\ell_i} \leq 1$, pueden formarse?*

Solución:

Para elegir la primera palabra, hay r^{ℓ_1} opciones.

De las r^{ℓ_2} palabras de longitud ℓ_2 hay que quitar las que están debajo de la primera palabra elegida: $r^{\ell_2} - r^{\ell_2-\ell_1}$, luego hay $r^{\ell_2}\left(1 - r^{-\ell_1}\right)$ opciones de elegir la segunda palabra.

De las r^{ℓ_3} palabras de longitud ℓ_3 hay que quitar las palabras que están debajo de la primera palabra elegida y las que están debajo de la segunda palabra elegida: $r^{\ell_3} - r^{\ell_3-\ell_1} - r^{\ell_3-\ell_2}$, resultando $r^{\ell_3}\left(1 - r^{-\ell_1} - r^{-\ell_2}\right)$ opciones de elegir la tercera palabra.

De los resultados anteriores, se puede inferir que hay

$$r^{\ell_1+\cdots+\ell_q}\left(1 - r^{-\ell_1}\right)\left(1 - r^{-\ell_1} - r^{-\ell_2}\right)\cdots\left(1 - r^{-\ell_1} - r^{-\ell_2} - \cdots - r^{-\ell_{q-1}}\right)$$

códigos r-ários instantáneos de q palabras de longitudes $\ell_1 < \ell_2 < \cdots < \ell_q$.

Por ejemplo, hay $3^{1+2+3}(1-3^{-1})(1-3^{-1}-3^{-2}) = 270$ códigos ternarios con una palabra de longitud 1, otra de longitud 2 y otra de longitud 3.

Esta fórmula no es válida si hay longitudes iguales, porque las maneras de escoger, por ejemplo, dos palabras de longitud 1 son las combinaciones de r elementos tomados de dos en dos y no el producto $r^{1+1}(1-r^{-1}) = r^2 - r = r(r-1)$ ya que hay que contar con las elecciones que se repiten.

5. Códigos exhaustivos

Un código C es exhaustivo si existe un nivel ℓ tal que cualquier secuencia de elementos del alfabeto código de longitud ℓ tiene una palabra código como prefijo.

Ejemplo 5.1 *El código binario $C = \{0, 00\}$ no es exhaustivo porque cualquier cadena que empiece con un 1 no tiene como prefijo a ninguna palabra código.*

El código $C = \{0, 10, 11\}$ es exhaustivo porque cualquier cadena de longitud 3 tiene como prefijo una palabra código: 0.00, 0.01, 0.10, 0.11, 10.0, 10.1, 11.0, 11.1. Verdaderamente, también es exhaustivo de nivel 2 porque todas las palabras de longitud 2 tienen como prefijo una palabra código: 0.0, 0.1, 10, 11.

Evidentemente, si un código C es exhaustivo de nivel ℓ, cualquier cadena de longitud superior a ℓ también tiene como prefijo una palabra código de C. Esta observación permite establecer definiciones alternativas.

Propiedad 5.1 *Un código C es exhaustivo de nivel ℓ si y solamente si cualquier cadena infinita de elementos del alfabeto código tiene como prefijo una palabra código de C.*

Un código C es exhaustivo de nivel ℓ si y solamente si cualquier cadena infinita de símbolos puede descomponerse como concatenación de palabras código.

En un código instantáneo de longitud máxima ℓ, no todas las cadenas de longitud ℓ tienen como prefijo una palabra código. En un código instantáneo, puede haber secuencias de palabras de longitud máxima ℓ que tengan como prefijo una o más palabras código.

Propiedad 5.2 *Si un código r-ario con palabras de longitudes $\ell_1, \ell_2, \ldots, \ell_q$ es exhaustivo, entonces*

$$\sum_{i=1}^{q} r^{-\ell_i} \geq 1.$$

Demostración:

Si C es exhaustivo de nivel ℓ, entonces la cantidad de secuencias de longitud ℓ que tienen como prefijo una palabra código, $\sum_{i=1}^{q} r^{\ell-\ell_i}$, tiene que ser superior al número de palabras de longitud ℓ, r^ℓ, por lo que

$$\sum_{i=1}^{q} r^{\ell-\ell_i} \geq r^\ell.$$

Dividiendo por r^ℓ, se obtiene la desigualdad buscada, $\sum_{i=1}^{q} r^{-\ell_i} \geq 1$.

Si $\sum_{i=1}^{q} r^{-\ell_i} = 1$, entonces cada palabra de longitud ℓ tiene como prefijo una y solamente una palabra código, por lo que C es prefijo e instantáneo. Recíprocamente, si C es instantáneo y $\sum_{i=1}^{q} r^{-\ell_i} = 1$, entonces C es exhaustivo porque como cada palabra de longitud ℓ no puede proceder de más de una palabra código, la suma de todas las palabras que proceden de palabras código es exactamente r^ℓ. □

Corolario 5.1 *Sea C un código r-ario con palabras de longitudes $\ell_1, \ell_2, \ldots, \ell_q$. Se cumple*

a) *Si C es exhaustivo y $\sum_{i=1}^{q} r^{-\ell_i} = 1$, entonces es instantáneo.*

b) *Si C es instantáneo y $\sum_{i=1}^{q} r^{-\ell_i} = 1$, entonces es exhaustivo.*

c) *Si C es instantáneo y exhaustivo, entonces $\sum_{i=1}^{q} r^{-\ell_i} = 1$.*

Sin embargo, las relaciones anteriores no son de equivalencia ya que

- Hay códigos instantáneos, como $C = \{0\}$ que no es exhaustivo y $\sum_{i=1}^{q} r^{-\ell_i} = 2^{-1} \neq 1$.
- Hay códigos exhaustivos, como $C = \{0, 1, 00\}$ que no es instantáneo y $\sum_{i=1}^{q} r^{-\ell_i} = \frac{5}{4} \neq 1$.
- Hay códigos, como $C = \{0, 00, 10\}$ que cumplen $\sum_{i=1}^{q} r^{-\ell_i} = 2^{-1} + 2^{-2} + 2^{-2} = 1$ y que no son ni exhaustivos ni instantáneos.