Universidad de Granada

Ingeniería Informática

Computación y Sistemas Inteligentes

Relacion Tema 3

Autor: José Antonio Ruiz Millán

Asignatura: Teoria de la Información y de la Codificación

2 de diciembre de 2018

1. **Explica el Teorema Fundamental de Shannon para canales sin ruido, y expón sus consecuencias más inmediatas.**

   Por dedfinición tenemos que el Teorema Fundamental de Shannon nos dice que *"Sea una fuente S con entropía $H(S)$ (bits por símbolo) y un canal con capacidad C (bits por segundo). Entonces, es posible codificar la salida de la fuente de tal modo que se transmita a una velocidad de $C/H(S) - \epsilon$ símbolos por segundo sobre el canal, donde ϵ es tan pequeño como se quiera. No es posible transmitir a un régimen promedio mayor que $C/H(S)$"*

   Esto hace que el teorema de Shannon para la codificación en canales sin ruido sea de extrema importancia ya que nos indica **el número de bits que se pueden codificar por un canal con una capacidad**.

   Gracias a este teorema, Shannon prueba la existencia de un límite a la eficiencia de la codificación de una fuente.

2. **Explica qué es un código de traducción única. Pon un ejemplo de un código que permita codificar 5 mensajes {A, B, C, D, E} que sea de traducción única. Pon un ejemplo de código para el mismo conjunto de mensajes, que no sea de traducción única y explica porqué no lo es. Indica, mediante un ejemplo que codifique la secuencia de mensajes {AACAD}, cuál es la principal desventaja de códigos que no son de traducción única.**

   Se dice que un **código de traducción única** es aque para el que cualesquiera sucesión única de mensajes a transmitir, corresponde a una única sucesión de símbolos transmitidos, es decir, no existe ninguna codificación distinta a la que se quiere enviar con la que se pueda interpretar el mensaje que se recive.

   Un ejemplo de código de traducción única sería el siguiente:

   A $\rightarrow$ 000
   B $\rightarrow$ 111
   C $\rightarrow$ 010
   D $\rightarrow$ 101
   E $\rightarrow$ 001

   Un ejemplo de código que **no** es de traducción única:

   A $\rightarrow$ 0
   B $\rightarrow$ 1
   C $\rightarrow$ 01
   D $\rightarrow$ 10
   E $\rightarrow$ 11

   Este ejemplo no es de traducción única porque no cumple la propia definición de código de traducción única, es decir, podemos decodificar un mensaje de distintas formas y no de una única forma, lo que hace que esta codificación no sea de traducción única.

   Por ejemplo, para el código "AACAD" tenemos lo siguiente:

   - Traducción única: 000000010000101

- No traducción única: 0001010

  Como podemos ver, la principal desventaja de este código es claramente que si no utilizamos bits de control para indicar qué estamos mandando, es confuso y no podemos decodificar el mensaje ya que tenemos varias opciones como por ejemplo, este mensaje se puede decodificar como "AACAD" pero también como "AAADD".

3. **Explica qué es un código instantáneo. Indica un procedimiento general para generar códigos instantáneos y utilízalo para generar un código instantáneo que permita codificar 5 mensajes {A, B, C, D, E}. Indica cómo se codificaría y decodificaría la secuencia de mensajes {AACAD}. Por otro ejemplo de código que no sea instantáneo para el mismo conjunto de mensajes. Indica cómo sería el proceso para decodificar con este nuevo código, y exponga un ejemplo de decodificación para la cadena de mensajes {AACAD}. Por último, en relación con el ejercicio anterior, ¿todo código instantáneo es de traducción única? ¿Y todo código de traducción única es instantáneo? Razone la respuesta y ponga ejemplos.**

   Se dice que un código **es instantáneo** si ninguna palabra codificada coincide con el comienzo de otra.

   Tenemos **un algoritmo** capaz de generar un código instantáneo:

   Sea $M = \{m_1, m_2, ..., m_N\}$ un conjunto de N mensajes a enviar (alfabeto de la fuente). Sea $A = \{a_1, a_2, ..., a_D\}$ el conjunto de D símbolos transmisibles por el canal (alfabeto del código).

   - Se divide M en D subjconjuntos $M_1 = \{m_1, m_2, ..., m_k\}$, $M_2 = \{m_{k+1}, m_{k+2}, ..., m_{2*k}\}, ..., M_D = \{m_r, m_{r+1}, ..., m_N\}$. A cada M_i se le asigna el símbolo a_i del alfabeto del código.
   - Repetir la operación para cada M_i, generando M_{i1}, M_{i2},...,M_{iD} y asignando a cada M_{ij} los códigos $\{a_i a_j\}$, hasta que cada subconjunto generado tenga un único elemento.

   Ahora utilizaré este algoritmo para generar un **código instantáneo**.

   - Partimos de que tenemos

     $$M = \{A, B, C, D, E\}$$

     Y un alfabeto

     $$A = \{0, 1\}$$

   - Ahora dividimos lso subconjuntos, obteniendo:

     $$M_1 = \{A, B, C\} \to 0$$

     $$M_2 = \{D, E\} \to 1$$

   - Ahora volvemos a dividir cada uno de ellos:
     - M1:

       $$M_{1_1} = \{A, B\} \to 00$$

       $$M_{1_2} = \{C\} \to 01$$

     - M2:

$$M_{2_1} = \{D\} \to 10$$
$$M_{2_2} = \{E\} \to 11$$

- Por último dividimos lo que nos ha quedado en M1:

$$M_{1_{1_1}} = \{A\} \to 000$$
$$M_{1_{1_2}} = \{B\} \to 001$$

Por lo que finalmente hemos obtenido el siguiente código.

A → 000

B → 001

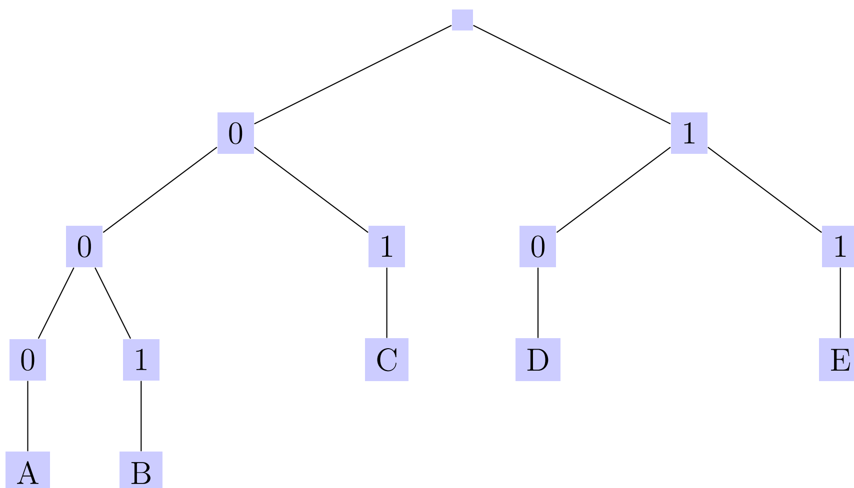
C → 01

D → 10

E → 11

Es facil comprobar que el código es instantáneo ya que no repite ninguna palabra como comienzo de otra.

Para **codificar y decodificar**, podemos utilizar un arbol que represente la secuencia que hemos creado, por ejemplo:



Por lo ahora **para codificar**, lo que tenemos que hacer es recorrer el mensaje alrevés e ir recorriendo el arbol de abajo a arriba. Por ejemplo para el mensaje *"AACAD"*, tenemos que empezar por la "D" he ir rellenando hacia la izquierda, quedando codificado como **0000000100010**. Tambien podemos empezar por la izquierda he ir rellenando hacia la derecha y al terminar, cambiar el sentido de la cadena codificada.

Por caso contrario, **para decodificar**, comenzamos por la izquierda y comenzando por la raíz del árbol, vamos decodificando.

Ahora, voy a crear un **código no instantáneo** para hacer el mismo proceso. Por ejemplo el código:

A → 0

B → 1

C → 00

D → 11

E → 01

Es fácil comprobar que este código no es instantáneo ya que por ejemplo la palabra C empieza como la palabra A.

En este caso no podemos representarlo como un árbol, porque por ejemplo el primer nodo 0 debería ser una A pero sin embargo éste nodo tiene hijos para poder codificar una C. Para decodificar en este caso la cadena *0000011* que es la que equivaldría a *"AACAD"* necesitaríamos algunos bits más de información para definir que carácter decodificar.

Por último, por definición sabemos que **todo código instantáneo es de traducción única**, pero no se cumple la condición en el otro sentido. Por ejemplo para el código instantáneo anterior vemos como cumple que ese código es de traducción única. Es facil verlo ya que el implicar que ninguna palabra codificada pueda ser el comienzo de otra, estás eliminando la incertidumbre a la hora de descodificar ya que cada palabra tiene un comienzo único.

Por otra parte por ejemplo en un conjunto con dos palabras tales $M = \{A, B\}$, $A = \{0, 1\}$ tenemos la siguiente codificación:

$A \to 0$ $\qquad\qquad$ $B \to 01$

Este código es un código de traducción única ya que cualquier cadena perteneciente a este alfabeto que se pueda decodificar únicamente va a tener una decodificación única, sin embargo, tenemos que no es instantáneo poruqe la codificación de la palabra B comienza como la codificación completa de la palabra A.

4. **Explica la desigualdad de Kraft y qué implicaciones tiene en códigos instantáneos. Pon un ejemplo de código que permita codificar 5 mensajes {A, B, C, D, E} y que cumpla la desigualdad de Kraft, y explica qué propiedades tiene. Pon otro ejemplo de código que permita codificar 5 mensajes {A, B, C, D, E} y que no cumpla la desigualdad de Kraft, y explica qué propiedades tiene. Pon un ejemplo de código que permita codificar 5 mensajes {A, B, C, D, E} y que cumpla la igualdad de Kraft, y explica qué propiedades tiene ¿Qué relación guardan los códigos completos con la desigualdad de Kraft? Expón un ejemplo.**

La desigualdad de Kraft nos dice que:

- Sea $M = \{m_1, m_2, ..., m_N\}$ un conjunto de N mensajes a enviar (alfabeto de la fuente). Sea $A = \{a_1, a_2, ..., a_D\}$ los D símbolos a utilizar por un codificador.
- Se desea crear un código con el alfabeto A sobre M cuyas palabras tengan longitudes $n_1, n_2, ..., n_N$ para cada mensaje $m_1, m_2, ..., m_N$.
- Entonces la condición necesaria y suficiente para que tal código exista es:

$$\sum_{i=1}^{N} D^{-n_i} \leq 1$$

Por lo que esto implica que si no se cumple la desigualdad de Kraft, es imposible que exista un código de cdecodificación instantánea con longitudes de palabra $n_1, n_2, ..., n_N$ para los mensajes a codificar, usando el alfabeto A para codificar los mensajes.

Un ejemplo de **código que cumple la desigualdad de Kraft** sería el código instantáneo que tenemos en el ejercicio 3:

A $\rightarrow$ 000

B $\rightarrow$ 001

C $\rightarrow$ 01

D $\rightarrow$ 10

E $\rightarrow$ 11

Como podemos comprobar, tenemos que:

$$\sum_{i=1}^{N} D^{-n_i} = 3 \cdot 2^{-2} + 2 \cdot 2^{-3} = 1 \leq 1$$

Las propiedades que tiene este código es que es completo ya que tenemos que $\sum_{i=1}^{N} D^{-n_i} = 1$ y también que podemos crear un código instantáneo no uniforme con la longitud de palabra que hemos establecido (en este caso ya tenemos el código) por cumplir la desigualdad de Kraft.

Un ejemplo de **código que no cumple la desigualdad de Kraft** sería el siguiente:

A $\rightarrow$ 00

B $\rightarrow$ 01

C $\rightarrow$ 10

D $\rightarrow$ 11

E $\rightarrow$ 000

No cumple la desigualdad ya que:

$$\sum_{i=1}^{N} D^{-n_i} = 4 \cdot 2^{-2} + 2^{-3} = 1{,}125 > 1$$

Esto implica lo contrario del caso anterior, nos dice que no existe un código instantáneo no uniforme que se pueda crear con estos valores que hemos seleccionado, por lo que éste código no es un código instantáneo no uniforme.

Para el ejemplo de la **igualdad de Kraft** podemos fijarnos en el ejemplo que he puesto sobre la desigualdad de Kraft ya que he puesto un ejemplo que cumple la igualdad de Kraft y también explico las propiedades.

La relación que guarda la desigualdad de Kraft con los códigos completos la he comentado en el primer apartado de este ejercicio, es que si la sumatoria de la desigualdad de Kraft es exactamente 1, podemos afirmar que el código es completo.

5. **Considere los siguientes códigos para codificar los mensajes {A, B, C, D, E}:**

   a) **110,1110,0,100,1111**

   b) **111,100,0,101,110**

   c) **10,110,01,111,00**

   d) **10,0,110,111,101**

   Indique qué propiedades cumplen los códigos anteriores, y también cuáles de ellos han podido ser generados mediante los métodos de Shannon-Fano o de Huffman. En caso de encontrar anomalías, indíquelas y explique qué repercusiones tienen.

   a) **110,1110,0,100,1111**

Podemos ver que este código es un código binario, no uniforme, instantáneo, y por lo tanto de traducción única.

Completitud:

$$2^{-1} + 2 \cdot 2^{-3} + 2 \cdot 2^{-4} = 0{,}875$$

Por lo que existe un código de traducción única no completo y estaríamos desaprovechando bits.

No ha podido ser creado con Huffman ni Shannon-Fano porque esta codificación no tiene estructura de árbol.

b) **111,100,0,101,110**

Podemos ver que este código es un código binario, no uniforme, instantáneo, y por lo tanto de traducción única.

Completitud:

$$2^{-1} + 4 \cdot 2^{-3} = 1$$

Por lo que existe un código de traducción única completo con estos parámetros.

Este código si puede ser generado tanto por Huffman como por Shannon-Fano. Tendríamos que saber si el código es óptimo para poder declinarnos más por uno o por otro.

c) **10,110,01,111,00**

Podemos ver que este código es un código binario, no uniforme, instantáneo, y por lo tanto de traducción única.

Completitud:

$$3 \cdot 2^{-2} + 2 \cdot 2^{-3} = 1$$

Por lo que existe un código de traducción única completo con estos parámetros.

Este código si puede ser generado tanto por Huffman como por Shannon-Fano. Tendríamos que saber si el código es óptimo para poder declinarnos más por uno o por otro.

d) **10,0,110,111,101**

Podemos ver que este código es un código binario, no uniforme, no instantáneo, y no de traducción única.

Completitud:

$$2^{-1} + 2^{-2} + 3 \cdot 2^{-3} = 1{,}125$$

Que nos dice que no existe código instantáneo para estos parámetros.

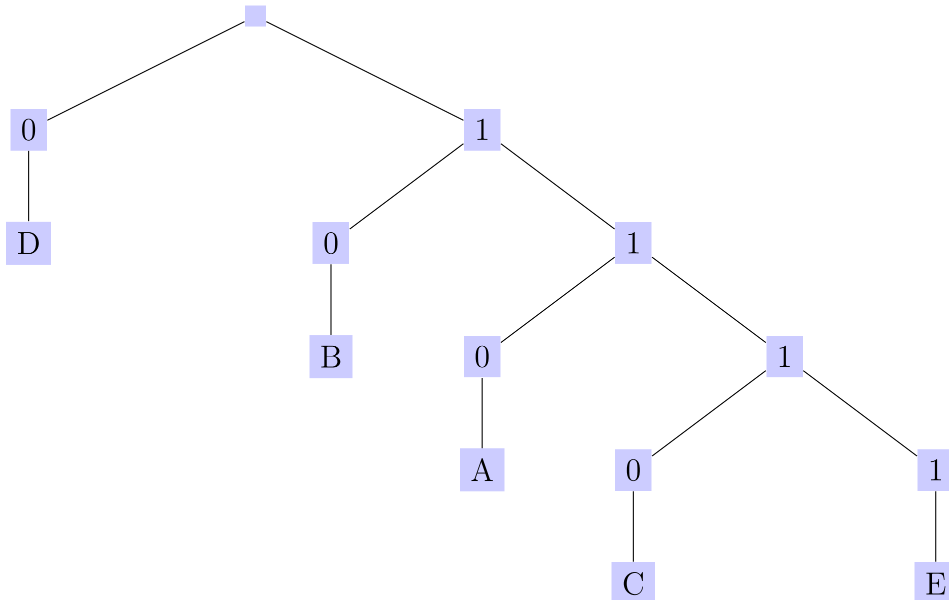
Al no ser un código instantáneao, podemos asegurar que no es un código devuelto ni por Huffman ni por Shannon-Fano.

6. **Se sabe que las longitudes de los códigos para los mensajes m_i en el conjunto{A, B, C, D, E}, tienen longitudes n_i = {3, 2, 4, 1, 4}, respectivamente. Halle un código Huffman compatible con este hecho, y dibuje el árbol de codificación.**

$$2^{-1} + 2^{-2} + 2^{-3} + 2 \cdot 2^{-4} = 1$$

Por lo que estos parámetros permiten obtener un código instantáneo no uniforme completo.

El árbol de Huffman sería el siguiente:



Por lo que tendríamos las siguientes codificaciones **{110,10,1110,0,1111}**

7. **Considerando las probabilidades de ocurrencia de los mensajes siguientes, desarrolle un código Shannon-Fano. Explique el algoritmo según construye el código, y dibuje el árbol de codificación:**

A	B	C	D	E	F	G	H
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{128}$	$\frac{1}{128}$

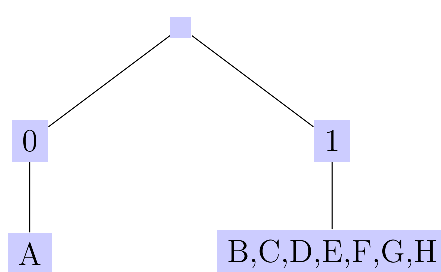
Finalmente, exponga un ejemplo para codificar la cadena de mensajes {HACED}. Explique también, apoyándose con un ejemplo, cómo decodificar la secuencia codificada.

- Como ya tenemos los elementos ordenados, comenzamos dividiendo el conjunto en 2 partes equiprobables:

$$M_1 = \{A\}$$

$$M_2 = \{B, C, D, E, F, G, H\}$$

Obteniendo como árbol:

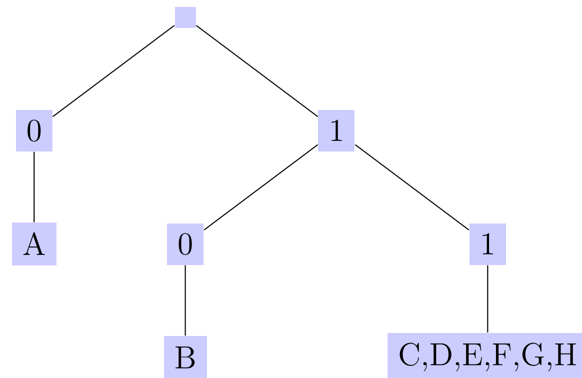


- La parte de M_1 ya la tenemos solucionada, por lo que seguimos haciendo lo mismo con la parte de M_2

$$M_{2_1} = \{B\}$$

$$M_{2_2} = \{C, D, E, F, G, H\}$$

Obteniendo como árbol:

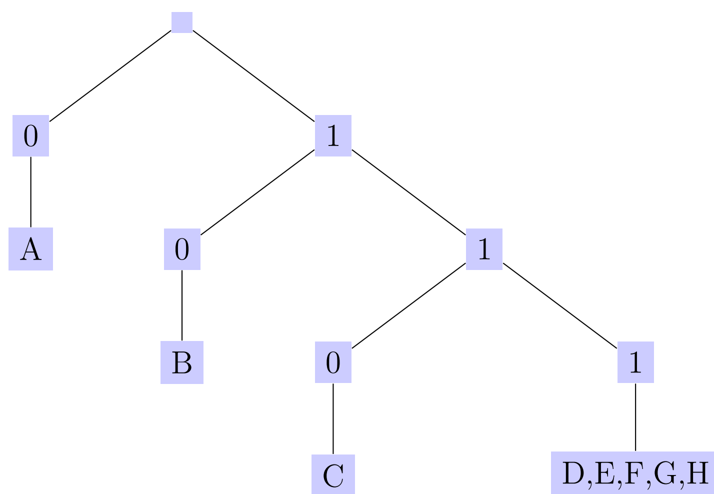


- De nuevo tenemos la parte de M_{2_1} solucionada, por lo que seguimos por M_{2_2}

$$M_{2_{2_1}} = \{C\}$$

$$M_{2_{2_2}} = \{D, E, F, G, H\}$$

Obteniendo como árbol:

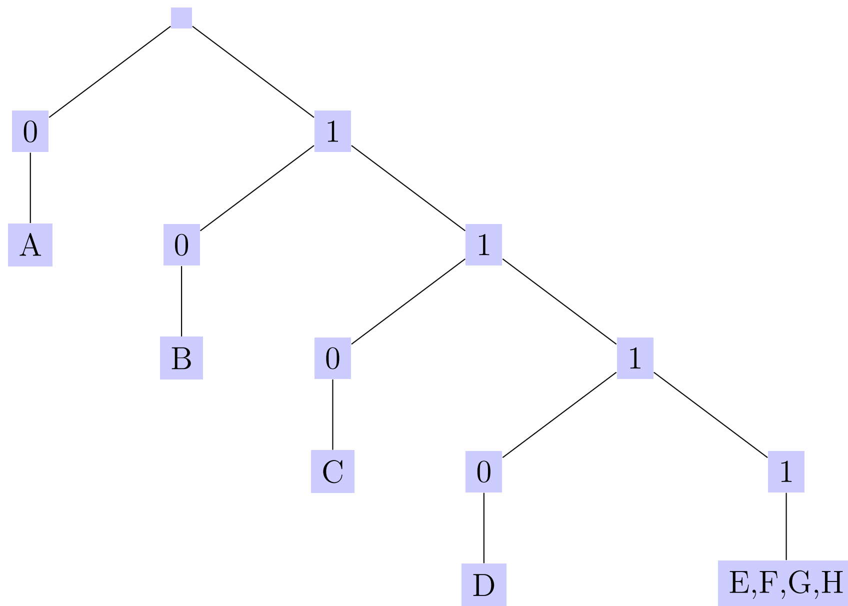


- Como en los casos anteriores volvemos a estar en la misma situación por lo que procedemos igual que antes:

$$M_{2_{2_{2_1}}} = \{D\}$$

$$M_{2_{2_{2_2}}} = \{E, F, G, H\}$$

Obteniendo como árbol:

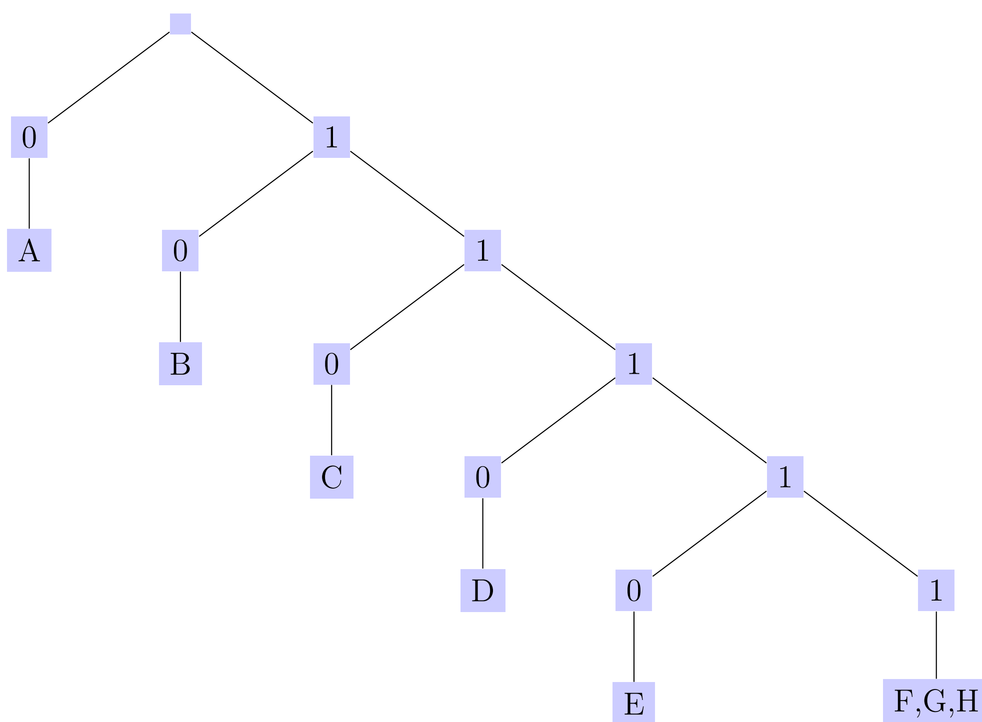


- Seguimos con el mismo proceso:

$$M_{2_{2_{2_1}}} = \{E\}$$

$$M_{2_{2_{2_2}}} = \{F, G, H\}$$

Obteniendo como árbol:

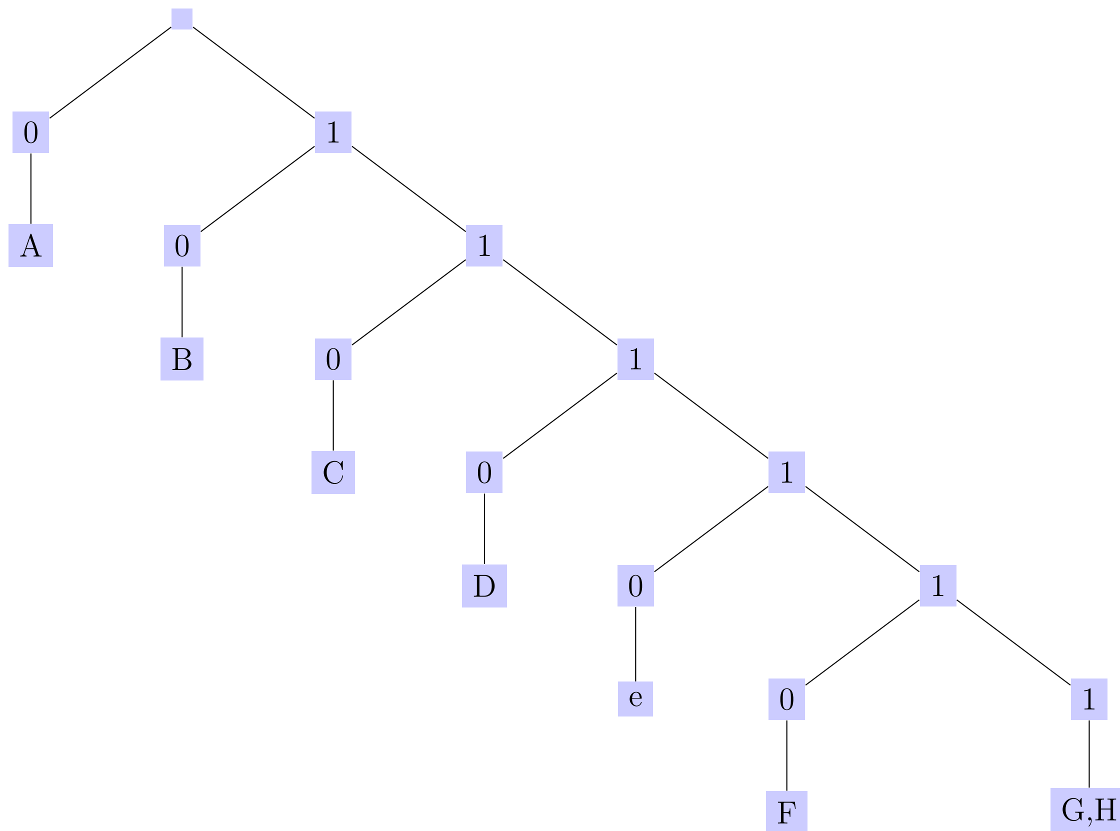


- De nuevo aplicamos el mismo proceso:

$$M_{2_{2_{2_{2_1}}}} = \{F\}$$

$$M_{2_{2_{2_{2_2}}}} = \{G, H\}$$

Obteniendo como árbol:

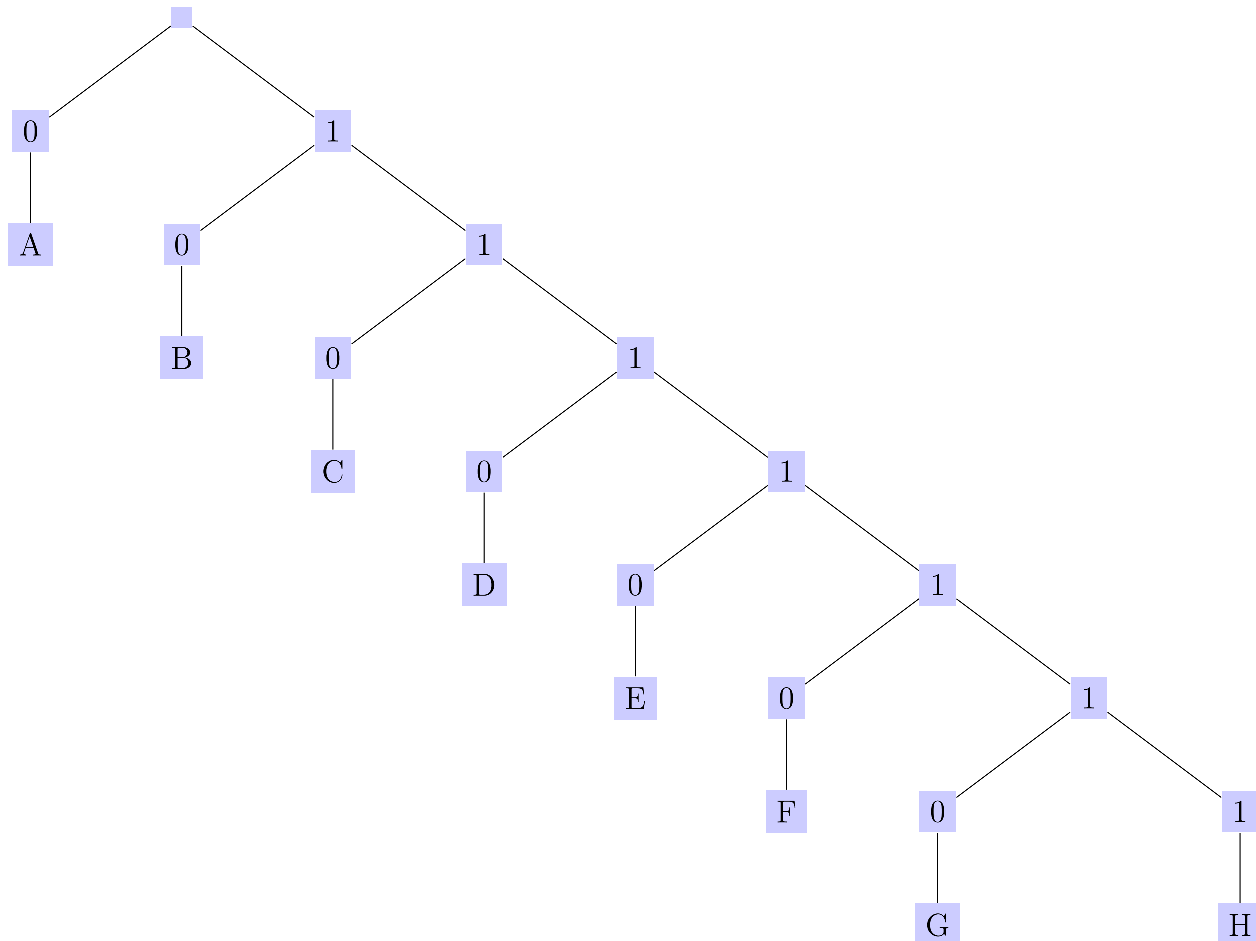


- Por último dividimos el último subconjunto que nos queda, obteniendo:

$$M_{2_{2_{2_{2_{2_{2_1}}}}}} = \{G\}$$

$$M_{2_{2_{2_{2_{2_{2_2}}}}}} = \{H\}$$

Obteniendo como árbol final:



Para **codificar** la cadena {HACED} lo que tenemos que hacer es partiendo de la letra a codificar en el árbol, vamos ascendiendo sobre el árbol hasta llegar a la raíz. Una vez en la raíz tenemos la codificación de esa palabra en sentido contrario, por lo que la rotamos y ya tenemos la codificación correcta. Por ejemplo para la letra *H*, empezaríamos en la *H* y vamos subiendo por el árbol hasta la raíz, obteniendo 1111111 como codificación, lo único que hay que hacer ahora es rotarla y tendremos la codificación correcta (aunque para este caso no se modifique hay que hacerlo).

Por lo que después de hacer eso concada un o de los elementos de la cadena, obtemos que su codificación es **1111111011011110111 0**.

Para decodificar esta cadena, el proceso es el inverso. Comenzando desde la raíz, vamos cogiendo cada uno de los bits y vamos avanzando por el árbol hasta llegar a un nodo hoja, cuando esto suceda se descodifica ese carácter y empezamos de nuevo desde la raíz con los carácteres que queden. Como se puede comprobar, si realizamos el proceso con la cadena codificada obtenemos de nuevo la cadena original.

8. **Considerando las probabilidades de ocurrencia dadas en el ejercicio anterior, desarrolle un código Huffman. Explique el algoritmo según construye el código, y dibuje el árbol de codificación.**

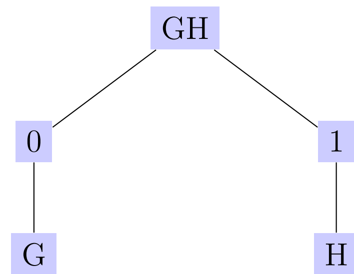
   - Como ya tenemos los elementos ordenados, pasamos a compactar el conjunto que tenemos:

$$M' = \{A, B, C, D, E, F, GH\} \text{ con } \{0.5,0.25,0.125,0.0625,0.03125,0.015625,0.015625\}$$

$$G \rightarrow 0 \; H \rightarrow 1$$

Como ya se quedan bien ordenados, no tenemos que reordenar.

Obteniendo como árbol:



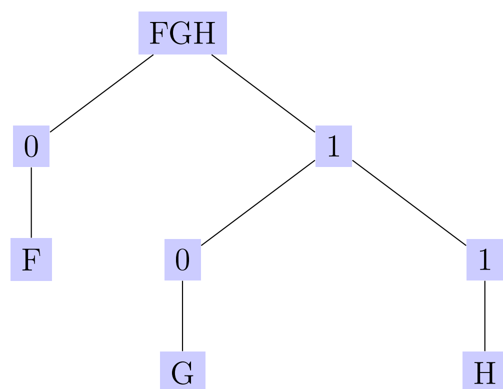
- Volvemos ha realizar el mismo paso.

$$M' = \{A, B, C, D, E, FGH\} \text{ con } \{0{,}5, 0{,}25, 0{,}125, 0{,}0625, 0{,}03125, 0{,}03125\}$$

$$F \rightarrow 0 \; GH \rightarrow 1$$

Como ya se quedan bien ordenados, no tenemos que reordenar.

Obteniendo como árbol:



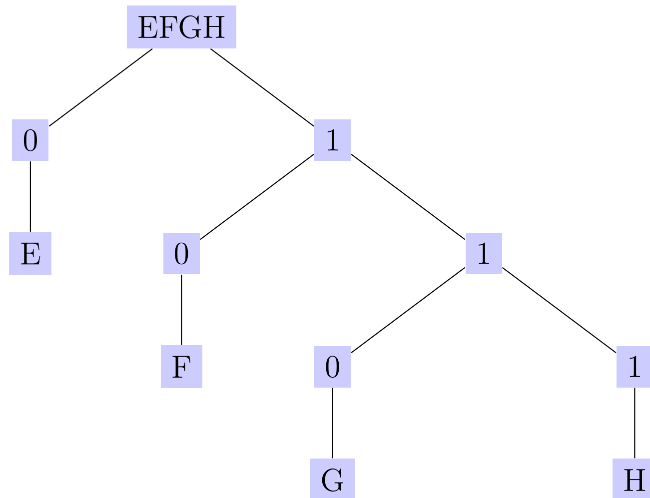
- Volvemos ha realizar el mismo paso.

$$M' = \{A, B, C, D, EFGH\} \text{ con } \{0{,}5, 0{,}25, 0{,}125, 0{,}0625, 0{,}0625\}$$

$$E \rightarrow 0 \; FGH \rightarrow 1$$

Como ya se quedan bien ordenados, no tenemos que reordenar.

Obteniendo como árbol:

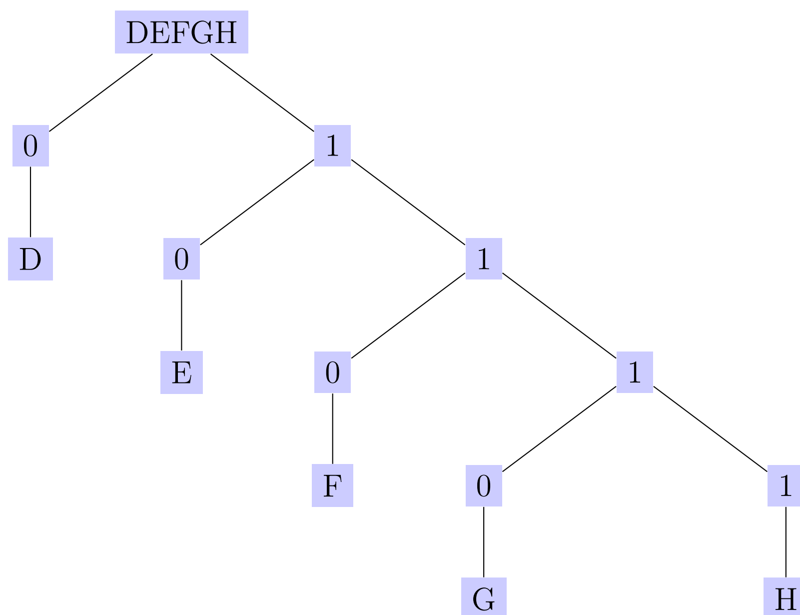


- Volvemos ha realizar el mismo paso.

$$M' = \{A, B, C, DEFGH\} \text{ con } \{0{,}5, 0{,}25, 0{,}125, 0{,}125\}$$

$$D \to 0 \; EFGH \to 1$$

Obteniendo como árbol:



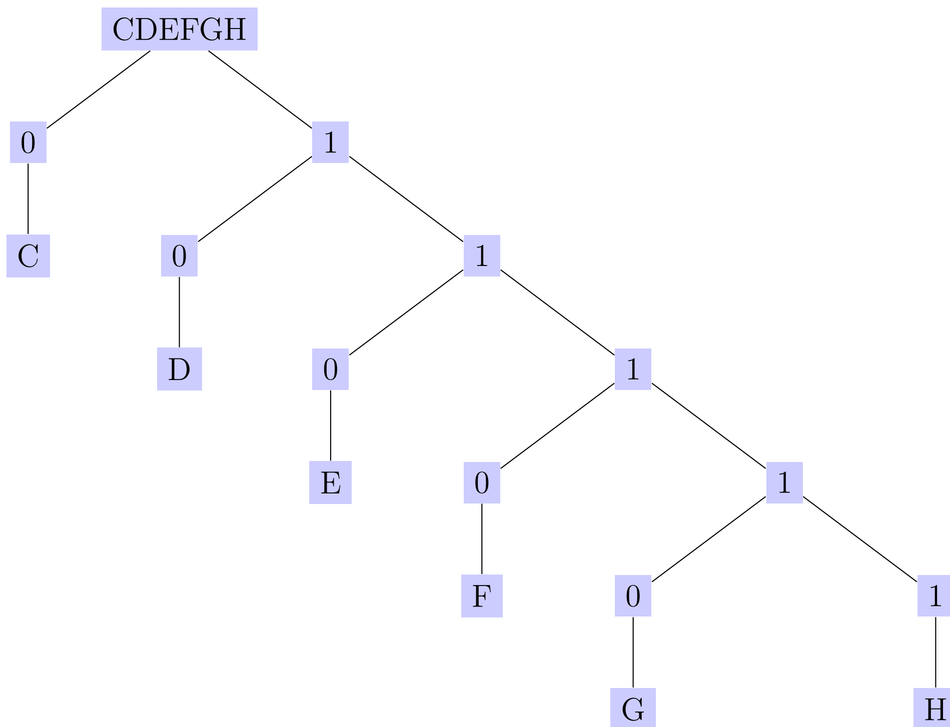
- Volvemos ha realizar el mismo paso.

$$M' = \{A, B, CDEFGH\} \text{ con } \{0{,}5, 0{,}25, 0{,}25\}$$

$$C \to 0 \; DEFGH \to 1$$

Como ya se quedan bien ordenados, no tenemos que reordenar.

Obteniendo como árbol:



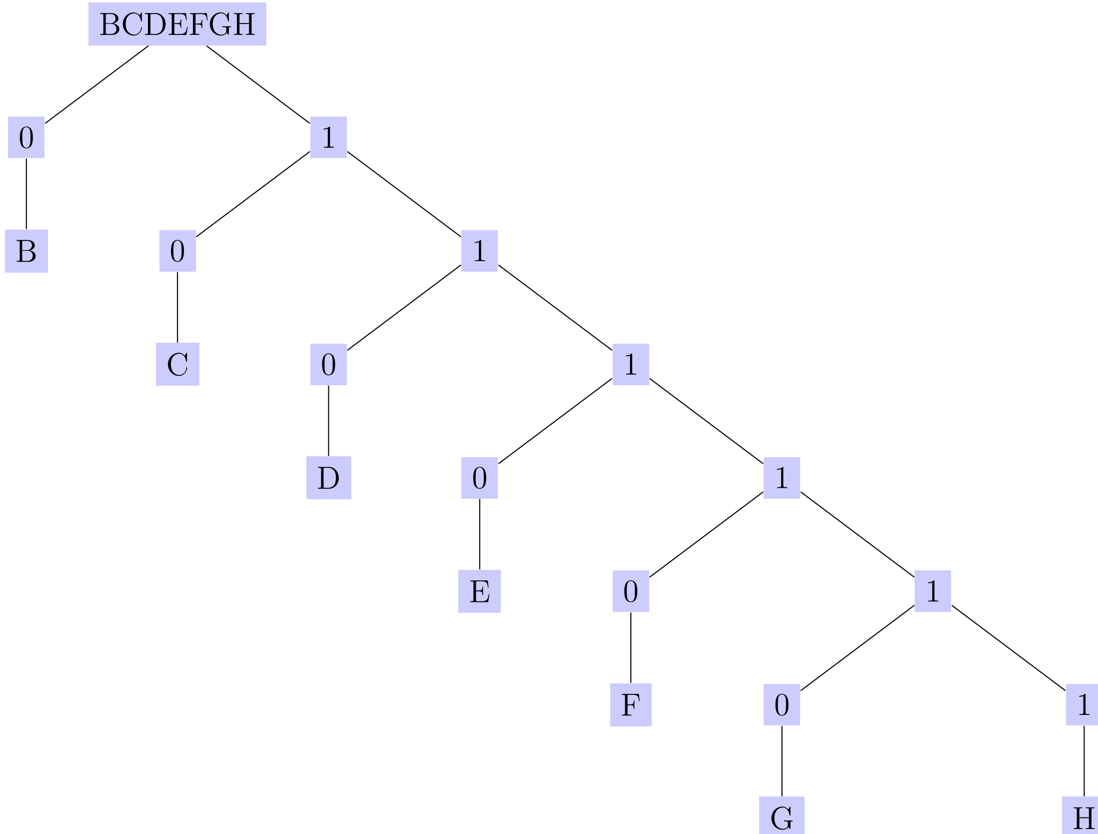
- Volvemos ha realizar el mismo paso.

$$M' = \{A, BCDEFGH\} \text{ con } \{0{,}5, 0{,}5\}$$

$$B \to 0 \; CDEFGH \to 1$$

Como ya se quedan bien ordenados, no tenemos que reordenar.

Obteniendo como árbol:



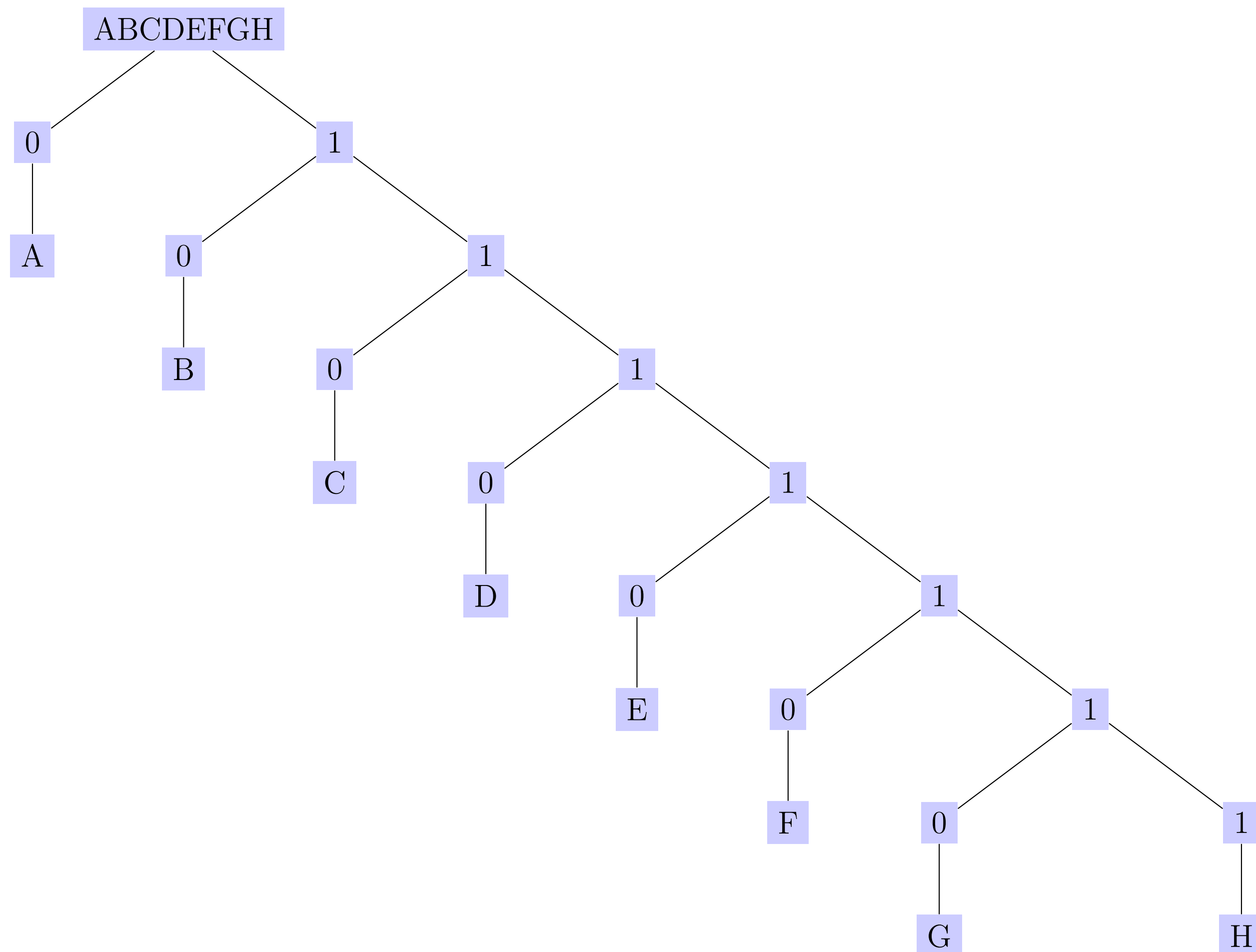
- Volvemos ha realizar el mismo paso.

$$M' = \{ABCDEFGH\} \text{ con } \{1\}$$

$$A \to 0 \; BCDEFGH \to 1$$

Como ya se quedan bien ordenados, no tenemos que reordenar.

Obteniendo como árbol final:



9. **Exponga un ejemplo de distribución de probabilidades para los mensajes {A, B, C, D, E, F} que, aplicando el método de Shannon-Fano, no proporcionen un código óptimo. Exponga también el código óptimo que se generaría utilizando codificación Huffman.**

10. **Atendiendo a los siguientes mensajes, y sus probabilidades de generación por la fuente:**

$$\{\ a\ ,\ b\ ,\ c\ ,\ d\ ,\ e\ ,\ f\ ,\ g\ \}$$
$$\{0.01, 0.24, 0.05, 0.20, 0.47, 0.01, 0.02\}$$

Explique cómo generar un código de Shannon-Fano y cree su árbol de codificación asociado. Explique cómo generar un código Huffman y cree su árbol de codificación asociado. ¿Son igualmente óptimos ambos códigos? Justifique su respuesta. Codifique y decodifique la secuencia de mensajes {bacafeg} con ambos métodos, explicando el procedimiento.

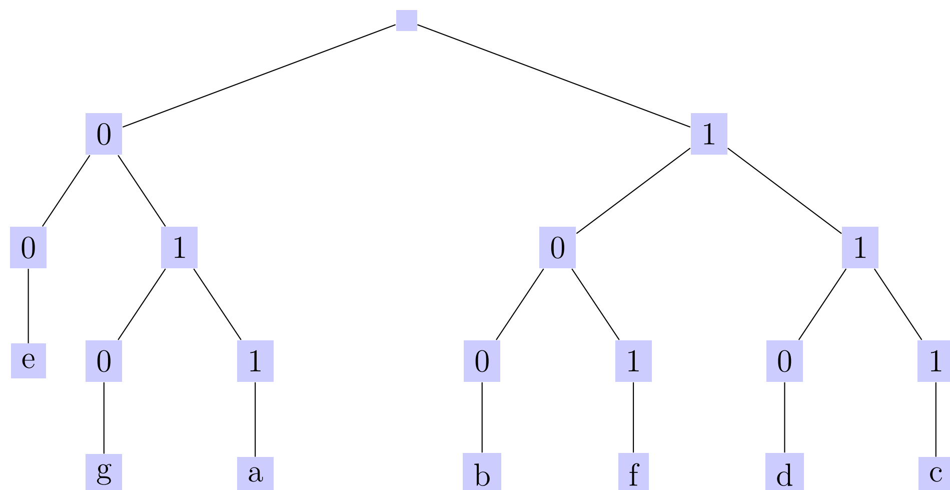
Como hemos visto en el ejercicio 8, siguiendo ese procedimiento podemos crear tanto el árbol Huffman como el árbol Shannon-Fano, asique como el procedimiento está bien explicado, pondré solo los algoritmos y el arbol resultante.

- **Shannon-Fano**:
    - Ordenamos los mensajes m_i en orden decreciente de probabilidades p_i. Llamaremos

M' al conjunto ordenado.

- Escoger el punto k de M' tal que divida M' en 2 partes equiprobables, denominadas $M'1$ y $M'2$.
- Asignar a_1 a M'_1 y a_2 a M'_2.
- Volver a realizar la operación desde el punto 2 sobre los subconjuntos generados M'_1 y M'_2, hasta que el número de símbolos en cada subconjunto sea 1.

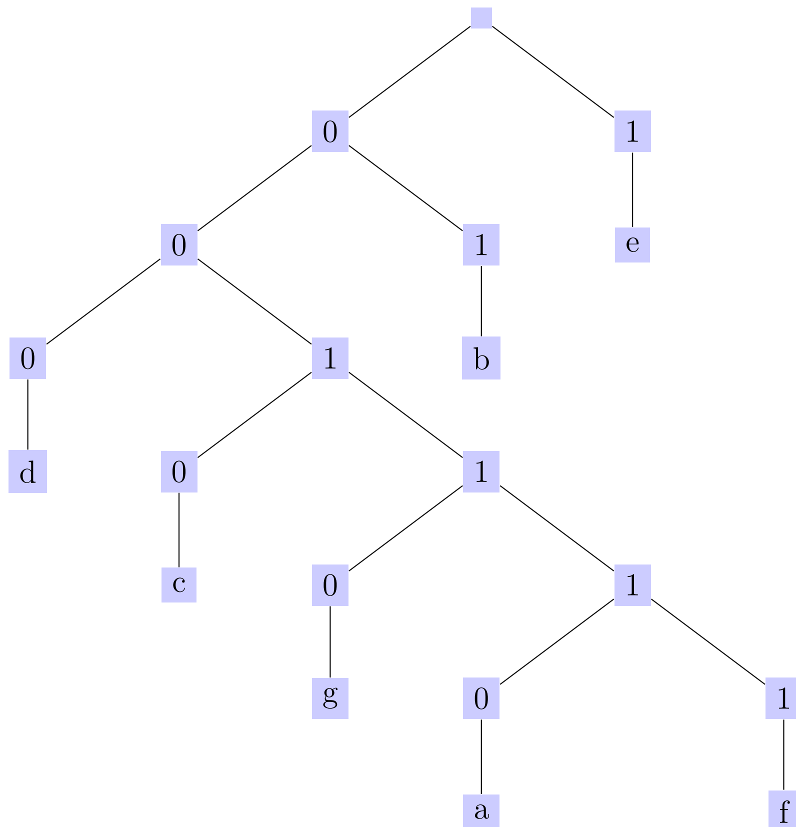
Obtenemos como árbol:



Podemos ver como Shannon-Fano nos devuelve un árbol donde el único elemento con preferencia es el elemento *"e"*, ya que el resto de elementos están todos al mismo nivel. Esto nos muestra que Shannon-Fano no nos ha creado un código óptimo comparado con Huffman que lo veremos en el siguiente apartado.

- **Huffman**:
  - Ordenamos los mensajes m_i en orden decreciente de probabilidades p_i. Llamaremos M' al conjunto ordenado.
  - Los dos últimos símbolos (de menos probabilidad) se agrupan en uno sólo, asignando 0 al primero y 1 al segundo.
  - Se reordenan las probabilidades de nuevo, y se vuelve al paso 2 hasta que sólo quede un símbolo.

  Se obtiene como árbol lo siguiente:



Podemos comprobar que Huffman si nos crea un código óptimo y dependiendo de las probabilidades conseguimos que el código codificado sea de más o menos longitud.

Para codificar y decodificar ya he explicado el procedimiento en ejercicios anteriores, para codificar tenemos que emepando desde la letra, ir subiendo hasta la raiz y cuando lleguemos a la raiz ya tendríamos el valor codificado para este símbolo. Lo único que tenemos que hacer posteriormente es rotarlo para tenerlo en el orden adeacuado.

Por otra parte para decodificar tenemos que ir pasando cada uno de los bits e ir avanzando desde la raiz del aŕbol hacia abajo hasta llegar a una hoja. Cuando lleguemos a una hoja ya tenemos decodificado un elemento, vovemos a la raíz y seguimos por donde nos habíamos quedado en la cadena codificada.

Por lo que una vez explicado como se realiza el procedimiento, podemos decir que la cadena {bacafeg} tiene la codificación **100011111011110100010 en Shannon-Fano** mientras que tendría la codificación **0100111000100011100011111100110 en Huffman**. Podemos ver como la cadena de Huffman es más larga que la de Shannon-Fano, esto se debe principalmente a que el mensaje que estamos codificando tiene más letras que se suponen muy poco probables y a penas ninguna de las que se suponen más probables.

Siguiendo el procedimiento indicando para decodificar, si utilizamos las cadenas descritas anteriormente, podemos comprobar como volvemos a obtener la cadena del principio.

11. **Sea una fuente S capaz de generar 4 símbolos, con las siguientes probabilidades:** $S = [P(A) = 0{,}4; P(B) = 0{,}3; P(C) = 0{,}2; P(D) = 0{,}1]$

    Indique, entre los siguientes códigos, cuáles son instantáneos, cuáles unívocamente decodificables, cuáles completos y cuáles tienen mejor rendimiento (H/longi-

tud promedia):

a) **Código 1: A = 001 ; B = 01 ; C = 11 ; D = 010**

Tenemos un código no instantáneo, unívocamente decodificable.

Completitud:

$$2 \cdot 2^{-2} + 2 \cdot 2^{-3} = 0{,}75$$

Por lo que con estos parámetros puede ser completo.

Rendimiento:

$$H = 1{,}85$$

$$rendimiento = \frac{1{,}85}{2{,}5} = 0{,}74$$

Este código tiene el mismo rendimiento que el tercero pero peor que el segundo.

b) **Código 2 : A = 0 ; B = 01 ; C = 011 ; D = 111**

Tenemos un código no instantáneo, unívocamente decodificable.

Completitud:

$$2^{-1} + 2^{-2} + 2 \cdot 2^{-3} = 1$$

El código no es instanstáneo por lo que esto nos dice que existe un código completo e instantáneo con estos parámetros, pero éste no lo es.

Rendimiento:

$$H = 1{,}85$$

$$rendimiento = \frac{1{,}85}{2{,}25} = 0{,}82$$

En rendimiento este código es mejor que los otros 2.

c) **Código 3 : A = 1 ; B = 01 ; C = 001 ; D = 0001** Tenemos un código instantáneo, y por lo tanto unívocamente decodificable.

Completitud:

$$2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} = 0{,}9375$$

Por lo que el código no es completo.

Rendimiento:

$$H = 1{,}85$$

$$rendimiento = \frac{1{,}85}{2{,}5} = 0{,}74$$

Por lo que en rendimiento este es igual que el primero pero ambos peor que el segundo.

12. **Sea una fuente S capaz de generar 4 símbolos, con las siguientes probabilidades:**

Esta fuente genera la siguiente secuencia de mensajes: AABAAADADDAAAAAB

a) **Desarrolle un código uniforme para codificar los mensajes de S. Demuestre si el código es completo. Explique cómo codificar la secuencia de mensajes dada con este código, y expóngalo como ejemplo. Explique también el método de decodificación, haciendo uso del mensaje codificado para su decodificación.**

b) **Desarrolle un código Huffman para codificar los mensajes de S. Demuestre si el código es completo. Explique cómo codificar la secuencia de mensajes dada con este código, y expóngalo como ejemplo. Explique también el método de decodificación, haciendo uso del mensaje codificado para su decodificación. Indique, en comparación con el método de codificación uniforme, cuál es la variación de eficiencia de ambos códigos.**