UNIVERSIDAD NACIONAL DE COLOMBIA

UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE CIENCIAS
TEORIA DE CODIFICACIÓN

Edgar Santiago Ochoa Quiroga
María Alejandra Rodríguez Ríos

Ejercicio 1
Suponga que una fuente genera dígitos binarios con distribución uniforme, los mensajes son enviados a través de un canal cambia los símbolos que genera una fuente binaria con las probalidades que se muestran en la gráfica.
poner foto Responda cada una de las siguientes preguntas justificando su razonamiento:
a). ¿Cuál es el número más probable de errores que se puede encontrar en un mensaje de 5000 bits que pase por el canal?
b). Si se codifica cada bit que genera la fuente triplicandolo, en qué porcentaje se reduce el número de errores? Suponga que para la decodificación se usa mayoría de bits por tripla.
c). Si se codifica cada bit que genera la fuente con una n-tupla y se usa mayoría de bit por n tupla en la decodificación, cuál sería la longitud mínima n - tupla para obtener por lo menos un 95 % de certeza en la decondificación?

Solución. Hola :p

Ejercicio 2
Se codifican los bits de una fuente de acuerdo a la siguiente función

$$\mathrm{Cod}(x_1,x_2,x_3,x_4,x_5) = (x_1,x_2,x_3,x_4,x_5,x_1{+}x_2{+}x_4,x_1{+}x_3{+}x_4,x_2{+}x_3{+}x_4,x_1{+}x_3{+}x_5,x_3{+}x_4{+}x_5)$$

Si el mensaje recibido es 1101110101 determine si hay un error en el mensaje y si es posible corregir el error . Use diagramas de Ven o el método de Galager para llegar a la solución. Deje todos los cálculos que le llevan a la solución en su hoja de respuesta.

Solución. Hola :p

Ejercicio 3
Clasifique los siguientes códigos de tal manera que puedan codificar la misma fuente. Luego indique por fuente cuál sería el mejor código. Justifique todas sus afirmaciones.

- $C_1 = \{0,1\}$
- $C_2 = \{00,01,10\}$
- $C_3 = \{000,111\}$
- $C_4 = \{000,011,101,110,001,100\}$
- $C_5 = \{00000,11111,22222\}$
- $C_6 = \{001,010,012,021,100,221\}$

Solución. Hola :p

Ejercicio 4
Determine justificando, si los códigos $C_1 = \{1, 10, 100, 1000\}$ y $C_2 = \{1, 01, 001, 0001\}$ son univocamente decodificables.

Solución. Procederemos con cada código por separado

- Sea $C = \{1, 10, 100, 1000\}$., Donde nuestro alfabeto para codificar es $D = \{0, 1\}$ Procedamos por medio se Sardinas-Patterson. Primero definimos $C_0 = C$, por el algoritmo tenemos que

$$C_1 = \{w \in D^+ | \, uw = v, \text{ con } u \in C_0 \text{ y } v \in C \text{ o } u \in C \text{ y } v \in C_0\}.$$

Note que como $C_0 = C$ Basta con resolver ecuaciones de concatenación de palabras donde solo revisamos los elementos de C. Procedemos por casos.

Caso 1: Si $u = 1000$, tenemos las siguientes cuatro ecuaciones

$$1000w = \begin{cases} 1 \\ 10 \\ 100 \\ 1000 \end{cases}$$

Note que $|1000w| \geq 5$ ya que $|w| \geq 1$. Pero las longitudes de las palabras al otro lado de la igualdad son a lo mas 4, por lo que estas ecuaciones no tienen solución.

Caso 2: Si $u = 100$, tenemos las siguientes cuatro ecuaciones

$$100w = \begin{cases} 1 \\ 10 \\ 100 \\ 1000 \end{cases}$$

Note que para las primeras tres al igual que en el caso anterior no hay solución simplemente viendo las longitudes, pero para la ultima tomando $w = 0$, tenemos nuestra primera solución.

Caso 3: Si $u = 10$, tenemos las siguientes cuatro ecuaciones

$$10w = \begin{cases} 1 \\ 10 \\ 100 \\ 1000 \end{cases}$$

Siguiendo la lógica de los anteriores casos nos damos cuenta que $w = 00$ es solución de la cuarta ecuación, mientras que $w = 0$ de la tercera y las otras dos no tienen.

Caso 4: Si $u = 1$, tenemos las siguientes cuatro ecuaciones

$$1w = \begin{cases} 1 \\ 10 \\ 100 \\ 1000 \end{cases}$$

Siguiendo la lógica de los anteriores casos nos damos cuenta que $w = 000$ es solución de la cuarta ecuación, mientras que $w = 00$ de la tercera y $w = 0$ de la segunda, mientras que la primera no tiene solución.

Así tenemos que $C_1 = \{0, 00, 000\}$. Siguiendo el algoritmo tenemos que

$$C_2 = \{w \in D^+ | uw = v, \text{ con } u \in C_1 \text{ y } v \in C \text{ o } u \in C \text{ y } v \in C_1\}.$$

Veamos los posibles casos.

Caso 1: Si $u \in C_1$ y $v \in C$, note que la palabra uw siempre empieza en 0, ya que u es igual a $0, 00$ o 000, mientras que todas las palabras v en C empiezan en 1, así ninguna ecuación tiene solución.

Caso 2: Si $u \in C$ y $v \in C_1$, pasa igual que en el caso anterior. Todas la posibles palabras uw empiezan por 1, ya que cualquier palabra en C empieza por uno, mientras que todas las palabras en C_1 empiezan por 0, así ninguna ecuación tiene solución. Así $C_2 = \varnothing$.

Por la construcción podemos concluir que $C_3 = C_4 = \cdots = \varnothing$. Por lo tanto

$$C_\infty = \bigcup_{i=1}^{\infty} C_i = C_1.$$

Así podemos notar que

$$C \cap C_\infty = \{1, 10, 100, 1000\} \cap \{0, 00, 000\} = \varnothing.$$

Concluyendo por el teorema de Sardinas-Patterson que C es unívocamente decodificable.

- Sea $C = \{1, 01, 001, 0001\}$, donde nuestro alfabeto para codificar es $D = \{0, 1\}$. Siguiendo el algoritmo, tomamos $C_0 = C$, siguiendo el algoritmo tenemos que

$$C_1 = \{w \in D^+ | uw = v, \text{ con } u \in C_0 \text{ y } v \in C \text{ o } u \in C \text{ y } v \in C_0\}.$$

Veamos los casos.

Caso 1: Sea $u = 0001$, Note que $|uw| \geq 5$, y las posibles longitudes para $v \in C$ son menores o iguales a 4, luego para este u, las ecuaciones no tienen solución.

Caso 2: Sea $u = 001$, note que por el argumento de las longitudes la única ecuación que se podría considerar es

$$001w = 0001.$$

Pero note que el tercer símbolo de izquierda a derecha de la palabra de la izquierda es 1, mientras que el de la derecha es 0, por lo que ninguna ecuación tiene solución.

Caso 3: Sea $u = 01$, nuevamente por el argumento de las longitudes las ecuaciones que valen la pena considerar son

$$01w = \begin{cases} 001 \\ 0001 \end{cases}$$

Pero nuevamente el segundo símbolo de izquierda a derecha de la palabra de la izquierda es 1, mientras que en ambas palabras de la derecha el símbolo es 0, así para estas ecuaciones nuevamente no tenemos soluciones.

Caso 4: Para este ultimo caso, siguiendo la misma idea de las longitudes, cuando $u = 1$ basta con considerar las siguientes ecuaciones

$$1w = \begin{cases} 01 \\ 001 \\ 0001 \end{cases}$$

Pero la palabra de la izquierda empieza en 1, mientras que las de la derecha en 0, por lo que nuevamente no ha soluciones.

Como en ningún caso existe w que cumpla las ecuaciones, tenemos que $C_1 = \varnothing$. Luego tenemos que $C_2 = C_3 = \cdots = \varnothing$. De esta manera tenemos que

$$C_\infty = \bigcup_{i=1}^{\infty} C_i = \varnothing.$$

Luego es claro que

$$C \cap C_\infty = \varnothing.$$

Así por el teorema de Sardinas-Patterson podemos concluir que C es unívocamente decodificable.

De esta manera concluimos por medio de Sardinas-Patterson que los C_1 y C_2, presentados en el enunciado, son unívocamente decodificables.

$\cup_{\vee}\cup$

Ejercicio 5

Pruebe, diseñando un algoritmo (sin usar el algoritmo de SardinasPaterson), que el código triario $\{aa, b, ba, abc\}$ es univocamente decodificable

Solución. Hola :p

Ejercicio 6
Determine si el código triario $C = \{ab, cb, abbc, cbc, abb\}$ es unívocamente decodificable. Existe un código binario instantáneo con las longitudes de palabras de C ? Si existe construyalo.

Solución. Procedamos por medio del algoritmo. Al igual que hicimos el el punto 4 cuando consideremos ecuaciones, solo tendremos en cuenta los casos con longitudes de palabra que al menos puedan coincidir. Definimos $C_0 = C$, ahora por definición

$$C_1 = \{w \in D^+ | uw = v, \text{ con } u \in C_0 \text{ y } v \in C \text{ o } u \in C \text{ y } v \in C_0\}.$$

Por lo que tenemos los siguientes casos:

- Si $u = ab$, las ecuaciones con longitudes de palabra coherente son

$$abw = \begin{cases} abbc \\ cbc \\ abb \end{cases}$$

  Luego posibles soluciones para w son bc en la primera y b en la ultima, la segunda ecuación note que la palabra de la izquierda empieza por a mientras que la derecha por c, así no hay solución.

- Si $u = cb$, por las longitudes de palabra consideramos solamente

$$cbw = \begin{cases} abbc \\ cbc \\ abb \end{cases}$$

  Note que la primera ecuación y la ultima no tienen solución ya que la palabra de la izquierda empieza por c mientras la de la derecha por a. Pero note que para la segunda si tomamos $w = c$, cumplimos la ecuación

- Si $u = cbc$, la única ecuación por longitudes es $cbcw = abbc$, pero esta no tiene solución por que la palabra de la izquierda empieza por c, mientras la de la derecha empieza por a. En cambio si $u = abb$, la ecuación $abbw = abbc$ tiene como solución $w = c$.

Así con todos los casos inspeccionados tenemos que $C_1 = \{b, c, bc\}$. Ahora tenemos que

$$C_2 = \{w \in D^+ | uw = v, \text{ con } u \in C_1 \text{ y } v \in C \text{ o } u \in C \text{ y } v \in C_1\}.$$

Note que por las longitudes de palabra las ecuaciones de la forma $uw = v$, donde $u \in C$ y $v \in C_1$, ya que $|uw| \geq 3$ mientras que $|v| \leq 2$. Así que solo observaremos cuando $u \in C_1$ y $c \in C$.

- Note que para los casos donde $u = b$ y $u = bc$, la palabra uw empieza por b, mientras que todas las palabras del código C empiezan por a o c, así ninguna ecuación tiene solución en este caso.

- Si $u = c$, basta considerar las ecuaciones que empiezan por ese símbolo, es decir, $cw = cb$ y $cw = cbc$. Luego $w = b$ o $w = bc$.

De esta manera concluimos que $C_2 = \{b, bc\}$. Note ahora que para C_3 podemos argumentar de manera muy similar a C_2. Para el caso donde $u \in C_2$ y $v \in C$ note que $u = b$ o $u = bc$, luego uw empieza por b, pero ninguna palabra en C empieza por b, así no hay soluciones. En caso de que $u \in C$ y $v \in C_2$, tenemos que $|uw| \geq 3$, mientras que $|v| \leq 2$. Así concluimos que $C_3 = C_4 = \cdots = \varnothing$. Luego

$$C_\infty = \bigcup_{i=1}^{\infty} C_i = C_1 \cup C_2 = \{b, c, bc\}.$$

De esta manera $C \cap C_\infty = \varnothing$, y por el teorema de Sardinas-Patterson tenemos que C es unívocamente decodificable.

Ahora queremos ver si existe un código binario, es decir, $D = 2$, con longitudes de palabras $\ell_1 = 2, \ell_2 = 2, \ell_3 = 3, \ell_4 = 3$ y $\ell_5 = 4$. Que sea instantáneo, esto lo podemos ver por medio de la desigualdad de Kraft

$$\begin{aligned}\sum_{i=1}^{5} D^{-\ell_i} &= 2^{-2} + 2^{-2} + 2^{-3} + +2^{-3} + +2^{-4} \\ &= 2^{-4}(2^2 + 2^2 + 2 + 2 + 1) \\ &\frac{4+4+2+2+1}{16} \\ &= \frac{13}{16} \leq 1\end{aligned}$$

Luego tal código si existe, por lo tanto podemos construirlo. Mañana hago este grafo ya tengo sueñito ᗢ˯ᗢ

Ejercicio 7
Una fuente genera símbolos con una distribución de probabilidad

$$\{0{,}729, 0{,}081, 0{,}081, 0{,}081, 0{,}009, 0{,}009, 0{,}009, 0{,}001\}.$$

Determine un código binario que permita calcular la codificación de los símbolos de la fuente con menor longitud promedio. Cálcule la eficiencia del código.

Solución. Hola :p

Ejercicio 8
Diseñe un código de Fano para una fuente con la siguiente distribución de probabilidad

$$\{0{,}20, 0{,}19, 0{,}18, 0{,}17, 0{,}15, 0, 10, 0{,}01\}$$

.

Solución. Hola :p

Ejercicio 9
Resuelva el ejercicio 1.5.1 de las notas de clase. Construya un código que no sea instantáneo cuyas longitudes de palabra cumplen la desigualdad de Kraft.

Solución. Sea $C = \{1, 10, 100, 1000\}$. Este codigo claramente no es instantaneo, ya que no es prefijo, por que la palabra 1 es segmento inicial de todas la demas palabras del codigo. Note que para este codigo tenemos que $D = 2$ ya que es un codigo binario, y tenemos longitudes de palabra $\ell_1 = 1, \ell_2 = 2, \ell_3 = 3$ y $\ell_4 = 4$. Asi

$$\begin{aligned}\sum_{i=1}^{4} D^{-\ell_i} &= 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} \\ &= 2^{-4}(2^3 + 2^2 + 2 + 1) \\ &= \frac{8+4+2+1}{16} \\ &= \frac{15}{16} \leq 1.\end{aligned}$$

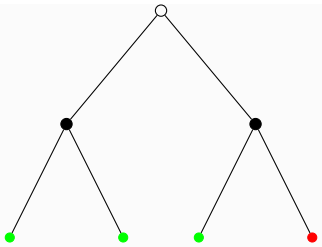
De esta forma C es un codigo no instantaneo que cumple la desigualdad de Kraft como queriamos. ∪˳∪

Ejercicio 10
Resuelva el ejercicio 1.5.2 de la notas de clase, calcule la eficiencia. Construya, en caso de que exista, un código binario instantáneo constituido por 5 palabras de longitudes: 2, 2, 2, 3 y 4.

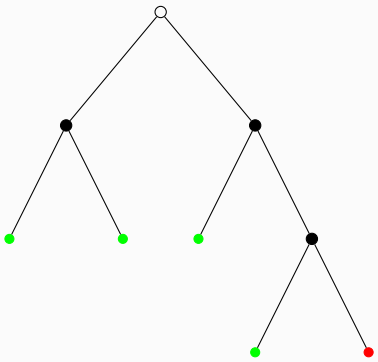
Solución. Primero verifiquemos que tal codigo exista. Tenemos que $D = 2$, por que es binario y tenemos longitudes de palabra $\ell_1 = 2, \ell_2 = 2, \ell_3 = 2, \ell_4 = 3$ y $\ell_5 = 4$. Asi

$$\begin{aligned}\sum_{i=1}^{5} D^{-\ell_i} &= 2^{-2} + 2^{-2} + 2^{-2} + 2^{-3} + 2^{-4} \\ &= 2^{-4}(2^2 + 2^2 + 2^2 + 2 + 1) \\ &= \frac{4+4+4+2+1}{16} \\ &= \frac{15}{16} \leq 1.\end{aligned}$$

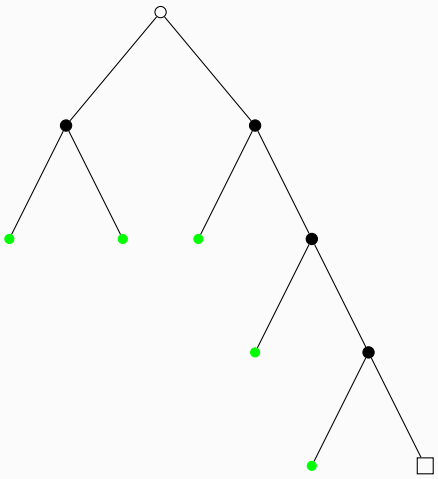
Como cumple la desigualdad de Kraft, sabemos que existe un codigo instantaneo (no completo ya que no se tiene la igualdad), con las longitudes de palabra indicadas. Siguiendo el algoritmo de construccion, tenemos para el paso uno el siguiente arbol de altura 2.



Donde los nodos verdes serán hojas, y el rojo es un nodo terminal. Para el segundo paso aumentamos a un árbol de altura 3 como se muestra a continuación



Para finalizar aumentamos el árbol a una altura 4 como se muestra a continuación



De esta forma si a las ramas que abren hacia la izquierda les asignamos el 0 y a las otras el 1, obtenemos el siguiente código instantáneo con las longitudes de palabra deseadas.

$$C = \{00, 01, 10, 110, 1110\}.$$

∪˙∪

Ejercicio 11

Resuelva el ejercicio 2.3.1 de la página 36 de las notas de clase, calcule la eficiencia en cada caso. Considere el alfabeto S con la distribución de probabilidades que se muestra en la tabla:

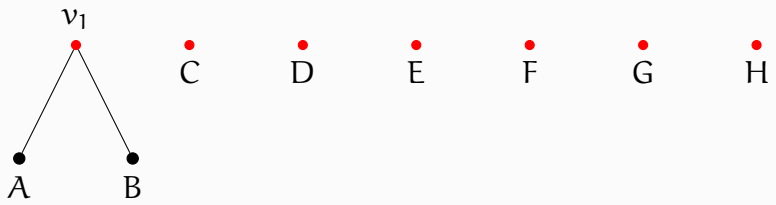
A	B	C	D	E	F	G	H
0,02	0,03	0,04	0,04	0,12	0,20	0,20	0,35

Suponga que se ha usado un código de Huffman para codificar los mensajes sobre un alfabeto binario.

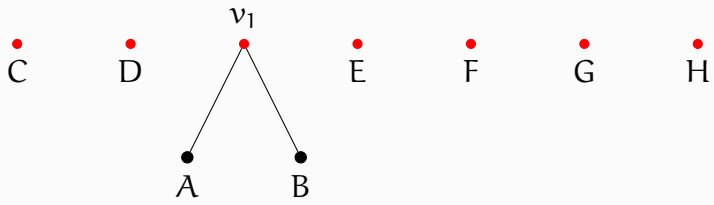
Si en el árbol de codificación se le asigna 1 a las ramas sobre la izquierda y 0 a las ramas sobre la derecha, ¿qué palabra representa la secuencia 111011111101110? Determine la longitud promedio de palabra para este código. ¿Cuál sería la codificación de los símbolos sobre un código triario?

Solución. Primero procederemos para el caso binario por medio del algoritmo de Huffman. Pintemos los respectivos 8 nodos en orden de menor a mayot segun la distribucion de probabilidad.

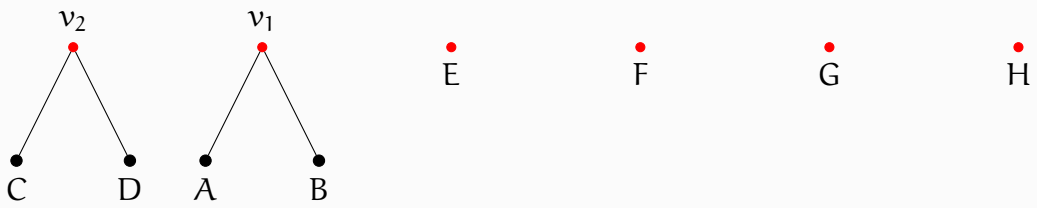
Donde el nodo rojo significa que esta disponible. Como estamos en el caso binario no tenemos que agregar nodos nuevos al inicio y podemos proceder a crear un nodo padre para los dos de menos probabilidad, que en este caso son A y B. Quedando como nodos activos



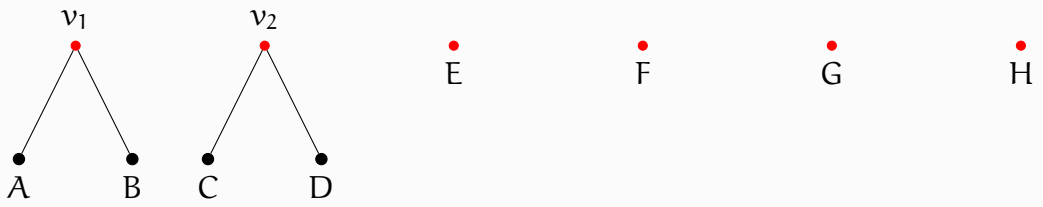
Ahora v_1 queda con las porbabilidades sumadas de A y B, es decir $0,02 + 0,03 = 0,05$. Por el algoritmo tenemos que reorganizar los nodos tal que queden nuevamente en orden de probabilidades, tal que



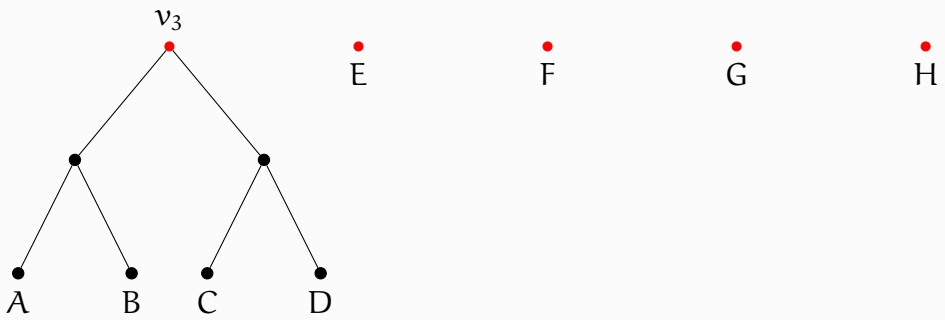
Ahora creamos un nodo padre para los dos de menor probabilidad, es decir C y D, a este lo llamaremos v_2 y tendra las probabilidad sumada, es decir $0,04 + 0,04 = 0,08$.



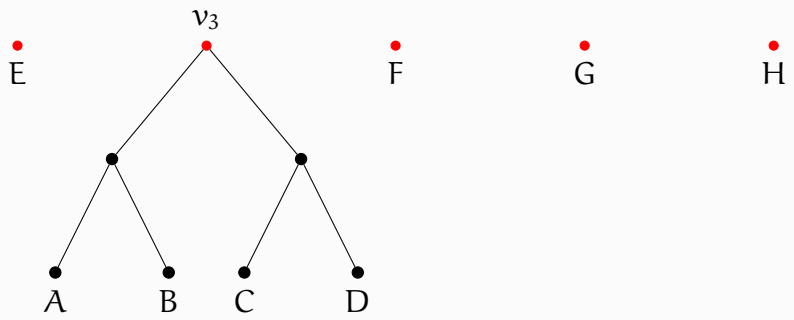
Note que nuevamente hay que reorganizar los nodos ya que la probabilidad de v_1 es menor a la de v_2.



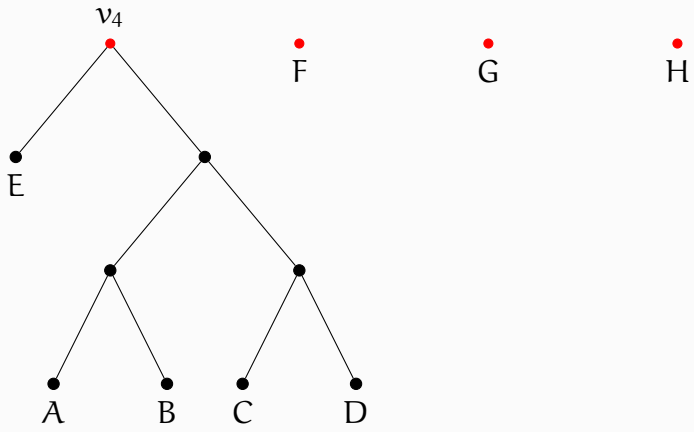
Y siguiendo el algoritmo ponemos un padre a v_1 y v_2, con probabilidad $0,05 + 0,08 = 0,13$.



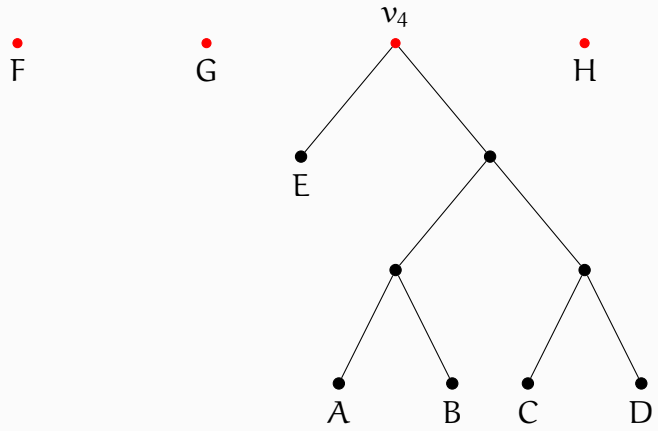
Ahora como la probabilidad de v_3 es mayor a la de E hay que cambiar el orden.



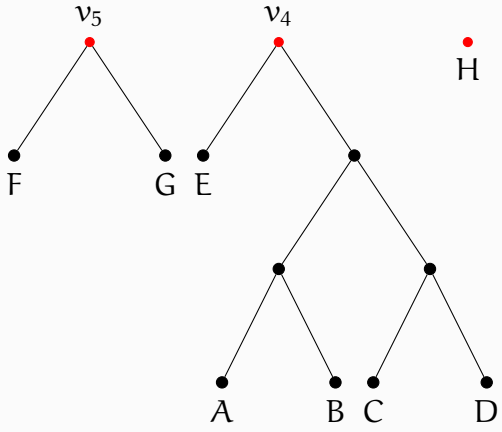
Ahora juntamos los nodos E y v_3 y les damos un padre con probabilidad de $0,12 + 0,13 = 0,25$.



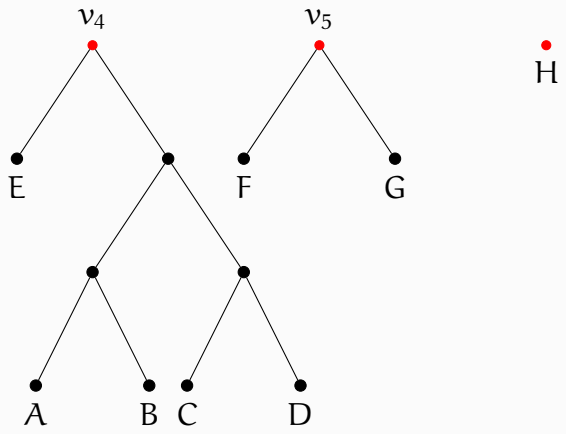
Luego la probabilidad de F y G es menor a la de v_4, asi reorganizando



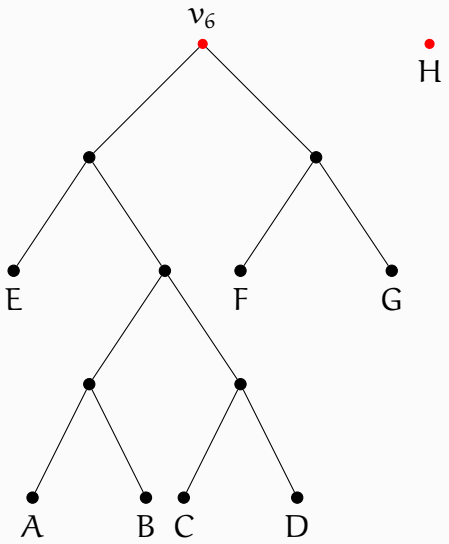
De esta manera tenemos que juntar los nodos F y G con nun nuevo nodo padre, de probabilidad $0,20 + 0,20 = 0,40$.



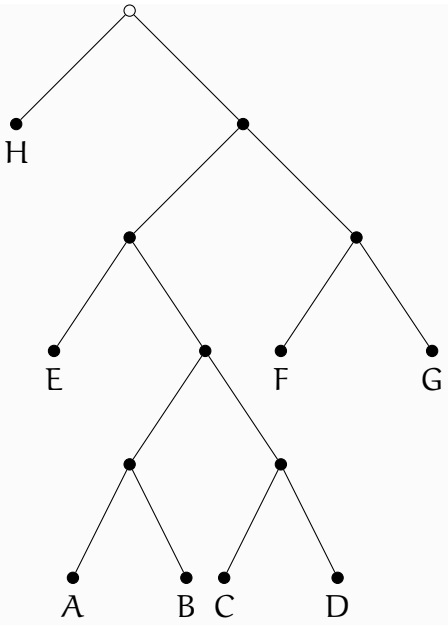
Nuevamente tenemos que reorganizar, ya que la probabilidad de v_5 es mayor a la de v_4, asi



Asi para los nodos v_4 y v_5, asignamos un padre con probabilidad de $0,25 + 0,40 = 0,65$.



Ahora solo hay dos nodos, reorganizando y creando un ultimo nodo activo llegamos a que el arbol de codificacion es



Ya con la codificacion hecha podemos determinar la longitud promedio de palabra ya que

$$\begin{aligned}
&|C(H)| = 1,\\
&|C(E)| = |C(F)| = |C(G)| = 3,\\
&|C(A)| = |C(B)| = |C(C)| = |C(D)| = 5.
\end{aligned}$$

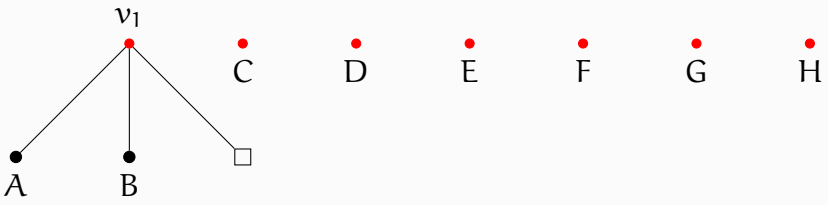
Luego por la formula tenemos que

$$\begin{aligned}
L(C) &= \sum_{i=1}^{8} p_i |C(s_i)|\\
&= 5 \cdot 0,02 + 5 \cdot 0,03 + 5 \cdot 0,04 + 5 \cdot 0,04 + 3 \cdot 0,12 + 3 \cdot 0,20 + 3 \cdot 0,20 + 1 \cdot 0,35\\
&= 5(0,02 + 0,03 + 0,04 + 0,04) + 3(0,12 + 0,20 + 0,20) + 0,35\\
&= 5 \cdot 0,13 + 3 \cdot 0,52 + 0,35\\
&= 2,56 \quad \texttt{bits/simbolo}
\end{aligned}$$

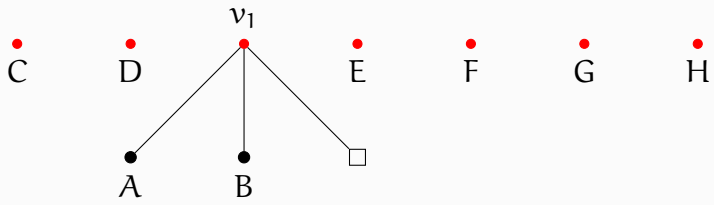
De esta manera $L(C) = 2,56 \quad \texttt{bits/simbolo}$.

Procedamos ahora con el algoritmo de Huffman pero para el caso triario. En este caso tenemos que $n - 1 = 8 - 1 = 7$ y que $D - 1 = 3 - 1 = 2$, luego el residuo de dividir 7 entre 2 es 1, por lo que en el proceso inicial hay que agregar un nodo nuevo.

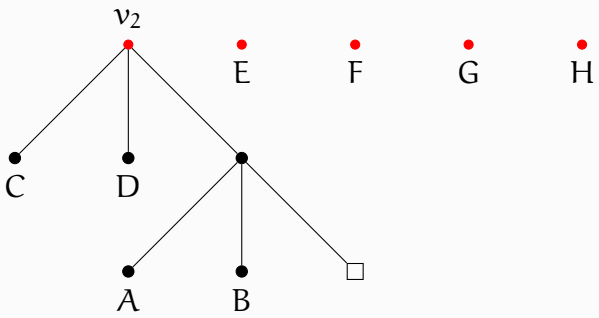
Procedemos igual que en el caso de binario, pero ahora hacemos un nuevo nodo padre v_1, para los nodos de menor probabilidad y el nodo nuevo que añadimos, tal que:



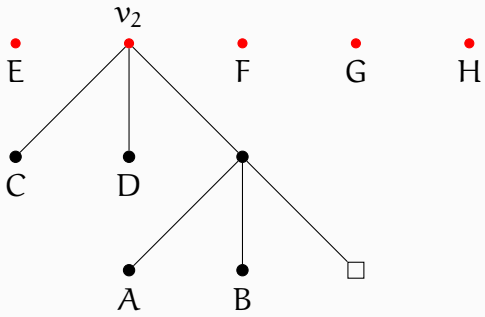
Donde la probabilidad de v_1 es la suma de la de los nodos hijos, es decir $0,05$. Luego toca reorganizar de acuerdo a las nuevas probabilidades



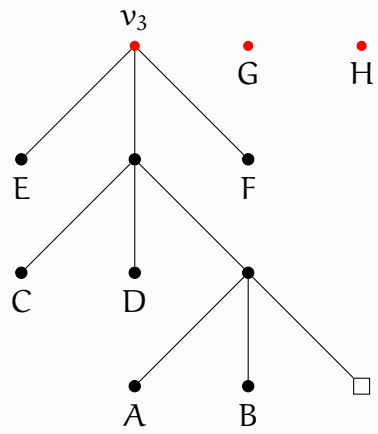
Ahora colocamos un nuevo nodo padre para los nodos C, D y v_1.



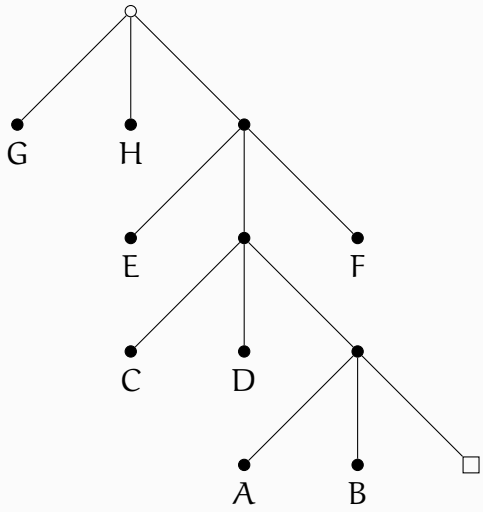
La probabilidad de este nuevo nodo es $0,04 + 0,04 + 0,05 = 0,13$. Por lo que nuevamente tenemos que reorganizar los nodos



Ahora le damos un padre a los nodos E, v_2 y F, que va a tener la probabilidad $0,12+0,013+0,20 = 0,45$.



Ahora como solo quedan tres nodos basta con ponerlos en orden de menor a mayor y juntarlos con el padre que sera la raiz, asi el arbol de codificacion es



De esta manera si a cada rama de a izquierda a derecha le asignamos 0,1 y 2, respectivamente, tenemos que la codificacion de los simbolos es

$$\begin{aligned} C(G) &= 0, \\ C(H) &= 1, \\ C(E) &= 20, \\ C(F) &= 22, \\ C(C) &= 210, \\ C(D) &= 211, \\ C(A) &= 2120, \\ C(B) &= 2121. \end{aligned}$$

De manera abreviada en orden seria $C = \{2120, 2121, 210, 211, 20, 22, 0, 1\}$. Por hacer la comparativa hallemos la longitud promedio de palabra en este caso

$$\begin{aligned} L(C) &= 4 \cdot 0,02 + 4 \cdot 0,03 + 3 \cdot 0,04 + 3 \cdot 0,04 + 2 \cdot 0,12 + 2 \cdot 0,20 + 1 \cdot 0,20 + 1 \cdot 0,35 \\ &= 4 \cdot 0,05 + 3 \cdot 0,08 + 2 \cdot 0,32 + 0,55 \\ &= 1,63 \quad \text{tribits/simbolo.} \end{aligned}$$

$\cup_\smile\cup$

Ejercicio 12
Resuelva el ejercicio 3.5.1 de la página 61 de las notas de clase, calcule la eficiencia en cada caso. Considere una fuente que genera símbolos del alfabeto $S = \{0, 1\}$ con probabilidades $p(0) = 0,9$ y $p(1) = 0,1$. Diseñe códigos de Huffman y de Shannon-Fano para los alfabetos extendidos S^2, S^3 y S^4, en cada caso calcule la eficiencia.

Solución. Hola :p