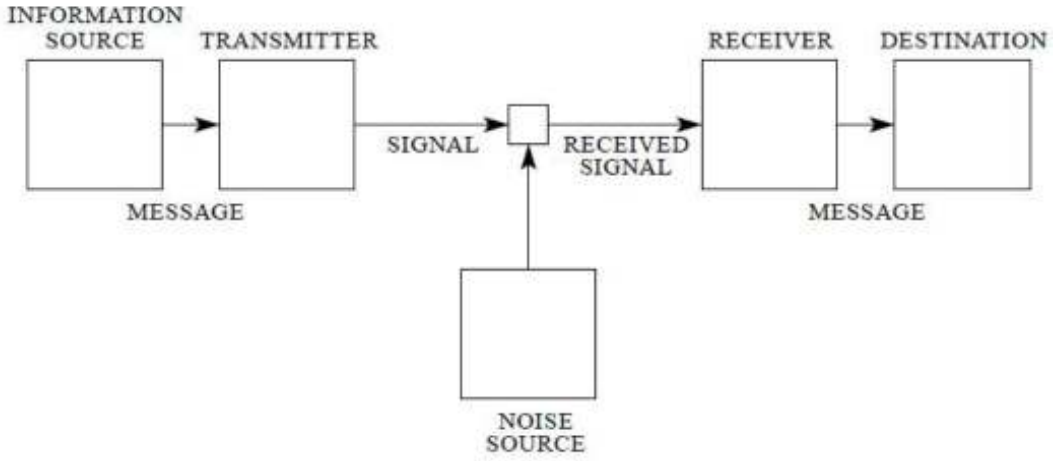
TRABAJO PRÁCTICO N°1 “Codificación de Fuente”

Alumno: Ramirez, Fernando (R-3888/1)

Año: 2020

1) Esquema de un sistema de comunicación punto a punto.



2)

Para este caso tenemos una plaza automotriz compuesta por:

Vehículo	Cantidad	Probabilidad
Autos	500	500/1000=0.5
Colectivos	250	250/1000=0.25
Motos	100	100/1000=0.1
Taxis	100	100/1000=0.1
Bicicletas	50	50/1000=0.05
Total	1000	1

Llamando S a la fuente que contiene los vehículos, es decir, S ={Autos, Colectivos, Motos, Taxis, Bicicletas} y p a la distribución de probabilidades de los elementos de la fuente S, ósea, p ={0.5,0.25,0.1,0.1,0.05}.

Teniendo en cuenta la tabla de probabilidades anterior se implementó el siguiente método de codificación, el diagrama de árbol se encuentra más adelante:

Vehículo	Probabilidad	Código
Autos	0.5	0
Colectivos	0.25	10
Motos	0.1	110
Taxis	0.1	1110
Bicicletas	0.1	1111

Para calcular la eficiencia de este código debemos calcular la entropía(H) de la fuente y la longitud L del código que esta emite.

La entropía de la fuente se calcula como:

$$H(s) = \sum_{i=1}^{5} p_i . log_2(\frac{1}{p_i})$$
$$= 0{,}5 . log_2(\frac{1}{0.5}) + 0.25 . log_2(\frac{1}{0.25}) + 0{,}1 . log_2(\frac{1}{0.1}) + 0{,}1 . log_2(\frac{1}{0.1})$$
$$+ 0{,}05 . log_2(\frac{1}{0.05}) = 1.8804\ [bits/simbolo]$$

Y la longitud L del código como:

$$\bar{L} = \sum_i p_i l_i = 0.5*1 + 0.25*2 + 0.1*3 + 0.1*4 + 0.1*4 = 1.95[bits]$$

Con esto, podemos calcular la eficiencia del código de la siguiente manera:

$$n = \frac{H(s)}{\bar{L}} = \frac{1.8804[bits/simbolo]}{1.95[bits]} = 0.964*100\% = 96.4\%$$

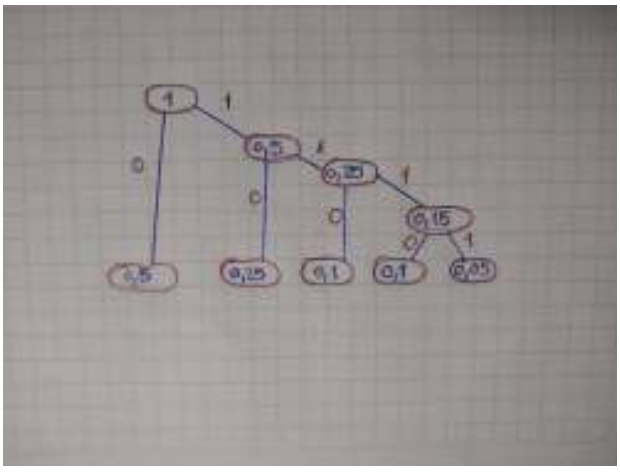
Este código cumple con el teorema de Kraft y con el de Codificación de Fuente para Códigos Prefijos C, es decir:

$$H(s) \le \bar{L} \le H(s) + 1 \rightarrow 1.8804 \le 1.95 \le 2.8804$$

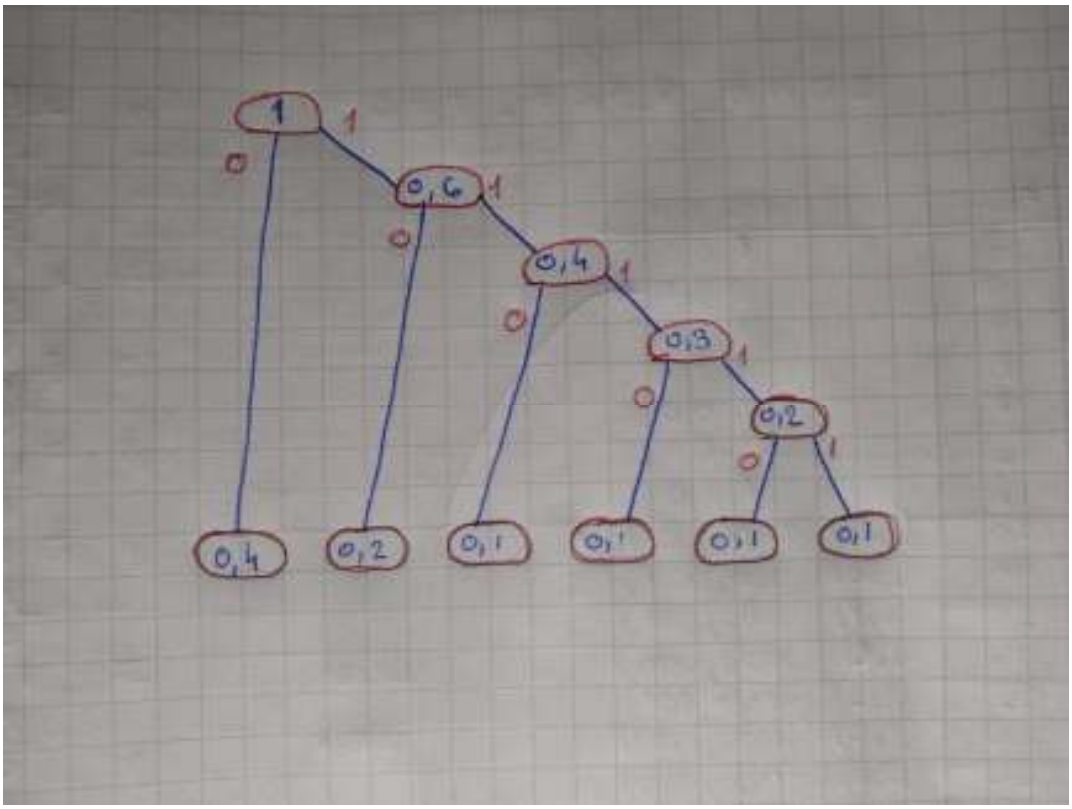
y la desigualdad de Kraft:

$$\sum_{k=0}^{K-1} 2^{-l_k} \le 1 = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} = 1$$

El diagrama de árbol que logra esta codificación se presenta a continuación.



3a) La primera codificación se realizo con el siguiente árbol:



Resultando un código como el que sigue

Elemento	Probabilidad	Código
S1	0.4	0
S2	0.2	10
S3	0.1	110
S4	0.1	1110
S5	0.1	11110
S6	0.1	11111

La longitud del código, calculada con la formula del ejercicio 2, es:

$$\bar{L} = 0.4*1 + 0.2*2 + 0.1*3 + 0.1*4 + 0.1*5 + 0.1*5 = 2.5[bits]$$

La desigualdad de Kraft para este caso queda como:

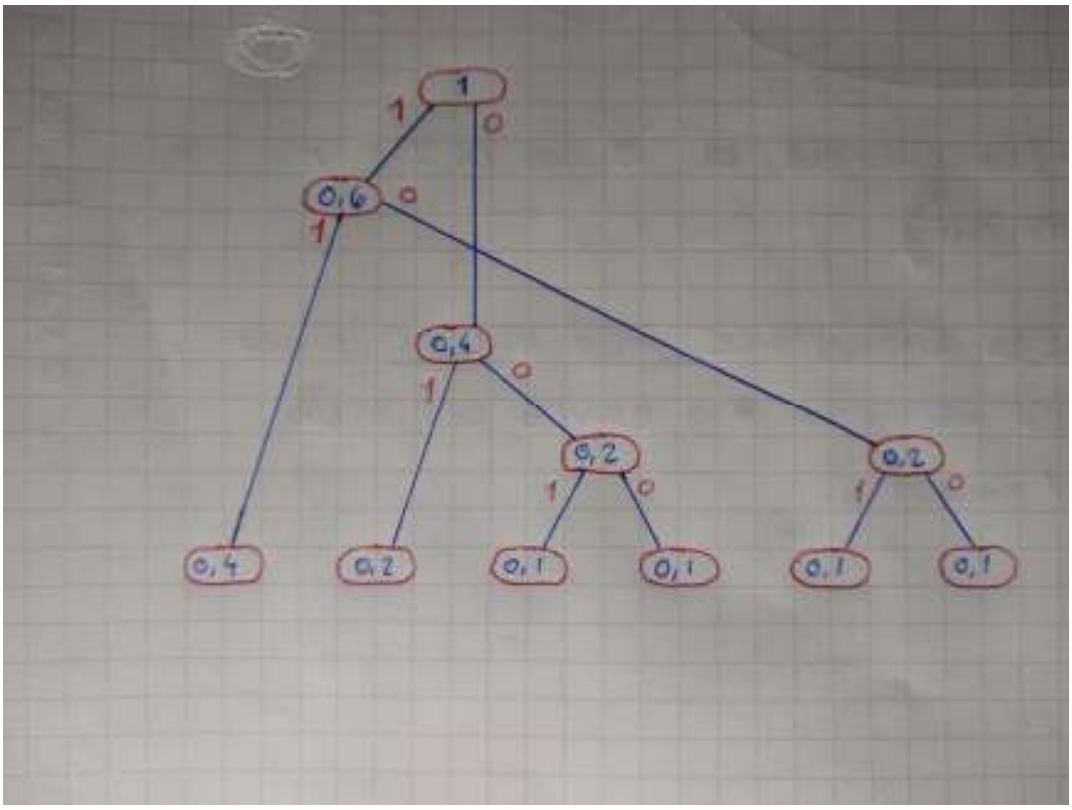
$$\sum_{k=0}^{K-1} 2^{-l_k} \leq 1 = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-5} = 1$$

Se cumple la desigualdad de Kraft.

Y la varianza queda como:

$$\sigma^2 = \sum_{k=0}^{K-1} p_k \left(l_k - \bar{L}\right)^2 = 0.4*(1-2.5)^2 + 0.2*(2-2.5)^2 + 0.1*(3-2.5)^2 + 0.1*(4-2.5)^2 + 0.1*(5-2.5)^2 + 0.1*(5-2.5)^2 = 2.45[bits^2]$$

Otra codificación para estos elementos puede ser la siguiente:



Resultando un código como el que sigue

Elemento	Probabilidad	Código
S1	0.4	11
S2	0.2	01
S3	0.1	001
S4	0.1	000
S5	0.1	101
S6	0.1	100

La longitud del código, calculada con la formula del ejercicio 2, es:

$$\bar{L} = 0.4*2 + 0.2*2 + 0.1*3 + 0.1*3 + 0.1*3 + 0.1*3 = 2.4[bits]$$

La desigualdad de Kraft para este caso queda como:

$$\sum_{k=0}^{K-1} 2^{-l_k} \leq 1 = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} + 2^{-3} + 2^{-3} = 1$$

Se cumple la desigualdad de Kraft.

Y la varianza queda como:

$$\sigma^2 = \sum_{k=0}^{K-1} p_k \left(l_k - \bar{L} \right)^2 = 0.4*(2-2.4)^2 + 0.2*(2-2.4)^2 + 0.1*(3-2.4)^2 + 0.1*(3-2.4)^2 + 0.1*(3-2.4)^2 + 0.1*(3-2.4)^2 = 0.24[bits^2]$$

La entropía de la fuente es, según la formula utilizada en el ejercicio 2:

$$\mathrm{H(s)} = \sum_{i=1}^{6} p_i \log_2 \left(\frac{1}{p_i} \right) = 2.322[bits / simbolo]$$

Y el rendimiento para el caso 1 es del :

$$n = \frac{H(s)}{\bar{L}} = \frac{2.322[bits / simbolo]}{2.5[bits]} = 0.9288*100\% = 92.88\%$$

Mientras que para el caso 2:

$$n = \frac{H(s)}{\bar{L}} = \frac{2.322[bits / simbolo]}{2.4[bits]} = 0.9675*100\% = 96.75\%$$

Como se puede observar por los cálculos, los códigos obtenidos no tienen la misma longitud L, lo que arroja una eficiencia diferente, pero además, la variancia de ambos códigos también es distinta. Es fácil de ver que la variancia de la codificación 2, es 10 veces menor que la del código 1.

Que la variancia sea la menor posible es altamente deseable, ya que influye directamente en los costos del sistema y en la eficiencia de la transmisión, al tener que agregar un buffer cuando esta es muy grande, para equilibrar las tasas de transferencia de los códigos por existir códigos muy grandes y pequeños dentro de las posibilidades.

Siempre lo que se desea es mantener una uniformidad en las velocidades de transferencia de las distintas etapas del sistema de comunicación.

B)

Si la fuente emite S2,S1,S4,S1,S5,S3,S6 y se codifica con los códigos obtenidos en el apartado a) se puede decir que su decodificación es UD(Únicamente Decodificable), en primer lugar, porque si observamos los códigos obtenidos para cada uno de ellos, vemos que ninguno es prefijo del otro. En segundo lugar, porque realizando una codificación Huffman, una de las cosas que esta asegurada es justamente que los códigos son UD, ya que verifican los teoremas de Kraft y MacMillan y el de Codificación de Fuente para Códigos Prefijos C.

4)a) Para este caso tenemos una imagen la cual se quiere transmitir. A partir de esta imagen podemos armar una tabla de probabilidades de los elementos de la fuente como sigue:

Símbolo	Probabilidad
C	P1=14/30=0.47
A	P2=7/30=0.23
R	P3=8/30=0.27
N	P4=1/30=0.03

La entropía de la fuente la obtenemos aplicando la misma formula de los ejercicios pasados con las probabilidades obtenidas en la tabla anterior. Esto nos da el siguiente resultado:

$$\mathrm{H(s)}=\sum_{i=1}^{4} p_i \log_2\left(\frac{1}{p_i}\right) = 1.66[bits/simbolo]$$

B)

La entropía de una fuente extendida es

$$H(s^n) = n * H(s)$$

Entonces, para el caso de la entropía de una fuente de orden 2,3 y 4 tenemos que es:

$$H(s^2) = 2 * H(s) = 3.32[bits/simbolo]$$

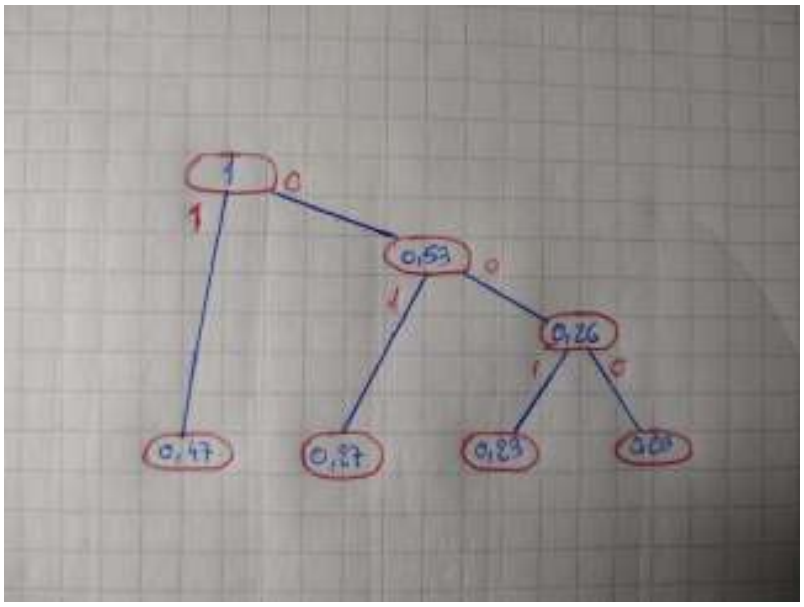
$$H(s^3) = 3 * H(s) = 4.98[bits/simbolo]$$

$$H(s^4) = 4 * H(s) = 6.64[bits/simbolo]$$

C)

<u>Orden 1</u>

Para esta codificación se desarrolló el siguiente árbol



Y se obtuvo la siguiente codificación

Elemento	Probabilidad	Código
C	0.47	1
R	0.27	01
A	0.23	001
N	0.03	000

Para calcular la eficiencia de este código debemos calcular la entropía(H) de la fuente y la longitud L del código que esta emite.

La entropía de la fuente se calcula como:

$$H(s) = \sum_{i=1}^{4} p_i . log_2(\frac{1}{p_i})$$
$$= 0{,}47 . log_2(\frac{1}{0.47}) + 0.27 . log_2(\frac{1}{0.27}) + 0{,}23 . log_2(\frac{1}{0.23})$$
$$+ 0{,}03 . log_2(\frac{1}{0.03}) = 1.66\ [bits/simbolo]$$

Y la longitud L del código como:

$$\bar{L} = \sum_i p_i l_i = 0.47*1 + 0.27*2 + 0.23*3 + 0.03*3 = 1.79[bits]$$

Con esto, podemos calcular la eficiencia del código de la siguiente manera:

$$n = \frac{H(s)}{\bar{L}} = \frac{1.66[bits/simbolo]}{1.79[bits]} = 0.927*100\% = 92.7\%$$

Este código cumple con el teorema de Kraft y con el de Codificación de Fuente para Códigos Prefijos C, es decir:

$$H(s) \le \bar{L} \le H(s) + 1 \rightarrow 1.66 \le 1.79 \le 2.66$$

y la desigualdad de Kraft:

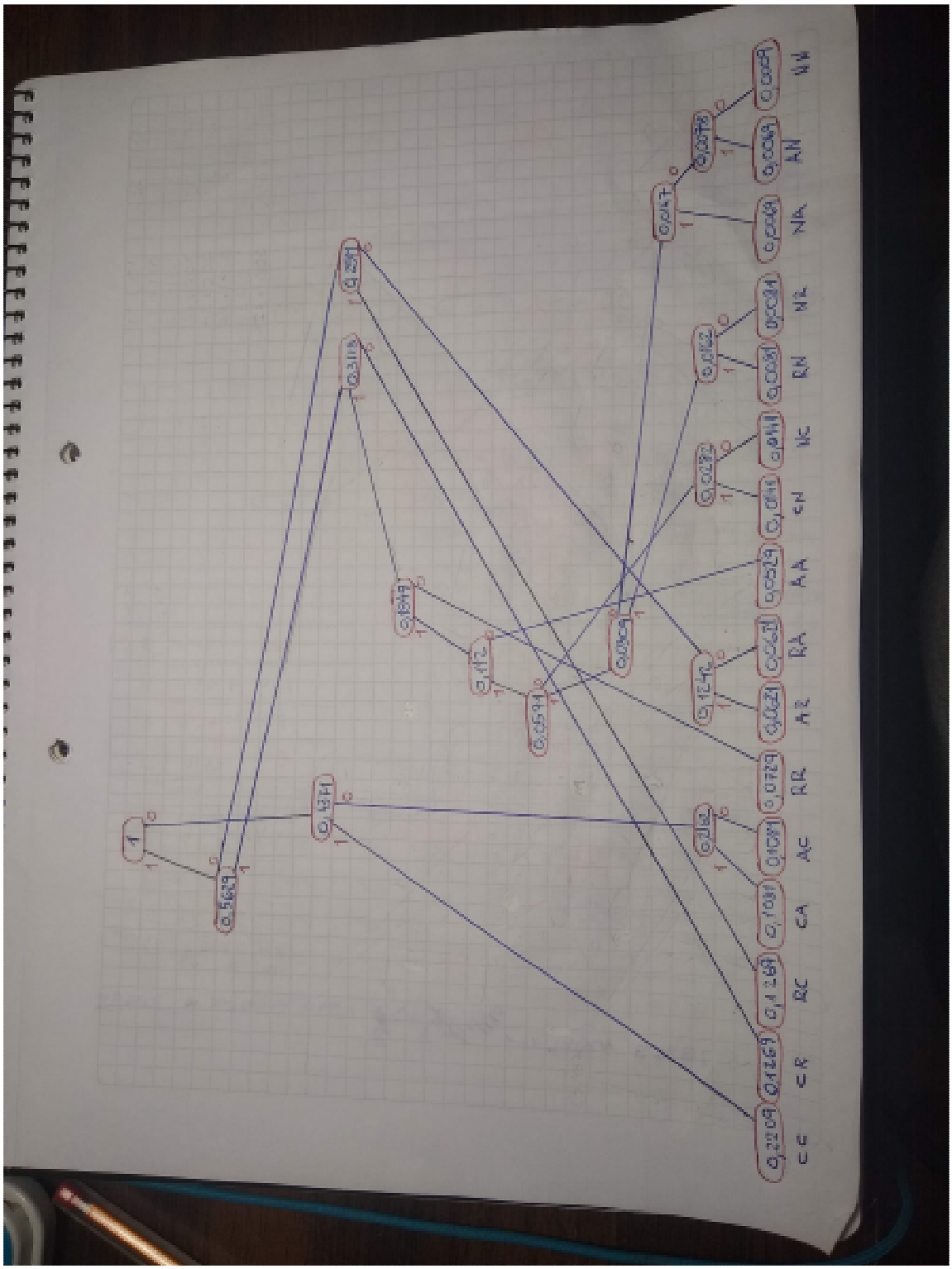
$$\sum_{k=0}^{K-1} 2^{-l_k} \le 1 = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} = 1$$

<u>Orden 2</u>

Para la fuente de orden 2, dada la extensión de los cálculos a realizar, se implemento una planilla de Excel donde se confeccionaron las tablas y las distintas funciones para calcular todos los valores.

Se coloca una captura de pantalla de dicha planilla de Excel, junto con las formulas implementadas en cada caso.

También se coloca una foto del diagrama de árbol que se utilizó para lograr la codificación



	C	A	R	N
C	0,2209	0,1081	0,1269	0,0141
A	0,1081	0,0529	0,0621	0,0069
R	0,1269	0,0621	0,0729	0,0081
N	0,0141	0,0069	0,0081	0,0009

suma prob=	1

p1	p2	p3	p4
0,47	0,23	0,27	0,03

Símbolo	N°bits	código	Probabilidad
CC	2	01	0,2209
CR	3	110	0,1269
RC	3	101	0,1269
CA	3	001	0,1081
AC	3	000	0,1081
RR	4	1110	0,0729
AR	4	1001	0,0621
RA	4	1000	0,0621
AA	5	11110	0,0529
CN	7	1111101	0,0141
NC	7	1111100	0,0141
RN	8	11111111	0,0081
NR	8	11111110	0,0081
NA	8	11111101	0,0069
AN	9	111111001	0,0069
NN	9	111111000	0,0009

Long. L =	3,3571
Varianza2=	1,8115756
Entropia=	3,3228234
DES KRAFT=	1
n=	98,979%

Para armar la matriz 4x4 se implemento , en cada celda la multiplicación de las probabilidades de los respectivos elementos conformantes, por ejemplo: La casilla CC=p1*p1 a la NR=p4*p3 y así sucesivamente.

Para la casilla “suma prob=” se implemento la siguiente función: =SUMA(F7:I10) que suma el total de las probabilidades expresadas en la matriz de probabilidades.

Para la casilla “Long L=” se implementó la siguiente función: “=(O12*M12+O13*M13+O14*M14+O15*M15+O16*M16+O17*M17+O18*M18+O19*M19+O20*M20+O21*M21+O22*M22+O23*M23+O24*M24+O25*M25+O26*M26+O27*M27)”

Que calcula la suma de la longitud de cada código por su probabilidad, entregando la Longitud promedio de un string cualquiera de esta fuente.

Para la casilla “Varianza2=” se implemento la siguiente función: =SUMA(O12:O27*POTENCIA(M12:M27-S10;2)) que calcula varianza de la codificación obtenida.

Para la casilla “Entropia=” se implemento la siguiente función: =SUMA(-O12:O27*LOG(O12:O27;2)) que calcula la entropía de la fuente.

Para la casilla “Des Kraft=” se implementó la siguiente función: =SUMA(POTENCIA(2;-M12:M27)) que calcula la desigualdad de Kraft de la codificación.

Para la casilla “n=” se implemento la siguiente operación: =S14/S10, que calcula el rendimiento al dividir la entropía por la longitud promedio L. En este caso se configuro la celda para que muestre el resultado en porcentaje.

5) Para este ejercicio se pedía codificar en binario, utilizando el método LZ78, el siguiente string:

como_poco_coco_como,_poco_coco_compro

Como hipótesis se tenía que el diccionario se tenía precargado en el codificador, con lo cual, de entrada, se puede confeccionar la siguiente tabla.

Índice bin.	Índice	Frases	Cod	Cod bin
00001	1	C		000
00010	2	O		001
00011	3	M		010
00100	4	P		011
00101	5	R		100
00110	6	_		101
00111	7	,		110

Con esta tablase identificaron los caracteres simples del diccionario que van a ser utilizados por el algoritmo y su respectiva codificación binaria.

Luego lo que resta hacer es ejecutar el algoritmo, para lo cual se realizó una subdivisión del mensaje de la manera que lo hubiera hecho el algoritmo. Y luego se actualizó el diccionario quedando de la siguiente manera.

co | mo | _p | oc | o_ | coc | o_c | om | o, | _po | co_ | coco | _c | omp | ro

8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

Con esa subdivisión se actualizó el diccionario.

Índice bin.	Índice	Frases	Cod	Cod bin
00001	1	C		000
00010	2	O		001
00011	3	M		010
00100	4	P		011
00101	5	R		100
00110	6	_		101
00111	7	,		110
01000	8	CO	(1,O)	(0001, 001)
01001	9	MO	(3,O)	(0011, 001)
01010	10	_P	(6,P)	(0110, 011)
01011	11	OC	(2,C)	(0010, 000)
01100	12	O_	(2,_)	(0010, 101)
01101	13	COC	(8,C)	(1000, 000)
01110	14	O_C	(12,C)	(1100, 000)
01111	15	OM	(2,M)	(0010, 010)
10000	16	O,	(2, “,”)	(0010, 110)
10001	17	_PO	(10,O)	(1010, 001)
10010	18	CO_	(8,_)	(1000, 101)
10011	19	COCO	(13,O)	(1101, 001)
10100	20	_C	(6,C)	(0110, 000)
10101	21	OMP	(15,P)	(1111, 011)
10110	22	RO	(5,O)	(0101, 001)

Notar que al escribir la codificación en binario de cada uno de los elementos del diccionario se omitió el primer 0 en todos los índices, esto es dado que el máximo índice utilizado es 15 con lo cual el primero de todos los strings es un 0 que se puede omitir.

Con esta tabla podemos armar la secuencia binaria encontrada a la salida del decodificador colocando cada codificación binaria una al lado de la otra de la siguiente manera:

000100100110010110011001000000101011000000110000000010010001011010100011000101110100101100001111011010101001

La principal diferencia entre LZ78 y LZW es que el LZW inicializa con un diccionario precargado, normalmente el ASCII aunque puede ser cualquiera dependiendo de la aplicación. Por esto, no tiene necesidad de mostrar el siguiente carácter como el LZ78, sino solamente el índice de diccionario correspondiente. En promedio, esto hace que la compresión LZW sea más eficiente que la de LZ78.