*“C. Shannon transformó ntra manera d entender la info al demostrar q en el caos y la incertidumbre existen patrones q podemos usar p/comprimir y optimizar los datos. Su teoría revolucionó telecoms y mostró q cn correcta interpretación, es posible reducir la complejidad sin perder el mensaje esencial. La compresión es el arte d decir + cn -, y Shannon mostró q este principio es el corazón d la eficiencia n el mundo digital”*

### **ÁREA TEMÁTICA No1**

**PRÁCTICO 3 TEORÍA DE LA INFORMACIÓN: “Compactadores”**

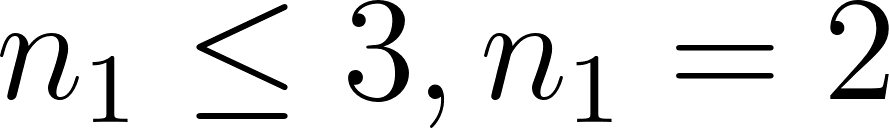
1. Sean los siguientes códigos:

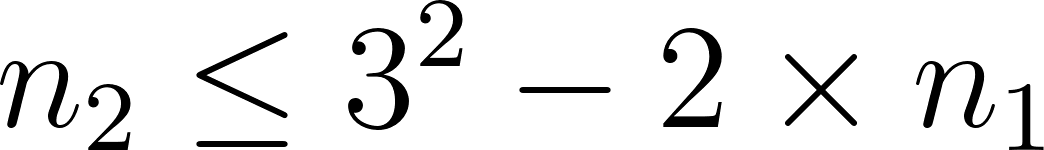
| **s** | **pi** | **a** | **b** | **c** | **d** | **e** | **f** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **s1** | **1/2** | **000** | **0** | **0** | **0** | **0** | **0** |
| **s2** | **1/4** | **001** | **01** | **10** | **10** | **10** | **100** |
| **s3** | **1/16** | **010** | **011** | **110** | **110** | **1100** | **101** |
| **s4** | **1/16** | **011** | **0111** | **1110** | **1110** | **1101** | **110** |
| **s5** | **1/16** | **100** | **01111** | **11110** | **1011** | **1110** | **111** |
| **s6** | **1/16** | **101** | **011111** | **111110** | **1101** | **1111** | **001** |
| **Bloque** |  | **X** |  |  |  |  |  |
| **No singular** |  | **X** | **X** | **X** | **X** | **X** | **X** |
| **Instantáneo** |  | **X** |  | **X** |  |  |  |
| **Univocamente decodificable** |  | **X** | **X** | **X** | **X** | **X** |  |
| **Longitud Promedio** |  | **3** | **2.125** | **2.125** | **1.9375** | **2** | **2** |

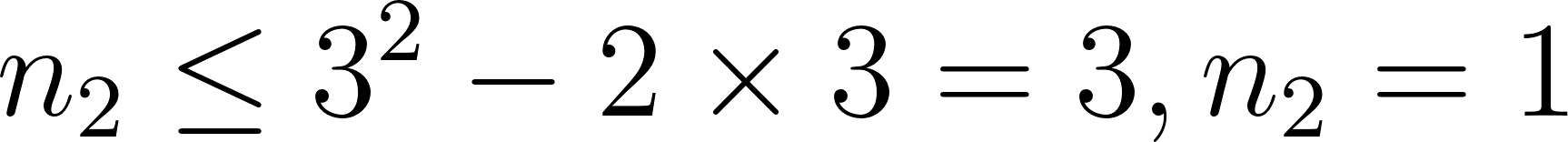
* 1. Clasificar cada uno de ellos ( Bloque, No singular, Instantáneos, Unívocamente decodif.)
  2. Calcular la longitud promedio de los unívocamente decodificables.

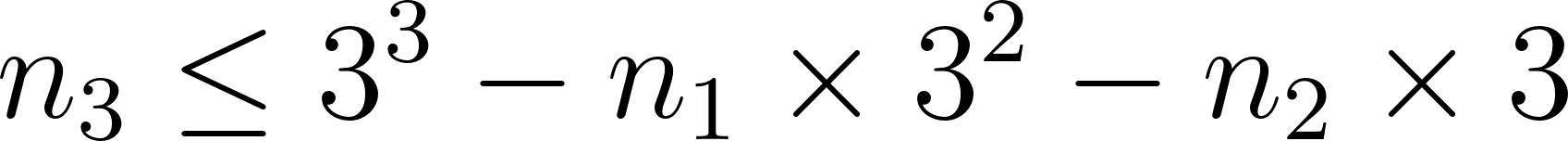
1. Ejercicios
   1. ¿Cuál de los conjuntos de longitudes de la siguiente tabla es válida para un código unívocamente decodificable con base X{0,1,2}?

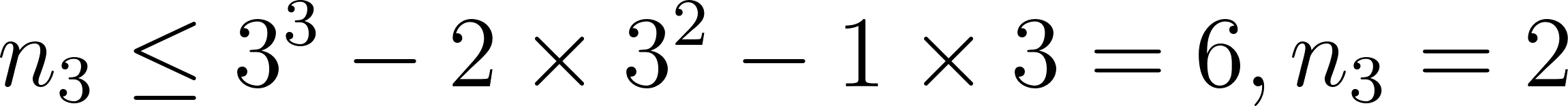
Para Código A:

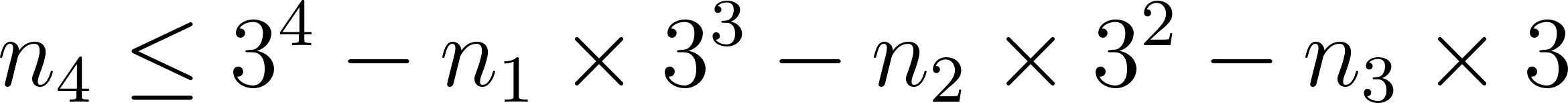
[](https://www.codecogs.com/eqnedit.php?latex=%20n_1%20%5Cleq%203%20%2C%20n_1%20%3D%202%20#0), es válido

[](https://www.codecogs.com/eqnedit.php?latex=%20n_2%20%5Cleq%203%5E2%20-%202%20%5Ctimes%20n_1%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20n_2%20%5Cleq%203%5E2-2%20%5Ctimes%203%20%20%3D%203%2C%20n_2%20%3D%201%20#0), es válido

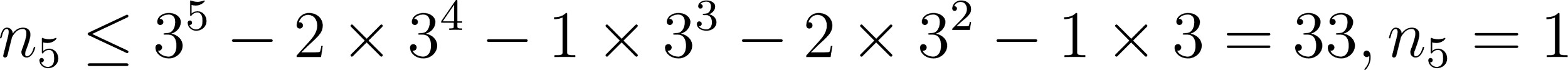
[](https://www.codecogs.com/eqnedit.php?latex=%20n_3%20%5Cleq%203%5E3-%20n_1%20%5Ctimes%203%5E2%20-%20n_2%20%5Ctimes%203%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20n_3%20%5Cleq%203%5E3%20%20-%202%20%5Ctimes%20%203%5E2%20-%201%20%5Ctimes%203%20%3D%206%2C%20n_3%20%3D%202%20#0), es válido

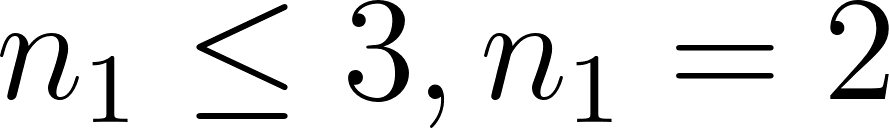
[](https://www.codecogs.com/eqnedit.php?latex=%20n_4%20%5Cleq%203%5E4%20-%20n_1%20%5Ctimes%203%5E3%20-%20n_2%20%5Ctimes%203%5E2%20-%20n_3%20%5Ctimes%203%20#0)

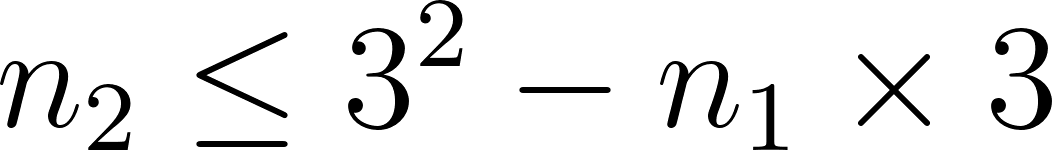
[](https://www.codecogs.com/eqnedit.php?latex=%20n_4%20%5Cleq%203%5E4%20%20-%202%20%5Ctimes%203%5E3%20-%201%20%5Ctimes%203%5E2%20-%202%20%5Ctimes%203%20%3D%2012%2C%20n_4%20%3D%201%20#0), es válido

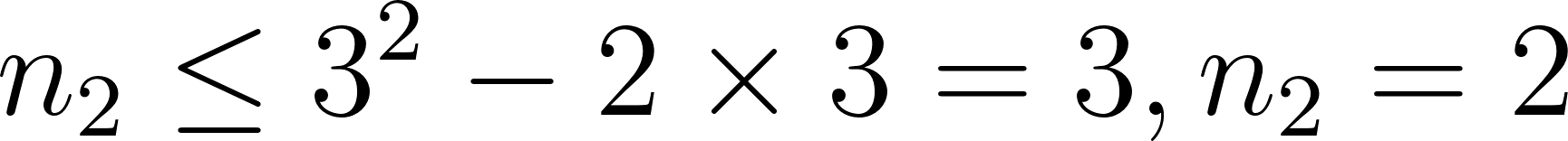
[](https://www.codecogs.com/eqnedit.php?latex=%20n_5%20%5Cleq%203%5E5%20%20-%20n_1%20%5Ctimes%203%5E4%20-%20n_2%20%5Ctimes%203%5E3%20-%20n_3%20%5Ctimes%203%5E2%20-%20n_4%20%5Ctimes%203%20#0)

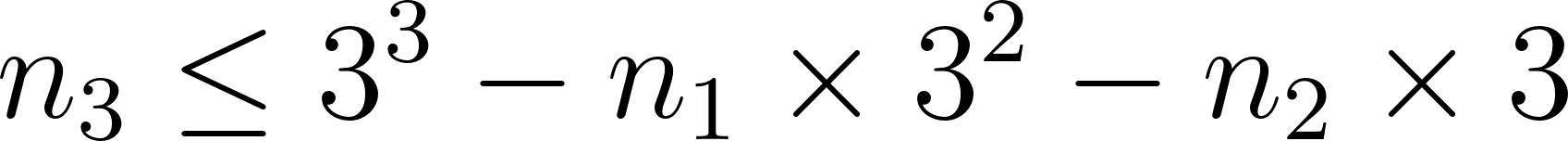
[](https://www.codecogs.com/eqnedit.php?latex=%20n_5%20%5Cleq%20%203%5E5%20%20-%202%20%5Ctimes%203%5E4%20-%201%20%5Ctimes%203%5E3%20-%202%20%5Ctimes%203%5E2%20-%201%20%5Ctimes%203%20%3D%2033%2C%20n_5%20%3D%201%20#0), es válido

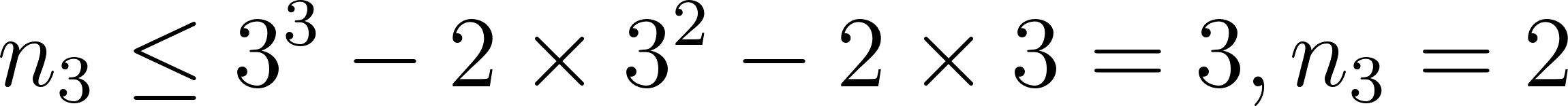
Para Código B:

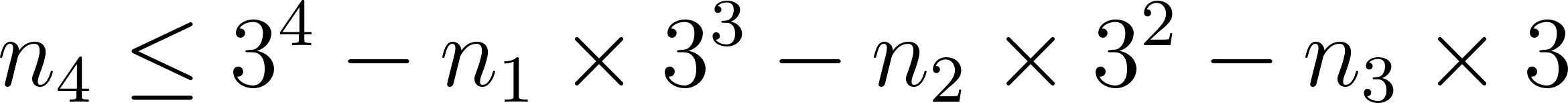
[](https://www.codecogs.com/eqnedit.php?latex=%20n_1%20%5Cleq%203%2C%20n_1%20%3D%202%20#0), es válido

[](https://www.codecogs.com/eqnedit.php?latex=%20n_2%20%5Cleq%203%5E2%20-%20n_1%20%5Ctimes%203%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20n_2%20%5Cleq%203%5E2%20-%202%20%5Ctimes%203%20%3D%203%2C%20n_2%20%3D%202%20#0), es válido

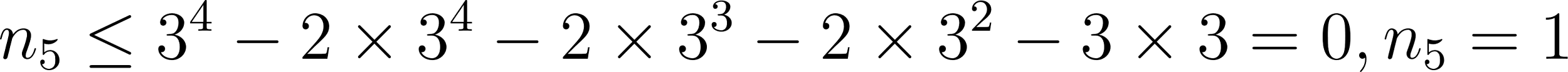
[](https://www.codecogs.com/eqnedit.php?latex=%20n_3%20%5Cleq%203%5E3%20-%20n_1%20%5Ctimes%203%5E2%20-%20n_2%20%5Ctimes%203%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20n_3%20%5Cleq%203%5E3%20-%202%20%5Ctimes%203%5E2%20-%202%20%5Ctimes%203%20%3D%203%2C%20n_3%20%3D%202%20#0), es válido

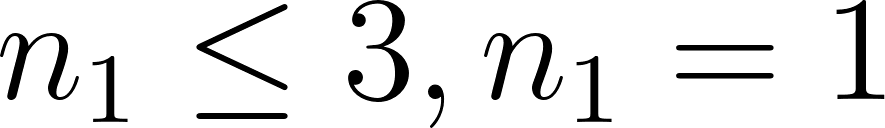
[](https://www.codecogs.com/eqnedit.php?latex=%20n_4%20%5Cleq%203%5E4%20-%20n_1%20%5Ctimes%203%5E3%20-%20n_2%20%5Ctimes%203%5E2%20-%20n_3%20%5Ctimes%203%20#0)

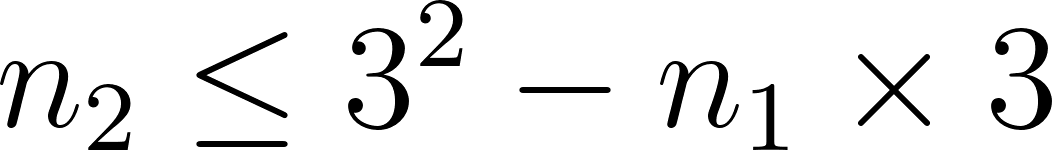
[](https://www.codecogs.com/eqnedit.php?latex=%20n_4%20%5Cleq%203%5E4%20-%202%20%5Ctimes%203%5E3%20-%202%20%5Ctimes%203%5E2%20-%202%20%5Ctimes%203%20%3D%203%2C%20n_4%20%3D%203%20#0), es válido

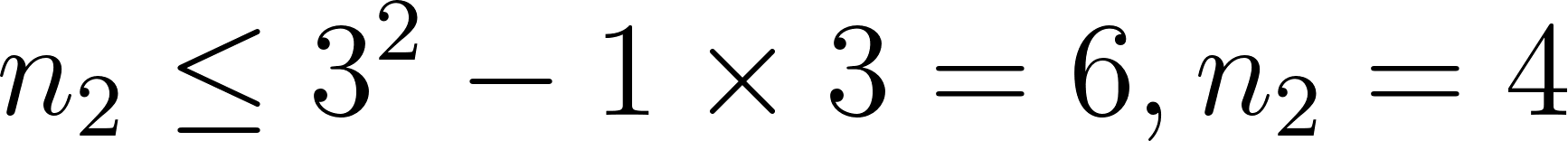
[](https://www.codecogs.com/eqnedit.php?latex=%20n_5%20%5Cleq%203%5E4%20-%20n_1%20%5Ctimes%203%5E4%20-%20n_2%20%5Ctimes%203%5E3%20-%20n_3%20%5Ctimes%203%5E2%20-%20n_4%20%5Ctimes%203%20#0)

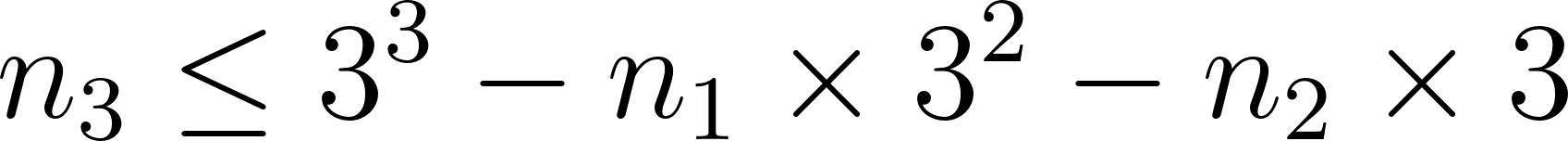
[](https://www.codecogs.com/eqnedit.php?latex=%20n_5%20%5Cleq%203%5E4%20-%202%20%5Ctimes%203%5E4%20-%202%20%5Ctimes%203%5E3%20-%202%20%5Ctimes%203%5E2%20-%203%20%5Ctimes%203%20%3D%200%2C%20n_5%20%3D%201%20#0), No es válido

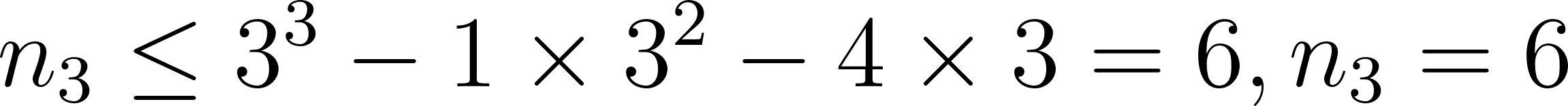
Para Código C:

[](https://www.codecogs.com/eqnedit.php?latex=%20n_1%20%5Cleq%203%2C%20n_1%20%3D%201%20#0), es válido

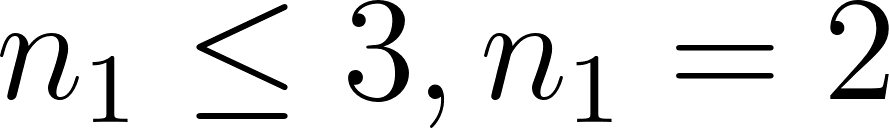
[](https://www.codecogs.com/eqnedit.php?latex=%20n_2%20%5Cleq%203%5E2%20-%20n_1%20%5Ctimes%203%20#0)

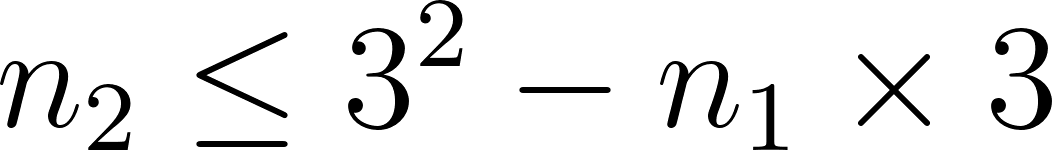
[](https://www.codecogs.com/eqnedit.php?latex=%20n_2%20%5Cleq%203%5E2%20-%201%20%5Ctimes%203%20%3D%206%2C%20n_2%20%3D%204%20#0), es válido

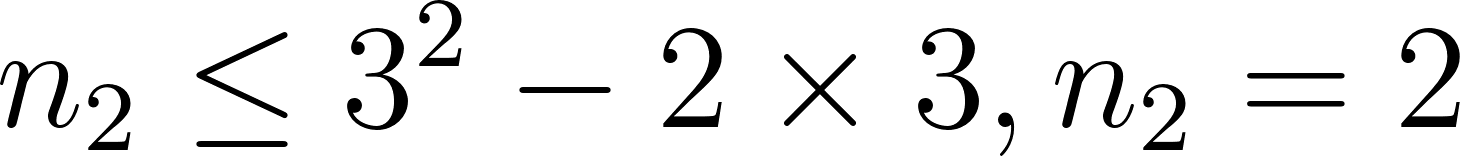
[](https://www.codecogs.com/eqnedit.php?latex=%20n_3%20%5Cleq%203%5E3%20-%20n_1%20%5Ctimes%203%5E2%20-%20n_2%20%5Ctimes%203%20#0)

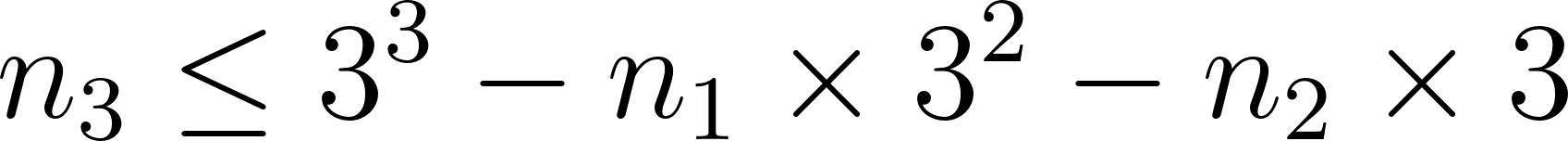
[](https://www.codecogs.com/eqnedit.php?latex=%20n_3%20%5Cleq%203%5E3%20-%201%20%5Ctimes%203%5E2%20-%204%20%5Ctimes%203%20%3D%206%2C%20n_3%20%3D%206%20#0), es válido

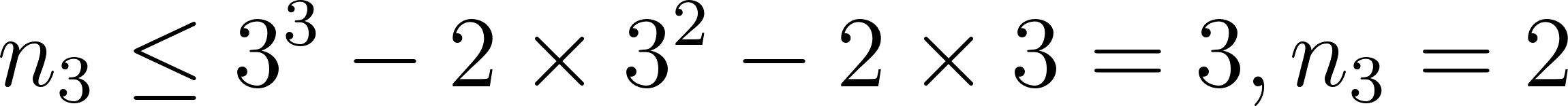
Para Código D:

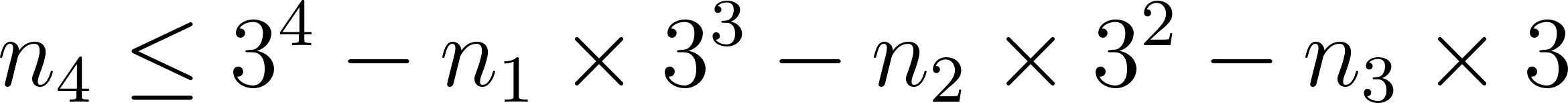
[](https://www.codecogs.com/eqnedit.php?latex=%20n_1%20%5Cleq%203%2C%20n_1%20%3D%202%20#0), es válido

[](https://www.codecogs.com/eqnedit.php?latex=%20n_2%20%5Cleq%203%5E2%20-%20n_1%20%5Ctimes%203%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20n_2%20%5Cleq%203%5E2%20-%202%20%5Ctimes%203%2C%20n_2%20%3D%202%20#0), es válido

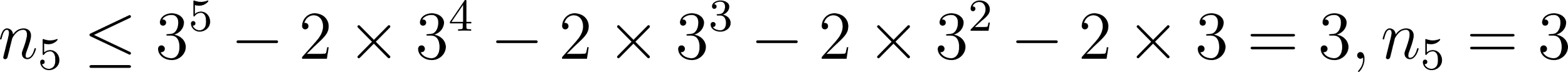
[](https://www.codecogs.com/eqnedit.php?latex=%20n_3%20%5Cleq%203%5E3%20-%20n_1%20%5Ctimes%203%5E2%20-%20n_2%20%5Ctimes%203%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20n_3%20%5Cleq%203%5E3%20-%202%20%5Ctimes%203%5E2%20-%202%20%5Ctimes%203%20%3D%203%2C%20n_3%20%3D%202%20#0), es válido

[](https://www.codecogs.com/eqnedit.php?latex=%20n_4%20%5Cleq%203%5E4%20-%20n_1%20%5Ctimes%203%5E3%20-%20n_2%20%5Ctimes%203%5E2%20-%20n_3%20%5Ctimes%203%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20n_4%20%5Cleq%203%5E4%20-%202%20%5Ctimes%203%5E3%20-%202%20%5Ctimes%203%5E2%20-%202%20%5Ctimes%203%20%3D%203%2C%20n_4%20%3D%202%20#0), es válido

[](https://www.codecogs.com/eqnedit.php?latex=%20n_5%20%5Cleq%203%5E4%20-%20n_1%20%5Ctimes%203%5E4%20-%20n_2%20%5Ctimes%203%5E3%20-%20n_3%20%5Ctimes%203%5E2%20-%20n_4%20%5Ctimes%203%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20n_5%20%5Cleq%203%5E5%20-%202%20%5Ctimes%203%5E4%20-%202%20%5Ctimes%203%5E3%20-%202%20%5Ctimes%203%5E2%20-%202%20%5Ctimes%203%20%3D%203%2C%20n_5%20%3D%203%20#0), es válido

* 1. Construir un código instantáneo con cada conjunto de longitudes válidas

| **Long. palabra →** | **1** | **2** | **3** | **4** | **5** |
| --- | --- | --- | --- | --- | --- |
| **Código A** | **2** | **1** | **2** | **1** | **4** |
| **Código B** | **2** | **2** | **2** | **3** | **1** |
| **Código C** | **1** | **4** | **6** | **0** | **0** |
| **Código D** | **2** | **2** | **2** | **2** | **3** |

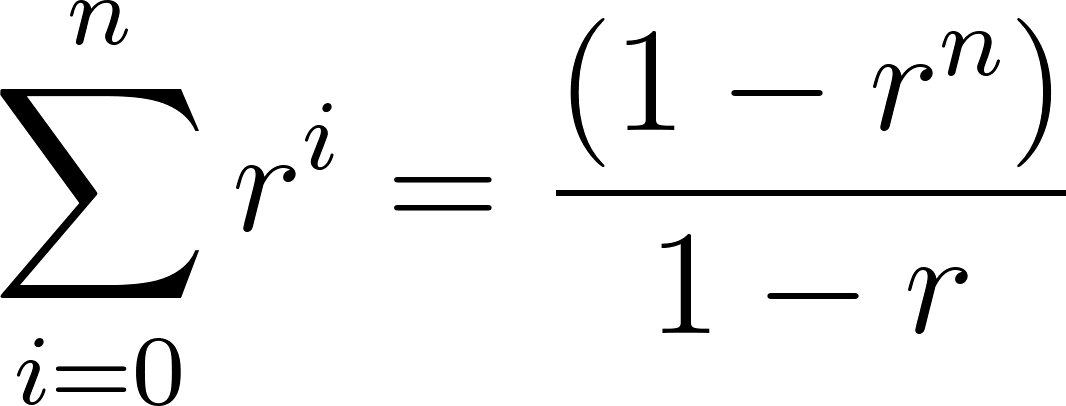
| **Símbolo** | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** | **S9** | **S10** | **S11** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Código A** | **0** | **1** | **20** | **210** | **211** | **2120** | **21210** | **21211** | **21212** | **21220** | **-** |
| **Código C** | **0** | **10** | **11** | **12** | **20** | **210** | **211** | **212** | **220** | **221** | **222** |
| **Código D** | **0** | **1** | **20** | **21** | **220** | **221** | **2220** | **2221** | **22220** | **22221** | **22222** |

1. Demostrar que el código coma cumple Kraft

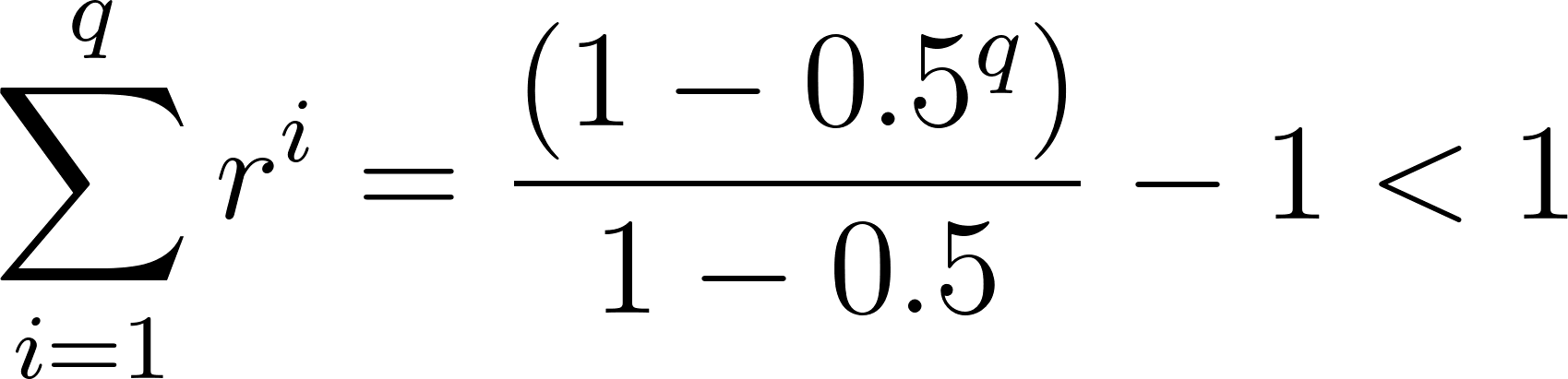
La fórmula para la inecuación de Kraft es

Vamos a demostrarlo para cualquier q

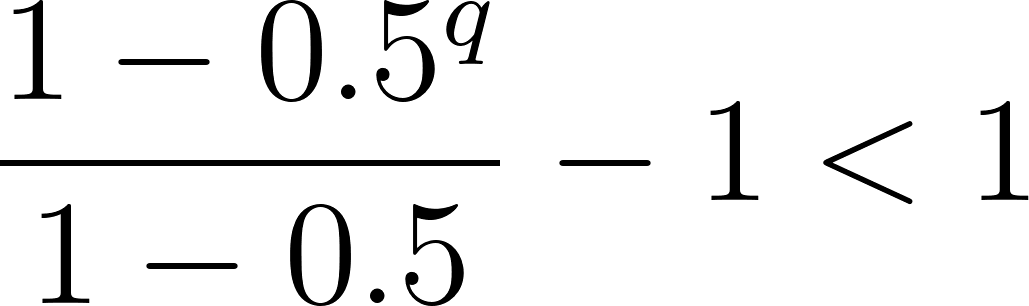
La suma de los primeros n términos de una serie geométrica es

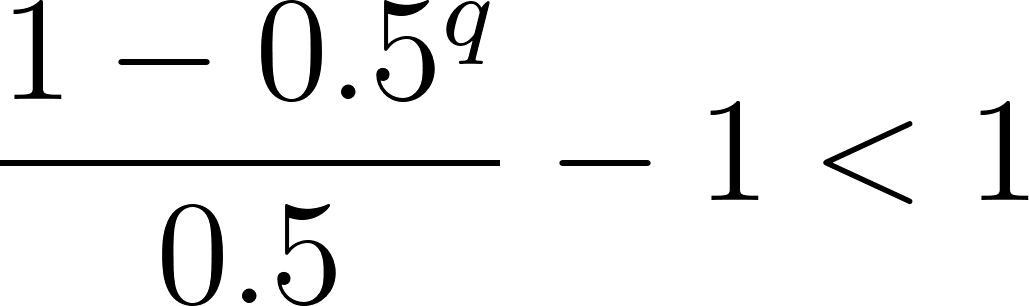
[](https://www.codecogs.com/eqnedit.php?latex=%20%5Csum_%7Bi%3D0%7D%5En%20r%5Ei%20%3D%20%7B(1-r%5En)%20%5Cover%201-r%7D%20#0)

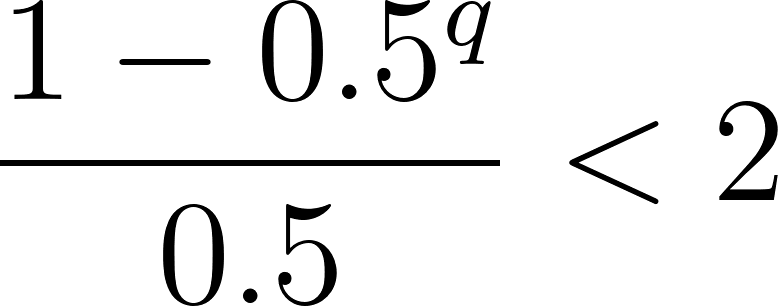
Luego, en nuestro caso queremos demostrar que

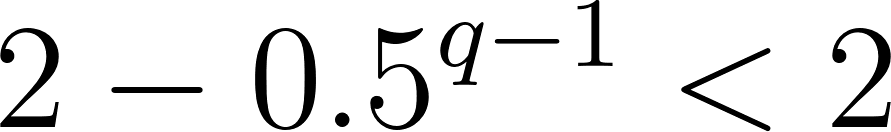
[](https://www.codecogs.com/eqnedit.php?latex=%20%5Csum_%7Bi%3D1%7D%5Eq%20r%5Ei%20%3D%20%7B(1-0.5%5Eq)%20%5Cover%201-0.5%7D%20-%201%20%3C%201#0)

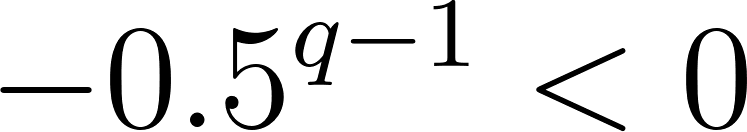
Ya que estamos trabajando con la codificación coma vamos a tener que li = i

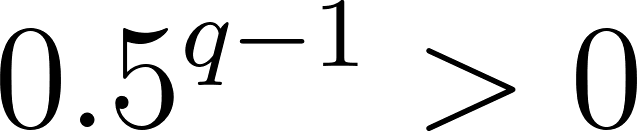
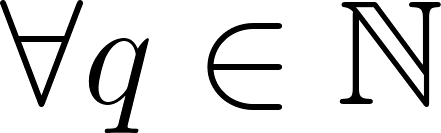
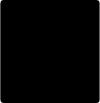
[](https://www.codecogs.com/eqnedit.php?latex=%20%7B1-0.5%5Eq%20%5Cover%201-0.5%7D%20-%201%20%3C%201%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20%7B1-0.5%5Eq%20%5Cover%200.5%7D%20-1%20%3C%201%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20%7B1-0.5%5Eq%20%5Cover%200.5%7D%20%3C%202%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%202%20-%200.5%5E%7Bq-1%7D%20%3C%202%20#0)

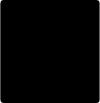
[](https://www.codecogs.com/eqnedit.php?latex=%20-%200.5%5E%7Bq-1%7D%20%3C%200%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%200.5%5E%7Bq-1%7D%20%3E%200%20#0), lo cual es válido [](https://www.codecogs.com/eqnedit.php?latex=%20%5Cforall%20q%20%5Cin%20%5Cmathbb%7BN%7D%20#0)[](https://www.codecogs.com/eqnedit.php?latex=%20%5Cblacksquare%20#0)

Ahora demostramos que la igualdad sólo se cumpliría con un número infinito de símbolos

Por propiedad de la serie geométrica:

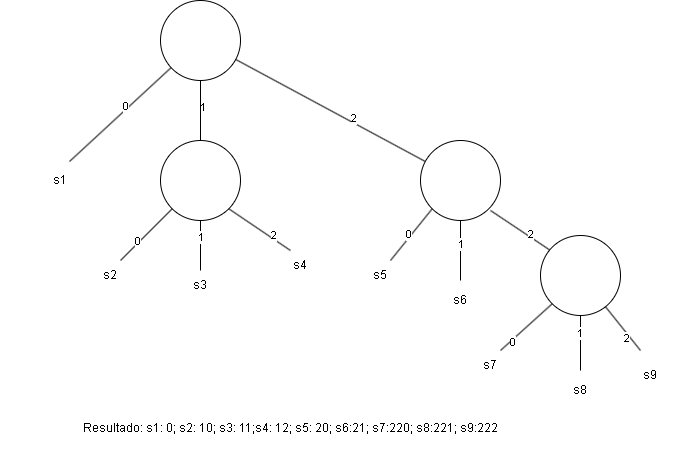
Para este caso tenemos que a = 1 y r = 0.5

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Cblacksquare%20#0)

1. Sea un código trinario. Codificar los diez dígitos decimales con longitudes 1, 2, 2, 2, 2, 2, 3, 3, 3,3. Si no se pudiera ¿hasta cuantos dígitos podría codificar?, Realice el árbol de codificación.

No se pueden codificar los diez dígitos con las longitudes pedidas, sin embargo, se pueden codificar nueve dígitos, el primero con longitud 1, los siguientes cinco de longitud 2 y los últimos tres de longitud 3.

Árbol resultante:

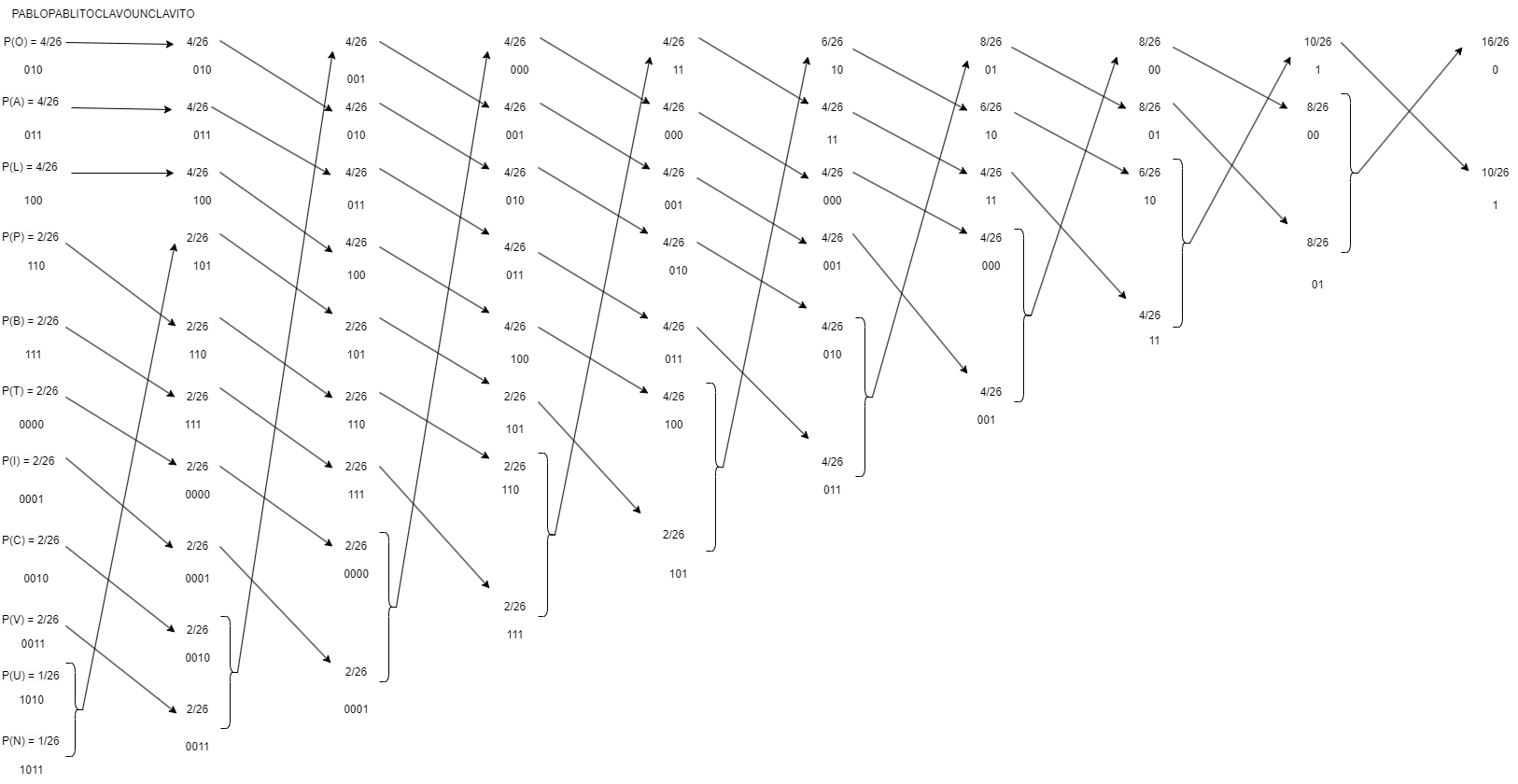


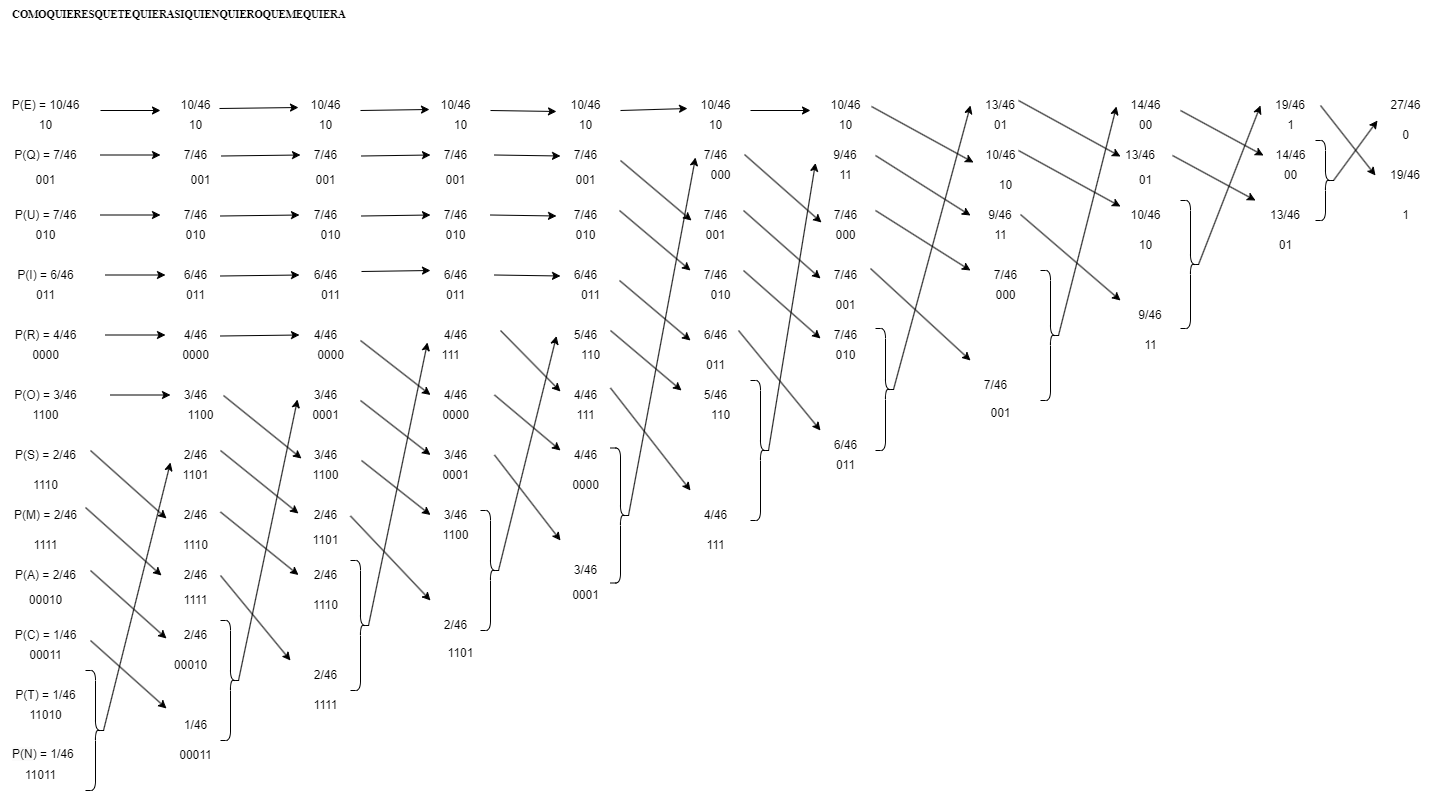
1. Sean los siguientes textos:

| **PABLOPABLITOCLAVOUNCLAVITO** |
| --- |
| **COMOQUIERESQUETEQUIERA SIQUIENQUIEROQUEMEQUIERA** |

* 1. Realizar la codificación usando los siguientes métodos:
     1. Estáticos: Huffman - Shannon - Fano
     2. Dinámicos: LZ - LZW – Huffman - Aritmético
  2. Para cada uno de los métodos calcular el porcentaje de compresión.
  3. Comparar los porcentajes obtenidos en cada uno de los métodos.

ESTÁTICOS:  
HUFFMAN

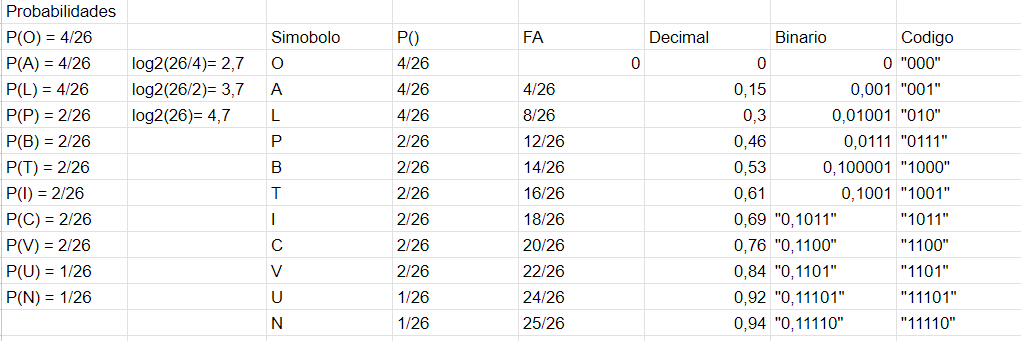


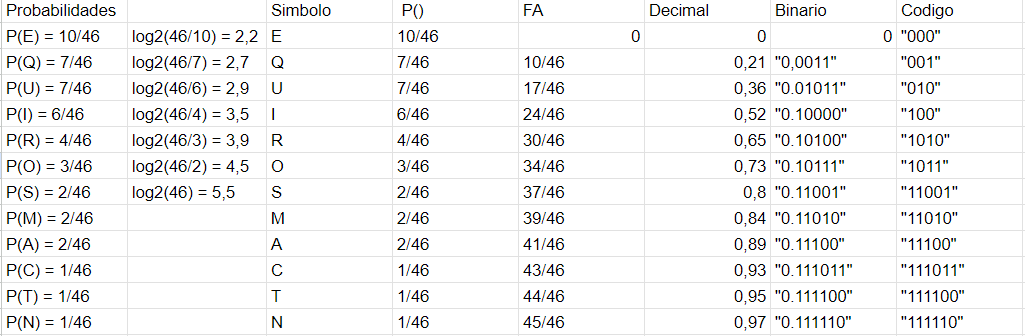


Porcentaje de compresión Pablito: = 0.4230769231

Porcentaje de compresión Quiera: = 0.3695652174

Shannon





Porcentaje de compresión Pablito =

Porcentaje de compresión Quien=

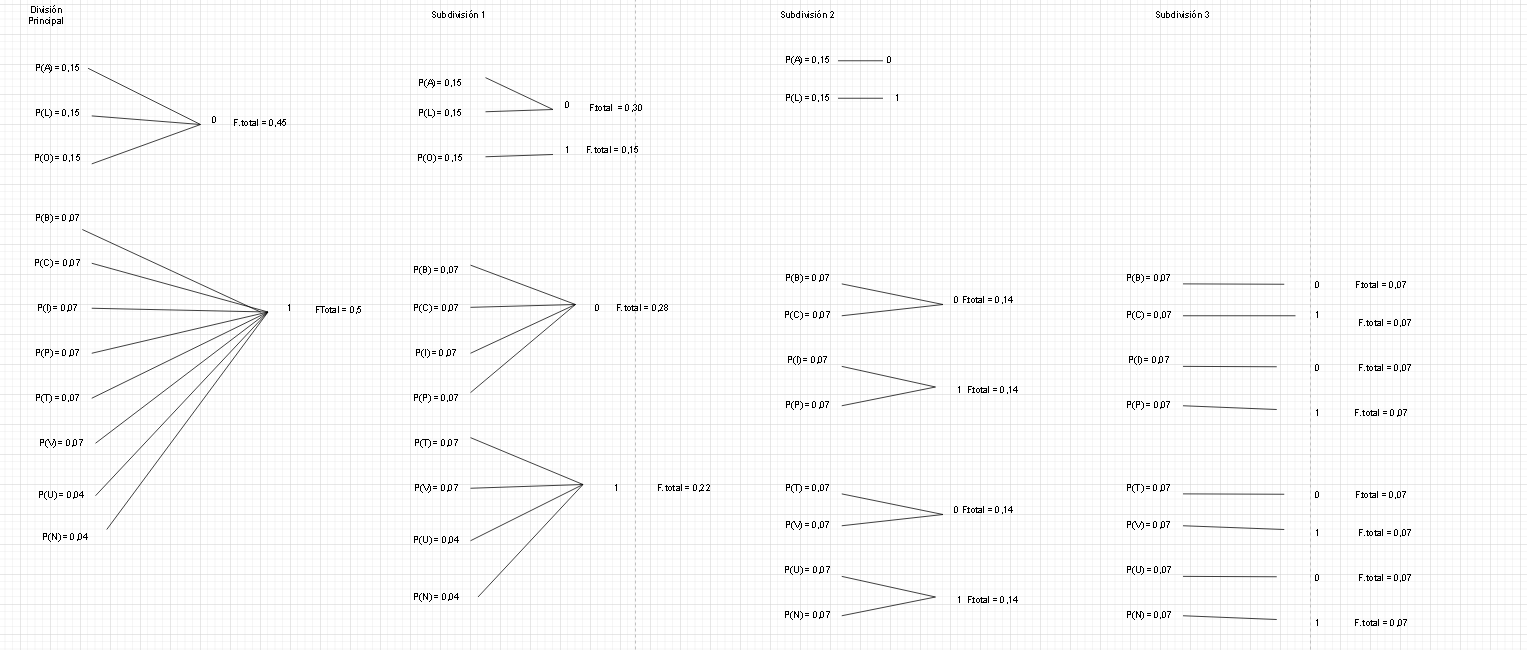
Fanno:

Texto 1:

Obtenemos las probabilidades de cada uno de los carácteres del texto:

P(P)= 2/26= 0,07; P(A)=4/26 = 0.15; P(B) = 0,07; P(L) = 0,15; P(I)= 0,07; P(T) = 0,07; P(O)=0,15; P(C) = 0,07; P(V) = 0,07; P(U)=1/26 = 0,04; P(N) = 0,04

Creamos las subdivisiones y obtenemos:



Codificación resultante:

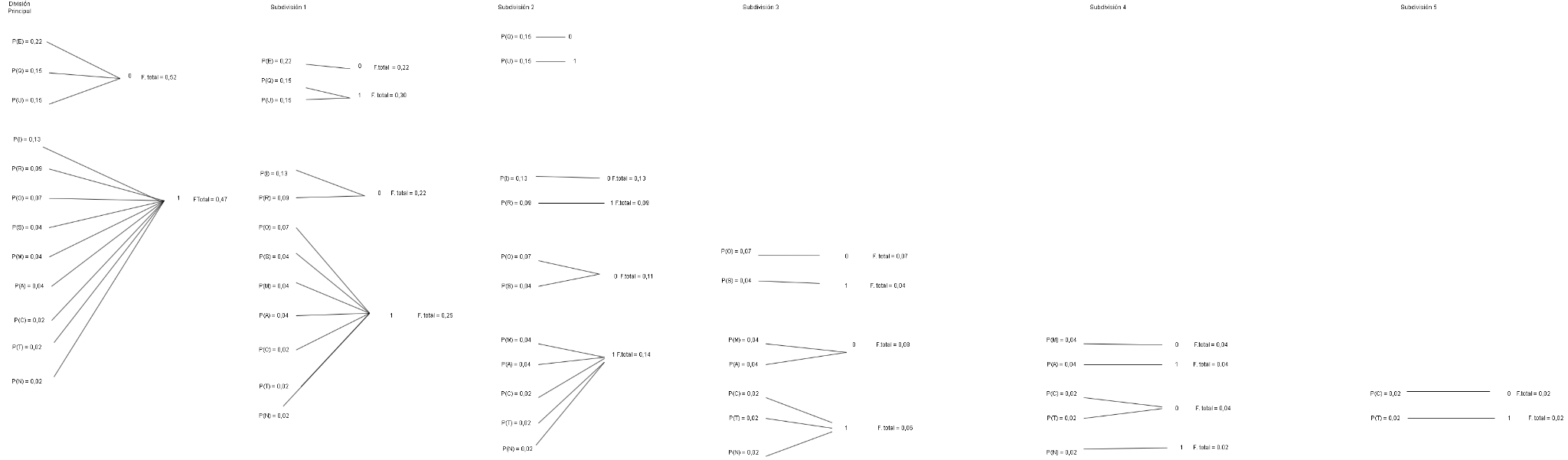
| Símbolo | Código |
| --- | --- |
| A | 000 |
| L | 001 |
| O | 01 |
| B | 1000 |
| C | 1001 |
| I | 1010 |
| P | 1011 |
| T | 1100 |
| V | 1101 |
| U | 1110 |
| N | 1111 |

Texto 2:

Obtenemos las probabilidades de cada símbolo:

P(E) = 10/46 = 0,22; P(Q) = 7/46 = 0,15; P(U) = 0,15; P(I) = 6/46 = 0,13; P(R) = 4/46 = 0,09; P(O) = 3/46 = 0,07; P(S) = 2/46 = 0,04; P(M) = 0,04; P(A) = 0,04; P(C) = 1/46 = 0,02; P(T) = 0,02; P(N) = 0,02

Creamos las subdivisiones y obtenemos:



Codificación resultante

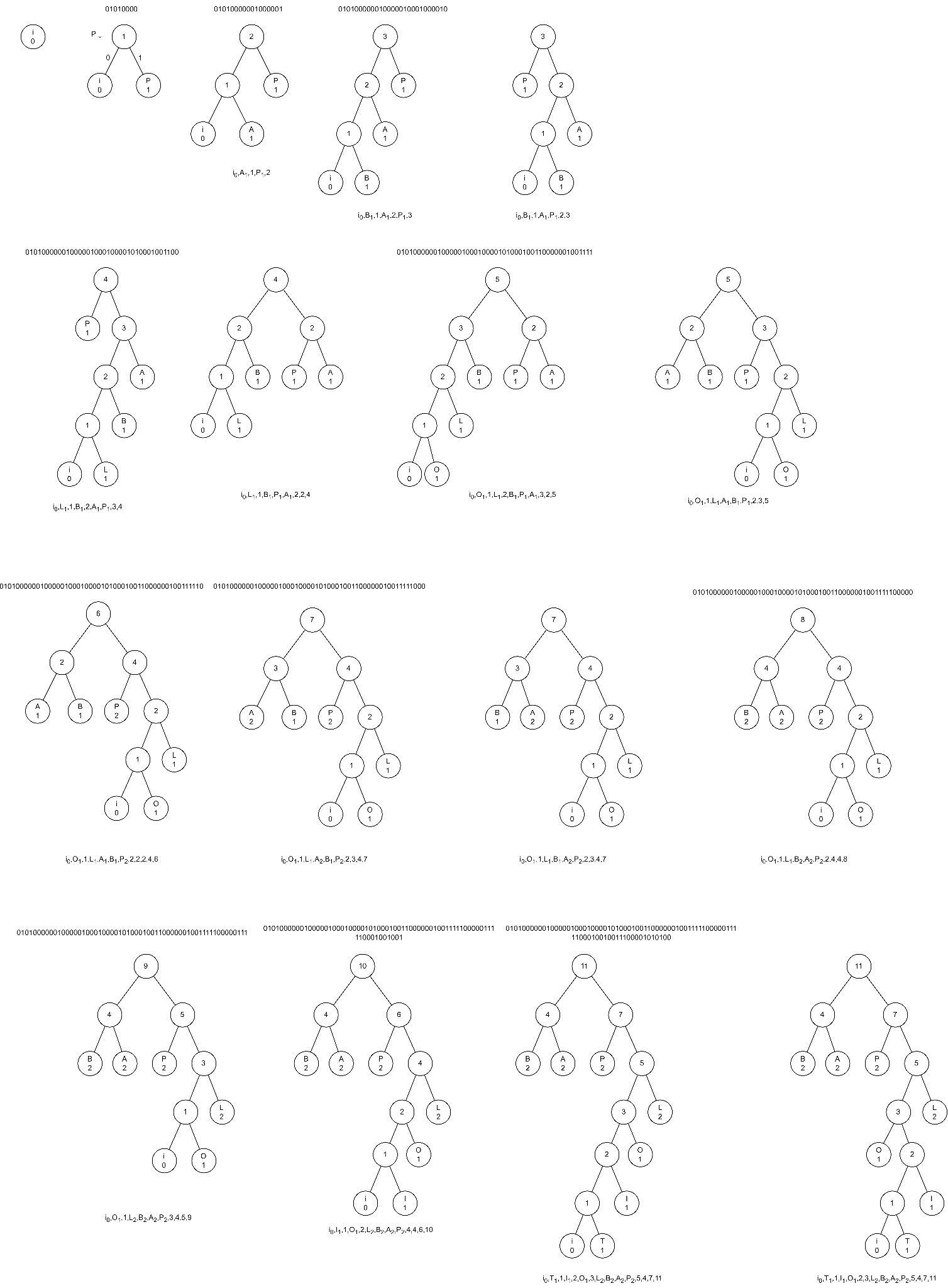
| Símbolo | Código |
| --- | --- |
| E | 00 |
| Q | 010 |
| U | 011 |
| I | 100 |
| R | 101 |
| O | 1100 |
| S | 1101 |
| M | 11100 |
| A | 11101 |
| C | 111100 |
| T | 111101 |
| N | 11111 |

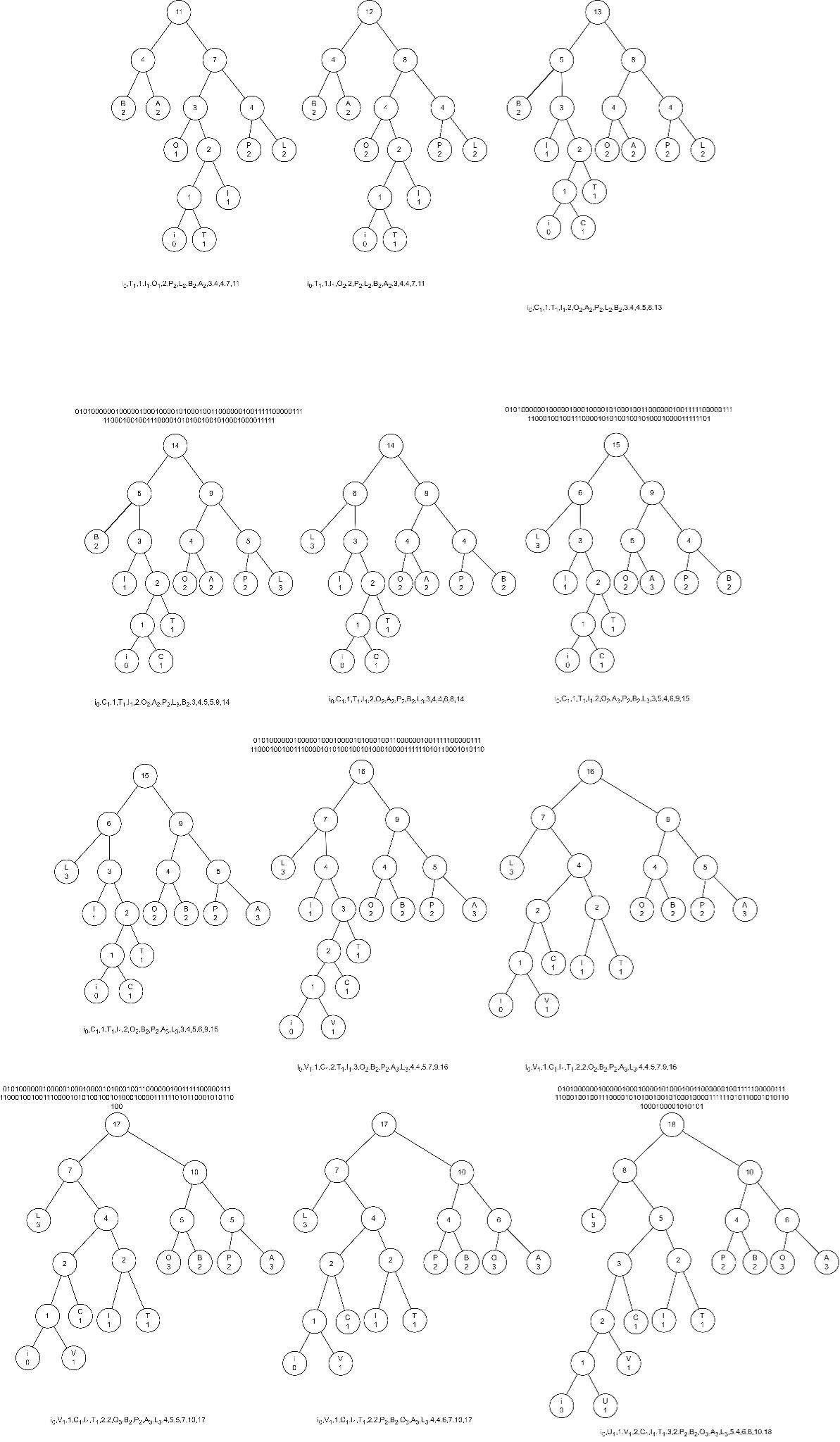
Porcentaje de compresión Pablito =

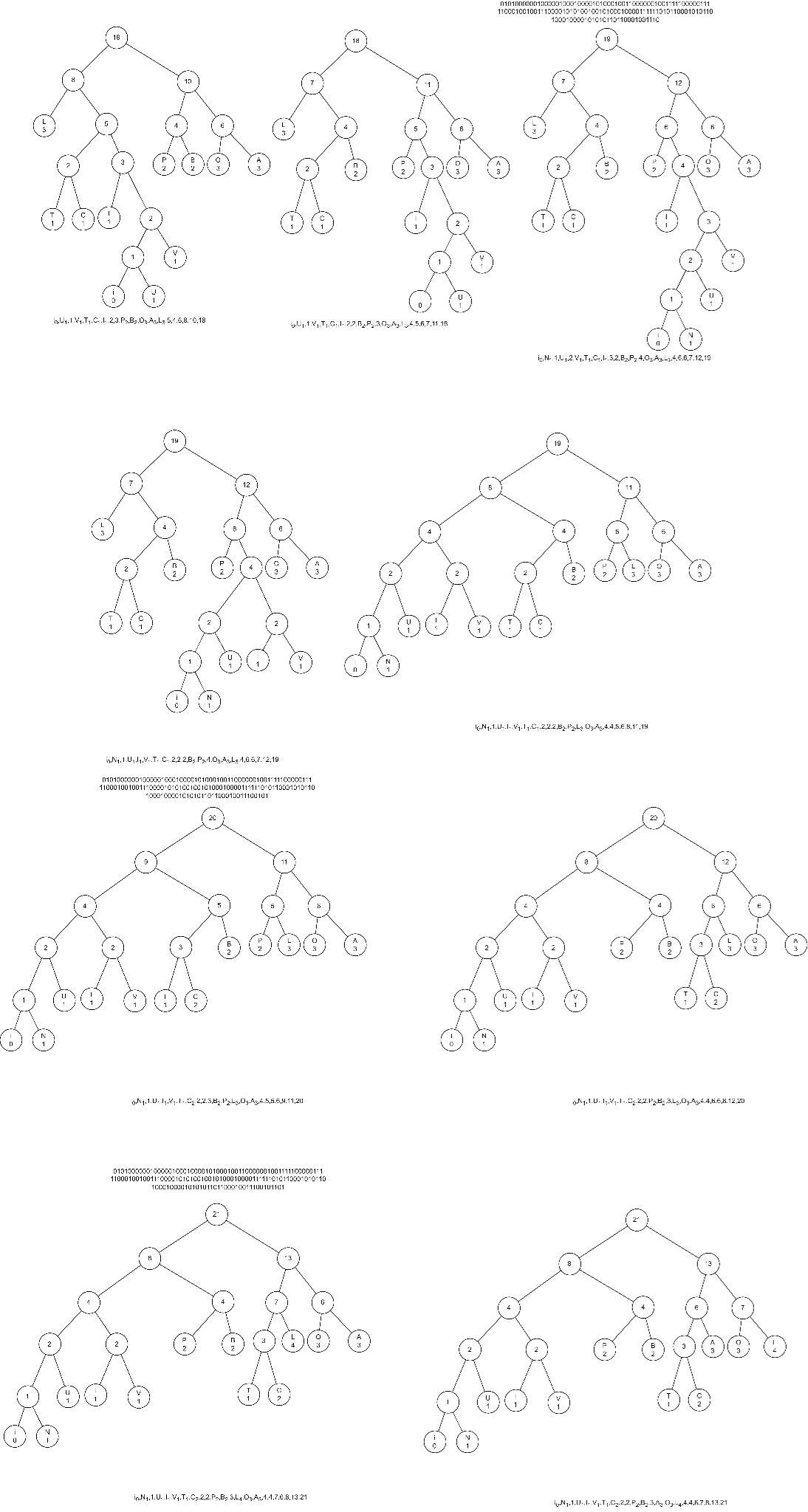
Porcentaje de compresión Quien=

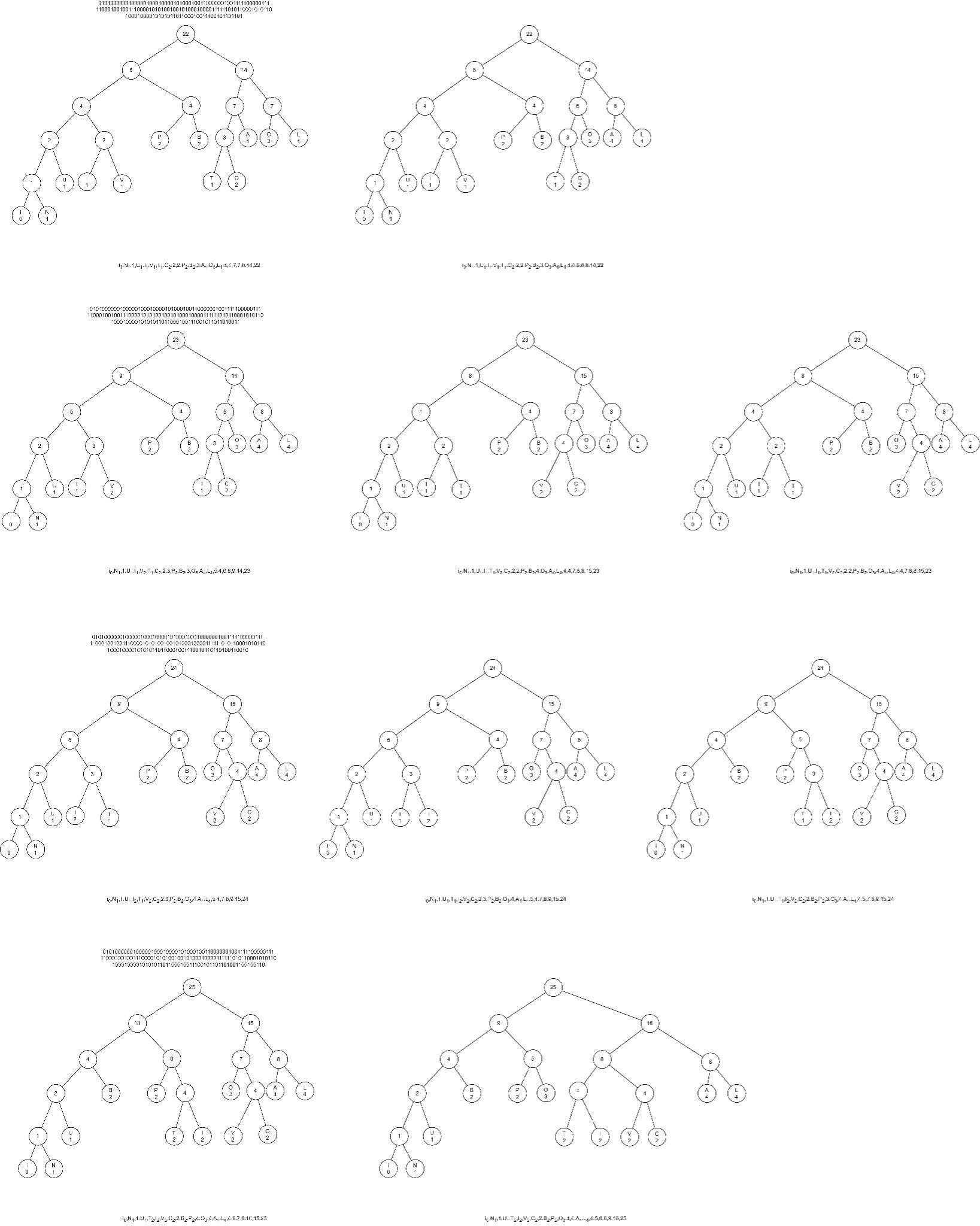
DINÁMICOS:

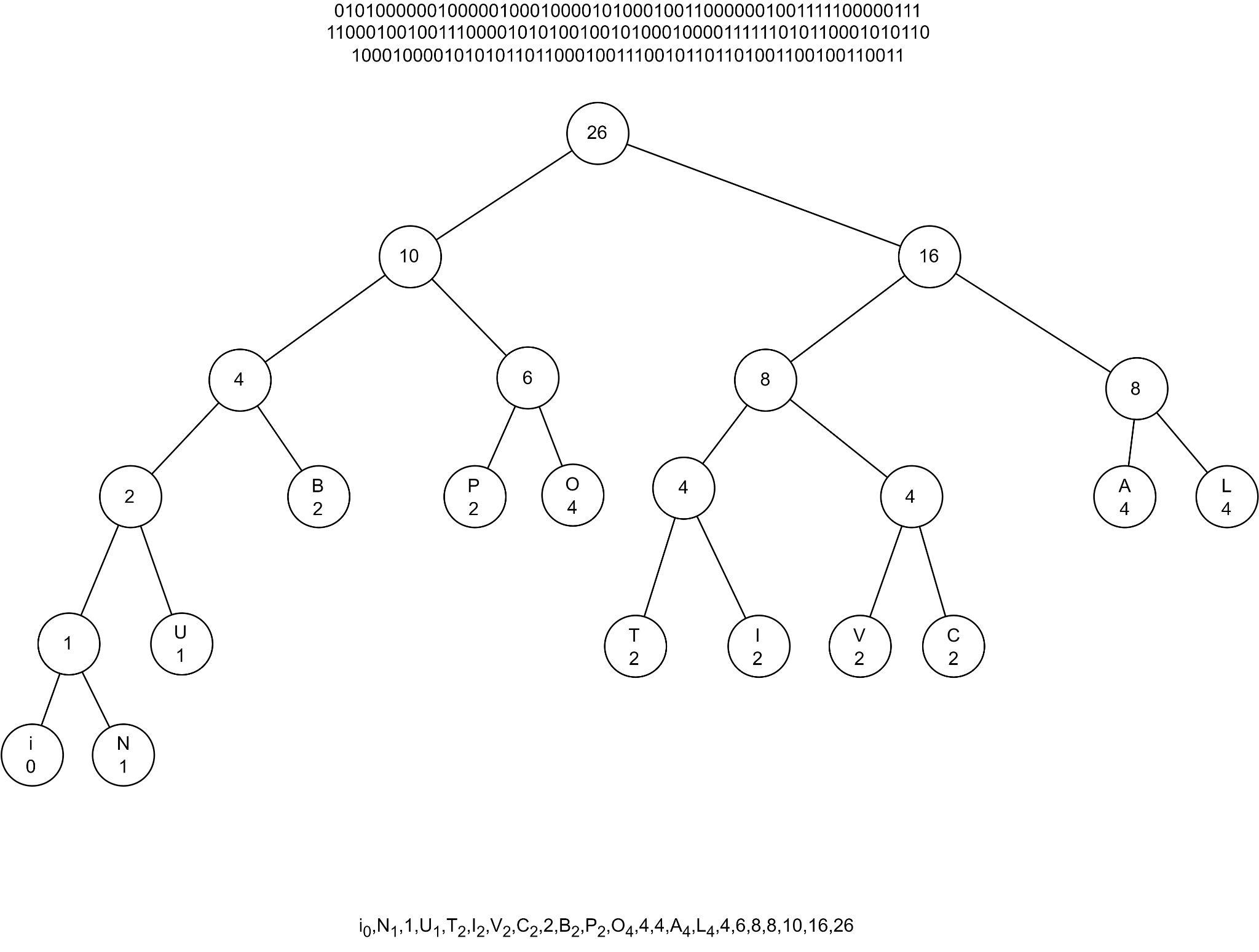
HUFFMAN











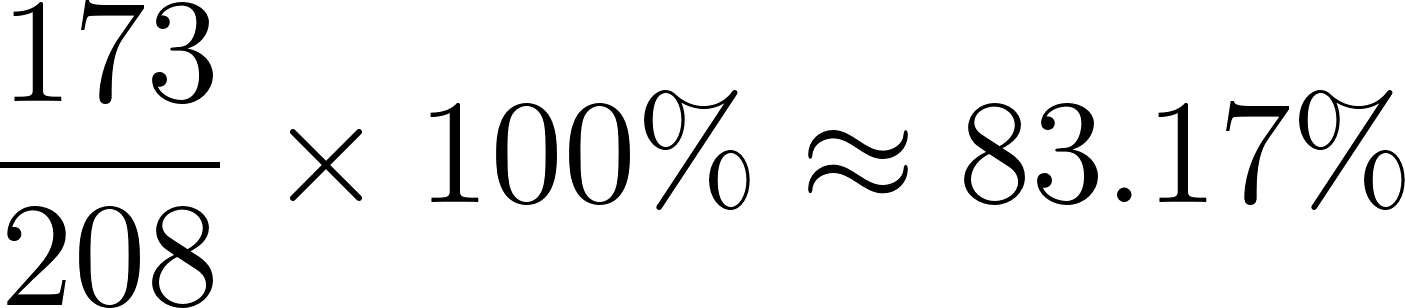
La secuencia de bits obtenida para la primer cadena tras la compresión con Huffman dinámico es:

01010000001000001000100001010001001100000010011111000001111100010010011100001010100100101000100001111110101100010101101000100001010101101100010011100101101101001100100110011

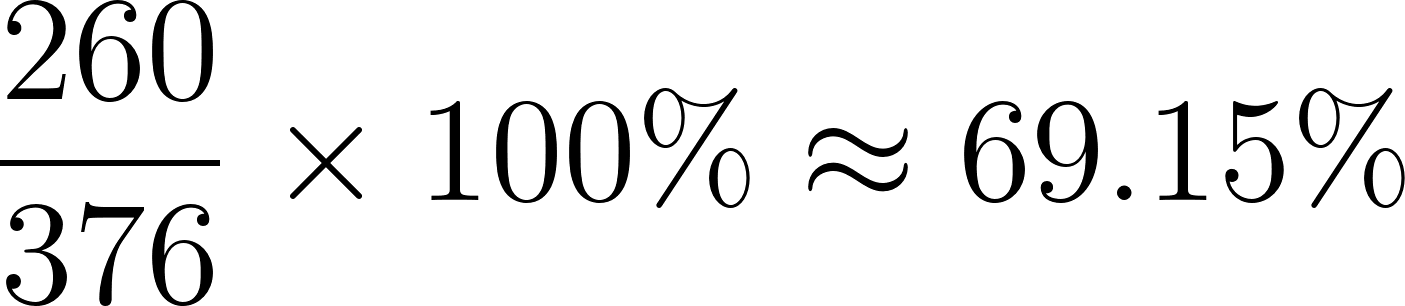
Para la segunda cadena se creó un script que realizara la compresión, obteniéndose:

01000011001001111000100110111100010100011100010101011000010010010100010001010000010100101111111000101001100011101101010001010100001000111000001000000000100000110010000100000000111111011000100011000001001110101100011000110011110010000011100011010110000101100110

Porcentaje de compresión de Pablito:

[](https://www.codecogs.com/eqnedit.php?latex=%20%7B173%20%5Cover%20208%7D%20%5Ctimes%20100%25%20%5Capprox%2083.17%25%20#0)

Porcentaje de compresión de Quien:

[](https://www.codecogs.com/eqnedit.php?latex=%20%7B260%20%5Cover%20376%7D%20%5Ctimes%20100%25%20%5Capprox%2069.15%25%20#0)

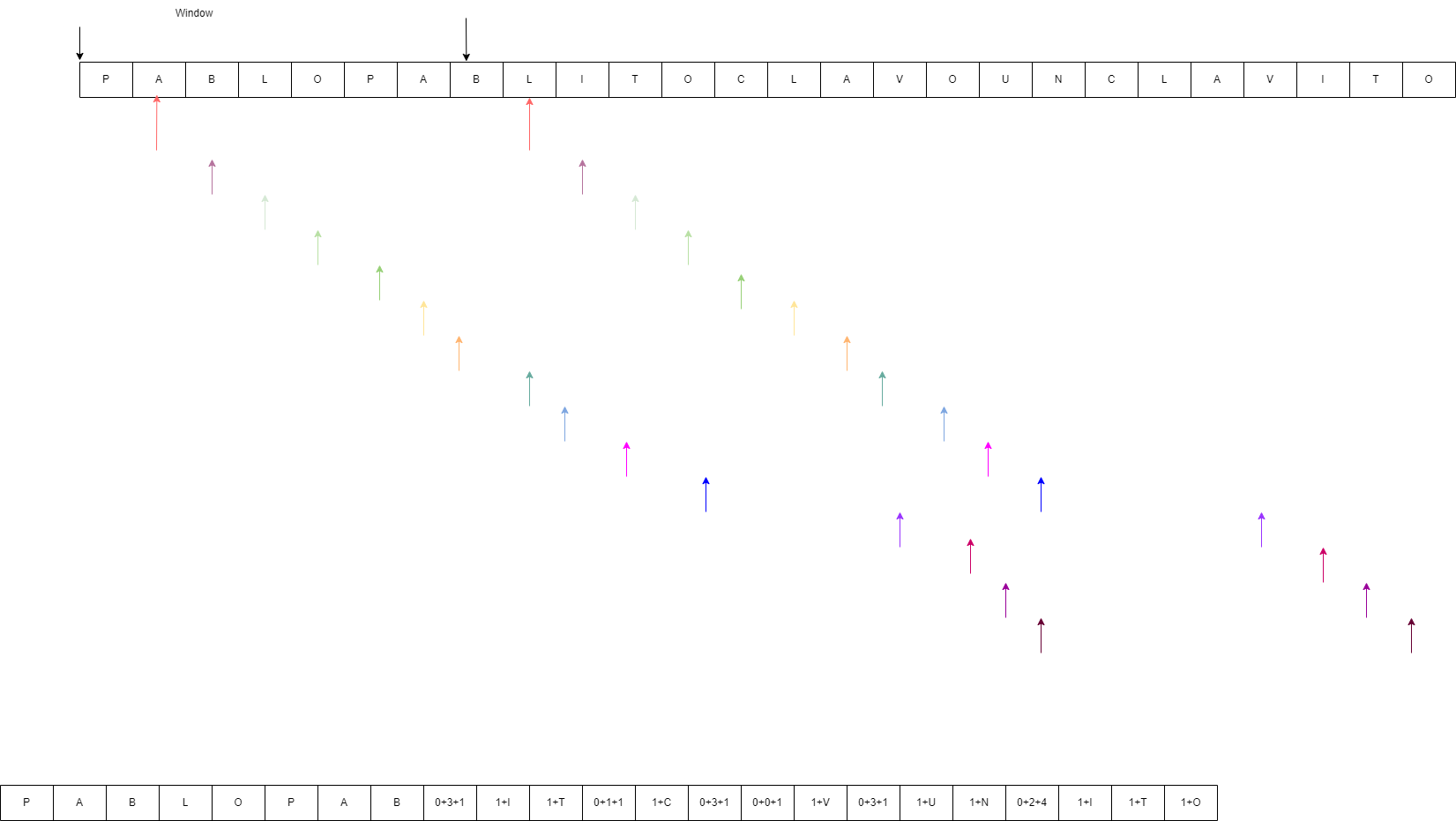
Algoritmo LZ:

Cadena 1:

Tamaño de ventana= 8

cantidad de bits para representar el flag = 1; cantidad de bits para representar los desplazamientos = 2; cantidad de bits para representar la cantidad de coincidencias = 2;

Total de bits para cada carácter = 5



Tamaño archivo original = 8\*26 = 208 bits

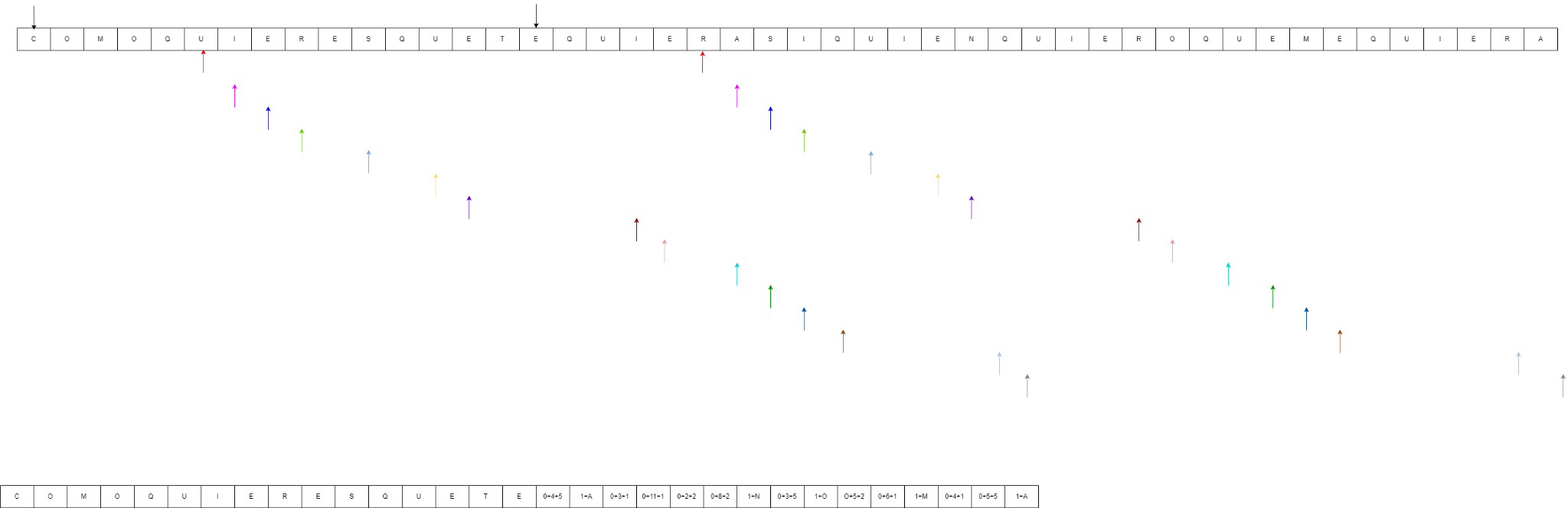
Tamaño de archivo comprimido = 5\*23 = 115 bits

Porcentaje de compresión = (1-

Cadena 2:

Tamaño de ventana = 16; Cantidad de bits para los desplazamientos = 4; cantidad de bits para el flag = 1; Cantidad de bits para las coincidencias = 3

Cantidad total de bits por cada caracter = 8



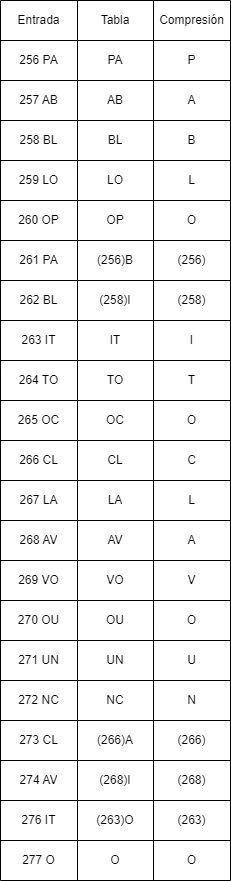
Tamaño del archivo original = 8\*46 = 368 bits

Tamaño del archivo comprimido = 8\*31 = 248 bits

Porcentaje de compresión = (1-

Algoritmo LZW:

Cadena 1



Cantidad de bits para los caracteres cuyo código ASCII es menor a 256 = 8 bits

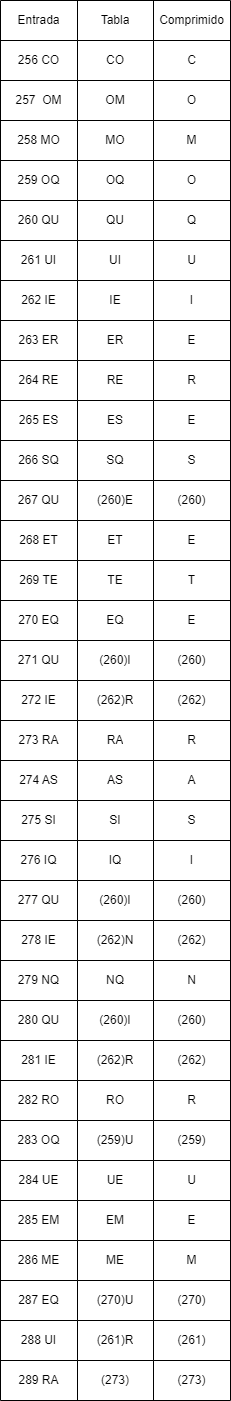
Para los mayores e iguales a 256 = 9 bits

Tamaño del archivo original = 8 bits x 26 = 208 bits

Tamaño del archivo comprimido = (8 bits x 16) + (9 bits x 5) = 173 bits

Porcentaje de compresión = (1-

Cadena 2:



Cantidad de bits para los caracteres cuyo código ASCII es menor a 256 = 8 bits

Para los mayores e iguales a 256 = 9 bits

Tamaño del archivo original = 8 bits x 46= 368 bits

Tamaño del archivo comprimido = (8 bits x 23) + (9 bits x 11) = 283 bits

Porcentaje de compresión = (1-

Algoritmo aritmético (Los cálculos se realizaron a través de código):

PABLOPABLITOCLAVOUNCLAVITO:

Codigo ; 0.45029877782024625583601148199068430728793765432448

Tamaño del texto original: 27 B

Tamaño código: 104 B

Porcentaje de compresión: (1- 104/27 ) \* 100= -285.1851852%

**COMOQUIERESQUETEQUIERA SIQUIENQUIEROQUEMEQUIERA:**

Codigo: 0.9520377532921862689512056444354239380775735913129250090455246490825545385189719504648646021813831845

Tamaño texto original: 46 B

Tamaño código: 152 B

Porcentaje de compresión: (1- 152/46 ) \* 100 = -230.4347826%

En ambos casos por lo corto del texto, por utilizar un solo código para todo el texto y por la implementación en python no se consigue una compresión real del texto

**FUENTES DE MARKOV**

1. Dado el siguiente enunciado construir un conjunto de probabilidades condicionadas (con cuatro estados) en un proceso de Markov de primer orden, encontrar la matriz de transición y el grafo.

**aaddasadsadsdssadasasadadas|a**

|  | **a** | **d** | **s** |
| --- | --- | --- | --- |
| **a** | **2/12** | **6/12** | **4/12** |
| **d** | **4/8** | **1/8** | **3/8** |
| **s** | **6/8** | **1/8** | **1/8** |

**Huffman a cada columna para encontrar los árboles**

1. A partir del ejercicio n°1
   1. Obtener la matriz estacionaria (distribución a la que tiende).

|  | **a** | **d** | **s** |
| --- | --- | --- | --- |
| **a** | **0.42857142** | **0.28571429** | **0.28571428** |
| **d** | **0.42857142** | **0.28571429** | **0.28571428** |
| **s** | **0.42857142** | **0.28571429** | **0.28571428** |

* 1. Comprobar que las potencias sucesivas de la matriz de transición tienen menor variabilidad en los valores de sus componentes.

Matriz al cuadrado

|  | **a** | **d** | **s** |
| --- | --- | --- | --- |
| **a** | **0.52777778** | **0.1875** | **0.28472222** |
| **d** | **0.42708333** | **0.3125** | **0.26041667** |
| **s** | **0.28125** | **0.40625** | **0.3125** |

Matriz al cubo

|  | **a** | **d** | **s** |
| --- | --- | --- | --- |
| **a** | **0.39525463** | **0.32291667** | **0.2818287** |
| **d** | **0.42274306** | **0.28515625** | **0.29210069** |
| **s** | **0.484375** | **0.23046875** | **0.28515625** |

Matriz a la cuarta

|  | **a** | **d** | **s** |
| --- | --- | --- | --- |
| **a** | **0.43870563** | **0.27322049** | **0.28807388** |
| **d** | **0.43211082** | **0.28352865** | **0.28436053** |
| **s** | **0.40983073** | **0.30664062** | **0.28352865** |

Como se puede observar, cada vez los valores de las distintas filas son más similares entre sí.

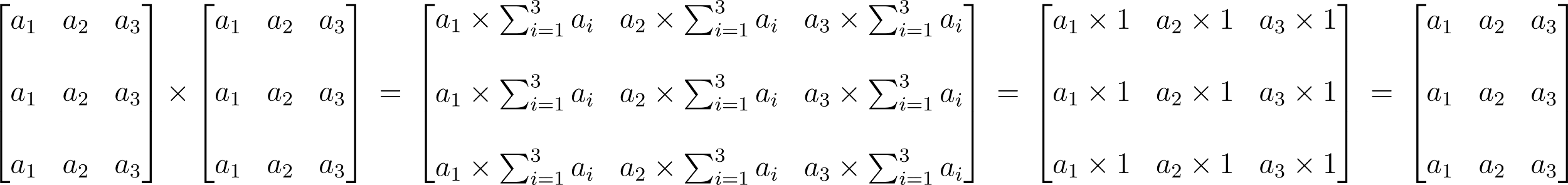
* 1. Comprobar que la distribución estacionaria es la misma para el cuadrado de la matriz de transición.

Al elevar al cuadrado la matriz correspondiente a la distribución estacionaria, se obtiene la siguiente matriz

|  | **a** | **d** | **s** |
| --- | --- | --- | --- |
| **a** | **0.42857142** | **0.28571429** | **0.28571428** |
| **d** | **0.42857142** | **0.28571429** | **0.28571428** |
| **s** | **0.42857142** | **0.28571429** | **0.28571428** |

La cual es la matriz de la distribución estacionaria en sí misma

También se puede demostrar formalmente

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Cbegin%7Bbmatrix%7Da_1%20%26%20a_2%20%26%20a_3%5C%5C%5C%5C%5C%5C%20a_1%20%26%20a_2%20%26%20a_3%5C%5C%5C%5C%5C%5C%20a_1%20%26%20a_2%20%26%20a_3%20%5Cend%7Bbmatrix%7D%5C%5C%5C%5C%20%5Ctimes%5C%5C%5C%5C%20%5Cbegin%7Bbmatrix%7Da_1%20%26%20a_2%20%26%20a_3%5C%5C%5C%5C%5C%5C%20a_1%20%26%20a_2%20%26%20a_3%5C%5C%5C%5C%5C%5C%20a_1%20%26%20a_2%20%26%20a_3%20%5Cend%7Bbmatrix%7D%5C%5C%5C%5C%20%3D%5C%5C%5C%5C%20%5Cbegin%7Bbmatrix%7Da_1%20%5Ctimes%20%5Csum_%7Bi%3D1%7D%5E3%7Ba_i%7D%26%20a_2%20%5Ctimes%20%5Csum_%7Bi%3D1%7D%5E3%7Ba_i%7D%20%26%20a_3%20%5Ctimes%20%5Csum_%7Bi%3D1%7D%5E3%7Ba_i%7D%5C%5C%5C%5C%5C%5C%20a_1%20%5Ctimes%20%5Csum_%7Bi%3D1%7D%5E3%7Ba_i%7D%20%26%20a_2%20%5Ctimes%20%5Csum_%7Bi%3D1%7D%5E3%7Ba_i%7D%20%26%20a_3%20%5Ctimes%20%5Csum_%7Bi%3D1%7D%5E3%7Ba_i%7D%5C%5C%5C%5C%5C%5C%20a_1%20%5Ctimes%20%5Csum_%7Bi%3D1%7D%5E3%7Ba_i%7D%20%26%20a_2%20%5Ctimes%20%5Csum_%7Bi%3D1%7D%5E3%7Ba_i%7D%20%26%20a_3%20%5Ctimes%20%5Csum_%7Bi%3D1%7D%5E3%7Ba_i%7D%20%5Cend%7Bbmatrix%7D%5C%5C%5C%5C%20%3D%5C%5C%5C%5C%20%5Cbegin%7Bbmatrix%7Da_1%20%5Ctimes%201%20%26%20a_2%20%5Ctimes%201%20%26%20a_3%20%5Ctimes%201%5C%5C%5C%5C%5C%5C%20a_1%20%5Ctimes%201%20%26%20a_2%20%5Ctimes%201%20%26%20a_3%20%5Ctimes%201%5C%5C%5C%5C%5C%5C%20a_1%20%5Ctimes%201%20%26%20a_2%20%5Ctimes%201%20%26%20a_3%20%5Ctimes%201%20%5Cend%7Bbmatrix%7D%5C%5C%5C%5C%20%3D%5C%5C%5C%5C%20%5Cbegin%7Bbmatrix%7Da_1%20%26%20a_2%20%26%20a_3%5C%5C%5C%5C%5C%5C%20a_1%20%26%20a_2%20%26%20a_3%5C%5C%5C%5C%5C%5C%20a_1%20%26%20a_2%20%26%20a_3%20%5Cend%7Bbmatrix%7D%20#0)

* 1. ¿Es un proceso ergódico?

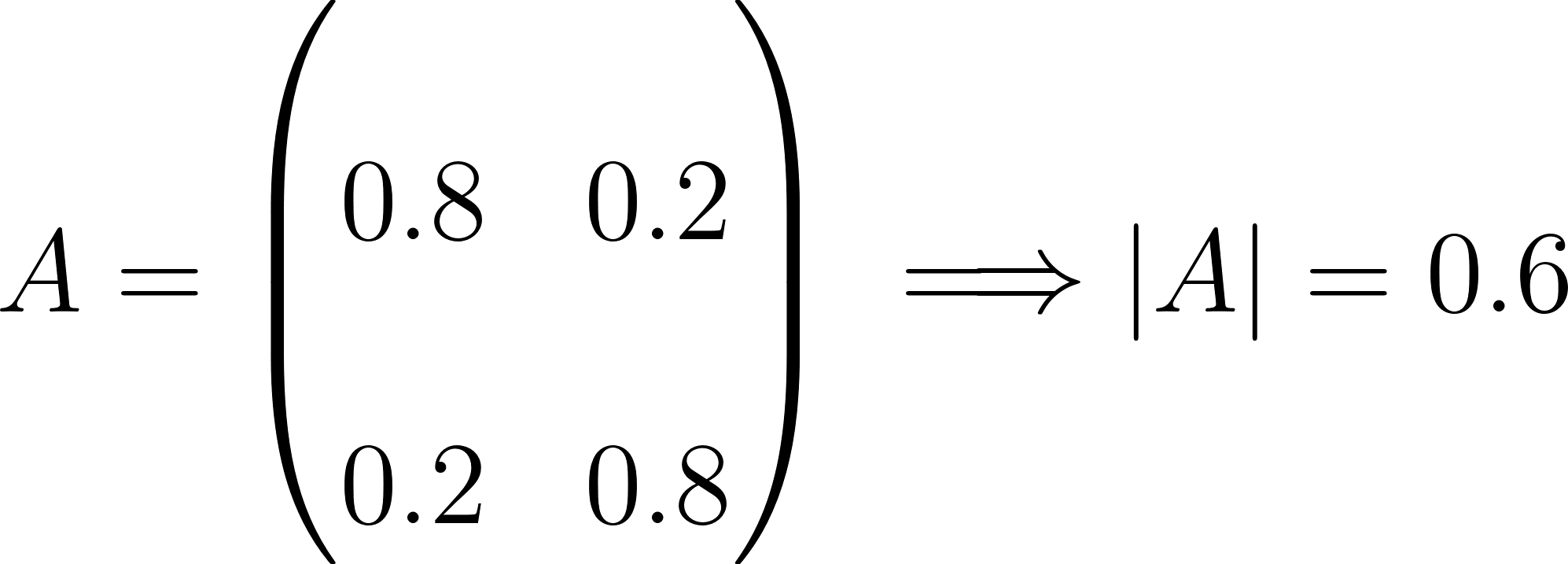
cuando todos los elementos de la matriz de transición son distintos de cero el proceso es ergódico, cualquier transición es factible sin tiempo infinito y P = 1

Probabilidad de (adadad)= p(a)\*p(a/a)\*p(d/a)\*p(a/d)\*p(d/a)\*p(a/d)\*p(d/a)+ p(d)\*p(a/d)\*p(d/a)\*p(a/d)\*p(d/a)\*p(a/d)\*p(d/a)+p(s)\*p(a/s)\*p(d/a)\*p(a/d)\*p(d/a)\*p(a/d)\*p(d/a)

**0.42857142\*2/12\*1/2\*1/2\*1/2\*1/2\*1/2+0.28571429\*1/2\*1/2\*1/2\*1/2\*1/2\*1/2+0.28571428\*4/12\*1/2\*1/2\*1/2\*1/2\*1/2**= 0.00967261901

1. Demostrar que las filas de la matriz de transición son linealmente dependientes en un proceso de Markov.

Las filas de una matriz de transición de un proceso de Markov no son necesariamente linealmente dependientes, por ejemplo, considérese el siguiente caso particular de una matriz de transición.

[](https://www.codecogs.com/eqnedit.php?latex=%20A%20%3D%20%7B%5Cbegin%7Bpmatrix%7D%5C%5C%5C%5C%200.8%20%26%200.2%20%5C%5C%5C%5C%5C%5C%200.2%20%26%200.8%5C%5C%5C%5C%20%5Cend%7Bpmatrix%7D%7D%20%5C%5C%5C%5C%20%5Cimplies%5C%5C%5C%5C%20%7CA%7C%20%3D%200.6%20#0)

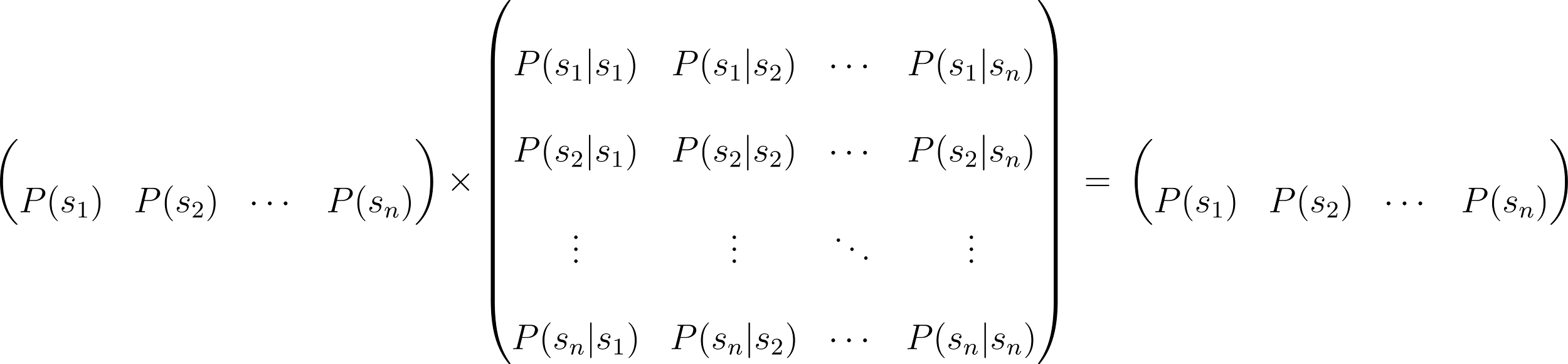
Como se puede ver, el determinante de esta matriz es distinto de cero, por lo que se trata de un contraejemplo de lo que se pidió demostrar y dicha demostración no es posible.

1. Dado el siguiente ejemplo decir:
   1. ¿De qué orden es el sistema de Markov?.  
      Orden 3
   2. Sea S = (s1,s2,s3,s4) p(si/sj1,sj2,sj3)
   3. ¿En cuántos estados puede encontrarse el sistema?

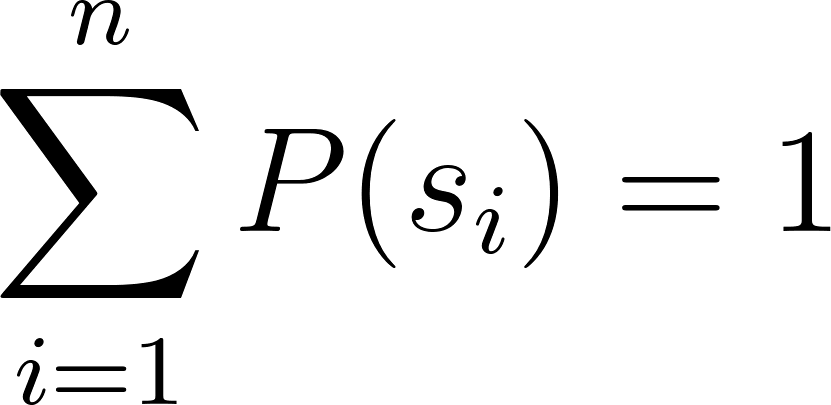
64\*64

* 1. Supuestas conocidas todas las probabilidades condicionales ¿cuál será la forma del sistema de ecuaciones para encontrar las p(si)?

La forma de este sistema de ecuaciones es

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Cbegin%7Bpmatrix%7D%5C%5C%5C%5C%20P(s_1)%20%26%20P(s_2)%20%26%20%5Ccdots%20%26%20P(s_n)%5C%5C%5C%5C%20%5Cend%7Bpmatrix%7D%20%5C%5C%5C%5C%20%5Ctimes%5C%5C%5C%5C%20%5Cbegin%7Bpmatrix%7D%5C%5C%5C%5C%20P(s_1%20%7C%20s_1)%20%26%20P(s_1%20%7C%20s_2)%20%26%20%5Ccdots%20%26%20P(s_1%20%7C%20s_n)%20%5C%5C%5C%5C%5C%5C%20P(s_2%20%7C%20s_1)%20%26%20P(s_2%20%7C%20s_2)%20%26%20%5Ccdots%20%26%20P(s_2%20%7C%20s_n)%20%5C%5C%5C%5C%5C%5C%20%5Cvdots%20%26%20%5Cvdots%20%26%20%5Cddots%20%26%20%5Cvdots%20%5C%5C%5C%5C%5C%5C%20P(s_n%20%7C%20s_1)%20%26%20P(s_n%20%7C%20s_2)%20%26%20%5Ccdots%20%26%20P(s_n%20%7C%20s_n)%5C%5C%5C%5C%20%5Cend%7Bpmatrix%7D%5C%5C%5C%5C%20%3D%5C%5C%5C%5C%20%5Cbegin%7Bpmatrix%7D%5C%5C%5C%5C%20P(s_1)%20%26%20P(s_2)%20%26%20%5Ccdots%20%26%20P(s_n)%5C%5C%5C%5C%20%5Cend%7Bpmatrix%7D%20%5C%5C%5C%5C%20#0)

Con la restricción

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Csum_%7Bi%3D1%7D%5En%20%7BP(s_i)%7D%20%3D%201%20#0)

1. Supongamos una máquina con dos estados, anda=1, no anda=2. Estudiamos el proceso día a día llegando a comprobar que:
   * 1. si anda hoy, hay un 75% de chances que ande mañana
     2. si no anda hoy la arreglan mañana
   1. Encontrar la matriz de transición

|  | A | NA |
| --- | --- | --- |
| A | 3/4 | 1/4 |
| NA | 1 | 0 |

* 1. Calcular hasta la quinta potencia de la matriz de transición y analizar los resultados

|  | A | NA |
| --- | --- | --- |
| A | 0.79980469 | 0.20019531 |
| NA | 0.80078125 | 0.19921875 |

* 1. Encontrar la matriz de equilibrio o estacionaria

|  | A | NA |
| --- | --- | --- |
| A | 4/5 | 1/5 |
| NA | 4/5 | 1/5 |

1. Dado un proceso de Markov con tres estados a,b,c; con las siguientes probabilidades: p(a/a)=3/4; p(c/c)=p(b/b)=1/2; p(b/a)=p(c/a)=1/8; p(a/b)=p(c/b)=p(a/c)=p(b/c)=¼ comprobarque la longitud de Markov para un código compacto LM = 11/8 < 12/8 = L
2. Usando el texto del ejercicio 5, Práctico No 3
   1. Codificar mediante los métodos: Huffman, Fano y Shannon como fuente de Markov de orden 1.
   2. Relevar para la fuente la estadística de primer orden y realizar una codificación predictiva.
   3. Realizar una transformación Burrows-Wheller y codificar mediante cuenta de símbolos iguales.
   4. Obtener los porcentajes de compresión y comparar los resultados obtenidos.

**PRÁCTICO No 4**

1. Comprobar para los siguientes conjuntos de probabilidades que Hr <= l.
   1. p1 = 0,4; p2 = 0,2; p3 = 0,2; p4 = 0,1; p5 = 0,1
   2. p1 = 0,4; p2 = 0,4; p3 = 0,1; p4 = 0,1
   3. p1 = 0,4; p2 = 0,3; p3 = 0,2; p4 = 0,1
2. Sea una fuente S con eventos s1, s2, s3, s4 de p(si) = 1/4 ¿es posible encontrar un L=1,86 para cada codificación? ¿Cuál es el mínimo valor que en ese caso puede tomar L?

Sabemos que por lo que empezamos por comprobar el valor de la entropía para la fuente.

H = 1/4\*log2(4)\*4 = log2(4) = 2

Por lo tanto No es posible conseguir un L de 1,86, ya que como minimo debera de ser de 2

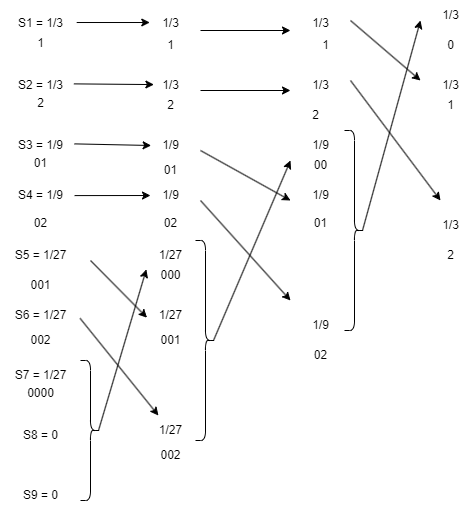
1. Repetir el ejemplo anterior pero con p(s1)=1/2; p(s2)=1/4; p(s3)=p(s4)=1/8

H = 1/2\*log2(2)+ 1/4\*log2(4)+ 1/8\*log2(8)\*2 = 1.75

Ahora si es posible conseguir un L de 1,86 con un mínimo de L = 1,75

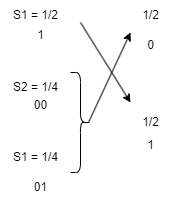
1. Sea la siguiente fuente S=(s1,s2,s3,s4,s5,s6,s7); p(s1)=p(s2)=1/3; p(s3)=p(s4)=1/9; p(s5)=p(s6)=p(s7)=1/27. Encontrar un código trinario, para ello encontrar H(S), L y codificar según Huffman.

H = 1/3 \* log3(3)\*2+1/9\*log3(9)\*3+1/27\*log3(27)\*3 = 1,66

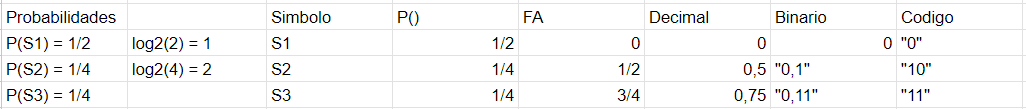


L = 1/3\*2+1/9\*2\*2+1/27\*3\*2+1/27\*4 = 1.481

1. Dada una fuente S=(s1,s2,s3); p(s1)=1/2; p(s2)=p(s3)=1/4. Codificar por Huffman, Shannon y Fano, encontrar LH, LS y LF.

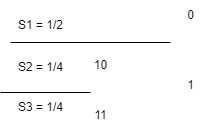


LH = 1/2+1/4\*2\*2 = 0,5



LS = 1/2+1/4\*2\*2 = 0,5

Fano



LF = 1/2\*1+1/4\*2\*2 = 0,5

1. Para el ejemplo anterior codificar por ambos métodos la segunda y tercera extensión viendo la tendencia de la longitud de la fuente original.
2. Dados los textos del ejercicio 5, Práctico No 3:

| **PABLOPABLITOCLAVOUNCLAVITO** |
| --- |
| **COMOQUIERESQUETEQUIERASIQUIENQUIEROQUEMEQUIERA** |

* 1. Codificar la segunda extensión usando el método de Fano.

La codificación se realizó con código en python:

Codigo extención 2: 10001100111010011010001111001010010011100111110010010101011011111011100011011011111

Codigo extencion 1: 1100001000100111100001000101011110101110100100011100111111011111101001000111010111101011

* 1. Calcular el porcentaje de compresión y comparar los resultados con los obtenidos codificando Fano como fuente de Markov de orden 1.

Porcentaje de compresión ext1: 57.692307692307686

Porcentaje de compresión ext2: 60.09615384615385

* 1. Fundamente la diferencia obtenida en el apartado anterior.

Esto se justifica ya que:

Con 𝐿𝑛 la longitud media de un código compacto para la 𝑛 − é𝑠𝑖𝑚𝑎 extensión de la fuente

Por lo que:

Por eso al hacer compresiones de extensiones mayores mejora el porcentaje de compresión, en este caso extensión 1 y 2