

Práctica 1. “Universo”

Carmona Serrano Ian Carlo
Ingeniería en Inteligencia Artificial 5BM1
Teoría de la Computación , ESCOM- IPN

14 de Octubre 2023

1 Introduction

Este programa se enfoca en la generación y análisis de cadenas binarias con un valor "n" específico. El usuario puede definir "n" o dejar que el programa lo seleccione automáticamente dentro del rango $[0, 1000]$. Los resultados se almacenan en un archivo de texto y se graficaran

2 Marco Teórico

Dentro del campo de la teoría de la computación, se explora la noción de un "universo" de cadenas o lenguajes, en el cual se contemplan todas las posibles combinaciones de símbolos de un alfabeto dado. Un ejemplo concreto es el "universo" de todas las cadenas binarias (Σ^n), donde Σ representa el alfabeto binario $\{0, 1\}$, y "n" es la longitud de la cadena.

Cuando se analiza este universo, se llega a una comprensión fundamental en la teoría de la computación: que todos los programas de computadora posibles, y por extensión, todas las computaciones posibles, pueden considerarse como un subconjunto de este universo. Esto se debe a que un programa de computadora se puede ver como una cadena de símbolos (código fuente) que puede ser ejecutada para realizar cálculos o tareas específicas.

La teoría del "universo" de todos los programas posibles es fundamental en áreas como la teoría de la computabilidad, donde se exploran los límites de lo que es computable y lo que no lo es. La noción de este universo también está relacionada con el concepto de máquinas de Turing, que son modelos abstractos de computación utilizados para analizar la computabilidad y la complejidad de los problemas.

3 Desarrollo

Paso 1: Selección de Opción

El programa ofrecerá al usuario dos opciones:

1. Ingresar un valor "n" manualmente en el rango [0, 1000].
2. Calcular automáticamente un valor "n" dentro del mismo rango.

```
BIENVENIDO

1) ingrese k de forma manual
2) ingrese k de forma automatica

Seleccione la opción a realizar: 1
Seleccionaste la opción manual
```

Figure 1: Paso 1

Paso 2: Entrada de Datos

Si elige la opción de ingresar "n" manualmente, se le pedirá que introduzca un valor "n" en el rango [0, 1000].

```
BIENVENIDO

1) ingrese k de forma manual
2) ingrese k de forma automatica

Seleccione la opción a realizar: 1
Seleccionaste la opción manual
Ingrese la k: 28
El valor de k -> 28
Presione enter para continuar...
```

Figure 2: Paso 2

Paso 3: Generación de Conjuntos Binarios

A continuación, el programa generará el conjunto de cadenas binarias (Σ^n) utilizando el valor de "n" proporcionado por el usuario o calculado automáticamente.

Paso 4: Almacenamiento de Resultados

Los resultados se expresarán en notación de conjunto y se guardarán en un archivo de texto. Cada cadena se incluirá en el conjunto, representando así todas las combinaciones posibles de cadenas binarias de longitud "n".

33554410	111111111111111111101010
33554411	111111111111111111101011
33554412	111111111111111111101100
33554413	111111111111111111101101
33554414	111111111111111111101110
33554415	111111111111111111101111
33554416	111111111111111111100000
33554417	111111111111111111100001
33554418	111111111111111111100010
33554419	111111111111111111100011
33554420	111111111111111111100100
33554421	111111111111111111100101
33554422	111111111111111111100110
33554423	111111111111111111100111
33554424	111111111111111111100000
33554425	111111111111111111100001
33554426	111111111111111111100010
33554427	111111111111111111100011
33554428	111111111111111111100100
33554429	111111111111111111100101
33554430	111111111111111111100110
33554431	111111111111111111100111

Figure 3: Paso 4

Paso 5: Graficación del Número de "Unos"

1. Se calculará el número de unos en cada cadena binaria y se registrarán estos valores.
2. Se generará una gráfica con el eje x representando las cadenas binarias y el eje y representando el número de unos en cada cadena.

3. Se calculará el logaritmo en base 10 de los valores en el eje y y se generará otra gráfica.

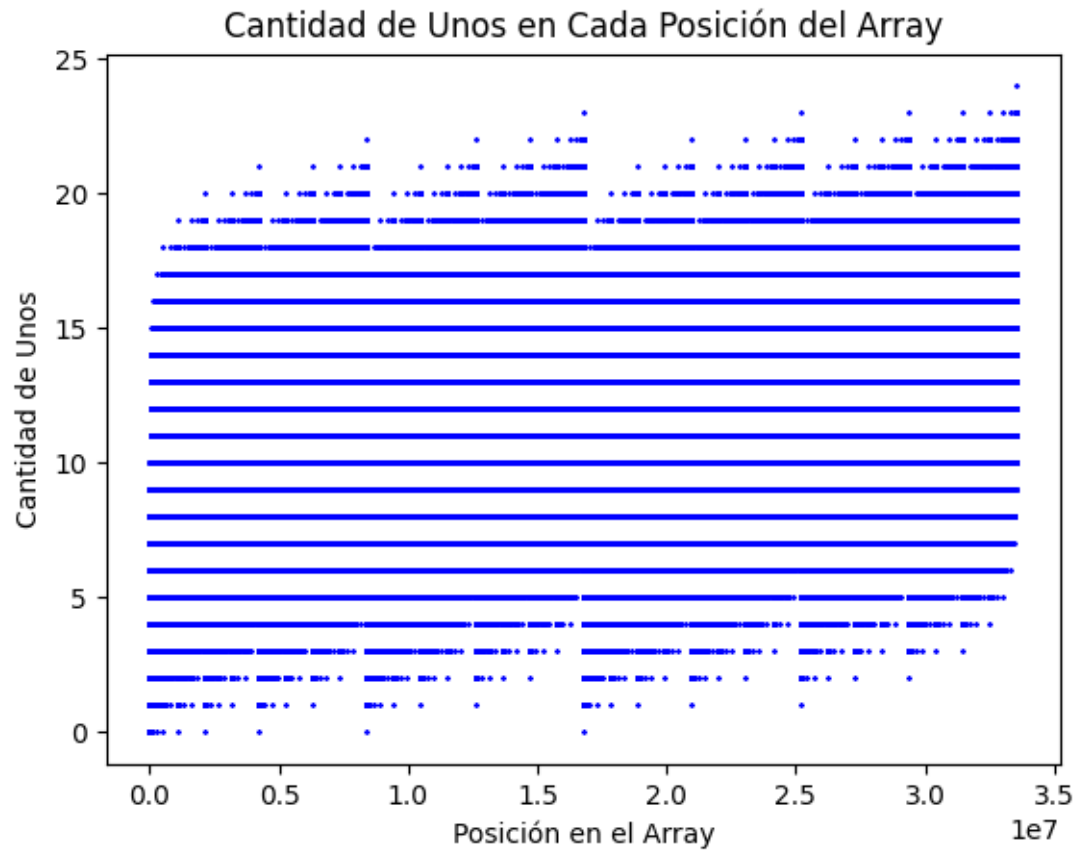


Figure 4: Numero de 1's

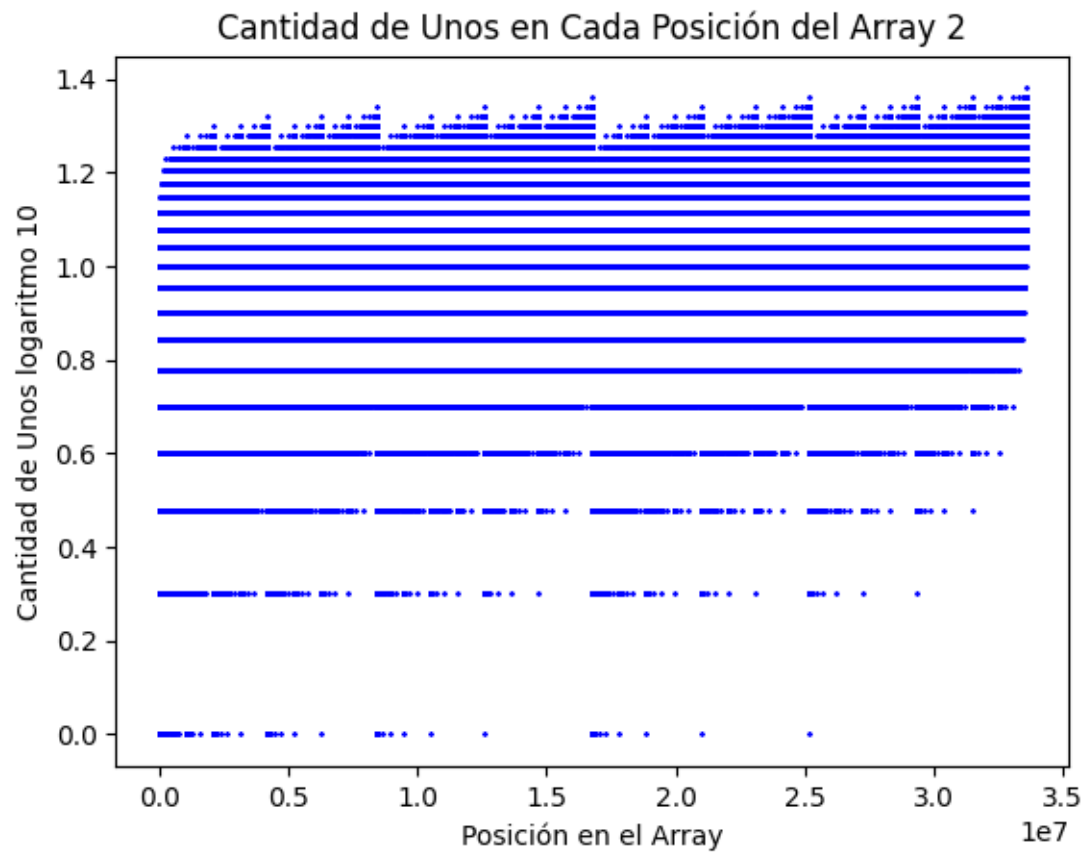


Figure 5: Numero de 1's con log10

Código

```

1 import random
2 import matplotlib.pyplot as plt
3 import math
4
5 k = 0
6 opcion = 0
7 path = "binarios.txt"
8
9
10 def generar_secuencia_binaria(k):
11     if k == 0:
12         return [" "]
13     elif k == 1:

```

```

14         return [" ", "0", "1"]
15     else:
16         nueva_secuencia = [" ", "0", "1"]
17         k_aux = 2
18         k_pos = 0
19
20         while k_aux <= k:
21             for i in range(k_pos + 1, pow(2, k_aux), 1):
22                 nueva_secuencia.append(nueva_secuencia[i] +
23 "0")
24                 nueva_secuencia.append(nueva_secuencia[i] +
25 "1")
26
27                 k_pos = pow(2, k_aux) - 1
28                 k_aux = k_aux + 1
29
30         del nueva_secuencia[-2:]
31         return nueva_secuencia
32
33 def contar_unos(cadena):
34     contador = 0 # Inicializamos el contador
35
36     for caracter in cadena:
37         if caracter == '1':
38             contador += 1 # Incrementamos el contador si
39             encontramos un '1'
40
41     return contador
42
43 print("BIENVENIDO".center(50))
44 print("\n1) ingrese k de forma manual")
45 print("2) ingrese k de forma automatica\n")
46
47 while True:
48     try:
49         opcion = int(input("Seleccione la opci n a realizar
50 : "))
51         if opcion == 1:
52             print("Seleccionaste la opci n manual")
53             k = int(input("Ingrese la k: "))
54             break
55         elif opcion == 2:
56             k = random.randint(0, 1000)
57             print("Seleccionaste la opci n random")
58             break
59         else:
60             print("Opci n no v lida. Por favor, elige 1 o
61 2.")
62     except ValueError:
63         print("Entrada no v lida. Por favor, ingresa un

```

```

    n mero entero.")
59
60 opcion = None
61 print(f"El valor de k -> {k}")
62 input("Presione enter para continuar...")
63
64 # Convertir secuencia_binaria en una lista
65 secuencia_binaria = list(generar_secuencia_binaria(k))
66
67 posiciones = []
68 cantidad_unos = []
69
70 with open(path, "w", encoding="utf-8") as file:
71     for item, elemento in enumerate(secuencia_binaria):
72         posiciones.append(item)
73         cantidad_unos.append(contar_unos(elemento))
74         file.write(f"{elemento}\n")
75
76 file.close()
77
78 plt.scatter(posiciones, cantidad_unos, marker='*', s=1,
79             color='blue')
79 plt.xlabel('Posici n en el Array')
80 plt.ylabel('Cantidad de Unos')
81 plt.title('Cantidad de Unos en Cada Posici n del Array')
82
83 plt.show()
84
85
86 posiciones = []
87 cantidad_unos_log = []
88
89 for i in range(len(cantidad_unos)):
90     if cantidad_unos[i] != 0:
91         posiciones.append(i)
92         cantidad_unos_log.append(math.log10(cantidad_unos[i]
93     ))
94
95 plt.scatter(posiciones, cantidad_unos_log, marker='*', s=1,
96             color='blue')
97 plt.xlabel('Posici n en el Array')
98 plt.ylabel('Cantidad de Unos logaritmo 10')
99 plt.title('Cantidad de Unos en Cada Posici n del Array 2')
100
101 plt.show()

```

Listing 1: Practica 2

4 Conclusión

En esta práctica, hemos explorado el mundo de las cadenas binarias (Σ^n) en acción, aplicando conceptos clave de la teoría de la computación. Hemos aprendido que estas cadenas son mucho más que secuencias de 0s y 1s; son una ventana hacia un vasto universo de combinaciones posibles.

Mediante la programación, hemos demostrado que las cadenas binarias pueden ser estudiadas, visualizadas y comprendidas en profundidad. Hemos comprobado cómo estas cadenas se comportan en función de su longitud (n), y cómo la graficación, incluso con logaritmo base 10, puede revelar patrones y relaciones interesantes.

5 Bibliografía

Ullman, J.D. (2009-10). "CS154: Introduction to Automata and Complexity Theory". Sitio web: <http://infolab.stanford.edu/~ullman/ialc/spr10/spr10.html>LECTURE