INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Computo

Práctica 1

Teoría de la Computación

Autor:
Hernández Vergara Eduardo

Marzo 2022

Índice

1. Introdución

1.1. Objetivo

En la práctica desarrollada tuvimos que hacer uso de conceptos fundamentales en la teoía de autómatas los cuales son el alfabeto, las cadenas de caracteres y el lenguaje, pues en este programa tuvimos que construir el universo de cadenas binarias.

1.2. Alfabeto

Un alfabeto es un conjunto de símbolos fínito y no vacío. Convencionalmente, utilizamos en simbolo Σ para designar un alfabeto. El alfabeto que nos intera para nuestra practica es el alfabeto binario, $\Sigma = \{1, 0\}$

1.3. Cadenas de caracteres

Una cadena de caracteres es una secuencia fínita de símbolos seleccionados de algún alfabeto. Por ejemplo, 01101 es una cadena del alfabeto binario $\Sigma = \{0,1\}$. La cadena 111 es otra cadena de dicho alfabeto.

1.3.1. Potencias de un alfabeto

Si Σ es un alfabeto, podemos expresar el conjunto de todas las cadenas de una determinada longitud de dicho alfabeto utilizando una notación exponencial. Definimos Σ^k para que sea el conjunto de las cadenas de longitud k, tales que cada uno de los símbolos de las mismas pertenece a Σ.

Por convenio, el conjunto de todas las cadenas de un alfabeto Σ se designa mediante Σ^*. Por ejemplo, $\{0,1\}^* = \{\epsilon$,0,1,00,01,10,11,000, . . .$\}$. Expresado de otra forma, $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup$...

2. Solución

Para la implementación de la solución se uso C#, con .Net 6.0(para las gráficas se uso .Net 4.8), con el IDE Visual Studio 2022, como veremos a continuación con las capturas de resultado, veremos como salio.

2.1. Escritura de Datos

Figura 1: Escritura de datos

Como podemos ver, en esta corrida de programa, se tarda 7 minutos en escribir mi archivo, a costa de memoria RAM, ahora bien, el archivo de salida no se puede abrir ni con VIM por su gran tamaño, entonces no puedo abrirlo, pero, confirmamos su contenido con las gráficas. Como dije anteriormente el tamaño del archivo es muy grande, para ser abierto con VIM

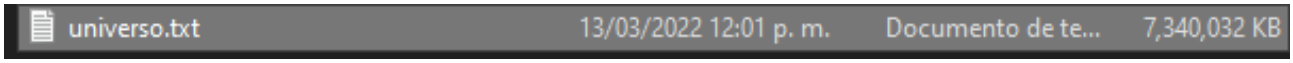

Figura 2: El tamaño del archivo

2.2. Lectura de Datos

Como podemos apreciar, una vez escritos los datos, toca leerlos y graficarlos, una vez hecho esto nos despliega la grafica correspondiente

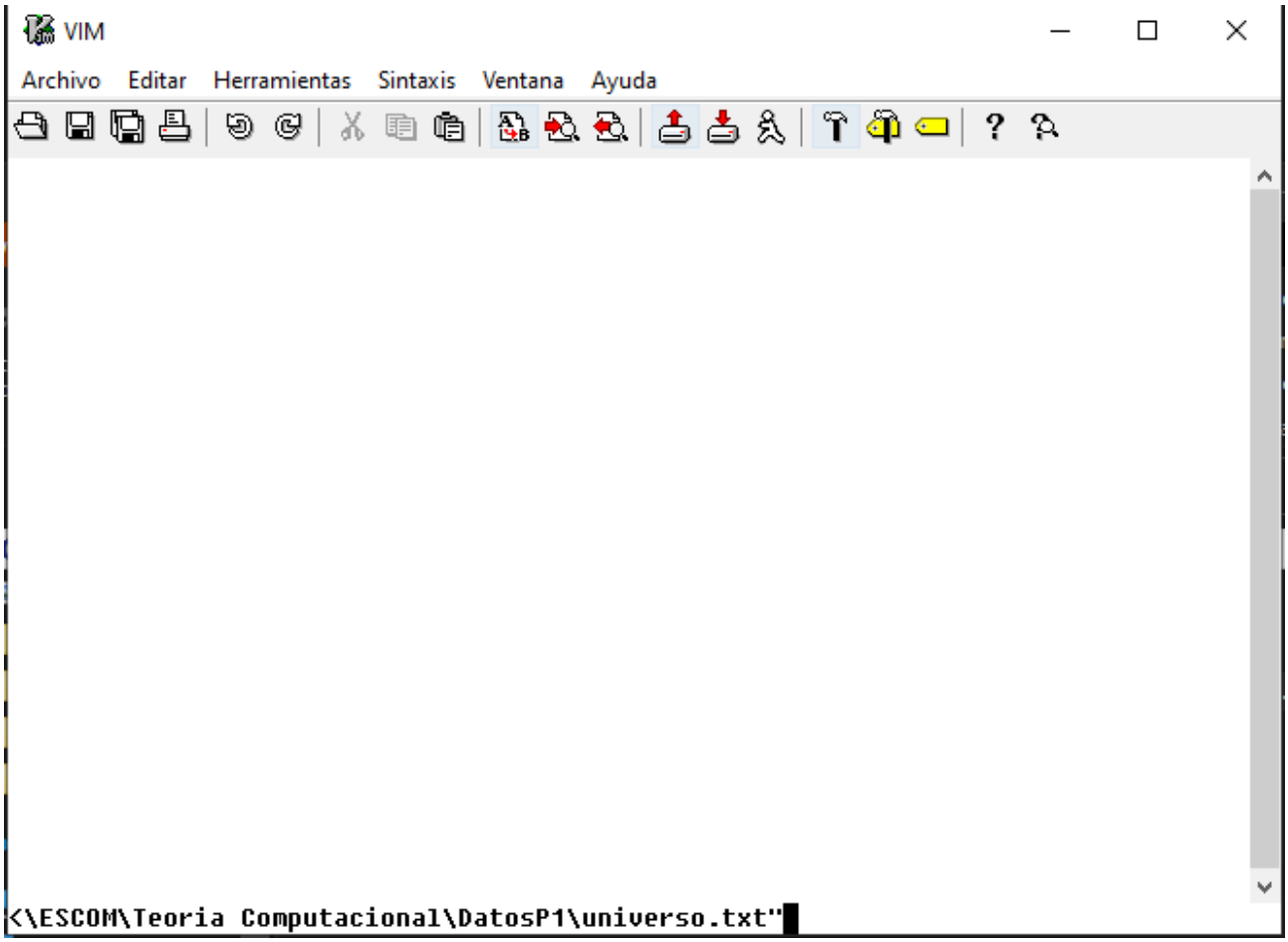

Figura 3: El archivo no puede ser abierto por VIM

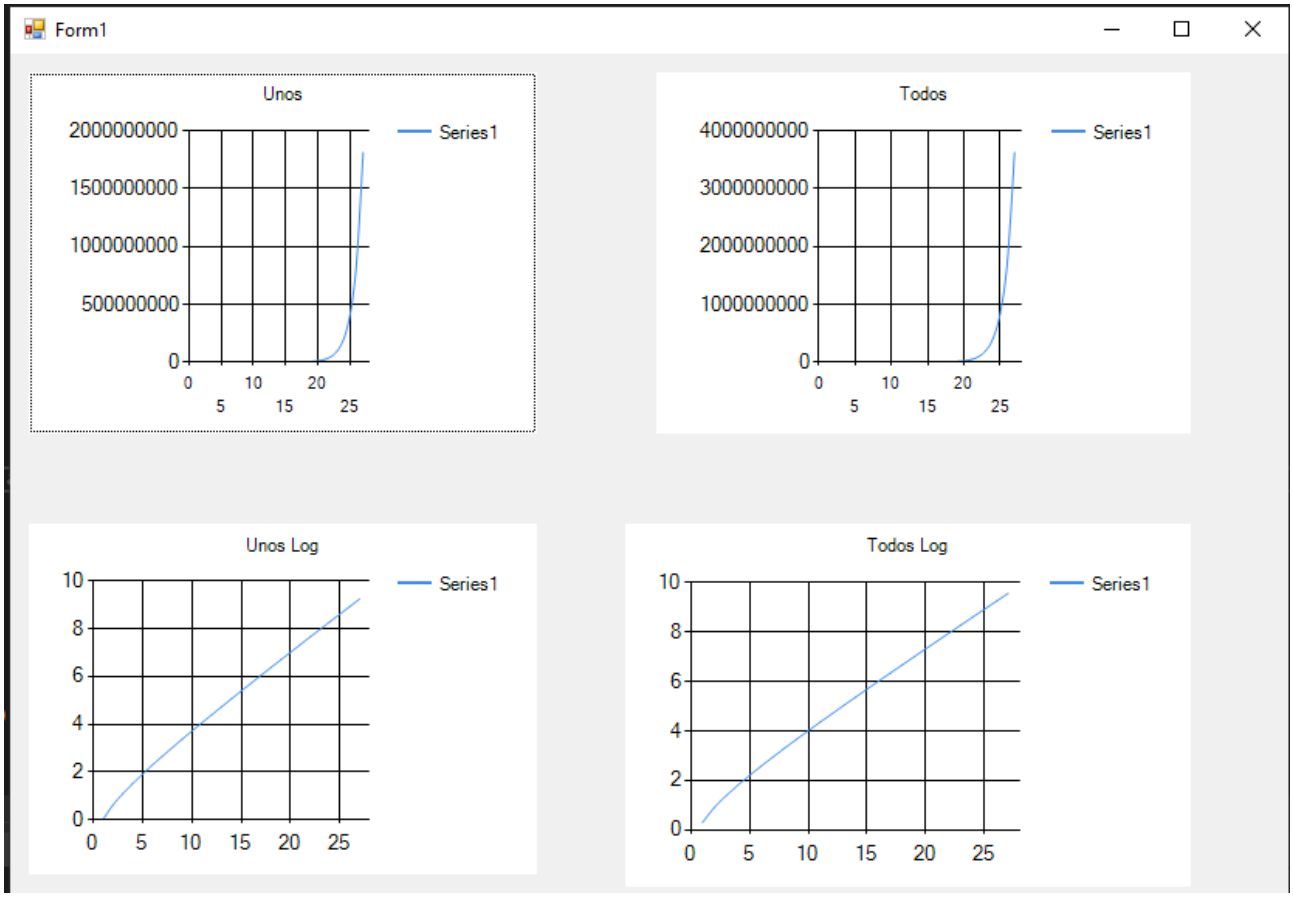

Figura 4: Las graficas para n = 27

3. Código

3.1. Escritura de Datos

Listing 1: Escritura de Datos C# .net 6.0

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
namespace Programa1
{
    public class Programa1
    {
        public static void Main(string[] args)
        {
            Menu();
        }
        public static void Menu()
        {
            int num = 0;
            const string Menus = "1. Escoger Numero:\n2. Al azar:\n3.Salir.";
            Console.WriteLine(">>Menu<<");
            Console.WriteLine("Eliga una opcion");
            Console.WriteLine(Menus);
            int opc = Convert.ToInt32(Console.ReadLine());
            switch (opc)
            {
                case 1:
                    Console.WriteLine("Ingresa k: ");
                    num = Convert.ToInt32(Console.ReadLine() ?? "0");
                    Procesos(num);
                    break;
                case 2:
                    Random rnd = new Random();
                    num = rnd.Next(2, 30);
                    Console.WriteLine($"Se escogio un numero al azar: {num}");
                    Procesos(num);
                    break;
                case 3:
                    Console.WriteLine("Adios");
                    break;
                default:
                    Menu();
                    break;
            }
        }
        public static void Procesos(int num)
        {
            Stopwatch stopwatch = new Stopwatch();
            stopwatch.Start();
            string alfabeto = "01";
            List<string> cadenas = new();
            foreach (char c in alfabeto)
                cadenas.Add(c.ToString());
            string header = $"Escogio \u03A3^{num} \n siendo esto: \u03A3^* = ( ";
```

```
51              bool siono = File.Exists("D:\\Documentos\\ESCOM\\Teoria
                    Computacional\\DatosP1\\universo.txt");
52              if (siono == true)
53              {
54                  File.Delete("D:\\Documentos\\ESCOM\\Teoria Computacional\\
                        DatosP1\\universo.txt");
55              }
56              File.AppendAllLines("D:\\Documentos\\ESCOM\\Teoria
                    Computacional\\DatosP1\\universo.txt", cadenas);
57              for (int i = 1; i < num; i++)
58              {
59                  List<string>[] auxiliar = new List<string>[alfabeto.Length
                        ];
60                  for (int j = 0; j < auxiliar.Length; j++)
61                  {
62                      auxiliar[j] = new();
63                      auxiliar[j].AddRange(cadenas);
64                      for (int k = 0; k < auxiliar[j].Count; k++)
65                          auxiliar[j][k] = string.Concat(alfabeto[j],
                                auxiliar[j][k]);
66                  }
67                  cadenas = new();
68                  foreach (List<string> lista in auxiliar)
69                      cadenas.AddRange(lista);
70                  File.AppendAllLines("D:\\Documentos\\ESCOM\\Teoria
                        Computacional\\DatosP1\\universo.txt", cadenas);
71              }
72              Console.WriteLine("Exito, se ha guardado en universo.txt");
73              cadenas.Clear();
74              cadenas = new();
75              GC.Collect(GC.MaxGeneration);
76              GC.Collect(GC.GetGeneration(cadenas));
77              GC.Collect();
78              stopwatch.Stop();
79              Console.WriteLine("Se tardo en escribir: {0}", stopwatch.
                    Elapsed.ToString("hh\\:mm\\:ss\\.fff"));
80              Console.Beep(200, 5000);
81              Menu();
82          }
83      }
84  }
```

3.2. Grafica/Lectura Datos

Listing 2: Grafica/Lectura Datos C# (.net 4.8)

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Diagnostics;
6  using System.Drawing;
7  using System.IO;
8  using System.Linq;
9  using System.Text;
10 using System.Threading.Tasks;
11 using System.Windows.Forms;
12 using System.Windows.Forms.DataVisualization.Charting;
```

```
13
14 namespace GraficoP1
15 {
16     public partial class Form1 : Form
17     {
18         static List<UInt64> cuentas;
19         static List<UInt64> cusas;
20         static List<Double> cuentasLog;
21         static List<Double> cusasLog;
22         public Form1()
23         {
24             InitializeComponent();
25         }
26         public static void leer()
27         {
28             Stopwatch s2 = new Stopwatch();
29             s2.Start();
30             StreamReader lectura = new StreamReader("D:\\Documentos\\ESCOM
                   \\Teoria Computacional\\DatosP1\\universo.txt");
31             string linea;
32             int conjuto = 1, cuenta = 0;
33             Int64 usass = 0;
34             double cuentaLog = 0, usassLog = 0;
35             cuentas = new List<UInt64>();
36             cusas = new List<UInt64>();
37             cuentasLog = new List<Double>();
38             cusasLog = new List<Double>();
39             while ((linea = lectura.ReadLine()) != null)
40             {
41                 if (linea.Length > conjuto)
42                 {
43                     cusas.Add(Convert.ToUInt64(usass));
44                     cuentas.Add(Convert.ToUInt64(cuenta));
45                     cuenta = 0;
46                     usass = 0;
47                     conjuto++;
48                 }
49                 usass += linea.Count();
50                 cuenta += linea.Count(caracter => caracter.Equals('1'));
51                 //Console.WriteLine(linea);
52                 //Console.WriteLine(usass);
53             }
54             cusas.Add(Convert.ToUInt64(usass));
55             cuentas.Add(Convert.ToUInt64(cuenta));
56             for (int i = 0; i < cuentas.Count(); i++)
57             {
58                 cuentaLog = Math.Log(Convert.ToDouble(cuentas[i]), 10);
59                 cuentasLog.Add(cuentaLog);
60                 usassLog = Math.Log(Convert.ToDouble(cusas[i]), 10);
61                 cusasLog.Add(Convert.ToDouble(usassLog));
62                 Console.WriteLine(cuentaLog);
63             }
64             s2.Stop();
65             Console.WriteLine("Se tardo en leer: {0}", s2.Elapsed.ToString
                   ("hh\\:mm\\:ss\\.fff"));
66             Console.Beep(300, 1000);
67             lectura.Close();
68         }
69         private void Form1_Load(object sender, EventArgs e)
70         {
```

```
71              //leo el archivo
72              leer();
73              //saco el eje X
74              List<string> series = new List<string>();
75              //cambiar la composicion de colores
76              chart1.Palette = ChartColorPalette.BrightPastel;
77              chart2.Palette = ChartColorPalette.BrightPastel;
78              chart3.Palette = ChartColorPalette.BrightPastel;
79              chart4.Palette = ChartColorPalette.BrightPastel;
80              chart1.Titles.Add("Unos");
81              chart2.Titles.Add("Todos");
82              chart3.Titles.Add("Unos Log");
83              chart4.Titles.Add("Todos Log");
84              for (int i = 0; i < cuentas.Count(); i++)
85              {
86                  series.Add(Convert.ToString(i + 1));
87                  chart1.Series["Series1"].Points.AddXY(Convert.ToInt32(
                        series[i]) , cuentas[i]);
88                  chart2.Series["Series1"].Points.AddXY(Convert.ToInt32(
                        series[i]), cusas[i]);
89                  chart3.Series["Series1"].Points.AddXY(Convert.ToInt32(
                        series[i]), cuentasLog[i]);
90                  chart4.Series["Series1"].Points.AddXY(Convert.ToInt32(
                        series[i]), cusasLog[i]);
91              }
92              Console.WriteLine(cuentas.Count());
93          }
94          private void chart1_Click(object sender, EventArgs e)
95          {
96
97          }
98
99          private void chart2_Click(object sender, EventArgs e)
100         {
101
102         }
103
104         private void chart3_Click(object sender, EventArgs e)
105         {
106
107         }
108
109         private void chart4_Click(object sender, EventArgs e)
110         {
111
112         }
113
114
115     }
116 }
```