INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Computo

Práctica 6: PushDown Automata

Teoría de la Computación

Autor:
Hernández Vergara Eduardo

Junio 2022

Índice

1. Introdución

1.1. Objetivo

En la práctica desarrollada tuvimos que diseñar un autómata de pila que nos evalue una cantidad n y m de 0's y 1's donde m = n.

1.2. Automatas de pila (Pushdown automatas)

Fundamentalmente, el autómata a pila es un autómata finito no determinista con transiciones-ϵ y una capacidad adicional: una pila en la que se puede almacenar una cadena de >símbolos de pila<. La presencia de una pila significa que, a diferencia del autómata finito, el autómata a pila puede >recordar< una cantidad infinita de información. Sin embargo, a diferencia de las computadoras de propósito general, que también tienen la capacidad de recordar una cantidad arbitrariamente grande de información, el autómata a pila sólo puede acceder a la información disponible en su pila de acuerdo con la forma de manipular una pila FIFO (first-in-first-out way, primero en entrar primero en salir).

1.3. Descripcion Instantanea

Hasta el momento, sólo disponemos de una noción informal de cómo calcula un autómata a pila. Intuitivamente, el autómata a pila pasa de una configuración a otra, en respuesta a los símbolos de entrada (o, en ocasiones, a ϵ), pero a diferencia del autómata finito, donde el estado es lo único que necesitamos conocer acerca del mismo, la configuración del autómata a pila incluye tanto el estado como el contenido de la pila. Siendo arbitrariamente larga, la pila es a menudo la parte más importante de la configuración total del autómata a pila en cualquier instante.

2. Solución

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

2.1. Correr el Programa

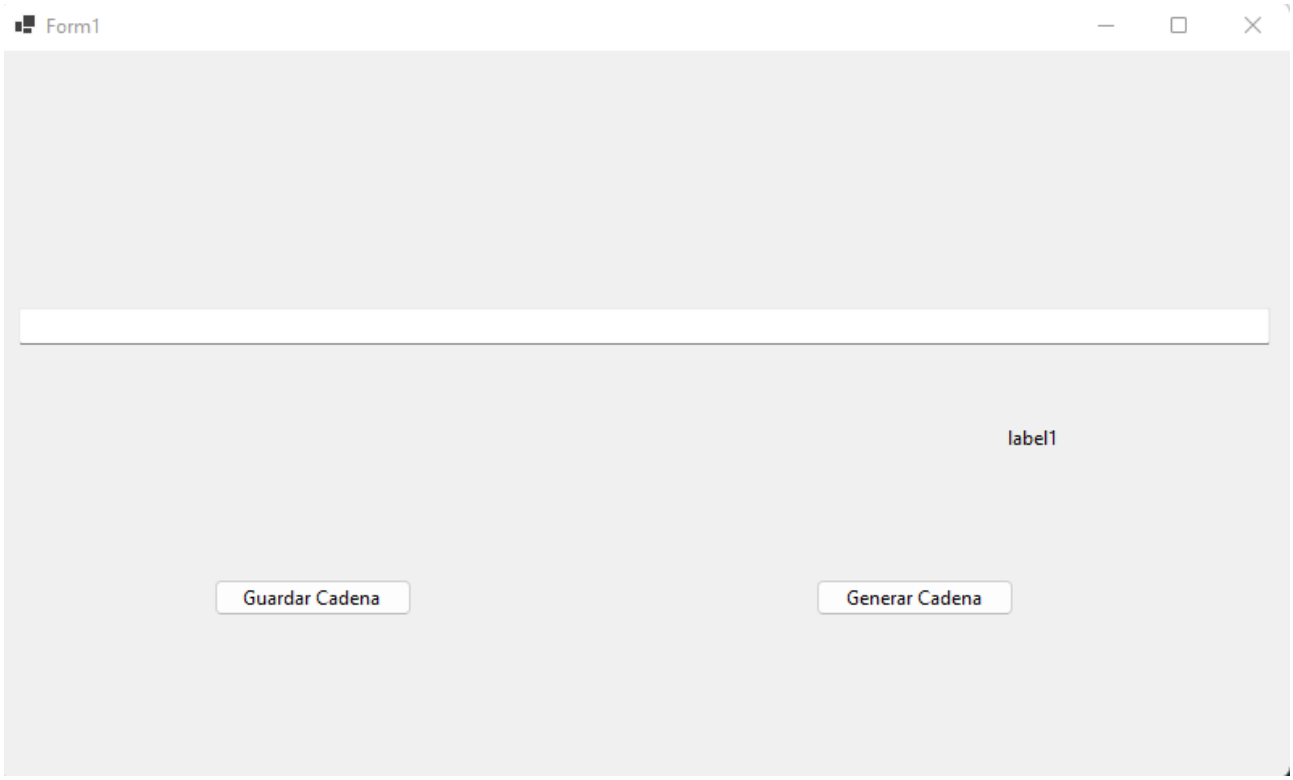

Figura 1: Es el menu principal

Primeramente tenemos nuestro menu principal, donde tenemos un textbox y 2 botones, el boton generar cadena es para generarlo automaticamente, mientras que el guardar cadena es para ingresar la cadena nosotros. Al darle clic nos llevara a la graficacion del automata Aqui podemos ver la animación del automata, claro que no podemos apreciar como cambia de estado por ser una imagen, sin embargo se verá en la revision del programa

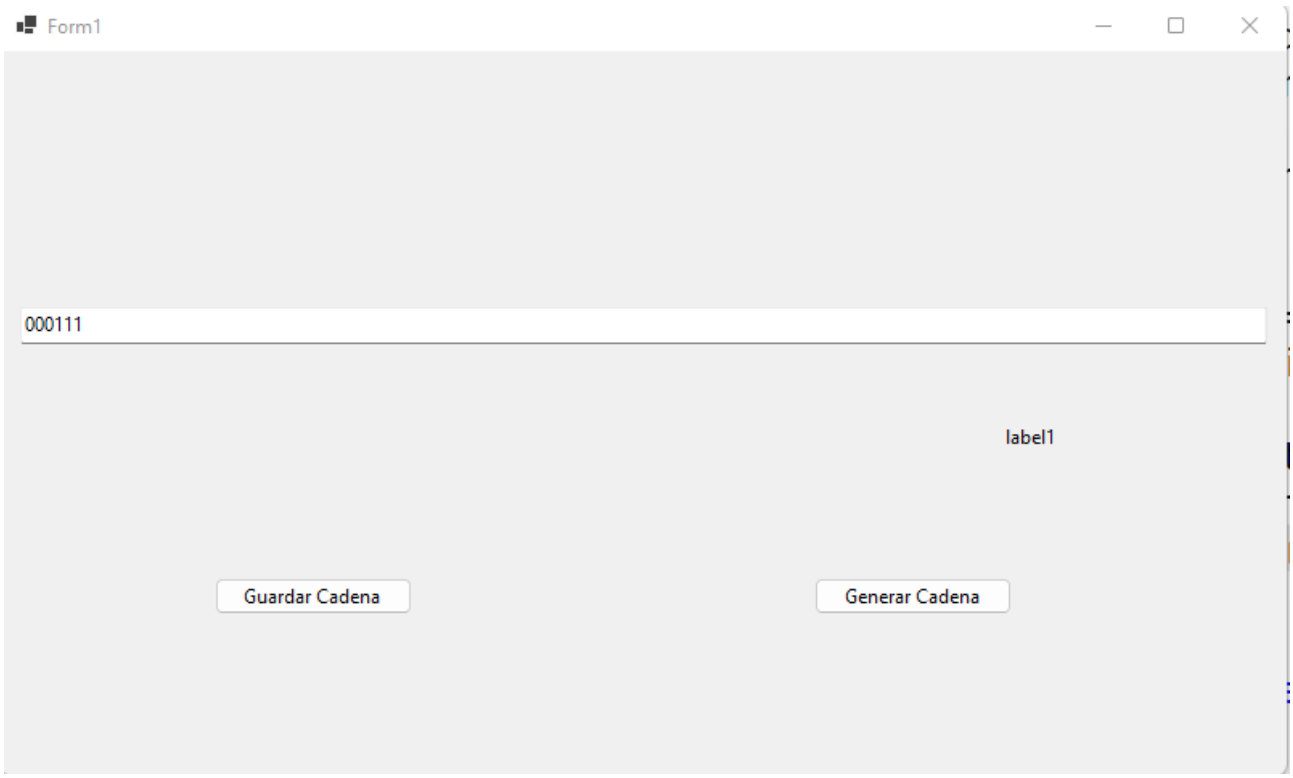

Figura 2: Le ingresamos una cadena a nuestro menu

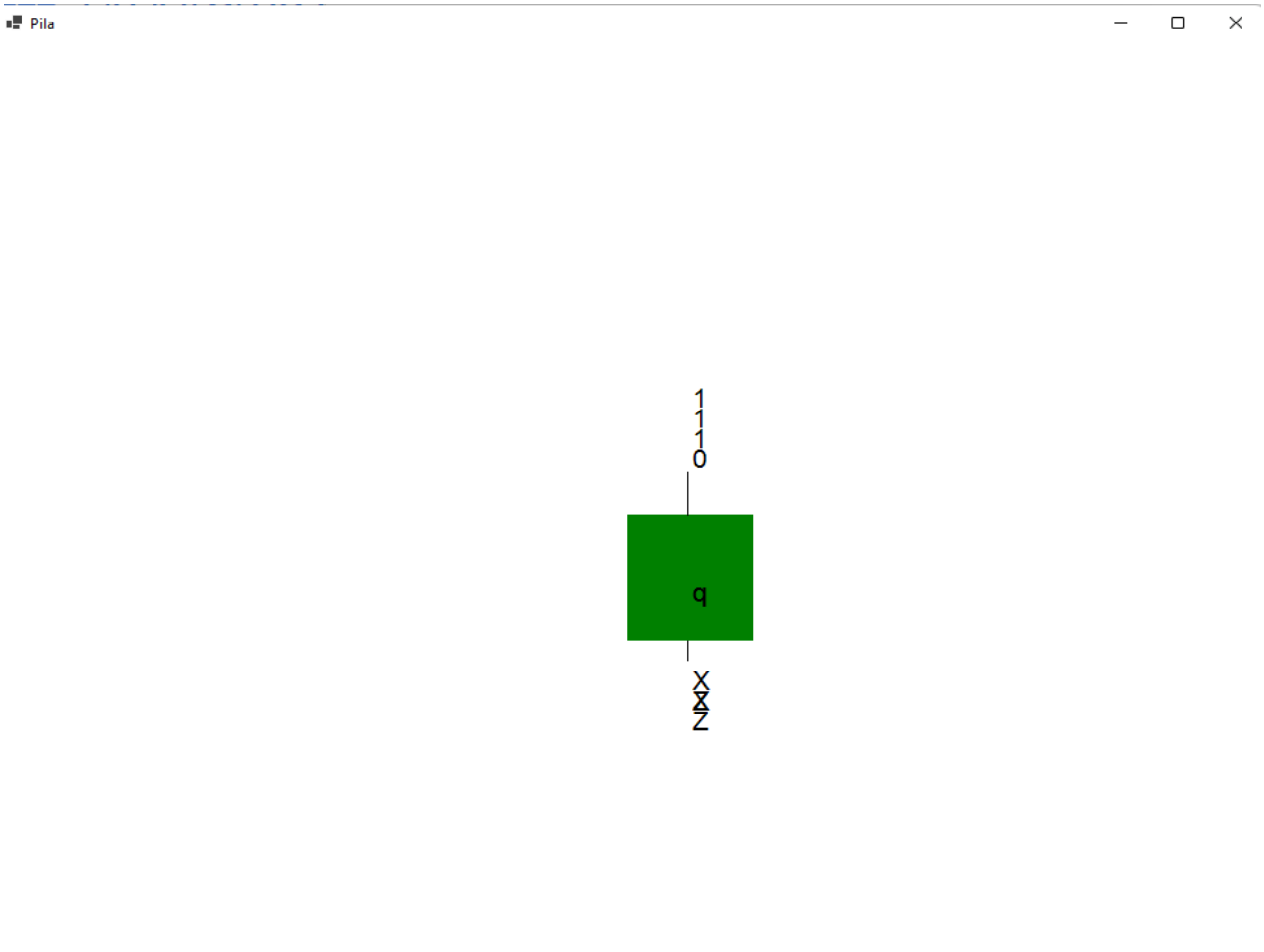

Figura 3: Animacion Automata

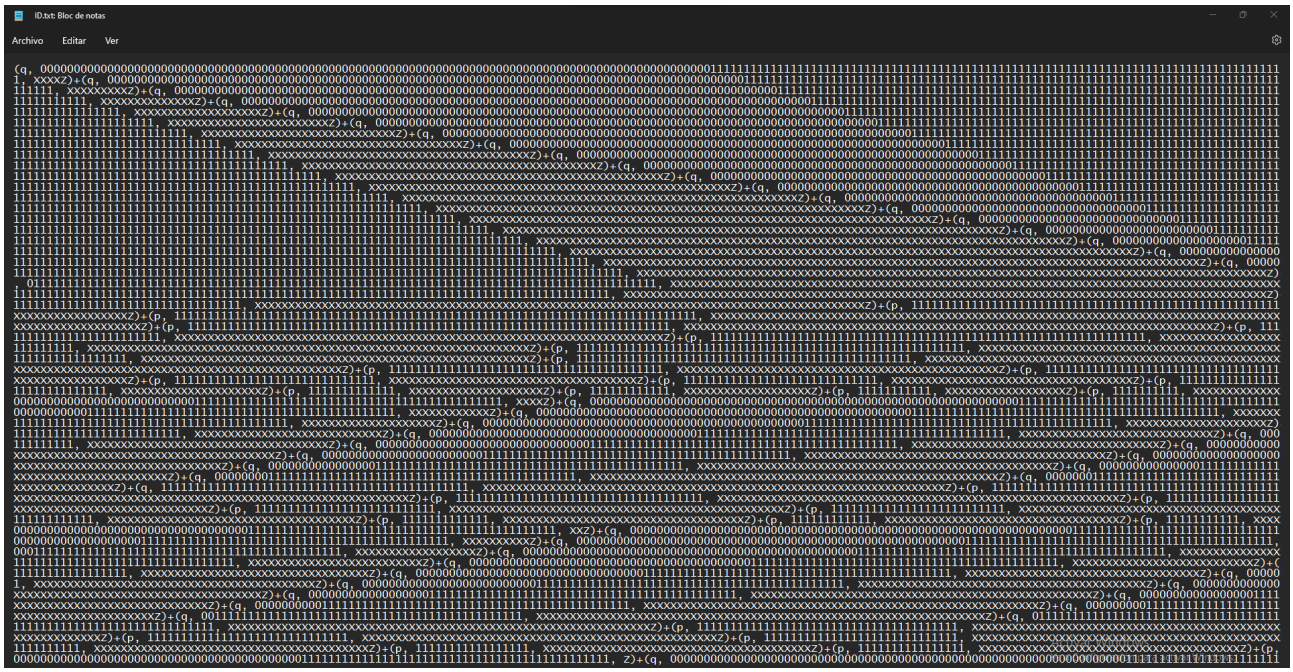

Figura 4: Descripcion Instantanea de mi automata

3. Código

3.1. El automata

Listing 1: Pantalla Principal del PDA C# .net 6.0

```
namespace Programa6
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Generar_Click(object sender, EventArgs e)
        {
            Utilidades.Utilidad utilidad = new();
            string aleatorio;
            aleatorio = utilidad.GenerarCadenas();
            Cant.Text = aleatorio.Length.ToString();
            textBox1.Text = aleatorio;
            Utilidades.Datos.cadena = aleatorio;
            App.Pila pila = new();
            Thread.Sleep(1000);
            pila.ShowDialog();
        }

        private void GuardarCad_Click(object sender, EventArgs e)
        {
            Utilidades.Datos.cadena = textBox1.Text;
            App.Pila pila = new();
            pila.ShowDialog();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Listing 2: Pila C# .net 6.0

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa6.App
{
    public partial class Pila : Form
    {
```

```
16          public Pila()
17          {
18              InitializeComponent();
19          }
20
21          private void Pila_Paint(object sender, PaintEventArgs e)
22          {
23              Graphics g = e.Graphics;
24              SolidBrush escr = new(Color.Black);
25              Pen pen = new(escr);
26              Font f = new("Arial", 16);
27              StringFormat fmt = new();
28              SolidBrush verde = new(Color.Green);
29              //g.DrawString(Utilidades.Datos.cadena, f, escr, 15, 15);
30              Rectangulo(g, verde);
31              var cP = Utilidades.Datos.cadena;
32              Stack pila = new();
33              Stack p2 = new();
34              String ruta = "D:\\Documentos\\ESCOM\\Teoria Computacional\\
                    DatosP6";
35              for (int i = cP.Length; i > 0; i--)
36              {
37                  p2.Push(cP[i - 1]);
38              }
39              AutomataPila(p2, f, escr, g);
40              string estado = "q";
41              string cadep = "", cadep2 = "";
42              foreach (var c in p2)
43              {
44                  cadep += c;
45              }
46              foreach (var c in pila)
47              {
48                  cadep2 += c;
49              }
50              File.AppendAllText($"{ruta}\\ID.txt", $"({estado}, {cadep}, {
                    cadep2}Z)+");
51              List<String> historia = new();
52              if (cP[0] == '0')
53              {
54                  for (int i = 0; i < cP.Length; i++)
55                  {
56                      try
57                      {
58                          if (cP[i] == '0')
59                          {
60                              estado = "q";
61                              g.DrawString(estado, f, escr, 550, 424);
62                              pila.Push("X");
63                              p2.Pop();
64                              string cade1 = "", cade2 = "";
65                              foreach (var c in p2) {
66                                  cade1 += c;
67                              }
68                              foreach (var c in pila)
69                              {
70                                  cade2 += c;
71                              }
72                              File.AppendAllText($"{ruta}\\ID.txt", $"({
                                    estado}, {cade1}, {cade2}Z)+");
```

```
                            BorrarX(g);
                            EscribirX(pila, f, escr, g);
                            Borrar(g);
                            AutomataPila(p2, f, escr, g);
                            historia.Add(estado);
                        }
                        else if (cP[i] == '1')
                        {
                            Rectangulo(g, verde);
                            pila.Pop();
                            BorrarX(g);
                            EscribirX(pila, f, escr, g);
                            estado = "p";
                            g.DrawString(estado, f, escr, 550, 424);
                            p2.Pop();
                            string cade1 = "", cade2="";
                            foreach (var c in p2)
                            {
                                cade1 += c;
                            }
                            foreach (var c in pila)
                            {
                                cade2 += c;
                            }
                            File.AppendAllText($"{ruta}\\ID.txt", $"({
                                estado}, {cade1}, {cade2}Z)+");
                            Borrar(g);
                            AutomataPila(p2, f, escr, g);
                            historia.Add(estado);
                        }
                    }
                    catch (SystemException ex)
                    {
                        MessageBox.Show("Rechazada");
                        break;
                    }

                }
            }


            else
            {
                MessageBox.Show("Rechazado");
            }
            estado = "f";
            string cadef1 = "", cadef2 = "";
            foreach (var c in p2)
            {
                cadef1 += c;
            }
            foreach (var c in pila)
            {
                cadef2 += c;
            }
            File.AppendAllText($"{ruta}\\ID.txt", $"({estado}, {cadef1}, {
                cadef2}Z)+");
            Rectangulo(g, verde);
            g.DrawString(estado, f, escr, 550, 424);
        }
```

```
131         public void AutomataPila(Stack p2, Font f, SolidBrush escr,
                Graphics g ) {
132             int j = 318;
133             foreach (var c in p2)
134             {
135                 g.DrawString(c.ToString(), f, escr, 550, j);
136                 if (Utilidades.Datos.cadena.Length <= 10)
137                 {
138                     Thread.Sleep(1000);
139                 }
140                 j -= 16;
141             }
142         }
143         public void Borrar(Graphics g) {
144             SolidBrush borra = new(Color.White);
145             g.FillRectangle(borra, 550, 0, 16, 340);
146         }
147         public void Rectangulo(Graphics g, SolidBrush verde) {
148             g.FillRectangle(verde, 502, 374, 100, 100);
149             g.DrawLine(new Pen(Color.Black), 550, 374, 550, 315);
150             g.DrawLine(new Pen(Color.Black), 550, 474, 550, 490);
151         }
152         public void EscribirX(Stack pila, Font f, SolidBrush escr,
                Graphics g) {
153             int j = 494;
154             foreach (var c in pila)
155             {
156                 g.DrawString("Z", f, escr, 550, j+16);
157                 g.DrawString(c.ToString(), f, escr, 550, j);
158                 if (Utilidades.Datos.cadena.Length <=10) {
159                     Thread.Sleep(1000);
160                 }
161                 j += 16;
162             }
163         }
164         public void BorrarX(Graphics g)
165         {
166             SolidBrush borra = new(Color.White);
167             g.FillRectangle(borra, 550, 490, 16, 768);
168         }
169     }
170 }
```

3.2. Utilidades (Paquete C#)

Listing 3: Clase Utilidades C# .net 6.0

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Programa6.Utilidades
8  {
9      internal class Utilidad
10     {
11         public string GenerarCadenas()
```

```
12          {
13              var tam = new Random();
14              var chars = "01";
15              var arr = new char[tam.Next(1, 100000)];
16              var rnd = new Random();
17              for (int i = 0; i < arr.Length; i ++)
18              {
19                  arr[i] = chars[rnd.Next(chars.Length)];
20              }
21              var v = new String(arr);
22              return v;
23          }
24      }
25  }
```

Listing 4: Clase Datos C# .net 6.0

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Programa6.Utilidades
8  {
9      internal class Datos
10     {
11         public static string cadena;
12         public static bool nya;
13     }
14 }
```