



INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo

Ing. en sistemas computacionales



Integrantes:

Josue Macias Castillo

Irving Arturo Aguiar Hernández

Profesora:

Luz María Sánchez García

Grupo:

2CM1

Boleta:

Josue: 2015301058

Irving: 2016630418

Materia:

Teoría Computacional

Fecha de entrega:

26 de mayo del 2017

Introducción:

Un autómata con pila, autómata a pila o autómata de pila es un modelo matemático de un sistema que recibe una cadena constituida por símbolos de un alfabeto y determina si esa cadena pertenece al lenguaje que el autómata reconoce. El lenguaje que reconoce un autómata con pila pertenece al grupo de los lenguajes libres de contexto en la clasificación de la Jerarquía de Chomsky.

Funcionamiento

Los autómatas de pila, en forma similar a como se usan los autómatas finitos, también se pueden utilizar para aceptar cadenas de un lenguaje definido sobre un alfabeto A . Los autómatas de pila pueden aceptar lenguajes que no pueden aceptar los autómatas finitos. Un autómata de pila cuenta con una cinta de entrada y un mecanismo de control que puede encontrarse en uno de entre un número finito de estados. Uno de estos estados se designa como estado inicial, y además algunos estados se llaman de aceptación o finales. A diferencia de los autómatas finitos, los autómatas de pila cuentan con una memoria auxiliar llamada pila. Los símbolos (llamados símbolos de pila) pueden ser insertados o extraídos de la pila, de acuerdo con el manejo last-in-first-out (LIFO). Las transiciones entre los estados que ejecutan los autómatas de pila dependen de los símbolos de entrada y de los símbolos de la pila. El autómata acepta una cadena x si la secuencia de transiciones, comenzando en estado inicial y con pila vacía, conduce a un estado final, después de leer toda la cadena x .

Autómata con pila determinístico

Nótese que, a diferencia de un autómata finito o una máquina de Turing, la definición básica de un autómata con pila es de naturaleza no determinista, pues la clase de los autómatas con pila determinísticos, a diferencia de lo que ocurría con aquellos modelos, tiene una potencia descriptiva estrictamente menor. Para calificar a un autómata con pila como determinístico deben darse dos circunstancias; en primer lugar, por supuesto, que en la definición de cada componente de la función de transición existan un único elemento lo que da la naturaleza determinista. Pero eso no es suficiente, pues además puede darse la circunstancia de que el autómata esté en el estado s y en la pila aparezca el símbolo sZ , entonces, si existe una definición de transición posible para algún símbolo cualquiera a del alfabeto de entrada, pero, además existe otra alternativa para la palabra vacía ϵ , también esto es una forma de no determinismo, pues podemos optar entre leer un símbolo o no hacerlo. Por eso, en autómata determinístico no debe existir transición posible con lectura de símbolo si puede hacerse sin ella, ni, al contrario.

Autómata con pila no determinista

Un autómata finito con pila no determinista (AFPN) consta de los mismos parámetros de un AFPD.

$P = (Q, \Sigma, \Gamma, \Delta, q_0, T, Z)$:

Donde la función de transición Δ es de la forma:

$$\Delta: Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow \text{Pf}(Q \times \Gamma^*)$$

Donde $\text{Pf}(Q \times \Gamma^*)$ es un conjunto de subconjuntos finitos de $Q \times \Gamma^*$

Para $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ y $s \in \Gamma$

$$\Delta(q, a, s) = \{(q_1, \gamma_1), (q_2, \gamma_2), \dots, (q_n, \gamma_n)\}$$

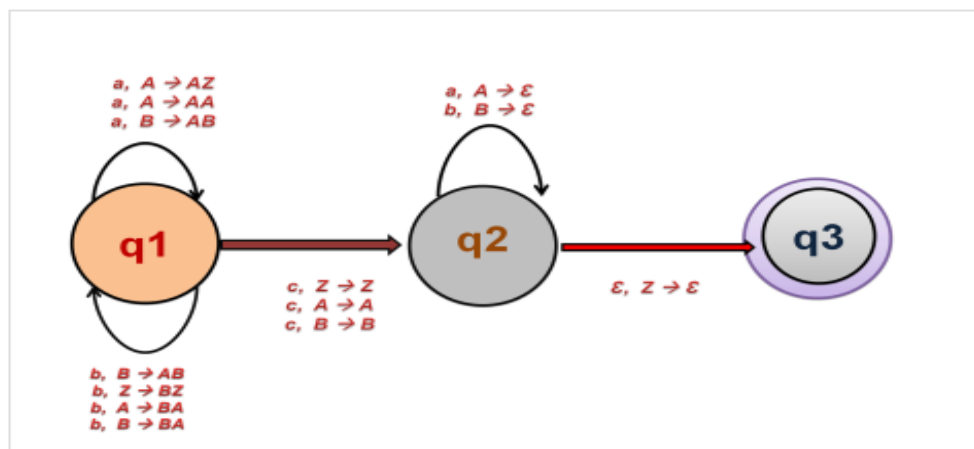
Donde $\gamma_i \in \Gamma^*$

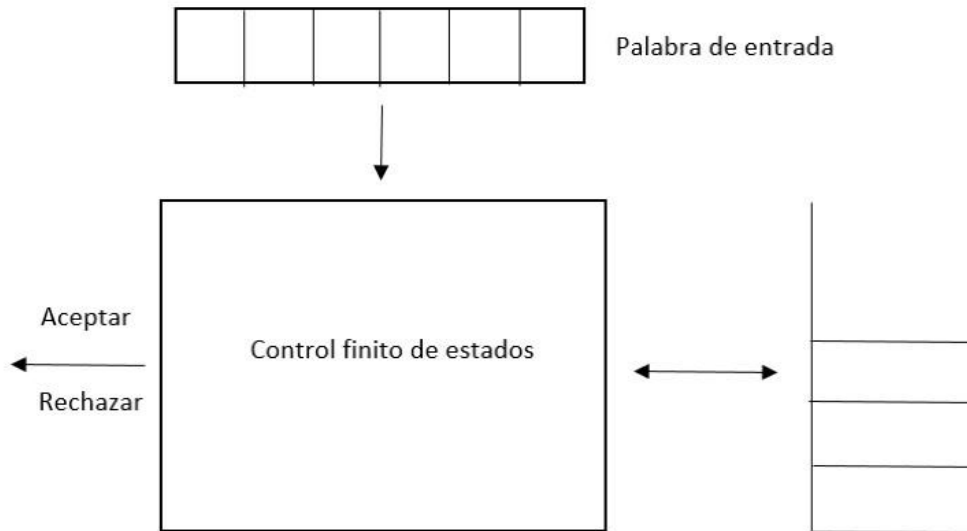
Planteamiento del problema:

Implementar un programa en C# que identifique si una cadena es válida en el autómata además de que imprima las transiciones, este autómata solo aceptaras cadenas con “a” y “b” el lenguaje será el siguiente:

$$L = \{a^n b^n | n > 0 \cup b^n a^n | n > 0\}$$

El autómata es el siguiente:





Implementación:

```
//Librerias que se utilizaran para el automata de pila
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;

namespace AUTOMATA_PILA
{
    /// Description of MainForm.

    public partial class MainForm : Form
    {
        public MainForm()
        {
            // Se llama al metodo
            InitializeComponent();

            // Se agrega el constructor
        }

        void TextBox1TextChanged(object sender, EventArgs e)
        {
        }

        void Button1Click(object sender, EventArgs e)
        {
            // Se manda un mensaje si es que no se ha escrito una cadena
        }
    }
}
```

```

        if(textBox2.Text==""){
            MessageBox.Show("NO SE INGRESO LA CADENA");
        }
        else{
            proseso();
        }
    //
    }

    public void proseso()
    {
        // Declaracion de las variables
        int ia;
        bool b1;
        b1 = false;
        ia = textBox2.TextLength - 1;
        int a;

        string s0;
        string s1;
        string s2;
        bool b;
        s1 = "q1";
        s2 = "Z";
        string z;
        textBox3.Text = s2;

        b = false;
        if (b1 == false)
        {
            ia = textBox2.TextLength;
            for (a = 0; a <= ia; a++)
            {
                textBox2.SelectionStart = 0;
                textBox2.SelectionLength = 1;
                s0 = textBox2.SelectedText;

                textBox3.SelectionStart = 0;

                textBox3.SelectionLength = 1;
                textBox3.SelectionLength = 1;

                z = textBox3.SelectedText;

                // Se manda un mensaje verificando cada caracter de la cadena
                MessageBox.Show("VERIFICANDO CADA CARACTER DE LA CADENA ");

                textBox1.Text=s1 + ", " + s0 + ", " +

                z.ToString();

                if ((s1 == "q1") && (z == "Z") && (s0 == "a") &&
                (b == false))

```

```

        {
            textBox2.SelectedText = "";
            s2 = "A" + s2;
            textBox3.Text = s2;

            b = true;

        }
        textBox3.SelectionStart = 0;

        textBox3.SelectionLength = 1;

// Se ponen los if para verificar si la cadena es
valida o no

(b == false))

        z = textBox3.SelectedText;
        if ((s1 == "q1") && (z == "A") && (s0 == "a") &&
        {
            textBox2.SelectedText = "";
            s2 = "A" + s2;
            textBox3.Text = s2;
            b = true;

        }
        textBox3.SelectionStart = 0;
        textBox3.SelectionLength = 1;
        z = textBox3.SelectedText;
        if ((s1 == "q1") && (z == "B") && (s0 == "a") &&
        {
            textBox2.SelectedText = "";
            s2 = "A" + s2;
            textBox3.Text = s2;
            b = true;

        }
        textBox3.SelectionStart = 0;
        textBox3.SelectionLength = 1;
        z = textBox3.SelectedText;
        if ((s1 == "q1") && (z == "Z") && (s0 == "b") &&
        {
            textBox2.SelectedText = "";
            s2 = "B" + s2;
            textBox3.Text = s2;
            b = true;

        }
        textBox3.SelectionStart = 0;
        textBox3.SelectionLength = 1;
        z = textBox3.SelectedText;

```

```

(b == false))
    if ((s1 == "q1") && (z == "A") && (s0 == "b") &&
    {
        textBox2.SelectedText = "";
        s2 = "B" + s2;
        textBox3.Text = s2;
        b = true;
    }
    textBox3.SelectionStart = 0;
    textBox3.SelectionLength = 1;
    z = textBox3.SelectedText;
    if ((s1 == "q1") && (z == "B") && (s0 == "b") &&
    {
        textBox2.SelectedText = "";
        s2 = "B" + s2;
        textBox3.Text = s2;
        b = true;
    }
    textBox3.SelectionStart = 0;
    textBox3.SelectionLength = 1;
    z = textBox3.SelectedText;
    if ((s1 == "q1") && (z == "Z") && (s0 == "c") &&
    {
        textBox2.SelectedText = "";
        s1 = "q2";
        b = true;
    }
    textBox3.SelectionStart = 0;
    textBox3.SelectionLength = 1;
    z = textBox3.SelectedText;
    if ((s1 == "q1") && (z == "A") && (s0 == "c") &&
    {
        textBox2.SelectedText = "";
        s1 = "q2";
        b = true;
    }
    textBox3.SelectionStart = 0;
    textBox3.SelectionLength = 1;
    z = textBox3.SelectedText;
    if ((s1 == "q1") && (z == "B") && (s0 == "c") &&
    {
        textBox2.SelectedText = "";
        s1 = "q2";
        b = true;
    }
    textBox3.SelectionStart = 0;
    textBox3.SelectionLength = 1;
    z = textBox3.SelectedText;
    if ((s1 == "q2") && (z == "A") && (s0 == "a") &&
    {
        textBox2.SelectedText = "";
        textBox3.SelectionStart = 0;

```

```

        textBox3.SelectionLength = 1;
        textBox3.SelectedText = "";
        b = true;
    }
    textBox3.SelectionStart = 0;
    textBox3.SelectionLength = 1;
    z = textBox3.SelectedText;
    if ((s1 == "q2") && (z == "B") && (s0 == "b") &&
(b == false))
    {
        textBox2.SelectedText = "";
        textBox3.SelectionStart = 0;
        textBox3.SelectionLength = 1;
        textBox3.SelectedText = "";
        b = true;
    }
    textBox3.SelectionStart = 0;
    textBox3.SelectionLength = 1;
    z = textBox3.SelectedText;
    if ((s1 == "q2") && (z == "Z") && (s0.Length ==
0))
    {
        s1 = "q3";
    }
    b = false;
}
if (s1 == "q3")
{
    //Imprime si la cadena es valida o no
    textBox3.Text="";
    MessageBox.Show("La cadena que ingreso es
valida");

}
else
{
    MessageBox.Show("La cadena que ingreso es
invalida");
}
}
}

```

```

//Codigo para agregar el menu (ComboBox)
void LimpiarToolStripMenuItemClick(object sender, EventArgs e)
{
    limpiar();
}

void limpiar(){

```



```

        textBox2.Text="";
        textBox3.Text="";

    }

    void SalirToolStripMenuItemClick(object sender, EventArgs e)
    {
        this.Close();
    }

    void IntegrantesToolStripMenuItemClick(object sender, EventArgs
e)
    {
        Form1 abrir=new Form1();
        abrir.ShowDialog();
    }

    private void listBox1_SelectedIndexChanged(object sender,
EventArgs e)
    {
    }

    //Interfaz
    private void MainForm_Load(object sender, EventArgs e)
    {
    }

    private void groupBox2_Enter(object sender, EventArgs e)
    {
    }

    private void groupBox1_Enter(object sender, EventArgs e)
    {
    }

}
}

```

Funcionamiento:

CADENAS VALIDAS:

1. Cadena : c

Practica 6

Menu

Area de trabajo

Movimiento de Transiciones

TRANSICIONES

- (q1.a,Z) -(q1,AZ)
- (q1.a,A) -(q1,AA)
- (q1.a,B) -(q1,AB)
- (q1.b,Z) -(q1,BZ)
- (q1.b,A) -(q1,BA)
- (q1.b,B) -(q1,BB)
- (q1.c,Z) -(q2,Z)
- (q1.c,A) -(q2,A)
- (q1.c,B) -(q2,b)
- (q2.a,A) -(q2,e)
- (q2.b,B) -(q2,e)
- (q2.e,Z) -(q3,e)

PILA

Practica 6

Menu

Area de trabajo

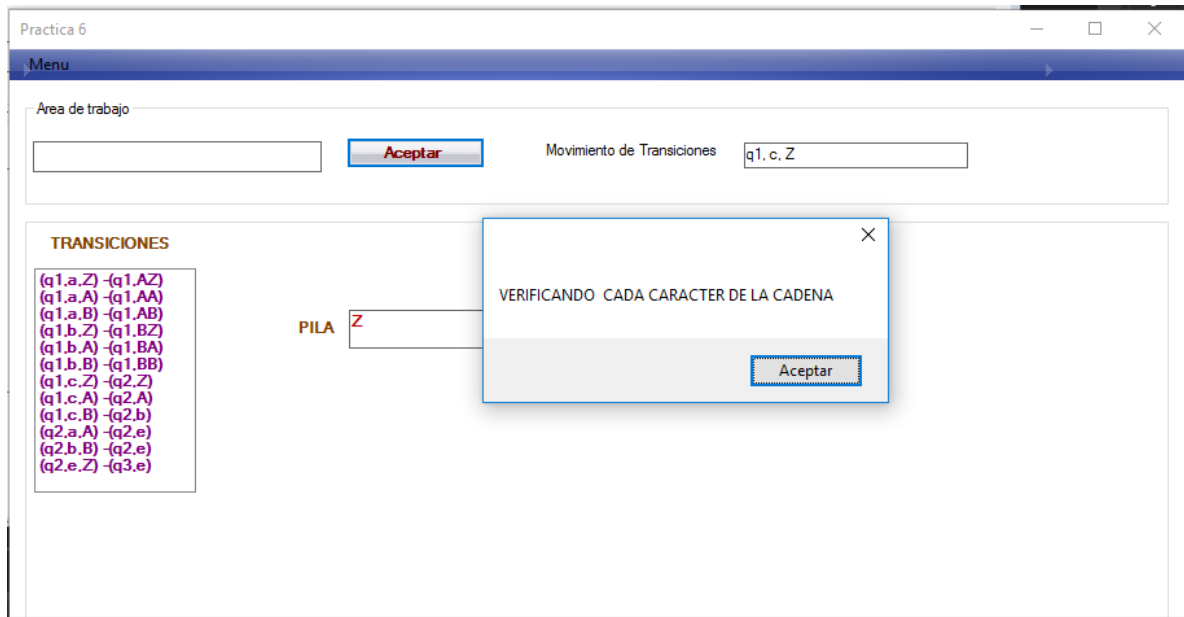
Movimiento de Transiciones

TRANSICIONES

- (q1.a,Z) -(q1,AZ)
- (q1.a,A) -(q1,AA)
- (q1.a,B) -(q1,AB)
- (q1.b,Z) -(q1,BZ)
- (q1.b,A) -(q1,BA)
- (q1.b,B) -(q1,BB)
- (q1.c,Z) -(q2,Z)
- (q1.c,A) -(q2,A)
- (q1.c,B) -(q2,b)
- (q2.a,A) -(q2,e)
- (q2.b,B) -(q2,e)
- (q2.e,Z) -(q3,e)

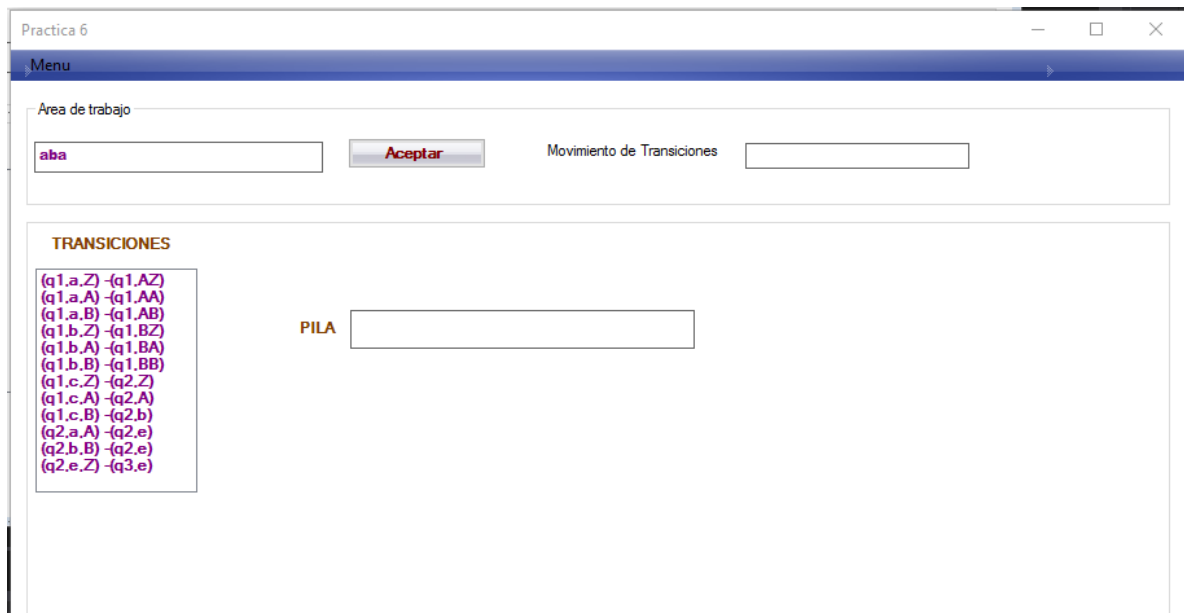
PILA

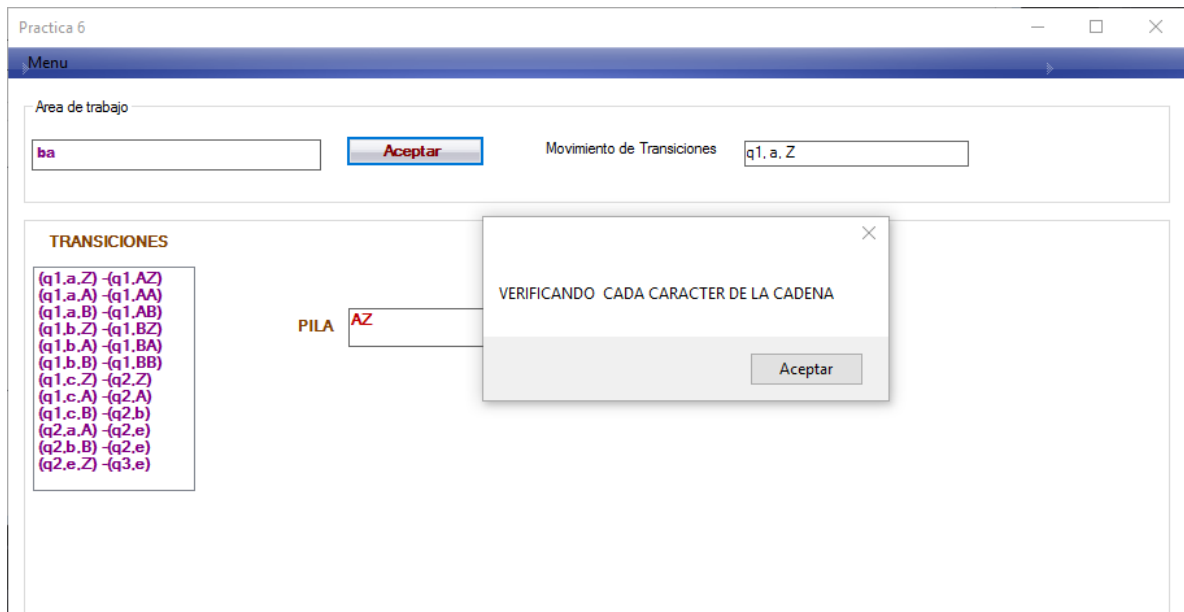
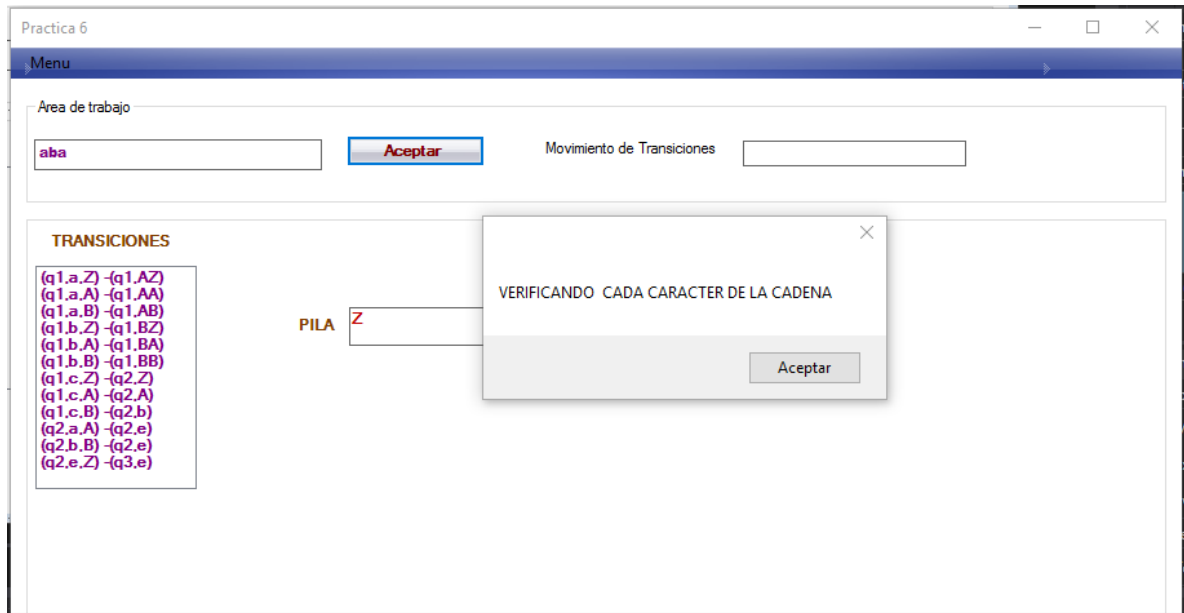
VERIFICANDO CADA CARACTER DE LA CADENA

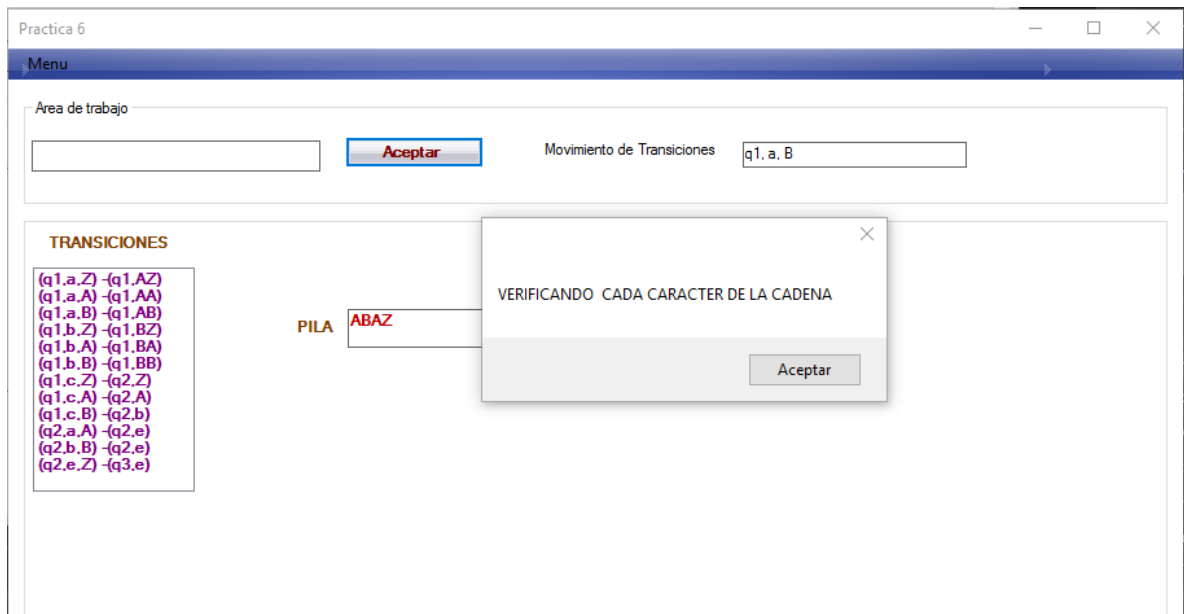
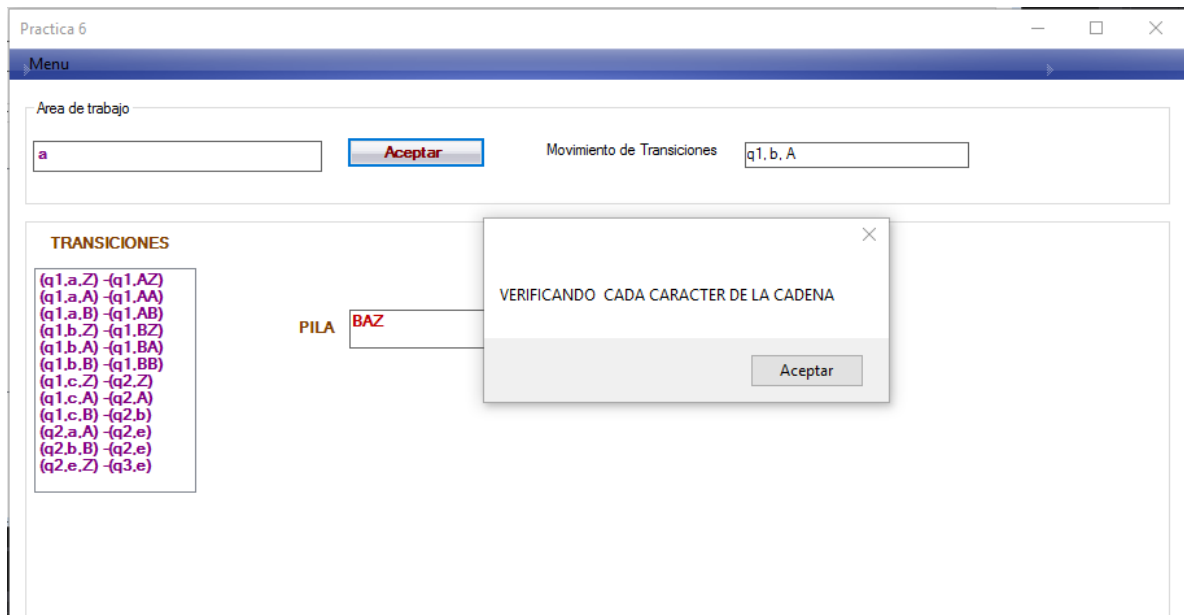


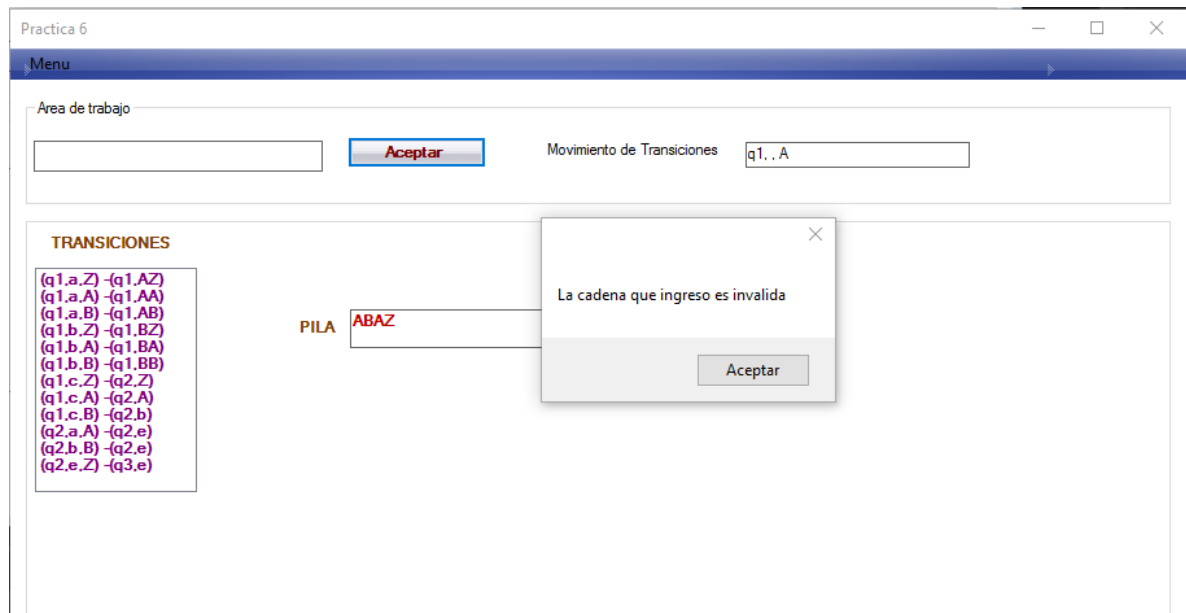
CADENAS INVALIDAS

1. Cadena: aba









Conclusión:

Esta práctica estuvo muy interesante ya que se pudo comprobar algunos ejercicios que se realizaron en clase, la parte más interesante de esta práctica fue implementar una pila dentro de un autómata esa estructura de datos la vi el semestre pasado y no creí que se utilizara para los autómatas, en resumen, esta práctica estuvo algo difícil, pero se logró.

Bibliografía

- Jeff Ferguson, B. P. (2003). *La biblia de C#*. Madrid: Anaya Multimedia (Grupo Anaya. S.A.).
- John E. Hopcroft, R. M. (2007). *Teoria de autómatas, lenguajes y computación*. Madrid: PEARSON EDUCACION S.A.