Práctica 6. Programa - Automata pila

Autor: Victor Ulises Miranda Chávez. Grupo: 5BM1

Fecha de entrega: 06/11/2022

1. Introducción

Un autómata de pila es un modelo de computación que utiliza una pila adicional para almacenar información y tomar decisiones durante su ejecución. Los autómatas de pila pueden ser deterministas o no deterministas. En el caso de los autómatas de pila deterministas, para cada estado y símbolo de entrada hay un único siguiente estado y una única acción de pila. En el caso de los autómatas de pila no deterministas, pueden haber varios siguientes estados y varias acciones de pila para un mismo estado y símbolo de entrada.

En este programa se utiliza un autómata de pila para reconocer el lenguaje libre de contexto $0^n1^n|n >= 1$, es decir, para aceptar cadenas de 0s y 1s de tal manera que la cantidad de 0s sea igual a la cantidad de 1s.

1.1. Objetivo

Programar un autómata de pila que sirva para reconocer el lenguaje libre de contexto $0^n1^n|n >= 1$.

Adicionalmente, el programa debe de contar con las siguientes características:

1. La cadena puede ser ingresada por el usuario o automáticamente. Si es aleatoriamente, la cadena no podrá ser mayor a 100,000 caracteres.
2. Mandar a un archivo y en pantalla la evaluación del autómata a través de descripciones instantáneas (IDs).
3. Animar el autómata de pila, solo si la cadena es menor igual a 10 caracteres.
4. En el reporte deben de estar pantallas del programa en ejecución de todas las características solicitadas.

2. Desarrollo

2.1. Lenguaje utilizado

- Python

2.2. Capturas de resultados:

```
Elija el modo:
1. Automatico
2. Manual
Inserte su opcion deseada: 2
Inserte su cadena: 00001111
Su entrada fue:  00001111
Tope:  Z0
Cadena aceptada
```

Figura 1: Selección del modo y cadena deseada

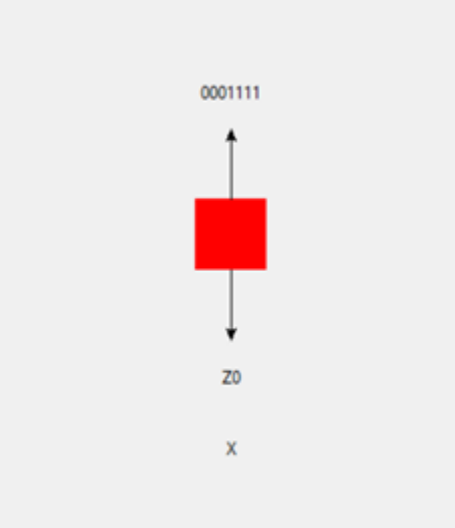

Figura 2: Animación de la pila en progreso

evaluacionPila.txt

```
00001111
d(q, 0, Z0) = [(q, XZ0)]
0001111
d(q, 0, X) = [(q, XXZ0)]
001111
d(q, 0, X) = [(q, XXXZ0)]
01111
d(q, 0, X) = [(q, XXXXZ0)]
1111
d(q, 1, XXXX) = [(p, XXX)]
d(p, 1, XXXX) = [(p, XXX)]
111
d(p, 1, XXX) = [(p, XX)]
11
d(p, 1, XX) = [(p, X)]
1
d(p, 1, X) = [(p, e)]
d(p, e, Z0) = [(f, Z0)]
```

Figura 3: Evaluación de la pila

3. Conclusiones:

Desarrollar esta práctica fue retroalimentativa porque que pude conocer la función de un autómata de pila el cual utiliza una pila para almacenar información y tomar decisiones, en este caso para reconocer el lenguaje libre de contexto $0^n1^n|n >= 1$. Por la parte práctica, fue bueno para conocer como elaborar y utilizar una pila hecha por nuestra cuenta.

4. Bibliografía:

Hopcroft, J. E., Motwani, R. & Ullman, J. D. (2008). Teoría de autómatas, lenguajes y computación 3/E. Pearson Educación.

5. Código:

```
import random
import time
from tkinter import Canvas, Tk

# Creamos la clase Pila
class Pila:
    def __init__(self):
        self.items = []

    def estaVacia(self):
        return self.items == ['Z0']

    def empujar(self, item):
        self.items.append(item)

    def sacar(self):
        return self.items.pop()

    def verTope(self):
        return self.items[len(self.items)-1]

    def tamano(self):
        return len(self.items)

def automataPila(pila: Pila, entrada):
    equis = "X"
    bandera = True
    for i in range(len(entrada)):
        bit = entrada[i]

        if bit == "0":
            if i == 0:
                f.write(f"d(q, 0, Z0) = [(q, {equis}Z0)]")
            else:
                equis += 'X'
                f.write(f"d(q, 0, X) = [(q, {equis}Z0)]")

            pila.empujar('X')
        else:
            if bandera == True:
                f.write(f"d(q, 1, {equis}) = [(p, {equis[:-1]})]\n")
                bandera = False

            if equis[:-1] != '':
                f.write(f"d(p, 1, {equis}) = [(p, {equis[:-1]})]")
            else:
                f.write(f"d(p, 1, {equis}) = [(p, e)]")

```

```
            equis = equis[:-1]

            if pila.estaVacia():
                return False

            pila.sacar()

        f.write("\n")

    if pila.verTope() != 'Z0':
        return False

    return True

def automataPila_Grafica(pila, entrada):
    i = 0
    aux = entrada

    canv.create_rectangle(300, 150, 350, 200, width=0, fill='red')
    canv.create_line(325, 150, 325, 100, arrow="last")
    canv.create_line(325, 200, 325, 250, arrow="last")
    canv.create_text(325, 75, text=entrada, tags="entrada1")
    canv.create_text(325, 275, text="Z0", tags="pila{i}")
    canv.pack()

    x = 325
    y = 325
    x2 = 325
    y2 = 75

    canv.update()
    time.sleep(2)

    equis = "X"
    bandera = True
    for j in range(len(entrada)):
        bit = entrada[j]
        f.write(aux + "\n")
        canv.delete("entrada1")

        if bit == '0':
            if j == 0:
                f.write(f"d(q, 0, Z0) = [(q, {equis}Z0)]")
            else:
                equis += 'X'
                f.write(f"d(q, 0, X) = [(q, {equis}Z0)]")
            canv.create_text(x, y, text="X", tags=f"pila{i}")
            pila.empujar('X')
            i += 1
            y += 50

        else:
            if bandera == True:
                f.write(f"d(q, 1, {equis}) = [(p, {equis[:-1]})]\n")
                bandera = False

            if equis[:-1] != '':
                f.write(f"d(p, 1, {equis}) = [(p, {equis[:-1]})]")
            else:
                f.write(f"d(p, 1, {equis}) = [(p, e)]")

            equis = equis[:-1]

            y -= 50
            if (pila.estaVacia()):
                return False
```

```
            i -= 1
            canv.delete(f"pila{i}")
            pila.sacar()

        canv.delete("entrada")

        aux = entrada[j + 1:]
        canv.create_text(x2, y2, text=str(aux), tags="entrada")
        canv.update()

        time.sleep(3)
        f.write("\n")

    canv.update()
    if pila.verTope() != 'Z0':
        return False

    f.write("d(p, e, Z0) = [(f, Z0)]")
    return True


# Creamos un menu para elejir el modo
print("Elija el modo:")
print("1. Automatico")
print("2. Manual")
opc = input("Inserte su opcion deseada: ")
entrada = ""

# Creamos una cadena aleatoria o le pedimos al usuario su cadena
if opc == '1':
    cantidadCeros = random.randint(1, 50)
    cantidadUnos = random.randint(1, 50)
    print("Cantidad ceros: ", cantidadCeros)
    print("Cantidad unos: ", cantidadUnos)

    for i in range(cantidadCeros):
        entrada += '0'
    for i in range(cantidadUnos):
        entrada += '1'
else:
    entrada = input("Inserte su cadena: ")

# Mostramos la cadena creada
print("Su entrada fue: ", entrada)

# Inicializamos la pila
pila = Pila()
pila.empujar('Z0')

# Abrimos el archuivo de texto donde guardamos los estados
f = open("evaluacionPila.txt", 'w')

# Evaluamos la cadena en el automata
# Si la cadena es menor que 10, animamos la pila, en otro caso no
esValida = False
if len(entrada) < 10:
    #Creamos la ventana para la animacion
    ventana = Tk()
    canv = Canvas(ventana, width=800, height=800)
    ventana.geometry("600x600")
    esValida = automataPila_Grafica(pila, entrada)
    canv.update()
    canv.place(x=0, y=0)
    ventana.mainloop()
else:
    esValida = automataPila(pila, entrada)
```

```
181
182 # Verificamos los resultados
183 if esValida:
184     print("Tope: ", pila.verTope())
185     print("Cadena aceptada")
186 else:
187     print("Tope: ", pila.verTope())
188     print("No es valida")
```