

## **Documentación: Actividad 3.2 Programando un DFA**

### **Materia:**

Implementación de Métodos Computacionales

### **Integrantes:**

Guillermo Tafoya Milo - A01633790

Engels Emiliano Miranda Palacios - A01423398

Jorge Hernández Montero - A01733616

### **Campus:**

Monterrey

### **Fecha:**

13 de marzo del 2022

## **Introducción**

En este documento se presentará en resumidas cuentas el algoritmo implementado para la resolución de la actividad 3.2 “Programando un DFA” y se explicarán algunos puntos importantes como el lenguaje de programación en que fue desarrollado, la manera de ejecutarlo y los requerimientos para que funcione correctamente.

## **Lenguaje de Programación: Python 3.10.4**

### **Requerimientos:**

1. Dispositivo con Python 3+ instalado y que sea capaz de ejecutarlo.
2. Archivo main.py con el algoritmo.
3. Archivo de prueba para correr el programa.

Nota: No es necesario instalar nada más además de Python.

### **Indicaciones para Ejecutar el Programa:**

1. Tener en una carpeta el archivo “main.py” y un archivo de prueba “test.txt”.
2. Abrir una consola de comandos en la ruta de la carpeta.
3. Ejecutar el comando “python main.py test.txt resultados.txt”, donde “resultados.txt” será el archivo de salida donde se guardará el resultado arrojado por el algoritmo.

Nota: Es importante ejecutar el comando de la manera en que se indica, ya que si sólo se pone “python man.py” lanzará un error.

## **Resultados**

El programa dará como salida una lista en la cual, en cada renglón se mostrará el contenido del token detectado y el tipo de token, de la siguiente forma:

Token	Tipo
b	Variable
=	Asignación
7	Entero
a	Variable
=	Asignación
32.4	Real

Y cumpliendo las siguientes reglas:

### Reglas de formación de algunos tokens

#### 1. Variables:

- Deben empezar con una letra (mayúscula o minúscula).
- Sólo están formadas por letras, números y *underscore* ('\_').

#### 2. Números reales (de punto flotante):

- Pueden ser positivos o negativos
- Pueden o no tener parte decimal pero deben contener un punto (e.g. 10. o 10.0)
- Pueden usar notación exponencial con la letra E, mayúscula o minúscula, pero después de la letra E sólo puede ir un entero positivo o negativo (e.g. 2.3E3, 6.345e-5, -0.001E-3, .467E9).

#### 3. Comentarios:

- Inician con // y todo lo que sigue hasta que termina el renglón es un comentario

## Anexos

### ***Entrada: test.txt***

b=7

a = 32.4 \* (-8.6 - b) / 6.1E-8 // Esto tmb es un comentario

d = a ^ b // Esto es un comentario

variable1 = 22.0E17 \* 45.1

variable2=-45 +32

### **Salida: resultados.txt**

b,Variable

=,Asignacion

7,Entero

a,Variable

=,Asignacion

32.4,Real

\*,Multiplicacion

(,Parentesis Izquierdo

-,Resta

8.6,Real

-,Resta

b,Variable

),Parentesis Derecho

/,Division

6.1E-8,Real

// Esto tmb es un comentario,Comentario

d,Variable

=,Asignacion

a,Variable

^,Potencia

b,Variable

// Esto es un comentario,Comentario

variable1,Variable

=,Asignacion

22.0E17,Real

\*,Multiplicacion

45.1,Real

variable2,Variable

=,Asignacion

-,Resta

45,Entero

+,Suma

32,Entero

