

## Autómata Finito Determinista



Tecnológico  
de Monterrey

### Link del código

GitHub 

### Propósito

*Una de las aplicaciones de los autómatas finitos determinísticos es la implementación de reconocedores de tokens en un lenguaje de programación (conocido como Lexer en los compiladores).*

### Reglas de formación de algunos tokens

#### Variables:

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

#### 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).*

#### Comentarios

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

## Algoritmo

- **Variable:** Inicia con una letra (mayúscula/minúscula), para después tener N cantidad de letras(mayúscula/minúscula), números o guiones bajos.
- **Número:** El número puede iniciar con negativo o punto o que directamente inicie con un dígito, después tener N cantidad de dígitos y si es entero, que acabe. Para los decimales tiene que estar seguidos por más números y para las exponenciales puede iniciar con negativo y ser decimal.
- **Símbolo:** Identifica el símbolo que se encuentra en la iteración actual y regresa el valor basado en la siguiente tabla:

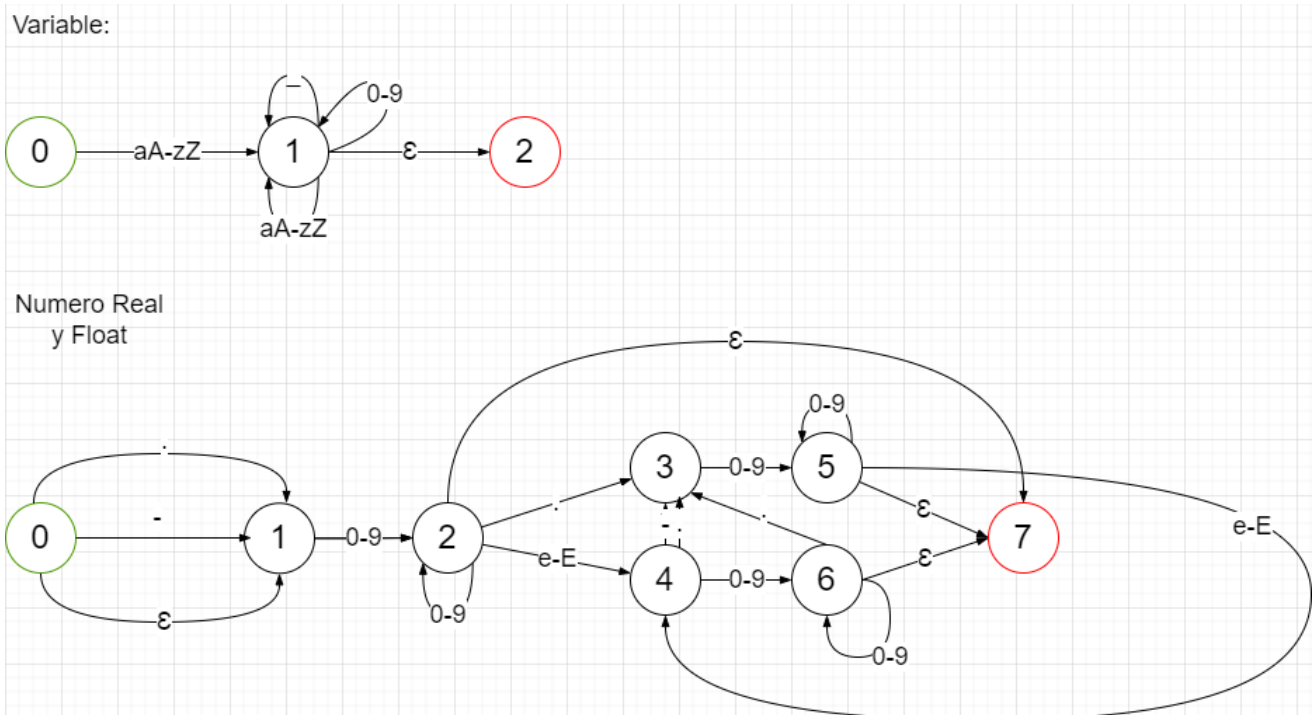
Token	Tipo
b	Variable
=	Asignación
7	Entero
a	Variable
=	Asignación
32.4	Real
*	Multiplicación
(	Paréntesis que abre
-8.6	Real
-	Resta
b	Variable
)	Paréntesis que cierra

## Integrantes

- Marco Antonio Camalich Pérez - A01351725
- José Ángel Rico Mendieta - A01707404

/	División
6.1E-8	Real
d	Variable
=	Asignación
^	Potencia
b	Variable
// Esto es un comentario	Comentario

### Visualización de autómatas



### Integrantes

- Marco Antonio Camalich Pérez - A01351725
- José Ángel Rico Mendieta - A01707404

## Manual de usuario

*El programa está hecho con el lenguaje C++, por lo que se necesita del compilador MinGW para que este pueda ser ejecutado. En la consola, se necesita ejecutar los siguientes comandos en la terminal ubicada en el directorio de los archivos:*

*1.- El primer paso equivale a compilar el programa, lo cuál se puede realizar por medio de la ejecución del siguiente código en terminal.*

`g++ main.cpp -o main`

*2.- Si la compilación es exitosa, el paso siguiente es ejecutar el archivo resultado de la compilación, el cuál posee el nombre de:*

`main.exe`

*3.- Tras concluir el paso dos,, el programa pregunta por el nombre del archivo de texto a utilizar, por lo que, aparte de que debe de estar en la misma carpeta que los archivos y que este tiene que terminar con la extensión .txt, se debe colocar como respuesta la siguiente indicación en negritas.*

Nombre del archivo (con extensión .txt): **prueba.txt**

*El programa se ejecutará como lo menciona la documentación y la salida de este será que por cada línea, el código presentará el tipo de token y el resultado que es. Un ejemplo sería el siguiente:*

LINE 1

Variable : b

Asignacion : =

Entero : 7

LINE 3

Variable : a

Asignacion : =

Real : 32.4

Multiplicacion : \*

Parentesis que abre : (

## Integrantes

- Marco Antonio Camalich Pérez - A01351725
- José Ángel Rico Mendieta - A01707404

Real : -8.6

Resta : -

Variable : b

Paréntesis que cierra : )

Division : /

Real : 6.1E-8

LINE 5

Variable : d

Asignacion : =

Variable : a

Potencia : ^

Variable : b

Comentario: // Esto es un comentario

## Integrantes

- *Marco Antonio Camalich Pérez - A01351725*
- *José Ángel Rico Mendieta - A01707404*