Instituto Tecnológico de Estudios Superiores de Monterrey

Materia:
Implementación de métodos computacionales

Grupo:
604

Maestro:

Dr. Jesús Guillermo Falcón Cardona

Nombre del Trabajo:

Actividad 3.2 Programando un DFA

Autores:
Jorge Del Barco Garza | A01284234
Patricio mendoza Pasapera | A00830337
José Emilio Alvear Cantú | A01024944

Fecha:
24 de abril del 2022

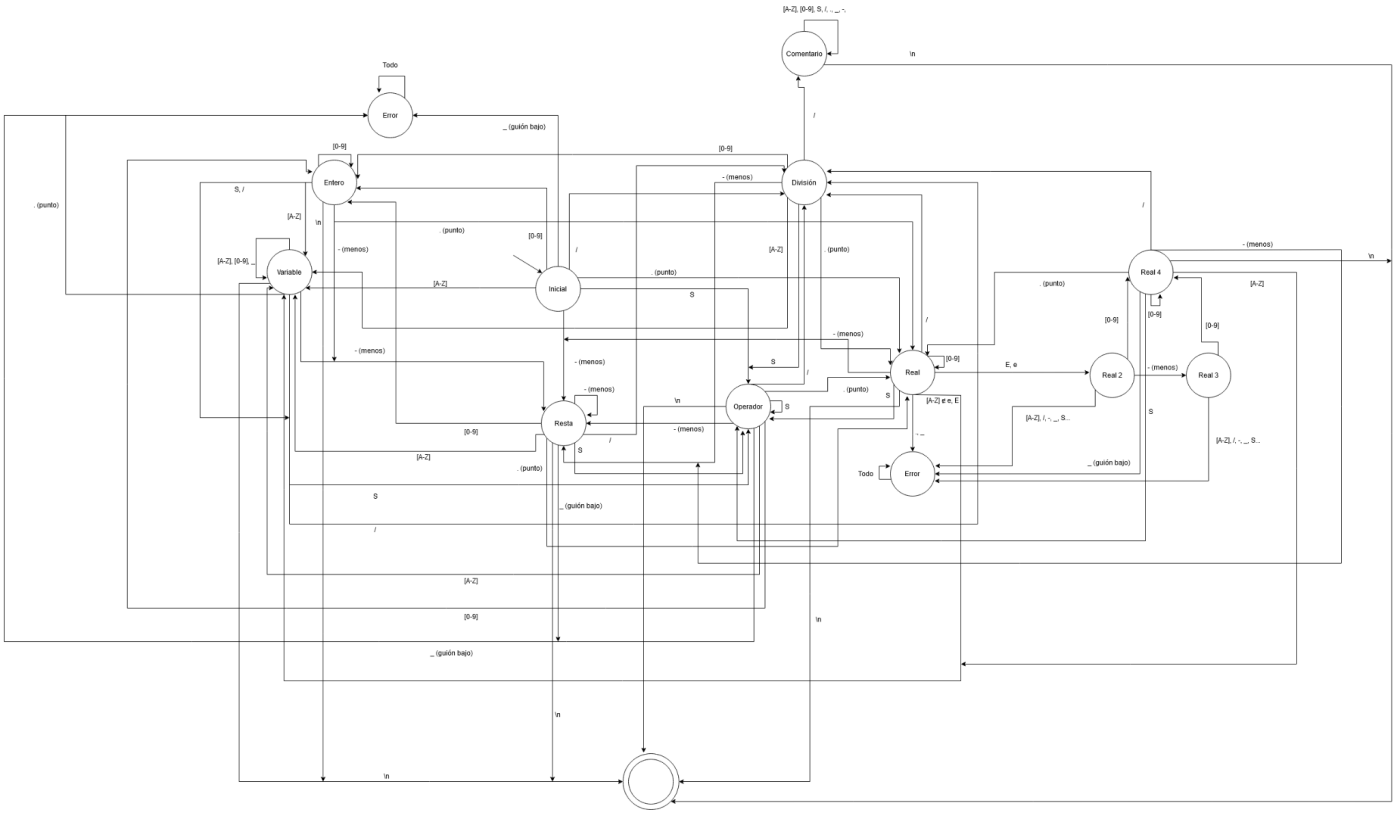
DFA

Para la solución del problema, se realizó un DFA con las siguientes consideraciones:

- S = [+, =, *, ^, (,)]
- [A-Z] = letras alfabéticas, incluyendo mayúsculas y minúsculas
- [0-9] = números del cero al nueve
- _ = Guión bajo

El DFA considera las siguientes reglas:

- Solo se llega a un estado de aceptación una vez que se haya leído todo el archivo y no se hayan encontrado errores de sintaxis.
- Se consideró el caracter de nueva línea \n para transitar al estado de aceptación.



Link a una mejor visualización:
https://drive.google.com/file/d/1DJfd1pJALXhxahelBVbUFqETOwRYiKt3/view?usp=sharing

Código

El código se basa en el DFA de la sección anterior. Se lee el archivo línea por línea. De cada línea, se va revisando caracter por caracter y se intercambia el estado dependiendo de lo que lea. Los caracteres se van agregando a una lista para imprimirlos en caso de que formen un token válido. Si se detecta un cambio de estado cuando hay algún token válido, se imprime el caracter en el archivo.. Si se encuentra un caracter no válido, se imprimirá un error de sintaxis y se terminará el programa.

La lógica más importante se encuentra en la función `processLine()`. Se recomienda tener el DFA anterior a la mano para entender la lógica del código mejor.

En el .zip se incluye un archivo expresiones.txt, el cual se usó para los casos de prueba. Corriendo el código, se generará otro archivo de salida con los resultados.

Manual de Uso

El código de DFA.py recibe el nombre de archivo.txt o su directorio como entrada. Este archivo.txt debe contener las expresiones que se quieran leer. Una vez que se haya corrido el código, como salida se imprimirán los tokens y su tipo en otro archivo.txt llamado "output.txt". El contenido de output.txt será limpiado cada vez que el código vuelva a ser utilizado.

El código puede ser usado de dos maneras diferentes:

1. **A través de la línea de comandos**

DFA.py puede ser utilizado a través de la línea de comandos. Se deberá pasar el nombre del archivo como un argumento de línea de comando. Lo siguiente es un ejemplo de cómo podría ser usada esta opción en la terminal CMD de windows. Recuerde estar en el mismo directorio que el archivo DFA.py:

```
python -m DFA "expresiones.txt"
```

Debido a que no contamos con una computadora Mac, no podemos confirmar que el siguiente ejemplo sea funcional. Sin embargo, de acuerdo a nuestra investigación, el siguiente comando debería funcionar para ejecutar DFA.py en una línea de comandos de una Mac:

```
python DFA.py expresiones.txt
```

2. **Importando la función `lexerAritmetico()` en otro script**

Se puede importar la función `lexerAritmetico()` en el script de su preferencia de la siguiente manera:

```
from DFA import lexerAritmetico
```

La función `lexerAritmetico()` toma un parámetro: el nombre (o el directorio) del archivo.txt de donde se leerán las expresiones. En caso de que se pase el directorio a la función, es recomendable utilizar un raw string de Python (es decir, prefijar el string del directorio con una r).

Es importante recalcar que el archivo.txt con las expresiones que se quieran leer o el script en el que esté importando la función debe estar en el mismo directorio que DFA.py.