

CASTRO CRUCES JORGE E.	INSTITUTO POLITÉCNICO NACIONAL	ESCUELA SUPERIOR DE CÓMPUTO
PRIMER PARCIAL	PRACTICA 1 – GENERADOR DE AF	25/03/2021
3CV18	COMPILADORES	Hernández Olvera Luis E

## SECCIÓN TEÓRICA O INTRODUCCIÓN

### Autómatas finitos:

- 1) Los autómatas finitos son reconocedores; sólo dicen “sí” o “no” en relación con cada posible cadena de entrada.
- 2) Los autómatas finitos pueden ser de dos tipos:
  - a) Los autómatas finitos no deterministas (AFN) no tienen restricciones en cuanto a las etiquetas de sus líneas. Un símbolo puede etiquetar a varias líneas que surgen del mismo estado, y, la cadena vacía, es una posible etiqueta.
  - b) Los autómatas finitos deterministas (AFD) tienen, para cada estado, y para cada símbolo de su alfabeto de entrada, exactamente una línea con ese símbolo que sale de ese estado.

Tanto los autómatas finitos deterministas como los no deterministas son capaces de reconocer los mismos lenguajes. De hecho, estos lenguajes son exactamente los mismos lenguajes, conocidos como lenguajes regulares, que pueden describir las expresiones regulares.[1]

### Autómatas finitos no deterministas:

Un autómata finito no determinista (AFN) consiste en:

- 1) Un conjunto finito de estados  $S$ .
- 2) Un conjunto de símbolos de entrada  $\Sigma$ , el alfabeto de entrada. Suponemos que, que representa a la cadena vacía, nunca será miembro de  $\Sigma$ .
- 3) Una función de transición que proporciona, para cada estado y para cada símbolo en  $\Sigma \cup \{\epsilon\}$ , un conjunto de estados siguientes.
- 4) Un estado  $s_0$  de  $S$ , que se distingue como el estado inicial.
- 5) Un conjunto de estados  $F$ , un subconjunto de  $S$ , que se distinguen como los estados aceptantes (o estados finales).[1]

### AFD y lenguajes:

Para trabajar con los AFD es necesario usar ciertas definiciones y notaciones. Si  $M$  es un AFD, entonces el lenguaje aceptado por  $M$  es

$$L(M) = \{w \in \Sigma^* | w \text{ es aceptada por } M\}$$

Por tanto,  $L(M)$  es el conjunto de cadenas que hacen que  $M$  pase de su estado inicial a un estado de aceptación. Por ejemplo, el lenguaje aceptado por el AFD (\*), presentado en la última sección, es

$$L(M) = \{w \in \{a, b\}^* | w \text{ no contiene tres } b \text{ consecutivas}\}$$

Merece la pena hacer hincapié en que  $L(M)$  está formado por todas las cadenas aceptadas por  $M$ , y no que es un conjunto de cadenas que son todas aceptadas por  $M$ . [2]

Python tiene muchos tipos de estructura de datos, una de ellas es la lista, es por ello por lo que en este tutorial aprenderemos a usarlas y veremos sus métodos más utilizados. Si no conoces qué es Python, te lo explico a continuación.

*"Python es un lenguaje de programación interpretado de alto nivel y orientado a objetos, con el cual podemos crear todo tipo de aplicaciones."*

En algunos lenguajes de programación se las conocen como arreglos o matrices; y se caracterizan porque los elementos están entre corchetes y separados por una coma.

***lista = [1, 2, 3, 4]***

### **¿Qué es una lista?**

Una lista es una estructura de datos y un tipo de dato en Python con características especiales. Lo especial de las listas en Python es que nos permiten almacenar cualquier tipo de valor como enteros, cadenas y hasta otras funciones; por ejemplo:

***lista = [1, 2.5, 'DevCode', [5,6],4] [3]***

## **PROBLEMAS RELACIONADOS AL TEMA A LOS CUALES TE ENFRENTASTE AL PROGRAMAR LA PRÁCTICA**

A todos, desde el hecho de que no manejo ni conozco la Programación Orientada a Objetos, hasta el punto de que son pocos los recuerdos de la materia de Teoría Computacional, que curse hace dos años.

Dicho lo anterior, tuve que referirme a libros gratuitos publicados en la internet que me permitieran recordar y refrescar los vagos recuerdos de dicha materia que nadie me dijo que volvería a retomar en la materia de Compiladores.

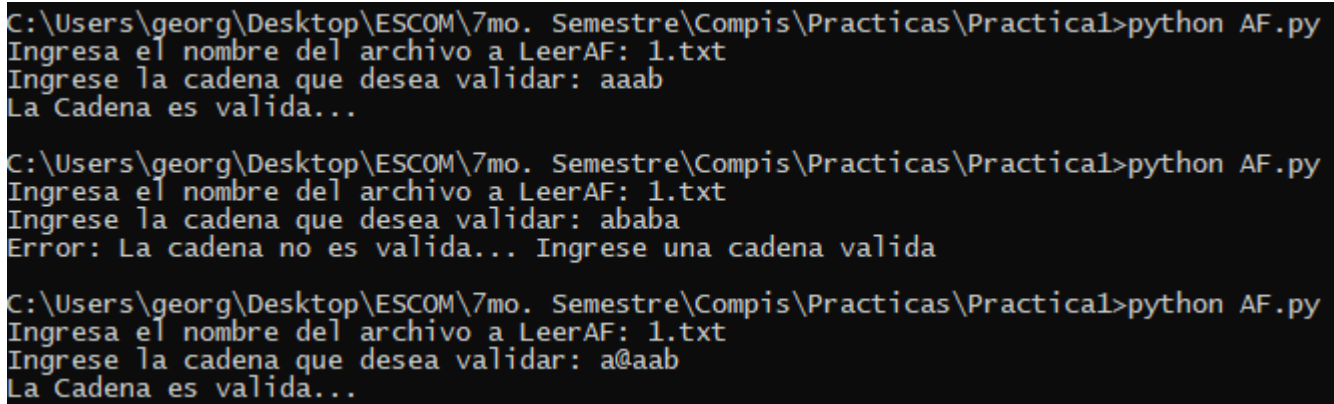
Gracias a que en semestres anteriores curse materias que requerían que yo aprendiese este famoso lenguaje de alto nivel que es Python, y que debo reconocer que es bastante amigable y sencillo de aprender, y programar (Por ejemplo: Análisis de Imágenes, Algoritmos Genéticos y actualmente Reconocimiento de patrones). Es por eso por lo que al menos en el lenguaje de programación no tuve tantos problemas, pero si en la técnica de Programación Orientada a Objetos.

Por supuesto, que me tuve que referir a repositorios que me permitiesen darme una idea de como empezar a programar está práctica, lo que cual no significa que yo haya copiado y pegado código de forma descarada y sin escrúpulos. En cambio me permitieron aprender algo nuevo que para nada fue fácil, porque repito, tuve que aprender POO en poco tiempo, y espero que con el paso de las semanas y las prácticas, yo pueda afinar y perfeccionar mis conocimientos.

## CAPTURAS DE PANTALLA

Ejecución del programa leyendo el AF 1 con las cadenas siguientes:

- aaab
- ababa
- a@aab



```
C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 1.txt
Ingresa la cadena que desea validar: aaab
La Cadena es valida...

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 1.txt
Ingresa la cadena que desea validar: ababa
Error: La cadena no es valida... Ingresa una cadena valida

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 1.txt
Ingresa la cadena que desea validar: a@aab
La Cadena es valida...
```

Es la imagen anterior, se muestra la ejecución del código analizando el Autómata Finito 1, y el mismo, arrojó los siguientes resultados:

- 1) La cadena es valida
- 2) La cadena no es valida
- 3) La cadena es valida

Cabe señalar que la segunda cadena no es válida, ya que no pertenece al Lenguaje que reconoce este AF.

Ejecución del programa leyendo el AF 2 con las cadenas siguientes:

- ab
- a@b
- aaab
- abab

```
C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 2.txt
Ingresa la cadena que desea validar: ab
La Cadena es valida...
La Cadena es valida...

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 2.txt
Ingresa la cadena que desea validar: a@b
La Cadena es valida...
La Cadena es valida...

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 2.txt
Ingresa la cadena que desea validar: aaab
La Cadena es valida...

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 2.txt
Ingresa la cadena que desea validar: abab
Error: La cadena no es valida... Ingresa una cadena valida
```

Es la imagen anterior, se muestra la ejecución del código analizando el Autómata Finito 2, y el mismo, arrojó los siguientes resultados:

- 1) La cadena es valida
- 2) La cadena es valida
- 3) La cadena es valida
- 4) La cadena no es valida

Cabe señalar que la cuarta cadena no es válida, ya que no pertenece al Lenguaje que reconoce este AF.

Ejecución del programa leyendo el AF 2 con las cadenas siguientes:

- a.aa
- +a.a
- +a@.@a
- +aa.aa

```
C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 3.txt
Ingresa la cadena que desea validar: a.aa
La Cadena es valida...

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 3.txt
Ingresa la cadena que desea validar: +a.a
La Cadena es valida...
La Cadena es valida...

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 3.txt
Ingresa la cadena que desea validar: +a@.@a
La Cadena es valida...
La Cadena es valida...

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Compis\Practicas\Practical>python AF.py
Ingresa el nombre del archivo a LeerAF: 3.txt
Ingresa la cadena que desea validar: +aa.aa
La Cadena es valida...
La Cadena es valida...
```

Es la imagen anterior, se muestra la ejecución del código analizando el Autómata Finito 3, y el mismo, arrojó los siguientes resultados:

- 1) La cadena es valida
- 2) La cadena es valida
- 3) La cadena es valida
- 4) La cadena es valida

Cabe señalar que todas las cadenas ingresadas fueron validas, ya que pertenecen al Lenguaje que reconoce este AF.

## CONCLUSIÓN

### ¿Qué aprendí de esta práctica?

Principalmente aprendí a programar con Programación Orientada a Objetos, ya que como mencioné en la sección de **Problemas relacionados al tema a los cuales te enfrentaste al programar la práctica**, no conocía para nada la terminología, ni mucho menos como aplicarlo en otras materias, este caso Compiladores; Y siendo honestos me pareció bastante fascinante e interesante el imaginar en que otras cosas o aplicaciones puedo llegar a implementarlo.

### ¿En qué me ayudó esta práctica?

Me ayudo con muchísimos ámbitos, cabe resaltar el tema de Listas, tema que vi hace varios años en una materia que no aproveché al máximo. Las listas me parecieron muy útiles para programar este analizador, y más en el Lenguaje Python, lenguaje que cada día me sigue sorprendiendo.

Por último, quiero concluir esta practica diciendo que nunca creí que llegaría a conocer que hay detrás de un compilador de lenguajes de programación, y es que esto apenas comienza y estoy deseoso de conocer con mayor amplitud esta materia tan cotidiana en nuestro día a día como programadores.

## REFERENCIAS

- [1] Aho, A., 2008. Compiladores. 2nd ed. México: Pearson Educación de México, SA de CV, p.147.
- [2] Kelley, D., Joyanes Aguilar, L. and Diéz Platas, L., 1995. Teoría de autómatas y lenguajes formales. 1st ed. Madrid: Prentice Hall, p.59.
- [3] Mariños Urquiaga, J., 2021. Listas en Python. [online] DevCode Tutoriales. Available at: <https://devcode.la/tutoriales/listas-python/> [Accessed 23 March 2021].