

Ellioth Romero A01781724

Luis Daniel Filorio Luna A01028418

Instrucciones:

1. Selecciona un lenguaje de programación que te resulte familiar, y determina las categorías léxicas que tienen en común (por ejemplo, palabras reservadas, operadores, identificadores, comentarios, etc.).
2. Usando Python implementa un resaltador léxico que logre escanear los elementos léxicos de cualquier archivo fuente provisto.
3. El programa debe convertir su entrada en documentos de HTML+CSS que resalten su léxico con diferentes colores de acuerdo al tipo de token (por ejemplo, palabras reservadas, operadores, identificadores, comentarios, etc.).
4. Utiliza las convenciones de codificación del lenguaje en el que está implementado tu programa.
5. Reflexiona sobre la solución planteada, los algoritmos implementados y sobre el tiempo de ejecución de estos.
6. Calcula la complejidad de tu algoritmo basada en el número de iteraciones y contrástala con el tiempo obtenido en el punto 5 para ver si son parecidos.
7. Plasma en un breve reporte de una página las conclusiones de tu reflexión en los puntos 7 y 8. Agrega además una breve reflexión sobre las implicaciones éticas que el tipo de tecnología que desarrollaste pudiera tener en la sociedad.

Para el ejercicio del primer punto decidimos comparar C++ con Python 3, ambos comparten muchas categorías léxicas, como por ejemplo identificadores, palabras clave, literales, operadores, separadores (aunque se ven muy diferentes en ambos lenguajes, cumplen la misma función) y comentarios. Un tipo de categoría léxica que tiene C++ pero que no tiene Python son los preprocesadores, que son palabras como **#include**, es importante no confundirlas con **import** en python, que tiene una función distinta.

Mi compañero y yo implementamos un resaltador léxico que toma un archivo externo de Python y lo interpreta como string. Ese string se analiza mediante un proceso el cual detallaremos más adelante, el cual nos ayudará a determinar las categorías léxicas de las palabras dentro del script de Python. El objetivo final es que una vez identificadas las categorías, se puedan pasar a código de HTML, el cual nos ayudará, mediante los colores que nos ofrece, a tener una representación visual clara de la categoría a la que cierto conjunto de caracteres pertenece.

FSM diagrama:

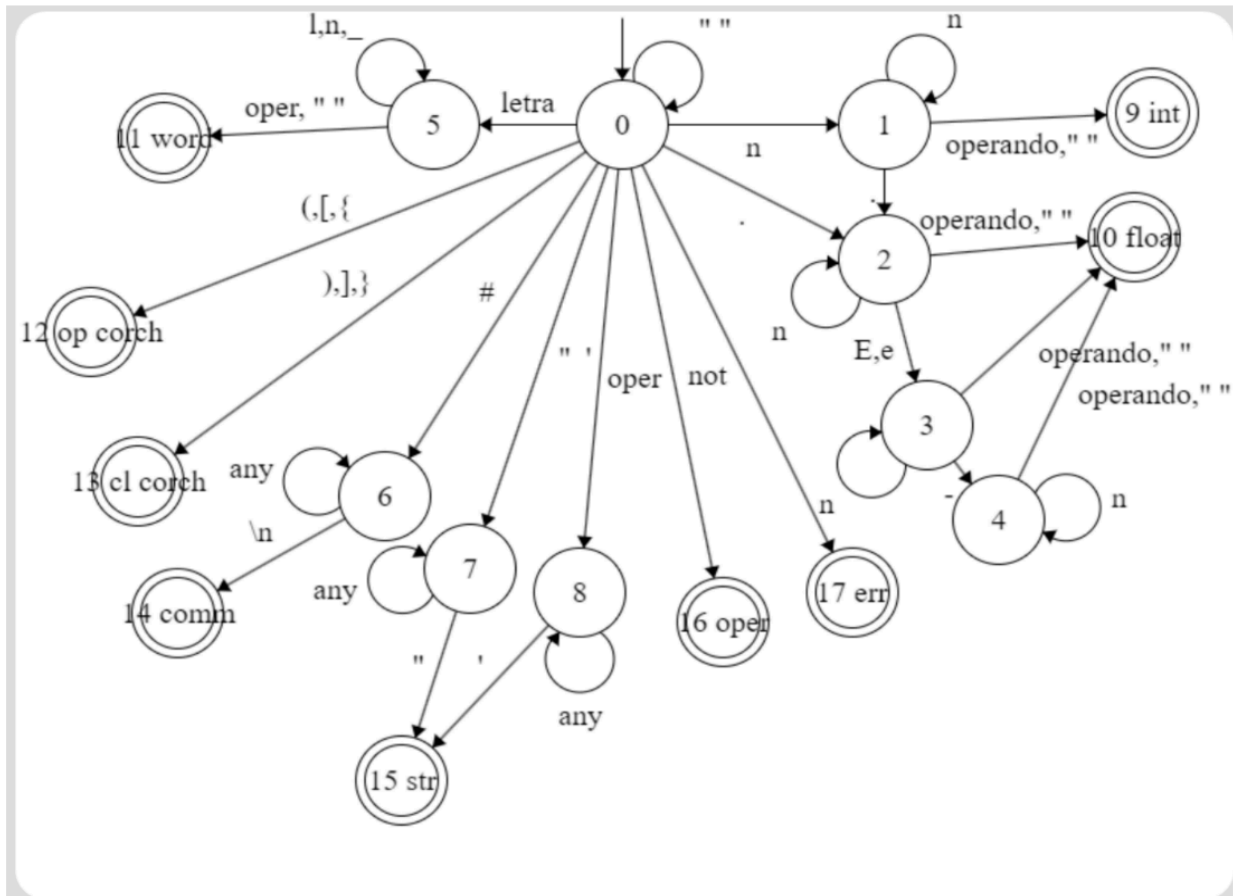


Tabla de estados:

	“ “	num	Letra && !Ee &&!e	{ ([#	})]	“	‘	Operat or && !-	.	else	Ee e	-	_	\n
0	0	1	5	12	6	13	7	8	16	2	17	5	16	5	0
1	9	1	9	9	9	9	9	9	9	2	9	9	9	9	9
2	10	2	10	10	10	10	10	10	10	10	10	3	10	10	10
3	10	3	10	10	10	10	10	10	10	10	10	10	4	10	10
4	10	4	10	10	10	10	10	10	10	10	10	10	10	10	10
5	11	5	5	11	11	11	11	11	11	11	11	5	11	5	11
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	14
7	7	7	7	7	7	7	15	7	7	7	7	7	7	7	17
8	8	8	8	8	8	8	8	15	8	8	8	8	8	8	17

La solución propuesta funciona mediante la implementación de la tabla de estados, que nos ayuda a replicar el comportamiento del FSM dentro del script de Python. Esta solución es más eficiente que otras que se pudieran proponer y que darían el mismo resultado, como por ejemplo iterar sobre el archivo (total de valores) veces, buscando en cada iteración si existen coincidencias con cada tipo de valor. Es más sencillo de implementar, pero en este caso solo pasamos una sola vez por el string obtenido.

La complejidad del código va a ser $O(n)$ ya que su tiempo de ejecución tiene una relación lineal con el número de elementos que haya en el archivo que estamos leyendo, y solo se iterará una vez sobre el string. El tiempo de ejecución crecerá proporcionalmente al número de caracteres del archivo obtenido.

A primera vista, podría parecer que la creación de un lexer no escapa de lo técnico, que la ética no tiene nada que ver con estas tecnologías, pero en realidad es todo lo contrario. Toda tecnología debe ser creada con una meta y valores en mente, en este caso, es importante crear tecnologías de lectura de calidad, pues en el caso de descuidar este aspecto en un lexer, éste se puede convertir en un arma en contra de quien lo usa, pues el mal código puede estar abierto a vulnerabilidades y con esto violar principios de privacidad y seguridad.