

ANÁLISIS DE HERRAMIENTAS USADAS PARA LA SOLUCIÓN DE LAS SITUACIONES PROBLEMA

LUIS DANIEL FILORIO A01028418, ELLIOTH DENZEL ROMERO A01781724

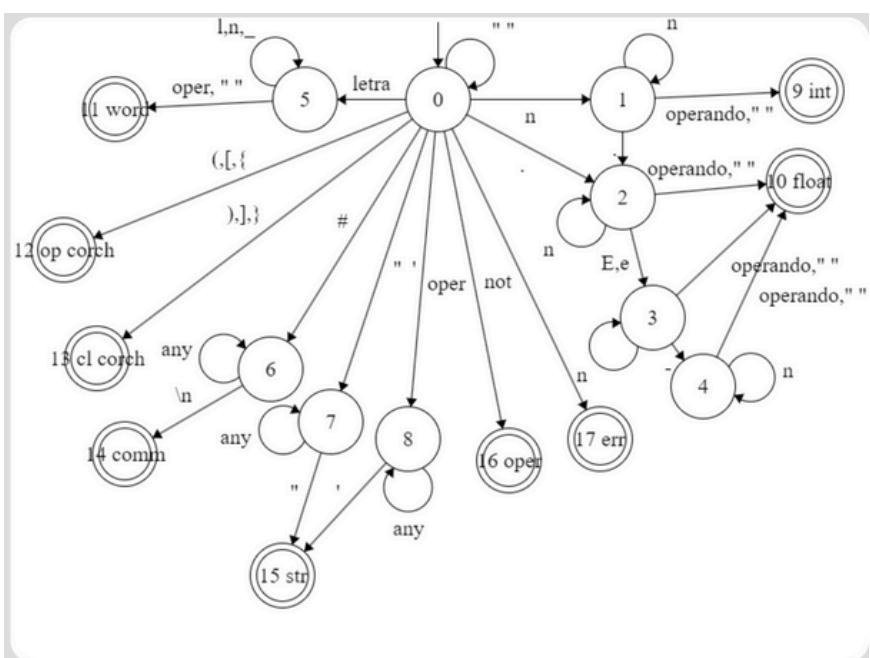
FASE 1

Para la primera fase de la evidencia desarrollamos un resaltador de sintaxis para el lenguaje de python, este lo realizamos con el uso de un automáta NFA, para la lógica del resaltador, a partir de este automáta realizamos una tabla de estados para ayudarnos a pasar este problema más sencillo a código. Realizamos un código en python para poder desarrollar este lexer que lo que hace es leer un archivo .py, mete lo que lee a un archivo HTML y colorea las palabras dependiendo de la sintaxis de python

FASE 2

Para la segunda fase de la evidencia se escribieron 3 programas que hacen la misma tarea, pero de diferente manera donde todos los programas revisan todos los archivos dentro de un folder y coleccionan una lista de direcciones donde haya solo archivos de python y de esta manera implementar la fase 1 de la evidencia a todos estos archivos. En un programa no hubo procesamiento paralelo, en otro usamos multiprocesos y en el último usamos hilos.

AUTOMÁTA



Este es el automata que realizamos para la fase 1, este automata tiene 17 estados totales. Puede leer números enteros, flotantes, palabras en las cuales puede detectar las que son clave para python, abrir y cerrar corchetes y parentésis, leer comentarios y tambien la forma de escribir strings en python, asi como operadores.

RESULTADO FASE 1

El resultado de la fase 1 fue un archivo de tipo HTML el cual contuviera el código de python leído y coloreado dependiendo de que fuera con base al autómat. Un ejemplo de esto es el siguiente

```
import
math
# False, True, None
a=
False
b=
True
c=
None
n=
42.3
# and, or, not
if a and not b or c:
pass
# as, with
with open('example.txt','w') as f:
f.write('Hello, world!')
```

RESULTADO FASE 2

Se midieron los tiempos para múltiples ejecuciones de cada programa y se llegó a lo siguiente:

- La solución sin procesamiento paralelo tuvo un tiempo de ejecución promedio de 0.03 s
- La solución de procesamiento paralelo usando multiprocesos, tuvo un tiempo de ejecución promedio de 0.09 s a veces 0.011 s
- La solución del procesamiento paralelo utilizando hilos tuvo un tiempo de ejecución promedio de 0.012 s