

Universidad de San Carlos de Guatemala
Facultad de Ingeniería en Ciencias y Sistemas
Lenguajes Formales
Ingeniero David Morales
Auxiliar: Bryan López



Luis Mariano Moreira García
202010770

Índice

Objetivos y requisitos.....	3
Explicación clases.....	4
Explicación automatas.....	6

Objetivo

El objetivo de este programa es el poder realizar encuestas de maneras totalmente eficiente mediante el uso de una interfaz gráfica hecha en Python.

El programa por medio de estados es capaz de identificar las palabras e identificar los tokens, ya sea una cadena, un símbolo del lenguaje o una palabra reservada del programa.

Requisitos:

- Contar con al menos Windows 7
- 4GB de memoria RAM
- **1 gb de almacenamiento (Variable)**
- **Intel Atom**
- **Python versión 3.00 en adelante**

Funcionamiento del programa:

Main:

Funciona para mandar a llamar a la clase window, que es la que realiza todo el funcionamiento.

Window:

La clase responsable de generar la ventana que se despliega, al ser aquí donde se trabaja, se mandan a llamar los metodos de las clases lexical, errors, reports.

Lexical:

Esta clase no solo se encarga de realizar un estudio léxico, aquí es donde se trabaja el estudio semantico para realizar el html.

Se procedera a explicar los estados de la aplicación:

Estado 0: Determina que camino escoger

Estado 1: Determina las cadenas

Estado 2-10: Determina las palabras reservadas por medio de booleanos, esto con el motivo de no hacer más estados ya que verifica carácter por carácter, por ejemplo tenemos la palabra tipo:

Se determinan sus booleanos:

```
tTipo=False  
iTipo=False  
pTipo=False
```

Luego de determinar arranca por el tipo 0:

```
elif letter== 't' or letter=='T':  
    newLetter+=letter  
    column+=1  
    state=2  
    tTipo=True
```

Luego va por el tipo 2 y es aquí por medio de los booleanos que aclara cual

tomar:

```
#Ti -> Tipo
elif tTipo:
    if letter=='i' or letter=='I':
        iTipo=True
        newLetter+=letter
        column+=1
        state=3
    else:
        column+=1
        state=0
        newLetter+=letter
        tTipo=False
        error=Error("Entrada incorrecta: {} -> {}".format(newLetter,letter),"Se esperaba: I|i para formar: Tipo",line,column)
        self.errorsList.append(error)
        newLetter=""
```

Else es el error que puede generarse y lo guarda en una lista de errores

Como la palabra Tipo, tiene 4 letras, esta le toma hasta 3 estados para poder formar la palabra reservada:

```
newLetter=""
#Tipo -> Tipo
elif pTipo:
    if letter=='o' or letter=='O':
        column+=1
        newLetter+=letter
        token=Token("Palabra Reservada",newLetter,line, column)
        self.tokensList.append(token)
        newLetter=""
        tTipo=False
        iTipo=False
        pTipo=False
        state=0
    else:
        column+=1
        state=0
        newLetter+=letter
        tTipo=False
        iTipo=False
        pTipo=False
        error=Error("Entrada incorrecta: {} -> {}".format(newLetter,letter),"Se esperaba: O|o para formar: Tipo",line,column)
        self.errorsList.append(error)
        newLetter=""
```

Esta regresa a estado 0, así sucesivamente para cada carácter que no es acompañado por ningún “[

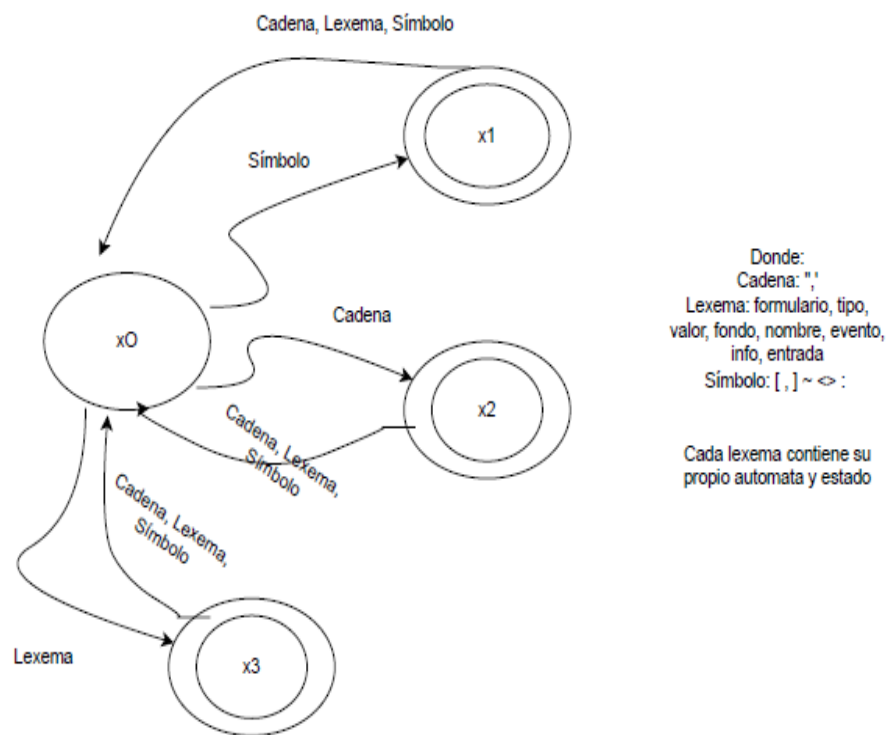
Estado 11: Funciona para los símbolos del programa, a continuación se explicaran los automatas y expresiones regulares que se utilizaron.

Explicación del automata:

Como se menciona con anterioridad, el programa funciona a través de estados los cuales están formados por la siguiente expresión regular:

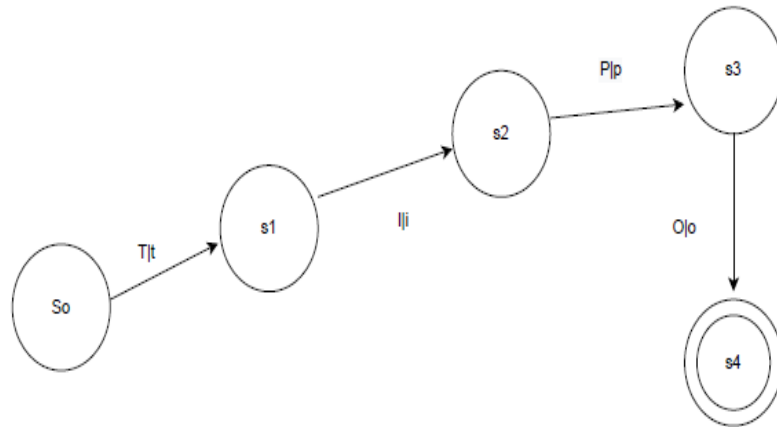
Expresión regular=(Palabra reservada, cadena, símbolo del lenguaje)*

Esta expresión regular se satisface por el siguiente automata:



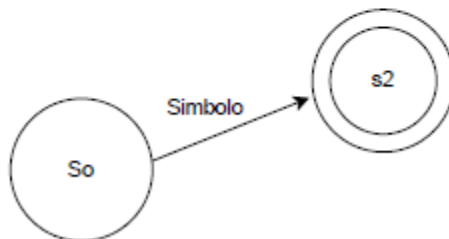
Como se puede intuir, este autómata está formado por más autómatas que se van cambiando entre estados acorde al flujo del programa, por ejemplo para formar la palabra reservada "Tipo" cada carácter es un estado y su expresión regular es la palabra en sí, veamos el autómata del estado x_3 :

Automata de la palabra reservada "Tipo"



El lenguaje es de tipo case insensitive, en el código para no formar muchas más condiciones es que por medio de verificaicones booleanas se iban a un mismo estado y así formar la palabra reservada, al final la cantidad de estados dependio de la palabra reservada más larga "Formulario", ahora veamos el automata para cualquier simbolo que se presente:

Automata de un símbolo



Por último veamos el automata de la cadena y que tiene la siguiente expresión regular:

Expresión regular="|" ([a-z]|[A-Z])* "|"

Automata de una cadena

