

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Lenguajes Formales de Programación

Sección A+

Ingeniero Otto Rodríguez

Aux. Elmer Real

16 de Noviembre de 2019, Guatemala



Manual Técnico

Nombre: Jackeline Alexandra Benitez Benitez

Carnet: 201709166

Contenido

Introducción.....	1
Objetivos y Alcances del Sistema.....	2
Diagramas de Flujo y Diagramas de Clase	¡Error! Marcador no definido.
Listado de Metodos y funciones implementadas	11
Dudas o Consultas:	11

Introducción

Es importante conocer las funcionalidades de un nuevo programa, sobretodo si no conocemos nada de el. En este caso este programa es destinado para las personas que desean traducir de un lenguaje de programacion a otro en este caso del lenguaje C# a Phyton. En la aplicación tambien se es capaz de ejecutar e interpretar el lenguaje traducido.

Objetivos y Alcances del Sistema

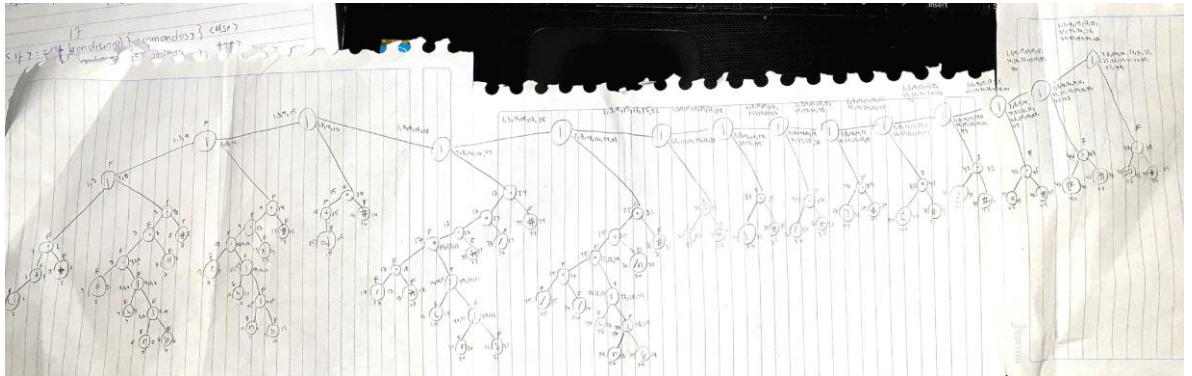
1. Facilitar la traducción de un código de un lenguaje a otro
2. Optimizar la forma de visualizar los tokens y la lista de errores de los programas
3. Organizar de mejor forma el nuevo código, con su respectiva indentación y espacios correspondientes
4. Tener una interfaz atractiva que nos ayude a visualizar todo el contenido.

Método del Arbol

Expresión regular:

$(L \# | "(L | n | Simbolo)" \# | '(L | n | Simbol)' \# | (d+) \# | /*(L | n | Simbol)*/ \# | /(L | n | Simbol) \# | (d+) \# | (simbolos) \#$

Arbol:



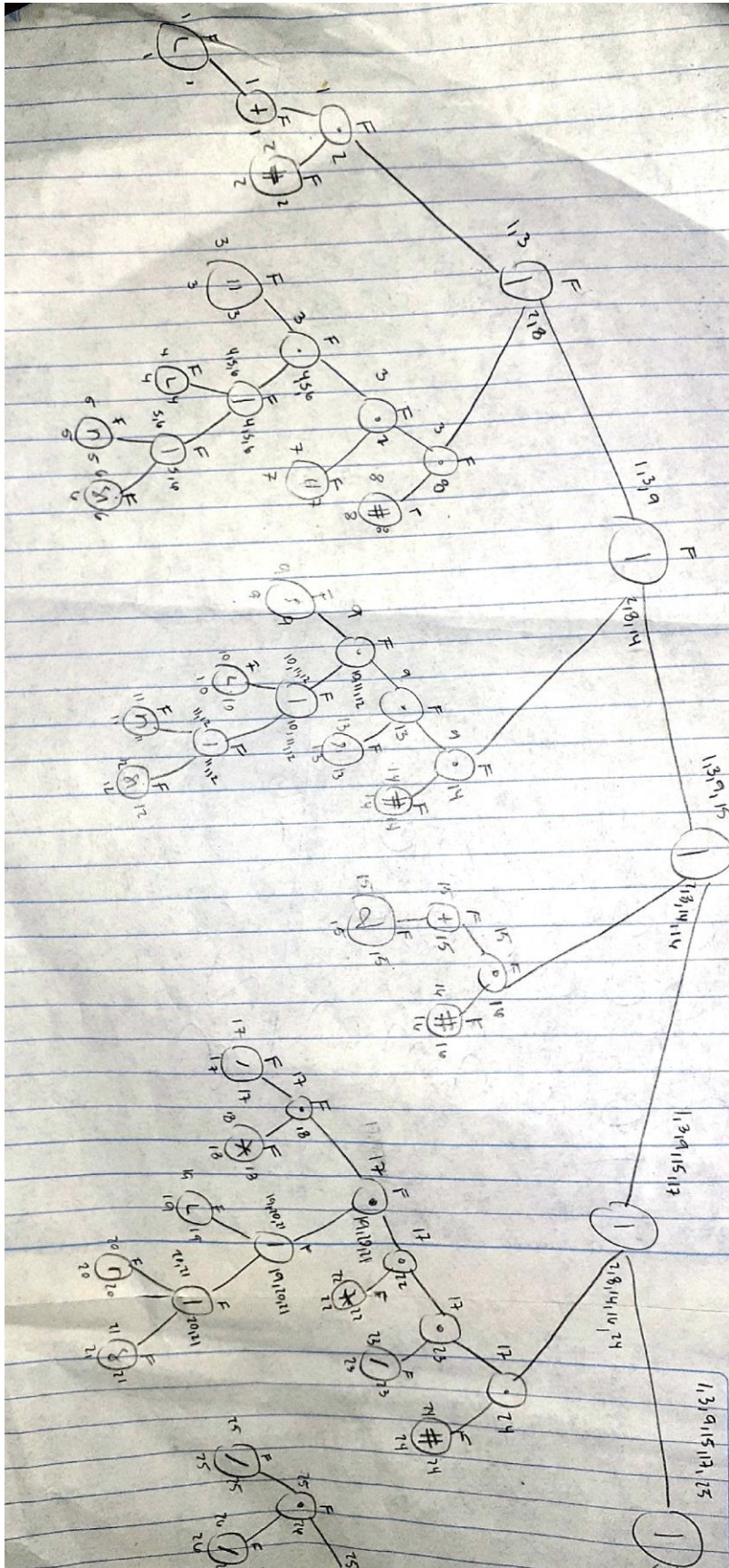




Tabla de Siguietes:

i	Siguiente(i)
1	2
2	
3	4,5,6
4	7
5	7
6	7
7	8
8	
9	10,11,12
10	13
11	13
12	13
13	14
14	
15	16
16	
17	18
18	19,20,21
19	22
20	22
21	22
22	23
23	24
24	
25	26
26	27,28,29
27	30
28	30
29	30
30	31
31	
32	33
33	
34	35
35	
36	37
37	
38	39
39	
40	41

41	
42	43
43	
44	45
45	
46	47
47	
48	49

$S_0 = \{1, 9, 20, 27, 37, 48, 56, 58, 61, 63, 65, 67, 69\}$

$S_1 = \{2\}$

$S_2 = \{3\}$

$S_3 = \{4\}$

$S_4 = \{5\}$

$S_5 = \{6\}$

$S_6 = \{7\}$

$S_7 = \{8\}$

$S_8 = \{10\}$

$S_9 = \{11\}$

$S_{10} = \{12\}$

$S_{11} = \{13\}$

$S_{12} = \{14\}$

$S_{13} = \{15\}$

$S_{14} = \{16\}$

$S_{15} = \{17\}$

$S_{16} = \{18\}$

$S_{17} = \{19\}$

$S_{18} = \{21\}$

$S_{19} = \{22\}$

$S_{20} = \{23\}$

$S_{21} = \{24\}$

$S_{22} = \{25\}$

$S_{23} = \{26\}$

$S_{24} = \{28\}$

$S_{25} = \{29\}$

$S_{26} = \{30\}$

$S_{27} = \{31\}$

$S_{28} = \{32\}$

$S_{29} = \{33\}$

$S_{30} = \{34\}$

$S_{31} = \{35\}$

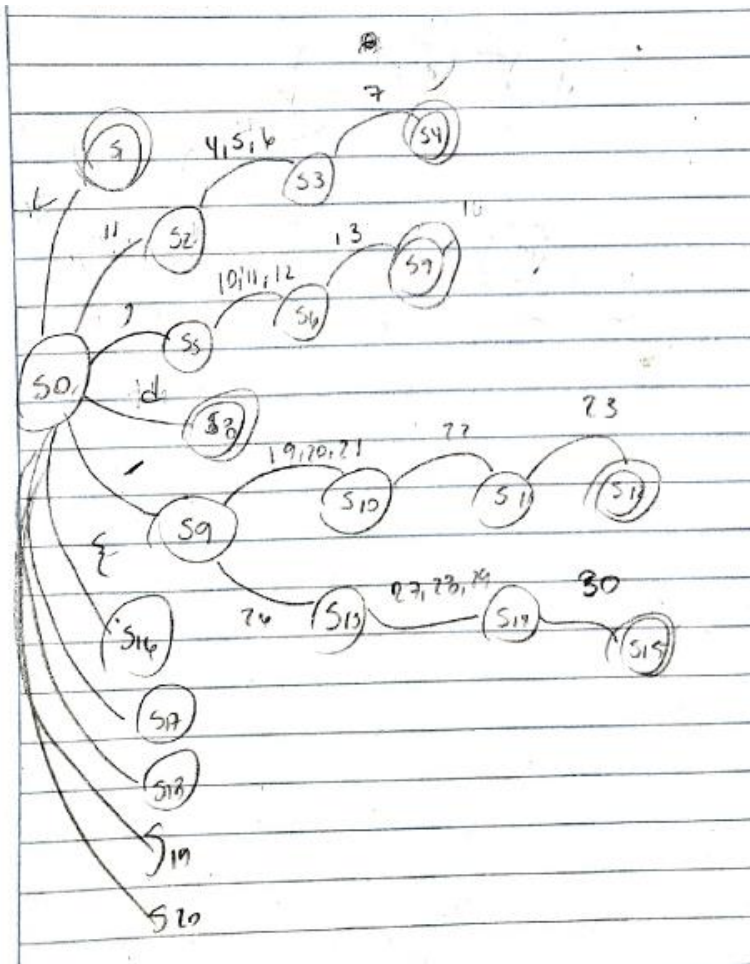
$S_{32} = \{36\}$

$S_{33} = \{38\}$

$S_{34} = \{39\}$

S35={40}
S36={41}
S37={42}
S38={43}
S39={44}
S40={45}
S41={46}
S42={47}
S43={49}
S44={50}
S45={51}
S46={52}
S47={53}
S48={54}
S49={55}
S50={57,59}
S51={60}
S52={62}
S53={64}
S54={66}
S55={68}
S56={70,71,72,73}
S57={74}

AFD Minimo



Gramatica Utilizada:

```

<inicio> := class <identificador> { <metodostatic> { <body> } }
<identificador> : identificador
<metodostatic> := static void <identificador> (<tipo>[] args)
<tipo> := int
        | bool
        | char
        | string
        | String
        | float
<body> := <comandos>
<comandos> := <declaracion> <subcomandos>
        | <asignacion> <subcomandos>
        | <graficar-vector> <subcomandos>
        | <comentarios> <subcomandos>
        | <imprimir> <subcomandos>
        | <if> <subcomandos>
        | <for> <subcomandos>
        | <while> <subcomandos>
        | <operaciones> <subcomandos>
<subcomandos> := <declaracion>; <subcomandos>
        | <asignacion>; <subcomandos>
        | <graficar-vector>; <subcomandos>
        | <comentarios>; <subcomandos>
        | <if>; <subcomandos>
        | <switch>; <subcomandos>
        | <for>; <subcomandos>
        | <while>; <subcomandos>
        | <operaciones>; <subcomandos>
        | ε

```

switch

```
<switch> := switch (<identificador>) { <cases> }  
<cases> := 'case numero : <comandos> break; <case>  
<case> := case numero : <comandos> break; <case>  
| default : <comandos> break;  
| ε
```

for

```
<for> := for (<ini>; <condicion>; <incr-decre>) { <comandos> }  
<ini> := <tipo> <identificador> = numero  
| <declaracion>
```

VER CORRECCION

```
<incr-decre> := <identificador> signo incremento  
| <identificador> signo decremento
```

while

```
<while> := while (<condicion>) { <comandos> }
```

operaciones

```
<operaciones> := <valores> <operador> <valores>
```

Listado de Metodos y funciones implementadas

- Getter&Setters: Para obtener valores especificos del Bean
- LinkedList: Se utilizaron para guardar la lista de tokens y errores
- Switch: Para analizar cada estado del automata
- AgregarError(): Servia para guardar un error a la lista
- AgregarToken(): Sirve para guardar una lista de tokens a la lista
- WriteLine: Para escribir la salida en consola
- CompareTo(): Para comparar un Chat con x elemento
- TokenHtml(): Para mostrar la lista en un reporte de html
- ErrorHTML(): Para mostrar la lista en un reporte de html
- InitializeComponent(): Para inicializar todos los componentes del form
- TreeNodo: para añadir nodos al arbol
- TabControl: para añadir nueva pestaña a mi ventana
- OpenFileDialog: para busca en el fichero un archivo
- SaveFileDialog: para guardar en el fichero un archvi
- FileName: Nombre de archivo
- ReadAllText: leer todo el archivo

Dudas o Consultas:

Jackelinebenitez112@gmail.com

56919369

System Developer