Pablo Blanco | A01637761

Diego Velázquez | A01632240

Sebastián Rojas | A01637557

Actividad Integradora 3.4 Resaltador de sintaxis

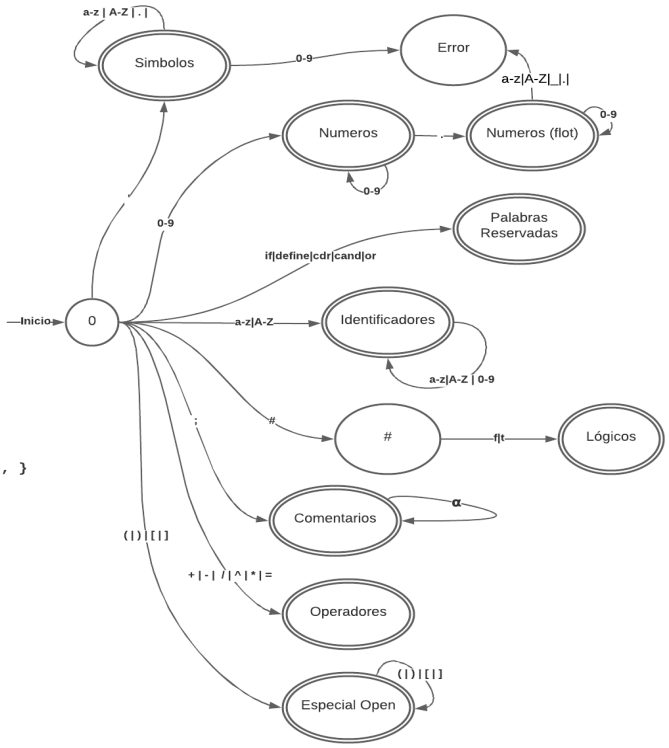
Enlace al resaltador de sintaxis: [Brucelords (tc2037.herokuapp.com)]

[Diagrama]:

Vocabulario Scheme

1. Números [#ffe119]
2. Lógicos (#t, #f) [#4363d8]
3. Símbolos ('palabra) [#469990]
4. Operadores: [#911eb4]
   - Asignación
   - Suma
   - Resta
   - Multiplicación
   - División
   - Potencia
5. Identificadores: [#3cb44b]
   - Variables
6. Especiales: [#9A6324]
   - (
   -)
7. Comentarios: [#808000]
   - ; seguido de caracteres hasta que se acabe el renglón
8. Palabras reservadas: [#f032e6]
   - define - if - cdr- cand - or
9. Errores [#e6194B]

α = {a-z, A-Z, 0-9, _, +, -, *, ^, =, /, (,), }



[Tabla]:

states	alpha-minus	alpha-mayus	number	_	+	-	*	^	(	)	#ERROR!	/	E	.	SPACE		;	
0	1	1	2	16	8	7	11	12	14	15	13	9	1	16	0	20	10	
1	1	1	1	1	0	0	0	0	0	0	0	0	1	16	0	0	0	
2	16	16	2	16	0	0	0	0	0	0	0	0	4	3	0	0	0	
3	16	16	3	16	0	0	0	0	0	0	0	0	4	16	0	0	0	
4	16	16	6	16	16	5	16	16	16	16	16	16	16	16	16	16	16	
5	16	16	6	16	16	16	16	16	16	16	16	16	16	16	16	16	16	
6	16	16	6	16	0	0	0	0	0	0	0	0	16	16	0	0	0	
7	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	16	16	16	16	0	0	0	0	0	0	0	0	16	16	0	0	0	
17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	0	0	0	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20	17	17	17	0	0	0	0	0	21	0	0	0	17	0	0	0	0	
21	21	21	21	21	21	21	21	21	21	22	21	21	21	21	21	21	0	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Pablo Blanco | A01637761

Diego Velázquez | A01632240

Sebastián Rojas | A01637557

(Se agregaron más estados para cubrir ciertos casos y uno para simular un salto de línea que no se encuentra en scheme).

Durante la realización del Resaltador de Sintaxis notamos que utilizando los conocimientos de los autómatas finitos es posible agregar las representaciones léxicas y elaborar un código que nos permita analizar un archivo de texto basado en un documento de “Scheme” resaltando con colores las palabras dependiendo de lo que representan (token).

Tomando nuestro autómata pasado como base realizamos modificaciones en la tabla de transiciones y el diagrama para que se adaptara al nuevo vocabulario basado en scheme. Ya con estas modificaciones nuestro programa lograba identificar los nuevos tokens y clasificarlos en la tabla de la entrega anterior. Posterior a esto tuvimos que alterar la forma en que se desplegaba dicha modificación a mostrarlo en un documento HTML+CSS. La modificaciòn del output al principio resultò sencilla pero nos percatamos que tenìamos que mostrar el resultado igual que el documento analizado. Esto nos llevó a tener que tomar en cuenta tokens como saltos de línea y espacios para que no los clasifique el programa pero que si los despliegue en el resultado.

Algoritmos implementado

La función lexer Aritmético lee el archivo línea por línea y dentro de un for manda a llamar la función analyze la cual lee caracter por caracter de cada línea y dependiendo de estos altera el estado y asigna tokens. Los cambios de estados se basan en la tabla de transiciones. Dentro de la función analyze se llama a la función output con la palabra y el estado que le corresponde. La función output es la que se encarga de escribir el resultado final en un archivo HTML dependiendo su estado previamente comparado y el color en que debería ser escrito, se crea un archivo de html para su correcto funcionamiento.

Tiempo de ejecución:

0.032360076904296875 segundos con archivo test.scm proporcionado por el profesor y fue calculado con la librería time hasta que terminara la ejecución del programa.

Tomando en cuenta las iteraciones que nuestro programa toma definimos que su complejidad era de O(n^2) porque en su implementación tenemos una función con un *for* que revisa cada línea del archivo y este llama una función con un *while* que va checando caracter por caracter para establecer un token. Con lo anterior establecido en caso de que se aumente el tamaño del archivo que leerá el tiempo aumentará en un O(n^2).

Pablo Blanco | A01637761

Diego Velázquez | A01632240

Sebastián Rojas | A01637557

- Video : https://youtu.be/a9ogdYH23ug