

Analizador Lexico

April 2019

El Codigo esta dividido en 3 partes

- A_lexico.cpp
- Utils.h
- Token.h

1 Tabla de Símbolos

No usa una tabla hash por el problema de colisiones. Por lo tanto se usa la estructura MAP (que viene a ser una árbol balanceado frecuentemente el RED BLACK) cuyo costo de insert delete and find (O)logn. La generación de la tabla se realiza desde codigo para no cargar nada de disco. Se usa *ascii table* para no escribir caracter por caracter simplemente se recorren con *fors*

Dentro de la tabla de símbolos como Palabras reservadas están:

- if
- while
- for
- else
- int
- char
- double
- bool
- and
- or
- =
- >

- <
- ==
- <=
- >=
- ”
- ’
- (
-)
- {
- }
- [
-]
- +
- −
- *
- /
- :
- ;
- ,

2 Token.h

Aqui solo estan las funciones para obtener identificador num o numero_real estan basados en automatas.

3 Utils.h

Aqui estan las estructuras de los datos y las funciones de formacion la unica funcion no entendible es esta bool Skip_or_Operator(int ascii) es usada para identificar si hay algun salto de linea espacio o si el caracter es un simbolo aritmetico, bool o (){}[] por ejemplo si la entrada de la linea es 1+1 de forma que cuando se llegue al caracter + la funcion retornara un bool=true esta funcion es llamada al momento de leer un archivo de entrada para poder separar palabras