



**Tecnológico
de Monterrey**

“Actividad Integradora 3.4 - Resaltador de Sintaxis ”

Implementación de Métodos Computacionales(Gpo 3)

Profesor:

Pedro Oscar Perez Murueta

Alumno:

Ricardo Andrés Cáceres Villibord A01706972

10 de junio del 2022

Para este proyecto, se ha tomado en cuenta el lenguaje de programación de C y C + +, para poder analizar los elementos léxicos del lenguaje, para así poder crear un analizador lexico en Scheme donde su objetivo sea poder resaltar la sintaxis del lenguaje. El analizador léxico tiene como principal tarea leer los caracteres de entrada de un programa fuente (en este caso un archivo de texto escrito en el lenguaje C), y producir una secuencia de tokens para cada lexema. El token consiste en el token y un atributo especial, que en nuestro caso será el tag de HTML que le corresponde al lexema.

Es por esto que se necesita primero identificar las diferentes categorías del lexema.

Las categorías identificadas fueron las siguientes:

Palabras Reservadas

namespace|if|else|for|while|do|switch|case|continue|break|int|float|double|long|string|cin|cout|char|const|void|bool|false|true|static|virtual|new|class|private|public|throw|this|catch|try|return|volatile|extern|union|sizeof|pair|printf|scanf

Operadores

+, ++, -, --, !, <, >, <<, >>, <=, >=, =, ==, &, *, /, ^, ?:, +=, -=, <<=, >>=, &=, |=, ^=, *=, /=, |, ||, &&

Identificadores

Son los nombres que se le proporcionan a las variables.

Includes

Son librerías que se incluyen en el documento para usar en el código. Están formadas de la siguiente forma: #include <LIBRERIA>

Integers y Decimals

Números enteros y flotantes

Comentarios

Oraciones en donde se empieza con //.

Para poder detectar a qué categoría pertenece cada token, se ha utilizado la librería de Expresiones Regulares que viene incluida en Racket para poder formular expresiones donde podamos identificar si el token pertenece a una categoría u otra.

El programa consiste en recibir un archivo de texto, el cual es pasado por una función que separa todo el contenido del archivo en una lista de strings. Luego esta lista es pasada a la función maestra, la cual a su vez hace la verificación de cada elemento de la lista con las funciones de las expresiones regulares. Con ayuda de la recursión, podemos ir iterando sobre nuestra lista y así verificar a qué categoría pertenece cada elemento de la lista. Si llega a hacer match con una expresión regular, significa que pertenece a cierta categoría. Por lo que a nuestra string que sería que output de HTML, se le concatena un tag en donde la clase del tag sea la categoría del token. Ejemplo: while . De esta manera en nuestro archivo CSS, podemos decir que todo lo que pertenezca a cierta clase, lo pinte de cierto color.

La complejidad de esta solución es de $O(n)$ en donde n es igual al tamaño de la lista que el algoritmo principal recibe para analizar. Eso es porque el algoritmo hace uso de la recursión para iterar sobre toda la lista. Es decir, primero analiza el primer elemento de la lista, luego hace la llamada recursiva pero ahora con el resto de la lista, y así itera hasta el final de la lista. Es decir hasta llegar a n .

Este proyecto tiene varias implicaciones éticas, y es una tecnología que puede llegar a ser de mucha utilidad para la sociedad. Una de las utilidades es para las personas que no tienen las capacidades visuales adecuadas. Al poder resaltar de diferentes colores los diferentes lexemas del lenguaje de programación, estas personas pueden identificar y diferenciar más rápido a qué categoría pertenece cada token. Y de la misma manera, esta es una solución muy versátil ya que los colores se pueden ajustar para las necesidades de cada persona.

Link del Repositorio de Github:

<https://github.com/Caceres-A01706972/TC2037/tree/main/Situación%20Problema/Resaltador%20de%20Sintaxis%201>