Valeria Conde A01281637

Resaltador sintáctico

Para realizar el analizador sintáctico, necesitamos utilizar un analizador léxico, en este caso el que utilizamos en la actividad anterior que proveía los tokens que después fueron procesados por el analizador sintáctico, de manera que podamos aseguramos de que el orden de las operaciones sea válido. Además, reconociendo el tipo de token que se provee como input podemos realizar un resaltador que marque de cierto color dependiendo del tipo que se leyó.

Para realizarlo, primero construimos una matriz que contiene en valores enteros una representación de los estados del autómata que construimos. Además, al construirla también hacemos un mapeo de dichos valores enteros a lo que representan en valor de tipo string, para poder identificarlos posteriormente.

Después el programa recibe un archivo de entrada que contiene los tokens a analizar y por cada línea en el archivo lee cada caracter y lo busca en la matriz de estados para identificarlo. Antes de realizar el parser se debe establecer la gramática que se respetará, la cual en establecimos como:

assignation: varID = EXP
EXP: term = term((+|-) term)*
term: factor((*|/|^)factor)*
factor: "(" EXP ")" | (+|-)? constant
constant: varID | int | real

Y significa que tomará todo como asignación o como comentario. Una expresión como a+b no sería válida, al no ser una asignación.

Finalmente, el programa llama a la función paser() la cual crea un documento HTML con estilos CSS para hacer un resaltador sintáctico. Esta función lee cada token (que fue guardado durante la ejecución del lexer en un arreglo) y llama a la función assignation(), la cual identifica si se trata de una variable y en su caso la resalta como tal y llama a la función match() que matchea el token con la gramática y avanza la variable que tenemos un apuntador para iterar el arreglo donde guardamos los tokens. Al tratarse de una asignación, la función assignation() espera una expresión por lo que finalmente llama a la función exp(), la cual llama a la función term() al esperar un término, y resalta y matchea si se trata de una suma o resta. La función term() manda llamar la función factor() y a su vez resalta y matchea si se trata de una multiplicación, división o potencia, utilizando jerarquía de operaciones. La función factor() resalta y matchea si se trata de un operador, y llama a la función constant() que resalta y matchea un token de variable, entero, flotante o potencia.

Se trata de un algoritmo de complejidad O(n), es decir, lineal, ya que cada token que lee del archivo de entrada se visita una sola vez para el cómputo del resultado.

Los parsers son tecnología que abrió paso al desarrollo de muchas tecnologías, al ser una parte fundamental de un compilador, el cual se utiliza para procesar lenguajes de programación, con los cuales se desarrollan muchas tecnologías potentes. Por ello, como todo, hay implicaciones éticas en el desarrollo de ellas, por lo que tenemos una responsabilidad social y siempre hay que vigilar el desarrollo y uso ético de las herramientas de trabajo que tenemos como ingenieros de software, ya que podemos hacer uso tanto adecuado como malicioso de la tecnología y mucho más cuando se trata de desarrollo.

Daniela Garza A00829404

Los algoritmos planteados fueron los siguientes: hicimos una función llamada filter que lee un caracter ‘c’ que regresa las columnas de la matriz, hicimos una función match la cual lee un token el cual es un string en el que si hay un error sale de output “error parsing”, una función la cual lee el archivo línea por línea y hace algo con el string, guarda el token y el tipo en vectores y marca el final de una línea, la función de la matriz, una función que lee la gramática y les hace match con funciones y varias funciones que leen el tipo de token, y las regresa, le dan estilo al html y se hace el output del código en un archivo html.
El código corre rápido, por lo que la complejidad del sistema es O(n).

Las implicaciones éticas que el tipo de tecnología desarrollada podría tener en la sociedad son que se puede obtener información más rápido, por ejemplo en sistema IA y/o en la robótica. Por ello, se deben de tener valores de tolerancia y respeto por el bienestar de los seres humanos al ser incorporados como elementos de la programación de agentes de inteligencia artificial.