



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Organización de Lenguajes y Compiladores 1
Catedrático: Ing. Kevin Adiel Lajpop Ajpacaja
Tutor académico: Moises Gonzalez Fuentes

SECCION C

Segundo semestre 2022

MANUAL TÉCNICO

Proyecto 1

Pseudo-Parser

Nombre: Moises David Maldonado de León

Carné: 202010833

Guatemala, 17 de septiembre de 2022

Introducción

En el presente documento, se describe la estructura y los demás aspectos técnicos del Psedo-parser creado que implementa un analizador léxico y sintáctico para la obtención de información y su respectiva traducción a los lenguajes de Golang y Python. Se conocerá no solo el detalle de estructura del programa sino también el proceso previo, es decir, las expresiones regulares utilizadas en cuanto a la parte del análisis léxico y también la gramática utilizada definida e implementada para el análisis sintáctico. También se conocerá la lógica para la implementación del árbol sintáctico.

OBJETIVOS

Generales

Familiarizar al lector con la lógica planteada para el funcionamiento del sistema realizado mediante el uso del lenguaje de programación JAVA y aplicando las herramientas de Jflex y Cup, para la parte del análisis léxico y para la creación del análisis sintáctico respectivamente así como la estructura de datos para el árbol sintáctico.

Específicos

- Mostrar la estructura del programa realizado.
- Dar a conocer con detalle los métodos y variables empleadas, así como su funcionalidad dentro del programa.
- Mostrar la tabla de tokens, expresiones regulares.
- Mostrar la gramática creada e implementada para la correcta ejecución del programa.

Alcances

El programa se realizó con el fin de funcionar como traductor de código mediante una entrada de pseudocódigo. Además, fue creado con el propósito de crear un entorno agradable donde el usuario puede ingresar interactuar y posteriormente verificar si existen errores en las entradas y a su vez, con una sola acción visualizar la información deseada.

Especificaciones

Requisitos del Hardware

- Mínimo 2GB de memoria RAM
- Procesador Intel Core 2 Duo 2 o superior
- Espacio en disco duro: Mínimo 50 MB

Requisitos del Software

- El programa fue realizado en Java, específicamente con la versión 1.8
- El IDE utilizado fue NetBeans, versión 8.2
- Funcional en sistemas operativos como Windows XP, 7,8,10 o MAC OS
- Tener instaladas o con acceso a las siguientes librerías y paquetes: java-cup-11b.jar, java-cup-11b-runtime.jar, jflex-full-1.7.0.jar y Graphviz.

Lógica del programa

Descripción de Clases	
Clase	Descripción
A_sintactico()	Clase generada mediante la herramienta CUP, maneja toda la parte del análisis sintáctico, así como la inserción de nodos en el árbol sintáctico.
Analizador_Lexico()	Clase generada mediante la herramienta JFLEX, maneja toda la parte del análisis léxico.
Símbolos()	Clase generada mediante la herramienta JFLEX, maneja la parte de los símbolos (tokens) definidos para el análisis léxico.
claseErrores()	Clase que maneja tipo de dato abstracto para el guardado de errores léxicos y sintácticos.
Nodo()	Clase nodo, almacena los nodos del árbol sintáctico
arbolSintactico()	Clase que maneja la creación del árbol sintáctico, se implementó la estructura de datos recursiva de un árbol n-ario.
Proyecto_OLC1 ()	Clase main, es la principal e inicializa la interfaz gráfica.

Archivos extras e interfaz gráfica

lexico.jflex	<p>Archivo que maneja la creación de Símbolos y reconocimiento de tokens. Para crear la clase java a partir de este archivo se debe ejecutar el siguiente comando en la consola con cada cambio:</p> <pre>java -jar jflex-full-1.7.0.jar lexico.jflex</pre>
sintactico.cup	<p>Para crear la clase java a partir de este archivo se debe ejecutar el siguiente comando en la consola con cada cambio:</p> <pre>java -jar java-cup-11b.jar -parser A_sintactico -symbols Simbolos sintactico.cup</pre>
ReporteErrores.java	<p>Archivo JFrame Form, maneja la clase secundaria de la interfaz gráfica. Despliega la ventana que muestra la tabla de errores.</p>
Interfaz.java	<p>Archivo JFrame Form, maneja la clase principal de la interfaz gráfica.</p>

Descripción de las funciones/métodos		
Clase	Método	Descripción
claseErrores()	claseErrores	Constructor de la clase, inicializa las variables
	getLexema(), getTipo() getDescripcion(), getFila() getColumna()	Métodos que retornan los datos respectivos de la clase.
Nodo()	Nodo()	Constructor de la clase, inicializa las variables
	Redimensionar()	Método que redimensiona los hijos de un nodo
	aumentarHijo()	Método que agrega un hijo al nodo
	getValor()	Método que retorna el valor del nodo
	setValor()	Método para cambiar el valor del nodo
	verNodo()	Método que imprime en consola el dato del nodo y sus hijos

Descripción de las funciones/métodos		
Clase	Método	Descripción
Sintactico()	insertarRaiz()	Función de tipo Nodo que permite la inserción de la raíz del árbol y retorna el mismo.
	getRaiz()	Función de tipo Nodo que retorna el nodo raíz.
	verHijosRecursoivo()	Método recursivo que muestra los hijos de un nodo. Para ver todo el árbol se le envía como parámetro el nodo raíz.
	InsertarRecursoivo()	Método recursivo para insertar un nodo al árbol.
	VerificarExistencia()	Función de tipo booleano que recorre el árbol, retorna falso si no se encuentra el nodo y en caso contrario verdadero
	Graficar()	Método principal que crea el archivo .dot , llama al método generarNodos() y generarArchivoDot()
	generarNodos()	Función de tipo string que genera una cadena de texto con la creación de los nodos y sus apuntadores del archivo .dot. Retorna la cadena generada.
	generarArchivoDot()	Método que crea el archivo .dot
	crearImagen()	Método que ejecuta un comando en la consola que permite la ejecución de un script de Python para la generación de la imagen del árbol sintáctico.
	def tempEquipo()	Genera un HTML con los partidos disputados por un equipo en una temporada específica.
	def topCSV()	Retorna el top de mejores o peores equipos clasificados.

Flujo del programa

Análisis Sintáctico

El archivo de gramática se encuentra en la carpeta de Documentación como Gramática.txt

Repositorio: <https://github.com/Hrafnyr/OLC1-202010833>