Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Estructuras De Datos, 1er. Semestre 2022.

PROYECTO 1 - CONSTRUCTOR DE ANÁLISIS LÉXICO Y SINTÁCTICO

202000119 – Sergie Daniel Arizandieta Yol

Resumen

Como el titulo lo describe el proyecto se enfoca en análisis léxico y sintáctico, siendo más precisos para expresiones regulares y sus validaciones de cadenas para las mismas mediante un autómata finito determinista.

Dicho software fue desarrollado en JAVA implementando Jflex y Cup para el análisis del archivo de entrada. Así mismo generando archivos de salida sobre las cadenas y métodos utilizados para la generación del AFD mediante Graphviz, implementando el método del árbol y Thomson.

Dicho proyecto hace utilización de estas tecnologías de una manera novedosa para la búsqueda, lecturas y escrituras optimas de los datos a gestionar para la Constructor de análisis léxico y sintáctico requerido en el 2022 (Se recalca que el proyecto no es una fase final).

Palabras clave

JAVA, JFLEX, CUP, AFD, AFND

Abstract

As the title describes, the project focuses on lexical and syntactic analysis, being more precise for regular expressions and their string validations for them by means of a deterministic finite automaton.

Said software was developed in JAVA implementing Jflex and Cup for the analysis of the input file. Likewise, generating output files on the strings and methods used to generate the AFD using Graphviz, implementing the tree and Thomson method.

This project makes use of these technologies in a novel way for the optimal search, reading and writing of the data to be managed for the Lexical and Syntactic Analysis Builder required in 2022 (It is emphasized that the project is not a final phase).

Keywords

JAVA, JFLEX, CUP, AFD, AFND

Introducción

Los analizadores léxico y sintácticos no son los únicos manejados en un compilador, pero para el enfoque del proyecto se utilizaron estos para la generación de autómatas finitos deterministas y no deterministas, mediante el método del árbol y Thomson respectivamente, así pudiendo comprobar una serie de cadenas propuestas con las expresiones regulares definidas.

Así mismo la salida de las cadenas se verá reflejado en archivo JSON, teniendo como salidas también grafos generados mediante la herramienta Graphviz sobre cada uno de los pasos del método del árbol, AFD y AFND.

Teniendo como métodos para el almacenamiento de datos listas TDA generadas por el programador y algunas LinkedList de JAVA.

Desarrollo del tema

Las tecnologías requeridas y fundamentales para el desarrollo del software son:

- Implementación de TDA
  - Listas
  - Manipulación de las tuplas de las TDA
- Archivos Json
  - Definición de los archivos Json
- Generación de grafos mediante Graphviz
- Método del árbol
- Método de Thomson

Conceptos generales

- Tipo de dato abstracto (TDA)

  Son el nivel más alto de abstracción y son independientes del lenguaje de programación, básicamente son una colección de valores y de operaciones que se definen mediante una especificación en el presente proyecto implementado en listas dinámicas.

- TDA Lista:

  Se define como una tupla de n elementos los cuales son asignados dinámicamente como se requiera, donde se pueden crear operaciones para cada una de las tuplas como:

  - Obtener datos
  - Mostrar
  - Eliminar
  - Agregar
  - Otros

- Manipulación de las tuplas de una lista TDA

  Es muy similar a la POO (Programación Orientada Objetos), ya que cada tupla cuenta con atributos específicos, los cuales se pueden hacer referencia gracias a que son definidos en su constructor el cual se encarga de definir los atributos de las tuplas cuando llamadas para crear una instancia de estas.

- Archivos JSON

  JSON sus siglas en ingles son “JavaScript Object Notation” el cual traducido seria “Notación de Objetos de JavaScript”. Se utiliza para estructurar datos en forma de texto y permite el intercambio

de información entre aplicaciones de manera sencilla, liviana y rápida. Es una alternativa al XML.

- Grafos en Graphviz:
  Graphviz es un software de visualizador de gráficos de código abierto, el cual implementa su extensión DOT para sus grafos.

  - DOT

    Es un lenguaje descriptivo de texto plano el cual es una forma simple de describir los grafos.

- Método del árbol:
  Es uno de los métodos existentes para la generación de Autómatas Finitos Deterministas, a partir de una expresión regular el cual posee una serie de pasos los cuales son:

  - Agregar simbolo de finalización

    Se le agrega un simbolo de finalización a la expresión regular
  - Construir árbol sintáctico

    Se genera el árbol binario asociado a la expresión regular enumerando sus hojas.
  - Calcular Anulables

    Se define que nodos son anulables o no, basado en reglas estipuladas.
  - Calcular primeros

    Se calcula los primero de cada nodo basado en reglas estipuladas.
  - Calcular Ultimo

    Se calcula los últimos de cada nodo basado en reglas estipuladas.
  - Calcular los siguientes

    Se calcula los siguientes de cada hoja basado en reglas estipuladas.
  - Tabla de transiciones

    Se genera una tabla de transición de estados basados en reglas estipuladas
  - Generar AFD

    Se genera el autómata finito determinaste a partir de la tabla de transiciones

  Para tener las reglas estipuladas al método se puede referir a: https://www.youtube.com/watch?v=Re-s4DgPKaU - Autor: Kevin Lajpop, 25 de feb. del 2022. Ingeniero de la Escuela de Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala

- Método de Thomson:
  Es uno de los métodos existentes para la generación de Autómatas Finitos No Deterministas, a partir de una expresión regular el cual solo consiste en la construcción del AFND respetando los operadores *,|,..

  Para tener las reglas estipuladas al método se puede referir a: https://www.youtube.com/watch?v=IfqTXVD4qQ4 - Autor: Kevin Lajpop, 27 se ago. del 2020. Ingeniero de la Escuela de Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala

Representación de los módulos del programa

Tabla I.

Módulos del proyecto.

Clases
Main
Analizador_Lexico
Analizador_sintactico
errorList
SimpleER
Cadenas
SimpleCalcSiguientes
SimpleSiguientesTransiciones
AFND
Automata
CrearArchivo
Reportes

Descripción de módulos:

- Main
  Se encarga solo de la instancia de la clase de general para comenzar todo el gestiona miento y donde se almacena todo el diseño del software.

- **Analizador_Lexico**
  Archivo generado mediante Jflex en el cual se define el lenguaje del analizador.

- **Analizador_sintactico**
  Archivo generado mediante Cup en el cual se define una gramática utilizando el lenguaje y no terminales aceptados en el Analizador_Lexico que definirá al analizador.

- **errorList**

  Es una objeto que se utilizó para la creación de una LinkedList de java para el almacenamiento de errores del analizador,

- **SimpleER**

  Es una lista simplemente enlazada donde se almacena cada dato de la expresión regular para luego agregar cada una de estad en una LinkedList de java para el almacenamiento de las expresiones regulares.

- **Cadenas**

  Es un objeto que almacena la cadena a evaluar junto su expresión regular para comprobar si esta es correcta o no.

- **SimpleCalcSiguientes**

  Esta es una lista simple enlazada para la creación de los siguientes aceptados en cada hoja.

- **SimpleSiguientesTransiciones**

  Es una lista simplemente enlazada para las tradiciones de los autómatas finitos deterministas, simplemente almacena los siguientes de cada estado.

- **AFND**

  Es clase que implementa una pila para la generación del autómata finito no determinista guardando el orden de los operadores.

- **Automata**

  Es un objeto que básicamente los nodos del autómata.

- **CrearArchivo**

  Es una clase con el propósito de generar los autómatas mediante Graphviz y generar el grafo a png.

- **Reportes**

  Es una clase que genera el archivo HTML de errores.

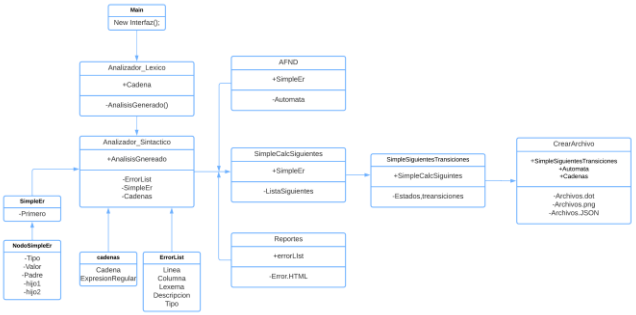


Figura 1. Diagrama de Clases.

Fuente: elaboración propia.

Conclusiones

El uso de nuevas tecnologías, implementaciones inclusive librerías que puedan hacer el trabajo de desarrollo es vital para una empresa viéndolo a gran escala, como son las TDA lista, ya que la mayoría de las veces el cliente siempre requerirá que se tenga el menor estrés en sus computadoras al ejecutar los softwares necesarios.

Estas combinadas con los analizadores utilizados en este proyecto abren a un gran campo para no solo la generación de autómatas surgidos de una expresión regular, por lo cual teniendo en cuenta que hay muchos mas analizados y tipo de datos TDA ¿Qué otro tipo de herramientas se te ocurren a ti?

Referencias bibliográficas

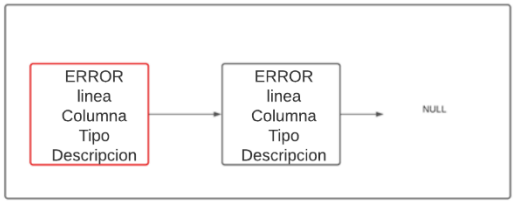
Arias Guerra, D. (2008) Estructura de datos Avanzadas (Revisado, ed., Vol. 9). Universidad de las Ciencias Informáticas. https://cutt.ly/eWdkzvt

Ellson, J. (2005) Graphviz. Graphviz org. https://graphviz.org/doc/info/lang.html

Ferris Castell, R. (2004) Algoritmos y Estructura de Datos I (Revisado ed., Vol. 1) Universidad de Valencia http://informatica.uv.es/iiguia/AED/oldwww/2001_02/Teoria/Tema_10.pdf

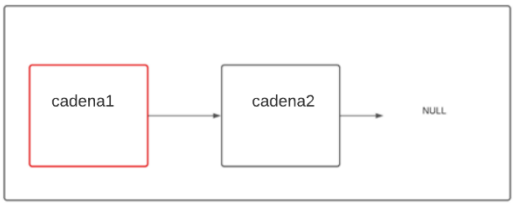
Anexos

Modelo “errorList”



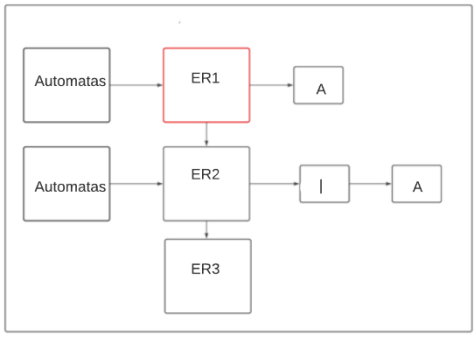
Fuente: elaboración propia.

Modelo “Cadenas”



Fuente: elaboración propia.

Modelo “SimpleER”



Fuente: elaboración propia.