**UNIVERSIDAD NACIONAL DE ASUNCION FACULTAD POLITECNICA**

*Ingeniería en Informática*

**Asignatura:** Diseño de Compiladores

**Profesor:** Ing. Sergio A. Aranda

*“Convertidor”*

**Integrantes:**

* Melisa María Rosa Villamayor Paredes
* Luis Víctor Maidana Benítez

**Semestre:** 9no

Año 2018

**Introducción**

El objetivo de este trabajo es implementar un analizador léxico, donde se ingresa una expresión regular, esta debe ser procesada a través del algoritmo de Thompson para obtener el AFN correspondiente y luego con el algoritmo de subconjuntos para obtener el AFD, además se debe minimizar este AFD si es posible y luego introducir un lexema para escanear si este pertenece o no al lenguaje definido.

**Descripción**

El software Conversor está realizado en lenguaje Java, sobre el IDE Netbeans.

El programa cuenta con 4 paquetes, de los cuales a continuación resaltamos y explicamos las partes más importantes.

1. **algoritmos:** allí están los algoritmos de Thompson, Subconjuntos, validación, etc. que son necesarios para la creación de los autómatas.
2. **preparacion:** allí están definidas las estructuras necesarias para los datos del sistema.
3. **representacion:** están las validaciones del alfabeto, el analizador sintáctico y el léxico, además de tokens y expresiones regulares.
4. **principal:** aquí está el main del proyecto.
5. **Tabla de tokens/lexemas/patrones:**

|  |  |  |
| --- | --- | --- |
| **Token** | **Patrón** | **Lexema** |
| PARENTESIS\_DERECHO |  | ( |
| PARENTESIS\_IZQUIERDO |  | ) |
| UNION |  | | |
| CERRADURA\_KLEENE |  | \* |
| CONCATENACION |  | Ningún símbolo |
| ALFABETO |  | Cualquier símbolo del alfabeto. |
| FINAL |  | Finalizador de una ER |
| DESCONOCIDO |  | Token desconocido (invalido) |

1. **Representación de ER de cada patrón**
2. **Proceso y Resultado de aplicar el Algoritmo de Thompson**
3. **Proceso y Resultado de aplicar el Algoritmo de Algoritmo de Subconjunto**
4. **Proceso y Resultado de aplicar el Algoritmo de Algoritmo de minimización**
5. **Código fuente del programa**

**Representación**

Las expresiones soportan los siguientes operadores:

* Unión ( | )
* Agrupaciones (a través de paréntesis)
* Concatenación (sin carácter representativo)
* Cerradura de Kleene ( \* )

Explicar A. Lexico

Explicar A Sintactico

Explicar Thompson

Explicar Subconjuntos

Explicar minimización

Explicar Validacion

1. **Adicionales**

El alfabeto es personalizado, lo define el usuario.

Además de los operadores mencionados anteriormente, también soporta + y ?.

Para mayor comprensión la salida imprime todos los cálculos de todos los procesos de algoritmos utilizados.

En las tablas se pueden apreciar los estados en grupos.

El código está debidamente documentado.