LENGUAJES FORMALES

|  |  |
| --- | --- |
| Integrantes: | Fecha |

Proyecto final

Tabla de contenido

[Parte I. Investigación. 1](#_Toc89157465)

[Parte II. Sintaxix lenguajes de alto nivel 1](#_Toc89157466)

[Parte III. Caso de estudio 1](#_Toc89157467)

[Especificaciones 2](#_Toc89157468)

[Entrada: 2](#_Toc89157469)

[Salida: 2](#_Toc89157470)

[Reporte de resultados 3](#_Toc89157471)

[Paso 1. 3](#_Toc89157472)

[Paso 2. 3](#_Toc89157473)

[Paso 3. 3](#_Toc89157474)

[Paso 4. 3](#_Toc89157475)

[Conclusiones. 4](#_Toc89157476)

[Bibliografía. 4](#_Toc89157477)

## Parte I. Investigación aplicaciones.

Describe diversas aplicaciones de autómatas, gramáticas libres de contexto , expresiones regulares, autómatas de pila, máquinas de Turing. Recuerda citar tus referencias.

## Parte II. Sintaxix lenguajes de alto nivel

[En esta sección escoge 3 lenguajes de programación y anota su gramática libre de contexto]

## Parte III. Caso de estudio

Objetivo

Resolver un problema que requiera el reconocimiento de patrones de texto y estructuras gramaticales , utilizando software de generación de analizadores léxicos y sintácticos.

Enunciado del problema

Crear un programa, utilizando el generador de analizadores léxicos JFlex y el de analizadores sintácticos CUP, que transforme una expresión regular a notación postfija.

## Especificaciones

La sintaxis para definir las expresiones regulares será de la siguiente manera:

Una ER válida será válida de la siguiente forma:

Base:

La cadena vacía ε , representada como $

Cualquier símbolo del alfabeto mayuscula o minúsculula y cualquier dígito es una ER.

Inductivo

1. Si E y F son expresiones regulares, entonces E,F también lo es y representa la unión de L(E)∪L(F).
2. Si E y F son expresiones regulares, entonces EF también lo es y representa la concatenación de L(E)L(F). (Para imprimirlo como operador en la expresión postfija utilizaremos el punto)
3. Si E es una expresión regular , entonces E∗ también lo es y representa L(E)∗ .
4. Si E es una expresión regular , entonces (E), también lo es.

### Entrada:

Serán un listado de expresiones regulares separados por un punto y coma.

a\*;

a , b c ;

a,(bca,a)\*;

a,(bca,a)\*\*;

a,(bca,a)\*,K\*;

### Salida:

Transformación a notación postfija.

a \*

a b c . ,

a b c . a . a , \* ,

a b c . a . a , \* \* ,

a b c . a . a , \* , K \* ,

## Reporte de resultados

### Paso 1.

Poner en funcionamiento las herramientas de JFLEX y CUP con el ejemplo de una calculadora de números enteros.

a. Insertar captura de pantalla, con una entrada distinta a la asignada en el ejemplo de los archivos descargados.

### Paso 2.

Modificar la calculadora de números enteros para generar en notación postfija cada una de las expresiones aritméticas en el archivo input.txt

b. Inserta una captura de pantalla con la misma entrada en el inciso a.

### Paso 3.

Genera la gramática para generar expresiones regulares, recuerda que se comporta de manera muy similar a la de aritmética, sólo hay que considerar los nuevos operadores y su jerarquía.

c. Inserta la gramática nueva con formato solicitado en la página , adicional , muestra capturas de pantalla en este inciso con los resultados que muestra la página sobre tu gramática:

<https://mdaines.github.io/grammophone/>

### Paso 4.

Modifica los archivos input.txt, ycalc.cup, lcalc.flex para que ahora generan la notación postfija de la expresión regular.

d. Lcalc.flex modificado

e. Ycalc.cup modificado

f. Input.txt (ejemplos necesarios, anota más ejemplos conforme a tus pruebas)

a\*;

a , b c ;

a,(bca,a)\*;

a,(bca,a)\*\*;

a,(bca,a)\*,K\*;

g. Captura de pantalla que genere la notación postfija de las expresiones regulares en input.txt

## Conclusiones.

[Anota los el resumen el analisis de los resultados que obtuviste, trabajo futuro, recomendaciones ]

## Bibliografía.

Para las tareas en las que consultan referencias externas a las notas de la clase, inclúyanlas en el documento (libros, notas, páginas web, videos, etc. ). Deben estar citadas de acuerdo al estandar IEEE.