

Proyecto No. 1

Instrucciones generales:

- Este proyecto es en **parejas**. Deberá inscribirse en un grupo de Canvas asociado a esta entrega.
- Cree un repositorio **privado** en GitHub.com para alojar el código correspondiente para este proyecto.
- Añada los usuarios de GitHub de su catedrático y de su(s) auxiliar(es) con rol de *collaborator*. Estos se encuentran en la sección de información del curso en Canvas.
- Coloque todas las instrucciones que considere pertinentes en un archivo denominado README.md en la raíz del repositorio.

Descripción

Este proyecto es una recopilación del trabajo efectuado en laboratorios anteriores y agrega nuevos elementos para terminar de completar la parte inicial de un analizador léxico, que se dedica a verificar la aceptación de cadenas en lenguajes regulares.

El proyecto consiste en la implementación de los algoritmos básicos para construcción de autómatas finitos a partir de expresiones regulares. El programa aceptará como entrada una expresión regular r y una cadena w . A partir de la expresión regular r se transformará su notación infix a postfix y luego se construirá un AFN; posteriormente, se transformará el AFN a AFD. Luego, este AFD deberá minimizarse. Con dichos autómatas (AFN y AFD) se determinará si la cadena w pertenece o no a $L(r)$.

Objetivos

- **Generales**
 - Implementación de algoritmos básicos de autómatas finitos y expresiones (lenguajes) regulares.
- **Específicos**
 - Implementación del **Algoritmo de Shunting Yard de conversión infix a postfix**.
 - Implementación del **Algoritmo de Construcción de Thompson**.
 - Implementación del **Algoritmo de Construcción de Subconjuntos**.
 - Implementación del **Algoritmo para minimización de AFD's**.
 - Implementación de simulación de AFN.
 - Implementación de simulación de AFD.

Especificación del funcionamiento del programa

- **Entrada**
 - Una expresión regular r .
 - Una cadena w .
 - Por ejemplo, se ingresa la expresión regular $r = (b|b)^*abb(a|b)^*$ y la cadena $w = babbaaaaa$.
 - El símbolo que represente a ϵ será designado por el programador o programadora (debe ser algo razonable, no una letra o un número con altas probabilidades de ser usado en otro aspecto del proyecto).

- Todos los símbolos del alfabeto son de longitud 1. El alfabeto de símbolos para la expresión regular será conformado por todos los símbolos (no operadores) distintos que se encuentren en la expresión regular.
- Salida
 - Por cada AF generado a partir de r:
 - Una imagen con el Grafo correspondiente para el AFN y para los AFDs generados (por construcción de subconjuntos y su minimización), mostrando el estado inicial, los estados adicionales, el estado de aceptación y las transiciones con sus símbolos correspondientes.
 - La simulación del AFN y de los AFDs al colocar la cadena w: el programa debe indicar si $w \in L(r)$ con un "sí" en caso el enunciado anterior sea correcto, de lo contrario deberá mostrar un "no".
- Su programa debe leer un archivo de texto y procesar cada línea en este archivo. Cada línea deberá de tener una expresión regular. Estas serán brindadas por su calificador al momento de presentar su proyecto.

Ponderación

Característica	Ponderación
Shunting Yard para transformar de infix a postfix	3 puntos
Generación de AFN con Thompson	3 puntos
Generación de AFD con Subconjuntos	3 puntos
Minimización de AFD	3 puntos
Simulación de AFN y AFDs	3 puntos
Total	15 puntos