

Proyecto 1 (propuesto originalmente por Bidkar Pojoy)

Descripción

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 construirá un AFN, se transformará el AFN a AFD; y se generará un AFD directamente. Ambos AFD's deberán minimizarse. Con dichos autómatas se determinará si la cadena w pertenece o no a $L(r)$.

Este proyecto podrá ser desarrollado de manera individual, en parejas o, como máximo, en tríos. Una única persona del grupo deberá subir los entregables a Canvas, pero todas las personas presentarán sus resultados. Para facilitar la calificación, es requerido cargar todo el código entregable directamente a Canvas.

Objetivos

- **Generales**
 - Implementación de los algoritmos básicos de autómatas finitos y expresiones regulares.
- **Específicos**
 - Implementación de algoritmo de **Construcción de Thompson**.
 - Implementación de algoritmo de **Construcción de Subconjuntos**.
 - Implementación de algoritmo de **Construcción de AFD a partir una expresión regular r** .
 - Implementación de un algoritmo para **minimización de AFD's**.
 - Implementación de simulación de un AFN.
 - Implementación de simulación de un AFD.

Especificación

• Entrada

Solamente se ingresarán textualmente una expresión regular r y 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

- Para cada autómata generado, un **SÍ** si la cadena w pertenece al lenguaje descrito por el autómata ($L(r)$) o un **NO** en caso contrario. También, para cada autómata, el tiempo que se tarda en realizar dicha validación y las transiciones realizadas durante la validación.
- Para cada autómata generado se debe escribir un archivo de texto que describa al autómata con la siguiente estructura:

```
ESTADOS = {0, 1, ... n}
SIMBOLOS = {a, b, c, ... z}
INICIO = {0}
ACEPTACION = {0, 1, ... n}
TRANSICIONES = (0, a, 1) - (0,  $\epsilon$ , 2) - ... (3, b, n)
```

Ponderación

El proyecto en total tiene un valor de **20 puntos** netos, distribuidos de la siguiente forma:

Característica	Ponderación
Transformación de expresión regular a <i>postfix</i>	2
Generación de AFN (archivo)	3
Conversión AFN a AFD (archivo)	4
Generación de AFD Directo (archivo)	4
Minimización de AFD's (archivo)	4
Simulación AFN y AFD	3
Total	20

Debe entregarse documentación para tener derecho a nota. La documentación debe de abarcar:

- Diseño de la aplicación (el método de modelado a discreción del desarrollador o desarrolladora).
- Discusión (obstáculos encontrados, recomendaciones, etc.).
- Ejemplos y pruebas realizadas. **Importante:** se requerirá realizar las mismas pruebas de simulación sobre cada autómata generado. Es decir que, para una cadena w dada y una expresión regular r dada, se deberá efectuar la simulación sobre el AFN generado a partir de r , el AFD construido directamente a partir de r , el AFD obtenido por construcción de subconjuntos a partir del AFN de r ; y los AFD's mínimos obtenidos a partir del AFD construido directamente y el AFD obtenido por construcción de subconjuntos.