

Teoría de la Computación 2024

Primer Proyecto

14.agosto.2024

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. Este AFD deberá minimizarse. Con dichos autómatas se determinará si la cadena w pertenece o no a $L(r)$.

Objetivos

- Implementación de los algoritmos básicos de autómatas finitos y expresiones regulares.
 - Implementación del algoritmo *Shunting Yard* para convertir a postfix.
 - Implementación del algoritmo de conversión regexp a AFD (Thompson o Glushkov).
 - Implementación del algoritmo de Construcción de Subconjuntos.
 - Implementación de un algoritmo para minimización de un AFD.
 - 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 los programadores (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 estructurado (en formato .json, .yaml, xml, o similares) de texto que describa al autómata con la siguiente estructura:

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

A entregar:

1. **Documentación**: para tener derecho a nota. La documentación o informe debe de abarcar:
 - Diseño de la aplicación (el método de modelado a discreción de los desarrolladores).
 - Discusión (obstáculos encontrados, recomendaciones, etc.).
 - Ejemplos y pruebas realizadas.

- Referencias utilizadas.

2. **Código:** Este puede subirse como un archivo empaquetado (.zip, .rar, .tar o similares) directo al enlace en Canvas. También pueden dejar su código como un repositorio en Git. En este caso, en el informe deberán indicar el enlace a su repositorio.

Importante! El repositorio debe estar público (si no desean que esté público deberán incluir mi usuario con permisos de lectura). Si al momento de la revisión el repositorio no se puede acceder, se asignará automáticamente una nota de 0.

Ponderación

El Proyecto tiene un valor de 15 puntos.

Característica	Ponderación
Conversión de <i>regex</i> a <i>postfix</i> (Shunting Yard)	3 puntos
Generación de AFN (archivo)	2 puntos
Conversión de AFN a AFD (archivo)	4 puntos
Minimización del AFD (archivo)	3 puntos
Simulación AFN, AFD y AFD minimal	3 puntos
Total	15 puntos

Fecha de Entrega

En la semana del 16 al 20 de septiembre.

Se asignará un horario específico para cada grupo.