

Laboratorio No. 3

Para los siguientes incisos, siga estas instrucciones:

- Cree un repositorio **privado** en GitHub.com para alojar el código correspondiente para este laboratorio.
- Añada a los siguientes usuarios con rol de *collaborator*:
 - o Aristondo01
 - o Daniel14gonc
 - o gbrolo
- Coloque todas las instrucciones que considere pertinentes en un archivo denominado README.md en la raíz del repositorio.
- Para los incisos que no involucren la generación de código, cree una carpeta por separado en su repositorio y coloque las respuestas en un documento con formato PDF
- Grabe un video de no más de 10 minutos donde muestre la ejecución de sus programas para el ejercicio 1. Súbalo a YouTube como video no listado y adjúntelo en el README de su repositorio.

Ejercicio No. 1 (75%) – En el laboratorio anterior usted tuvo que implementar el algoritmo de Shunting Yard para convertir de una expresión infix a postfix. Ahora deberá de implementar un algoritmo en el lenguaje de programación de su elección que utilice el código que usted ya creó para convertir una expresión regular de infix a postfix y con ese output utilizarlo ahora como input para poder crear un árbol abstracto sintáctico correspondiente a la expresión regular en cuestión.

Para ello, deberá de crear diversos objetos que le permitan guardar la información correspondiente para cada nodo del árbol sintáctico y deberá de utilizar una pila para ir alojando los elementos de la expresión regular en notación postfix que obtuvo anteriormente. Consulte la sección “Relaciones entre RE, Postfix y AST” en el documento de lecciones magistrales del curso, que brinda un flujo visual de las operaciones que debería de implementar en este laboratorio.

Deberá utilizar una librería que le permita dibujar un árbol en pantalla, ya que debe mostrar el árbol generado. Si utiliza Python se recomienda que utilice la librería de Graphviz. De igual forma, todo queda a su elección.

De esta forma la ejecución esperada de su programa es la siguiente:

- Su programa debe leer un archivo de texto y procesar cada línea en este archivo. Cada línea deberá de tener una expresión regular, correspondiente a la siguiente lista:
 - o $(a * | b *) +$
 - o $((\epsilon | a) | b *) *$
 - o $(a | b) * a b b (a | b) *$
 - o $0? (1?)? 0 *$
- Cada expresión regular deberá de convertirse de infix a postfix utilizando su implementación de Shunting Yard que realizó en el laboratorio anterior. Recuerde que las extensiones de expresiones regulares (+ y ?) deberán de ser transformadas a su simplificación (i.e., $a+$ es aa^* y $a?$ es $a | \epsilon$).
- Luego de convertir a postfix, deberá implementar código que convierta la expresión postfix a un árbol sintáctico y mostrarlo en pantalla de forma gráfica.

- Muestre la ejecución completa de su programa y explique brevemente en el video su código.

Ejercicio No. 2 (25%) – Utilice el Lema de Arden para encontrar el lenguaje generado por el siguiente Autómata Finito, i.e., convierta el autómata a su correspondiente expresión regular utilizando el Lema de Arden y el algoritmo visto en clase. Deje todo su procedimiento.

