

Laboratorio #2

En este laboratorio debe entregar una implementación del algoritmo de construcción de Thompson. Su programa generará archivos de texto que contengan la descripción de un AFN generado a partir de una expresión regular.

La salida será un archivo de texto con la descripción del AFN generado, y debe incluir la lista de estados, la lista de símbolos encontrados (alfabeto), el estado inicial, la lista de estados de aceptación, y la lista de transiciones. A continuación una muestra:

```
ESTADOS = {0, 1,... n}
SIMBOLOS = {a, b, c,... z}
INICIO = {0}
ACEPTACION = {0, 1,... n}
TRANSICION = (0, a, 1)-(0, e, 2)- ... (3, b, n)
```

Debe definir qué símbolo utilizará para representar ε , y recuerde que este símbolo no debe formar parte del alfabeto. Opcionalmente puede implementar el uso de abreviaturas, la concatenación por yuxtaposición y/o la generación de diagramas de transición. Ninguna de estas características le otorgará puntos para este laboratorio, pero cuentan como puntos extra para la primera entrega de proyecto.

A continuación se presentan algunas expresiones regulares con las que puede probar la correcta funcionalidad de su programa:

- $(a^*|b^*)c$
- $(b|b)^*abb(a|b)^*$
- $(a|\varepsilon)b(a+)^c?$
- $(a|b)^*a(a|b)(a|b)$
- $b^*ab?$
- $b+abc+$
- ab^*ab^*
- $0(0|1)^*0$
- $((\varepsilon|0)1^*)^*$
- $(0|1)^*0(0|1)(0|1)$
- $(00)^*(11)^*$
- $(0|1)1^*(0|1)$
- $0?(1|\varepsilon)?0^*$
- $((1?)^*)^*$
- $(01)^*(10)^*$

Puede probar las expresiones de este listado que tengan abreviaturas realizando, antes, la transformación correspondiente. Por ejemplo, $b+abc+= (bb^*)ab(cc^*)$.

Universidad del Valle de Guatemala
Diseño de Lenguajes de Programación
Catedrático: Tomás Gálvez
Semestre 2, 2015



Para facilitar la lectura y manejo de las expresiones regulares se recomienda pasar las expresiones recibidas a notación *postfix*. Esta característica no forma parte del contenido del curso, por lo que deberá investigar si desea implementarla.