

TRABAJO PRÁCTICO GRUPAL

Deberá codificar un programa, que implemente dos algoritmos dados en clase:

- Dado un AFD encontrar la expresión regular que describe el lenguaje aceptado por el autómata.
- Dado un AFD, encontrar la gramática G3, que acepta el mismo lenguaje que el autómata.

Item a:

El programa leerá de un archivo los datos correspondiente a un AFD y generará una expresión regular mediante el método de eliminación de estados, mostrando por pantalla la expresión regular encontrada y grabando en un archivo de salida la secuencia de los estados del AFD a medida que fue eliminándolos.

El archivo de entrada, debe ser un txt, que representa al autómata como una secuencia de definiciones, cada una de las cuales se compone de un número de estado seguido por:

- El símbolo !, que indica que el estado es inicial;
- El símbolo #, que indica que el estado es final;
- Un símbolo de entrada del alfabeto del lenguaje, seguido de un número de estado, que indica que hay una transición para ese símbolo al nuevo estado.

La gramática en EBNF para definir el archivo de entrada, es la siguiente:

```
Autómata ::= Secuencia Secuencia Secuencia+
Secuencia ::= Definición SaltoRenglón
Definición ::= Estado (! | # | Entrada Estado)
Entrada ::= a | b | c
Estado ::= Dígito Dígito*
Dígito ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

El SaltoRenglón, significa que deben pasar al renglón siguiente. Por lo tanto cada Definición está en un renglón.

La expresión regular estará formada por símbolos del alfabeto $\Sigma = \{a, b, c\}$ y utilizará los operadores |, *, y los paréntesis (,).

El archivo de salida deberá ser un txt, formado por una secuencia de renglones, donde cada renglón posee el dato de los estados que quedan luego de cada eliminación. Los números de estados estarán separados por comas.

Ejemplo:

Siendo el AFD del archivo de entrada, el siguiente:

```
0!
0a0
0b1
1a1
1b0
0c2
2c3
3c4
4c2
4#
```

TRABAJO PRÁCTICO GRUPAL

Una posibilidad de archivo de salida (que representa los estados que quedan después de cada eliminación) puede ser

0,2,3,4	Elimino 1, por lo tanto quedan los estados 0,2,3 y 4
0,3,4	Elimino 2, por lo tanto quedan los estados 0,3,4
0,4	Elimino 3, por lo tanto quedan los estados 0 y 4

y la salida por pantalla (que representa la expresión regular) es

$(alb.a^*b).c.c.c.(c.c.c)^*$

Item b:

El item b, deberá tomar como entrada el mismo archivo que tomó el item a, y generará un archivo de salida con la especificación de una Gramática Regular o G3:

$A \rightarrow a$

$A \rightarrow aB$

Donde $a \in T$ y $A, B \in N$

El archivo de salida tendrá formato .txt y utilizará la siguiente nomenclatura:

Los No terminales (como se corresponden con los estados) serán representados por números.

Se utilizará el símbolo : para separar la parte izquierda de la parte derecha de las reglas de producciones.

Se deberá escribir una regla por renglón.

Ejemplo:

Continuando con el ejemplo planteado, el pasaje a AFD, producirá el siguiente archivo:

```
0:a0
0:b1
1:1a
1:b0
0:c2
2:c3
3:c4
4:c3
3:c
```

Aclaraciones:

- No es necesario que el AFD de entrada tenga el estado trampa.
- Considere que el AFD puede tener como máximo 3 estados finales. Incluso el estado inicial puede ser a su vez estado final.
- Si el AFD tiene más de un estado final, el archivo de salida deberá contener todas las secuencias de eliminación de estados. Si tomamos el ejemplo con el que veníamos trabajando y consideramos que 3 también es estado final, el archivo de salida sería:

TRABAJO PRÁCTICO GRUPAL

0,2,3,4 Elimino 1, por lo tanto quedan los estados 0,2,3 y 4
0,3,4 Elimino 2, por lo tanto quedan los estados 0,3,4
0,4 Elimino 3, considerando que 4 es final, encuentro así una parte de la expr. regular
0,3 Elimino 4 (tomando desde el 2do renglón) considerando 3 como final. Encuentro otra parte de la expresión regular.

- El orden de los registros (renglones) en los archivos, solo es importante para el caso del archivo de salida del ítem a; en todos los otros casos el orden no tiene importancia.

Desarrollo

- Lenguaje de programación a utilizar: C ó C++
- Controlar, situaciones de error. Por ejemplo que el formato del archivo de entrada no sea el correcto, o que el AF no sea Determinístico.

Entregar:

- Archivos ejecutables.
- Archivos de código fuente.
- Archivos de pruebas.
- Carpeta con:
 - Carátula, con Nombres y Apellido de los Integrantes y e-mail de cada uno.
 - Código fuente impreso,
 - Diagrama modular de funciones implementadas.
 - Breve comentario sobre la complejidad del TP, y del tiempo utilizado para su resolución.
 - Opinión del grupo sobre el tema y desarrollo del TP

Fecha de entrega: 16/11/2009

Forma de entrega: La entrega será en forma personal a algún docente de la cátedra, en lugar a convenir.

Cantidad de integrantes por grupo: 3 (sin excepciones).