Trabajo Práctico
Teoría de Lenguajes

Segundo cuatrimestre 2011

1. Introducción

El objetivo de este trabajo práctico es implementar una herramienta de búsqueda en archivos de texto. Dicha herramienta funcionará desde la línea de comandos recibiendo como parámetros una expresión regular que especificará el patrón a buscar, y el nombre del archivo en el que se efectuará la búsqueda.

Se deben mostrar por la salida standard todas las líneas del archivo aceptadas por la expresión regular.

2. Descripción de la entrada

El programa a implementar debe recibir dos argumentos: una expresión regular y el nombre de un archivo.

Definimos el lenguaje de las expresiones regulares válidas con esta gramática
$G =< \{E\}, \{\texttt{caracter}, \texttt{|}, \texttt{*}, \texttt{+}, \texttt{?}, \texttt{.}, \texttt{(}, \texttt{)}\}, P, E >$

$$
\begin{array}{rcl}
P: \quad E & \rightarrow & EE \\
 & | & E\texttt{|}E \\
 & | & E\texttt{*} \\
 & | & E\texttt{+} \\
 & | & E\texttt{?} \\
 & | & \texttt{(}E\texttt{)} \\
 & | & \texttt{caracter} \\
 & | & \texttt{.}
\end{array}
$$

Los caracteres permitidos en la expresión regular deben incluir las letras mayúsculas y minúsculas, los dígitos decimales y el espacio en blanco.

Interpretamos los operadores de la manera usual, teniendo en cuenta que

- no hay un operador explícito para la concatenación
- el signo de pregunta indica que su operando es opcional
- el punto representa un caracter cualquiera
- el operador `|` tiene menor precedencia que la concatenación, y esta menor precedencia que los operadores unarios
- los operadores binarios son asociativos a izquierda.

3. Salida

Al ejecutar el programa se debe procesar el archivo secuencialmente escribiendo en la salida standard todas las líneas que pertenezcan al lenguaje de la expresión regular.

Notar que la expresión regular debe coincidir con la línea completa. Para buscar una subcadena α se podría ingresar como parámetro la expresión $.^*\alpha.^*$.

Enviar los mensajes de error o cualquier mensaje informativo al error standard (stderr).

4. Implementación

Para implementar la búsqueda convertiremos la expresión regular de la entrada en un autómata finito no determinístico. Luego podemos determinizarlo y usarlo para procesar cada línea, devolviendo las líneas que acepte.

Deberán completarse las siguientes tareas:

- Modificar la gramática de manera que sea adecuada para construir el parser elegido.
- Implementar un parser que reconozca expresiones regulares y construya un autómata finito no determinístico equivalente. Se puede usar un generador de parsers (yacc, bison, cup, antlr, javacc) o implementarlo directamente con los métodos vistos en clase.
- Implementar un Autómata finito no determinístico y determinístico con operaciones adecuadas.
- La herramienta en sí debe parsear la expresión regular y procesar el archivo línea por línea usando un autómata finito determinístico generado a partir del no–determinístico.

5. Entregas

Habrá una sola entrega que debe incluir:

- Un programa que cumpla con lo solicitado
- El código fuente del programa. Si se usaron herramientas generadoras de código, imprimir la fuente ingresada a la herramienta, no el código generado.
- Informe conteniendo:
  - Las modificaciones a la gramática o indicaciones adicionales que hayan sido necesarias para construir el parser.
  - Descripción de cómo se implementó la solución.
  - Información y requerimientos de software para ejecutar y recompilar el tp (versiones de compiladores, herramientas, plataforma, etc).
  - Casos de prueba con expresiones sintácticamente correctas e incorrectas, resultados obtenidos y conclusiones.

Deben entregar el informe impreso. Pueden enviar los archivos que componen la entrega por mail a tptleng@gmail.com.

La fecha de entrega del trabajo práctico es el miércoles 7 de diciembre.