Trabajo Práctico
Teoría de Lenguajes

Primer cuatrimestre 2021

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 estándar todas las líneas del archivo que contengan el patrón buscado.

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 carácter 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 estándar todas las líneas que contengan al menos una subcadena que pertenezca al lenguaje denotado por la expresión regular.

Notar que la subcadena encontrada no puede extenderse más allá del fin de línea.

Enviar los mensajes de error o cualquier mensaje informativo al error estándar (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.

Como la búsqueda es de una subcadena que pertenezca al lenguaje de la expresión regular (no la línea completa), si α es la expresión regular se podría generar un autómata para $.^*\alpha.^*$ y devolver 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, ply) 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 analizar 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 deberá incluir:

- Un programa que cumpla con lo solicitado, para lo cual se puede usar C++, C#, Java, Python o Go, con comentarios relevantes, más el código ingresado a la herramienta de parsing en caso de usar alguna. Si desean utilizar otro lenguaje, u otra herramienta, consulten con el JTP.
- 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.

Grupos: de 3 personas. Modo de entrega: por e-mail a tptleng@gmail.com hasta el día 5/7, en un archivo .rar o .zip., con subject: *entrega de TP grupo...*