

Trabajo Práctico Especial 1

Conversor de autómatas a gramática y viceversa

- Castro Peña, Gonzalo --- 49398
- Pintos, Esteban – 51048
- De Santi, Matías – 51051

Consideraciones realizadas	3
Descripción del desarrollo del TP	3
Dificultades encontradas	3
Futuras extensiones	4
Conclusiones.....	4

Consideraciones realizadas

Dado que las especificaciones del trabajo práctico fueron precisas, no fue necesario hacer consideraciones especiales.

Descripción del desarrollo del TP

Para realizar el trabajo se realizó una implementación de un autómata y una implementación de una gramática. Para simplificar el manejo de los datos se crearon ADTs.

Por un lado, el autómata se implementó como un grafo que contenía las siguientes estructuras: nodo y transición. Un nodo contiene un arreglo de transiciones hacia sus nodos vecinos además de otros datos necesarios dentro de un nodo.

Por otro lado, para la implementación de la gramática se decidió usar ADTs para guardar los datos sobre una producción. De esta manera, se pueden guardar todas las producciones con la información necesaria en cada una.

Una vez que la estructura está armada, se procede a realizar los algoritmos correspondientes para pasar al tipo requerido (si la entrada fue un autómata, el algoritmo que se corre es para generar una gramática).

Una vez finalizados dichos algoritmos, se imprimen los datos solicitados y el archivo de salida correspondiente (.gr o .png).

Dificultades encontradas

La mayor dificultad se presentó a la hora de realizar el algoritmo de conversión de gramática a autómata ya que varias veces se tuvo que revisar el algoritmo debido a pequeñas fallas a la hora de eliminar producciones unitarias. Sin embargo, estas fallas no fueron grandes dificultades y el trabajo se pudo realizar satisfactoriamente.

Por otra parte, el desarrollo del parser en lex también tuvo varias versiones. En un principio se había generado uno sin una máquina de estados pero fue

evidente rápidamente que esto no funcionaba. Fue por ello que se decidió por hacer una implementación con máquinas de estados.

Futuras extensiones

Una posible extensión de este trabajo práctico es la posibilidad de recibir dos autómatas y retornar una gramática que genere la concatenación o la unión de los lenguajes descritos por dichos autómatas. Esta extensión requiere de un análisis de los estados finales y el inicial del siguiente automata para unirlos con una transición lambda. No debería tener una complejidad elevada.

Otra posibilidad es la de, dado un autómata o una gramática, generar la expresión regular correspondiente al lenguaje descrito por el archivo de entrada. Si la entrada es un automata, se debe plantear el sistema de ecuaciones apropiado para dicho automata y resolverlo. En caso de que la entrada sea una gramática, se pasa a automata y luego se aplica el mismo razonamiento. Tampoco debería resultar en una complejidad elevada.

Conclusiones

Este trabajo sirvió como una primera aproximación a Lex. Es un lenguaje muy potente y simplifica mucho el parseo de archivos para extraer datos de los mismos. Luego de haber tenido la experiencia de desarrollar nuestros propios parsers para otros trabajos prácticos, podemos concluir que Lex proporciona grandes ventajas y simplifica el proceso de extracción de datos de archivos.