

Trabajo práctico de Autómatas, teoría de lenguajes y compiladores

Parser de Gramáticas LL(1)

Autores:

Alejo Hegouaburu
Gustavo Maldonado
Gisela de la Villa

Índice

Índice	2
Consideraciones realizadas	3
Descripción del desarrollo del TP.....	3
Dificultades encontradas en el desarrollo del TP	3
Gramáticas de ejemplo.....	3

Consideraciones realizadas

- Se realiza el cálculo completo de Primeros y Siguietes, sin tomar las consideraciones de simplificación sugeridas en el enunciado
- Por el punto anterior, las producciones de la gramática no necesitan empezar con un símbolo terminal
- Se considera que todos los símbolos no terminales que aparecen en la producción están en el lenguaje.
- Se asume que nunca viene una gramática recursiva izquierda (si no muere en la recursión). No se implementó el eliminador de recursividad izquierda.

Descripción del desarrollo del TP

El trabajo consiste en generar un programa que genere un parser de una gramática de tipo LL(1) dada, de modo que pueda recibir una cadena como entrada y determinar si ésta pertenece al lenguaje descrito por la gramática o no. Este programa fue escrito en lenguaje Java utilizando el API de la Java Standard Edition v1.6.

El programa lee de un archivo las producciones correspondientes a la gramática. Asumiendo que S es el símbolo inicial, que los símbolos no terminales son letras mayúsculas y que los terminales son letras minúsculas, identifica cómo están compuestos los conjuntos V_T y V_N . A su vez, almacena las producciones en una colección de instancias de la clase Production. A continuación, calcula el conjunto de Primeros de cada símbolo no terminal y el conjunto de Siguietes de los mismos. Por último, se calculan los símbolos directrices de cada producción, se genera la tabla y se espera a las palabras de entrada para verificar su pertenencia al lenguaje utilizando el algoritmo de análisis sintáctico descendente.

Durante el análisis de la palabra, se imprime por salida estándar el estado de la pila y, al final, el resultado del análisis. Esta decisión se tomó a fin de facilitar el seguimiento del proceso de análisis por parte del usuario.

Dificultades encontradas en el desarrollo del TP

No se han encontrado dificultades en la realización del trabajo. Siguiendo los algoritmos provistos en la teoría de la cátedra, el pasaje a código Java fue casi automático y funcionó sin necesidad de refactorizar.

Gramáticas de ejemplo

Archivo:

error2

Producciones:

S xSz

S ySz

S

Genera:

Ejemplos válidos:

"XZ"

"xyzz"

"yyxyyxxyxxxxyzoooooooooooo"

"Z"

"X"

"xyyz"