Sintaxis y Semántica de los Lenguajes
Facultad Regional Delta,
Universidad Tecnológica Nacional

Trabajo de Laboratorio N°1
(22 de Mayo de 2023)

Objetivo

Implementar un analizador lexicográfico para una gramática especificada.

Enunciado

La implementación del analizador lexicográfico se realizará en grupos de 4 integrantes como máximo. Cada grupo recibirá una gramática para implementar el analizador. Se pretende desarrollar un Autómata Finito(AF) por cada tipo de token y luego el analizador deberá ser implementado mediante un AFD que incluye a todos los AF construidos para cada token.

El programa que resulte de la implementación deberá aceptar una cadena que representa código escrito en el lenguaje generado por la gramática provista. Este código, visto como una cadena de caracteres ASCII, deberá ser convertido a una cadena de tokens correspondiente a la gramática provista.

Entrega de informe

Una vez aprobado el código del analizador lexicográfico, deberá ser entregado un informe donde figure:

- ➔ Carátula (Universidad, Regional, Carrera, Materia, integrantes del grupo, docentes, año)
- ➔ Enunciado con la gramática que se les asignó.
- ➔ Explicación, observaciones y comentarios sobre el trabajo realizado (Por ejemplo: cómo se implementa el analizador, explicación de las ideas usadas en la implementación del algoritmo, estructuras de datos utilizadas, cuáles y porque fueron elegidas, más otras explicaciones que crean convenientes para entender su trabajo).
- ➔ Ejemplos de cadenas de prueba con los resultados pertinentes.

Recomendaciones

Aquí enumeramos algunas consideraciones a tener en cuenta a la hora de desarrollar el trabajo, respecto al formato del código, versión del software, formato y nombre de las entregas, etc , a modo de recordatorio de lo sugerido en clase:

- ➔ No utilizar funciones tipo input o cualquier interactividad que requiera la intervención del usuario más allá de correr el programa, pues esto dificulta el desarrollo para los alumnos y la evaluación por parte del profesor.

- ➔ Enviar todos los archivos necesarios para correr el programa comprimidos en un único archivo con formato zip, rar o similares.
- ➔ El archivo debe llamarse: ***{apellido o nombre del grupo}_tp1_{version}***, donde cada nueva ronda de correcciones tiene que tener una nueva versión (1, 2, 3, etc)
- ➔ Agregar un minimo de 10 pruebas para el lexer

Gramática

Si bien se especifica a continuación la gramática en la notación usual, detallando cada una de los 4 elementos de la gramática, se hacen las siguientes aclaraciones para que puedan leer la gramática más fácilmente desde las producciones:

- ➔ El símbolo distinguido es Programa.
- ➔ Los terminales se hallan entre “ ”, y en negrita, por ejemplo “**var**” es un terminal llamado *var*.
- ➔ Los no terminales no se hallan entre “ ” y siempre comienzan en mayúsculas, pudiendo contener mayúsculas intermedias para aclarar su significado.
- ➔ Terminales y No Terminales se hallan separados por espacios en blanco para claridad de la gramática.

La gramática G=<VN, VT, P, S> se halla completa para mejor visualización en la siguiente página, siendo:

VT = {id, num, si, entonces, sino, finsi, repetir, hasta, equal, leer, mostrar, func, finfunc, (,), ;, oprel, opsuma, opmult}
VN = {Program, ListaSentencias, Sentencia, SentenciaSi, SentenciaRepetir, SentenciaAsig, SentenciaLeer, SentenciaMostrar, SentenciaFun,Proc, ListaPar, Expression, Expresion2, Factor, Termino}
S = Program
P = {

Program → ListaSentencias

ListaSentencias → ListaSentencias;Sentencia
| Sentencia

Sentencia → SentenciaSi
| SentenciaRepetir
| SentenciaAsig
| SentenciaLeer
| SentenciaMostrar
| SentenciaFun

SentenciaSi → **“si”** Expresion **“entonces”** ListaSentencias **“sino”** ListaSentencias **“finsi”**
| **“si”** Expresion **“entonces”** ListaSentencias **“finsi”**

SentenciaRepetir → **“repetir”** ListaSentencias **“hasta”** Expresion

SentenciaAsig → **“id” “equal”** Expresion

SentenciaLeer → **“leer” “id”**

SentenciaAsig → **“mostrar”** Expresion

SentenciaFun → **“func”** Proc **“finfunc”**

Proc → **“id”** “(“ ListaPar “)” ListaSentencias

ListaPar → ListaPar **“;” “id”**
| **“id”**

Expresion → Expresion2 **“oprel”** Expresion2
| Expresion2

Expresion2 → Expresion2 **“opsuma”** Termino
| Termino

Termino → Termino **“opmult”** Factor | Factor

Factor → **“(“** Expresion **“)”** | **“num”** | **“id”**

}