

## Sintaxis y Semántica de los Lenguajes

Facultad Regional Delta,  
Universidad Tecnológica Nacional

### Trabajo de Laboratorio N°2

(14 de Agosto de 2023)

#### Objetivo

Implementar un analizador sintáctico descendente ***predictivo por procedimientos***.

#### Enunciado

- La implementación del analizador sintáctico descendente predictivo por procedimientos se realizará en grupo de hasta cuatro alumnos, el mismo grupo que el formado para el TP N°1 de laboratorio.
- Cada grupo implementará el analizador para la misma gramática utilizada en el TP N°1, utilizando ***el analizador lexicográfico producido para dicho trabajo***.
- Debe verificarse que ***la gramática dada cumple las condiciones teóricas*** para poder analizarse con el tipo de analizador sintáctico solicitado.
- El programa que resulte de la implementación deberá recibir una cadena y luego ***indicar si dicha cadena pertenece al lenguaje generado por la gramática*** y además ***deberá indicar qué producciones de la gramática deben ser usadas para derivar la cadena de entrada***, en caso de pertenecer.
- Conjuntamente cada grupo producirá un conjunto de cadenas de testeo, de las cuales, algunas pertenecerán al lenguaje generado por la gramática y otras no, indicando la salida de su parser en cada caso.

#### Entrega de informe

Deberá ser entregado un informe donde figure:

- Carátula (Universidad, Regional, Carrera, Materia, integrantes del grupo, docentes, año)
- 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}\_tp2\_{version}***, donde cada nueva ronda de correcciones tiene que tener una nueva versión (1, 2, 3, etc)
- Agregar un mínimo de 10 pruebas para el analizador sintáctico.

## **Gramática**

El analizador sintáctico se realizará utilizando la misma gramática que en el TP N°1, con las mismas consideraciones que allí fue indicado.

Por cuestiones de claridad, la gramática  $G = \langle VN, VT, P, S \rangle$  se repite completa para mejor visualización en la siguiente página:

```

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”

SentenciaMostrar → “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”

}

```