

## **Sintaxis y Semántica de los Lenguajes**

Facultad Regional Delta,  
Universidad Tecnológica Nacional

### **Trabajo de Laboratorio N°2**

(28 de septiembre de 2015)

#### **Objetivo**

Implementar un analizador sintáctico descendente para una gramática especificada.

#### **Enunciado**

La implementación del analizador sintáctico descendente con retroceso (ASDR) se realizará en grupo de hasta tres alumnos (preferentemente dos alumnos).

Cada grupo recibirá una gramática para implementar el analizador. El tipo de ASDR deberá ser implementado mediante procedimientos, esto es, deberá haber un procedimiento Principal, un procedimiento Procesar y luego un procedimiento por cada no terminal de la gramática.

El programa que resulte de la implementación deberá aceptar 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.

Conjuntamente con la gramática, a cada grupo se le dará un conjunto de cadenas de testeo, de las cuales, algunas pertenecerán al lenguaje generado por la gramática y otras no.

#### **Entrega de informe**

El día 17 de noviembre deberá ser entregado un informe donde figure:

- Carátula (Universidad, Regional, Carrera, Materia, integrantes, docentes, año)
- Enunciado con la gramática que se les asignó.
- Pseudocódigo del programa
- Estructura de datos utilizadas
- Corrida de cada una de las cadenas que se les suministró conjuntamente con la gramática.
- Conclusiones. En esta última sección se describe qué se realizó y qué conclusiones se obtuvieron a partir del trabajo realizado.

## ***Gramática***

*Símbolos terminales* = { (, ), {, }, +, \*, -, O, Y, NO, A..Z, a..z, 0..9, VERDADERO, FALSO, Para, desde, hasta, Si, entonces, =, ==, <, >, >=, <=, ,, entero, real, logica, return }

*Símbolos no terminales* = { P, DV, DF, LC, Tipo, NombreVariable, NombreVariableCont, DF, NombreFuncion, LP, NombreVariableCont, LPCont, CAsignacion, Cpara, CSiEntonces, CLLamadoFuncion, ExpEntera, ExpLogica, LLPar, LLParCont, ExpEntera, Termino, Factor, ConstEntera, ConstEnteraCont, ExpLogica, OperandoLogico, FactorLogico, ConstLogica, Comparacion, Operador }

*Símbolo distinguido*: P

*Conjunto de producciones*:

**P** → DV DF LC

**DV** → Tipo NombreVariable ; DV /  $\lambda$

**Tipo** → entero / real / logica

**NombreVariable** → a NombreVariableCont / .. / z NombreVariableCont

**NombreVariableCont** → a NombreVariableCont / .. / z NombreVariableCont /  $\lambda$

**DF** → Tipo NombreFuncion LP { LC ; return NombreVariable ; } DF /  $\lambda$

**NombreFuncion** → A NombreFuncionCont / .. / Z NombreFuncionCont /  $\lambda$

**NombreFuncionCont** → a NombreFuncionCont / .. / z NombreFuncionCont /  $\lambda$

**LP** → ( LPCont ) / ( )

**LPCont** → Tipo NombreVariable , LPCont / Tipo NombreVariable

**LC** → CAsignacion LC / CPara LC / CSiEntonces LC / CLLamadoFuncion LC /  $\lambda$

**CAsignacion** → NombreVariable = ExpEntera ;

**CPara** → Para NombreVariable desde ExpEntera hasta ExpEntera { LC }

**CSiEntonces** → Si ExpLogica entonces { LC }

**CLlamadoFuncion** → NombreFuncion LLPar ;

**LLPar** → ( LLParCont ) / ( )

**LLParCont**  $\rightarrow$  NombreVariable , LLParCont / NombreVariable

**ExpEntera**  $\rightarrow$  ExpEntera + Termino / ExpEntera - Termino / Termino

**Termino**  $\rightarrow$  Termino \* Factor / Factor

**Factor**  $\rightarrow$  ( ExpEntera ) / ConstEntera / NombreVariable

**ConstEntera**  $\rightarrow$  1 ConstEnteraCont / .. / 9 ConstEnteraCont / 0

**ConstEnteraCont**  $\rightarrow$  0 ConstEnteraCont / .. / 9 ConstEnteraCont /  $\lambda$

**ExpLogica**  $\rightarrow$  ExpLogica O OperandoLogico / OperandoLogico

**OperandoLogico**  $\rightarrow$  OperandoLogico Y FactorLogico / FactorLogico / NO FactorLogico

**FactorLogico**  $\rightarrow$  (ExpLogica) / ConstLogica / NombreVariable / Comparacion

**ConstLogica**  $\rightarrow$  VERDADERO / FALSO

**Comparacion**  $\rightarrow$  ExpEntera Operador ExpEntera

**Operador**  $\rightarrow$  == / > / < / >= / <=