



## PROCESADORES DE LENGUAJE

Ingeniería Informática  
Especialidad de computación  
Tercer curso, segundo cuatrimestre



Departamento de Informática y Análisis Numérico  
Escuela Politécnica Superior  
Universidad de Córdoba  
Curso académico 2017 - 2018

### Hoja de ejercicios nº 2.- FUNDAMENTOS TEÓRICOS DEL ANÁLISIS SINTÁCTICO

#### Derivaciones y árboles sintácticos

1. Una gramática de contexto libre  $G$  posee el siguiente conjunto de producciones:

$$P = \{ \begin{array}{l} \langle \text{oración} \rangle \rightarrow \langle \text{sujeto} \rangle \langle \text{predicado} \rangle \\ \langle \text{sujeto} \rangle \rightarrow \langle \text{grupo\_nominal} \rangle \\ \langle \text{sujeto} \rangle \rightarrow \langle \text{grupo\_nominal} \rangle \text{ ADJETIVO} \\ \langle \text{grupo\_nominal} \rangle \rightarrow \text{NOMBRE} \mid \text{ARTÍCULO NOMBRE} \\ \langle \text{predicado} \rangle \rightarrow \text{VERBO} \langle \text{complementos} \rangle \\ \langle \text{complementos} \rangle \rightarrow \langle \text{directo} \rangle \langle \text{indirecto} \rangle \langle \text{circunstanciales} \rangle \\ \langle \text{complementos} \rangle \rightarrow \langle \text{indirecto} \rangle \langle \text{circunstanciales} \rangle \\ \langle \text{complementos} \rangle \rightarrow \langle \text{directo} \rangle \langle \text{circunstanciales} \rangle \\ \langle \text{complementos} \rangle \rightarrow \langle \text{circunstanciales} \rangle \\ \langle \text{directo} \rangle \rightarrow \langle \text{grupo\_nominal} \rangle \mid \langle \text{grupo\_nominal} \rangle \text{ ADJETIVO} \\ \langle \text{indirecto} \rangle \rightarrow \text{PREPOSICIÓN} \langle \text{grupo\_nominal} \rangle \\ \langle \text{circunstanciales} \rangle \rightarrow \varepsilon \\ \langle \text{circunstanciales} \rangle \rightarrow \langle \text{circunstancial} \rangle \langle \text{circunstanciales} \rangle \\ \langle \text{circunstancial} \rangle \rightarrow \text{PREPOSICIÓN} \langle \text{grupo\_nominal} \rangle \end{array} \}$$

- Indica cuáles son los símbolos terminales y cuáles los no terminales
- ¿Cuál es el símbolo inicial?
- ¿Es la gramática recursiva por la izquierda o por la derecha?
- Muestra las derivaciones por la izquierda y por la derecha de la siguiente cadena:

**La niña entregó la llave mágica a un amigo en el bosque**

- Dibuja el árbol sintáctico asociado a la derivación por la izquierda.

2. La siguiente gramática genera expresiones aritméticas con notación prefija:

$$P = \{ \begin{array}{l} E \rightarrow ( O L ) \\ O \rightarrow + \mid - \mid * \mid / \\ L \rightarrow A \mid A L \\ A \rightarrow \text{IDENTIFICADOR} \mid \text{NÚMERO} \mid E \end{array} \}$$

- Indica cuáles son los símbolos terminales y cuáles los no terminales
- ¿Cuál es el símbolo inicial?
- ¿Es la gramática recursiva por la izquierda o por la derecha?
- Muestra las derivaciones por la izquierda y por la derecha de la siguiente expresión:

(+ (\* a a ) (\* b b))

e. Muestra los árboles sintácticos asociados a las derivaciones del apartado anterior y comprueba si son iguales.

3. La siguiente gramática genera sentencias del lenguaje Pascal:

```
P = { <asignación_lógica> → IDENTIFICADOR := <predicado>
    <predicado> → <predicado> OR <disyunción>
    <predicado> → <disyunción>
    <disyunción> → <disyunción> AND <conjunción>
    <disyunción> → <conjunción>
    <conjunción> → <simple> | NOT ( <predicado> )
    <simple> → ( <predicado> )
    <simple> → <operando> <operador_relacional> <operando>
    <operador_relacional> → = | < | <= | > | >= | <>
    <simple> → true | false
    <operando> → IDENTIFICADOR | NÚMERO | true | false
}
```

- Indica cuáles son los símbolos terminales y cuáles los no terminales
- ¿Cuál es el símbolo inicial?
- ¿Es la gramática recursiva por la izquierda o por la derecha?
- Muestra las derivaciones por la izquierda de las siguientes sentencias:
  - estado := ( final <> true )*
  - apto := (not ((teoría < 4) or (prácticas < 4))) and (media >= 5)*
- Dibuja el árbol sintáctico asociado a la derivaciones por la izquierda

**Recomendación:** renombra los símbolos no terminales

### Diseño de gramáticas

4. Diseña gramáticas de contexto libre que generen los lenguajes que se indican:

- $L_1 = \{ x \mid x = a y b \wedge y \in \{0,1\}^* \}$
- $L_2 = \{ a^i c^{2j} b^i \mid i, j > 0 \}$
- $L_3 = \{ a^{2i} b^i \mid i > 0 \}$
- $L_4 = \{ a^i b^j c^k \mid i, j, k > 0 \wedge j = i + k \}$
- $L_5 = \{ x \mid x \text{ tiene igual número de ceros que de unos} \}$
- $L_6 = \{ w w^R \mid w \in \{0,1\}^* \wedge w^R \text{ es la palabra inversa o refleja de } w \}$

5. Diseña gramáticas de contexto libre que permitan generar las siguientes sentencias del lenguaje de programación C:

- Proposiciones lógicas, como por ejemplo:  
 $(a == b) \ \&\& \ (c != 0 \ || \ d >= 1)$
- Sentencias de control de C: **if**, **while**, **for** y **switch**.

6. Diseña gramáticas que permitan generar algunas de las declaraciones del lenguaje de programación Pascal:

- Declaraciones de variables simples  
*a, b, c: integer;*  
*d, e: integer := 9;*  
*x, y: real;*  
*z: real := 7.5;*
- Declaraciones de arrays  
*vector\_rango: array [-10 .. 10] of real;*

datos: array [7 .. 12] of integer := [90, 18, 23, -12, 37, 10];

## Ambigüedad

7. Una gramática de contexto libre  $G$  posee el siguiente conjunto de producciones:
- $P = \{S \rightarrow a \mid S a \mid b S S \mid S S b \mid S b S\}$ 
    - a. Comprueba que es ambigua generando dos derivaciones por la izquierda (o por la derecha) diferentes.
    - b. Construye los árboles sintácticos asociados a esas derivaciones.
8. Una gramática de contexto libre  $G$  posee el siguiente conjunto de producciones:
- $P = \{S \rightarrow A \mid B$   
     $A \rightarrow a A b \mid a b$   
     $B \rightarrow a b B \mid \varepsilon$   
     $\}$ 
    - a. Indica el lenguaje que genera esta gramática.
    - b. Comprueba que la gramática es ambigua y diseña otra equivalente que no lo sea.
9. Demuestra que si una gramática de contexto libre posee la siguiente característica entonces ha de ser ambigua:  
*"Existe un símbolo no terminal "A" que posee, simultáneamente, alguna producción recursiva por la izquierda ( $A \rightarrow A a$ ) y alguna producción recursiva por la derecha ( $A \rightarrow b A$ )".*

## Operaciones de limpieza

10. Dadas las siguientes gramáticas, construye otras equivalentes sin símbolos inútiles.
- $P = \{S \rightarrow A C B d \mid B a B,$   
     $A \rightarrow a A d \mid B C a \mid a b,$   
     $B \rightarrow b B b \mid a,$   
     $C \rightarrow C A C \mid A C c\}$
  - $P = \{S \rightarrow A B \mid C A d,$   
     $A \rightarrow a A b \mid b b A \mid a a,$   
     $B \rightarrow b A C \mid a B \mid B A,$   
     $C \rightarrow b a C a c \mid a b D,$   
     $D \rightarrow b D b c \mid c C a\}$
11. Dadas las siguientes gramáticas:
- $P = \{S \rightarrow L \text{ IDENTIFICADOR } := E ; , L \rightarrow L \text{ IDENTIFICADOR } := \mid \varepsilon ,$   
     $E \rightarrow E + T \mid T , T \rightarrow \text{IDENTIFICADOR} \mid \text{NÚMERO} \}$
  - $P = \{S \rightarrow a A a \mid b B b \mid A B , A \rightarrow a A a \mid \varepsilon , B \rightarrow b B b \mid \varepsilon \}$ 
    - a. Obtén otras gramáticas equivalentes sin reglas épsilon.
    - b. Suprime las reglas unitarias de las gramáticas obtenidas en el apartado anterior.

## Recursividad y factorización

12. Una gramática de contexto libre  $G$  posee el siguiente conjunto de producciones:
- $$P = \{$$
- $\langle \text{expresión-relacional} \rangle \rightarrow (\langle \text{operador-relacional} \rangle \langle \text{argumentos} \rangle )$
  - $\langle \text{operador-relacional} \rangle \rightarrow < \mid \leq \mid = \mid > \mid \geq$
  - $\langle \text{argumentos} \rangle \rightarrow \langle \text{argumentos} \rangle \langle \text{argumento} \rangle \mid \langle \text{argumento} \rangle$
  - $\langle \text{argumento} \rangle \rightarrow \text{NÚMERO} \mid \text{IDENTIFICADOR} \}$

- Elimina la recursividad por la izquierda y factorízala por la izquierda.
- Utiliza la gramática obtenida en el apartado anterior para generar la derivación por la izquierda y el árbol sintáctico de las siguiente sentencia:  
( $\leq 0$  temperatura 100)

- **Recomendación:** renombra los símbolos no terminales

13. La siguiente gramática permite generar asignaciones de expresiones aritméticas:

$$P = \{$$

$$A \rightarrow \text{IDENTIFICADOR} = E$$

$$E \rightarrow T \mid E + T$$

$$T \rightarrow P \mid T * P$$

$$P \rightarrow F \mid F \wedge P$$

$$F \rightarrow ( E ) \mid \text{NÚMERO} \mid \text{IDENTIFICADOR}$$

$$\}$$

- Elimina la recursividad por la izquierda y factoriza la gramática por la izquierda.
- Utiliza la gramática obtenida en el apartado anterior para generar la derivación por la izquierda y el árbol sintáctico de las siguiente sentencia:  
 $h = (a^2 + b^2) \wedge 0.5$

14. La siguiente gramática permite generar algunas de las enumeraciones del lenguaje C.

$$P = \{$$

$$S \rightarrow S E$$

$$S \rightarrow E$$

$$E \rightarrow \text{enum IDENTIFICADOR} \{ L \};$$

$$L \rightarrow L, I$$

$$L \rightarrow I$$

$$I \rightarrow \text{IDENTIFICADOR}$$

$$I \rightarrow \text{IDENTIFICADOR} = \text{NÚMERO}$$

$$\}$$

- Elimina la recursividad por la izquierda y factoriza la gramática por la izquierda.
- Utiliza la gramática obtenida en el apartado anterior para generar la derivación por la izquierda y el árbol sintáctico de las siguiente sentencia:  
`enum color { blanco, negro = -1, amarillo = 9, rojo };`

## Formas normales

15. Dada la siguiente gramática de contexto libre G:

$$P = \{ S \rightarrow T L ;$$

$$T \rightarrow \text{INT} \mid \text{FLOAT}$$

$$L \rightarrow \text{IDENTIFICADOR} \mid \text{IDENTIFICADOR } L'$$

$$L' \rightarrow , \text{IDENTIFICADOR} \mid , \text{IDENTIFICADOR } L'$$

$$\}$$

- Obtén la forma normal de Chomsky
- Obtén la forma normal de Greibach