



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 2018 - 2019

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 reglas de producción:

$$P = \{ \begin{aligned} &\langle \text{oración} \rangle \rightarrow \langle \text{sujeto} \rangle \langle \text{predicado} \rangle \\ &\langle \text{sujeto} \rangle \rightarrow \langle \text{grupo_nominal} \rangle \\ &\langle \text{sujeto} \rangle \rightarrow \langle \text{grupo_nominal} \rangle \text{ ADJETIVO} \\ &\langle \text{grupo_nominal} \rangle \rightarrow \text{NOMBRE} \mid \text{ARTÍCULO NOMBRE} \\ &\langle \text{predicado} \rangle \rightarrow \text{VERBO} \langle \text{complementos} \rangle \\ &\langle \text{complementos} \rangle \rightarrow \langle \text{directo} \rangle \langle \text{indirecto} \rangle \langle \text{circunstanciales} \rangle \\ &\langle \text{complementos} \rangle \rightarrow \langle \text{indirecto} \rangle \langle \text{circunstanciales} \rangle \\ &\langle \text{complementos} \rangle \rightarrow \langle \text{directo} \rangle \langle \text{circunstanciales} \rangle \\ &\langle \text{complementos} \rangle \rightarrow \langle \text{circunstanciales} \rangle \\ &\langle \text{directo} \rangle \rightarrow \langle \text{grupo_nominal} \rangle \mid \langle \text{grupo_nominal} \rangle \text{ ADJETIVO} \\ &\langle \text{indirecto} \rangle \rightarrow \text{PREPOSICIÓN} \langle \text{grupo_nominal} \rangle \\ &\langle \text{circunstanciales} \rangle \rightarrow \varepsilon \\ &\langle \text{circunstanciales} \rangle \rightarrow \langle \text{circunstancial} \rangle \langle \text{circunstanciales} \rangle \\ &\langle \text{circunstancial} \rangle \rightarrow \text{PREPOSICIÓN} \langle \text{grupo_nominal} \rangle \end{aligned} \}$$

- 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:

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

- 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?

- d. 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> → <constante>
 <constante> → true | false
 <operando> → IDENTIFICADOR | NÚMERO | <constante>
 }

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

```
6. Diseña una gramática que permita definir los atributos privados de una clase de C++ como la siguiente:

```

class Persona
{
 private:
 string _nombre;
 string _apellidos;
 int _edad;
 bool _sexo;
}

```

donde *Persona*, *\_nombre*, *\_apellidos*, *\_edad* y *\_sexo* son identificadores

## 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.
  - $P_1 = \{ S \rightarrow a A a$   
 $A \rightarrow a A a \mid b B b$   
 $B \rightarrow b B b \mid b b \mid c C d$   
 $C \rightarrow c C c \mid d D d$   
 $D \rightarrow c C c \mid d D d$   
 $E \rightarrow e E e \mid f \}$
  - $P_2 = \{ S \rightarrow A B$   
 $A \rightarrow a a A \mid a a$   
 $B \rightarrow b b B \mid b b \mid c C \mid d D$   
 $C \rightarrow d D$   
 $D \rightarrow c C$   
 $E \rightarrow f F$   
 $F \rightarrow e E \mid g \}$
  - Después de eliminar los símbolos inútiles, ¿qué lenguajes generan las gramáticas  $P_1$  y  $P_2$ ?

11. Dadas las siguientes gramáticas:

- $P = \{ S \rightarrow L \text{ IDENTIFICADOR} := E ; , L \rightarrow L \text{ IDENTIFICADOR} := | \varepsilon , E \rightarrow E + T | T , T \rightarrow \text{IDENTIFICADOR} | \text{NÚMERO} \}$
- $P = \{ S \rightarrow a A a | b B b | A B , A \rightarrow a A a | \varepsilon , B \rightarrow b B b | \varepsilon \}$ 
  - 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:

- $P = \{$   
 $\text{<expresión-relacional>} \rightarrow (\text{<operador-relacional>} \text{ <argumentos> } )$   
 $\text{<operador-relacional>} \rightarrow < | <= | = | > | >=$   
 $\text{<argumentos>} \rightarrow \text{<argumentos>} \text{ <argumento>} | \text{<argumento>}$   
 $\text{<argumento>} \rightarrow \text{NÚMERO} | \text{IDENTIFICADOR} \}$ 
  - a. Elimina la recursividad por la izquierda y factorízala por la izquierda.
  - b. Utiliza la gramática obtenida en el apartado anterior para generar la derivación por la izquierda y el árbol sintáctico de las siguiente sentencia:  
 $(<= 0 \text{ temperatura } 100)$
- **Recomendación:** renombra los símbolos no terminales

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

- $P = \{$   
 $A \rightarrow \text{IDENTIFICADOR} = E$   
 $E \rightarrow T | E + T$   
 $T \rightarrow P | T * P$   
 $P \rightarrow F | F \wedge P$   
 $F \rightarrow ( E ) | \text{NÚMERO} | \text{IDENTIFICADOR}$   
 $\}$ 
  - a. Elimina la recursividad por la izquierda y factoriza la gramática por la izquierda.
  - b. 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. Dada una gramática que genera algunas de las enumeraciones del lenguaje C.

- $P = \{$   
 $S \rightarrow S E$   
 $S \rightarrow E$   
 $E \rightarrow \text{enum IDENTIFICADOR} \{ L \} ;$   
 $L \rightarrow L, I$   
 $L \rightarrow I$   
 $I \rightarrow \text{IDENTIFICADOR}$   
 $I \rightarrow \text{IDENTIFICADOR} = \text{NÚMERO}$   
 $\}$ 
  - a. Elimina la recursividad por la izquierda y factoriza la gramática por la izquierda.
  - b. Utiliza la gramática obtenida en el apartado anterior para generar la derivación por la izquierda y el árbol sintáctico de las siguiente sentencia:  
 $\text{enum color} \{ \text{blanco}, \text{negro} = -1, \text{amarillo} = 9, \text{rojo} \};$

### Formas normales

15. Dada la siguiente gramática de contexto libre G:

- $P = \{ \begin{array}{l} S \rightarrow T L ; \\ T \rightarrow \text{int} \mid \text{float} \\ L \rightarrow \text{IDENTIFICADOR} \mid \text{IDENTIFICADOR } L' \\ L' \rightarrow , \text{IDENTIFICADOR} \mid , \text{IDENTIFICADOR } L' \end{array} \}$
- a. Obtén la forma normal de Chomsky
- b. Obtén la forma normal de Greibach