



PROCESADORES DE LENGUAJE

Ingeniería Informática
Primer curso de segundo ciclo



Departamento de Informática y Análisis Numérico
Escuela Politécnica Superior
Universidad de Córdoba
Curso académico 2010 - 2011

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

El embajador regaló el libro dorado a la princesa en el palacio

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

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

$$P = \{ \begin{array}{l} E \rightarrow (O L) \\ O \rightarrow + \mid - \mid * \mid / \\ L \rightarrow I \mid I L \\ I \rightarrow \text{identificador} \mid \text{número} \mid E \end{array} \}$$

- 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?

- d. Muestra la derivación por la izquierda la siguiente expresión:
 $(+ (* a a) (* b b))$
 e. Muestra el árbol sintáctico asociado a la derivación del apartado anterior.

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>
      <simple> → true | false
      <operando> → identificador | número
    }
```

- 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 := (salida = true)**
 ▪ **apto := (not ((teoría < 4) or (prácticas < 4))) and (media >= 5)**
 e. Dibuja el árbol sintáctico asociado a la derivaciones por la izquierda

Diseño de gramáticas

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

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

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 en 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
 $a, b, c: \text{integer};$
 $d, e: \text{integer} = 9;$
 $x, y: \text{real};$
 $z: \text{real} = 7.5;$

- 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 ($A \rightarrow b A$)".

Recursividad y factorización

10. Una gramática de contexto libre G posee el siguiente conjunto de producciones:
- $$P = \{$$
- $$\text{<expresión-relacional>} \rightarrow (\text{<operador-relacional> } \text{<argumentos> })$$
- $$\text{<operador-relacional>} \rightarrow < \mid <= \mid = \mid > \mid >=$$
- $$\text{<argumentos>} \rightarrow \text{<argumentos> } \text{<argumento>} \mid \text{<argumento>}$$
- $$\text{<argumento>} \rightarrow \text{número} \mid \text{identificador}$$
- $$\}$$
- 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
11. La siguiente gramática permite generar asignaciones de expresiones aritméticas:
- $$P = \{$$
- $$A \rightarrow \text{identificador} = E$$

```

E → T | E + T
T → P | T * P
P → F | F ^ P
F → ( E ) | número | 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:

$$h = (a^2 + b^2)^{0.5}$$

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

```

P = {
  S → E S
  S → E
  E → enum identificador { L } ;
  L → L, I
  L → I
  I → identificador
  I → identificador = 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, amarillo = 9, rojo };

Formas normales

13. Dada la siguiente gramática de contexto libre G:

```

P = {
  S → T L ;
  T → int
  T → float
  L → identificador L'
  L' → , identificador L'
  L' → ε
}

```

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