

# Serie de Análisis Sintáctico.

Semestre 2015-1

## Compiladores

### Diseño de Gramáticas libres de contexto

1. Diseñe una gramática libre de contexto para el lenguaje de todas las cadenas de  $a$  y  $b$ , de tal forma que justo antes de cada  $a$  vaya por lo menos una  $b$ .
2. Diseñe una gramática de contexto libre para el lenguaje de todas las cadenas de  $a$  y  $b$  donde el número de  $a$ 's es igual a número de  $b$ 's.
3. Diseñe una gramática de contexto libre que genera el lenguaje de los paréntesis y corchetes balanceados. Ejemplo:  $([[[()[(())]]])$

### Derivaciones y árboles de derivación

1. Dada la siguiente gramática libre de contexto

$$S \rightarrow SS + | SS * | a$$

y la cadena  $aa + a*$

- a) Proporcione una derivación por la derecha para la sentencia.
  - b) Proporcione una derivación por la izquierda para la sentencia.
  - c) Proporcione el árbol de análisis sintáctico para la sentencia.
  - d) Diga si la gramática es ambigua o no.
2. Dada la siguiente gramática libre de contexto

$$S \rightarrow S(S)S|\varepsilon$$

y la cadena  $((()))$

- a) Proporcione una derivación por la derecha para la sentencia.
  - b) Proporcione una derivación por la izquierda para la sentencia.
  - c) Proporcione el árbol de análisis sintáctico para la sentencia.
  - d) Diga si la gramática es ambigua o no.
3. Dada la siguiente gramática libre de contexto

$$\begin{aligned} \text{lexp} &\rightarrow \text{atom} \mid \text{list} \\ \text{atom} &\rightarrow \text{num} \mid \text{id} \\ \text{list} &\rightarrow (\text{lexp-sec}) \\ \text{lexp-sec} &\rightarrow \text{lexp-sec lexp} \mid \text{lexp} \end{aligned}$$

y la cadena  $(a23(mxy))$

- a) Proporcione una derivación por la derecha para la sentencia.
- b) Proporcione una derivación por la izquierda para la sentencia.
- c) Proporcione el árbol de análisis sintáctico para la sentencia.
- d) Diga si la gramática es ambigua o no.

## Ambigüedad

Usando la precedencia y asociatividad de operadores, eliminar la ambigüedad en las siguientes gramáticas.

1.  $S \rightarrow S + S | SS | (S) | S * | letra$
2.  $B \rightarrow B \text{ and } B | B \text{ or } B | \text{not } B | (B) | \text{true} | \text{false}$
3. **Hint:** Considere a *if* y *else* como operadores.

$$\begin{aligned} stmt &\rightarrow stmt\_if | otro | \varepsilon \\ stmt\_if &\rightarrow if( exp ) stmt | if(exp) stmt else stmt \\ exp &\rightarrow 0 | 1 \end{aligned}$$

## Notación EBNF y Diagramas de sintaxis

Representar las siguientes gramáticas en su notación EBNF y obtener los diagramas de sintaxis respectivos.

1.  $\begin{aligned} bexp &\rightarrow bexp' | 'bterm | bterm \\ bterm &\rightarrow bterm \& \& bfact | bfact \\ bfact &\rightarrow !bfact | (bexp) | \text{true} | \text{false} \end{aligned}$
2.  $\begin{aligned} decls &\rightarrow decls decl | decl \\ decl &\rightarrow type \text{ var-list } | \varepsilon \\ type &\rightarrow \text{int} | \text{float} | \text{char} \\ \text{var-list} &\rightarrow \text{id}, \text{var-list} | \text{id} \end{aligned}$
3.  $\begin{aligned} program &\rightarrow decls stmt \\ decls &\rightarrow decls decl ; | \varepsilon \\ stmt &\rightarrow asig ; | \varepsilon \\ asig &\rightarrow \text{id} = part \\ part &\rightarrow expresion \\ expresion &\rightarrow expresion + expresion | expresion - expresion | \text{id} | num \end{aligned}$