Gramáticas libres de contexto.

Contenido

- Introducción
- Definición Gramáticas tipo 2: Independientes o libres del contexto
- Árboles de derivación
- Ejercicios

Introducción.

- Retomando la definición formal de gramáticas:

Se llama gramática formal definida sobre un alfabeto $\sum$ a una tupla de la forma:

$$G = \{ \textstyle\sum_T, \sum_N, S, P \}$$

donde :

$\sum_T$ es el alfabeto de símbolos terminales

$\sum_N$ es el alfabeto de símbolos no terminales

(aparecen en los ejemplos encerrados entre **<>**)

S es el símbolo inicial de la gramática

P es un conjunto de producciones gramaticales

Gramáticas libres o independientes del contexto GIC (Tipo 2).

- En estas gramáticas, la parte izquierda de las producciones solo puede tener un símbolo no terminal:

$$P = \{ (S ::= \lambda) \text{ ó } (A ::= v) \mid S, A \in \Sigma_N, v \in \Sigma^+ \}$$

- Estas gramáticas se denominan de contexto libre, porque a la hora de transformar una palabra en otra, **el símbolo no terminal que se transforma no depende de los que están a la izquierda o derecha**. Así, cuando se realicen derivaciones en las que se transforme el símbolo **A** de la fórmula anterior no hace falta saber qué hay alrededor de él.

Gramática tipo 2: GIC

- Ejemplo 1:

$\mathbf{G_1}$ = ({A, B}, { 0, 1}, A, { **A → 1B1 | 11**, **B → 0|1** })

Esta gramática describe el lenguaje:

$\mathbf{L_1}$ = { 11, 101, 111 }

Gramática tipo 2. GIC

- Las gramáticas independientes del contexto generan lenguajes libres o independientes del contexto.
- Los lenguajes libres del contexto pueden ser reconocidos por un **autómata de pila** determinístico o no determinístico.

Gramática tipo 2. GIC

- Ejemplo 2. Encontrar una GIC que genere el lenguaje **a^+b^+**
- $G = \{ \sum_T, \sum_N, S, P \}$ $\sum_N = \{ a, b \}$

$\sum_N = \{S,A,B\}$
P = {
$S \longrightarrow aA$
$A \longrightarrow aA \mid B$
$B \longrightarrow bB \mid b$
}

$\sum_N = \{S,A,B\}$
P = {
$S \longrightarrow AB$
$A \longrightarrow aA \mid a$
$B \longrightarrow bB \mid b$
}

$\sum_N = \{S,A\}$
P = {
$S \longrightarrow aS \mid aA$
$A \longrightarrow bA \mid b$
}

Gramática tipo 2. GIC

- Ejemplo 3. Diseñar una GIC para modelar expresiones en lenguaje C.
- Usaremos:
  - Operadores: *****, **+**
  - Identificadores de variable: **a**, **b**
  - Paréntesis: **(**, **)**
  - Dígitos: **0**, **1**
- Ejemplo de expresión: **a*(a+b00)**

Ejemplo 3. Una GIC para expresiones

***G* = { Σ_T, Σ_N, *S*, *P* }**

Σ_T = { +, *, (,), a, b, 0, 1 }

Σ_N = { E, ID } (**E** usado para expresión; **ID** para identificador)

P = {

E ⟶ ID | E+E | E*E | (E)

ID ⟶ a | b | IDa | IDb | ID0 | ID1

}

Ejemplo 3. Una GIC para expresiones

P = {

E ⟶ ID | E+E | E*E | (E)

ID ⟶ a | b | IDa | IDb | ID0 | ID1

}

Derivación de a*(a+b00)

E ⟶ E*E ⟶ ID*E ⟶ a*E ⟶ a*(E) ⟶ a*(E+E) ⟶ a*(ID+E) ⟶ a*(a+E) ⟶ a *(a+ID) ⟶ a*(a+ID0) ⟶ a*(a+ID00) ⟶ a*(a+b00)

Ejemplo 3. Una GIC para expresiones

P = { E ⟶ ID | E+E | E*E | (E), ID ⟶ a | b | IDa | IDb | ID0 | ID1 }

Derivación de a*(a+b00) por la izquierda:

E → E*E → ID*E → a*E → a*(E) → a*(E+E) → a*(ID+E) → a*(a+E) →
a *(a+ID) → a*(a+ID0) → a*(a+ID00) → a*(a+b00)

Derivación de a*(a+b00) por la derecha:

E → E*E → E*(E) → E*(E+E) → E*(E+ID) → E*(E+ID0) → E*(E+ID00) →
E*(E+b00) → E*(ID+b00) → E*(a+b00) → ID*(a+b00)→⇒ a*(a+b00)

Árboles de derivación

- Son una forma de representar las derivaciones, siendo utilizados, por ejemplo, en la **construcción de compiladores** para representar el análisis sintáctico de los programas fuente y sirven de base para la **generación de código**.

- Sólo se pueden definir árboles de derivación para gramáticas de tipo 1 o más restrictivas (tipos 2 y 3).

Árboles de derivación

- Para construir un árbol de derivación:
  - El axioma se representa en la raíz del árbol.
  - Los nodos hojas del árbol son símbolos terminales de la gramática.
  - Los nodos intermedios son símbolos no terminales de la gramática.
  - Las derivaciones se representan creando tantos sucesores del símbolo no terminal de la izquierda de las producciones como símbolos (terminales y no terminales) aparezcan en la parte derecha de las producciones.

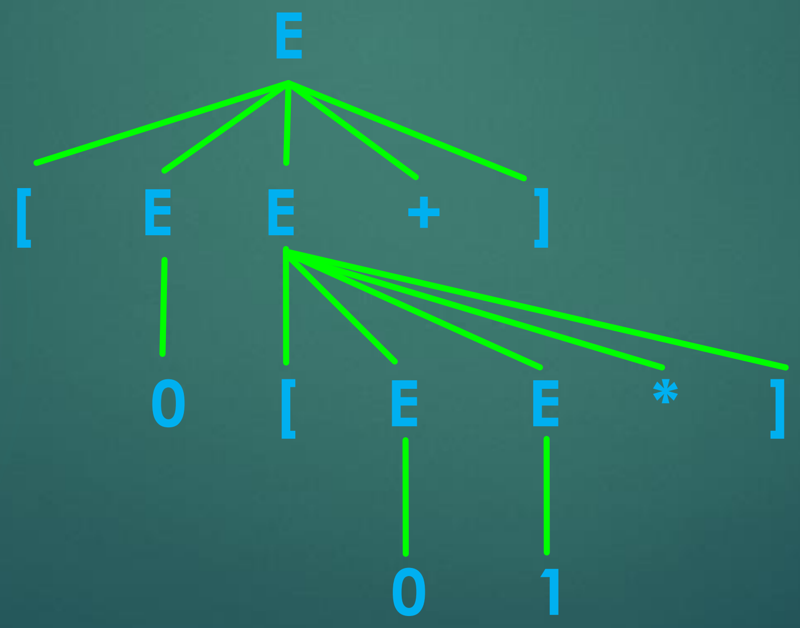
Árboles de derivación. Ejemplo

- Supóngase la siguiente gramática:

G = ({ 0, 1, [,], +, * }, { E }, E, P)

P = { E → [EE+] | [EE*] | 0 | 1 }

Dibujar el árbol de derivación para la palabra **[0 [0 1 *] +]**



Ejercicios. Realizar el árbol de derivación para las siguientes gramáticas.

1) ***G = { $\sum_T$, $\sum_N$, S, P }***

$\sum_T$ = { +, *, (,), a, b, 0, 1 }

$\sum_N$ = { E, ID } (**E** usado para expresión; **ID** para identificador)

P = {

E ⟶ ID | E+E | E*E | (E)

ID ⟶ a | b | IDa | IDb | ID0 | ID1

}

Palabra: a*(a+b00)

Derivar dos palabras más y dibujar sus árboles correspondientes.

Ejercicios. Realizar el árbol de derivación para las siguientes gramáticas.

2) G=({a,b},{S}, S, P}

P= {

$S \rightarrow aSa$

$S \rightarrow bSb$

$S \rightarrow \varepsilon$

}

Derivar dos palabras y dibujar sus árboles correspondientes.

Ejercicios. Realizar el árbol de derivación para las siguientes gramáticas.

3) $G=(\{a,b\},\{S, A\}, S, P\}$

$P= \{$

$S \to bSbb \mid A$

$A \to aA \mid \varepsilon$

$\}$

Derivar dos palabras y dibujar sus árboles correspondientes.

Ejercicios. Realizar el árbol de derivación para las siguientes gramáticas.

4) G=({a,b,c},{A, B, E, C, F}, A, P}

P= {

$A \rightarrow BF$

$B \rightarrow EC$

$E \rightarrow a$

$C \rightarrow b$

$F \rightarrow c$

}

Derivar tres palabras y dibujar sus árboles correspondientes.