

# INF155: Informática Teórica

## Clase 13: Árbol de Derivación

Aldo Berrios Valenzuela

Martes 19 de Abril de 2016

### 1. Árboles de Derivación

---

**Definición 1.1.** /\* Colocar la definición de derivación \*/

**Definición 1.2.** Sea  $G = (N, \Sigma, P, S)$  un CFG. El *árbol de derivación* de  $\sigma \in \mathcal{L}(G)$ . Se construye poniendo el símbolo de partida como raíz, y bajo cada no-terminal los símbolos que deriva (en orden).

**Ejemplo 1.1.** Consideremos la siguiente gramática:

$$E \rightarrow E + T \quad (1)$$

$$E \rightarrow T \quad (2)$$

$$T \rightarrow T * F \quad (3)$$

$$T \rightarrow F \quad (4)$$

$$F \rightarrow (E) \quad (5)$$

$$F \rightarrow a \quad (6)$$

Queremos ver los pasos por los que tenemos que pasar para llegar a  $a * (a + a)$  (es sólo un ejemplo). Para ello considere lo siguiente:

- Colocamos el no-terminal inicial que nos ayude a armar el cuento.
- Anotaremos la producción que usaremos en el lado derecho de la ecuación para no perdernos.
- Subrayamos el no-terminal que vamos a reemplazar.
- Reemplazamos.

Entonces:

$$E \Rightarrow \underline{T} \quad (2)$$

$$\Rightarrow T * \underline{F} \quad (3)$$

$$\Rightarrow T * (\underline{E}) \quad (5)$$

$$\Rightarrow T * (\underline{E} + T) \quad (1)$$

$$\Rightarrow T * (T + \underline{T}) \quad (2)$$

$$\Rightarrow T * (T + \underline{F}) \quad (4)$$

$$\Rightarrow \underline{T} * (T + a) \quad (6)$$

$$\Rightarrow F * (\underline{T} + a) \quad (4)$$

$$\Rightarrow F * (\underline{F} + a) \quad (4)$$

$$\Rightarrow \underline{F} * (a + a) \quad (6)$$

$$\Rightarrow a * (a + a) \quad (6)$$

Luego, el árbol de derivación del caso sería:

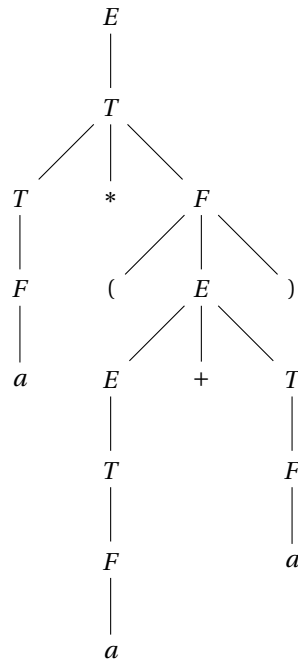


Figura 1: Árbol de Derivación resultante de las *relaciones de derivaciones* ( $\Rightarrow$ ) aplicadas en el ejemplo anterior.

**Definición 1.3** (Derivación de extrema izquierda). En cada iteración aplicamos la producción al no-terminal del extremo izquierdo. Para la gramática del ejemplo 1.1, la derivación de extrema izquierda es:

$$\begin{aligned}
 E &\Rightarrow \underline{T} \\
 &\Rightarrow \underline{T} * F \\
 &\Rightarrow \underline{F} * F \\
 &\Rightarrow a * \underline{F} \\
 &\Rightarrow a * (\underline{E}) \\
 &\Rightarrow a * (\underline{E} + T) \\
 &\Rightarrow a * (\underline{T} + T) \\
 &\Rightarrow a * (\underline{F} + T) \\
 &\Rightarrow a * (a + \underline{T}) \\
 &\Rightarrow a * (a + \underline{F}) \\
 &\Rightarrow a * (a + a)
 \end{aligned}$$

**Definición 1.4** (Derivación de extrema derecha). En cada iteración, aplicamos la producción al no-terminal del extremo derecho. Para la gramática del ejemplo 1.1, la derivación de extrema derecha es:

$$\begin{aligned}
 E &\Rightarrow \underline{T} \\
 &\Rightarrow T * \underline{F} \\
 &\Rightarrow T * (\underline{E}) \\
 &\Rightarrow T * (E + \underline{T}) \\
 &\Rightarrow T * (E + \underline{F}) \\
 &\Rightarrow T * (\underline{E} + a) \\
 &\Rightarrow T * (\underline{T} + a) \\
 &\Rightarrow T * (\underline{F} + a) \\
 &\Rightarrow \underline{T} * (a + a) \\
 &\Rightarrow \underline{F} * (a + a) \\
 &\Rightarrow a * (a + a)
 \end{aligned}$$

Algunas observaciones:

- Si podemos derivar una palabra, entonces podemos construir un árbol de derivación.
- Para encontrar la palabra del árbol de derivación debemos recorrerlo en **post-orden** (más detalles en sus apuntes de estructura de datos). Esta serán los no-terminales en el orden que los vayamos encontrando.
- Sólo en las gramáticas de contexto libre podemos aplicar las producciones en distinto orden (extrema izquierda o derecha por ejemplo), ya que son libres al contexto (valga la redundancia). Como es de esperar, en un lenguaje sensible al contexto esto no ocurre.

Árbol de derivación  $\equiv$  derivación de extrema izquierda  $\equiv$  derivación de extrema derecha (son equivalentes).

$\rightsquigarrow$  Colección de derivaciones que “hacen lo mismo”

**Definición 1.5** (Gramática Ambigua). Sea  $G = (N, \Sigma, P, S)$  una CFG. Si hay  $\sigma \in \mathcal{L}(G)$  con dos árboles de derivación distintos, se dice que  $G$  es ambigua.

**Ejemplo 1.2.** Consideremos la gramática  $G = (N, \Sigma, P, E)$  con el siguiente conjunto de producciones:

$$\begin{aligned}
 E &\rightarrow E + E & (1) \\
 E &\rightarrow E * E & (2) \\
 E &\rightarrow (E) & (3) \\
 E &\rightarrow a & (4)
 \end{aligned}$$

Supongamos además, que queremos formar la palabra  $a * a + a$  con la gramática  $G$ . Si comenzamos a derivar usando (1), obtendremos el árbol de la Figura 2a; sin embargo, si empezamos a derivar usando (2), el árbol de derivación resultante sería el de la Figura 2b.

La Figura 2, deja en manifiesto que para aplicaciones prácticas se debe tener cuidado con los árboles de derivación. Si la gramática ambigua, entonces no sirve (por ejemplo, una palabra con varios significados posibles... considere la gramática del ejemplo 1.2)

**Ejercicio 1.1.** Demuestre que existen lenguajes regulares generados por gramáticas de tipo 2 y que no son de tipo 3.

*Demostración.* Sea  $G = (N, \Sigma, P, S)$  con las siguientes producciones:

$$\begin{aligned}
 S &\rightarrow aSb \\
 S &\rightarrow ab \\
 S &\rightarrow aa \\
 S &\rightarrow ba \\
 S &\rightarrow bb
 \end{aligned}$$

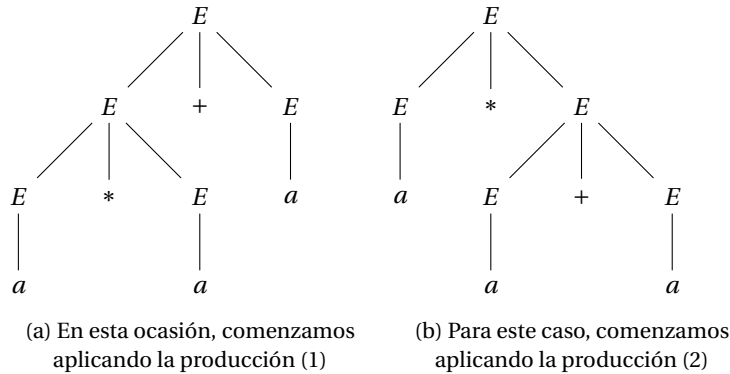


Figura 2: Gramática ambigua del ejemplo 1.2.

La gramática anterior es el lenguaje regular  $\mathcal{L}(a^*(a|b)(b|a)b^*)$ . □

**Ejercicio 1.2.** Demuestre que el lenguaje generado por la gramática

$$E \rightarrow E + E \quad (1)$$

$$E \rightarrow E * E \quad (2)$$

$$E \rightarrow (E) \quad (3)$$

$$E \rightarrow a \quad (4)$$

no es regular.

Algunos comentarios al margen antes de terminar:

- Determinar si dos lenguajes de contexto libre generan el mismo lenguaje es imposible
- La libertad de cómo aplicar las producciones en una CFG, se representa a través del árbol de derivación

## 2. Forma Normal de Chomsky (CNF)

**Definición 2.1.** Una CFG  $G = (N, \Sigma, P, S)$  se dice en *forma normal de Chomsky* si todas sus producciones tienen las formas:

$$A \rightarrow BC \quad A, B, C \in N$$

$$A \rightarrow a \quad A \in N, a \in \Sigma$$

La forma normal de Chomsky (CNF) nos sirve para realizar demostraciones con gramáticas, ya que es mucho más sencillo si hay pocos tipos de lados derechos. /\* Tengo la duda en “si hay pocos tipos de lados derechos”... no entiendo esa parte. Considero que estaba mejor con “si sólo existe 1 árbol de derivación para la CFG.” \*/

**Teorema 2.1.** Sea  $G$  una CFG. Entonces puede construirse una CFG en CNF  $G'$  tal que  $\mathcal{L}(G) = \mathcal{L}(G')$ .

*Demostración.* Consideremos:

- Producciones:

$$A \rightarrow \alpha$$

con  $|\alpha| \geq 2$ , contiene terminales y no-terminales.

Inventar nuevos no-terminales  $A_a \rightarrow a$ , para todo  $a \in \Sigma$ , y reemplazar  $a$  por  $A_a$ .

- Producciones unitarias:  $A \rightarrow B$

Podemos identificar todos los no-terminales con producciones unitarias ( $A$ ) y sus producciones ( $A \rightarrow B$ ).

Para cada cadena,  $A \rightarrow B, B \rightarrow C, \dots, U \rightarrow V$  y  $V \rightarrow \alpha$ , con  $|\alpha| \geq 2$ , podemos reemplazar  $A \rightarrow \alpha, B \rightarrow \alpha, \dots$

- Producciones:

$$A \rightarrow B_1 B_2 \cdots B_n$$

Nuevos no-terminales  $X, \dots, Y$ :

$$A \rightarrow B_1 X_1$$

$$X_1 \rightarrow B_2 X_2$$

$$\vdots$$

$$X_{n-2} \rightarrow B_{n-1} B_n$$

□