

# Teoría de Autómatas y Lenguajes Formales

## Unidad 2 — CFG: derivaciones, árboles, ambigüedad y desambiguación (con práctica)

Docente: Helder Octavio Fernández Guzmán

- Revisión operativa: **derivación** vs **árbol de derivación**.
- **Leftmost** vs **rightmost**: utilidad práctica (evidencia estandarizada).
- **Ambigüedad**: evidencia correcta  $\Rightarrow$  **2 árboles** distintos.
- **Desambiguación (técnica distinta)**: forzar estructura con **bloques + lista**.
- **Quiz corto** (antes de practicar): 8–10 min.
- **Cierre**: práctica del estudiante (actividad guiada en clase).

# Recordatorio mínimo: evidencia (IN / NOT IN)

Periodo 1/6

- Derivación (un paso): si  $A \rightarrow \alpha$ , entonces  $xAy \Rightarrow x\alpha y$ .
- Derivación (cero o más pasos):  $\Rightarrow^*$ .
- **IN** requiere evidencia: derivación o árbol.
- **NOT IN** requiere argumento estructural breve (2–4 líneas).

## Criterio correcto

Una CFG es **ambigua** si existe una cadena  $w$  con **dos árboles de derivación distintos**.

- **Sí prueba:** dos árboles distintos para el mismo  $w$ .
- **No basta:** “dos derivaciones diferentes” (pueden ser dos órdenes de expansión del *mismo* árbol).

Derivación:  $S \Rightarrow^* w$

Árbol de derivación

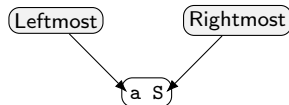


Ambos sirven como evidencia (IN), pero la ambigüedad se decide con árboles.

# Leftmost vs Rightmost (utilidad real)

Periodo 2/6

- **Leftmost:** expandir siempre el **no terminal** más a la izquierda.
- **Rightmost:** expandir siempre el **no terminal** más a la derecha.
- Utilidad: estandarizan el procedimiento de derivación (evidencia **reproducible**).
- Importante: leftmost/rightmost **no** crean ambigüedad; solo cambian el orden de expansión.



**Forma sentencial:** mezcla terminales y no terminales

Aquí solo hay un no terminal: siempre se expande S.

# G1: caso limpio para practicar derivaciones

Periodo 2/6

Gramática  $G_1$

$$S \rightarrow aS \mid a$$

Cadena objetivo

$$w = aaaa$$

- Derivación leftmost: escribir pasos hasta llegar a solo terminales.
- Derivación rightmost: mismo  $w$ , (en este caso) mismo orden por estructura simple.

# G1: derivación para aaaa

Periodo 2/6

$$\begin{aligned} S &\Rightarrow aS \\ &\Rightarrow aaS \\ &\Rightarrow aaaS \\ &\Rightarrow aaaa \end{aligned}$$

## Lectura operativa

Cada paso expande el no terminal disponible. El cierre ocurre al aplicar  $S \Rightarrow a$ .

# Ambigüedad sin operadores: concatenación pura (G2)

Periodo 3/6

Gramática ambigua  $G_2$

$$S \rightarrow SS \mid a$$

Cadena fija

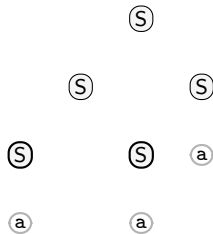
$$w = aaa$$

**Idea:** el problema es el reagrupamiento posible por  $S \rightarrow SS$ .



## G2: Árbol A para aaa

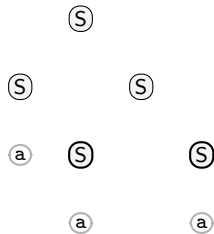
Periodo 3/6



**Agrupación:  $(SS)S$ .**

## G2: Árbol B para aaa

Periodo 3/6



**Agrupación:  $S(SS)$ .**

## Conclusión: evidencia de ambigüedad ( $G_2$ )

Periodo 3/6

- Para  $w = aaa$  mostramos **dos árboles distintos**.
- Por el criterio práctico,  $G_2$  es **ambigua**.

### Mensaje clave

La ambigüedad aparece por **reagrupamiento estructural** (no hace falta hablar de significado).

# Desambiguación (técnica distinta): problema $\rightarrow$ bloque $\rightarrow$ lista

Periodo 4/6

## Paso 1: identificar la fuente

En  $G_2$ ,  $S \rightarrow SS$  permite agrupar como  $(SS)S$  o  $S(SS)$ .

## Paso 2: introducir una unidad (bloque)

Definimos un bloque fijo:  $B \rightarrow aa$ .

## Paso 3: forzar concatenación única (lista de bloques)

Modelamos “uno o más bloques”:  $S \rightarrow BS \mid B$ .

## Gramática $G_3$

$$S \rightarrow BS \mid B \quad B \rightarrow aa$$

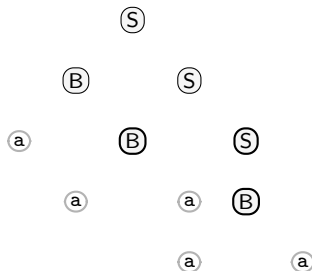
## Cadena objetivo

$$w = aaaaaa \quad (\text{tres bloques } aa)$$

**Idea:** la estructura queda determinada por la lista de bloques.

G3: árbol único para aaaaaa

Periodo 4/6



**Lectura:**  $S \Rightarrow BS \Rightarrow BBS \Rightarrow BBB$  (y cada  $B$  produce aa).

# Comparación breve: niveles vs bloques

Periodo 4/6

## Niveles (E/T/F)

- Útil para **operadores** (precedencia/asociatividad).
- Cambia la **jerarquía** de combinaciones.

## Bloques + lista (G3)

- Útil para **concatenación** y “segmentación”.
- Fuerza estructura al definir **unidades** ( $B$ ) y su repetición.

Ambas técnicas buscan un resultado: **una sola estructura válida**

# NOT IN: ejemplo A (argumento estructural por terminales)

Periodo 5/6

## Gramática $G_1$

$$S \rightarrow aS \mid a$$

## Cadena candidata

$$w = \text{aaaaaab}$$

## Argumento estructural (breve)

- 1 En  $G_1$ , las producciones solo introducen la terminal  $a$ .
- 2 Por tanto, toda cadena generada por  $G_1$  tiene la forma  $a^n$  con  $n \geq 1$ .
- 3 La cadena  $\text{aaaaaab}$  contiene el símbolo  $b$ .
- 4 Como  $b$  no puede ser producido por ninguna regla,  $\text{aaaaaab} \notin L(G_1)$ .



# NOT IN: ejemplo B (argumento estructural por balanceo)

Periodo 5/6

Gramática  $G_4$

$$S \rightarrow SS \mid [S] \mid x$$

Cadena candidata

$$w = [x$$

Argumento estructural (breve)

- 1 En  $G_4$ , el único modo de introducir corchetes es  $[S]$ , que agrega siempre un  $[$  y un  $]$  como pareja.
- 2 La concatenación  $SS$  solo concatena cadenas completas; no crea ni elimina corchetes.
- 3 Entonces, toda cadena en  $L(G_4)$  tiene corchetes balanceados: mismo número de  $[$  y  $]$ .
- 4 La cadena  $[x$  tiene un  $[$  sin su  $]$ .

# Quiz corto (antes de practicar)

Periodo 5/6

- Duración: **8–10 minutos**.
- Total: **10 preguntas** (mixtas: V/F + selección + 1 mini-árbol).
- Enfoque:
  - evidencia correcta de ambigüedad (2 árboles),
  - diferenciar derivación vs árbol,
  - reconocer leftmost/rightmost,
  - idea de “ajuste mínimo” (bloques + lista).

## Revisión express: errores típicos (2–3 min)

Periodo 5/6

- Confundir “**dos derivaciones**” con “**dos árboles**”.
- Entregar IN sin evidencia (ni derivación ni árbol).
- En desambiguación: cambiar símbolos sin cambiar la **estructura** (no resuelve).

# Práctica del estudiante (cierre): gramática nueva con corchetes (G4)

Periodo 6/6

Gramática  $G_4$  (ambigua)

$$S \rightarrow SS \mid [S] \mid x$$

Cadena fija

$$w = [[x]] [x] [x]$$

**Pista estructural:** piensa en “bloques” naturales:  $[[x]]$ ,  $[x]$ ,  $[x]$ .

- 1 Dos árboles distintos para  $w$  (evidencia de ambigüedad).
- 2 Una derivación **leftmost** completa para  $w$ .
- 3 **Ajuste mínimo** (propuesta): 3–5 líneas explicando cómo reducir el reagrupamiento por  $SS$ .

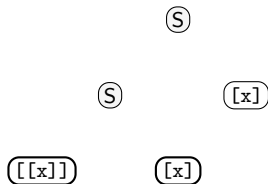
Checklist antes de entregar

- ☐ 2 árboles distintos
  - ☐ 1 leftmost completa
    - ☐ Ajuste mínimo (3–5 líneas)

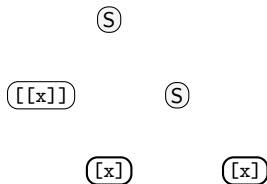
# Guía visual (sin “spoiler” completo): dos agrupaciones posibles

Periodo 6/6

**Agrupación A**



**Agrupación B**



Completa cada bloque usando la regla [S] hasta llegar a x.