

## 4. Gramáticas Independientes del Contexto

Copyright (c) 2025 Adrián Quiroga Linares Lectura y referencia permitidas; reutilización y plagio prohibidos

Una **gramática** es un conjunto de reglas que nos permite generar todas las palabras válidas de un lenguaje. Es como un "recetario" para crear frases correctas.

### 4.1 Definición Formal:

Una gramática se define siempre como:

$$G = (V, T, P, S)$$

1.  **$V$  (Variables/No Terminales):** Los "ingredientes intermedios". Se escriben en MAYÚSCULAS ( $S, A, B$ ).
2.  **$T$  (Terminales):** El "plato final". Símbolos que forman las cadenas reales ( $a, b, 0, 1$ ). **Nunca** aparecen a la izquierda de una flecha en gramáticas estándar (Context Free).
3.  **$P$  (Producciones):** Las reglas de sustitución. Estructura: Cabeza  $\rightarrow$  Cuerpo.
4.  **$S$  (Axioma):** El estado inicial (siempre pertenece a  $V$ ).

### 4.2 La Jerarquía de Chomsky

Chomsky clasificó las gramáticas según **cuán estrictas son sus reglas**.

- **Regla de oro:** Cuanto mayor es el número ( $0 \rightarrow 3$ ), **más restricciones** tiene y **menos potente** es.

Tipo	Nombre	¿Qué ves a la IZQUIERDA de la flecha?	La Regla de Oro (En español)	Ejemplo
------	--------	---------------------------------------	------------------------------	---------

Tipo 0	<b>Sin Restricciones</b>	<b>Cualquier cosa.</b>	Vale todo. Puedes borrar, cambiar, mezclar... Caos total.	$aAb \rightarrow c$
--------	--------------------------	------------------------	---	---------------------

(3 cosas se convierten en 1. Se puede encoger).

<b>Tipo</b>	<b>Nombre</b>	<b>¿Qué ves a la IZQUIERDA de la flecha?</b>	<b>La Regla de Oro (En español)</b>	<b>Ejemplo</b>
<b>Tipo 1</b>	<b>Sensible al Contexto</b>	<b>Un grupo de cosas.</b>  <i>(Contexto)</i>	<b>"No puedes encoger".</b>  Lo que entra debe ser igual o menor a lo que sale. Necesitas "vecinos" para cambiar.	<b>xAy → xBy</b>  <i>(La 'A' cambia a 'B' solo si está entre x e y).</i>
<b>Tipo 2</b>	<b>Indep. del Contexto (GIC)</b>	<b>UNA SOLA variable.</b>  <i>(Mayúscula)</i>	<b>"Sustitución simple".</b>  Cambias una variable por lo que quieras, sin importar qué tenga al lado.	<b>A → aB</b> ó <b>A → a</b>  <i>(Siempre que veas una A, cámbiala).</i>
<b>Tipo 3</b>	<b>Regular</b>	<b>UNA SOLA variable.</b>  <i>(Mayúscula)</i>	<b>"Cola india".</b>  Muy rígido. Solo puedes poner un terminal y (opcional) una variable al final.	<b>A → aB</b> ó <b>A → a</b>  <i>(La variable siempre va a un extremo).</i>

**Nota:** Tipo 3 ⊂ Tipo 2 ⊂ Tipo 1 ⊂ Tipo 0. *Toda gramática regular es independiente del contexto, pero no al revés.*

## 4.3. Gramáticas Regulares (Tipo 3)

Son las que generan los Lenguajes Regulares (los mismos que vimos en el Tema 2 y 3).

### Estructura Rígida:

- Lineal Derecha:**  $A \rightarrow \text{terminal} \cdot \text{Variable}$  (ej:  $A \rightarrow aB$ ) ó  $A \rightarrow \text{terminal}$  ( $A \rightarrow a$ ).
- Lineal Izquierda:**  $A \rightarrow \text{Variable} \cdot \text{terminal}$  (ej:  $A \rightarrow Ba$ ).

**¡Ojo!** No puedes mezclar reglas lineales por derecha e izquierda en la misma gramática. Si lo haces, se convierte en Tipo 2 (GIC) y deja de ser Regular.

## 4.4 Lenguaje de una Gramática

$$L(G) = \{w \in T^* \mid S \Rightarrow *w\}$$

Esto significa: El lenguaje generado por  $G$  es el conjunto de todas las palabras formadas **solo por terminales** que se pueden derivar desde  $S$ .

### Ejemplo:

$$S \rightarrow aSb \mid \epsilon$$

$$T = \{a, b\}$$

### Derivaciones:

- $S \Rightarrow \epsilon \rightarrow$  palabra: "" (cadena vacía)
- $S \Rightarrow aSb \Rightarrow ab \rightarrow$  palabra: "ab"
- $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb \rightarrow$  palabra: "aabb"
- $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabb \rightarrow$  palabra: "aaabb"

$$L(G) = \{a^n b^n \mid n \geq 0\} = \{\epsilon, ab, aabb, aaabb, \dots\}$$

## 4.5 Árboles de Derivación vs Derivaciones

Es importante distinguir entre el **proceso** y la **estructura**.

1. **Derivación (Texto):** Es la secuencia de pasos paso a paso.

- $S \Rightarrow aA \Rightarrow abB \Rightarrow abb$

2. **Árbol de Derivación (Gráfico):** Es la estructura jerárquica.

- Raíz:  $S$ .
- Hojas: La palabra final ( $a, b$ ,  $b$ ).
- Nodos: Variables intermedias.

## 4.6 Ambigüedad

**Definición de Examen:** Una gramática es ambigua si existe **al menos una cadena** que tiene **dos o más árboles de derivación distintos**.

**Caso típico:** Operaciones matemáticas sin paréntesis ni precedencia.

- $3 + 4 * 5 \rightarrow$  ¿Es  $(3+4)*5$  o  $3+(4*5)$ ? Si la gramática permite ambos árboles, es mala (ambigua).

## 4.7 Protips para ejercicios

### Patrón 1: El Espejo y la Cebolla (Palíndromos y $a^n b^n$ )

Si necesitas que el principio coincida con el final, o que la cantidad de letras del principio sea igual a la del final, usas la **recursividad envolvente**.

La Regla de Oro:

$$S \rightarrow x S y$$

Esto genera  $x$  a la izquierda y  $y$  a la derecha, sincronizados.

### Ejemplo Práctico Palíndromo:

1. Si añado una 'a' al principio, debo añadir una 'a' al final:  $S \rightarrow aSa$
2. Si añado una 'b' al principio, debo añadir una 'b' al final:  $S \rightarrow bSb$
3. ¿Cómo termino? Con el centro. Puede ser 'a', 'b', o vacío ( $\varepsilon$ ):  $S \rightarrow a | b | \varepsilon$

**Gramática final:**  $S \rightarrow aSa | bSb | a | b | \varepsilon$

## Patrón 2: Ecuaciones Lineales ( $k = i + j$ o $k = i + 2j$ )

Estos ejercicios te piden relacionar contadores.

- **Truco:** Traduce la ecuación a "**quién consume a quién**".
- **Orden:** Fíjate MUY bien en el orden de las letras en el alfabeto ( $a^i b^j c^k$ ).

**Caso A:** Suma simple ( $k = i + j$  en  $a^i b^j c^k$ )

Significa que por cada 'a' hay una 'c', Y por cada 'b' hay una 'c'. Pero las 'a' están lejos de las 'c'.

- Solución: Anidamiento. Tratamos la cadena como  $a^i(b^j c^j)c^i$ .
- Las 'a' envuelven a todo el bloque de 'b' y 'c'.
- Las 'b' se emparejan con las 'c' del medio.

Caso B: Multiplicación ( $k = i + 2j$  en  $a^i b^j c^k$ )

- Interpretación:
  - Por cada 1 'a', genero 1 'c'. (Exterior)
  - Por cada 1 'b', genero 2 'c's. (Interior)
- Diseño:
  1. Estado inicial ( $S$ ): Genera 'a' izquierda y 'c' derecha. Cuando acaben las 'a', pasamos al bloque central ( $B$ ).

$$S \rightarrow aSc | B$$

2. Estado central ( $B$ ): Genera 'b' izquierda y dos 'c' derecha.

$$B \rightarrow bBcc | \varepsilon$$

## Patrón 3: La "O" Lógica (Unión de Casos)

Si ves un "o", una coma, o condiciones alternativas ( $i = j$  o  $j = k$ ), **NO** intentes hacerlo todo en una sola regla. Divide y vencerás.

**Estrategia:** El símbolo inicial solo sirve para elegir camino.

$$S \rightarrow S_1 \mid S_2$$

**Ejemplo** ( $a^i b^j c^k$  donde  $i = j$  O  $j = k$ ):

- Camino 1 ( $S_1$ ):  $i = j$  (y  $k$  va por libre). Cadena tipo  $a^n b^n c^m$ .
  - Empareja 'a' y 'b'. La 'c' se genera aparte libremente.
  - $S_1 \rightarrow XC$
  - $X \rightarrow aXb \mid \epsilon$  (pareja a-b)
  - $C \rightarrow cC \mid \epsilon$  (c libre)
- Camino 2 ( $S_2$ ):  $j = k$  (y  $i$  va por libre). Cadena tipo  $a^m b^n c^n$ .
  - La 'a' va libre al principio. Luego empareja 'b' y 'c'.
  - $S_2 \rightarrow AY$
  - $A \rightarrow aA \mid \epsilon$  (a libre)
  - $Y \rightarrow bYc \mid \epsilon$  (pareja b-c)

## Patrón 4: Desigualdades (El Truco del "Sobra Algo")

Las gramáticas no saben hacer "mayor que". Solo saben hacer "igual". **Truco Matemático:**

- $k > i \rightarrow$  significa  $k = i + m$  (donde  $m \geq 1$ ).
- Es decir: "Hay tantas 'k' como 'i', y luego **sobran** más 'k'".

**Ejemplo** ( $a^i(b+c)^k$  donde  $k > i$ ): Vamos a simplificar  $(b+c)$  llamándolo  $X$ . La estructura es  $a^i X^k$ .

1. Parte equilibrada: Por cada 'a', pongo una  $X$ .
2. Parte sobrante: Añado más  $X$  al final (o al lado de las  $X$ ).

$$S \rightarrow aSX \mid A \quad (\text{Emparejo a con } X)$$

$$A \rightarrow XA \mid X \quad (\text{Genero las } X \text{ sobrantes, al menos una})$$

$$X \rightarrow b \mid c \quad (\text{Defino qué es } X)$$

**Truco para "Distinto"** ( $\neq$ ). "Distinto" significa "Mayor que" O "Menor que".

$$S \rightarrow S_{\text{mayor}} \mid S_{\text{menor}}$$

Haces dos gramáticas (como en el Patrón 4) y las unes.

## Patrón 5: Desorden y Mezcla (Conteo $N(a) = N(b)$ )

Aquí NO hay orden  $a^i b^j$ . Las letras pueden estar mezcladas "aababb...". Esto se resuelve con **Inserción Relativa**.

**Regla Maestra para  $N(a) = N(b)$ :** Si quiero mantener el equilibrio, donde ponga una 'a', debo poner una 'b'. Pero como no hay orden, la 'b' puede ir antes, después, o alrededor.

La forma estándar segura es:

$$S \rightarrow aSbS \mid bSaS \mid SS \mid \epsilon$$

Significado: Si pongo 'a', debo "deber" una 'b' (el estado S intermedio se encarga de resolver esa deuda).

**Ejemplo** ( $N(0) = N(1) + 1$ ): Esto es: "Equilibrado + un 1 extra".

- Definimos un equilibrio perfecto  $B$  (mismo nº de 0 y 1).

$$B \rightarrow 0B1B \mid 1B0B \mid \epsilon$$

(Ojo: simplificado para el concepto, a veces se requiere más rigor con  $SS$ ).

- Estado Inicial: Es el equilibrio  $B$ , pero forzando un '0' extra en algún sitio.

$$S \rightarrow B0B$$

## 4.7 Simplificación de Gramáticas (GIC)

Antes de pasar a formas normales, **siempre** debes limpiar la gramática en este orden estricto:

### 1. Eliminar Producciones $\epsilon$ (Vacías)

Si  $A \rightarrow \epsilon$ , entonces  $A$  es "anulable".

Método:

- Busca todas las variables que pueden volverse  $\epsilon$  (directa o indirectamente).
- Si tienes  $S \rightarrow Ab$ , y  $A$  es anulable, añade una nueva regla  $S \rightarrow b$  (versión donde  $A$  desaparece).
- Borra las reglas  $A \rightarrow \epsilon$  originales (salvo si  $S \rightarrow \epsilon$  es necesario para el lenguaje)

### 2. Eliminar Producciones Unitarias ( $A \rightarrow B$ )

Reglas que solo cambian el nombre de la variable sin añadir terminales.

Método:

- Si  $A \rightarrow B$  y  $B \rightarrow \text{algo}$ , entonces añade  $A \rightarrow \text{algo}$ .
- Borra  $A \rightarrow B$ .

### 3. Eliminar Símbolos Inútiles

Se hace en dos pasadas:

- No Generadores:** Variables que entran en bucle y nunca llegan a terminales ( $A \rightarrow aA$ ). Bórralas.
- Inalcanzables:** Variables a las que no puedes llegar empezando desde  $S$ . Bórralas.

**Ejemplo que usa los pasos 1 y 2:**

$$a) S \rightarrow XC | AY | C | X | Y | A$$

$$X \rightarrow aXb | ab$$

$$C \rightarrow Cc | c$$

$$A \rightarrow Aa | a$$

$$Y \rightarrow bYc | bc | C$$

$$b) S \rightarrow XC | AY | Cc | c | aXb | ab | bYc | bc | Aa | a$$

$$X \rightarrow aXb | ab$$

$$C \rightarrow Cc | c$$

$$A \rightarrow Aa | a$$

$$Y \rightarrow bYc | bc | Cc | c$$

### 4.8 Formas Normales

**Objetivo:** Estandarizar la gramática para que todas las reglas sean "cortas" y binarias. Es fundamental para el algoritmo CYK (análisis sintáctico).

**Solo se permiten 2 tipos de reglas:**

- $A \rightarrow BC$  (Dos variables).
- $A \rightarrow a$  (Un terminal).

### Algoritmo de Conversión a FNC (Práctico)

Supongamos que ya has simplificado la gramática (paso 4.8).

**Paso 1:** Terminales solitarios en reglas mixtas. Si tienes  $S \rightarrow aB$ , eso está prohibido (mezcla terminal y variable).

- Crea una variable nueva:  $X_a \rightarrow a$ .

- Cambia la regla a:  $S \rightarrow X_a B$ .

**Paso 2:** Acortar cadenas largas. Si tienes  $S \rightarrow ABC$  (3 variables), es demasiado largo.

- Rompe la cadena creando variables intermedias.
- $S \rightarrow AZ$
- $Z \rightarrow BC$

Ejemplo Rápido:

Original:  $S \rightarrow aSb$

1. Crear variables para terminales:  $X_a \rightarrow a$ ,  $X_b \rightarrow b$ .
2. Sustituir:  $S \rightarrow X_a S X_b$ .
3. Romper cadena de 3:
  - $S \rightarrow X_a Y$
  - $Y \rightarrow S X_b$

Resultado FNC:

$$\begin{aligned} S &\rightarrow X_a Y \\ Y &\rightarrow S X_b \\ X_a &\rightarrow a \\ X_b &\rightarrow b \end{aligned}$$

## 4.9 Forma Normal de Greibach (FNG)

La **FNG** es otra forma de estandarizar gramáticas (como la de Chomsky), pero con una filosofía distinta: "**Producir letra a letra**".

Imagina una máquina expendedora. En la **FNG**, cada vez que la gramática hace un movimiento (aplica una regla), está **obligada** a soltar exactamente **una moneda (símbolo terminal)** y quedarse con el cambio (variables).

La regla siempre tiene esta forma:

$$A \rightarrow a\alpha$$

- $A$ : Variable actual.
- $a$ : **Un solo terminal** (la moneda que suelta).
- $\alpha$ : Una cadena de cero o más variables (el cambio que te queda por procesar).

**Ejemplo:**  $S \rightarrow aAB$  (Soltó una 'a', le queda procesar A y B).

## Problema 1: Eliminación de Símbolos Inútiles

**Enunciado:** Encontrar gramática equivalente a:

$$S \rightarrow AB \mid CA$$

$$A \rightarrow a$$

$$B \rightarrow BC \mid AB$$

$$C \rightarrow aB \mid b$$

**Objetivo:** Limpiar la gramática. Para ello buscamos variables que sean "basura" (no generan terminales) o "fantasmas" (nadie llega a ellas).

Paso 1: Detectar símbolos NO Generadores. (¿Qué variables son capaces de convertirse en una cadena de solo terminales al final?)

1. ***A es generador:***  $A \rightarrow a$  (directo).
2. ***C es generador:***  $C \rightarrow b$  (directo).
3. ***S es generador:***  $S \rightarrow CA \rightarrow ba$  (indirecto).
4. **¿Y la B?**

- Sus reglas son:  $B \rightarrow BC$  y  $B \rightarrow AB$ .
- Fíjate bien: Para deshacerte de  $B$ , necesitas aplicar una regla. Pero **todas** sus reglas vuelven a invocar a  $B$  ( $BC$  o  $AB$ ).
- Es un bucle infinito:  $B \rightarrow AB \rightarrow aB \rightarrow aAB \dots$  Nunca desaparece.
- **Conclusión:**  $B$  es un símbolo inútil (no generador).

Paso 2: Eliminar la B

Borramos todas las reglas donde aparezca  $B$  (tanto a la izquierda como a la derecha).

- Borrar  $S \rightarrow AB$  (contiene B).
- Borrar todo el bloque de  $B \rightarrow \dots$
- Borrar  $C \rightarrow aB$  (contiene B).

**Gramática resultante:**

$$S \rightarrow CA$$

$$A \rightarrow a$$

$$C \rightarrow b$$

Paso 3: Símbolos Inaccesibles

Desde  $S$ , ¿podemos llegar a todos?

- $S \rightarrow CA$  (Llegamos a C y a A).
- Todo es accesible.

**Resultado Final (Problema 1):**

$$S \rightarrow CA$$

$$A \rightarrow a$$

$$C \rightarrow b$$

## Problema 2: Paso a Forma Normal de Chomsky (FNC)

### Enunciado:

$$S \rightarrow ASB \mid \varepsilon$$

$$A \rightarrow aAS \mid a$$

$$B \rightarrow SbS \mid A \mid bb$$

Este es un ejercicio largo. Se hace en 3 fases estrictas.

### Fase 1: Eliminar producciones $\varepsilon$ (Vacías)

La única regla vacía es  $S \rightarrow \varepsilon$ .  $S$  es "anulable".

Debemos reescribir las reglas imaginando "qué pasa si  $S$  desaparece".

#### 1. En $S \rightarrow ASB$ :

- Si la  $S$  del centro desaparece: queda  $AB$ .
- Nueva regla:  $S \rightarrow ASB \mid AB$ .

#### 2. En $A \rightarrow aAS$ :

- Si la  $S$  del final desaparece: queda  $aA$ .
- Nueva regla:  $A \rightarrow aAS \mid aA$ .

#### 3. En $B \rightarrow SbS$ :

- Si la 1<sup>a</sup>  $S$  desaparece:  $bS$ .
- Si la 2<sup>a</sup>  $S$  desaparece:  $Sb$ .
- Si ambas desaparecen:  $b$
- Nueva regla:  $B \rightarrow SbS \mid bS \mid Sb \mid b$ .

Gramática tras Fase 1 (sin  $\varepsilon$ ):

(Nota: asumimos  $S \rightarrow \varepsilon$  se elimina o se mantiene aparte si el lenguaje acepta vacío, aquí lo quitamos para limpiar).

$$S \rightarrow ASB \mid AB$$

$$A \rightarrow aAS \mid aA \mid a$$

$$B \rightarrow SbS \mid bS \mid Sb \mid b \mid A \mid bb$$

### Fase 2: Eliminar producciones Unitarias ( $X \rightarrow Y$ )

Buscamos reglas donde una variable lleva a una sola variable.

Aquí detectamos:  $B \rightarrow A$ .

- Como  $B$  se convierte en  $A$ ,  $B$  debe heredar todo lo que hace  $A$ .
- Producciones de  $A$ :  $\{aAS, aA, a\}$ .

- Añadimos eso a  $B$  y borramos  $B \rightarrow A$ .

\*Gramática tras Fase 2:

$$\begin{aligned} S &\rightarrow ASB \mid AB \\ A &\rightarrow aAS \mid aA \mid a \\ B &\rightarrow SbS \mid bS \mid Sb \mid b \mid bb \mid \mathbf{aAS} \mid \mathbf{aA} \mid \mathbf{a} \end{aligned}$$

## Fase 3: Convertir a FNC

Reglas permitidas:  $Var \rightarrow VarVar$  o  $Var \rightarrow terminal$ .

1. Crear variables para terminales:

Creamos  $X_a \rightarrow a$  y  $X_b \rightarrow b$ .

Sustituimos todos los terminales en reglas largas.

### 2. Ajustar reglas largas (Romper cadenas):

- $S \rightarrow ASB \Rightarrow S \rightarrow AZ_1$ , donde  $Z_1 \rightarrow SB$ .
- $A \rightarrow aAS \Rightarrow Sustituir'a' : A \rightarrow X_aAS \Rightarrow A \rightarrow X_aZ_2$ , donde  $Z_2 \rightarrow AS$ .
- $B \rightarrow SbS \Rightarrow B \rightarrow SZ_3$ , donde  $Z_3 \rightarrow X_bS$ .
- (Y así con el resto...).

### Resultado Final (Esquemático):

Variables auxiliares:  $X_a \rightarrow a$ ,  $X_b \rightarrow b$ .

Transformación de **S**:

$$S \rightarrow AZ_1 \mid AB \quad (Z_1 \rightarrow SB)$$

Transformación de **A**:

$$A \rightarrow X_aZ_2 \mid X_aA \mid a \quad (Z_2 \rightarrow AS)$$

Transformación de **B**:

$$B \rightarrow SZ_3 \mid X_bS \mid SX_b \mid b \mid X_bX_b \mid X_aZ_2 \mid X_aA \mid a \quad (Z_3 \rightarrow X_bS)$$