4. Gramáticas Independientes del Contexto

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: $\text{Cabeza} \rightarrow \text{Cuerpo}$.

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.

Ejemplo	Tipo	Nombre	¿Qué ves a la IZQUIERDA de la flecha?	La Regla de Oro (En español)
`aAb -> c` *(3 cosas se convierten en 1. Se puede encoger).*	**Tipo 0**	**Sin Restricciones**	**Cualquier cosa.**	Vale todo. Puedes borrar, cambiar, mezclar... Caos total.

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

Nota: Tipo 3 $\subset$ Tipo 2 $\subset$ Tipo 1 $\subset$ 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:** $A \to \text{terminal} \cdot \text{Variable}$ (ej: $A \to aB$) o $A \to \text{terminal}$ ($A \to a$).
- **Lineal Izquierda:** $A \to \text{Variable} \cdot \text{terminal}$ (ej: $A \to Ba$).

> **¡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

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

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 → aSb | ε
T = {a, b}
```

Derivaciones:

- S ⇒ ε → palabra: "" (cadena vacía)
- S ⇒ aSb ⇒ ab → palabra: "ab"
- S ⇒ aSb ⇒ aaSbb ⇒ aabb → palabra: "aabb"
- S ⇒ aSb ⇒ aaSbb ⇒ aaaSbbb ⇒ aaabbb → palabra: "aaabbb"

L(G) = {$a^n b^n$ | n ≥ 0} = {ε, ab, aabb, aaabbb, ...}

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:

$$S \to x\ S\ y$$

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

Ejemplo Práctico Palídromo:

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

> **Gramática final:** $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$

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).

     $$S \to aSc \mid B$$

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

     $$B \to bBcc \mid \varepsilon$$

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.

$$S \to S_1 \mid S_2$$

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.
  - $S_1 \to XC$
  - $X \to aXb \mid \varepsilon$ (pareja a-b)
  - $C \to cC \mid \varepsilon$ (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'.
  - $S_2 \to AY$
  - $A \to aA \mid \varepsilon$ (a libre)
  - $Y \to bYc \mid \varepsilon$ (pareja b-c)

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

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

- $k > i \to$ 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).

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

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

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

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

$$S \to S_{mayor} \mid S_{menor}$$

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:

$$S \to aSbS \mid bSaS \mid SS \mid \varepsilon$$

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).

  $$B \to 0B1B \mid 1B0B \mid \varepsilon$$

  (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.

  $$S \to B\,0\,B$$

4.7 Simplificación de Gramáticas (GIC)

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

1. Eliminar Producciones ε (Vacías)

Si $A \to \varepsilon$, entonces A es "anulable".
Método:

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

2. Eliminar Producciones Unitarias ($A \to B$)

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

1. Si $A \to B$ y $B \to \text{algo}$, entonces añade $A \to \text{algo}$.
2. Borra $A \to B$.

3. Eliminar Símbolos Inútiles

Se hace en dos pasadas:

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

Ejemplo que usa los pasos 1 y 2:

a) $S \to XC \mid AY \mid C \mid X \mid Y \mid A$
$X \to aXb \mid ab$
$C \to Cc \mid c$
$A \to Aa \mid a$
$Y \to bYc \mid bc \mid C$

b) $S \to XC \mid AY \mid Cc \mid c \mid aXb \mid ab \mid bYc \mid bc \mid Aa \mid a$
$X \to aXb \mid ab$
$C \to Cc \mid c$
$A \to Aa \mid a$
$Y \to bYc \mid bc \mid Cc \mid c$

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:

1. $A \to BC$ (Dos variables).
2. $A \to a$ (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 $S \to aB$, eso está prohibido (mezcla terminal y variable).

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

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

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

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

Ejemplo Rápido:
Original: $S \to aSb$

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

Resultado FNC:

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

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:

$$A \to a\alpha$$

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

Ejemplo: $S \to \mathbf{a}AB$ (Soltó una 'a', le queda procesar A y B).

Problema 1: Eliminación de Símbolos Inútiles

Enunciado: Encontrar gramática equivalente a:

$$S \to AB \mid CA$$

$$A \to a$$

$$B \to BC \mid AB$$

$$C \to aB \mid b$$

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:** $A \to a$ (directo).
2. C **es generador:** $C \to b$ (directo).
3. S **es generador:** $S \to CA \to ba$ (indirecto).
4. **¿Y la B?**
   - Sus reglas son: $B \to BC$ y $B \to AB$.
   - 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: $B \to AB \to aB \to aAB \ldots$ 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 $S \to AB$ (contiene B).
- Borrar todo el bloque de $B \to \ldots$
- Borrar $C \to aB$ (contiene B).

Gramática resultante:

$$S \to CA$$

$$A \to a$$

$$C \to b$$

Paso 3: Símbolos Inaccesibles
Desde S, ¿podemos llegar a todos?

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

Resultado Final (Problema 1):

$$S \to CA$$

$$A \to a$$

$$C \to b$$

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

Enunciado:

$$S \to ASB \mid \varepsilon$$

$$A \to aAS \mid a$$

$$B \to SbS \mid A \mid bb$$

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

Fase 1: Eliminar producciones ε (Vacías)

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

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

1. **En $S \to ASB$:**
   - Si la S del centro desaparece: queda AB.
   - Nueva regla: $S \to ASB \mid AB$.
2. **En $A \to aAS$:**
   - Si la S del final desaparece: queda aA.
   - Nueva regla: $A \to aAS \mid aA$.
3. **En $B \to SbS$:**
   - Si la 1ª S desaparece: bS.
   - Si la 2ª S desaparece: Sb.
   - Si ambas desaparecen: b
   - Nueva regla: $B \to SbS \mid bS \mid Sb \mid b$.

Gramática tras Fase 1 (sin ε):
(Nota: asumimos $S \to \varepsilon$ se elimina o se mantiene aparte si el lenguaje acepta vacío, aquí lo quitamos para limpiar).

$$S \to ASB \mid AB$$

$$A \to aAS \mid aA \mid a$$

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

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

Buscamos reglas donde una variable lleva a una sola variable.
Aquí detectamos: $B \to A$.

- 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 $B \to A$.

**Gramática tras Fase 2:*

$$S \to ASB \mid AB$$

$$A \to aAS \mid aA \mid a$$

$$B \to SbS \mid bS \mid Sb \mid b \mid bb \mid \mathbf{aAS} \mid \mathbf{aA} \mid \mathbf{a}$$

Fase 3: Convertir a FNC

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

1. Crear variables para terminales:

Creamos $X_a \to a$ y $X_b \to b$.
Sustituimos todos los terminales en reglas largas.

2. Ajustar reglas largas (Romper cadenas):

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

Resultado Final (Esquemático):
Variables auxiliares: $X_a \to a, X_b \to b$.
Transformación de **S**:

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

Transformación de **A**:

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

Transformación de **B**:

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