

7. Gramáticas GSC Y GSR

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

7.1 Gramáticas Sin Restricciones (GSR) - Tipo 0

Como su nombre indica, son las gramáticas más flexibles y potentes. No tienen límites en la forma de sus reglas, salvo que siempre debe haber algo que sustituir.

Definición y Reglas

Una GSR se define formalmente como $G = (NT, T, S, P)$.

Sus producciones tienen la forma genérica:

$$x \rightarrow y$$

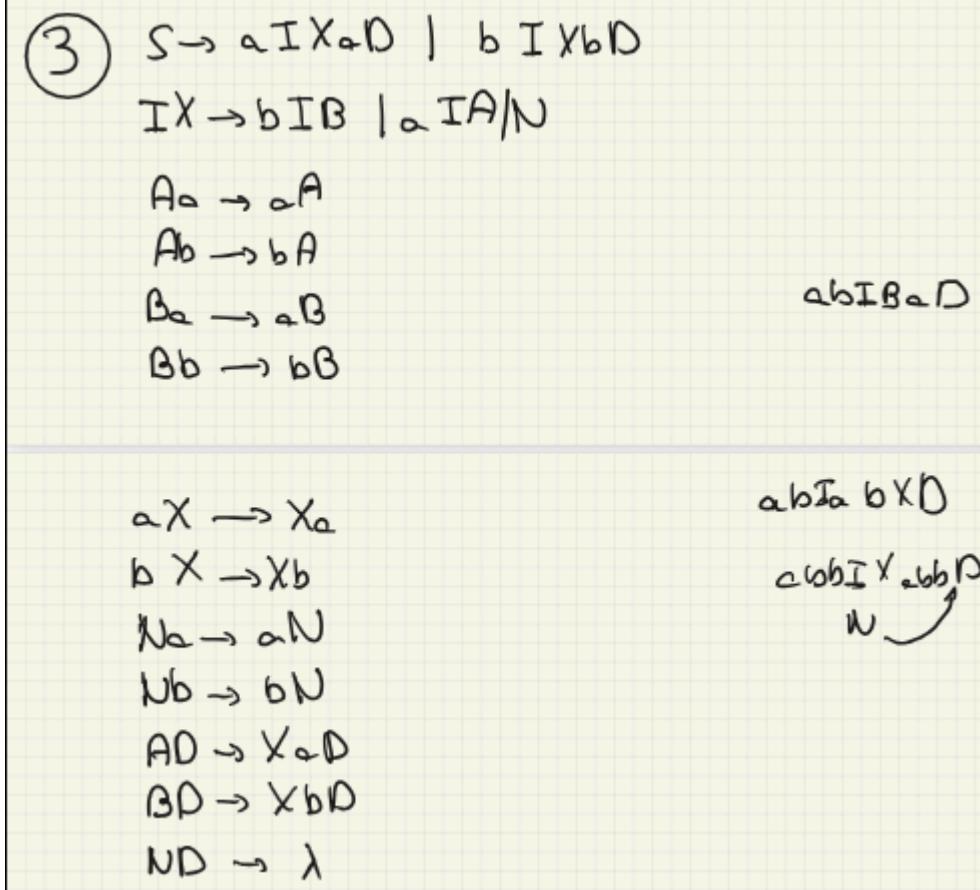
Donde:

- **x (Izquierda):** Es una cadena de símbolos terminales y no terminales (NT/T). La única condición es que **no puede estar vacía** (debe tener al menos un símbolo).
- **y (Derecha):** Es cualquier cadena de símbolos terminales y no terminales. **Puede ser vacía.**

Características Clave

- **Libertad total:** Puedes reemplazar un grupo de símbolos por otro grupo, acortar la cadena, alargarla o borrar símbolos completamente.
- **Lenguaje que generan:** Generan los **Lenguajes Recursivamente Enumerables (LRE)**.
- **Máquina Equivalente:** Son reconocidos por la **Máquina de Turing** estándar.

Ejemplo:



7.2 Gramáticas Sensibles al Contexto (GSC) - Tipo 1

Estas gramáticas imponen restricciones sobre las GSR. La idea principal es que las sustituciones dependen de lo que rodea a la variable ("el contexto") y la cadena nunca se hace más pequeña.

Definición y Reglas

Una GSC tiene producciones de la forma:

$$\alpha A \beta \rightarrow \alpha \gamma \beta$$

Desglosemos qué significa esto:

1. A : Es un **símbolo no terminal** (la variable que vamos a cambiar).
2. α y β (**El Contexto**): Son cadenas de terminales o no terminales (pueden ser vacías). Representan lo que está a la izquierda y derecha de A .
3. γ (**El Cambio**): Es una cadena **no vacía**. Es por lo que sustituimos a A .

La Regla de Oro: No Contracción

La característica más importante para identificar una GSC es la longitud.

- La longitud de la parte derecha (y) debe ser **igual o mayor** que la de la parte izquierda (x).
- Esto significa que la gramática **nunca "encoge"** la cadena (excepto posiblemente para generar la cadena vacía λ si el lenguaje lo permite)10.

Ejemplo:

1	$L = \{ a^{n+1} b^n c^{n-1} : n \geq 1 \}$
	$S \rightarrow aab \mid aaXbc$
	$aX \rightarrow aabY \mid aab$
	$Yb \rightarrow bY$
	$Yc \rightarrow Xcc$
	$bX \rightarrow Xb$
	$\in Xbc$
	$\in aabYbc$
	$\in aaabbYc$
	$\in aaabbXcc$

Relación con Máquinas

- **Lenguaje que generan:** Lenguajes Sensibles al Contexto (LSC).
- **Máquina Equivalente:** Son reconocidos por los **Autómatas Linealmente Acotados (ALA)**. Un ALA es básicamente una Máquina de Turing que solo tiene permiso para usar la parte de la cinta donde está escrita la entrada (espacio finito).

7.3 Protips

1. El Mensajero (Commutation)

Es el truco más importante. Si necesitas generar $a^n b^n c^n$, no puedes hacerlo de golpe.

Generas aBC y luego mueves la B o la C a su sitio.

- **Regla de tráfico:** $CB \rightarrow BC$ (Permite que la C "adelante" a la B).

2. El Muro (Boundary)

A veces necesitas saber dónde acaba la cadena para convertir variables en letras finales. Usamos marcadores de inicio/final.

- **Ejemplo:** $\#ABC\#$. Cuando una variable toca la pared ($\#$), se transforma.

3. El Clonador

Para ww o a^{2n} . Una variable se divide en dos, o transporta una copia de sí misma.

7.4 Identificar Gramáticas y Lenguajes

Gramáticas

¿Qué ves a la IZQUIERDA?	¿Qué ves a la DERECHA?	TIPO
Solo 1 Variable ($A \rightarrow \dots$)	Estricto: a ó aB	Tipo 3 (Regular)

¿Qué ves a la IZQUIERDA?	¿Qué ves a la DERECHA?	TIPO
Solo 1 Variable ($A \rightarrow \dots$)	Libre: aAb, BC , etc.	Tipo 2 (GIC)
Grupo ($aA \rightarrow \dots$)	Longitud \geq Izquierda	Tipo 1 (Sensible)
Grupo ($aA \rightarrow \dots$)	Longitud $<$ Izquierda	Tipo 0 (Irrestricta)

Lenguajes

1. Lenguajes Regulares (Tipo 3)

Memoria: Nula o Finitud.

Clave: "No necesito contar hasta el infinito".

- **Patrones:** Empieza por, termina en, contiene la subcadena "aba".
- **Conteo:** Solo cuenta hasta un número fijo. (Ej: "Longitud par", "múltiplo de 3").
- **NO PUEDE:** Contar cosas indefinidas y compararlas.
- *Ejemplo:* $L = \{ \text{cadenas de } a \text{ y } b \text{ que terminan en } ab \}$.

2. Lenguajes Independientes del Contexto (Tipo 2)

Memoria: Una Pila (Stack).

Clave: "Puedo comparar DOS cosas (o anidar)".

- **Patrones:** Equilibrio, espejos.
- **Conteo:** $a^n b^n$ (mismo número de a's que de b's).
- **Estructura:** Paréntesis equilibrados (()), palíndromos (ww^R).
- **Límite:** Solo puedo relacionar **dos** contadores a la vez o estructuras anidadas. No puedo cruzar relaciones.
- *Ejemplo:* $L = \{ a^n b^n \mid n \geq 0 \}$.

$$L = \{ a^i b^j c^k \mid i=j \text{ o } j=k, i, j, k > 0 \}.$$

$$L = \{ a^i b^i a^j b^j \mid i, j > 0 \},$$

$$L = \{ a^i b^j c^k \mid k = i + (2 * j) \}$$

3. Lenguajes Sensibles al Contexto (Tipo 1)

Memoria: Cinta acotada (puedo leer y volver atrás).

Clave: "Puedo comparar TRES cosas o COPIAR".

- **Patrones:** Relaciones triples o cruzadas.

- **Conteo:** $a^n b^n c^n$ (Tres cantidades iguales).
- **Copia:** ww (una palabra repetida exactamente igual, ej: "papa", "mama"). *Nota: El palíndromo es Tipo 2, la copia exacta es Tipo 1.*
- **Ejemplo:** $L = \{a^n b^n c^n \mid n \geq 1\}$.

$$L = \{a^{n+1} b^n c^{n-1} : n \geq 1\}.$$

4. Lenguajes Recursivamente Enumerables (Tipo 0)

Memoria: Ordenador completo.

Clave: "Cualquier algoritmo lógico".

- Si te dan un problema lógico complejo que no tiene restricciones de estructura simple. Generalmente, en los exámenes, se centran en los tres anteriores, salvo que pregunten por problemas de parada o indecidibilidad.

$$L = \{ww : w \in \{a, b\}^+\}.$$