

## 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:

$$x \rightarrow y$$

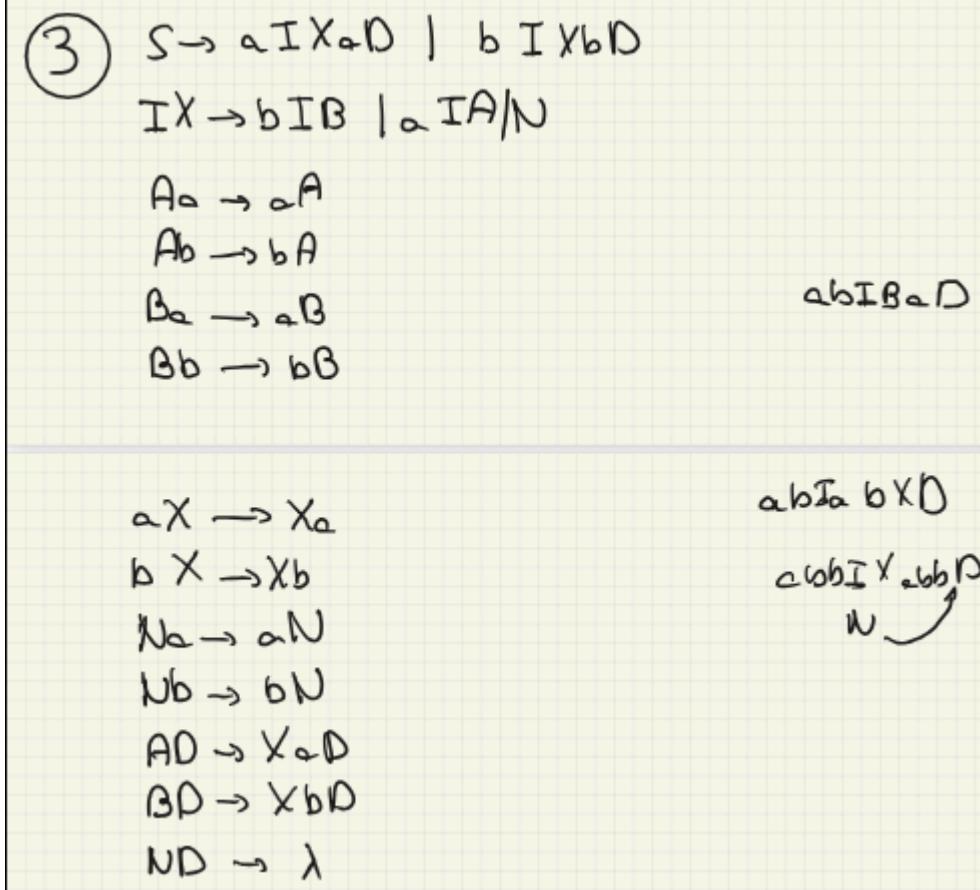
Donde:

- **$x$  (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.  $\alpha$  y  $\beta$  (**El Contexto**): Son cadenas de terminales o no terminales (pueden ser vacías). Representan lo que está a la izquierda y derecha de  $A$ .
3.  $\gamma$  (**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  $\lambda$  si el lenguaje lo permite).

**Ejemplo:**

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

$\Sigma \rightarrow aab \mid aaXbc$	$aX \rightarrow aabY \mid aab$	$Yb \rightarrow bY$	$Yc \rightarrow Xcc$	$bX \rightarrow Xb$
$aabb$	$aabbYbc$	$aaabbYbc$	$aaabbYcc$	$aaabbXcc$

**Relación con Máquinas**

- **Lenguaje que generan:** Lenguajes Sensibles al Contexto (LSC).
- **Máquina Equivalente:** Son reconocidos por los **Autómatas Linealmente Acotados (ALA)**. Es una Máquina de Turing (MT) no determinista con su cinta limitada por ambos extremos, siendo el tamaño de la cinta fijo. Imagina una MT normal, que tiene papel infinito para hacer cálculos. Un ABA, en cambio, solo tiene el espacio que ocupa la palabra de entrada (y quizás unas pocas casillas más fijas a los lados). No puede pedir más papel. Para que la cabeza lectora no se "caiga" de la cinta, tiene dos símbolos especiales (marcadores) en los extremos izquierdo y derecho que le dicen "hasta aquí puedes llegar".

**La fórmula del Lenguaje  $L(M)$ .** **L(M)** significa "**El Lenguaje de la Máquina M**". En términos sencillos: es la lista VIP de palabras que la máquina acepta.

$$L(M) = \{ w \in \Sigma^+ : q_0[w] \vdash^* [x_1 q_f x_2], q_f \in F \dots \}$$

- **[ (Marcador izquierdo):** Es el muro de la izquierda.
  - La fórmula  $\delta(q_i, [) = (q_j, [, D)$  significa: "Si la cabeza está en el estado  $q_i$  y lee el muro **[**, pasa al estado  $q_j$ , **deja el muro intacto** (**[**) y se mueve a la **Derecha** (**D**)".
  - En resumen: **Rebota hacia dentro**. No puede atravesarlo ni borrarlo.
- **] (Marcador derecho):** Es el muro de la derecha.
  - La fórmula  $\delta(q_i, ]) = (q_j, ], I)$  significa: "Si lee el muro **]**, **lo deja intacto** y se mueve a la **Izquierda** (**I**)".
  - En resumen: **Rebota hacia dentro**.

Se traduce así:

1. Tomas una palabra de entrada  $w$ .
2. La encierras entre los muros: **[w]**.
3. Empiezas en el estado inicial  $q_0$ .
4. La máquina procesa ( $\vdash^*$ ) moviéndose dentro de esos muros.

5. Si la máquina llega a un estado final ( $q_f \in F$ ) manteniendo los muros intactos ( $[x_1 \dots x_2]$ ), entonces **acepta** la palabra.

## 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 Grámaticas y Lenguajes

### Gramáticas

¿Qué ves a la IZQUIERDA?	¿Qué ves a la DERECHA?	TIPO
<b>Solo 1 Variable</b> ( $A \rightarrow \dots$ )	Estricto: $a$ ó $aB$	<b>Tipo 3 (Regular)</b>
<b>Solo 1 Variable</b> ( $A \rightarrow \dots$ )	Libre: $aAb$ , $BC$ , etc.	<b>Tipo 2 (GIC)</b>
<b>Grupo</b> ( $aA \rightarrow \dots$ )	Longitud $\geq$ Izquierda	<b>Tipo 1 (Sensible)</b>
<b>Grupo</b> ( $aA \rightarrow \dots$ )	Longitud $<$ Izquierda	<b>Tipo 0 (Irrestricta)</b>

### 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:**  $L = \{a^n b^n \mid n \geq 0\}$ .

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

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

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

## 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.

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