

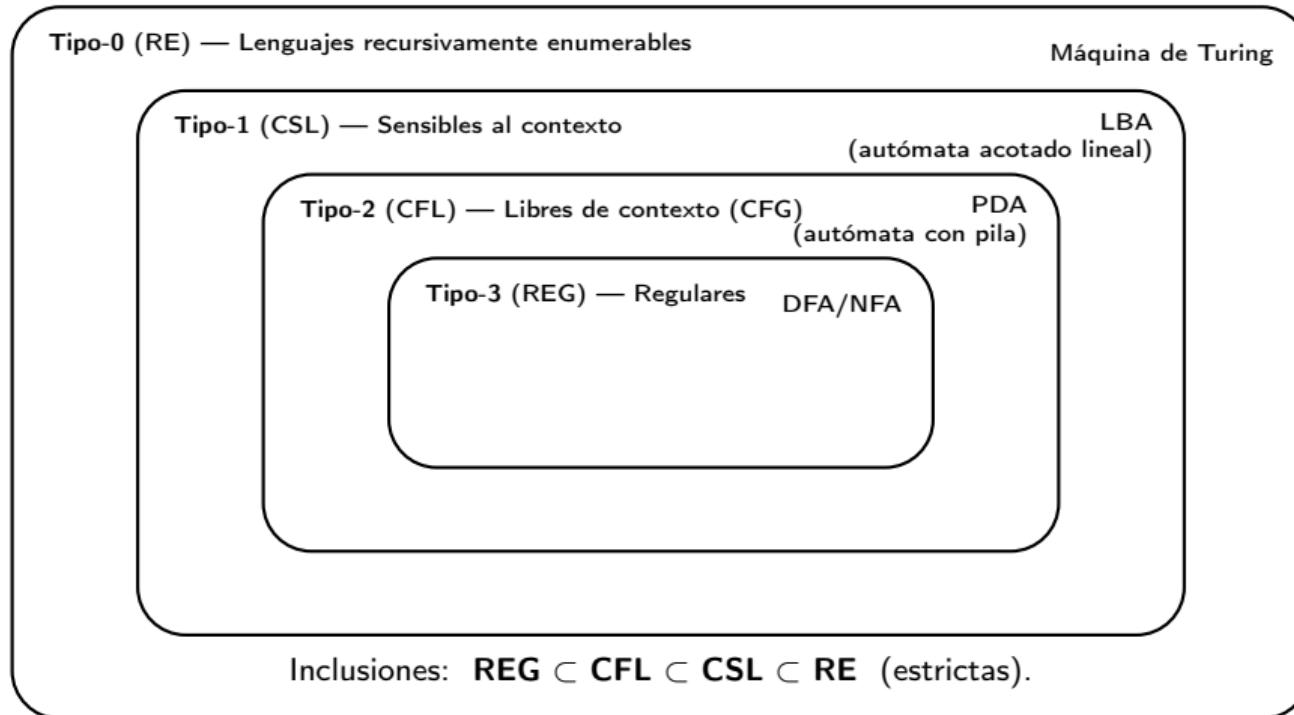
Teoría de Autómatas y Lenguajes Formales

Unidad 2 — Sesión 10 (26-ene-2026): CFG (definición + diseño) + pertenencia + práctica
+ MT10

Docente: Helder Octavio Fernández Guzmán

Jerarquía de Chomsky: mapa de lenguajes (contenidos)

Periodo 1/6



- Definición suficiente: $G = (V, \Sigma, P, S)$, derivación \Rightarrow , lenguaje $L(G)$.
- **Diseño guiado 1:** paréntesis balanceados con () .
- **Diseño guiado 2:** listas id, id, ..., id (sin coma final).
- **Práctica corta:** IN / NOT IN con evidencia mínima.
- **Quiz de cierre (Moodle):** 8–10 min (antes de MT10).
- **Bloque final (90 min):** MT10 (Avance en clase + trabajo final).

¿Qué es una CFG? (definición suficiente)

Periodo 2/6

Una **gramática libre de contexto** es un cuádruple:

$$G = (V, \Sigma, P, S)$$

- V : conjunto de **no terminales** (variables).
- Σ : conjunto de **terminales** (alfabeto), $V \cap \Sigma = \emptyset$.
- P : conjunto finito de **producciones** $A \rightarrow \alpha$, con $A \in V$ y $\alpha \in (V \cup \Sigma)^*$.
- $S \in V$: símbolo inicial.

- Una forma sentencial puede contener terminales y no terminales.
- **Derivación (un paso):** si $A \rightarrow \alpha \in P$, entonces

$$xAy \Rightarrow x\alpha y$$

- **Derivación (cero o más pasos):** \Rightarrow^* .
- **Lenguaje de la gramática:**

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

Idea guía: “ w está en $L(G)$ ” si puedo llegar a w desde S usando reglas, y terminando con solo terminales.

Patrón 1: paréntesis balanceados (idea)

Periodo 3/6

Alfabeto: $\Sigma = \{(,)\}$.

Intuición del lenguaje:

- Se permite **anidar**: $(())$
- Se permite **concatenar**: $() ()$
- Se permite **vacío**: ϵ

Buscamos una gramática que capture:

- “envolver” un bloque balanceado
- “pegar” dos bloques balanceados

CFG para paréntesis balanceados

Periodo 3/6

Gramática G_1 :

$$S \rightarrow SS \mid (S) \mid \epsilon$$

- (S) : crea **anidamiento**.
- SS : crea **concatenación**.
- ϵ : permite **terminar**.

Chequeo rápido: $)$ (**no** puede pertenecer (cierra antes de abrir)).

Derivación guiada (ejemplo): ()()

Periodo 3/6

Una derivación posible:

$$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ()S \Rightarrow ()(S) \Rightarrow ()()$$

- Usamos SS para “dos bloques”.
- En cada bloque: (S) y luego $S \Rightarrow \epsilon$.

Patrón 2: listas con separador (idea)

Periodo 4/6

Queremos modelar listas tipo:

id, id,id, id,id,id, ...

Restricción: no se permite coma final (id, es inválido).

Alfabeto: $\Sigma = \{\text{id}, ,\}$.

Idea:

- Caso base: una lista mínima (un solo id).
- Caso recursivo: id, seguido de otra lista.

Gramática G_2 :

$$L \rightarrow id \mid id, L$$

Lectura:

- $L \Rightarrow id$ produce una lista de 1 elemento.
- $L \Rightarrow id, L$ produce un id y “exige” otra lista a la derecha.

Chequeo rápido: id , no puede pertenecer (la coma obliga a continuar).

Derivación guiada (ejemplo): id, id, id

Periodo 4/6

Una derivación posible:

$$L \Rightarrow id, L \Rightarrow id, id, L \Rightarrow id, id, id$$

(donde en el último paso usamos $L \Rightarrow id$)

Importante: la gramática evita , al final porque cada coma viene pegada a un id y debe continuar con L .

Práctica (IN / NOT IN) con evidencia mínima

Periodo 4/6

Regla de evidencia:

- Para un caso **IN**: derivación (corta, pero completa).
- Para un caso **NOT IN**: argumento estructural (2–3 líneas).

G_1 (paréntesis)	G_2 (listas)
IN: ϵ , (), (()), ()()	IN: id, id,id, id,id,id
NOT IN:)(), (((), ()()	NOT IN: ,id, id,, id,,id

Meta: que al menos 1 caso IN por gramática quede derivado a mano (para entrenar EE2 presencial).

- Duración sugerida: **8–10 minutos.**
- Propósito: verificar notación mínima (terminal / no terminal / ϵ / $L(G)$) **antes** de pasar a MT10.
- Luego: revisión express de 2–3 preguntas frecuentes.

Nota logística: el quiz es parte de la sesión y sirve como “cierre” del bloque de clase (no reemplaza MT10).

MT10 se trabaja en clase y se completa hasta el final del día.

Entrega 1: “Avance en clase” (obligatoria antes de salir)

- CFG propuesta (borrador aceptable) + opción elegida (A, B o C).
- **Justificación breve** (2–3 líneas): por qué elegiste esa opción.
- 1 caso IN + 1 caso NOT IN (con evidencia mínima).
- Texto del archivo .cfg con cabecera (# Nombre — MT10 — Fecha — Avance).
- Evidencia mínima: captura del simulador o foto legible de apuntes.

Entrega 2: “Final” (hasta 23:55)

- 2 IN + 2 NOT IN (con justificación).
- 1 derivación manual corta (para un IN).
- Capturas del simulador (al menos 1 IN y 1 NOT IN) + .cfg final.

Opción A — Llaves balanceadas con contenido mínimo

Balanceadas y cada par encierra al menos una x.

Ejemplos IN: {x}, {{x}}, {x}{x} NOT IN: ϵ , {}, }{

Opción B — Sumas de números (sin + final)

$n + n + \dots + n$ (con n de 1 o más dígitos)

Ejemplos IN: 3, 12+7, 1+2+30 NOT IN: +3, 3+, 3++2

Opción C — Estructura if simple (sin anidamiento)

if(E)then(E)else(E), $E \in \{\text{id}, 0..9\}$

Ejemplos IN: if(id)then(id)else(id), if(3)then(id)else(7) NOT IN:
if(id)then()else(id), if(id)then(id)else

En el simulador, una CFG se escribe como un **mapping YAML**:

Ejemplo (paréntesis balanceados)

```
# CFG generating language of balanced () parentheses
S: [(S), SS, '']
```

Notas:

- Debe haber **espacio** después de :
- La cadena vacía se escribe como ''
- Usa # para comentarios (cabecera con tu nombre)

Checklist de calidad (para evitar pérdida de puntos)

Periodo 6/6

Antes de entregar, verifica:

- Elegiste **A, B o C** y lo indicaste en el PDF (con 2–3 líneas de justificación).
- Tu CFG genera los casos IN y **no** genera los NOT IN.
- Tu derivación manual es **corta** y termina en solo terminales.
- Incluiste el .cfg con cabecera de nombre (y fecha).
- Capturas legibles (simulador / apuntes).

La parte manual no es decorativa; es entrenamiento para el examen escrito.

- CFG: $G = (V, \Sigma, P, S)$ y $L(G)$.
- Dos patrones clave: **balanceo** y **lista**.
- Pertenencia siempre con evidencia: derivación o estructura.

Siguiente paso: derivaciones más sistemáticas (izq./der.) y puente a árboles.