

Teoría de Autómatas y Lenguajes Formales

Unidad 1 — Sesión 6 (15-ene-2026): Expresiones regulares (RE) + verificación + micro-lab

Docente: Helder Octavio Fernández Guzmán

Sesión 6: ¿qué haremos hoy?

Periodo 1/6

- Tema central: **Expresiones regulares (RE)** como **descripciones de lenguajes**.
- Meta práctica: **leer, construir y verificar RE** con casos de prueba.
- Herramientas:
 - **Automata Tutor: Words in RE** (verificación) y **RE to ε -NFA** (demo).
- Bloque final: **Micro-lab** con evidencia mínima.

Horario clave: curso 6:45–11:15; **micro-lab** 9:45–11:15.

- Alfabeto: $\Sigma = \{0, 1\}$.
- Σ^* : todas las cadenas sobre Σ (incluye ε).
- Lenguaje: **conjunto de cadenas**.
- Operaciones (intuición): **unión y concatenación**.

Objetivo del warm-up: asegurar base común antes de RE.

Warm-up: 4 preguntas rápidas

Periodo 1/6

- ① Si $\Sigma = \{0, 1\}$, ¿ $1010 \in \Sigma^*$? ¿y 2?
- ② ¿ ε es una cadena? ¿cuál es su longitud?
- ③ Si $L_1 = \{0, 00\}$ y $L_2 = \{1\}$, ¿cuál es $L_1 \cdot L_2$?
- ④ Si L = “cadenas que terminan en 1”, da 2 ejemplos que pertenecen y 2 que no.

Warm-up: respuestas (para alinear conceptos)

Periodo 1/6

- (1) $1010 \in \Sigma^*$ (usa solo símbolos de Σ). $2 \notin \Sigma^*$.
- (2) ε **sí** es una cadena; $|\varepsilon| = 0$.
- (3) $L_1 \cdot L_2 = \{01, 001\}$.
- (4) Pertenecen: 1, 000001. No pertenecen: ε , 10.

Mini-nota: un lenguaje es un **conjunto** (no un diagrama).

Al final de la sesión, podrás:

- Interpretar una RE como **lenguaje** (decidir pertenencia de cadenas).
- Construir RE desde enunciados frecuentes (patrones típicos).
- Verificar RE con **casos de prueba** (no solo “a ojo”).
- Conectar RE con un ε -NFA equivalente (demo con herramienta).

- Unión: |
- Concatenación: implícita (ej.: 01 significa 0 seguido de 1)
- Estrella de Kleene: * (cero o más repeticiones)
- Paréntesis: (...) para agrupar

Convención: $\Sigma = \{0, 1\}$ y $\Sigma^* \equiv (0|1)^*$.

Precedencia (evita la mayoría de errores)

Periodo 2/6

- Orden de precedencia:
 $*$ > concatenación > |
- Regla de oro: si dudas, **usa paréntesis.**

Mini-ejercicio: compara $0|11^*$ vs $(0|11)^*$ vs $0|(11^*)$.

RE como lenguaje: “acepta” / “rechaza”

Periodo 2/6

Ejemplo:

$$R = (0|1)*101(0|1)*$$

- Acepta: 101, 1101011
- Rechaza: ε , 1001

Idea: una RE describe un **conjunto de cadenas**.

Plantillas “de supervivencia” (para construir RE)

Periodo 3/6

- “contiene X ” $\rightarrow \Sigma^* X \Sigma^*$
- “empieza con X ” $\rightarrow X \Sigma^*$
- “termina en X ” $\rightarrow \Sigma^* X$
- “repite bloque B ” $\rightarrow (B)^*$
- “ A o B ” $\rightarrow (A|B)$

Consejo: primero escribe el enunciado como piezas; luego traduce a RE.

Ejemplo guiado 1: “contiene 101”

Periodo 3/6

- Lenguaje: cadenas binarias que contienen la subcadena 101.
- Construcción por plantilla:

$$\Sigma^* 101 \Sigma^* \equiv (0|1)^* 101 (0|1)^*$$

- Pruebas rápidas:
 - Acepta: 101, 0101
 - Rechaza: 10, 11111

Ejemplo guiado 2: “termina en 00 o 11”

Periodo 3/6

- Lenguaje: cadenas que terminan en 00 o en 11.
- Construcción:

$$\Sigma^*(00|11) \equiv (0|1)^*(00|11)$$

- Pruebas rápidas:
 - Acepta: 00, 01011
 - Rechaza: ε , 1010

Ejemplo semi-guiado 3: bloques 01 repetidos

Periodo 3/6

- Lenguaje: cadenas formadas por bloques 01 repetidos (incluye ε).
- RE:
$$(01)^*$$
- Pruebas rápidas:
 - Acepta: ε , 0101
 - Rechaza: 0, 010

Checklist mínimo de pruebas (8 casos):

- 2 bordes: ε y el mínimo que debería aceptar.
- 2 “casi cumple”: falla por 1 símbolo.
- 2 casos medios.
- 2 casos largos / con repetición.

Regla: si aparece un contraejemplo, ajusta la RE (o tu interpretación del enunciado).

- ① Ingresar la RE.
- ② Probar una lista de cadenas (positivas y negativas).
- ③ Si falla, ajustar RE o pruebas (buscar contraejemplos).
- ④ Guardar evidencia mínima (según criterio del curso).

Logro esperado: RE correcta + pruebas coherentes.

Actividad (momento): Automata Tutor *Words in Regex* — AT-S6-01 (2 ejercicios cortos: 3 IN + 3 NOT IN).

Demo breve: RE $\rightarrow \varepsilon$ -NFA (equivalencia de lenguaje)

Periodo 4/6

- Convertiremos una RE a un ε -NFA usando Automata Tutor.
- Probaremos 4 cadenas (2 acepta / 2 rechaza) para confirmar:

$$L(\text{RE}) = L(\varepsilon\text{-NFA})$$

Mensaje clave: cambia la representación, no el lenguaje.

Actividad (momento): Automata Tutor *RE to ε -NFA* — demo guiada (5–8 min) / bonus si terminan temprano.

Quiz S6 (Moodle) — termómetro antes del micro-lab

Periodo 5/6

- Plataforma: **Moodle (Quiz)**
- Preguntas: **8**
- Tiempo: **10–12 min**
- Intentos: **1**
- Objetivo: detectar errores típicos antes de iniciar el micro-lab.

Regla de uso: si fallas en precedencia/paréntesis o en “contiene vs termina”, ajusta tu idea antes de iniciar el micro-lab.

Actividad (momento): Moodle Quiz S6 — 8 preguntas, 10–12 min, 1 intento. (Se realiza aquí, antes del micro-lab).

Micro-lab S6: construir y verificar (evidencia mínima)

Periodo 5/6

- Alfabeto: $\Sigma = \{0, 1\}$.
- Lenguaje: cadenas que
 - **empiezan** con 1, y
 - **contienen** la subcadena 00 (en cualquier posición).
- Evidencia mínima (entrega):
 - ① RE final.
 - ② 8 casos de prueba (4 acepta / 4 rechaza) con justificación breve.
 - ③ Evidencia en Automata Tutor: **Words in RE**.

Micro-lab: 9:45–11:15 (trabajo del estudiante).

Actividad (momento): Automata Tutor *Regex Construction* — AT-S6-02 (RE oficial). Apoyo: UC Davis (pruebas).

Entrega: PDF en Moodle.

Checklist del micro-lab (antes de validar)

Periodo 5/6

Antes de validar, confirma:

- ① ¿Tu RE **obliga** a iniciar con 1?
- ② ¿Tu RE permite 00 **en cualquier posición** (no solo al final)?
- ③ ¿Estás usando correctamente $\Sigma^* = (0|1)^*$?
- ④ ¿Tus 8 pruebas incluyen bordes y “casi cumple”?

Tip: si una prueba falla, produce un contraejemplo y ajusta.

- Hoy:
 - leímos RE como lenguajes (acepta/rechaza),
 - construimos RE con plantillas típicas,
 - verificamos con casos de prueba y herramienta.
- Mañana (Sesión 7):
 - retroalimentación del micro-lab,
 - **propiedades de clausura** de lenguajes regulares,
 - práctica integrada (RE + justificación + verificación).