

Teoría de Autómatas y Lenguajes Formales

Unidad 1 — Sesión 7 (Vie 16-ene-2026): Clausura aplicada (regulares) + práctica integrada

Docente: Helder Octavio Fernández Guzmán

Sesión 7: ¿qué haremos hoy?

Periodo 1/6

- Tema central: **clausura** de los lenguajes regulares (idea operativa).
- Operaciones foco: **complemento, intersección y diferencia**.
- Meta del día:
 - construir un autómata para $L = A \setminus B$ usando

$$A \setminus B = A \cap \overline{B},$$

- validar con pruebas (cadenas IN / NOT IN) en el simulador.

Bloque final (90 min): micro-lab con UC Davis Automaton Simulator + evidencia en PDF.

- Sesión 6: usamos **RE** para describir lenguajes.
- Sesión 7: usamos **operaciones** para construir lenguajes nuevos y sus autómatas.
- Regla mental:
 - no invento un autómata desde cero,
 - lo **compongo** a partir de autómatas ya conocidos.

- Un conjunto de lenguajes es **cerrado** bajo una operación si al aplicarla, el resultado permanece en el conjunto.
- Para regulares:
 - si A y B son regulares, entonces \overline{A} , $A \cap B$, $A \cup B$, $A \setminus B$ siguen siendo regulares.
- Hoy usaremos clausura como **herramienta de construcción**.

- Definición:

$$A \setminus B = \{ w \mid w \in A \text{ y } w \notin B \}.$$

- Forma constructiva:

$$A \setminus B = A \cap \overline{B}.$$

- Lectura tipo checklist: **cumple A y no cumple B**.

Complemento en DFA completo (receta)

Periodo 2/6

- Si B está dado por un DFA completo $M = (Q, \Sigma, \delta, q_0, F)$, entonces:

\overline{B} se obtiene con $F' = Q \setminus F$.

- Todo lo demás se mantiene: Q, Σ, δ, q_0 .
- Condición importante: el DFA debe ser **completo** (sin transiciones faltantes).

- Para DFA de A y B , el producto construye $A \cap B$:
 - estados: pares (p, q) ,
 - inicial: (p_0, q_0) ,
 - transición: $\delta((p, q), a) = (\delta_A(p, a), \delta_B(q, a))$,
 - aceptación: (p, q) es final si p y q son finales.

Consejo: construir solo estados **alcanzables** desde el inicial.

Ejemplo del día (lenguajes)

Periodo 3/6

Sea $\Sigma = \{0, 1\}$.

- A : cadenas que terminan en 00.
- B : cadenas que contienen la subcadena 11.
- Objetivo:

$$L = A \setminus B = A \cap \overline{B}.$$

Lectura: termina en 00 y no contiene 11.

Referencia (YAML):

```
states: [s, s0, s00]
input_alphabet: ['0', '1']
start_state: s
accept_states: [s00]
delta:
  s:
    '0': s0
    '1': s
  s0:
    '0': s00
    '1': s
  s00:
    '0': s00
```

Referencia (YAML):

```
states: [q, q1, q11]
input_alphabet: ['0', '1']
start_state: q
accept_states: [q11]
delta:
    q:
        '0': q
        '1': q1
    q1:
        '0': q
        '1': q11
    q11:
        '0': q11
```

Paso 1: construir \overline{B} (solo cambia aceptación)

Periodo 3/6

- Mantener el mismo DFA (mismos estados y transiciones).
- Invertir aceptación:

$$F_{\overline{B}} = Q_B \setminus F_B.$$

- En este caso:

$$F_B = \{q11\} \Rightarrow F_{\overline{B}} = \{q, q1\}.$$

En YAML: cambia accept_states: [q11] por accept_states: [q, q1].

Paso 2: producto para $A \cap \overline{B}$ (receta operativa)

Periodo 3/6

- **Meta:** construir el DFA de $L = A \cap \overline{B}$ usando producto.
- **Estados del producto:** pares (p, q) con $p \in Q_A$ y $q \in Q_{\overline{B}}$.
- **Inicial:** (p_0, q_0) ; en el ejemplo: (s, q) .
- **Finales:** (p, q) es final si p es final en A y q es final en \overline{B} .

$$F_A = \{s00\}, \quad F_{\overline{B}} = \{q, q1\} \Rightarrow F_L = \{(s00, q), (s00, q1)\}$$

- **Regla unica de transicion (para $x \in \{0, 1\}$):**

$$\delta((p, q), x) = (\delta_A(p, x), \delta_{\overline{B}}(q, x))$$

Arranque guiado (primeras transiciones):

$$\delta((s, q), 0) = (s0, q) \quad y \quad \delta((s, q), 1) = (s, q1)$$

Tip: construir solo pares **alcanzables** desde (s, q) .

Prueba rápida (antes del quiz): 2 cadenas (criterio de aceptación)

Periodo 3/6

- Recordatorio: $L = A \cap \overline{B}$ acepta si:
 - termina en 00 (cumple A), y
 - **no** contiene 11 (cumple \overline{B}).
- **IN (debe aceptar):** 100
 - cumple A (termina en 00) y cumple \overline{B} (no aparece 11).
- **NOT IN (debe rechazar):** 1100
 - cumple A (termina en 00), pero **viola** \overline{B} (si contiene 11).

Lectura rápida: si aparece 11 en cualquier parte, queda fuera de $A \setminus B$.

- Duración sugerida: 10–12 minutos.
- Evalúa:
 - idea de clausura,
 - complemento en DFA,
 - producto para intersección,
 - identidad $A \setminus B = A \cap \overline{B}$,
 - clasificación de cadenas (IN / NOT IN).

Objetivo: detectar errores típicos antes del micro-lab.

- Herramienta: **UC Davis Automaton Simulator**.
- Construir:

- ① cargar DFA(A),
- ② cargar DFA(B),
- ③ construir \overline{B} (invertir aceptación),
- ④ construir producto para $A \cap \overline{B}$,
- ⑤ probar 8 cadenas (4 IN / 4 NOT IN).

Evidencia: 1 PDF en Moodle con capturas + tabla de pruebas.

Tabla compacta (para el PDF de evidencia)

Periodo 5/6

#	Cadena	Justificación breve (A y no B)
1	...	termina en 00; no contiene 11
2	...	termina en 00; contiene 11 (debe rechazar)
3	...	
4	...	
5	...	
6	...	
7	...	
8	...	

Requisito: 4 IN / 4 NOT IN, con al menos 1 caso trampa.

Checklist de calidad (antes de entregar)

Periodo 6/6

- \overline{B} : mismas transiciones, solo cambian estados finales.
- Producto: aceptación correcta (finales del producto).
- L acepta solo si:
 - termina en 00 y
 - no contiene 11.
- La evidencia incluye: capturas + tabla de 8 pruebas.

- Idea instalada: **componer** lenguajes regulares → **componer** autómatas.
- Operaciones dominadas hoy:
 - complemento,
 - intersección (producto),
 - diferencia como $A \cap \overline{B}$.
- Siguiente semana: seguir combinando construcciones y conectarlas con equivalencias/transformaciones del cronograma.