Teoría de Autómatas y Lenguajes Formales

Unidad 1 — Sesión 4 (13-ene-2026): AFN $\rightarrow$ AFD (método de subconjuntos) + micro-lab/verificación

Docente: Helder Octavio Fernández Guzmán

Sesión 4: ¿qué haremos hoy?

Periodo 1/6

- Convertir un **AFN** a un **AFD equivalente** usando el **método de subconjuntos**.
- Caso base (Sesión 3): lenguaje "contiene ab".
- Resultado esperado:
  - tabla de subconjuntos,
  - diagrama del AFD,
  - verificación de equivalencia (Automata Tutor).

Actividades en clase (bloque final): micro-lab (UC Davis + Automata Tutor) + evidencia.

Puente desde Sesión 3: la misma idea, ahora “congelada” como AFD

Periodo 1/6

- Ayer: al simular un AFN, el “estado actual” es un **conjunto** S de estados posibles.
- Hoy: cada conjunto S será un **estado del AFD**.
- Reglas clave:
  - Inicial: $S_0 = \{q_0\}$.
  - Final: S es de aceptación si $S \cap F \neq \emptyset$.

Regla mental: “un estado del AFD = un conjunto posible del AFN”.

Lenguaje objetivo: “contiene ab”

Periodo 2/6

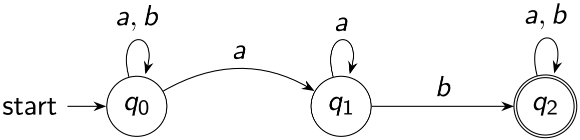
- Alfabeto: $\Sigma = \{a, b\}$.
- Lenguaje:

$$L = \{\, w \in \Sigma^* \mid w \text{ contiene la subcadena } \texttt{ab} \,\}$$

- Ejemplos:
  - Acepta: `aab`, `babab`
  - Rechaza: `bbb`, `bbaa`

AFN base (reuso Sesión 3): “contiene ab”

Periodo 2/6



- q_0: aún no confirmé ab
- q_1: acabo de ver una a candidata
- q_2: ya apareció ab (aceptación)

De “simular con conjuntos” a “definir transiciones del AFD”

- En simulación (Sesión 3), al leer un símbolo x:

$$S_{\text{nuevo}} = \delta(S, x)$$

- En subconjuntos, esa misma operación define la transición del AFD:

$$\delta'(S, x) = \delta(S, x) = \bigcup_{q \in S} \delta(q, x)$$

- Aceptación:

$$S \text{ es final} \iff S \cap F \neq \emptyset$$

Idea: no estamos cambiando el concepto; lo estamos **sistematizando**.

Definiciones para el método de subconjuntos (sin ε)

Periodo 3/6

- Nuestro AFN base **no usa** transiciones ε.
- Estado inicial del AFD: $\{q_0\}$.
- Transición del AFD:

$$\delta'(S, x) = \bigcup_{q \in S} \delta(q, x)$$

- Estados finales del AFD:

$$S \text{ es final} \iff S \cap F \neq \emptyset$$

Algoritmo de subconjuntos (versión operativa)

Periodo 3/6

1. Iniciar con el conjunto $\{q_0\}$ como único estado del AFD.
2. Mantener una lista (cola) de estados **pendientes** por expandir.
3. Para cada conjunto S pendiente:
   - calcular $\delta'(S, a)$ y $\delta'(S, b)$,
   - si aparece un conjunto nuevo, agregarlo a pendientes.
4. Marcar como finales los conjuntos que contengan algún estado final del AFN (aquí: q_2).
5. Terminar cuando no aparezcan conjuntos nuevos.

Plantilla de tabla (para construir el AFD)

Periodo 3/6

- Tabla de trabajo:
  - **Estado (conjunto)** | con a | con b | ¿final?
- Convención para nombrar conjuntos (para dibujar fácil):

$$A = \{q_0\},\ B = \{q_0, q_1\},\ C = \{q_0, q_2\},\ D = \{q_0, q_1, q_2\}$$

- Solo construimos estados **alcanzables** (no todos los $2^{|Q|}$).

Ejemplo guiado (Paso 0): estado inicial A

Periodo 4/6

- Definimos:

$$A = \{q_0\}$$

- Calculamos:

$$\delta'(A, a) = \delta(\{q_0\}, a) = \{q_0, q_1\} = B$$

$$\delta'(A, b) = \delta(\{q_0\}, b) = \{q_0\} = A$$

Lectura rápida: con b seguimos en q_0; con a aparece la hipótesis q_1.

Ejemplo guiado (Paso 1): conjunto B

Periodo 4/6

- Definimos:

$$B = \{q_0, q_1\}$$

- Con a:

$$\delta'(B, a) = \delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\} \cup \{q_1\} = B$$

- Con b:

$$\delta'(B, b) = \delta(q_0, b) \cup \delta(q_1, b) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\} = C$$

Ejemplo guiado (Paso 2): conjunto C (final)

Periodo 4/6

- Definimos:
$$C = \{q_0, q_2\} \quad \Rightarrow \quad \text{es final (contiene } q_2\text{)}$$
- Con a:
$$\delta'(C, a) = \delta(q_0, a) \cup \delta(q_2, a) = \{q_0, q_1\} \cup \{q_2\} = \{q_0, q_1, q_2\} = D$$
- Con b:
$$\delta'(C, b) = \delta(q_0, b) \cup \delta(q_2, b) = \{q_0\} \cup \{q_2\} = C$$

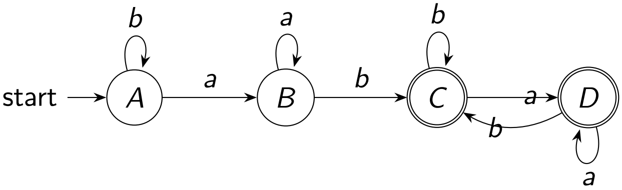
Ejemplo guiado (Paso 3): conjunto D (final)

Periodo 4/6

- Definimos:
$$D = \{q_0, q_1, q_2\} \quad \Rightarrow \quad \text{es final}$$
- Con a:
$$\delta'(D, a) = D$$
- Con b:
$$\delta'(D, b) = \{q_0, q_2\} = C$$

Cierre del ejemplo: ya no aparecen conjuntos nuevos $\Rightarrow$ terminamos la construcción.

AFD resultante (diagrama) + tabla compacta



Estado	con a	con b	¿final?
$A = \{q_0\}$	B	A	No
$B = \{q_0, q_1\}$	B	C	No
$C = \{q_0, q_2\}$	D	C	Sí
$D = \{q_0, q_1, q_2\}$	D	C	Sí

Micro-lab: construir y validar (herramientas)

Periodo 5/6

- **Parte A:** construye el **AFN** para el lenguaje "contiene `ba`" (sin ε).
- **Parte B:** completa la tabla de subconjuntos (nombra conjuntos alcanzables A,B,C,...).
- **Parte C (UC Davis):** dibuja el **AFD** resultante y prueba 2 cadenas:
  - una que **acepta** (sugerencia: `bba` o `aba`)
  - una que **rechaza** (sugerencia: `aaa` o `bbb`)
- **(Opcional) Automata Tutor:** usa **NFA Construction** y **DFA Construction** para recibir retroalimentación (no es parte de la evidencia mínima).

Evidencia mínima (entrega) + checklist

Periodo 5/6

- **Entregar 3 archivos:**
  1. **1 PDF** con: tabla de subconjuntos, AFD final, y 2 pruebas (acepta/rechaza) con justificación breve.
  2. `automata_s4_nfa.txt`: pseudocódigo del **AFN** "contiene `ba`".
  3. `automata_s4_dfa.txt`: pseudocódigo del **AFD** final (resultado por subconjuntos).
- Checklist:
  - el AFN realmente reconoce "contiene `ba`",
  - estados finales del AFD correctos (conjuntos que contienen q_2),
  - transiciones completas (sin celdas vacías),
  - solo conjuntos **alcanzables** (no inventar subconjuntos),
  - las pruebas concuerdan con el lenguaje (una acepta y una rechaza).

Cierre: lo que queda instalado

Periodo 6/6

- Hoy:
  - diseñamos un AFN para un patrón (`ba`),
  - construimos un AFD equivalente con el método de subconjuntos,
  - validamos con pruebas de cadenas (y opcionalmente con retroalimentación en herramientas).
- Próximo paso natural:
  - **minimización del AFD** (fusionar estados equivalentes) — lo veremos más adelante.
- Regla para llevarse:

"Un estado del AFD = un conjunto posible del AFN"