Teoría de Autómatas y Lenguajes Formales

Unidad 1 — Sesión 3 (12-ene-2026): AFN y ε-AFN (simulación) + puente a subconjuntos

Docente: Helder Octavio Fernández Guzmán

Sesión 3: ¿qué haremos hoy?

Periodo 1/6

- Entender qué es un **Autómata Finito No Determinista (AFN)**.
- Aprender a **ejecutar/simular** un AFN usando un **conjunto de estados**.
- Ver la idea de **transiciones** ε y el ε**-cierre** (solo lo esencial).
- Dejar listo el puente para mañana: **AFN** $\rightarrow$ **AFD** con **subconjuntos**.

Actividades en clase (bloque final): micro-lab paso a paso (UC Davis) + quiz Moodle (12).

Recordatorio express: AFD (DFA)

Periodo 1/6

- En un **AFD**, para cada estado y símbolo hay **exactamente 1** transición.
- Al ejecutar una cadena, existe **un único recorrido** posible.
- Esto hace que el diseño a veces sea **menos natural** cuando buscamos un patrón específico.

Idea guía: un AFN puede ser más cómodo de diseñar (y luego se puede convertir a AFD).

Motivación: ¿por qué AFN?

Periodo 1/6

- Un AFN puede **probar varias posibilidades** en paralelo.
- Ejemplo típico: reconocer cadenas que **contienen** un patrón (p. ej. `ab`).
- AFN **no es más poderoso** que AFD para lenguajes regulares, pero sí suele ser **más directo**.

Hoy lo veremos con un caso simple: "contiene `ab`".

¿Qué es un AFN? (definición operativa)

- Desde un estado, con un símbolo, puede haber:
  - **0** transiciones (ese camino se detiene),
  - **1** transición,
  - **varias** transiciones (*no determinismo*).
- En ejecución, el AFN puede estar en **varios estados a la vez**.

Por eso simularemos con un conjunto S de estados "posibles".

¿Cuándo acepta un AFN?

Periodo 2/6

- Un AFN **acepta** una cadena si:
  - **existe al menos un recorrido** que consume toda la cadena y termina en aceptación.
- Contraste mental:
  - AFD: “acepta si **su único recorrido** termina en aceptación”.
  - AFN: “acepta si **alguno de sus recorridos** termina en aceptación”.

Simulación con conjunto de estados (idea central)

Periodo 2/6

- Mantendremos un conjunto S de estados activos.
- Inicio: $S_0 = \{q_0\}$.
- Para cada símbolo x leído:

$$S_{\text{nuevo}} = \delta(S, x) \quad \text{(unión de transiciones desde todos los estados en } S \text{ con } x\text{)}$$

- Al final, el AFN acepta si S contiene **algún** estado de aceptación.

Ejemplo del día: lenguaje “contiene ab”

Periodo 3/6

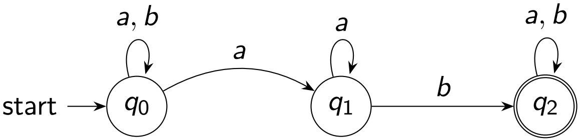
- Alfabeto: $\Sigma = \{a, b\}$.
- Lenguaje:

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

- Intuición: cuando aparece una a, el AFN puede “apostar” a que esa a inicia el patrón ab.

AFN para “contiene ab” (diagrama)

Periodo 3/6



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

Simulación guiada (acepta): aab

Periodo 3/6

Paso	**Conjunto de estados**
Inicio	$S_0 = \{q_0\}$
Leer `a`	$S_1 = \{q_0, q_1\}$
Leer `a`	$S_2 = \{q_0, q_1\}$
Leer `b`	$S_3 = \{q_0, q_2\}$

Conclusión: como $q_2 \in S_3$, la cadena `aab` **acepta**.

Simulación guiada (rechaza): bbb

Periodo 3/6

Paso	**Conjunto de estados**
Inicio	$S_0 = \{q_0\}$
Leer b	$S_1 = \{q_0\}$
Leer b	$S_2 = \{q_0\}$
Leer b	$S_3 = \{q_0\}$

Conclusión: $q_2 \notin S_3$, por tanto bbb **rechaza**.

¿Qué es una transición ε?

Periodo 4/6

- Una transición ε permite cambiar de estado **sin leer símbolo**.
- Se usa para modelar "saltos" o decisiones internas.
- No agrega símbolos nuevos al alfabeto: solo permite moverse con **entrada vacía**.

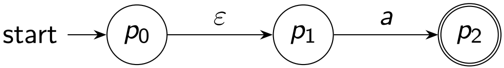
Hoy: entendemos la idea y cómo afecta a la simulación (sin formalismo pesado).

ε-cierre (intuición práctica)

Periodo 4/6

- ε-cierre(S): todos los estados alcanzables desde S usando solo transiciones ε (0 o más veces).
- En una simulación de ε-AFN, uno suele:
  - expandir el conjunto con ε-cierre,
  - leer un símbolo,
  - volver a expandir con ε-cierre.

Mini-ejemplo:



Puente a mañana: ¿por qué "subconjuntos"?

Periodo 4/6

- En AFN, el "estado actual" puede ser un **conjunto** S.
- Para construir un AFD equivalente:
  - cada estado del AFD puede representar un **subconjunto** de estados del AFN.
- Mañana haremos el método AFN $\rightarrow$ AFD (subconjuntos) y verificaremos con ejemplos.

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

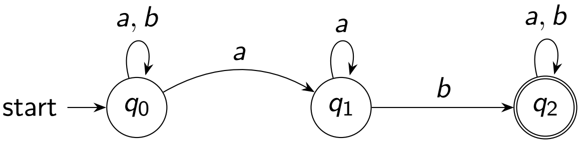
Micro-lab (UC Davis): instrucciones

Periodo 5/6

- En UC Davis, construye el AFN del ejemplo (mismos estados y transiciones).
- Simula **paso a paso**:
  - una cadena que **acepta** (sugerencia: `aab` o `bab`)
  - una cadena que **rechaza** (sugerencia: `bbb` o `baa`)
- Toma **dos capturas** (acepta / rechaza).
- Observa cómo cambia el **conjunto de estados** en cada símbolo.

AFN del micro-lab (diagrama grande)

Periodo 5/6



Tip: si dudas, verifica el conjunto de estados después de cada símbolo.

Quiz en clase (Moodle) + cierre

Periodo 6/6

- **Quiz Moodle (12 preguntas)** — se resuelve en clase.
- Cierre rápido:
  - ¿Sé explicar AFN vs AFD?
  - ¿Puedo simular un AFN con conjuntos de estados?
  - ¿Entiendo la idea de ε y del ε-cierre?
- Preparación para mañana:
  - AFN $\rightarrow$ AFD (subconjuntos) + práctica de verificación.

Apéndice: recordatorio AFD (4 patrones trabajados)

Periodo 6/6

- Termina en ab: memoria mínima $\rightarrow$ últimos 1–2 símbolos.
- Termina en ba: memoria mínima $\rightarrow$ últimos 1–2 símbolos.
- Contiene aa: memoria mínima $\rightarrow$ "ya vi aa" + si el último fue a.
- Contiene bb: memoria mínima $\rightarrow$ "ya vi bb" + si el último fue b.

Idea puente: en AFN la "memoria" puede explorarse con varios caminos (conjunto de estados).

Apéndice: analogía AFD $\leftrightarrow$ AFN (para la simulación)

- **AFD:** estado actual = **un** estado; recorrido único.
- **AFN:** estado actual = **conjunto** de estados; varios recorridos posibles.
- **Aceptación AFN:** basta con que exista **un** camino que llegue a aceptación.
- **Mañana (subconjuntos):** cada estado del AFD equivalente representa un **conjunto** posible del AFN.