

# Teoria de Automatas y Lenguajes Formales

Unidad 1 — Sesión 8 (Lun 19-ene-2026): Minimización básica de AFD

Docente: Helder Octavio Fernandez Guzman

- Objetivo: reducir un AFD sin cambiar el lenguaje que reconoce.
- Ruta (3 pasos):
  - ➊ **Limpieza**: eliminar estados **inalcanzables**.
  - ➋ **Particiones**: refinar bloques hasta estabilizar.
  - ➌ **AFD minimo**: construir un AFD por bloques equivalentes.
- Hoy: ejemplo guiado + actividad (evidencia con pruebas).

- Herramienta principal: **UC Davis Automaton Simulator**.
- Representacion recomendada: **YAML** (mismos nombres de estados que en las figuras).
- Evidencia minima para tu actividad:
  - lista de estados alcanzables (antes de minimizar),
  - particiones  $P_0, P_1, \dots$  hasta estable,
  - AFD minimo (diagrama o tabla),
  - tabla de pruebas (al menos 6 cadenas: 3 acepta / 3 rechaza).

- **Alcanzable:** existe un camino desde el estado inicial.
- **AFD completo:** en cada estado hay transicion para cada simbolo de  $\Sigma$ .
- **Estados equivalentes:** no existe cadena que los distinga (misma aceptacion futura).
- **Estados distinguibles:** existe una cadena que hace que uno acepte y el otro rechace.

**Idea:** la minimizacion *fusiona* estados equivalentes.

## Fase 1: limpieza

Eliminar estados **inalcanzables** (no aportan al lenguaje).

## Fase 2: particiones

- Particion inicial:

$$P_0 = \{F, Q \setminus F\}.$$

- Refinamiento: separar estados de un bloque si, con algun simbolo, caen a **bloques distintos**.
- Parar cuando la particion se estabiliza.

- Para cada bloque  $B$  de la particion actual:
  - para cada estado  $q \in B$ , registrar

$(\text{bloque}(\delta(q, a)), \text{bloque}(\delta(q, b)))$ .

- si dos estados de  $B$  tienen pares distintos, **separarlos**.
- Consejo: usar una mini-tabla de *firma* por estado.

# Errores tipicos (para evitarlos)

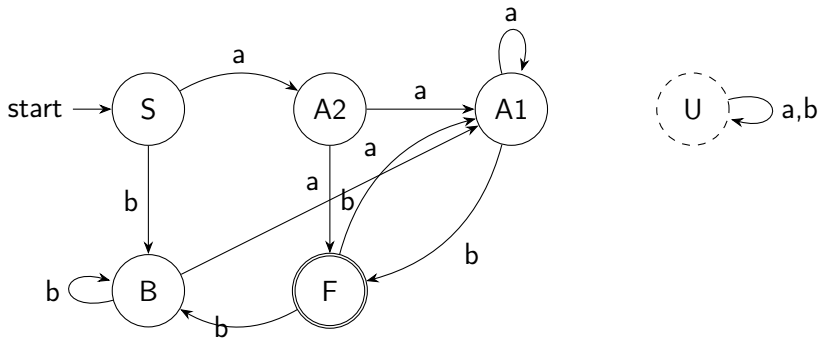
Periodo 2/6

- Olvidar eliminar inalcanzables.
- Refinar comparando *estados* en vez de *bloques destino*.
- Construir el AFD minimo sin actualizar transiciones al *bloque* correspondiente.
- No validar con pruebas (IN / NOT IN).

Sea  $\Sigma = \{a, b\}$ .

- Lenguaje  $L$ : cadenas que **terminan** en  $ab$ .
- Objetivo: minimizar un AFD (intencionalmente **no minimo**).
- Resultado final (referencia): AFD minimo con estados  $q_0, q_1, q_2$ .





*Nota:* U esta dibujado como estado aislado (candidato a inalcanzable).

# Tabla de transición del AFD (para refinar)

Periodo 3/6

Estado	<i>a</i>	<i>b</i>
S	A2	B
A2	A1	F
A1	A1	F
B	A1	B
F	A1	B
U	U	U

- Estado inicial: S. Finales: {F}.
- $\Sigma = \{a, b\}$  (AFD completo).

- Desde S:
  - con a llego a A2, luego a A1 y F.
  - con b llego a B.
- Por tanto, alcanzables:  $\{S, A2, A1, B, F\}$ .
- U no es alcanzable  $\Rightarrow$  **se elimina** antes de minimizar.

- Finales:  $F = \{F\}$ .
- No finales:  $Q \setminus F = \{S, A2, A1, B\}$ .

$$P_0 = \{\{F\}, \{S, A2, A1, B\}\}.$$

Particion actual:  $\{\{F\}, \{S, A2, A1, B\}\}$ .

Estado	$\text{bloque}(\delta(\cdot, a))$	$\text{bloque}(\delta(\cdot, b))$
S	(no final)	(no final)
A2	(no final)	(final)
A1	(no final)	(final)
B	(no final)	(no final)

Conclusiones:

- A2 y A1 tienen la misma firma  $\Rightarrow$  quedan juntos.
- S y B tienen la misma firma  $\Rightarrow$  quedan juntos.

$$P_1 = \{\{F\}, \{A2, A1\}, \{S, B\}\}.$$

- Verificacion rapida: al aplicar refinamiento otra vez, no se separa mas.
- Bloques equivalentes:
  - $A2 \equiv A1$
  - $S \equiv B$

- Cada bloque de  $P_1$  se vuelve un estado del AFD minimo.
- Transicion:

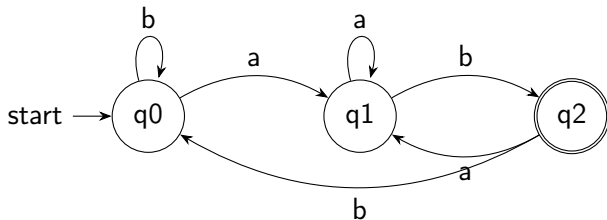
$$[q] \xrightarrow{\sigma} [\delta(q, \sigma)]$$

- Inicial: el bloque que contiene a S.
- Final: el bloque que contiene a F.

**Convencion para el minimo (referencia):**  $q_0, q_1, q_2$ .

# AFD minimo (TikZ): termina en ab

Periodo 4/6





Cadena $w$	Esperado (termina en $ab$ )
$ab$	Acepta
$aab$	Acepta
$bab$	Acepta
$\epsilon$	Rechaza
$a$	Rechaza
$aba$	Rechaza

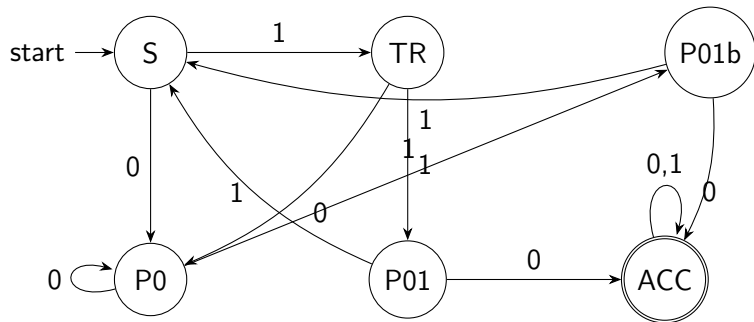
- La validacion debe dar lo mismo en el AFD original y en el minimo.

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

- Lenguaje: cadenas que **contienen** la subcadena 010.
- Se entrega un AFD intencionalmente **no minimo**.
- Tu tarea: limpiar alcanzables, refinar particiones, construir minimo y validar.

# AFD no minimo: contiene 010

Periodo 5/6



# Entregable minimo (checklist)

Periodo 5/6

- (1) Alcanzables: lista desde S.
- (2) Particiones:  $P_0, P_1, \dots$  hasta estable.
- (3) AFD minimo: diagrama o tabla de transicion.
- (4) Tabla de pruebas: **6 cadenas** (3 acepta / 3 rechaza) con esperado.

# Set minimo de pruebas sugerido (sin revelar el minimo)

Periodo 5/6

Cadena $w$	Esperado (contiene 010)
010	Acepta
0010	Acepta
1010	Acepta
$\epsilon$	Rechaza
01	Rechaza
111	Rechaza

# Checklist de depuracion (antes de entregar)

Periodo 6/6

- DFA completo: transicion para cada simbolo en cada estado.
- Alcanzables: elimine inalcanzables antes de minimizar.
- Refinamiento: compare **bloques destino**, no nombres de estados.
- Construccion final: reemplace transiciones por **transiciones entre bloques**.
- Validacion: pruebe 6 cadenas (IN / NOT IN).

- Hoy: minimizacion por particiones (limpieza + refinamiento + minimo).
- Proxima sesion: **Pumping lemma (regular)**: estructura y patrones.