

Teoría de Autómatas y Lenguajes Formales

Unidad 1 — Taller (Miércoles 14-ene-2026): AFN → AFD (subconjuntos) + 5 tareas
Moodle

Docente: Helder Octavio Fernández Guzmán

- Modalidad: **online sincrónica**, trabajo **individual**.
- Enfoque: **ABP** + construcción sistemática:
 - inferir el lenguaje que reconoce un AFN,
 - convertir a AFD por **método de subconjuntos**,
 - validar con **transición de estados** sobre el AFD.
- Control del avance: 5 tareas en Moodle (**P1–P5**).

Evidencias clave del día:

- `tabla_subconjuntos.txt` (P3)
- `P4_evidencia_afd.pdf` (diagrama + tabla + resumen)
- `P5_verificacion_afd.pdf` (8 pruebas + 2 capturas + reflexión)

- Alfabeto: $\Sigma = \{a, b\}$.
- Estados: $Q = \{q_0, q_1, q_2, q_3\}$.
- Inicial: q_0 .
- Final: $F = \{q_3\}$.

Transiciones δ :

Estado	con a	con b
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_1\}$	$\{q_2\}$
q_2	$\{q_3, q_1\}$	$\{q_0\}$
q_3	$\{q_3\}$	$\{q_3\}$

Inferir la condición del lenguaje con evidencia; luego formalizar con subconjuntos.

Convenciones del taller (para evitar ambigüedades)

Periodo 1/6

- En el AFD por subconjuntos, cada estado es un **conjunto** de estados del AFN.
- Estado inicial del AFD: $S_0 = \{q_0\}$.
- Estado final del AFD: S es final si $S \cap F \neq \emptyset$ (equiv.: **contiene** q_3).
- **Sin \emptyset como estado:**
 - Si alguna transición produjera “vacío”, se marca como – (sin transición).
 - En transición de estados: llegar a – implica **rechazo**.
- **Entregas:** únicamente por **Moodle** (5 tareas).

Propósito: inferir el lenguaje que reconoce el AFN con evidencia.

Qué hacer (individual):

- Escriba una **hipótesis** (frase clara): “Acepta cadenas que ...”.
- Proponga **6 cadenas**:
 - 3 que espera que el autómata **acepte**,
 - 3 que espera que el autómata **rechace**.
- Para cada cadena: **justificación breve** (1 línea).

Entrega (Moodle #1, preferencia): texto en línea con hipótesis + 6 cadenas (3A/3R) + justificación breve.

Objetivo del periodo: iniciar la tabla y fijar el mapeo de conjuntos sin errores.

Qué hacer:

- Defina el estado inicial del AFD: $A = \{q_0\}$.
- Calcule:

$$\delta'(S, x) = \bigcup_{q \in S} \delta(q, x) \quad \text{para } x \in \{a, b\}.$$

- Cree nombres para conjuntos nuevos: B, C, D, \dots (sin duplicar conjuntos existentes).
- Complete, como mínimo, transiciones de A y B con a y b .

Entrega (Moodle #2, preferencia): texto en línea con:

- mapeo de todos los conjuntos generados en el periodo (A, B, C, D, \dots),
- $\delta'(A, a), \delta'(A, b), \delta'(B, a), \delta'(B, b)$.

Objetivo del periodo: **cerrar** la tabla (ya no aparecen conjuntos nuevos).

Qué hacer:

- Expanda los conjuntos pendientes hasta que no aparezcan estados nuevos.
- Construya la tabla (solo alcanzables) con el formato:

Estado ; Conjunto ; con a -> ; con b -> ; Final?

- Marque **Final?** Sí si el conjunto contiene q_3 .
- Liste **Finales del AFD**.

Entrega (Moodle #3): subir `tabla_subconjuntos.txt` con tabla cerrada + finales identificados.

Objetivo del periodo: dejar evidencia verificable del AFD equivalente.

Entregar 1 PDF: P4_evidencia_afd.pdf

Debe incluir:

- 1 captura del **diagrama del AFD** (se debe ver estado inicial y estados finales).
- 1 captura de la **tabla de transiciones** del AFD (puede ser del simulador o la tabla generada desde tabla_subconjuntos.txt).
- **Resumen (3–6 líneas):** cuántos estados alcanzables obtuvo, cuáles son finales, y un comentario breve.

Plan B (si falla el simulador): diagrama dibujado + tabla desde tabla_subconjuntos.txt, todo en el mismo PDF.

Objetivo del periodo: verificar el AFD con evidencia y cerrar con reflexión ABP.

Entregar 1 PDF: P5_verificacion_afd.pdf

Debe incluir:

- **8 pruebas exactas** (4 A / 4 R), cada una con:
 - cadena,
 - **transición de estados** ($A \rightarrow B \rightarrow C \rightarrow \dots$),
 - decisión A/R,
 - etiqueta (corta / trampa / patrón al final / patrón al medio).
- **2 capturas del simulador:** 1 acepta y 1 rechaza (cadena visible + resultado).
- **Reflexión (obligatoria):** ¿confirmé mi hipótesis? (Sí/No) + razones + pasos clave/mejoras.

