

# AUTÓMATAS Y LENGUAJES

Lenguajes Regulares - Autómatas Finitos - AÑO 2024

LIC. EN CS. DE LA COMPUTACIÓN - 4to. AÑO

# Objetivos de la Materia

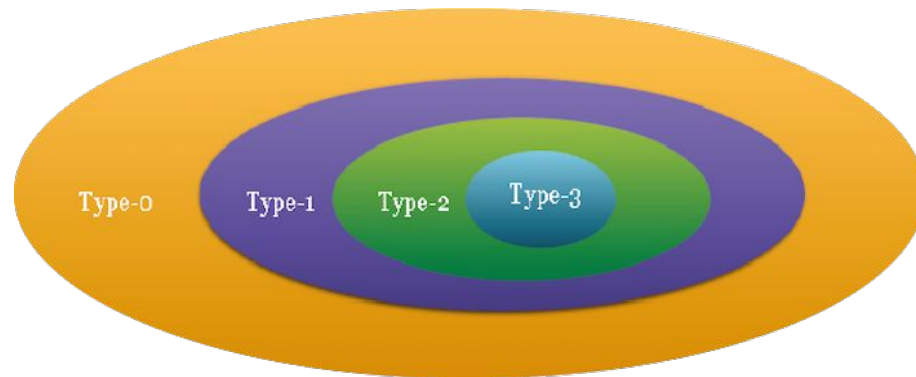
- Proporcionar herramientas conceptuales que permitan comprender los conceptos centrales de la Teoría de Lenguajes Formales, tales como: Autómatas, Gramáticas y su aplicación, en particular en el contexto de los Lenguajes de Programación.
- Analizar e individualizar las características de los distintos tipos de lenguajes según la Jerarquía de Chomsky. Propiedades y Equivalencias.
- Aplicar los conceptos necesarios para la implementación de un Analizador Lexicográfico.
- Hacer una introducción teórica al concepto de análisis sintáctico, junto con las respectivas técnicas de análisis: Top-Down y Bottom-Up, para sentar las bases para el estudio posterior de la asignatura Diseño y Construcción de Compiladores.
- Promover la revisión crítica de los temas abordados, propiciando la discusión y el trabajo en grupo y colaborativo.
- .....

# Objetivos de la clase

- ➔ Abordar el estudio de los diferentes tipos de lenguajes, comenzando por los Lenguajes Regulares (LR).
- ➔ Estudiar los siguientes dispositivos descriptores de los LR:
  - Autómatas Finitos Determinísticos (AFD).
  - Autómatas Finitos No Determinísticos (AFND).
  - Autómatas Finitos No Determinísticos con transiciones  $\epsilon$  (AFND- $\epsilon$ ).
  - Equivalencias entre estas máquinas.
- ➔ Comprender la teoría de autómatas para conocer su relación con los Lenguajes de programación.

# Lenguajes Regulares o Tipo 3

*Comenzamos el estudio de los diferentes lenguajes basados en la Jerarquía de Chomsky:*



Los lenguajes regulares son interesantes por su simplicidad, esto hace que sean fáciles de manipular. Además incluyen características relevantes, como: los mecanismos de búsqueda provistos por varios editores de texto (vi, emacs), así como por el shell de Unix y todas las herramientas asociadas para procesamiento de texto (sed, awk, perl), se basan en lenguajes regulares.

# Lenguajes Regulares o Tipo 3

*Los lenguajes regulares se pueden describir usando:*

- *Autómata Finitos Determinísticos y No Determinísticos, para reconocerlos.*
- *Expresiones Regulares, para denotarlos.*
- *Gramáticas Regulares, para generarlos.*

# Autómatas Finitos

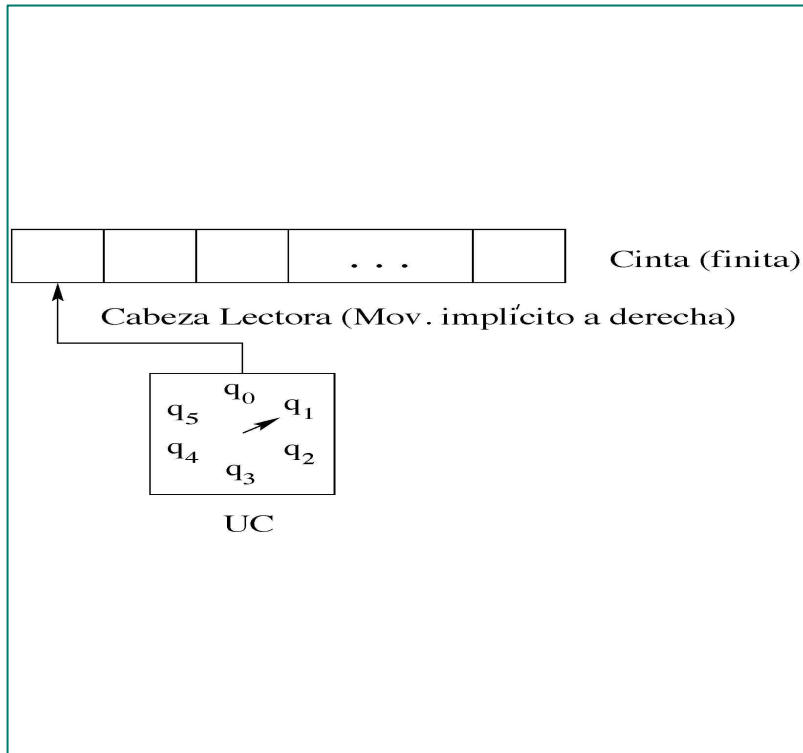
## Idea Intuitiva

Un autómatata finito tiene un conjunto de estados, y su “control” se mueve de estado en estado, en respuesta a “entradas” externas. Los autómatas finitos se dividen en diversas clases, dependiendo de si su control es “determinístico” (lo que significa que el autómatata no puede estar en más de un estado simultáneamente) o “no determinístico” (lo que significa que puede estar en varios estados al mismo tiempo).

Los estados representan la memoria, en estas máquinas.

# Autómatas Finitos Determinísticos

## Gráficamente



Cinta finita, cabeza lectora,  
movimiento implícito a  
derecha

Ejemplo

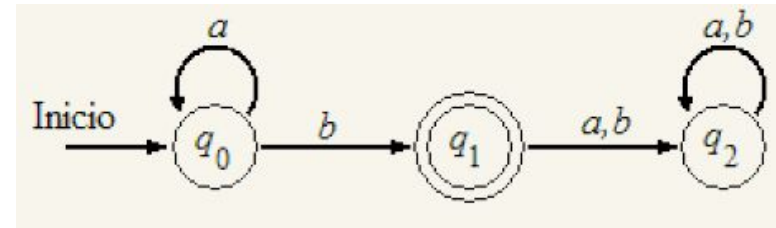


Diagrama de Transición

# Autómatas Finitos Determinísticos

## Formalización

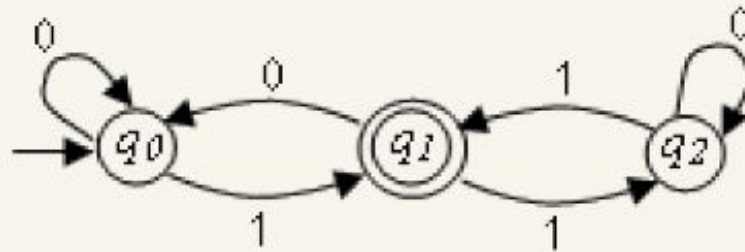
Un AF Determinístico (AFD) es una 5-tupla  $M = (Q, \Sigma, \delta, q_0, F)$ , donde  $Q$  es un conjunto finito de estados,  $\Sigma$  el alfabeto de entrada,  $q_0 \in Q$  el estado inicial,  $F \subseteq Q$  es el conjunto de estados finales. La función de transición,  $\delta : Q \times \Sigma \rightarrow Q$ , es una *función total*.

¿qué significa que la función de transición sea total?



# Autómatas Finitos Determinísticos

## Ejemplo



Representa al AFD  $M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$ , donde  $\delta$  esta dada por:

$$\delta(q_0, 0) = q_0 \quad \delta(q_0, 1) = q_1$$

$$\delta(q_1, 0) = q_0 \quad \delta(q_1, 1) = q_2$$

$$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_1$$

*¿Qué lenguaje aceptará este AFD?*

# Autómatas Finitos Determinísticos

## Tabla de Transición

$$A = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_0\})$$

Con transiciones:

$$\delta(q_0, 0) = q_2 \quad \delta(q_0, 1) = q_1$$

$$\delta(q_1, 0) = q_3 \quad \delta(q_1, 1) = q_0$$

$$\delta(q_2, 0) = q_0 \quad \delta(q_2, 1) = q_3$$

$$\delta(q_3, 0) = q_1 \quad \delta(q_3, 1) = q_2$$

Tabla de transiciones:

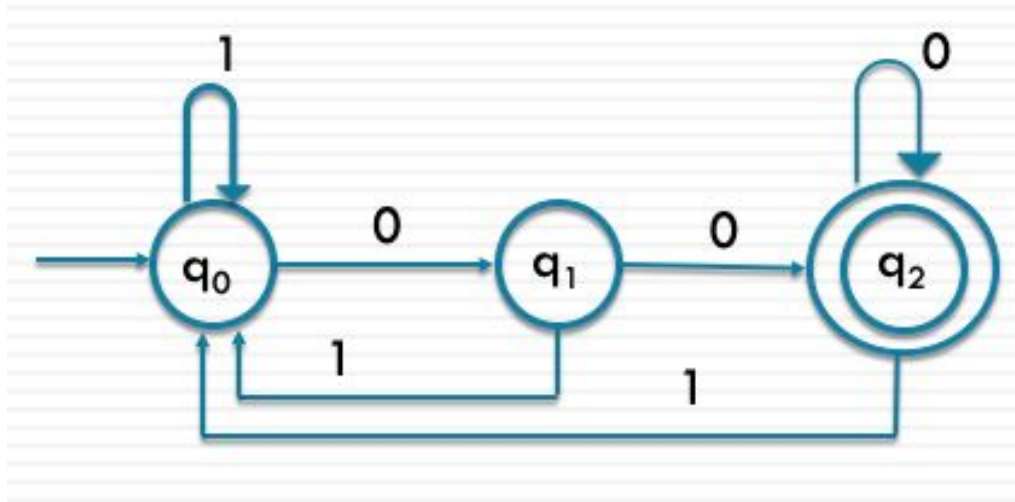
	0	1
* $\rightarrow$ $q_0$	$q_2$	$q_1$
$q_1$	$q_3$	$q_0$
$q_2$	$q_0$	$q_3$
$q_3$	$q_1$	$q_2$

# Autómatas Finitos Determinísticos

## Ejercicio

*Construir un AFD para el siguiente lenguaje:*

$$L_1 = \{w \in \{0,1\}^* \mid w \text{ termina en } 00\}$$



# Ejercicios

1. Construya un AFD que reconozca el conjunto de todas las cadenas sobre  $\Sigma = \{a,b\}$  que inicien con el prefijo **ab**.
2. Construya un AFD que acepte todas las cadenas sobre  $\Sigma = \{0,1\}$ , excepto aquellas que contengan la subcadena **001**.
3. Proponga un ejercicio, es decir piense en un lenguaje para el cual desee construir un AFD.

# Autómatas Finitos Determinísticos

## Función de transición extendida ( $\hat{\delta}$ )

$$\hat{\delta} : Q \times \Sigma^* \rightarrow Q$$

**Definición recursiva de  $\hat{\delta}$ :**

1.  $\hat{\delta}(q, \lambda) = q$
2.  $\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), \sigma)$  con  $\sigma \in \Sigma$  y  $x \in \Sigma^*$  y  $w = x\sigma$

**Luego, una cadena  $w$  es aceptada por un AFD**

**$M = (Q, \Sigma, \delta, q_0, F)$  si  $\hat{\delta}(q_0, w) \in F$ .**

En consecuencia, el lenguaje aceptado por  $M$  es:

$$L(M) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}.$$

Aplicar la función de transición extendida a la cadena 10100, para el AFD construido previamente.

# Autómatas Finitos Determinísticos

## Función de transición extendida (Ejemplo)

Aplicar la función de transición extendida ( $\hat{\delta}$ ), a la cadena 10000, para determinar si es aceptada por el autómata dado en el primer ejemplo:

$$\hat{\delta}(q_0, \lambda) = q_0$$

$$\hat{\delta}(q_0, 1) = \delta(\hat{\delta}(q_0, \lambda), 1) = q_0$$

$$\hat{\delta}(q_0, 10) = \delta(\hat{\delta}(q_0, 1), 0) = q_1$$

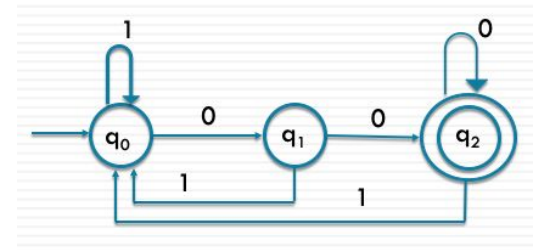
$$\hat{\delta}(q_0, 100) = \delta(\hat{\delta}(q_0, 10), 0) = q_2$$

$$\hat{\delta}(q_0, 1000) = \delta(\hat{\delta}(q_0, 100), 0) = q_2$$

$$\hat{\delta}(q_0, 10000) = \delta(\hat{\delta}(q_0, 1000), 0) = q_2$$

1.  $\hat{\delta}(q, \lambda) = q$

2.  $\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), \sigma)$  con  $\sigma \in \Sigma$  y  $x \in \Sigma^*$  y  $w = x\sigma$



# Autómatas Finitos Determinísticos

## Configuraciones o Descripciones Instantáneas

Una *configuración* representa el estado actual en el que se encuentra un AFD y la porción de cadena de entrada, que aún no se ha consumido. Por lo tanto es un par  $(q, x)$ . Se puede definir una relación que nos permite obtener una configuración, a partir de otra, de la siguiente manera:

$$(q, ax) \vdash (q', x) \text{ donde } a \in \Sigma, \text{ sii } \delta(q, a) = q'$$

También se puede definir esta relación, para indicar como obtener una configuración, a partir de otra, en 0 o más pasos  $(\vdash)^*$ , como la clausura reflexo transitiva de  $\vdash$ .

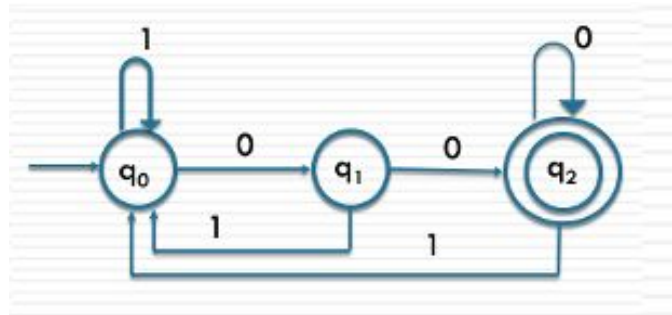
Entonces es posible dar la siguiente definición de lenguaje aceptado por un AFD  $M = (Q, \Sigma, \delta, q_0, F)$ , como:

$$L(M) = \{x \in \Sigma^* / \exists f \in F, (q_0, x) \vdash^* (f, \lambda)\}$$



# Autómatas Finitos Determinísticos

## Configuraciones o Descripciones Instantáneas (Ejemplo)



$(q_0, 101000) \vdash (q_0, 01000) \vdash (q_1, 1000) \vdash (q_0, 000) \vdash (q_1, 00)$   
 $\vdash (q_2, 0) \vdash (q_2, \lambda)$

$(q_2, \lambda)$  representa una configuración final, dado que  $q_2$  es un estado final y  $\lambda$  representa que se consumió la cadena de entrada dada, o lo que es lo mismo, que no hay más símbolos en la cinta.



Los autómatas finitos son un modelo útil para muchas aplicaciones tanto de hardware como de software, de las cuales podemos citar:

- Para examinar textos, para encontrar ocurrencias de palabras, frases, u otros patrones.
- Para análisis lexicográfico de un compilador típico.

- Para verificación de sistemas, tales como protocolos de comunicación o protocolos para intercambio seguro de información, entre otros.

- Para chequeo y diseño de circuitos digitales.
- Para modelización.

*Para reconocimiento de Lenguajes Regulares*

# Autómatas Finitos Determinísticos

## Ejercicio

Se desea diseñar un dispositivo que, dada una cadena formada por números binarios, encuentre las ocurrencias de la palabra clave 1011 y sirva de base para un recuento de sus apariciones. Nótese que si la cadena fuera, por ejemplo, 0101011011011, se detectarían dos ocurrencias de la palabra clave (subrayadas), no considerando el “1” de la séptima posición como inicio de otra ocurrencia.

Se pide construir el AFD correspondiente.

**Nota:** no es tarea de este AFD contabilizar el número de palabras claves encontradas, solo detectarlas.

Se deja como ejercicio.

# Autómata Finito No Determinístico

## Idea Intuitiva

- Un autómata finito no determinístico tiene la habilidad de estar en varios estados a la vez.
- Sin embargo, los AFND's son simplemente otra versión de AF's que no agregan potencia a los AFD's. Es decir, reconocen el mismo tipo de lenguajes, los regulares o tipo 3.
- Pero en general los AFND's permiten simplificar la tarea de diseño de autómatas finitos.

# Autómatas Finitos No Determinísticos

## Formalización

Al igual que un AFD, un AFND tiene un conjunto finito de estados, un conjunto finito de símbolos, un estado de comienzo y un conjunto de estados de aceptación.

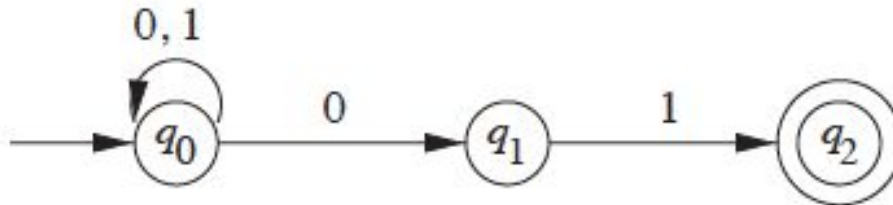
**Definición formal:** un AFND es una 5-tupla  $M = (Q, \Sigma, \delta, q_0, F)$ , donde  $Q$  es un conjunto finito de estados,  $\Sigma$  el alfabeto de entrada,  $q_0 \in Q$  el estado inicial,  $F \subseteq Q$  es el conjunto de estados finales. La función de transición,  $\delta : Q \times \Sigma \rightarrow 2^Q$ .

El concepto de no determinismo juega un rol central en teoría de lenguajes y computación.

# Autómatas Finitos No Determinísticos

## Ejemplo

- Construir un AFND que reconozca el lenguaje  $L \subseteq \{0,1\}^*$ , tal que las cadenas terminan en 01:

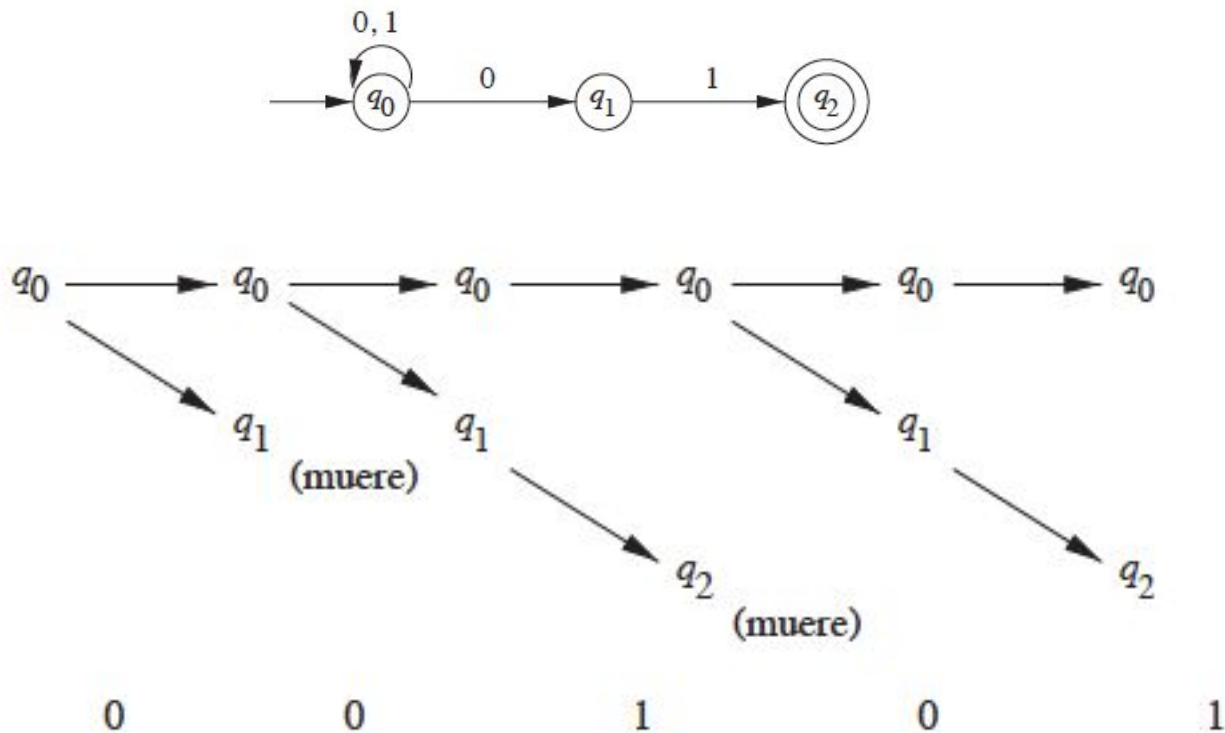


*Construir la tabla de transición para este AFND.*

# Autómatas Finitos No Determinísticos

## No determinismo - Gráficamente

Se puede visualizar gráficamente el funcionamiento del AFND:



# Autómatas Finitos No Determinísticos

## Lenguaje Aceptado

*Definición recursiva de  $\hat{\delta}$ :*

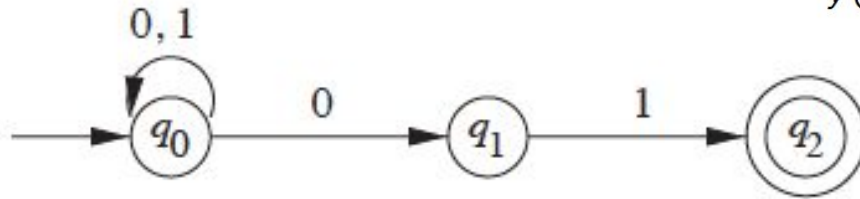
1.  $\hat{\delta}(q, \lambda) = \{q\}$
2.  $\hat{\delta}(q, w) = \{r_1, r_2, \dots, r_m\}$ ,  $w = x\sigma$ ,  $\hat{\delta}(q, x) = \{p_1, p_2, \dots, p_k\}$   
 $y \bigcup_{i=1}^k \delta(p_i, \sigma) = \{r_1, r_2, \dots, r_m\}$

Si  $M$  es un AFND, luego  $L(M) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$

# Autómatas Finitos No Determinísticos

Ejemplo  $\hat{\delta}$ :

1.  $\hat{\delta}(q, \lambda) = \{q\}$
2.  $\hat{\delta}(q, w) = \{r_1, r_2, \dots, r_m\}, w = x\sigma, \hat{\delta}(q, x) = \{p_1, p_2, \dots, p_k\}$   
 $y \bigcup_{i=1}^k \delta(p_i, \sigma) = \{r_1, r_2, \dots, r_m\}$



1.  $\hat{\delta}(q_0, \lambda) = \{q_0\}.$
2.  $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}.$
3.  $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}.$
4.  $\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}.$
5.  $\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}.$
6.  $\hat{\delta}(q_0, 00101) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}.$



# Equivalencia AFD - AFND

- ❖ A partir de cualquier AFND se puede obtener un AFD equivalente, esto involucra una importante construcción llamada **construcción del subconjunto**, la cual se basa en la idea de construir todos los subconjuntos del conjunto de estados del AFND.
- ❖ Además esta construcción permite garantizar que los lenguajes reconocidos por AFD's son también reconocidos por AFND's, o sea los lenguajes regulares.

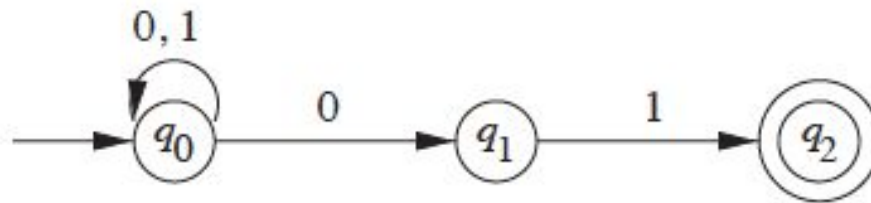
## Construcción de la máquina subconjunto

La construcción del subconjunto comienza a partir de un AFND  $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ . El objetivo es construir un AFD  $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$  tal que  $L(D) = L(N)$ .

- $Q_D$  es el conjunto de subconjuntos de  $Q_N$ , es decir es el conjunto potencia de  $Q_N$ .
- $F_D$  es el conjunto de subconjuntos  $S$  de  $Q_N$  tal que  $S \cap F_N \neq \emptyset$
- Para cada conjunto  $S \subseteq Q_N$  y para cada símbolo de entrada  $a \in \Sigma$ :  $\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$

# Aplicación de la construcción de la máquina subconjunto

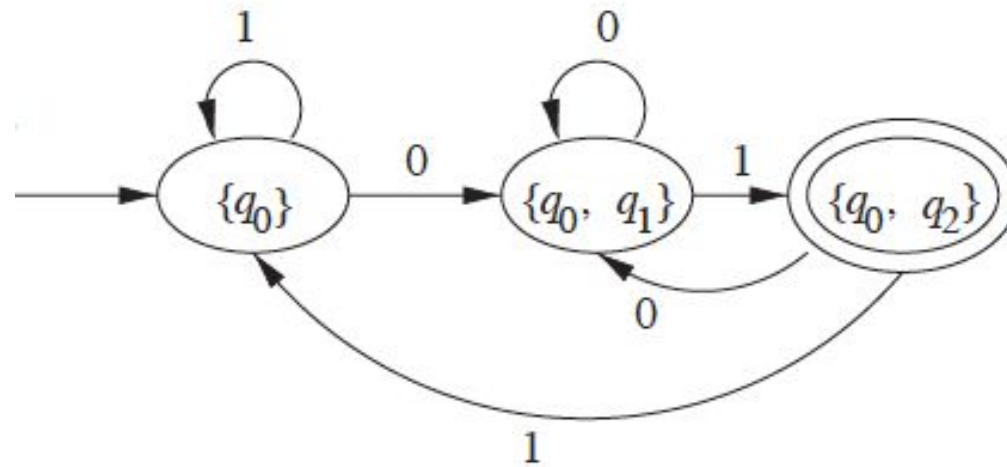
Aplicar esta construcción al siguiente AFND y corroborar que el resultado es correcto (desarrollado en clase):



# Aplicación de la construcción de la máquina subconjunto (Cont.)

El AFD resultante es:

AFD **D**



# Formalización equivalencia AFND - AFD

## Teorema:

Si  $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$  es el AFD construido a partir del AFND  $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$  por la construcción del subconjunto, entonces  $L(D) = L(N)$ .

## Demostración:

Lo que debemos probar es que  $\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$ , lo haremos por inducción sobre  $|w|$ .

Notar que cada una de las funciones  $\hat{\delta}$  retorna un conjunto de estados desde  $Q_N$ , pero  $\hat{\delta}_D$  interpreta este conjunto como uno de los estados de  $Q_D$  (el cual es el conjunto potencia de  $Q_N$ ), mientras  $\hat{\delta}_N$  interpreta este conjunto como un subconjunto de  $Q_N$ .

# Formalización equivalencia AFND - AFD

## Base:

Sea  $|w| = 0$ , esto es  $w = \lambda$ . Por las definiciones básicas de  $\hat{\delta}$  para AFD's y AFND's, ambas  $\hat{\delta}_D(\{q_0\}, \lambda)$  y  $\hat{\delta}_N(q_0, \lambda)$  son  $\{q_0\}$ .

## Inducción:

Sea  $w$  de longitud  $n + 1$ , asumamos que la sentencia se cumple para longitud  $n$ . Tomemos  $w = xa$ , con  $a$  el símbolo final de  $w$ . Por la hipótesis inductiva  $\hat{\delta}_D(\{q_0\}, x) = \hat{\delta}_N(q_0, x)$ . Sean ambos de estos conjuntos estados de  $N$ ,  $\{p_1, p_2, \dots, p_k\}$ . La parte inductiva de la definición de  $\hat{\delta}$  para AFND's nos dice que:

$$\hat{\delta}_N(q_0, w) = \cup_{i=1}^k \delta_N(p_i, a) \quad (1)$$

# Formalización equivalencia AFND - AFD

La construcción del subconjunto nos dice por otro lado que:

$$\delta_D(\{p_1, p_2, \dots, p_k\}, a) = \cup_{i=1}^k \delta_N(p_i, a) \quad (2)$$

Usemos (2) y la HI  $\hat{\delta}_D(\{q_0\}, x) = \{p_1, p_2, \dots, p_k\}$  :

$$\hat{\delta}_D(\{q_0\}, w) = \delta_D(\hat{\delta}_D(\{q_0\}, x), a) = \delta_D(\{p_1, p_2, \dots, p_k\}, a) = \cup_{i=1}^k \delta_N(p_i, a) \quad (3)$$

Las ecuaciones (1) y (3) demuestran que

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w).$$

Cuando observamos que  $D$  y  $N$  aceptan  $w$  sí y sólo si  $\hat{\delta}_D(\{q_0\}, w)$  o  $\hat{\delta}_N(q_0, w)$ , respectivamente, contienen un estado en  $F_N$ , completamos la prueba de que  $L(D) = L(N)$ .

# Formalización equivalencia AFD - AFND

**Teorema:** Un lenguaje  $L$  es aceptado por algún AFD sí y sólo sí  $L$  es aceptado por algún AFND.

**Demostración:**

(Sí) Es la demostración del teorema anterior.

(Sólo sí) Tenemos que convertir un AFD en un AFND equivalente. Intuitivamente si tenemos el diagrama de transición para un AFD, lo podemos interpretar como el diagrama de transición de un AFND, el cual tiene una única elección de transición en cualquier situación.

*Completar la demostración utilizando la bibliografía.*



# Autómata Finito No Determinístico- $\epsilon$

## Idea Intuitiva

- Es posible que en un AFND existan transiciones espontáneas ( $\epsilon$ ), es decir, que el autómata se mueva de un estado a otro sin necesidad de que sea vía un símbolo de la cadena de entrada. Se utilizan para ello las transiciones  $\epsilon$ .
- Esta capacidad no agrega potencia, es decir que no expande la clase de los lenguajes aceptados por autómatas finitos. Sin embargo, veremos cómo los AFND- $\epsilon$  son de gran utilidad en la prueba de la equivalencia entre las clases de lenguajes aceptados por autómatas finitos y las denotadas por expresiones regulares.

# Autómata Finito No Determinístico- $\epsilon$

## Formalización

Un AFND- $\epsilon$  se define de igual forma que un AFND, solo que es necesario incluir información sobre las transiciones  $\epsilon$ .

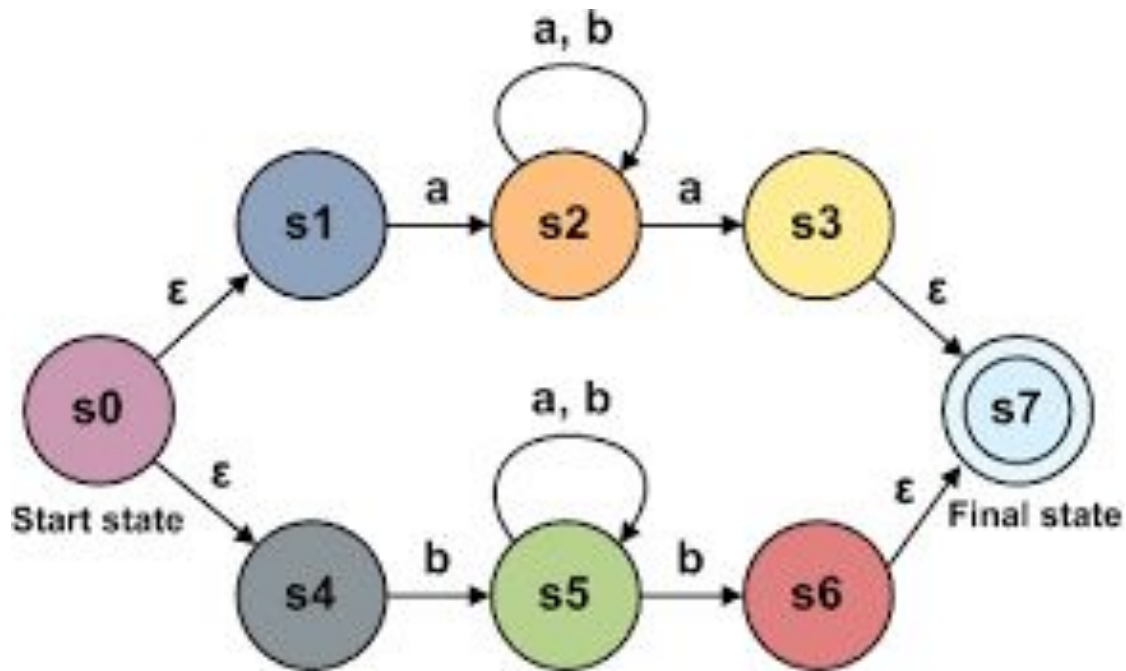
Formalmente un AFND- $\epsilon$   $A$  es  $A = (Q, \Sigma, \delta, q_0, F)$  con las componentes significando lo mismo que con AFND, excepto que  $\delta$  es ahora una función que toma como argumentos un estado de  $Q$  y un miembro de  $\Sigma \cup \{\epsilon\}$ , es decir,

$$\delta : Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q.$$

Nota: también se puede utilizar  $\lambda$  para denotar las transiciones espontáneas.

# Autómata Finito No Determinístico- $\epsilon$

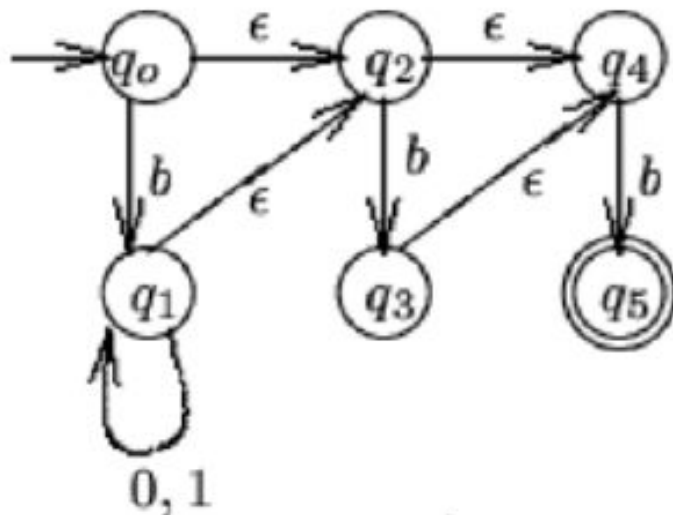
## Ejemplos



# Autómata Finito No Determinístico- $\epsilon$

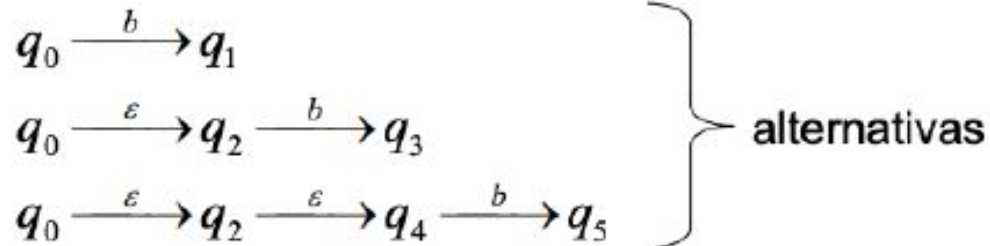
## Ejemplos

Analicemos cómo trabaja un AFND- $\epsilon$ , se muestra el diagrama de transición, la tabla de transición y una ejecución:



$\delta$	0	1	b	$\epsilon$
$\rightarrow q_0$	$\emptyset$	$\emptyset$	$\{q_1\}$	$\{q_2\}$
$q_1$	$\{q_1\}$	$\{q_1\}$	$\emptyset$	$\{q_2\}$
$q_2$	$\emptyset$	$\emptyset$	$\{q_3\}$	$\{q_4\}$
$q_3$	$\emptyset$	$\emptyset$	$\{q_3\}$	$\{q_4\}$
$q_4$	$\emptyset$	$\emptyset$	$\{q_5\}$	$\{\}$
F: $q_5$	$\emptyset$	$\emptyset$	$\{\}$	$\{\}$

- Si  $w=b$ ,
- $q_0$

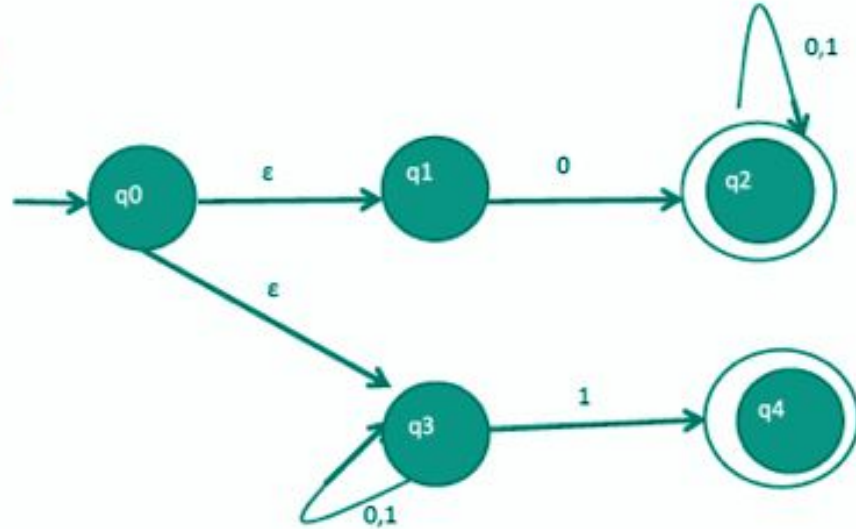


# Autómata Finito No Determinístico- $\epsilon$

## Ejercicios

**Ejercicio:** Construya un AFND- $\epsilon$  para el lenguaje en  $\Sigma = \{0,1\}$  cuyas cadenas empiezan con "0" o terminan en "1".

**SOLUCIÓN**



# Autómata Finito No Determinístico- $\epsilon$

## Clausura- $\epsilon$

- ✓ Es necesario extender la función de transición para poder definir el lenguaje aceptado por un AFND- $\epsilon$ , para ello se requiere definir **clausura- $\epsilon$** .
- ✓ Informalmente la **clausura- $\epsilon$**  de  **$q$** , es el conjunto de todos los estados  **$p$** , tal que hay un paso desde  **$q$**  a  **$p$**  rotulado  $\epsilon$ .

# Autómata Finito No Determinístico- $\epsilon$

## Clausura- $\epsilon$

Se define recursivamente *clausura* -  $\epsilon$ , de la siguiente manera:

- 1 El estado  $q$  está en la *clausura* -  $\epsilon(q)$ .
- 2 Si el estado  $p$  está en la *clausura* -  $\epsilon(q)$ , y hay una transición desde el estado  $p$  al estado  $r$  rotulado  $\epsilon$ , entonces  $r$  está en la *clausura* -  $\epsilon(q)$ .

Más precisamente, si  $\delta$  es la función de transición del AFND- $\epsilon$ , y  $p$  está en la *clausura* -  $\epsilon(q)$ , entonces la *clausura* -  $\epsilon(q)$  también contiene todos los estados en  $\delta(p, \epsilon)$ .

# Autómata Finito No Determinístico- $\epsilon$

## Lenguaje aceptado

Función de transición extendida  $\hat{\delta}$ :

- 1  $\hat{\delta}(q, \epsilon) = \text{clausura} - \epsilon(q)$
- 2 Suponemos que  $w$  es de la forma  $xa$ , donde  $a$  es el último símbolo de  $w$ , con  $a \in \Sigma$ , el cual no puede ser  $\epsilon$  dado que no pertenece a  $\Sigma$ . Entonces computamos  $\hat{\delta}(q, w)$  según los siguientes 3 pasos:
  - (I) Sea  $\hat{\delta}(q, x) = \{p_1, p_2, \dots, p_k\}$ .
  - (II) Sea  $\cup_{i=1}^k \delta(p_i, a) = \{r_1, r_2, \dots, r_m\}$
  - (III) Entonces  $\hat{\delta}(q, w) = \cup_{j=1}^m \text{clausura} - \epsilon(r_j)$ .

Sea  $E = (Q, \Sigma, \delta, q_0, F)$  un AFND- $\epsilon$ , se define lenguaje aceptado por  $E$  como:

$$L(E) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$



## Equivalencia entre AFND- $\epsilon$ y AFD

Dado cualquier AFND- $\epsilon$   $E$ , podemos encontrar un AFD  $D$  que acepte el mismo lenguaje que  $E$ .

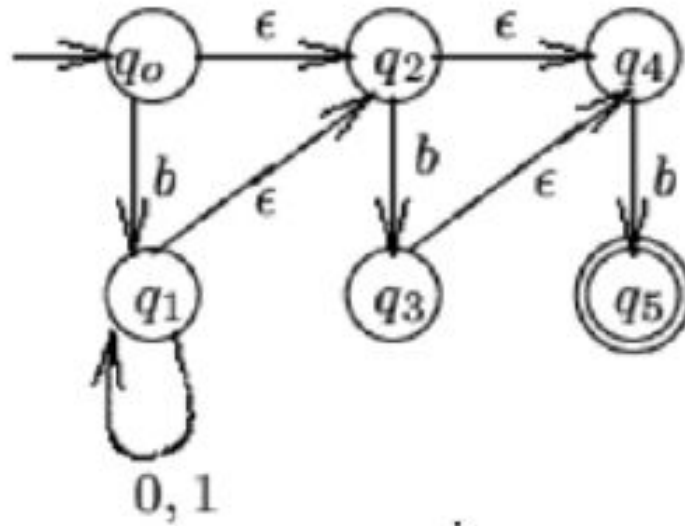
La construcción es muy similar a la de la construcción del subconjunto, sólo que debemos incorporar las transiciones  $\epsilon$  de  $E$ , para lo cual debemos utilizar la *clausura*  $- \epsilon$ :

Sea  $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ . Entonces el AFD equivalente  $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$  se define como sigue:

# Equivalencia entre AFND- $\epsilon$ y AFD

- 1  $Q_D$  es el conjunto de subconjuntos de  $Q_E$ .
- 2  $q_D = \text{clausura} - \epsilon(q_0)$ .
- 3  $F_D$  está formado por aquellos conjuntos de estados que contienen al menos un estado de aceptación de  $E$ . Es decir,  $F_D = \{S \mid S \text{ está en } Q_D \text{ y } S \cap F_E \neq \emptyset\}$ .
- 4  $\delta_D(S, a)$ , para todo  $a \in \Sigma$  y los conjuntos  $S$  en  $Q_D$ , es computada de la siguiente manera:
  - (I) Sea  $S = \{p_1, p_2, \dots, p_k\}$ .
  - (II) Obtener  $\cup_{i=1}^k \delta_E(p_i, a)$ , sea este conjunto  $\{r_1, r_2, \dots, r_m\}$ .
  - (III) Entonces  $\delta_D(S, a) = \cup_{j=1}^m \text{clausura} - \epsilon(r_j)$ .

## Ejemplo de aplicación



## Ejemplo (Cont.)

$$\text{clausura} - \epsilon(q_0) = \{q_0, q_2, q_4\}$$

*Ahora empezamos a calcular la función de transición, a partir del estado inicial del autómata que vamos a construir:*

$$\delta(\{q_0, q_2, q_4\}, b) = \{q_1\} \cup \{q_3\} \cup \{q_5\} = \{q_1, q_3, q_5\}$$

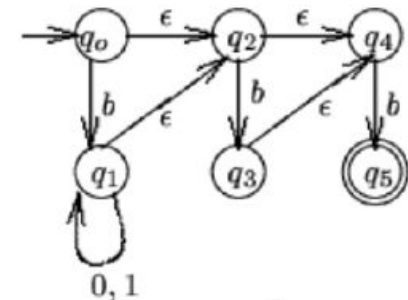
$$\text{clausura} - \epsilon(\{q_1, q_3, q_5\}) = \{q_1, q_2, q_4\} \cup \{q_3, q_4\} \cup \{q_5\} = \{q_1, q_2, q_3, q_4, q_5\}$$

$$\delta(\{q_0, q_2, q_4\}, 0) = \emptyset \cup \emptyset \cup \emptyset = \emptyset$$

$$\text{clausura} - \epsilon(\emptyset) = \emptyset$$

$$\delta(\{q_0, q_2, q_4\}, 1) = \emptyset \cup \emptyset \cup \emptyset = \emptyset$$

$$\text{clausura} - \epsilon(\emptyset) = \emptyset$$



## Ejemplo (Cont.)

$$\delta(\{q_1, q_2, q_3, q_4, q_5\}, b) = \emptyset \cup \{q_3\} \cup \emptyset \cup \{q_5\} \cup \emptyset = \{q_3, q_5\}$$

$$\text{clausura} - \epsilon(\{q_3, q_5\}) = \{\mathbf{q_3, q_4, q_5}\}$$

$$\delta(\{q_1, q_2, q_3, q_4, q_5\}, 0) = \{q_1\} \cup \emptyset \cup \emptyset \cup \emptyset \cup \emptyset = \{q_1\}$$

$$\text{clausura} - \epsilon(\{q_1\}) = \{\mathbf{q_1, q_2, q_4}\}$$

$$\delta(\{q_1, q_2, q_3, q_4, q_5\}, 1) = \{q_1\} \cup \emptyset \cup \emptyset \cup \emptyset \cup \emptyset = \{q_1\}$$

$$\text{clausura} - \epsilon(\{q_1\}) = \{\mathbf{q_1, q_2, q_4}\}$$

$$\delta(\{q_3, q_4, q_5\}, b) = \{q_5\}$$

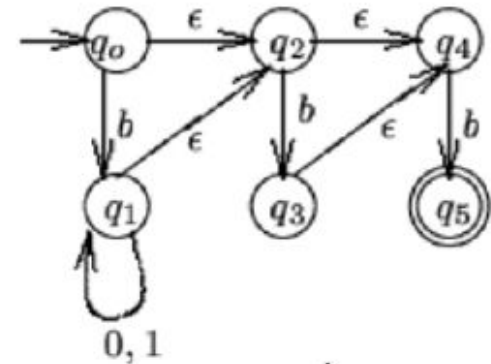
$$\text{clausura} - \epsilon(\{q_5\}) = \{\mathbf{q_5}\}$$

$$\delta(\{q_3, q_4, q_5\}, 0) = \emptyset$$

$$\text{clausura} - \epsilon(\emptyset) = \emptyset$$

$$\delta(\{q_3, q_4, q_5\}, 1) = \emptyset$$

$$\text{clausura} - \epsilon(\emptyset) = \emptyset$$



## Ejemplo (Cont.)

$$\delta(\{q_1, q_2, q_4\}, b) = \{q_3, q_5\}$$

$$\text{clausura} - \epsilon(\{q_3, q_5\}) = \{\mathbf{q_3}, \mathbf{q_4}, \mathbf{q_5}\}$$

$$\delta(\{q_1, q_2, q_4\}, 0) = \{q_1\}$$

$$\text{clausura} - \epsilon(\{q_1\}) = \{\mathbf{q_1}, \mathbf{q_2}, \mathbf{q_4}\}$$

$$\delta(\{q_1, q_2, q_4\}, 1) = \{q_1\}$$

$$\text{clausura} - \epsilon(\{q_1\}) = \{\mathbf{q_1}, \mathbf{q_2}, \mathbf{q_4}\}$$

$$\delta(\{q_5\}, b) = \emptyset$$

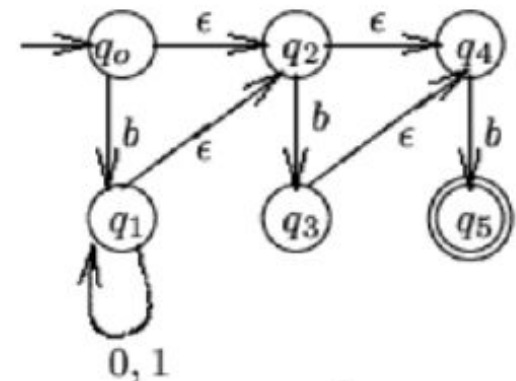
$$\text{clausura} - \epsilon(\emptyset) = \emptyset$$

$$\delta(\{q_5\}, 0) = \emptyset$$

$$\text{clausura} - \epsilon(\emptyset) = \emptyset$$

$$\delta(\{q_5\}, 1) = \emptyset$$

$$\text{clausura} - \epsilon(\emptyset) = \emptyset$$



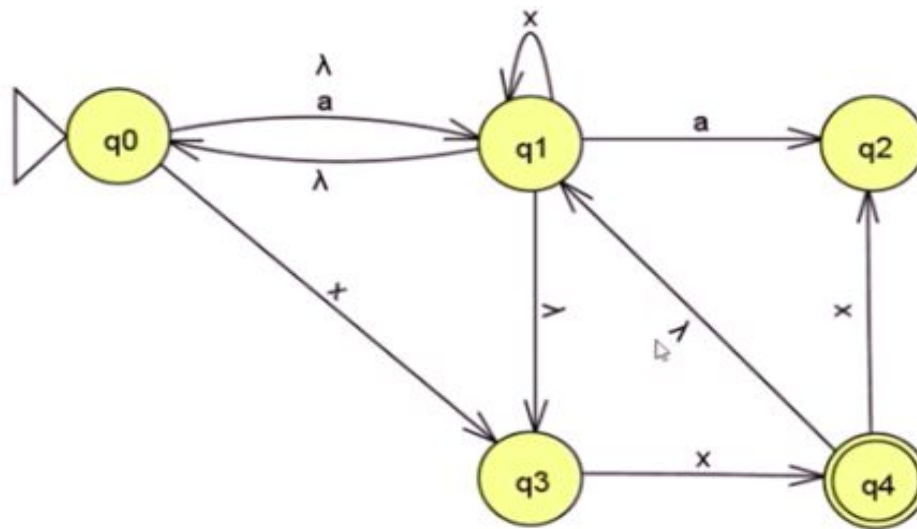
## Ejemplo (Cont.)

- ✓ *Se deja como ejercicio construir el diagrama de transición y determinar cuáles son los estados finales.*
- ✓ *Recordar que el conjunto vacío, representa un estado del AFD que construimos.*



# Ejercicio

Aplicar el algoritmo para transformar el siguiente AFND- $\epsilon$  a un AFD:





# Teorema

Un lenguaje  $L$  es aceptado por algún AFND- $\epsilon$  sí y sólo sí  $L$  es aceptado por algún AFD.

## *Demostración:*

Ver bibliografía, página 79, Introduction to Automata Theory, Languages and Computation - Hopcroft - Ullman - Motwani.

¿DUDAS?

