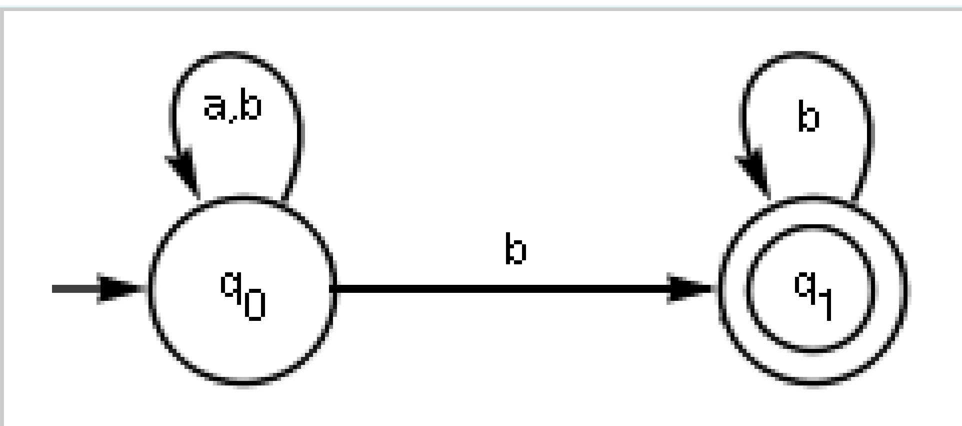
AUTÓMATAS FINITOS NO DETERMINÍSTICOS (AFND):

- es aquel que posee al menos un estado $q \in Q$, tal que para un símbolo a del alfabeto ($a \in \Sigma$), existe más de una transición $\delta(q,a)$ posible o ninguna.

- Ejemplo:

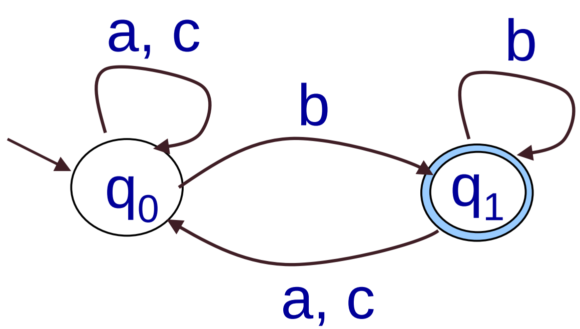


σ	a	b
q_0	q_0	$\{q_0,q_1\}$
q_1	-	q_1

Autómatas Finitos

Determinísticos

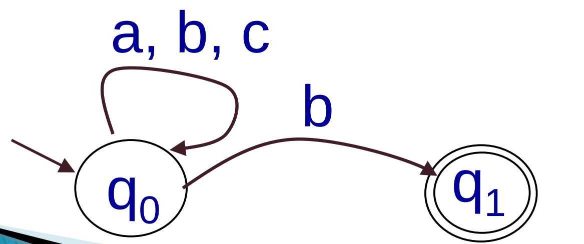
Para cada estado y para cada símbolo se determina unívocamente un solo estado:



Dada una cadena w existe sólo una forma de recorrer el diagrama de transición de estados del AF

No Determinísticos

Para algunos estados, dado un símbolo a $\in$ Σ, existe un conjunto de estados siguientes para elegir



Desde $\delta(q_0, b)$ se puede permanecer en q_0 ó pasar a q_1

- Un AFND es una quíntupla:

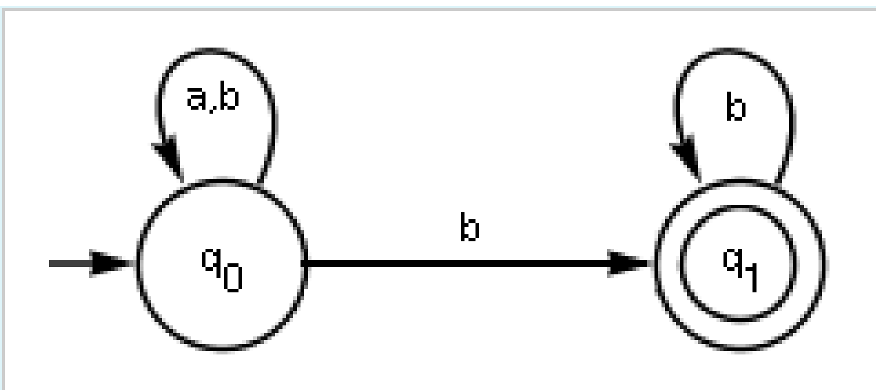
$$M=(\Sigma, Q, \delta, q0, F)$$

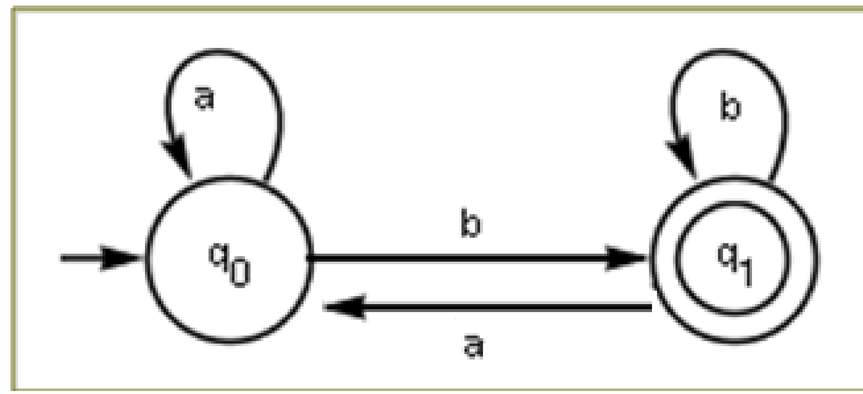
Donde:

- Σ es un alfabeto.
- Q es un conjunto finito no vacío de estados.
- δ es :
  - o una **relación**, es decir, $\delta \subseteq (Q \times \Sigma) \times Q$
  - o una **función**, es decir, $\delta: (Q \times \Sigma) \times \mathcal{P}(Q)$
- q0 es el estado inicial.
- F es el conjunto de estados finales.

AFD y AFND

- Los autómatas finitos deterministas son un caso particular de los no deterministas, puesto que *Q* pertenece al conjunto *P(Q)*.





σ_N	a	b
q_0	q_0	$\{q_0,q_1\}$
q_1	-	q_1

σ_D	a	b
q_0	q_0	q_1
q_1	q_0	q_1

Autómata finito no determinista

conjunto de estados finales $F \subseteq Q$

estado inicial, $q_0 \in Q$

$$A = (Q, \Sigma, \delta, q_0, F)$$

función de transición

δ: $Q \times \Sigma \rightarrow$ P(Q)

conjunto de estados

alfabeto

Extensión de δ a cadenas (δ: $Q \times \Sigma^* \rightarrow$ P(Q))

$$\forall q \in Q, x \in \Sigma^*, a \in \Sigma$$

$$\begin{cases} \delta(q, \varepsilon) = \{q\} \\ \delta(q, xa) = \bigcup\limits_{p \in \delta(q,x)} \delta(p, a) \end{cases}$$

AFD-N

- La interpretación que se suele hacer en el procesamiento de un AFND es que el autómata puede estar en varios estados a la vez, generándose una ramificación de las *configuraciones* existentes en un momento dado.

- En un autómata finito no determinista se puede aceptar la existencia de más de un nodo inicial, relajando aún más la definición original.

Ejercicio:

δ	0	1
$\rightarrow q_0$	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	ϕ	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_4\}$	ϕ
q_4	$\{q_4\}$	$\{q_4\}$

Se pide:

1. Dibujar en AFND asociado a la función de transición presentada.
2. Describir formalmente el lenguaje reconocido por el AFND.

AFND

Un AFN-D acepta una cadena si existe un camino que la acepta.

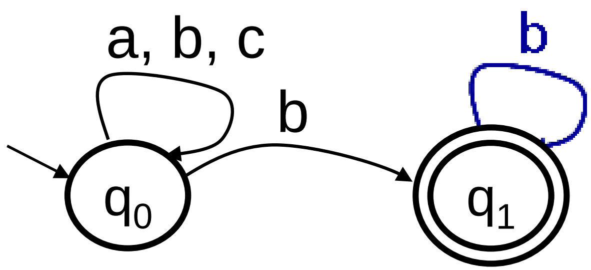
Son *más fáciles de especificar y claros de entender*….

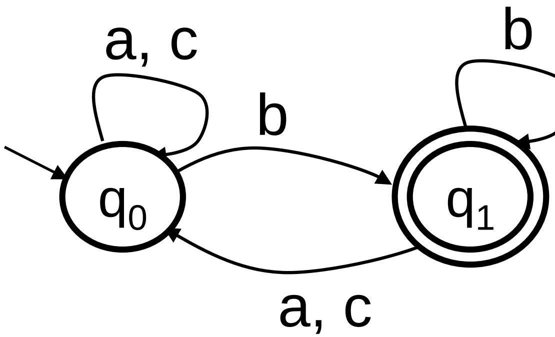
Pero, los computadores son deterministas !!!

Equivalencia entre AFND y AFD

Sea AFND = <{q_0, q_1}, {a, b, c}, δ, q_0, {q_1}> tal que

L(AFND) = { w / w $\in$ {a, b, c}* y w termina en b}





δ_{ND}	a	b	c
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	-	q1	-

Función de transición no determinística

δ_D	a	b	c
$[q_0]$	$[q_0]$	$[q_1]$	$[q_0]$
$[q_1]$	$[q_0]$	$[q_1]$	$[q_0]$

Función de transición determinística

Equivalencias entre autómatas finitos

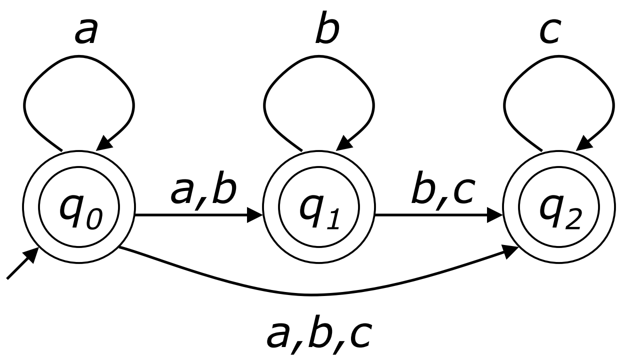
- Se dice que dos autómatas finitos son **equivalentes**, si ambos reconocen el mismo lenguaje regular.

- Toda **expresión regular** (que define un lenguaje regular) puede ser expresada como un **AFND** y viceversa.

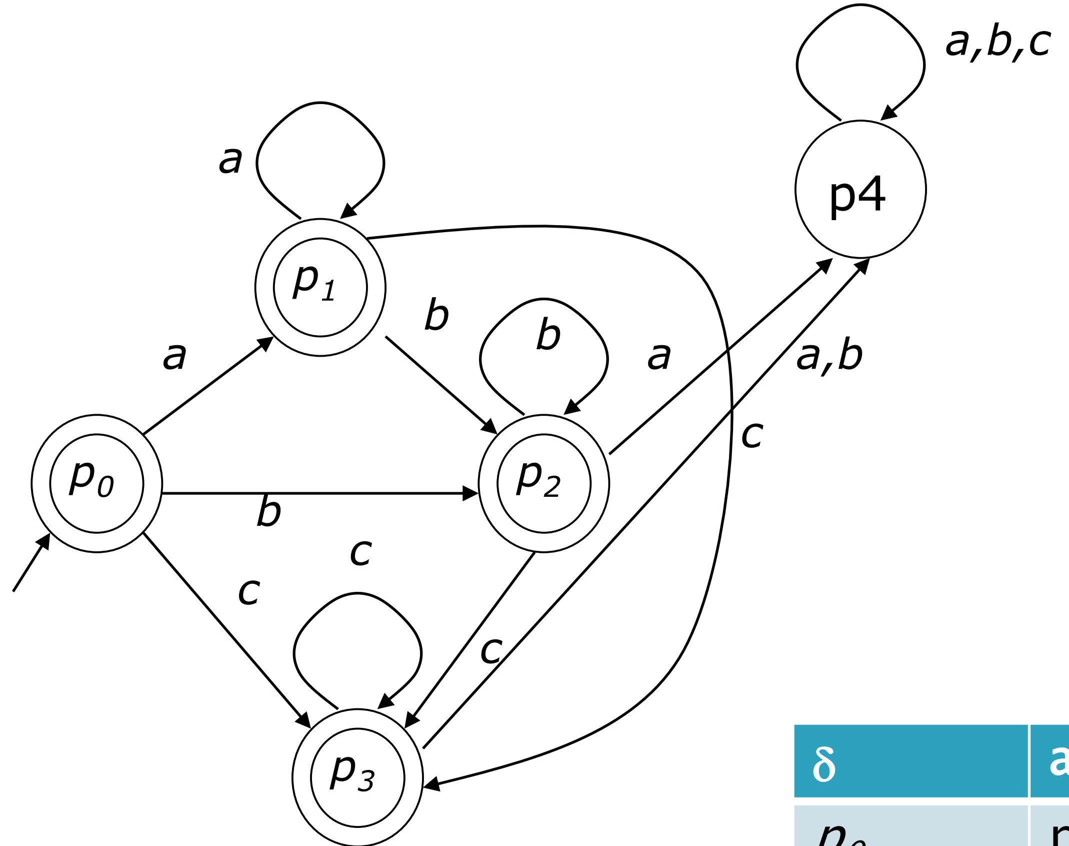
Equivalencia AFN - AFD

Dado un lenguaje L aceptado por un AFN, existe un AFD que acepta L.

Paso de AFN a AFD



δ	a	b	c
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	–	$\{q_1, q_2\}$	$\{q_2\}$
q_2	–	–	$\{q_2\}$



δ	a	b	c
p_0	p1	p2	p3
p_1	p1	p2	p3
p_2	p4	p2	p3
p_3	p4	p4	p3
p_4	p4	p4	p4

Autómatas Finitos No-Deterministas

- Para cada AFN-D existe un AFD equivalente, por lo tanto ambos aceptan los mismos lenguajes. Existen teoremas que demuestran este enunciado.

- Existen algoritmos para la conversión de un AFND en un AFD.

Por otro lado,...

- Un AFD reconoce palabras pertenecientes a un lenguaje L determinado.

- Las palabras pertenecientes a un lenguaje L determinado pueden ser reconocidas por uno o más AFD's.

Autómata finito no determinista con transiciones vacías

conjunto de estados finales $F \subseteq Q$

estado inicial, $q_0 \in Q$

$$\boxed{M = (Q, \Sigma, \delta, q_0, F)}$$

función de transición
$\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \wp(Q)$

alfabeto

conjunto de estados

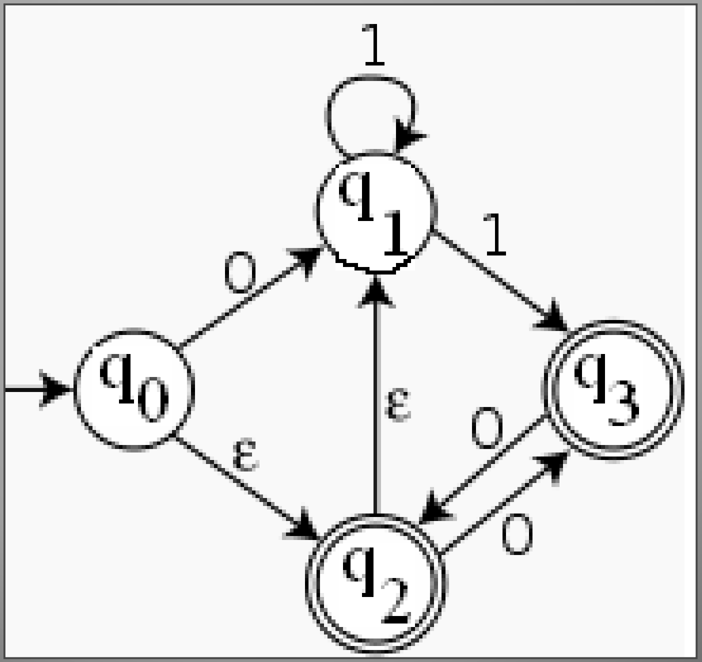
- ε -clausura de un estado, transición vacía
- Lenguaje aceptado: $\boldsymbol{L(M) = \{w \in \Sigma^*: \delta(q_0, w) \cap F \neq \emptyset\}}$

ε−AFND:

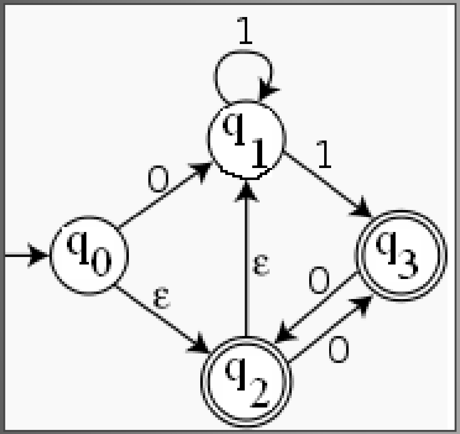
- Puede darse:
  - Que existan transiciones del tipo $\delta(q,a)=q1$ y $\delta(q,a)=q2$, siendo $q1 \neq q2$;
  - Que existan transiciones del tipo $\delta(q, \varepsilon)$, siendo q un estado no-final, o bien un estado final pero con transiciones hacia otros estados.
  - Cuando se cumple el segundo caso, se dice que el autómata es un **AFD-N con transiciones vacías** o **transiciones ε** (abreviado **ε-AFND**). Estas transiciones permiten al autómata cambiar de estado sin procesar ningún símbolo de entrada.

Ejercicio:

Describa formalmente el AFDN-**ε** descrito a continuación (Describa la función de transición σ):



Describa formalmente el AFDN-**ε** descrito a continuación (Descripción de la función de transición σ):



δ	0	1	ε
q0	{q1}	–	{q2}
q1	–	{q1, q3}	
q2	{q3}	–	{q1}
q3	{q2}	–	–

- Mientras en un AFD la función de transición se define como:

$$\delta : Q \times \Sigma \rightarrow Q$$

- en un AFND se define como:

$$\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$$

- Para los AFND-ϵ, se expresa la función de transición de la forma:

$$\delta : Q \times \{\Sigma \cup \epsilon\} \rightarrow \mathcal{P}(Q)$$

- donde *P(Q)* es el conjunto potencia de *Q*.

Equivalencias entre autómatas finitos

- Por transitividad, para cualquier autómata finito no determinista siempre existe un autómata finito determinista equivalente, y viceversa.

- En el diseño de autómatas finitos hay quienes sugieren, construir un AFND-ϵ para el problema en primer lugar, porque es el más sencillo, por poseer menos restricciones en su función de transiciones.

Equivalencias entre autómatas finitos

- En el diseño de autómatas finitos hay quienes sugieren, construir un AFND-ϵ para el problema en primer lugar, porque es el más sencillo, por poseer menos restricciones en su función de transiciones.

- Luego dicho autómata se reduce a un AFND, y finalmente a un AFD, el cual por sus características deterministas ya puede ser implementado sin problemas utilizando un lenguaje de programación.

Equivalencia AFλ - AFD

-Dado $q \in Q$, **ε-*clausura*(*q*)** es el conjunto de estados a los que se puede llegar desde q sin consumir símbolos.

-Dado $P \subseteq Q$ $\quad \varepsilon\text{-}clausura(P) = \bigcup_{p \in P} \varepsilon\text{-}clausura(p)$

Extensión a cadenas de la función de transición

$$-\delta'(q, \varepsilon) = \varepsilon\text{-}clausura(q)$$

$$-\delta'(q, xa) = \varepsilon\text{-}clausura\left(\bigcup_{p \in \delta'(q, x)} \delta(p, a)\right)$$

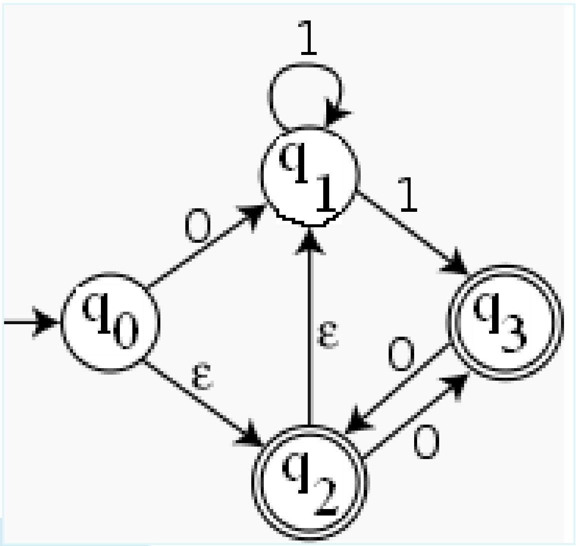
Equivalencia AFλ - AFD

Dado $A = (Q, \Sigma, \delta, q_0, F)$ *AFND* - ε , se define:

$$M' = (Q, \Sigma, \delta', q_0, F')$$

con $F' = F \cup \{q_0\}$ si ε-*clausura*$(q_0) \cap F \neq \emptyset$ o sólo F en caso contrario.

Se cumple que $L(A) = L(A')$



Ejemplo de AFND-ε

Sea M = $(\{q_0, q_1, q_2, q_3), \{0, 1\}, \delta, q_0, \{q_0\})$
con:

Tabla de transiciones

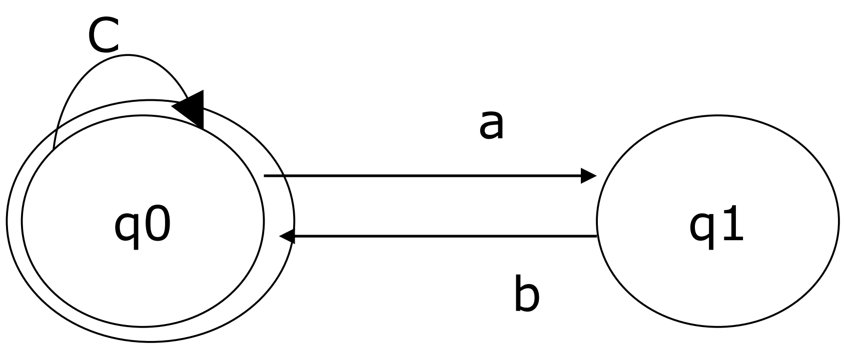
	0	1	ε
q_0	–	–	$\{q_1\}$
q_1	–	$\{q_3\}$	$\{q_2\}$
q_2	$\{q_1\}$	$\{q_2\}$	–
q_3	–	$\{q_3\}$	$\{q_0\}$

Se pide:
Dibujar el Diagrama de transiciones

Ejercicio:

Considere el AFND presentado a continuación, se le pide:

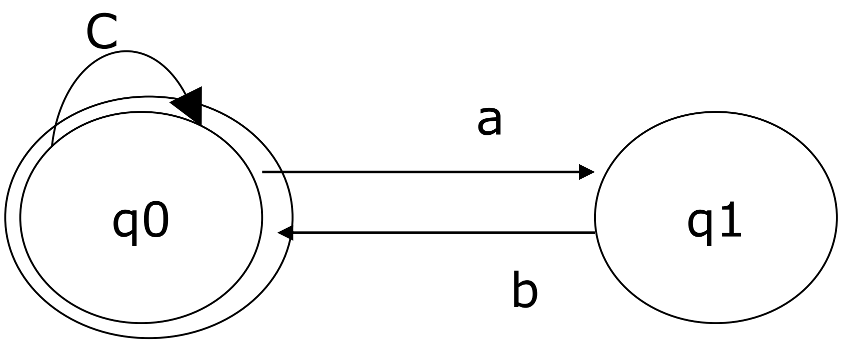
1. Identificar el lenguaje L(M) reconocido por el AFND
2. Transformarlo en un AFD .



σ	a	b	c
q0	q1	-	q0
q1	-	q0	-

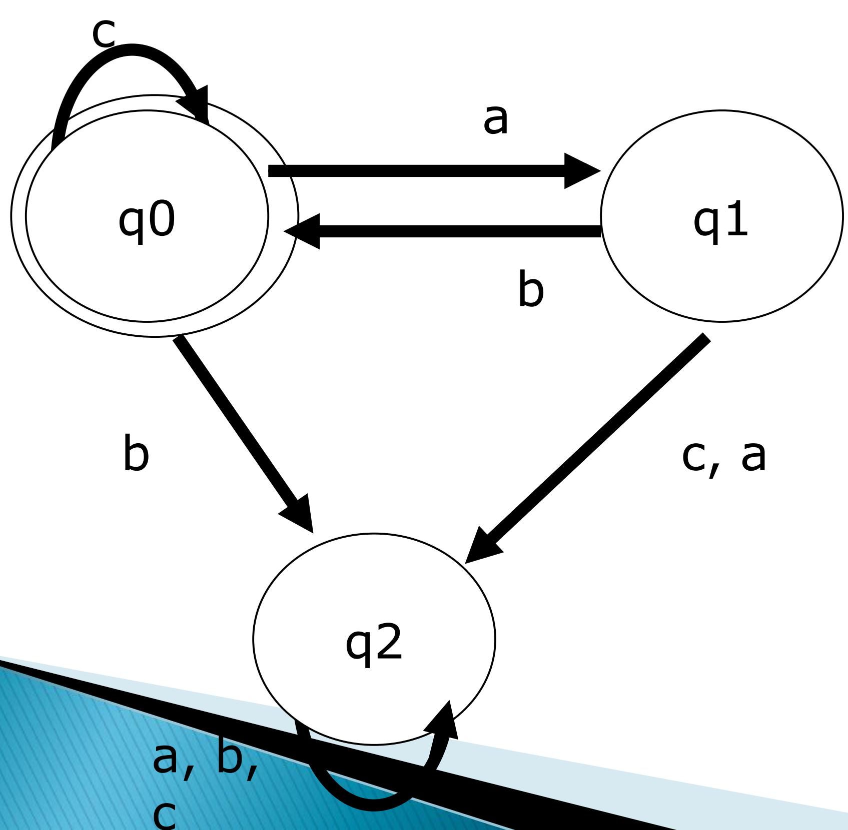
Función de transición asociada

L(M) = $\{\boldsymbol{w} \in \Sigma^* \,/\, \mathbf{(xc)(ab)(yc))}^{m} \,,\, \mathbf{x>=0,\ y>=0,\ m>=0}\}$



σ	a	b	c
q0	q1	-	q0
q1	-	q0	-

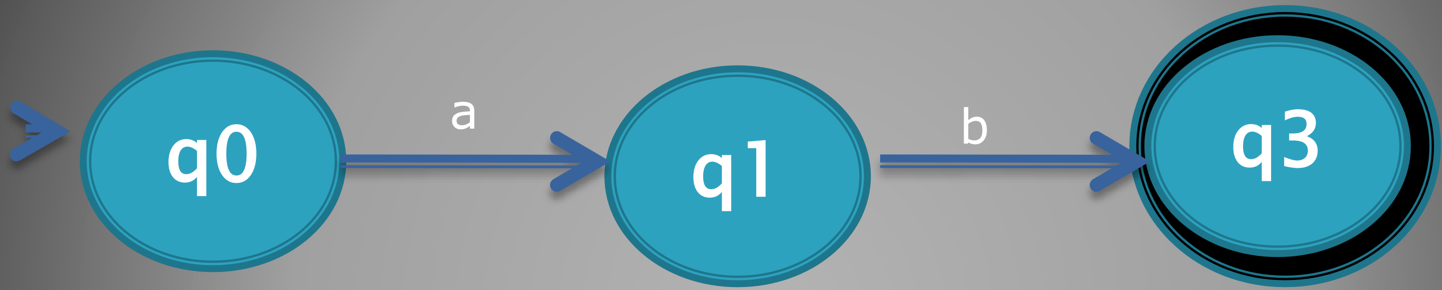
Función de transición asociada



σ	a	b	c
q0	q1	q2	q0
q1	q2	q0	q2
q2	q2	q2	q2

Función de transición asociada

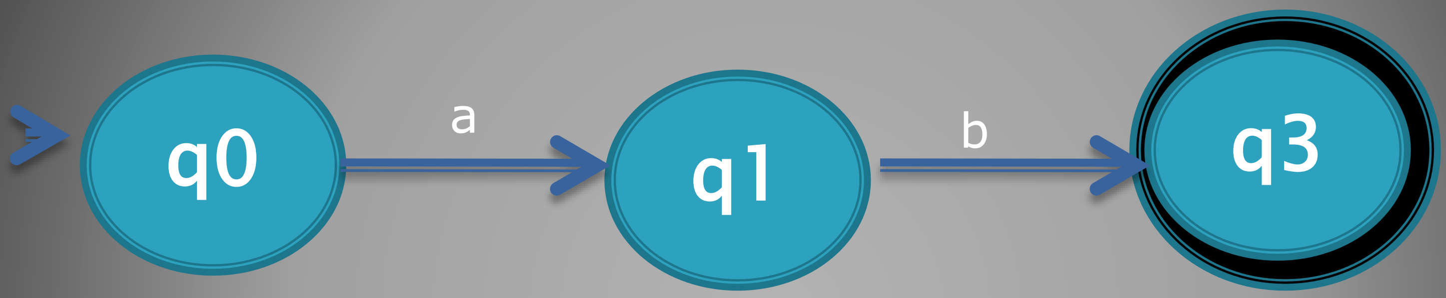
Ejemplo:



AFND

1. Transforme el AFND a un AFD.
2. Defina el lenguaje que reconoce formalmente.

Ejemplo:

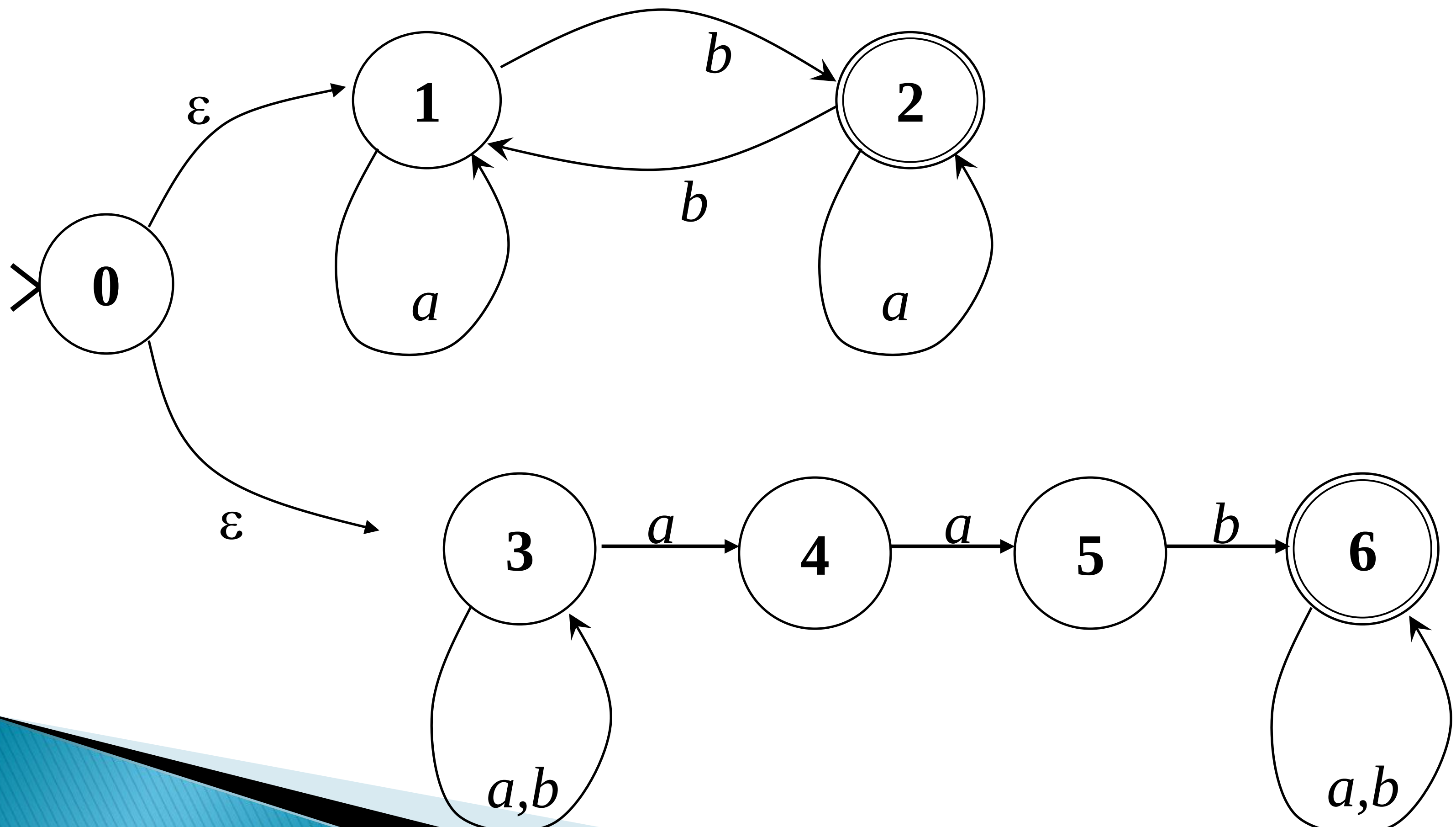


AFND

1. Transforme el AFND a un AFD.
2. Defina el lenguaje que reconoce formalmente.

Ejemplo

- Verifique que el AFND sobre $\Sigma=\{a, b\}$ reconoce palabras con un número impar de *b*'s o que contienen la cadena *aab*.



Ejemplo de AFND

Sea $M = (\{q_0 , q_1 , q_2), \{a, b, c\}, \delta , q_0 , \{q_0, q_1, q_2\})$
con:

$\delta(q_0 ,a) = \{q_0, q_1, q_2\}$, $\delta(q_0 , b) = \{q_1, q_2\}$, $\delta(q_0 ,c) = \{q_2\}$,
$\delta(q_1 ,a) = \emptyset$, $\delta(q_1, b) = \{q_1, q_2\}$, $\delta(q_1,c) = \{q_2 \}$,
$\delta(q_2 ,a) = \emptyset$, $\delta(q_2, b) = \emptyset$, $\delta(q_2,c) = \{q_2 \}$.

Se pide:

a) Construir la relación de transición δ en su forma matricial.
b) Dibujar el AFND asociado.
c) Describir formalmente el lenguaje $L(M) = \{x \in \Sigma^*: \delta(q_0 , x) \cap F \neq \emptyset \}$ que reconoce.
d) Transformar el AFND en un AFD equivalente.

Tabla de transiciones

	a	b	c
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\varnothing$	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\varnothing$	$\varnothing$	$\{q_2\}$

Ejemplo de AFND

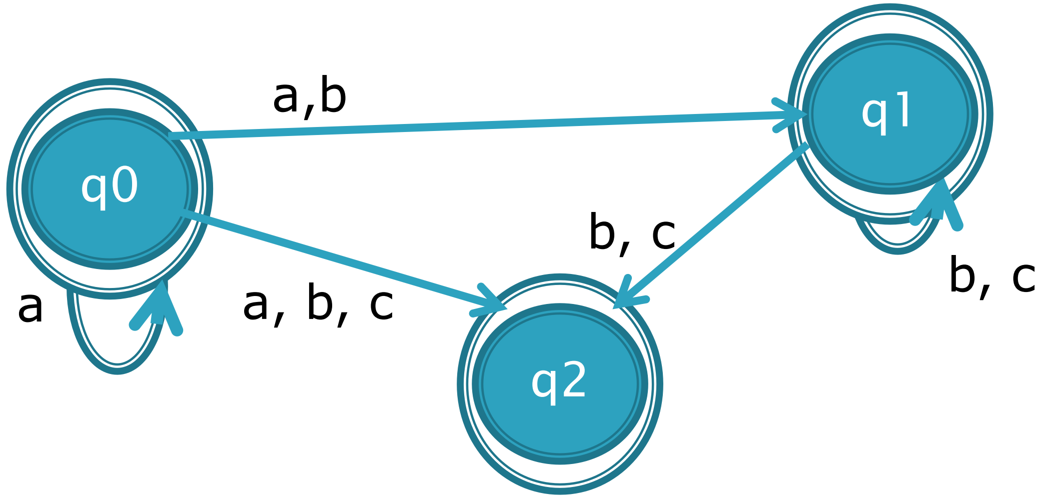


Diagrama de transiciones

Lenguaje aceptado: $L(A) = \{x \in \Sigma^*: \delta(q_0, x) \cap F \neq \varnothing \}$