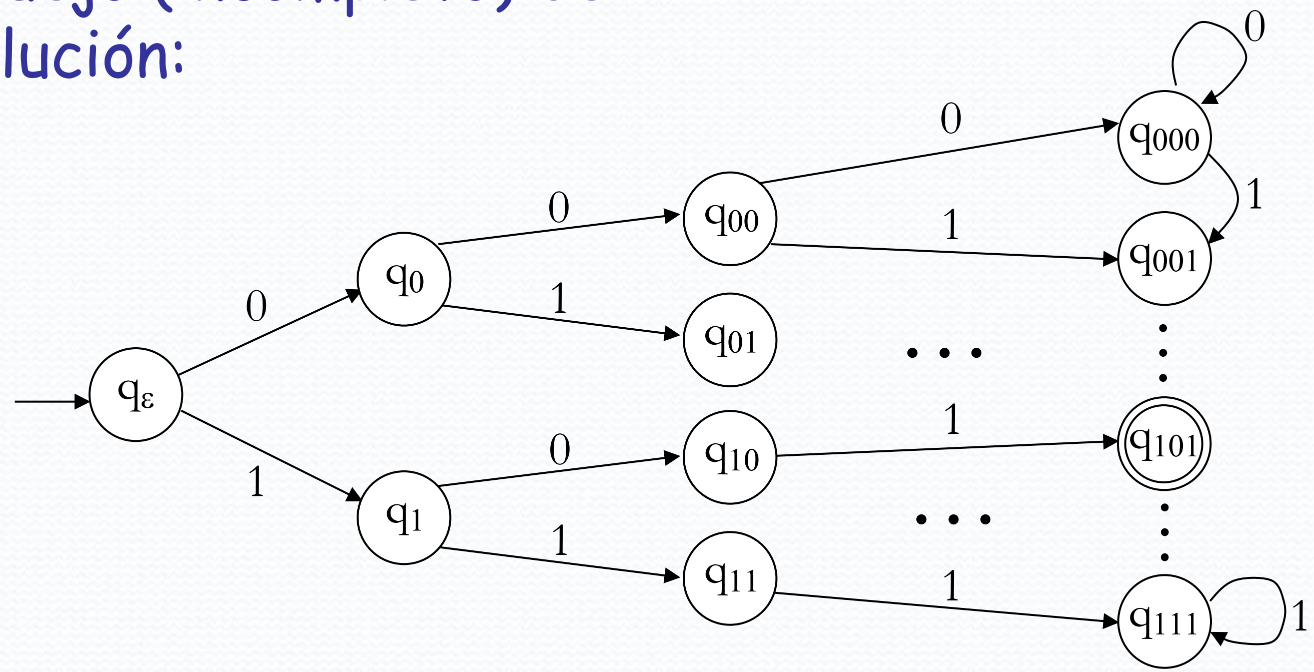
Autómatas Finitos No deterministas

Dr. José Lázaro Martínez Rodríguez

AFND: Autómatas finitos *no deterministas*

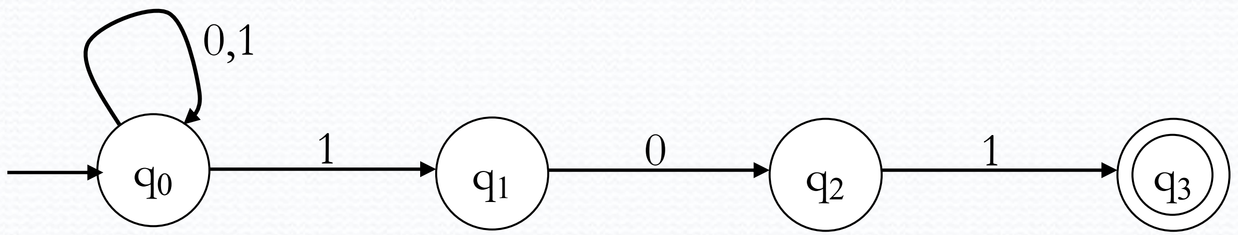
Veamos un ejemplo de AFD. Con $\Sigma=\{0,1\}$, queremos que acepte el lenguaje de los strings que terminan en 101.

Bosquejo (incompleto) de la solución:



AFND: Autómatas finitos *no deterministas*

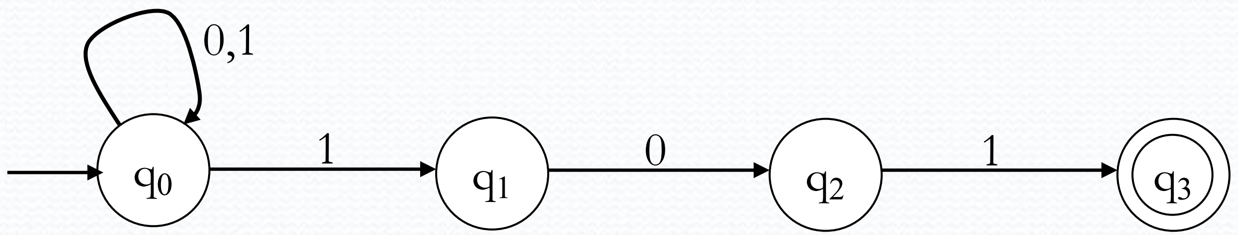
Sería más cómodo poder *adivinar* en que momento faltan sólo tres letras, y en ese momento comparar con 101.



•Si en q_0 leemos un "1", tenemos dos posibles opciones.

•Si "adivinamos" que estamos a tres letras del final, entonces escojo irme hacia q_1.

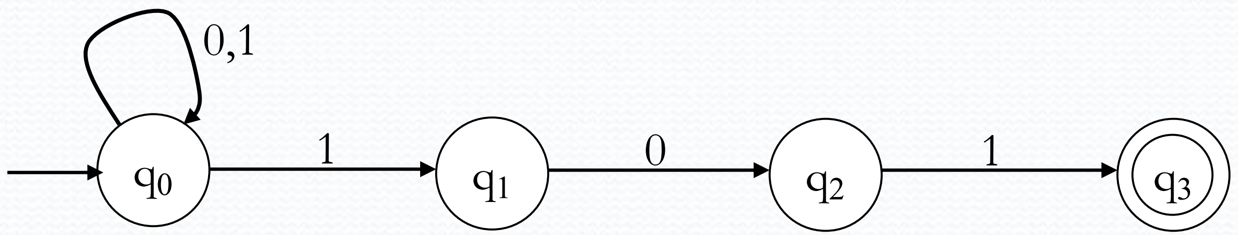
AFND: Autómatas finitos *no deterministas*



Además, hay transiciones no definidas, como $\delta(q_1,1)$.

$\rightarrow$ En ese caso el autómata "se cae".

AFND: Autómatas finitos *no deterministas*



Es más fácil verlo como generador: su lenguaje son las palabras que *podemos* obtener al recorrerlo y terminar en q_3.

AFND: Autómatas finitos *no deterministas*

Un AFND admite "bifurcaciones" en el comportamiento del autómata, y transiciones indefinidas.

Podemos interpretar el "no determinismo"

- como que existe un "oráculo" que permite adivinar el camino correcto

o bien

- como que exploramos *todas* las opciones.

- Diremos que el AFND acepta una palabra si existe *algún camino posible* que permite leer esa palabra y llegar a un estado de aceptación.

AFND: Autómatas finitos *no deterministas*

δ ya no es función

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

pues desde un mismo q, leyendo un mismo α, podemos pasar a más de un estado, o a ninguno

$\rightarrow$ ahora será una función

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

donde $\mathcal{P}(Q)$, también anotado 2^Q, son "las partes de Q" (o sea, $\delta(q,\alpha)$ entrega un subconjunto de Q).

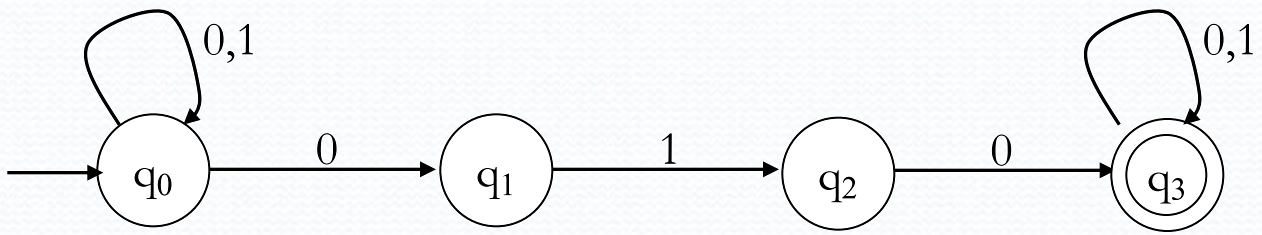
En particular, las transiciones indefinidas son $\delta(q,\alpha)=\emptyset$.

AFND: Autómatas finitos *no deterministas*

- El no determinismo significa que el siguiente movimiento no está determinado únicamente por el estado actual y el símbolo de entrada actual.
- En algunos casos, hay más de una transición posible a seguir.
- El autómata seguirá la "correcta", es decir, la que conduce a la aceptación, si es que existe.

AFND: Autómatas finitos *no deterministas*

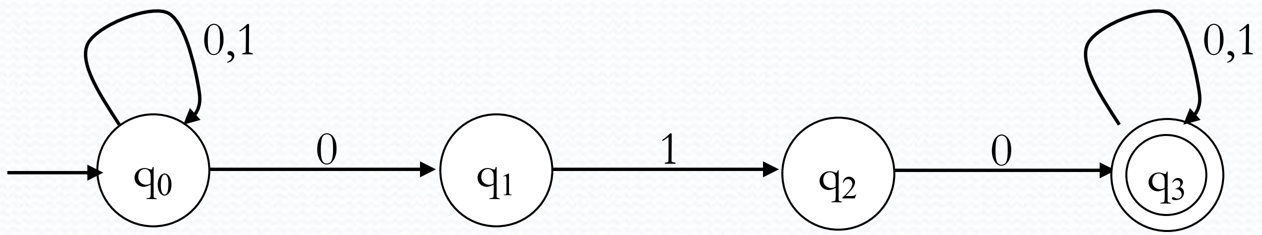
Otro ejemplo: reconocer los strings que incluyen 010 *en alguna parte.*



δ:	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
q_2	$\{q_3\}$	$\emptyset$
$*q_3$	$\{q_3\}$	$\{q_3\}$

AFND: Autómatas finitos *no deterministas*

Otro ejemplo: reconocer los strings que incluyen 010 *en alguna parte.*

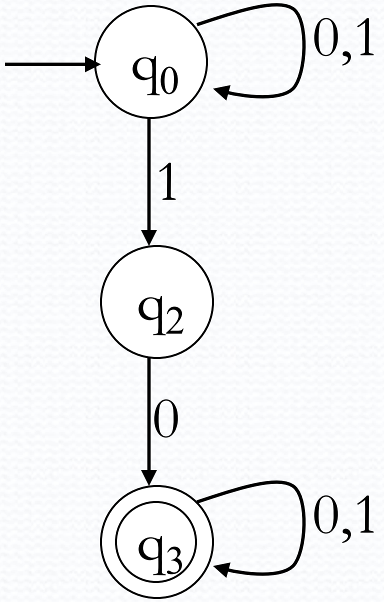


Ejercicio: hacer lo mismo con un AFD (también se puede), y hacerlo también para strings que incluyan 111, 101, 110, respectivamente. Verán que los AFND son iguales, pero los AFD son más variados (y menos obvios).

AFD, AFND

En AFD, $p = \delta(q,\alpha)$ ⟺ "del estado q, leyendo α, ***se pasa*** al estado p"

En AFND, $p \in \delta(q,\alpha)$ ⟺ "del estado q, leyendo α, se ***puede pasar*** al estado p"



Insistamos: en *ambos* casos se puede interpretar el AF como un descriptor de posibles "paseos", cada cual tiene asociada una palabra.

AFND: Autómatas finitos *no deterministas*

Nuevamente podemos extender δ para que lea más de una letra a la vez.

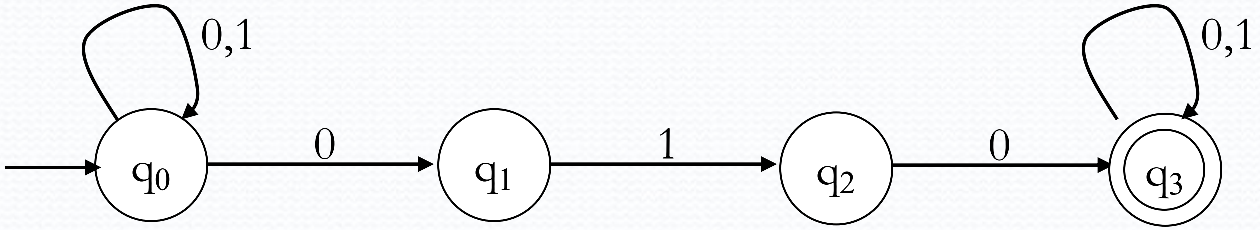
- $\delta'(q,\varepsilon)=\{q\}$
- $\delta'(q,w\alpha)= \{ p : \exists r \in \delta'(q,w) \text{ tal que } p \in \delta(r,\alpha) \} = \bigcup_{r \in \delta'(q,w)} \delta(r,\alpha)$

- $\delta'(q,w)$ es entonces el conjunto de estados a los que *puedo llegar* a partir del estado q, leyendo en el camino la palabra w desde el input.

- Nuevamente δ' resulta ser una generalización de δ, de modo que el apóstrofe no es necesario.

AFND: Autómatas finitos *no deterministas*

- $\delta'(q,\varepsilon)=\{q\}$
- $\delta'(q,w\alpha)= \{\ p : \exists r\in\delta'(q,w) \text{ tal que } p\in\delta(r,\alpha)\ \} = \displaystyle\bigcup_{r\,\in\,\delta'(q,w)} \delta(r,\alpha)$



δ:	0	1
$\to q_0$	$\{q_0,q_1\}$	$\{q_0\}$
q_1	$\varnothing$	$\{q_2\}$
q_2	$\{q_3\}$	$\varnothing$
$*q_3$	$\{q_3\}$	$\{q_3\}$

$\delta(q_0, 01) = ?$

$\{\ q_0, q_2\ \}$

AFND: Autómatas finitos *no deterministas*

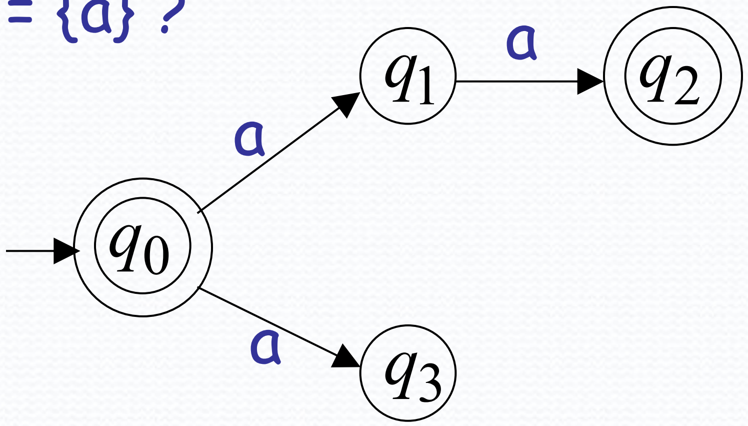
El lenguaje del AFND será

$$L(M) = \{ w : \delta(q_0, w) \cap F \neq \emptyset \}$$

O sea: "w pertenece al lenguaje, si a partir del estado inicial, y leyendo w, es posible llegar a *algún* estado que sea de aceptación".

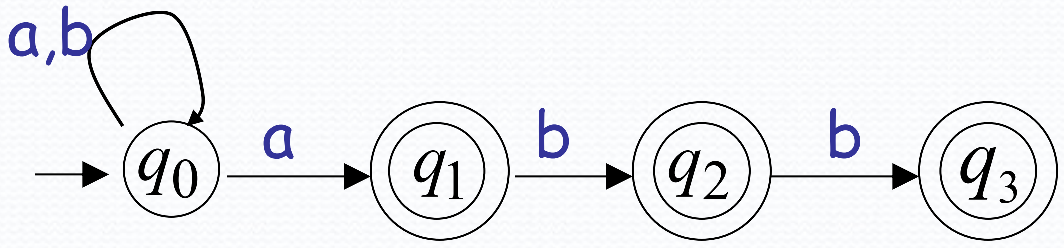
¿Qué acepta este AFND, con $\Sigma = \{a\}$?

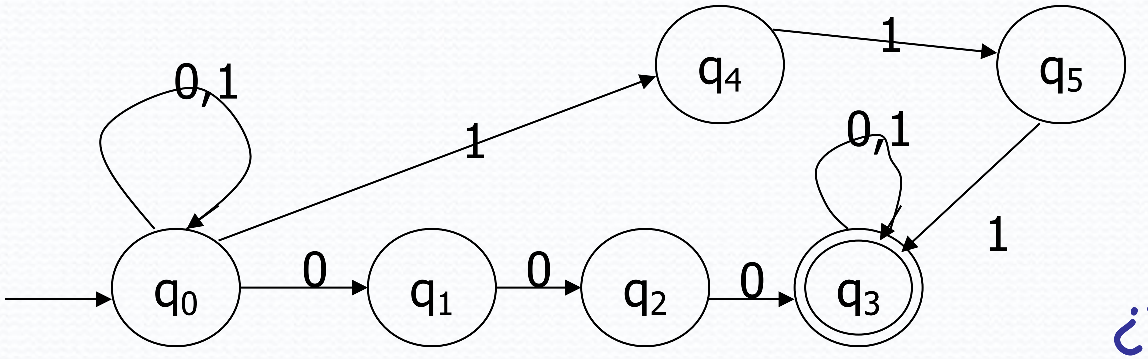
$$L = \{aa, \varepsilon\}$$



AFND: Autómatas finitos *no deterministas*

¿Y este, con
$\Sigma = \{a,b\}$?



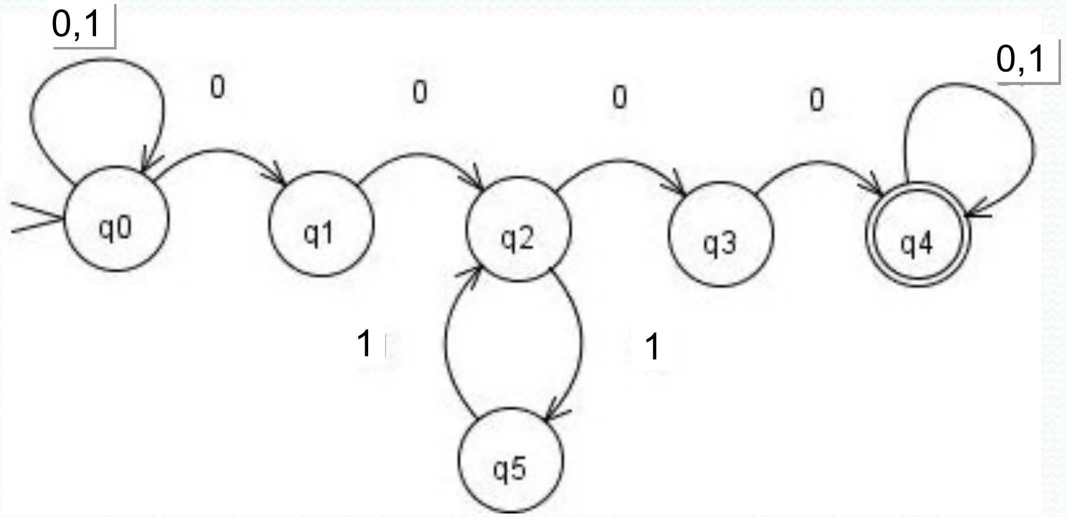


¿Y aquí, con $\Sigma = \{0,1\}$?

Más ejercicio: ver qué pasa si cambiamos a:
(1) $\delta(q_3,0) = \emptyset$, $\delta(q_3,1) = \emptyset$
(2) $\delta(q_3,0) = \emptyset$, $\delta(q_3,1) = q_5$

AFND: Autómatas finitos *no deterministas*

Construir un AFND que acepte palabras que contienen dos pares de 0's adyacentes, opcionalmente separados por una cantidad par de 1's: 01001100001 está, 1100111100 también, pero 0101001110010 no.



AFND: Autómatas finitos *no deterministas*

(dijimos que esos se llamaban "regulares")

Anotemos

- LR = { L: L es reconocido por algún AFD}
- LN = {L: L es reconocido por algún AFND}

¿Qué relación existirá entre estas clases?

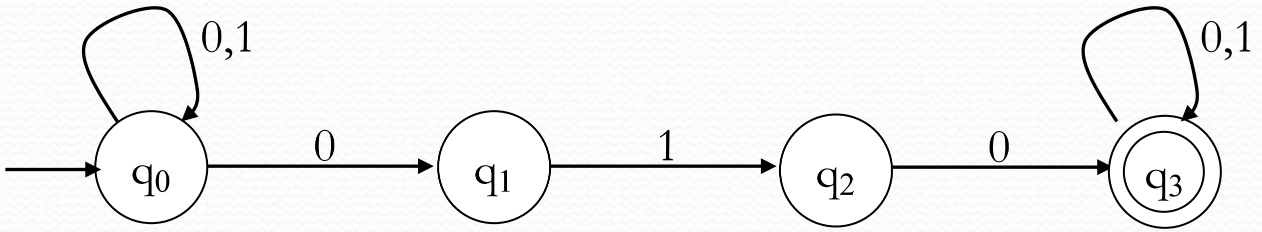
$\rightarrow$Todo AFD es un AFND (es el caso particular en que $|\delta(q,\alpha)|=1$ para cualquier q, α)

$\Rightarrow$ LR $\subseteq$ LN

Veremos que además LN $\subseteq$ LR (y por lo tanto, LN = LR).

AFND → AFD

Idea de por qué LN $\subseteq$ LR :
consideremos de nuevo el AFND que acepta el lenguaje de strings que incluyen 010.



Veamos los posibles recorridos al leer la palabra 0101:

$(q_0, 0101) \rightarrow (q_0, 101) \rightarrow (q_0, 01) \rightarrow (q_0, 1) \rightarrow (q_0, \varepsilon)$ rechaza

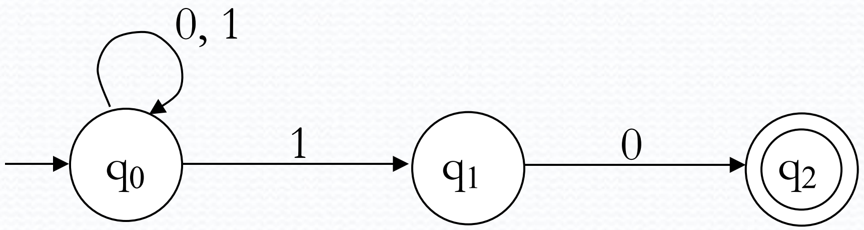
$(q_0, 01) \rightarrow (q_1, 1) \rightarrow (q_2, \varepsilon)$ rechaza

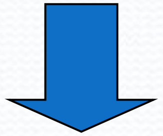
$(q_0, 0101) \rightarrow (q_1, 101) \rightarrow (q_2, 01) \rightarrow (q_3, 1) \rightarrow (q_3, \varepsilon)$ acepta

AFND → AFD

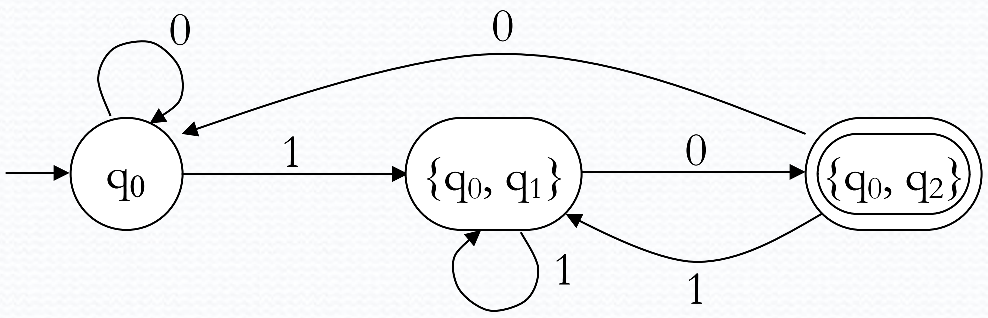
Otro ejemplo:

AFND





AFD



AFND $\to$ AFD

Método general para construir el AFD M′ equivalente a un AFND $M=(Q, \Sigma, \delta, q_0, F)$:

- $Q' = 2^Q$
- $\Sigma' = \Sigma$
- $q'_0 = \{q_0\}$
- $F' = \{ A \subseteq Q : A \cap F \neq \emptyset \}$
- $\delta'(A, \alpha) = \{ q: \exists p \in A, q \in \delta(p,\alpha) \} = \bigcup_{p \in A} \delta(p, \alpha)$

¿Será cierto que $L(M')=L(M)$?

Ejercicio

- Diseñe un autómata finito no determinista que acepte las cadenas que terminan en 1 del alfabeto {0,1}

Conversión de AFND a AFD

En los autómatas finitos no deterministas,

Para algún estado actual y símbolo de entrada, existe más de un estado de salida siguiente.

Una cadena se acepta sólo si existe al menos un camino de transición que comienza en el estado inicial y termina en el estado final..

Conversión de AFND a AFD

En los autómatas finitos no deterministas,

Para algún estado actual y símbolo de entrada, existe más de un estado de salida siguiente.

Una cadena se acepta sólo si existe al menos un camino de transición que comienza en el estado inicial y termina en el estado final..

Se siguen los siguientes pasos para convertir un AFND dado en un AFD

AFND a AFD – Paso 1

Sea Q' un nuevo conjunto de estados del AFD. Q' es nulo en el inicio.

Sea T' una nueva tabla de transición del AFD.

AFND a AFD – Paso 2

Añadir el estado de inicio de la AFND a Q'.

Añade las transiciones del estado inicial a la tabla de transición T'.

Si el estado inicial hace la transición a múltiples estados para algún alfabeto de entrada, entonces trata esos estados múltiples como un solo estado en el AFD.

AFND a AFD – Paso 2

Añadir el estado de inicio de la AFND a Q′.

Añade las transiciones del estado inicial a la tabla de transición T′.

Si el estado inicial hace la transición a múltiples estados para algún alfabeto de entrada, entonces trata esos estados múltiples como un solo estado en el AFD.

En NFA, si la transición del estado inicial sobre algún alfabeto de entrada es nula, entonces realiza la transición del estado inicial sobre ese alfabeto de entrada a un estado muerto en el DFA.

AFND a AFD – Paso 3

Si hay algún estado nuevo en la tabla de transición T',

Añade el nuevo estado en Q'.
Añade las transiciones de ese estado en la tabla de transición T'.

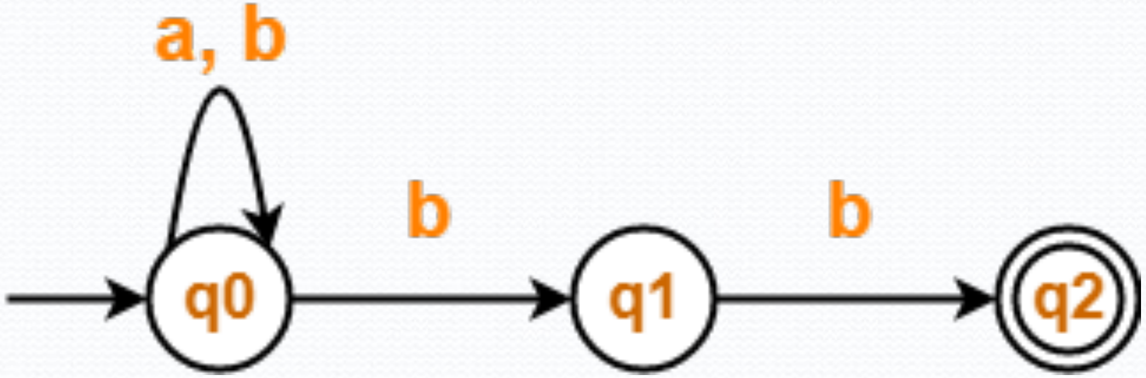
AFND a AFD – Paso 4

Se repite el Paso-03 hasta que no haya ningún estado nuevo en la tabla de transición T'.

Finalmente, la tabla de transición T' así obtenida es la tabla de transición completa del AFD requerido.

Ejemplo

Convertir el siguiente AFND a un AFD

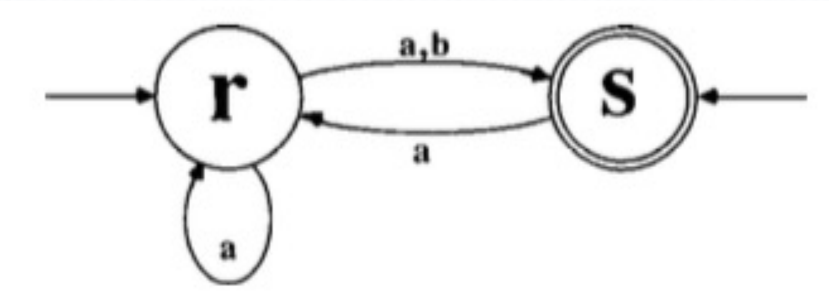


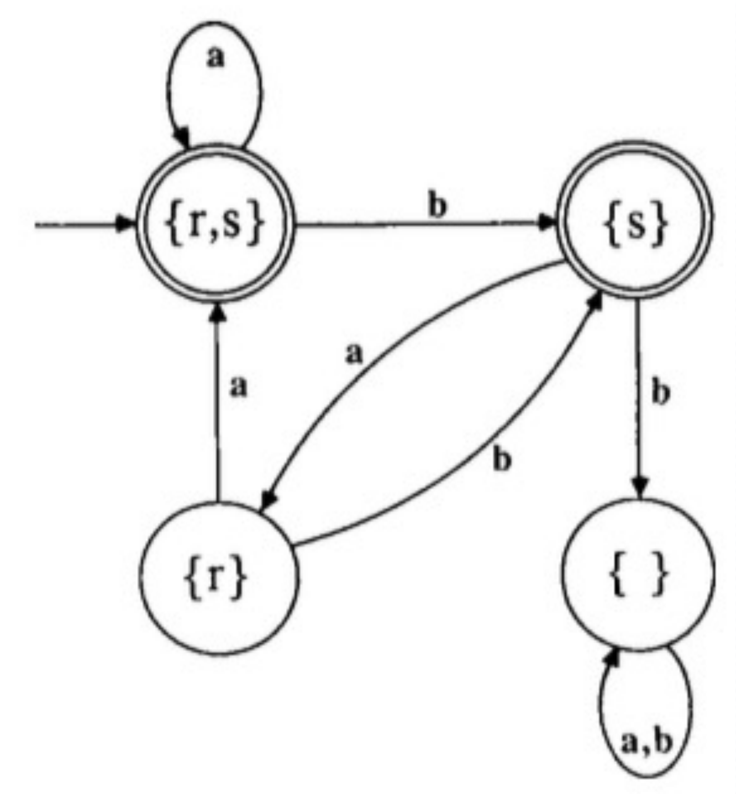
Qué cadenas acepta?

Estado / Alfabeto	a	b
→qo	qo	qo, q1
q1	–	*q2
***q2**	–	–

Ejemplo

Verificar





AFND a AFD – Paso 1

Inicializa Q' como nulo en el inicio.

Crea T' una nueva tabla de transición del AFD.

Estado / Alfabeto	a	b
→q0	q0	q0, q1
q1	–	*q2
*q2	–	–

AFND a AFD – Paso 2

Añadir las transiciones del estado inicial q0 a la tabla de transición T'.

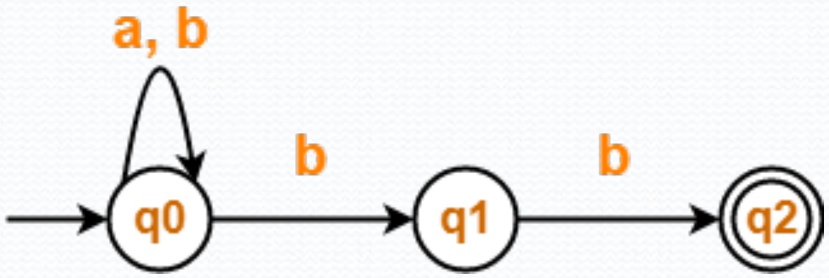
Estado / Alfabeto	a	b
→q0	q0	{q0, q1}

AFND a AFD – Paso 3

El nuevo estado presente en el estado Q' es {q0, q1}.

Añade transiciones para el conjunto de estados {q0, q1} a la tabla de transiciones T'.

Estado / Alfabeto	a	b
→**qo**	qo	{qo, q1}



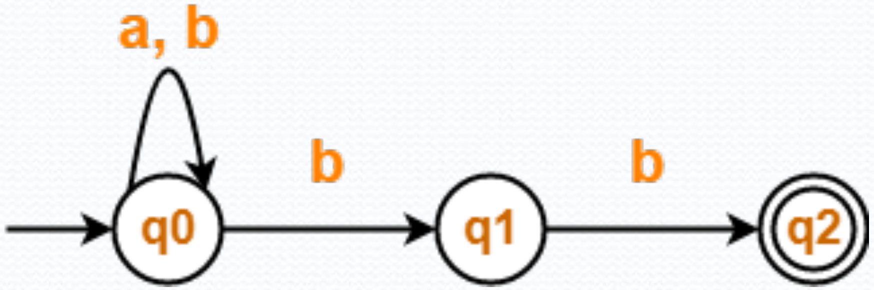
Estado / Alfabeto	a	b
→**qo**	qo	{qo, q1}
{qo, q1}	qo	{qo, q1, q2}

AFND a AFD – Paso 4

El nuevo estado presente en el estado Q' es {q0, q1, q2}.

Añade transiciones para el conjunto de estados {q0, q1, q2} a la tabla de transiciones T'.

Estado / Alfabeto	a	b
→q0	q0	{q0, q1}
{q0, q1}	q0	{q0, q1, q2}



Estado / Alfabeto	a	b
→q0	q0	{q0, q1}
{q0, q1}	q0	{q0, q1, q2}
{q0, q1, q2}	q0	{q0, q1, q2}

AFND a AFD – Paso 5

Como no quedan nuevos estados por añadir en la tabla de transición T', nos detenemos.

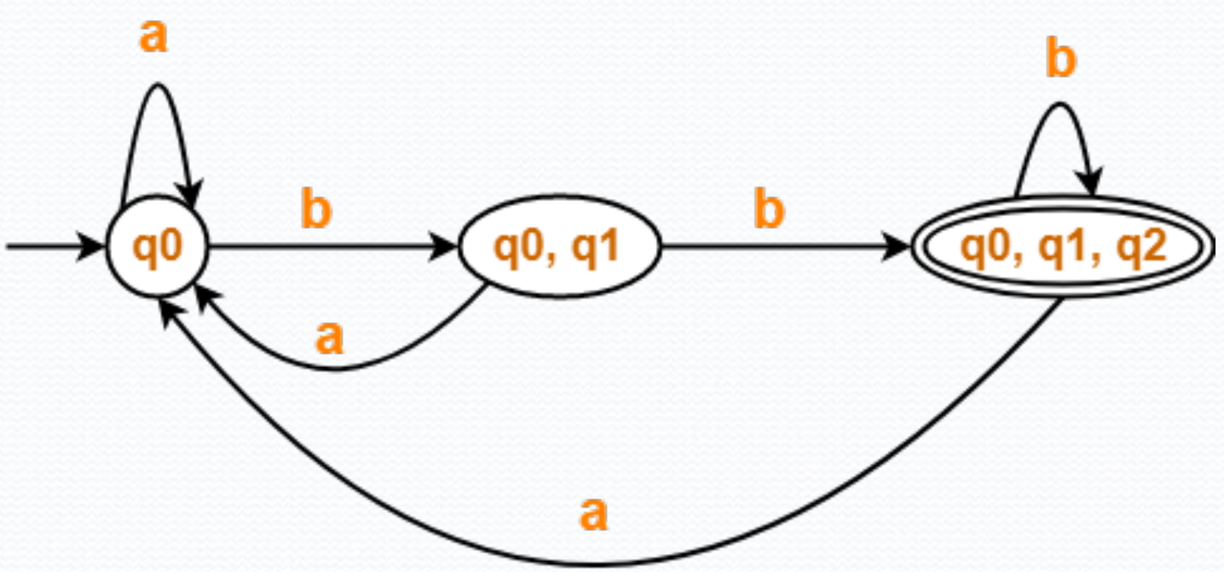
Los estados que contienen q2 como su componente son tratados como estados finales del AFD.

Estado / Alfabeto	**a**	**b**
→q_0	q_0	{q_0, q_1}
{q_0, q_1}	q_0	{q_0, q_1, q_2}
***{q_0, q_1, q_2}**	q_0	{q_0, q_1, q_2}

AFND a AFD – Paso 5

El AFD se dibuja de la forma

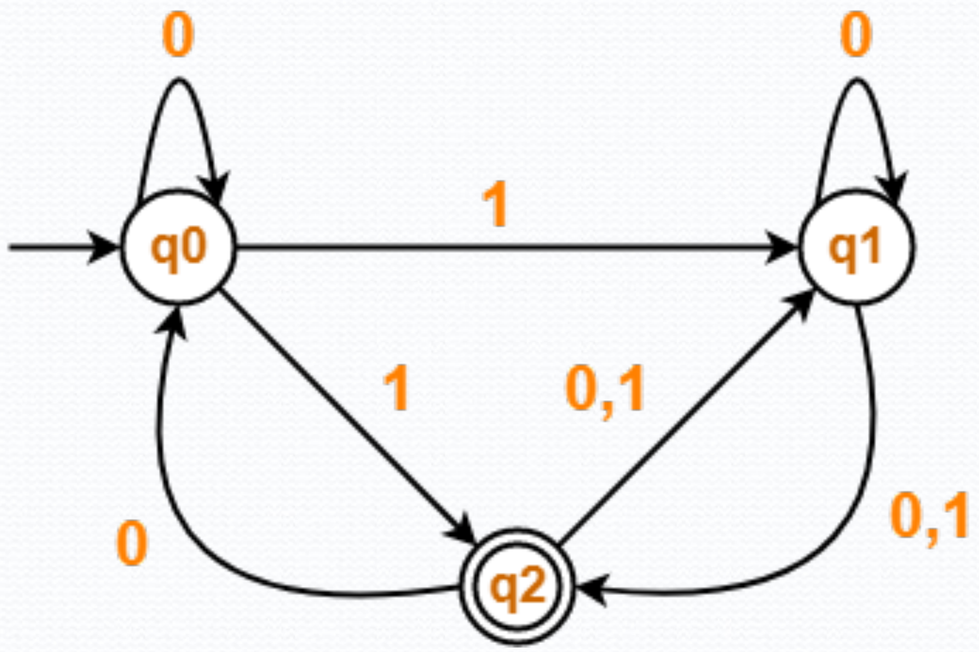
Estado / Alfabeto	a	b
→qo	qo	{qo, q1}
{qo, q1}	qo	{qo, q1, q2}
***{qo, q1, q2}**	qo	{qo, q1, q2}



Deterministic Finite Automata (DFA)

Ejemplo 2

Convertir el siguiente AFND a un AFD



Estado / Alfabeto	0	1
→q0	q0	q1, *q2
q1	q1, *q2	*q2
***q2**	q0, q1	q1

AFND a AFD – Paso 1

Inicializa Q' como nulo en el inicio.

Crea T' una nueva tabla de transición del AFD.

Estado / Alfabeto	0	1
→q0	q0	q1, *q2
q1	q1, *q2	*q2
***q2**	q0, q1	q1

AFND a AFD – Paso 2

Añadir las transiciones del estado inicial q0 a la tabla de transición T'.

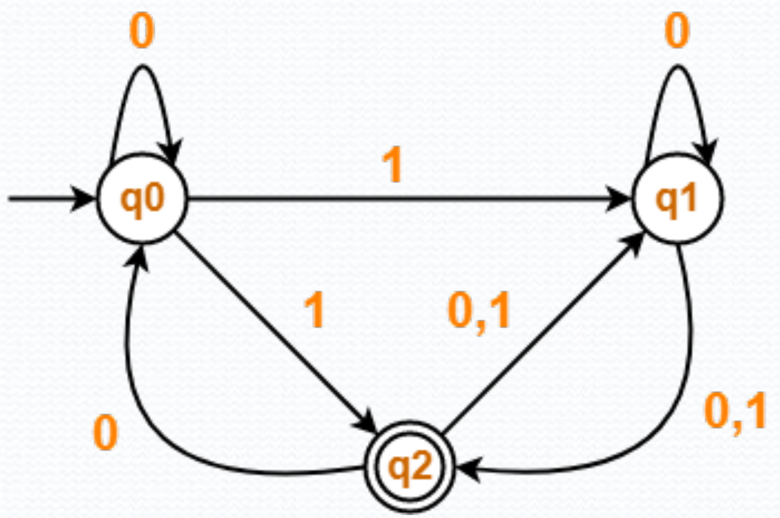
Estado / Alfabeto	0	1
→q0	q0	{q1, q2}

AFND a AFD – Paso 3

El nuevo estado presente en el estado Q' es {q1, q2}.

Añade transiciones para el conjunto de estados {q1, q2} a la tabla de transiciones T'.

Estado / Alfabeto	0	1
→q0	q0	{q1, q2}



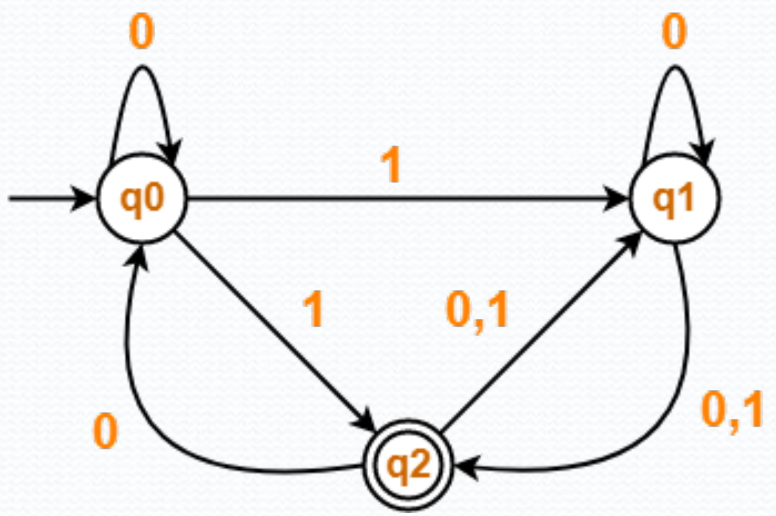
Estado / Alfabeto	0	1
→q0	q0	{q1, q2}
{q1, q2}	{q0, q1, q2}	{q1, q2}

AFND a AFD – Paso 4

El nuevo estado presente en el estado Q' es {q0, q1, q2}.

Añade transiciones para el conjunto de estados {q0, q1, q2} a la tabla de transiciones T'.

Estado / Alfabeto	0	1
→q0	q0	{q1, q2}
{q1, q2}	{q0, q1, q2}	{q1, q2}



Estado / Alfabeto	0	1
→q0	q0	{q1, q2}
{q1, q2}	{q0, q1, q2}	{q1, q2}
{q0, q1, q2}	{q0, q1, q2}	{q1, q2}

AFND a AFD – Paso 5

Como no quedan nuevos estados por añadir en la tabla de transición T', nos detenemos.

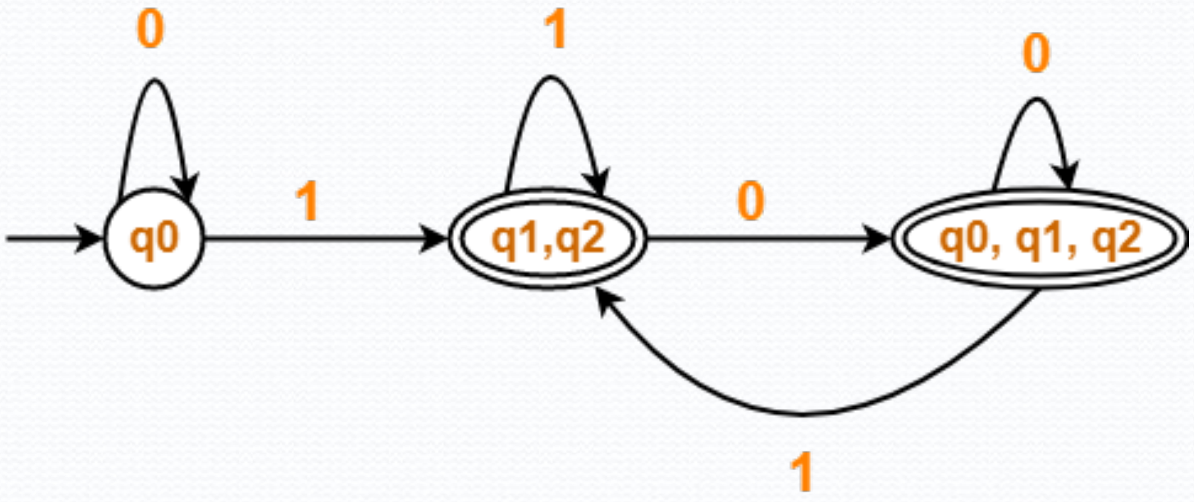
Los estados que contienen q2 como su componente son tratados como estados finales del AFD.

Estado / Alfabeto	0	1
→q0	q0	{q1, q2}
***{q1, q2}**	{q0, q1, q2}	{q1, q2}
***{q0, q1, q2}**	{q0, q1, q2}	{q1, q2}

AFND a AFD

El AFD se dibuja de la forma

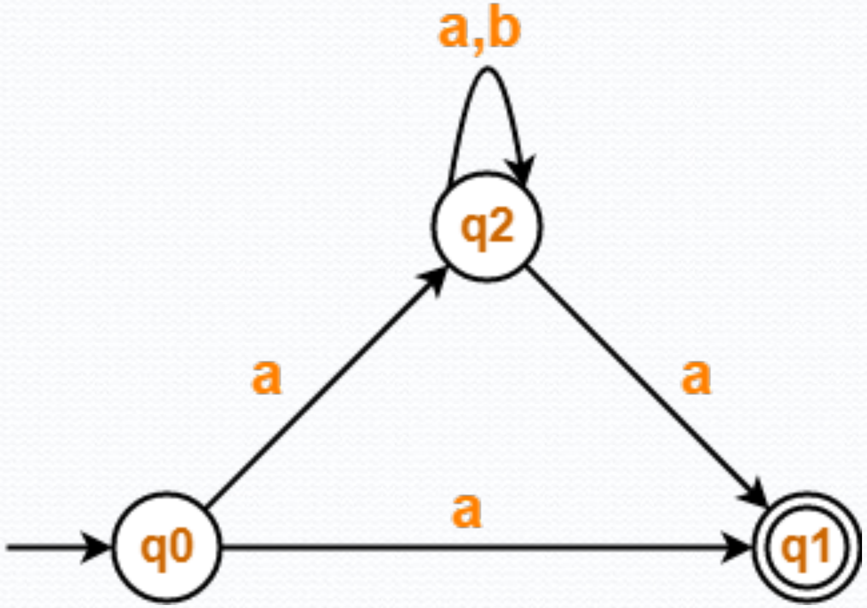
Estado / Alfabeto	0	1
→q0	q0	{q1, q2}
***{q1, q2}**	{q0, q1, q2}	{q1, q2}
***{q0, q1, q2}**	{q0, q1, q2}	{q1, q2}



Deterministic Finite Automata (DFA)

Ejemplo 3

Convertir el siguiente AFND a un AFD



Estado / Alfabeto	a	b
→q0	*q1, q2	–
*q1	–	–
q2	*q1, q2	q2

AFND a AFD – Paso 1

Inicializa Q' como nulo en el inicio.

Crea T' una nueva tabla de transición del AFD.

Estado / Alfabeto	a	b
→qo	*q1, q2	–
***q1**	–	–
q2	*q1, q2	q2

AFND a AFD – Paso 2

Añadir las transiciones del estado inicial q0 a la tabla de transición T'.

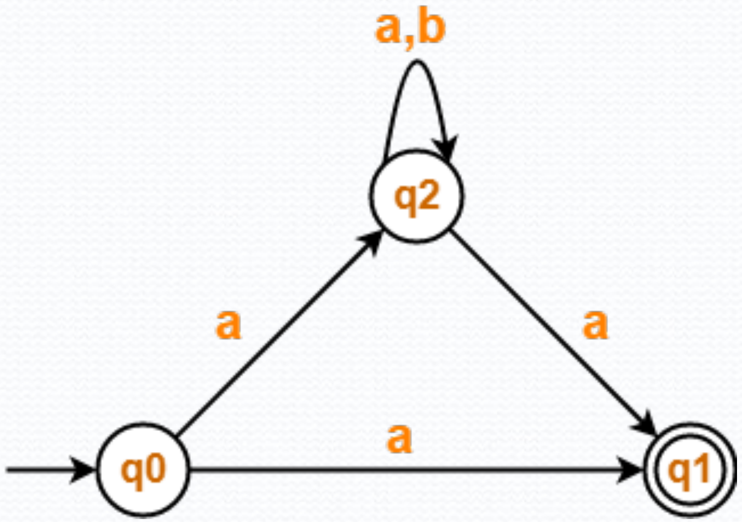
Estado / Alfabeto	a	b
→qo	{q1, q2}	Ø (Dead State)

AFND a AFD – Paso 3

El nuevo estado presente en el estado Q' es {q1, q2}.

Añade transiciones para el conjunto de estados {q1, q2} a la tabla de transiciones T'.

Estado / Alfabeto	a	b
→qo	{q1, q2}	Ø (Dead State)



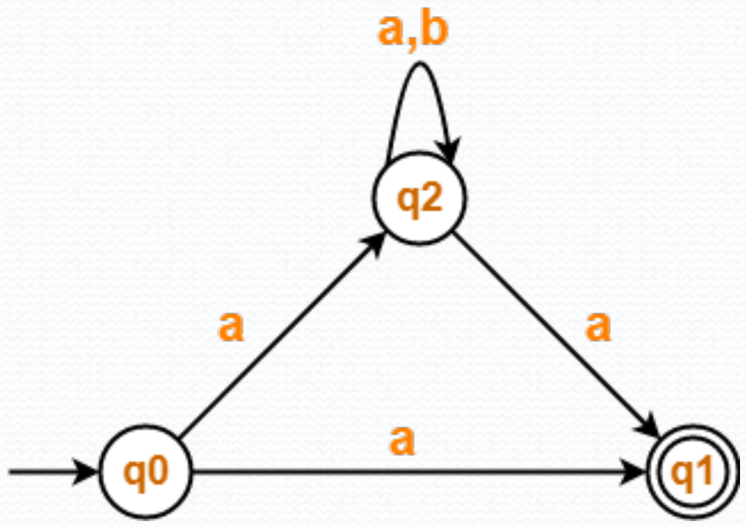
Estado / Alfabeto	a	b
→qo	{q1, q2}	Ø
{q1, q2}	{q1, q2}	q2

AFND a AFD – Paso 4

El nuevo estado presente en el estado Q' es {q2}.

Añade transiciones para el conjunto de estados {q2} a la tabla de transiciones T'.

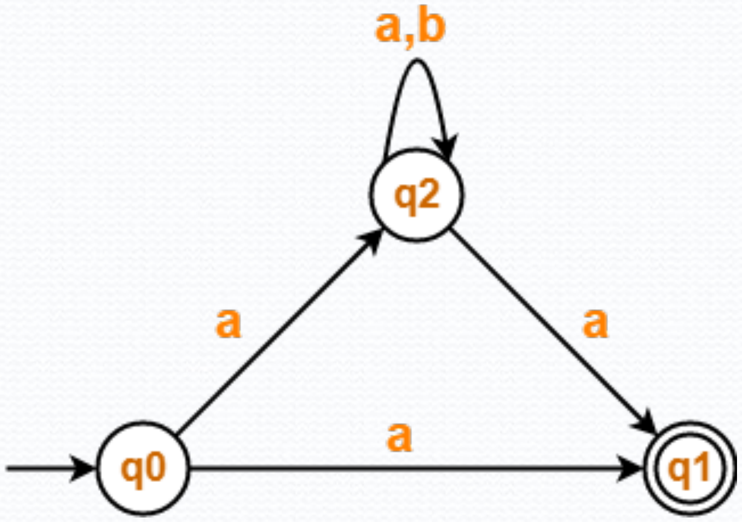
Estado / Alfabeto	a	b
→qo	{q1, q2}	Ø
{q1, q2}	{q1, q2}	q2



Estado / Alfabeto	a	b
→qo	{q1, q2}	Ø
{q1, q2}	{q1, q2}	q2
q2	{q1, q2}	q2

AFND a AFD – Paso 5

Añadir transiciones para el estado muerto {Ø} a la tabla de transición T'.

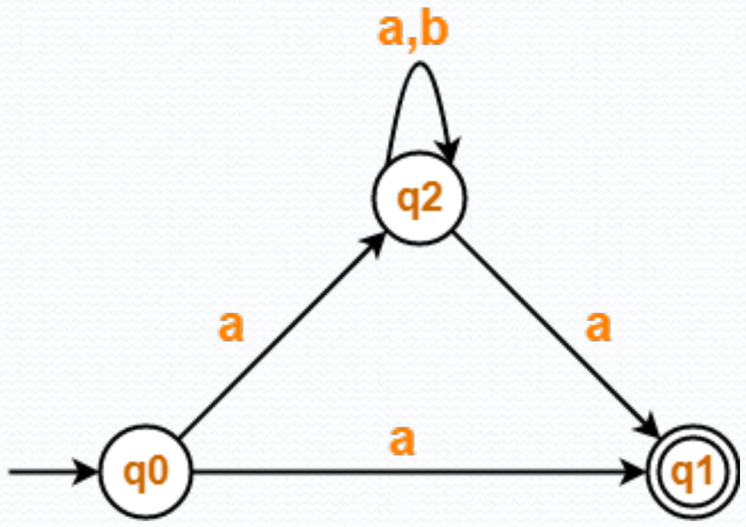


Estado / Alfabeto	a	b
→qo	{q1, q2}	Ø
{q1, q2}	{q1, q2}	q2
q2	{q1, q2}	q2
Ø	Ø	Ø

AFND a AFD – Paso 6

Como no quedan nuevos estados por añadir en la tabla de transición T', nos detenemos.

Los estados que contienen q1 como su componente son tratados como estados finales del AFD.

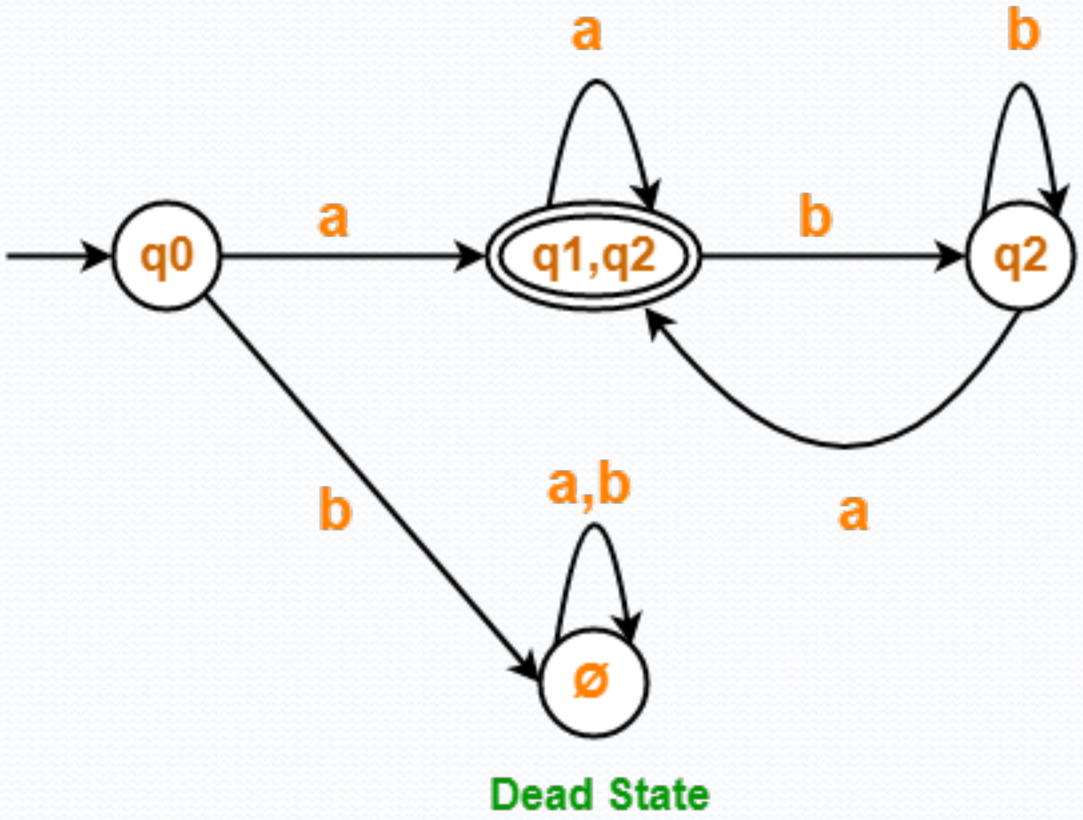


Estado / Alfabeto	a	b
→qo	{q1, q2}	Ø
***{q1, q2}**	{q1, q2}	q2
q2	{q1, q2}	q2
Ø	Ø	Ø

AFND a AFD

El AFD se dibuja de la forma

Estado / Alfabeto	a	b
→qo	{q1, q2}	Ø
*{q1, q2}	{q1, q2}	q2
q2	{q1, q2}	q2
Ø	Ø	Ø



Deterministic Finite Automata (DFA)

Notas

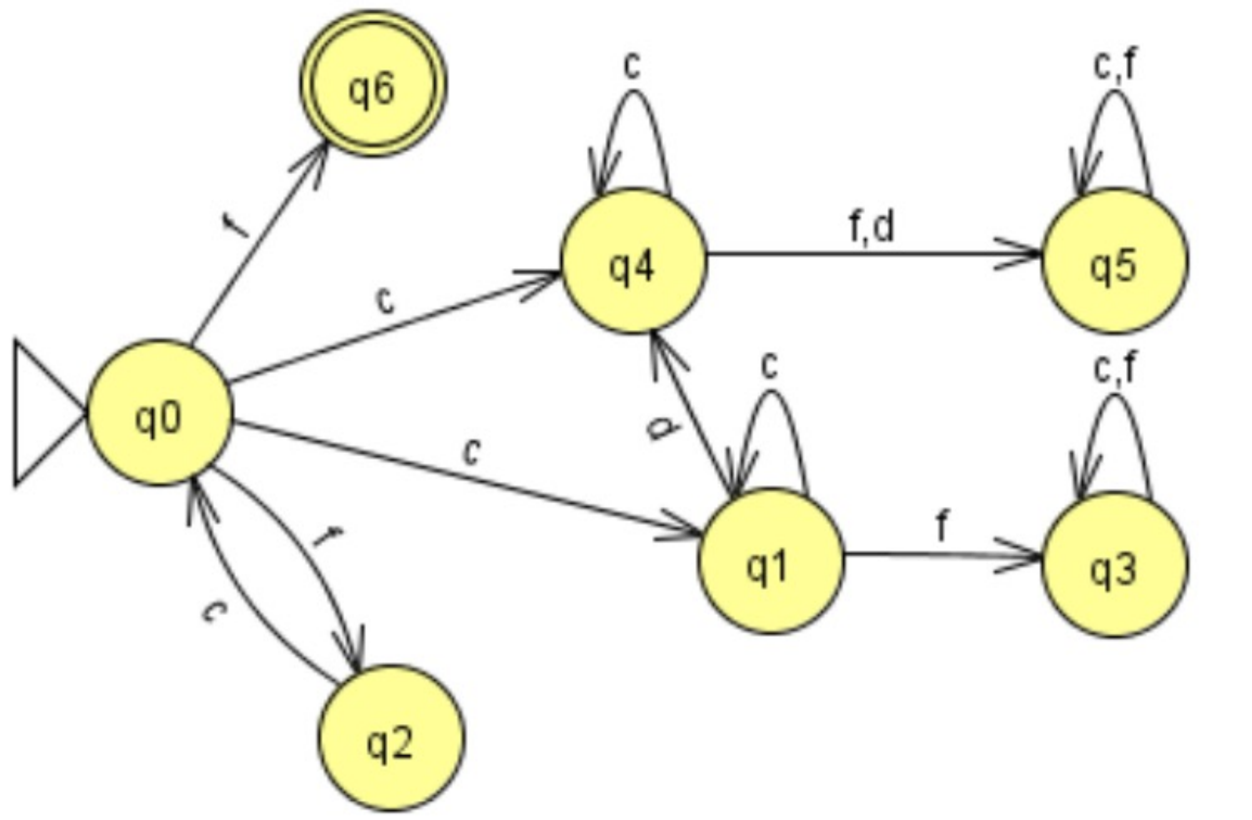
Después de la conversión, el número de estados en el ADF resultante puede o no ser el mismo que el AFN.

El número máximo de estados que puede haber en el ADF es $2^{\text{Número de estados en el AFN}}$.

Ejercicio

Convertir el siguiente AFND a un ADF

	c	f	d
→Q0	Q1,Q4	Q6,Q2	
Q1	Q1	Q3	Q4
Q2	Q0		
Q3	Q3	Q3	
Q4	Q4	Q5	Q5
Q5	Q5	Q5	
* Q6			



Ejercicio

Solución

		c	f	d
	→Q0	Q7	Q8	Q9
{Q1,Q4}=	**Q7**	Q7	Q10	Q11
{Q2,Q6}=	*** Q8**	Q0	Q9	Q9
{Q3,Q5}=	**Q10**	Q10	Q10	Q9
{Q4,Q5}=	**Q11**	Q11	Q5	Q5
	Q5	Q5	Q5	Q9
	Q9	Q9	Q9	Q9