Autómatas finitos no deterministas

Emanuel Alvaredo, Federico Lochbaum

Repaso de Autómata finito determinístico

- Mecanismos o sistemas estímulo-respuesta
- En cada momento la máquina está en un único y determinado estado
- Para cada estímulo y en cada estado, hay una única respuesta (cambio de estado; output)
- No existe actividad que no responda a un estímulo

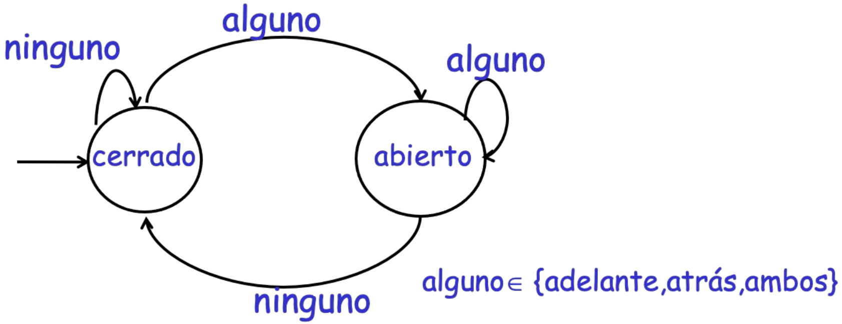
Autómata finito no determinístico

- Mecanismos o sistemas estímulo-respuesta
- ~~En cada momento la máquina está en un único y determinado estado~~
- ~~Para cada estímulo y en cada estado, hay una única respuesta (cambio de estado; output)~~
- ~~No existe actividad que no responda a un estímulo~~

El problema de la puerta

Pero... Qué significa el estímulo “ninguno”?

Significa que los sensores no detectan a ninguna persona cerca de la puerta, ni adelante, ni atrás. Es decir, el sistema detecta la ausencia de un estímulo.

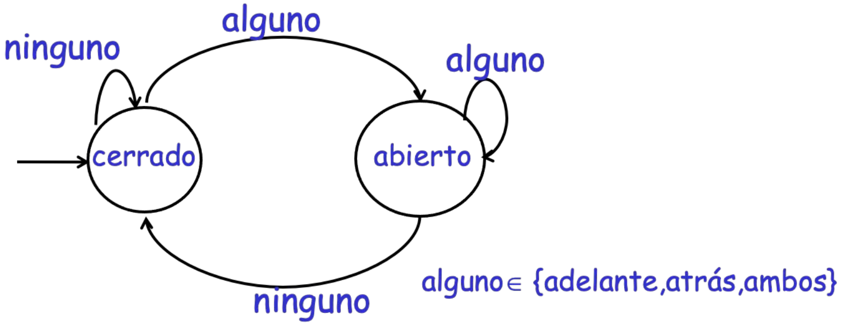


¿Cómo podemos representar esa situación?

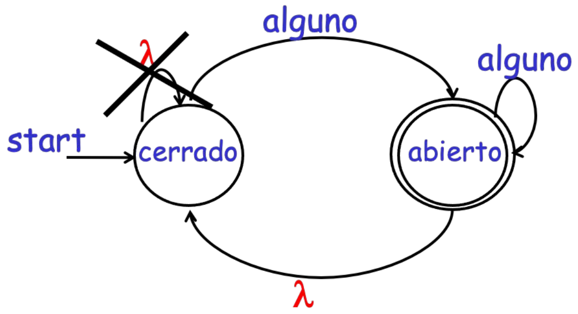
El problema de la puerta

Pero… Qué significa el estímulo “ninguno”?

Significa que los sensores no detectan a ninguna persona cerca de la puerta, ni adelante, ni atrás. Es decir, el sistema detecta la ausencia de un estímulo.



¿Cómo podemos representar esa situación? Con **no determinismo**!

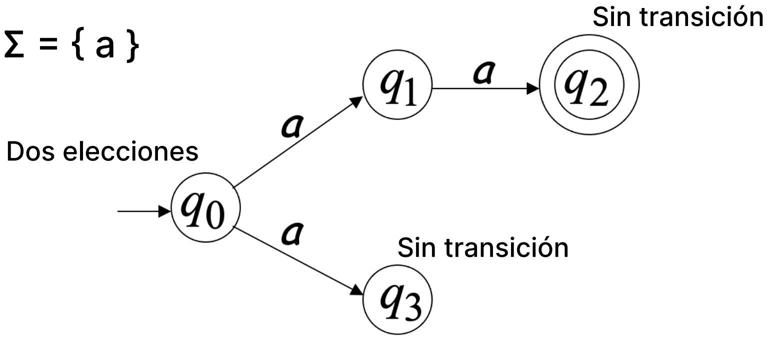


Lenguaje reconocido:

$(\{alguno\}\{alguno\}^{*})+ = (\{alguno\}^{+})^{+} = \{alguno\}^{+}$

Análisis de un AFND

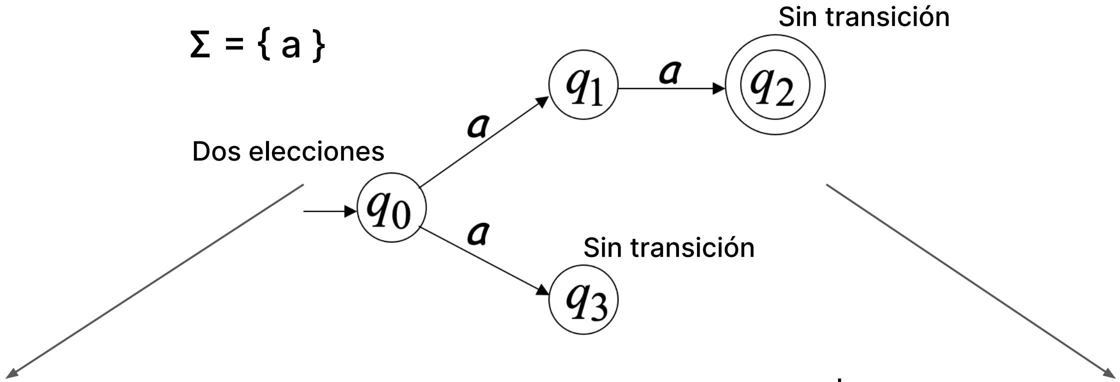
$\Sigma = \{ a \}$

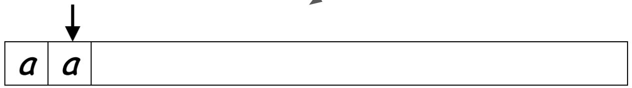


Análisis de un AFND

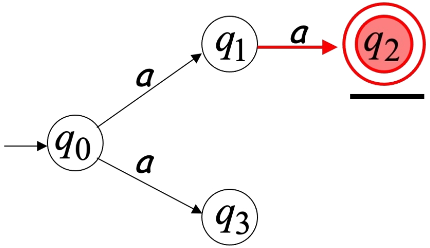
Input = aa

$\Sigma = \{ a \}$



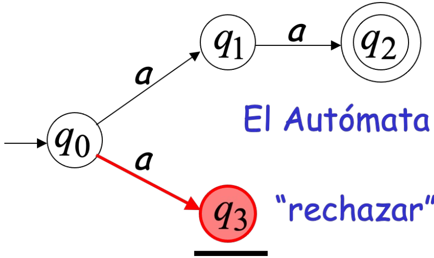


Todo el input consumido



"aceptar"

El input no puede ser consumido



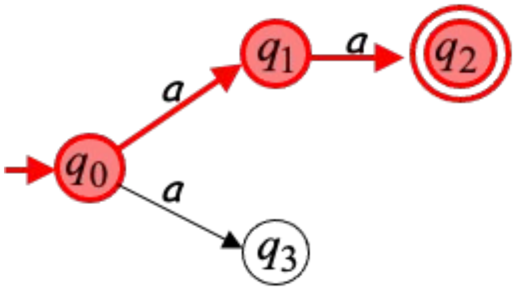
El Autómata Para

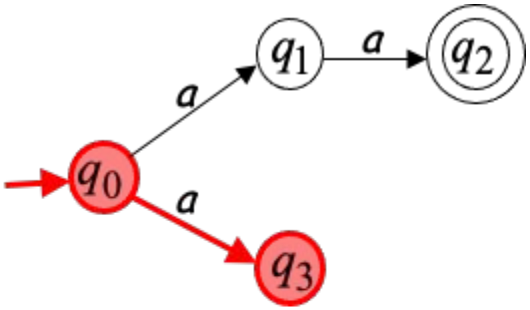
"rechazar"

Aceptación de un AFND

Vamos a decir que un AFND acepta una cadena **si existe** una computación de la cadena que es aceptada por el AFND

Ejemplo: aa es aceptado por el AFND



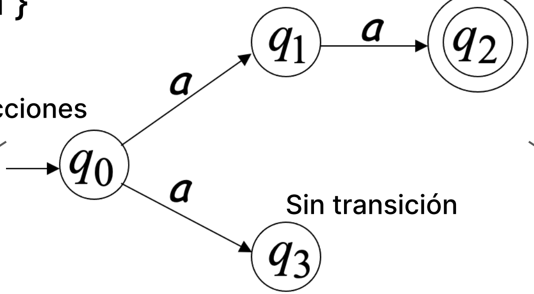


Análisis de un AFND

Input = aaa

$\Sigma = \{ a \}$

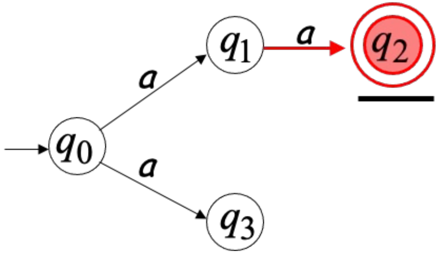
Sin transición

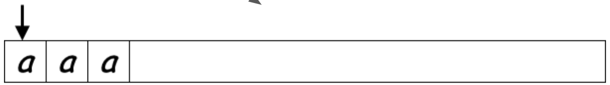


Dos elecciones

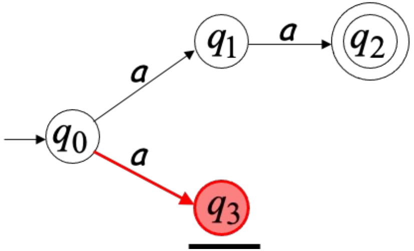
Sin transición

El input no puede seguir consumiéndose





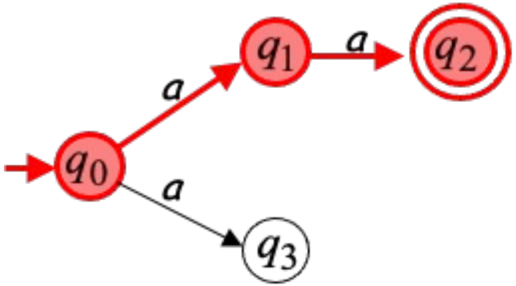
El input no puede seguir consumiéndose

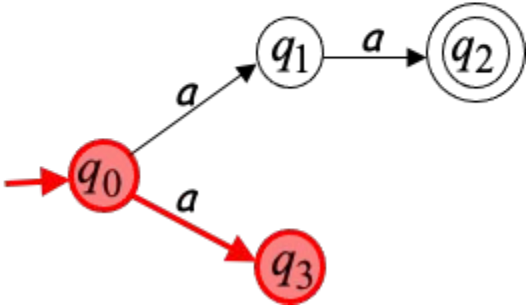


Rechazo de un AFND

Vamos a decir que un AFND rechaza una cadena **si NO existe** una computación de la cadena que es aceptada por el AFND

Ejemplo: aaa es rechazado por el AFND

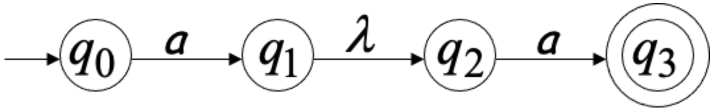




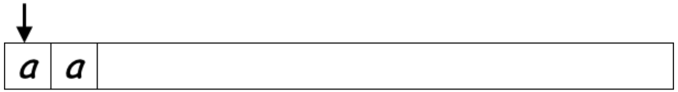
Todas las posibles computaciones rechazan

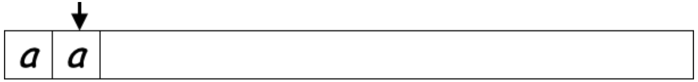
Transiciones lambda

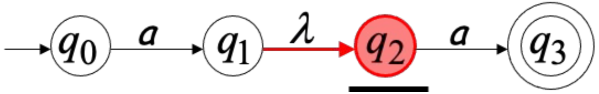
L(M) = { aa }

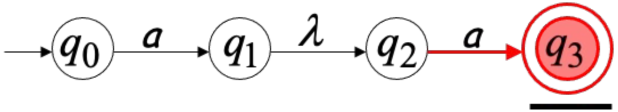


Notar que el cabezal no se mueve

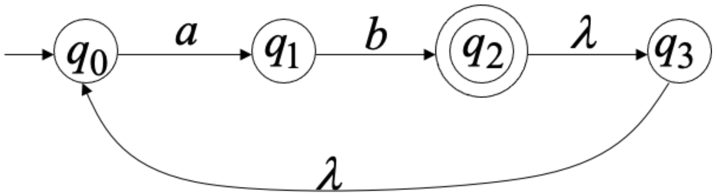




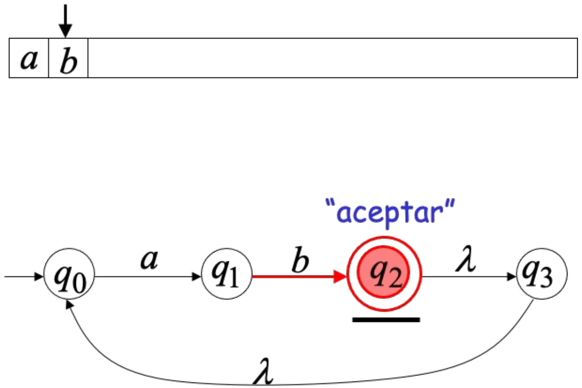


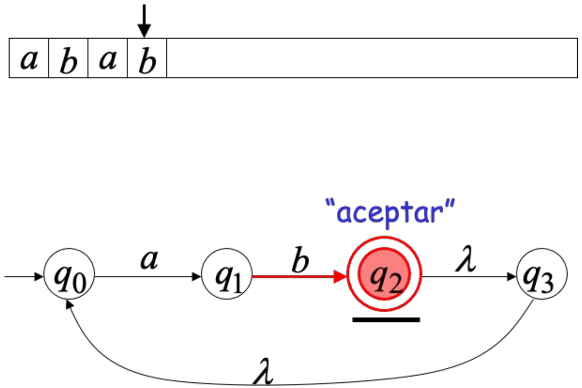


Otro ejemplo de AFND

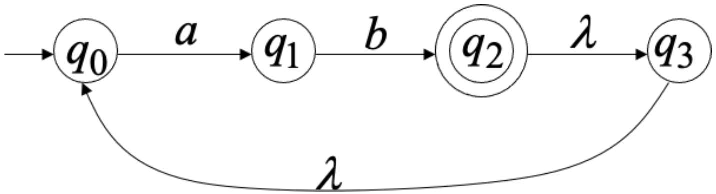


Cual es el lenguaje de M ?

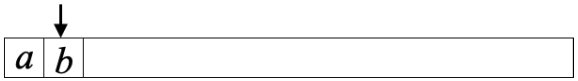


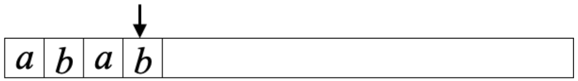


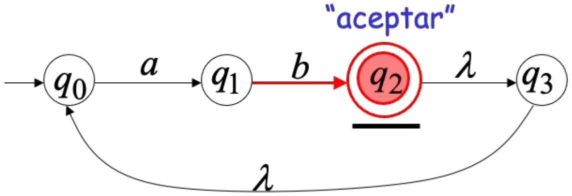
Otro ejemplo de AFND

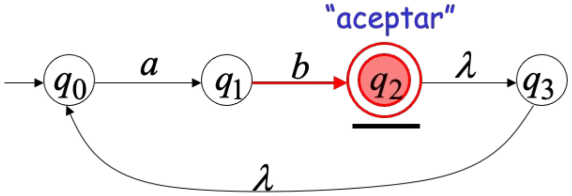


$$L = \{ab,\ abab,\ ababab,\ \ldots\}$$
$$= \{ab\}^{+}$$

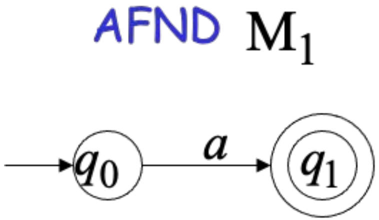




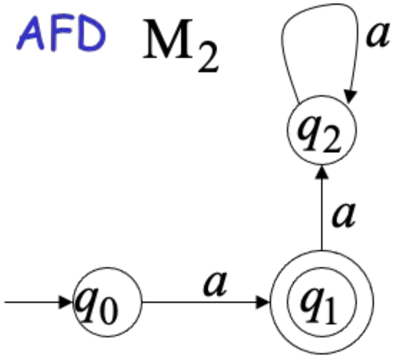




Los AFNDs son interesantes porque podemos expresar lenguajes con mayor facilidad que con los AFDs



$$L(M_1) = \{a\}$$



$$L(M_2) = \{a\}$$

Definición formal de AFND

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

***Q* : Conjunto de estados**

***Σ* : Alfabeto de input**

***δ* : Función de transición**

q_0 : Estado inicial ($q_0 \in Q$)

***F* : Conjunto de estados finales ($F \subset Q$)**

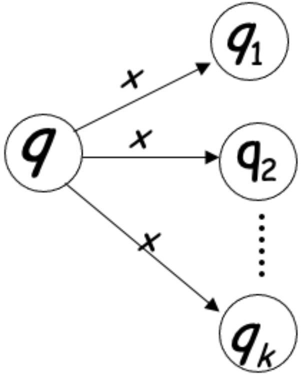
$$\delta : \mathrm{Q} \times \Sigma_{\{\lambda\}} \to \boldsymbol{P}(Q)$$

$$donde\ \Sigma_{\{\lambda\}} = \Sigma \cup \{\lambda\}$$

$$\lambda \notin \Sigma$$

Función de transición

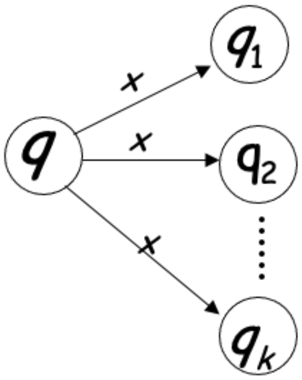
$$\delta(q, x) = \{q_1, q_2, \ldots, q_k\}$$



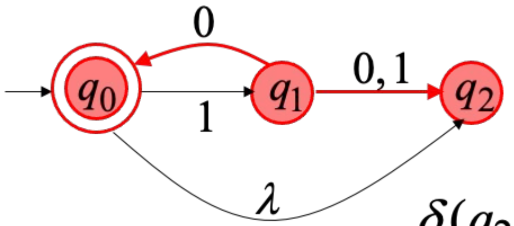
Estados resultantes de seguir UNA transición con el símbolo x

Función de transición

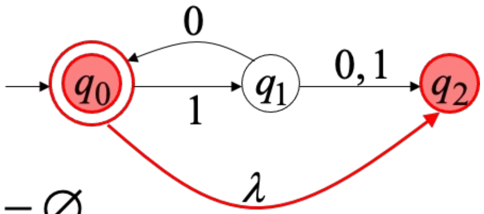
$$\delta(q, x) = \{q_1, q_2, \ldots, q_k\}$$



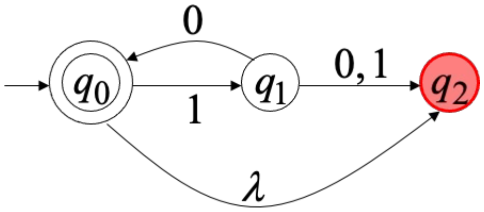
$$\delta(q_1, 0) = \{q_0, q_2\}$$



$$\delta(q_0, \lambda) = \{q_2\}$$

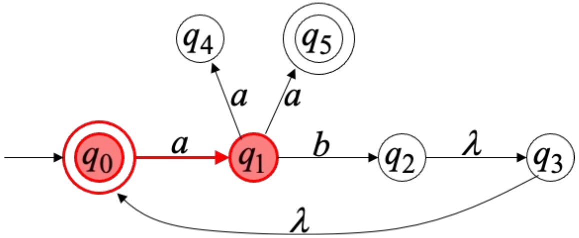


$$\delta(q_2, 1) = \varnothing$$

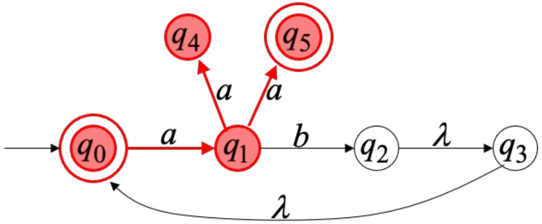


Función de transición extendida δ^*

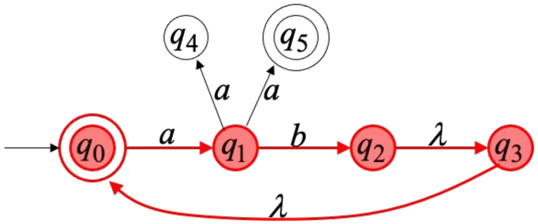
$\delta^*(q_0, a) = \{q_1\}$



$\delta^*(q_0, aa) = \{q_4, q_5\}$



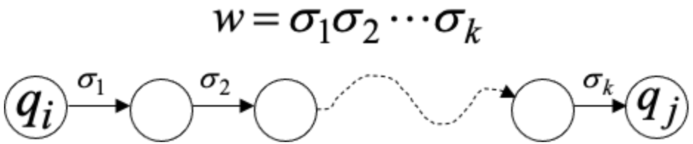
$\delta^*(q_0, ab) = \{q_2, q_3, q_0\}$



Caso especial: Para todo estado q, $q \in \delta^*(q, \lambda)$

Función de transición extendida δ^*

En general, $q_j \in \delta^*(q_i, w)$ indica que existe un camino desde q_i a q_j con la cadena w



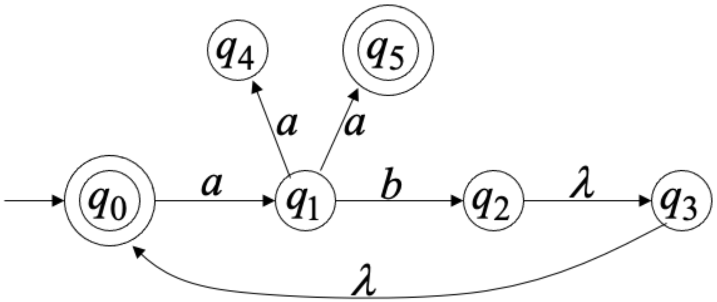
Lenguaje de un AFND

El lenguaje aceptado por un AFND M es $L(M) = \{w_1, w_2, ... w_n, ...\}$ donde $\delta^*(q_0, w_m) = \{q_i, ..., q_k, \ldots, q_j\}$

con $q_k \in F$

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

Lenguaje de un AFND



$$L(M) = \{(ab)^n \mid n \geq 0\} \cup \{(ab)^n \mid n \geq 0\}\{aa\}$$

o bien

$$L(M) = \{ab\}^* \cup \{ab\}^* \{aa\} = \{ab\}^* (\{\lambda\} \cup \{aa\})$$

Equivalencia entre AFNDs y AFDs

Definición: Una máquina M_1 es equivalente a M_2 si L(M1) = L(M2)

Teorema: Lenguajes aceptados por **AFNDs** = Lenguajes aceptados por **AFDs**

Nota: AFNDs y AFDs tienen **el mismo poder computacional**

Demo: Mostramos que

1. Lenguajes aceptados por **AFDs** $\subseteq$ Lenguajes aceptados por **AFNDs**
2. Lenguajes aceptados por **AFNDs** $\subseteq$ Lenguajes aceptados por **AFDs**

Lenguajes aceptados por **AFDs** $\subseteq$ Lenguajes aceptados por **AFNDs**

Trivial, todo AFD define trivialmente un AFND

Lenguajes aceptados por **AFNDs** $\subseteq$ Lenguajes aceptados por **AFDs**

Todo AFND puede ser transformado en un AFD equivalente

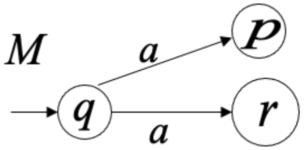
Intuición:

Si **M** es un AFND, construimos un AFD **M'** cuyos estados serán conjuntos de estados de **M,** tal que, por cada paso no determinista de M,

$\delta(q, a) = \{ p, r \}$ **definimos** $\delta'(\{ q \}, a) = \{ p, r \}$, donde $\{ q \}$ y $\{ p, r \}$ son estados de **M'**.

Ejemplo 1

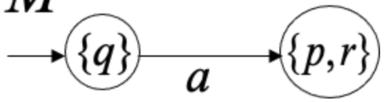
AFND M



$\delta(q,a)=\{p,r\}$

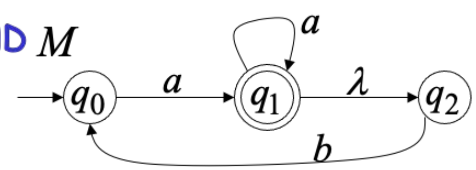
AFD M'

$\delta'(\{q\},a)=\{p,r\}$



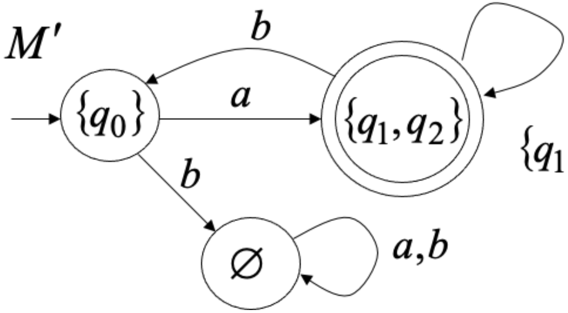
Ejemplo 2

AFND M



$q_1 \in F$

AFD M'



$\{q_1,q_2\} \in F'$

Procedimiento de conversión

Dado un AFND M, generar un AFD M’ tal que L(M) = L(M’)

Paso 1:

Si q_0 es el estado inicial del AFND M, **entonces** $\{q_0\}$ es el estado inicial del AFD M’

Paso 2:

Para todo estado actual del AFD ($\{ q_i, q_j, \ldots, q_m \}$), para cada a calculamos

$$\left.\begin{array}{l} \delta(q_i, a) \\ \cup\, \delta(q_j, a) \\ \ldots \\ \cup\, \delta(q_m, a) \end{array}\right\} = \{q'_k, q'_l, \ldots, q'_n\}$$

y entonces, agregamos al AFD M’ la transición

$$\delta'\left(\{q_i, q_j, \ldots, q_m\},\ a\right) = \{q'_k, q'_l, \ldots, q'_n\}$$

Procedimiento de conversión

Paso 3:

Repetimos el paso 2 para cada estado en AFD y cada símbolo en sigma hasta que no puedan agregarse más estados al AFD

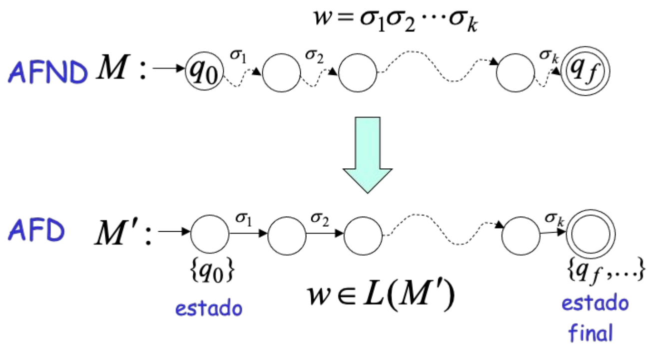
Paso 4:

Para cada estado del AFD M’ de la forma ({ q_i, q_j, ..., q_m }), **si** algún q_j **es estado final** en el AFND M, entonces ({ q_i, q_j, ..., q_m }) **es estado final** de M’

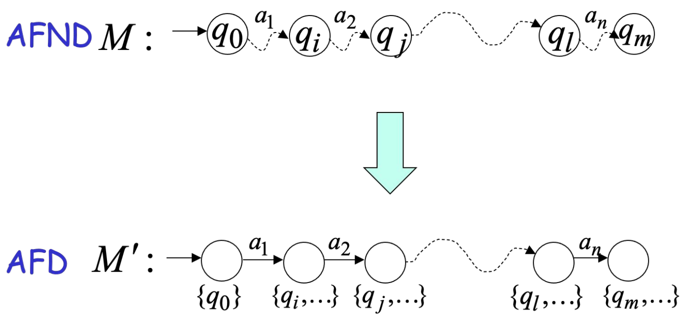
Lema: Si convertimos el AFND M en el AFD M’ de acuerdo con el procedimiento anterior, **entonces** L(M) = L(M’)

Demo: Mostramos que 1) $L(M) \subseteq L(M')$ y 2) $L(M) \supseteq L(M')$

Para mostrar 1, demostramos que si $w \in L(M)$ entonces $w \in L(M')$

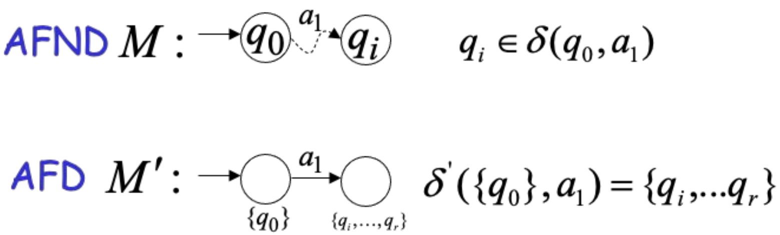


Más genéricamente si el string arbitrario v = $a_1a_2\ldots a_n$



Por inducción sobre | v |

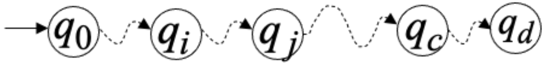
- Caso base | v | = 1, con v = a_1, es cierto por construcción de M’



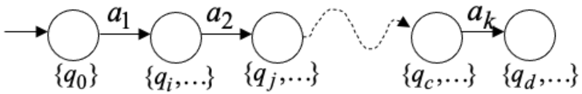
- Caso inductivo | v | = k + 1, $v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$

Hipótesis inductiva: 1 ≤ | v | ≤ k, $v = a_1 a_2 \cdots a_k$ (Asumimos que vale)

AFND M :

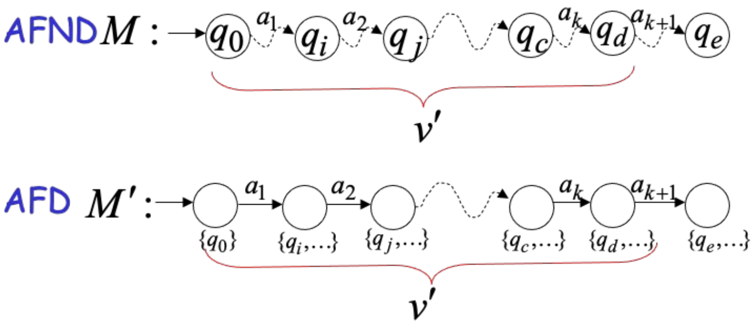


AFD M' :

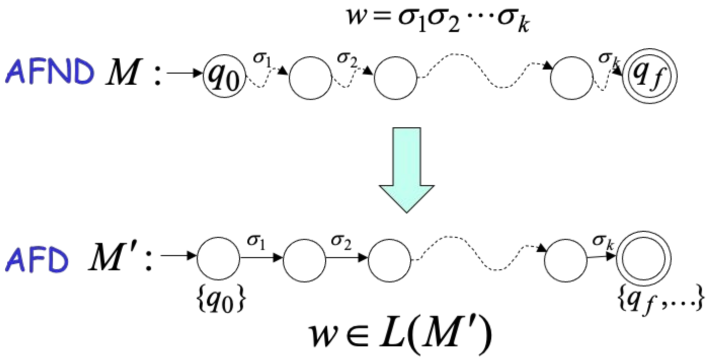


- Caso inductivo | v | = k + 1, $v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$

Vemos que



En consecuencia, $w \in L(M)$



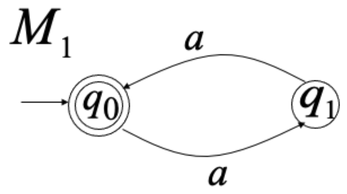
Por lo tanto, $L(M) \subseteq L(M')$ **. Análogamente podemos demostrar** $L(M) \supseteq L(M')$

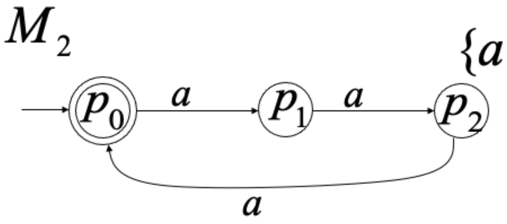
$$L(M) = L(M')$$

Ejemplo

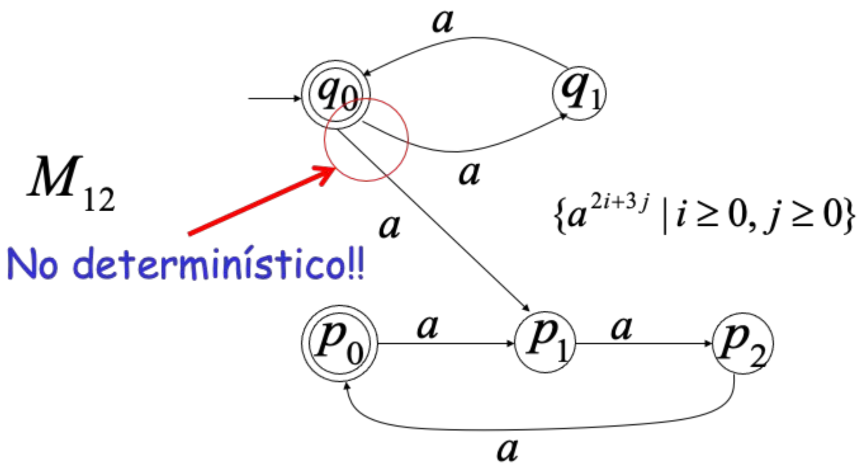
Construimos un AFD para $\{a^{2i+3j} \mid i \geq 0, j \geq 0\}$

Descomponemos el problema en dos subproblemas más sencillos y los **combinamos**





$\{a^{3j} \mid j \geq 0\}$



Conversión a AFD

$\delta(\{q_0\}, a) = \{q_1, p_1\}$

$\delta(\{q_1, p_1\}, a) = \{q_0, p_2\}$

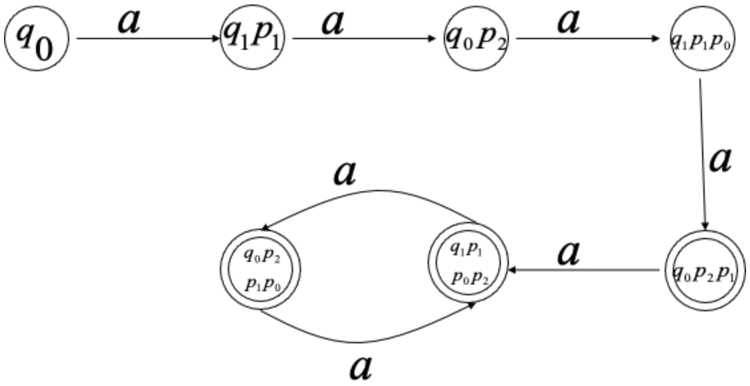
$\delta(\{q_0, p_2\}, a) = \{q_1, p_1, p_0\}$

$\delta(\{q_1, p_1, p_0\}, a) = \{q_0, p_2, p_1\}$

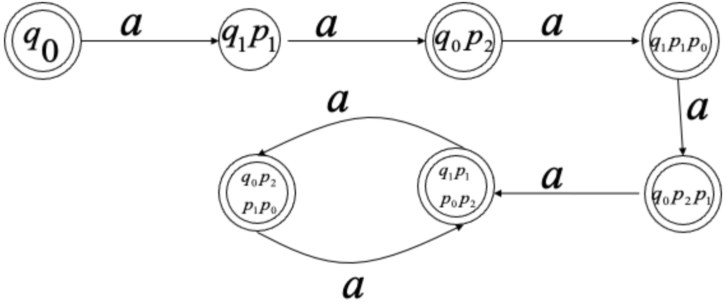
$\delta(\{q_0, p_2, p_1\}, a) = \{q_1, p_1, p_0, p_2\}$

$\delta(\{q_1, p_1, p_0, p_2\}, a) = \{q_0, p_2, p_1, p_0\}$

$\delta(\{q_0, p_2, p_1, p_0\}, a) = \{q_1, p_1, p_0, p_2\}$



Minimización



Con λ calculamos:

$E_{0N} =$

$$\{\{\{q_0\},\{q_0,p_2\},\{q_1,p_1,p_0\},\{q_0,p_2,p_1\},\{q_1,p_1,p_0,p_2\},\{q_0,p_2,p_1,p_0\}\},\{q_1,p_1\}\}$$

Agregando a obtenemos

$E_{1N} =$

$$\{\{q_0\},\{\{q_0,p_2\},\{q_1,p_1,p_0\},\{q_0,p_2,p_1\},\{q_1,p_1,p_0,p_2\},\{q_0,p_2,p_1,p_0\}\},\{q_1,p_1\}\}$$

Agregando aa obtenemos $E_{2N} = E_{1N}$ y termina el proceso