# II Autómatas Finitos y Gramáticas Regulares

## 5.1 Autómatas Finitos: FAs

Definición: Un autómata finito es un 5-tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, donde:

i) $Q$ es un conj. finito no vacío de estados

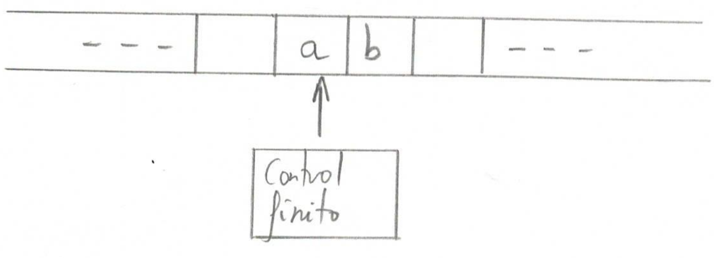
ii) $\Sigma$ es un conj. finito no vacío de símbolos de entrada

iii) $\delta : Q \times \Sigma \longrightarrow P(Q)$ es la función de transición

iv) $q_0 \in Q$ es el estado inicial

v) $F \subseteq Q$ es el conj. de estados finales (o de aceptación)

NOTA: Podemos representar un FA mediante un control finito, el cual se encuentra en un estado $q \in Q$, y lee una secuencia de símbolos en $\Sigma$ que se encuentran escritos en una cinta.




Cadena de entrada

Suponiendo que el símbolo a analizar sea "a", el control finito pasa al estado $p = \delta(q, a)$ y mueve su cabeza de lectura sobre la cinta una posición a la derecha para situarse en el siguiente símbolo de entrada "b"

NOTACION: $(q, a) \vdash (p, b)$

Para describir formalmente el comportamiento de un FA con una cadena de entrada, extenderemos la definición de la función de transición $\delta$.

<u>Definición</u>: Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un FA, definimos

$$\begin{aligned} \hat{\delta} : Q \times \Sigma^* &\longrightarrow P(Q) \\ (q, \varepsilon) &\rightsquigarrow q \\ (q, xa) &\rightsquigarrow \delta(\hat{\delta}(q,x), a) \end{aligned}$$

<u>Lema</u>: Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un FA, entonces:

i) $\hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q,x), y)$, $\forall x, y$

ii) $\hat{\delta}(q,x) = \hat{\delta}(q,y) \Rightarrow \hat{\delta}(q,xz) = \hat{\delta}(q,yz)$, $\forall x,y,z$

<u>demostración</u>

i) Lo haremos por inducción en $|y|$.

<u>$|y| = 1$</u> $\Rightarrow y = a \Rightarrow \hat{\delta}(q,xy) = \hat{\delta}(q,xa) := \delta(\hat{\delta}(q,x), a)$

<u>$|y| \leq n$</u> Supuesto cierto

<u>$|y| = n+1$</u> $\Rightarrow \exists a \in \Sigma \,/\, y = za, |z| = n$ por tanto:

$$\hat{\delta}(q,xy) = \hat{\delta}(q,xza) := \delta(\hat{\delta}(q,xz), a) \underset{\substack{\downarrow \\ \text{inducción por } |z|=n}}{:=} \delta(\hat{\delta}(\hat{\delta}(q,x),z),a)$$

$$:= \hat{\delta}(\hat{\delta}(q,x), za) = \hat{\delta}(\hat{\delta}(q,x), y) \quad \underline{\text{demostrado}}$$

ii) $$\hat{\delta}(q,xz) \underset{\substack{\downarrow \\ i)}}{=} \hat{\delta}(\hat{\delta}(q,x),z) \underset{\substack{\downarrow \\ \hat{\delta}(q,x) = \hat{\delta}(q,y)}}{=} \hat{\delta}(\hat{\delta}(q,y),z) \underset{\substack{\downarrow \\ i)}}{=} \hat{\delta}(q,yz)$$

<u>demostrado</u>

### 5.1.1 Configuraciones, movimientos, aceptación.

<u>Definición</u>: Sean $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un AF
$x \in \Sigma^*$ $\Big\}$, decimos que

<u>$\mathcal{A}$ acepta la cadena $x$</u> sii $(q_0, x) \vdash^* (q, \varepsilon)$ /
/ $q \in F$.

<u>Definición</u>: Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un AF, definimos una <u>configuración de $\mathcal{A}$</u> como un par $(q, w) \in Q \times \Sigma^*$, donde:

i) "$q$" es el estado actual del control finito.

ii) "$w$" es el contenido actual de la cinta de entrada, esto es, el conj. de símbolos que quedan por analizar.

<u>NOTA</u>: Se consideran dos tipos particulares de configuración:

i) Las <u>configuraciones iniciales</u>, esto es, aquellas en las que se encuentra $\mathcal{A}$ en un estado inicial. Son de la forma $(q_0, w)$.

ii) Las <u>configuraciones finales</u>, esto es, aquellas en las que se encuentra el control finito en uno de sus estados finales y la cadena que queda por analizar es $\varepsilon$. Son de la forma $(q, \varepsilon)$ / $q \in F$.

<u>Definición</u>: Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un FA. Entonces un <u>movimiento</u> de $\mathcal{A}$ es el paso de una configuración a otra mediante la aplicación de una transición. Esto es,

$(q, w) \vdash (p, x)$

<u>Definición</u>: Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un FA, definimos el conj. <u>$T(\mathcal{A})$</u> como el conj. de cadenas aceptadas por $\mathcal{A}$. Esto es, $T(\mathcal{A}) := \{x \in \Sigma^* / \delta(q_0, x) \in F\}$

## 5.1.2 Representación de FAs: tablas de transiciones, grafos de transiciones.

<u>Definición</u>: Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un FA, una <u>tabla de transiciones</u> para $\mathcal{A}$ es aquella que describe el funcionamiento de $\delta$ sobre los estados de $\mathcal{A}$.

<u>Ejemplo</u>: Sea $\mathcal{A} = (\{p, q, r\}, \{0, 1\}, \delta, p, \{r\})$, podemos considerar la tabla de transiciones siguiente

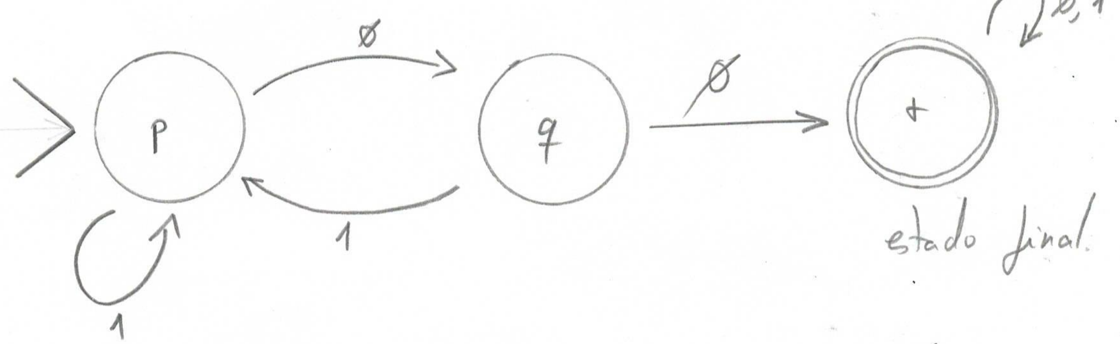
| | $\delta$ | $\emptyset$ | $1$ | Inputs |
|---|---|---|---|---|
| Estados | $p$ | $\{q\}$ | $\{p\}$ | |
| | $q$ | $\{r\}$ | $\{p\}$ | |
| | $r$ | $\{r\}$ | $\{r\}$ | |

**Definición**: Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un FA, el grafo de transiciones de $\mathcal{A}$ es un grafo desordenado etiquetado donde:

i) los nodos del grafo están etiquetados con los nombres de los estados.

ii) el grafo tiene un arco $(p, q)$ sii $\exists a \in \Sigma \,/\, / \, q \in \delta(p, a)$. En adelante, etiquetaremos los arcos con el conj. de $a \in \Sigma \,/\, q \in \delta(p, a)$.

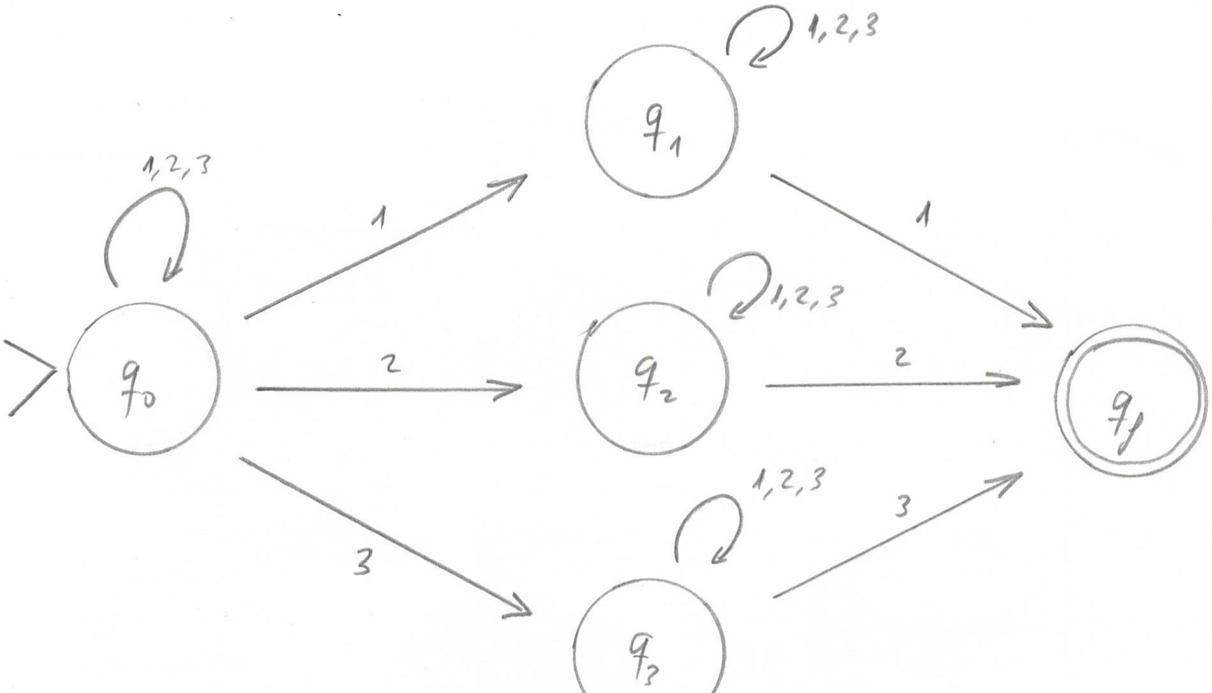
**Ejemplo**: Dado el autómata del ejemplo anterior, su grafo de transiciones vendrá dado por:




**Ejemplo**: Sea $\mathcal{A} = (\{q_0, q_1, q_2, q_3, q_f\}, \{1, 2, 3\}, \delta, q_0, \{q_f\})$ un FA, donde $\delta$ viene dada por la tabla:

| $\delta$ | 1 | 2 | 3 |
|---|---|---|---|
| $q_0$ | $\{q_0, q_1\}$ | $\{q_0, q_2\}$ | $\{q_0, q_3\}$ |
| $q_1$ | $\{q_1, q_f\}$ | $\{q_1\}$ | $\{q_1\}$ |
| $q_2$ | $\{q_2\}$ | $\{q_2, q_f\}$ | $\{q_2\}$ |
| $q_3$ | $\{q_3\}$ | $\{q_3\}$ | $\{q_3, q_f\}$ |
| $q_f$ | $\emptyset$ | $\emptyset$ | $\emptyset$ |

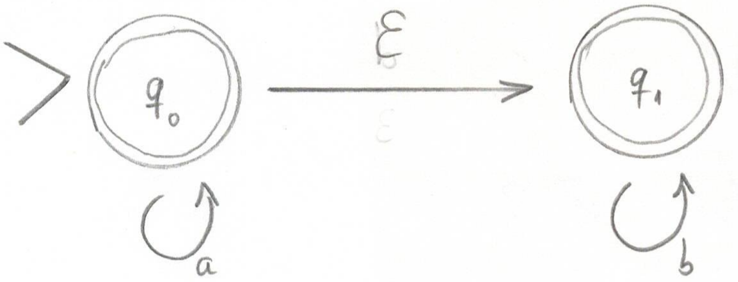
entonces su grafo de transiciones viene dado por:




## 5.1.3 Lenguajes reconocidos por FAs: conj. regulares

__Definición__: Sea $\Sigma$ un alfabeto, $L \subseteq \Sigma^*$ se dice __conj. regular__ sii $\exists\, \mathcal{A}$ un FA / $L = T(\mathcal{A})$.

__Ejemplo__: Sea $L := \{a^i b^j \,/\, i,j \geq 0\} = a^* b^*$. Veremos que es regular. Para ello bastará con construir un FA $\mathcal{A}$ / $L = T(\mathcal{A})$.



**Teorema:** El conjunto $L = \{a^i b^i \,/\, i \geq \emptyset\}$ no es regular.

demostración:

Supongamos $L$ conj. regular $\Rightarrow$: $\exists\, \mathcal{A} = (Q, \Sigma, \delta, q_0, F) \,/\, \boxed{L = T(\mathcal{A})}$.

$$\left.\begin{array}{l} \text{Consideremos el conj. } \{\delta(q_0, a^i) \,/\, i \geq \emptyset\} \subseteq Q \\ |Q| \text{ finito} \\ \exists \text{ infinitas cadenas } a^i \,/\, i \geq \emptyset \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{l} \Rightarrow \exists\, i,j,\ i>j \,/\, \delta(q_0, a^i) = \delta(q_0, a^j) \\ \text{Sea } z := b^i \end{array}\right\} \Rightarrow$$

$\Rightarrow \delta(q_0, a^j b^i) = \delta(q_0, a^i b^i)$, sea $q := \delta(q_0, a^i b^i)$ entonces:

**1º caso** $q \in F$

$$\left.\begin{array}{l} \Rightarrow: \boxed{a^j b^i \in T(\mathcal{A})} \\ i>j \Rightarrow \boxed{a^j b^i \notin L} \\ L = T(\mathcal{A}) \text{ por hipótesis} \end{array}\right\} \Rightarrow \boxed{L \neq T(\mathcal{A})}$$ absurdo demostrado

**2º caso** $q \notin F$

$$\left.\begin{array}{l} \Rightarrow: a^i b^i \notin T(\mathcal{A}) \\ a^i b^i \in L \end{array}\right\} \Rightarrow \boxed{L \neq T(\mathcal{A})}$$ absurdo demostrado

## 5.1.4 Autómatas Finitos No Deterministas: NFAs. Simulación de un NFA: método de las dos pilas.

En adelante, consideraremos que la def. dada para los FAs se refiere a los NFAs, en contraposición a los DFAs que serán introducidos más tarde.

**Teorema:** Sean $A = (Q, \Sigma, \delta, q_0, F)$ un NFA y sea $x \in \Sigma^*$, entonces podemos determinar si $x \in T(A)$ en un tiempo $O(|Q| \times |x|)$.

demo.

Bastará simular $A$ utilizando el método de las dos pilas.
Introduciremos primero el concepto de $\varepsilon$-clausura.

$\varepsilon$-clausura $(p) := \{ q \in Q \,/\, \delta(p, \varepsilon) \ni q \}$
$p \in Q$

El algoritmo es el siguiente:

```
S := ε-clausura ({q0});
c := leer_siguiente_caracter;

WHILE   c ≠ EOF  DO  BEGIN
        S := ε-clausura (δ(S,c));
        c := leer_siguiente_caracter;
END;
IF  S∩F ≠ Ø  THEN
    RETURN "yes"
ELSE RETURN "no"
```

NOTA: Dicho algoritmo puede simularse mediante dos pilas (de ahí su nombre):

1º PILA: Conjunto actual de estados.

2º PILA: " de estados siguientes.

Evidentemente, una vez hemos calculado el conj. siguiente de estados, podemos intercambiar los papeles de ambas pilas.

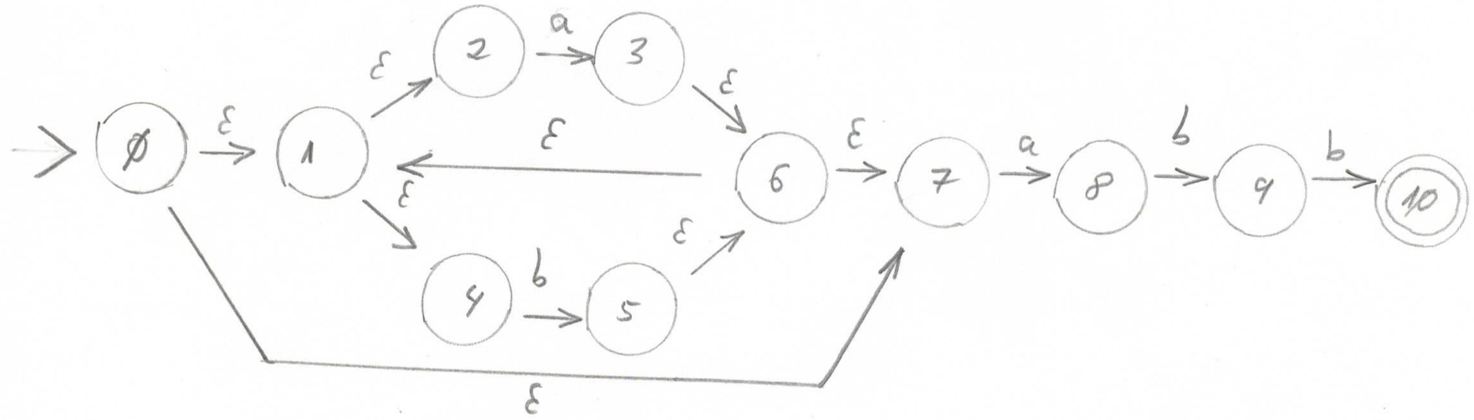
NOTA: Trivialmente dicho algoritmo simula el funcionamiento de $\mathcal{A}$.

Veamos ahora el tiempo invertido en el reconocimiento de $x \in \Sigma^*$.

Dado que al interior del bucle WHILE el número total de estados calculados es como máximo $|Q|$, tendremos trivialmente que dicho bucle invierte un tiempo $\Theta(|Q| \times |x|)$.

demostrado.

Ejemplo: Sea $\mathcal{A}$ el NFA dado por el esquema de transiciones siguiente:




para el cual $T(\mathcal{A}) = (a|b)^*abb$

Consideremos como input $x = a$, aplicando el algoritmo de las dos pilas obtendremos:

$S := \varepsilon\text{-clausura}(\{q_0\}) = \{q_0, q_1, q_2, q_4, q_7\}$

$S := \varepsilon\text{-clausura}(\delta(S, a)) = \varepsilon\text{-clausura}(\{q_3, q_8\}) =$

$= \{q_1, q_2, q_3, q_4, q_6, q_7, q_8\}$

$S \cap F = \emptyset$

RETURN "no"