8.4 Decidabilidad

8.4.1 Vaciacidad, finitud e infinitud

<u>Teorema (de Moore)</u>: El conj. de sentencias aceptadas por un FA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F) / |Q| = n$ es:

a) No vació sii $\mathcal{A}$ acepta un conj. de sentencias de longitud inferior a "n".

b) Infinito sii $\mathcal{A}$ acepta alguna sentencia de longitud l, $n \leq l < 2n$

<u>demo.</u>

a) "$\Leftarrow$" Trivial

"$\Rightarrow$" Sea $w \in L(\mathcal{A})$, tal que la long. de "w" sea la mínima para la que una sentencia es aceptada, entonces <u>$|w| < n$:</u>

Supongamos $\left.\begin{array}{l} |w| \geq n \\ |Q| = n \end{array}\right\} \Rightarrow$ (th. iteración) $\Rightarrow$

$$\Rightarrow w = xyz \Big/ \begin{array}{l} xy^k z \in L(\mathcal{A}), \forall k \geq 0 \\ y \neq \varepsilon \end{array}$$

En particular, $(k=0)$ $xz \in L(\mathcal{A})$, donde $|xz| < |w|$

<u>absurdo</u> <u>demostrado</u>

b) "$\Leftarrow$" Sea $w \in L(A) / n \leq |w| < 2n$, entonces por el th. de iteración

$$\exists\, x, y, z \in \Sigma^* \Big/ \begin{array}{l} w = xyz \\ y \neq \varepsilon \\ xy^kz \in L(A),\ \forall k \geq 0 \end{array} \Rightarrow L(A) \text{ es infinito.}$$

demostrado

"$\Rightarrow$" Por hipótesis $L(A)$ es infinito $\Rightarrow \exists\, w \in L(A) / |w| \geq n$.

1.º caso Si $|w| < 2n$ demostrado

2.º caso Si $|w| \geq 2n$ (esto es, $\nexists\, w \in L(A) / n \leq |w| \leq 2n-1$) (I)

En este caso, podemos suponer sin pérdida de generalidad que "w" es la palabra en $L(A)$ más corta, pero de longitud mayor o igual a $2n$. (II)

Aplicando el th. de iteración,

$$\exists\, x, y, z \in \Sigma^* \Big/ \begin{array}{l} w = xyz \\ y \neq \varepsilon \\ xy^kz \in L(A),\ \forall k \geq 0 \end{array} \Rightarrow (k = 0) \Rightarrow$$

$\Rightarrow xz \in L(A) / |xz| < |w|$, por tanto:

i) Si $|xz| \geq 2n$

absurdo con la hipótesis (II) sobre "w".

ii) Si $|xz| \leq 2n-1$

absurdo con la hipótesis (I) sobre $L(A)$

demostrado

Luego la $w \in L(A)$, $|w| \geq n$ y de long. mínima es tal que $|w| < 2n$ demostrado.

8.4.2 Equivalencia de dos autómatas finitos.

<u>Teorema (de Moore)</u>: Existe un algoritmo para determinar si dos FAs aceptan el mismo lenguaje.

<u>demo.</u> Sean los FAs, $\mathcal{A}_i = (Q_i, \Sigma_i, \delta_i, q_{0_i}, F_i)$, $i=1,2$ aceptando respectivamente los lenguajes $L(\mathcal{A}_1)$, $L(\mathcal{A}_2)$, entonces aplicando las propiedades de cierre, existe un FA $\mathcal{A}_3$ que acepta el lenguaje:

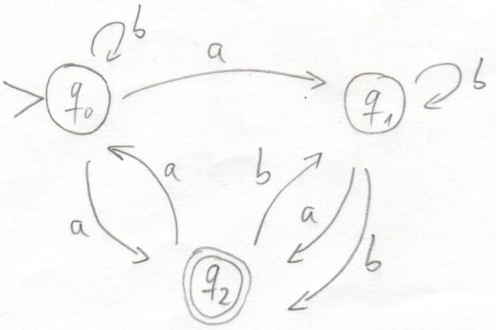
$$\left(L(\mathcal{A}_1) \cap \overline{L(\mathcal{A}_2)}\right) \cup \left(\overline{L(\mathcal{A}_1)} \cap L(\mathcal{A}_2)\right)$$

Donde trivialmente $L(\mathcal{A}_3)$ no es vacío sii $L(\mathcal{A}_1) \neq L(\mathcal{A}_2)$, luego demostrado, puesto que por el anterior th. de Moore es posible ver si $L(\mathcal{A}_3)$ es o no vacío.

<u>demostrado</u>

Ejercicio

Dado el NFA representado por el grafo adjunto, encontrar el DFA equivalente:



En nuestro caso las tablas de transición son las siguientes:

δ	a	b
q_0	q_1, q_2	q_0
q_1	q_2	q_1, q_2
q_2	q_0	q_1

Además no hay estados inaccesibles, por lo que podemos contar con todos para la construcción del DFA.

Para ello bastará con aplicar el th. de la pág. 50.

El nuevo DFA $A' = (\mathcal{P}(Q), \Sigma, \delta', \{q_0\}, \{S \subseteq Q / S \cap F \neq \emptyset\})$ donde

$$\forall S \subseteq Q, \ \delta'(S,a) = \{p \in Q / \exists q \in S, p \in \delta(q,a)\}$$

En nuestro caso:

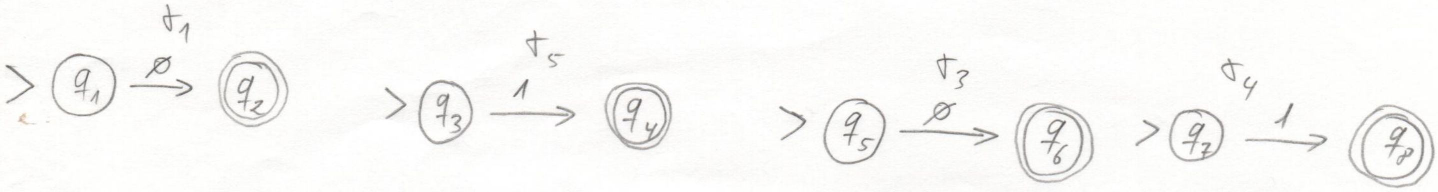
δ'	a	b
$[q_0]$	$[q_1, q_2]$	$[q_0]$
$[q_1, q_2]$	$[q_0, q_2]$	$[q_1, q_2]$
$[q_0, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$

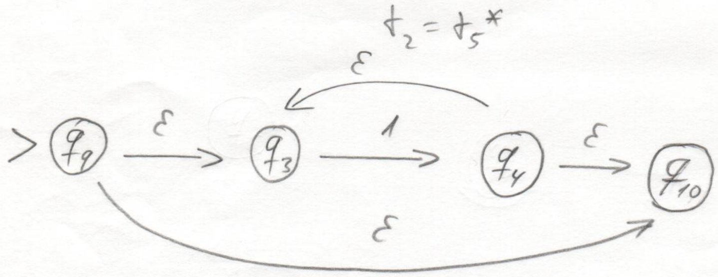
donde hemos construido el DFA, de manera a evitar la generación de estados inaccesibles.

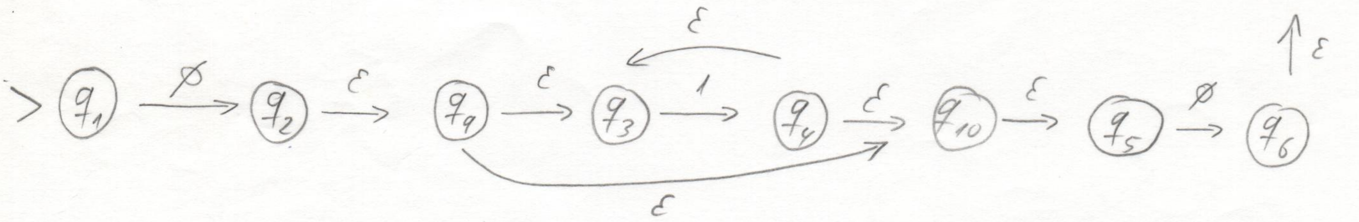
Ejercicio

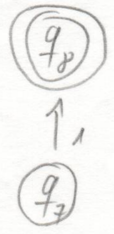
Dada la expresión regular $\emptyset 1^* \emptyset 1$ construir el AF correspondiente.

$$\emptyset 1^* \emptyset 1 = r_1 r_2 r_3 r_4 \quad \Big/ \quad \begin{array}{l} r_1 = \emptyset \\ r_2 = r_5^*\ ,\ r_5 = 1 \\ r_3 = \emptyset \\ r_4 = 1 \end{array}$$



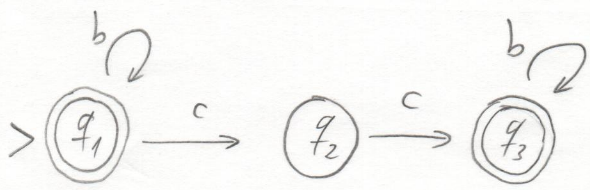






Ejercicio

Obtener la expresión regular asociada al AF dado por el grafo siguiente:



Bastará con construir las expresiones $r_{i,j}^k$ de acuerdo con la fórmula

$$r_{i,j}^k = r_{i,k}^{k-1} \left(r_{k,k}^{k-1}\right)^* r_{k,j}^{k-1} + r_{i,j}^{k-1}$$

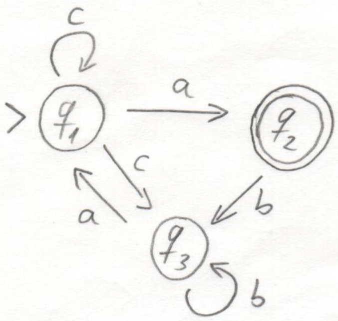
y el resultado buscado vendrá dado por

$$r_{\emptyset,j_1}^n + r_{\emptyset,j_2}^n + \ldots + r_{\emptyset,j_p}^n \quad \Big/ \quad \begin{array}{l} |Q| = n \\ \{q_{j_1}, q_{j_2}, \ldots, q_{j_p}\} = F \end{array}$$

De forma evidente el resultado es $b^* + b^* c c b^*$

Ejercicio:

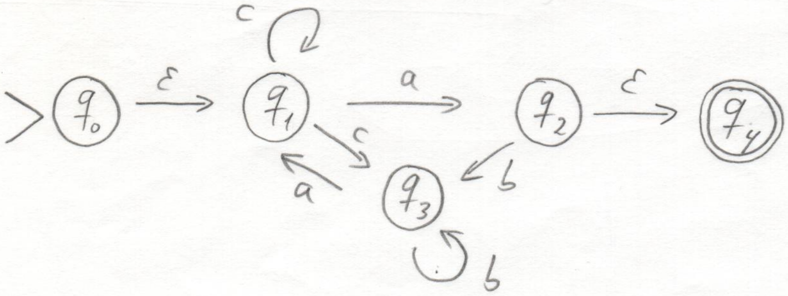
Dado el FA cuyo grafo de transiciones es el siguiente



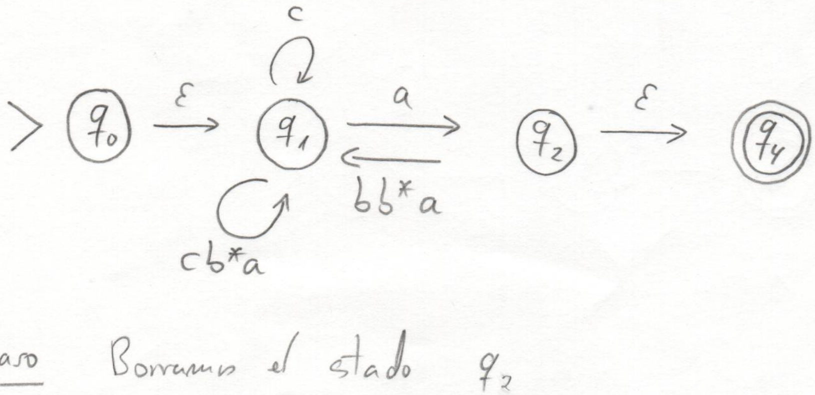
encontrar la expresión regular asociada.

Aplicaremos un método más fácil que el de las expresiones r_{ij}^k, al menos cuando el autómata es reducido.

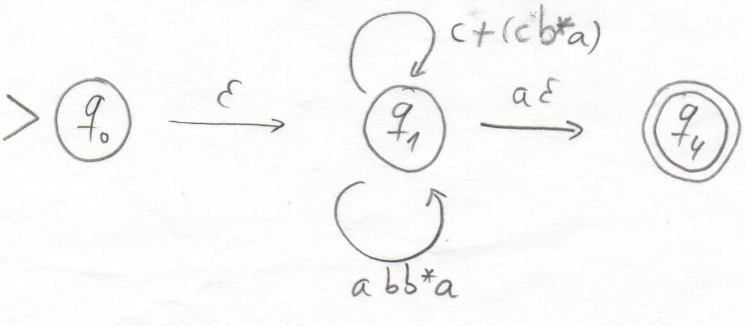
1º paso Añadimos un nuevo stado final e inicial



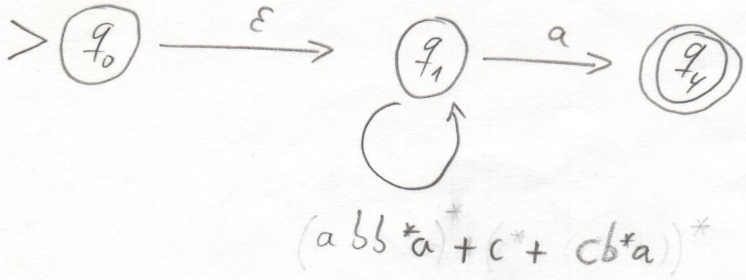
2º paso Borramos el estado q_3.



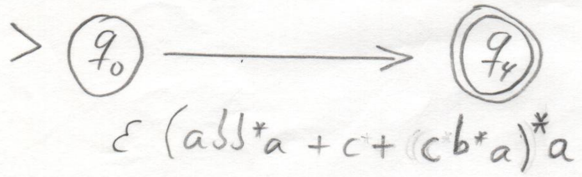
3º paso Borramos el stado q_2



equivalentemente



4.º paso Borramos el estado q_1



, esto es, la expresión regular buscada es

$$(abb^*a + c + cb^*a)^* a$$

Ejercicio: Sean L_i, $i=1,2$ conj. regulares. Ver si $L_1 \setminus L_2$ es un conj. reg.

Hay dos formas de verlo:

1.ª forma $L_1 \setminus L_2 = \underbrace{\underbrace{L_1}_{reg} \cap \underbrace{\overline{L_2}}_{reg}}_{reg}$

2.ª forma Construiremos el FA directamente:

$L_i = L(A_i), i=1,2$ donde $A_i = (Q_i, \Sigma, \delta_i, q_{0i}, F_i), i=1,2.$

que podemos suponer reducidos y deterministas.

Hemos visto (pag 102) que en esta hipótesis

$\overline{L_2} = L(\overline{A_2}) \,/\, \overline{A_2} = (Q_2, \Sigma, \delta_2, q_{02}, Q_2 \setminus F_2)$, entonces

$L_1 \cap \overline{L_2} = L(A)$ donde

$A = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times (Q_2 \setminus F_2))$ con

$\delta((q_1,q_2),a) := (\delta_1(q_1,a), \delta_2(q_2,a))$, $\forall q_i \in Q_i$, $i=1,2$, $\forall a \in \Sigma$