1) Determinismo vs no determinismo:

Cuando pensamos en automatas como mecanismos de cómputo lo que nos importa es su acción en los inputs $w \in \Sigma^*$

$$w \longrightarrow \boxed{M} \longrightarrow \begin{cases} \text{True (Acepta)} \\ \text{False} \end{cases}$$

Recordemos, para el caso no determinista qué significa

Def. Un NFA M acepta $w = w_1 \dots w_n \iff$ $\exists\, y_1, \dots, y_m \in \Sigma \cup \{\varepsilon\}$ con $w = y_1 \dots y_m$ y existe una ejecución de M que termina en un estado de aceptación con input $y_1 \dots y_m$.

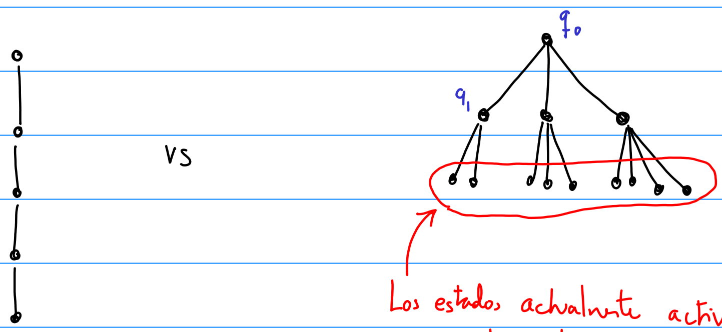
$L(M) = \{w \in \Sigma^* : M \text{ acepta a } w\}$

Def: Dos autómatas M_1 y M_2 son equivalentes $\iff L(M_1) = L(M_2)$

Teorema: Para todo NFA M existe un DFA M' con $L(M) = L(M')$.

[Es decir, el no determinismo no aporta nada para automatas finitos]

Idea:



Los estados actualmente activos son un subconjunto de todos los estados posibles. Queremos describir como evolucionan los estados activos?

Dem:

Más concretamente dado

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

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

Asumiendo que M no tiene ε - transitions

$Q' = \mathcal{P}(Q)$ ← subconjuntos de Q $(2^{|Q|})$

$$\delta'(\underbrace{R}_{\text{estados actualmente activos}}, a) = \underbrace{\bigcup_{r \in R} \delta(r,a)}_{\text{lugares a los que podemos llegar.}}$$

← $\emptyset$ queda en el $\emptyset$ así que "termina el cálculo con rechazo".

$$q_0' = \{q_0\}$$

$$F' = \{ R \in Q' : R \cap F \neq \emptyset \}.$$

Ahora le introducimos las ε- transitions :

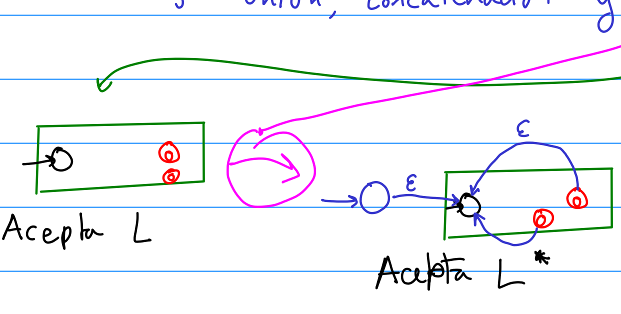
$E(R) = \{ q :$ q puede alcanzarse desde algun elemento de R usando 0 ó más ε- transitions $\}$ ~ se calcula en M.
(y tiene implícito cuantos ε's)

$$\delta'(R,a) = \bigcup_{r \in R} E(\delta(r,a))$$

El autómata M' es determinista y acepta exáctamente el mismo lenguaje que M.

Recuerda que un lenguaje $L \subseteq \Sigma^*$ se dice regular si $L = L(M)$ para algún DFA (ó equivalentemente NFA).
por el Teorema anterior

Teorema: El conjunto de lenguajes regulares está cerrado bajo unión, concatenación y estrella.



Def: R es una expresión regular sobre Σ si R cumple

(1) $R = a$ para $a \in \Sigma$ ó

(2) $R = \varepsilon$ ó

(3) $R = \emptyset$ ó

(4) $R = (R_1 \cup R_2)$ $\quad R_1, R_2$ expr regular ó

(5) $R = (R_1 \circ R_2)$ $\quad$ " " " ó

(6) $R = (R_1)^*$ $\quad R_1$ expr regular

es una expresión regular.

Ejemplo: $\underbrace{0 \circ (((10)^* \circ 1) \circ \cancel{\Sigma})}$ $= R$

Cada expresión regular en Σ describe un lenguaje. $L \subseteq \Sigma^*$

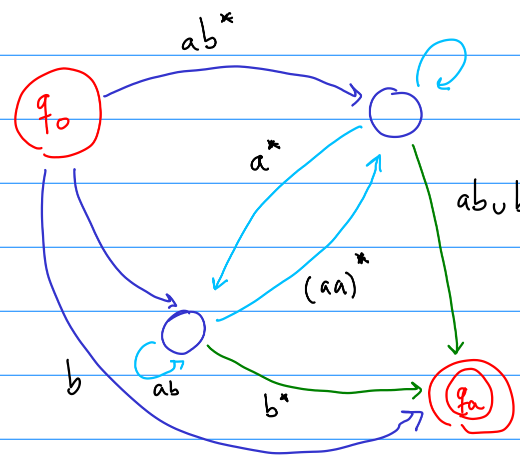
Teorema: Un lenguaje $L \subseteq \Sigma^*$ es regular $\iff$ existe una expresión regular que lo describe.

Dem: "$\Leftarrow$" Inducción en la longitud de la expresión regular que consideramos.

(Constructiva $(ab \cup a)^*$)

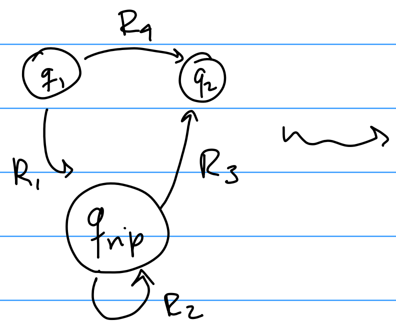
"$\Rightarrow$" Más difícil...

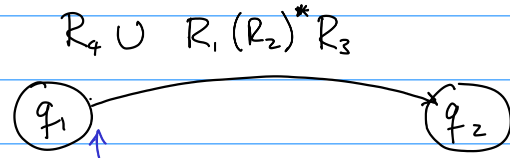
Def:



Avanza de un estado al otro leyendo un trozo de la palabra que está en el lenguaje generado por su expresión regular.

Podemos reducir el número de estados quitando cualquiera (salvo el inicial y el final)





todos los caminos en la máquina anterior, ya sea directamente ó a través de q_{rip}.

¿Cómo formalizar un GNFA?