(1) GNFAs [Automatas Finitos No-deterministas Generalizados]

Def: Un GNFA es una 5-tupla $(Q, \Sigma, \eta, q_0, q_F)$ donde

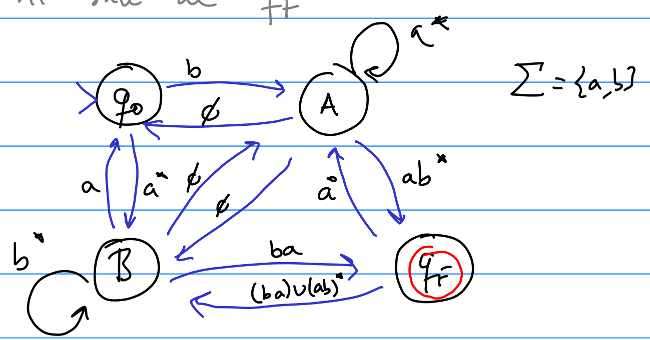
Q — conj. finito de estados.
Σ — alfabeto finito
$q_0 \in Q$ — estado inicial
$q_F \in Q$ — estado final de aceptación

Lo nuevo es la "función de transición" que es muy diferente:

$$\eta : Q \setminus \{q_F\} \times Q \setminus \{q_0\} \longrightarrow \mathcal{R}$$

conjunto de expresiones regulares en Σ (ver clase anterior)

En un dibujo tenemos finitos estados, incluyendo uno inicial y uno final. Flechas entre todos los pares con una expresión regular en cada flecha. Ninguna flecha entra a q_0 ni sale de q_F



Def: Un GNFA M acepta $w \in \Sigma^*$ si existen palabras $y_1, \dots, y_k \in \Sigma^*$ con $w = y_1 \cdot \ldots \cdot y_k$ y una sucesión de estados $q^{(0)}, q^{(1)}, \dots, q^{(k)} \in Q$ con:

(1) $q^{(0)} = q_0$

(2) Para cada $i \geq 1$ tenemos:

$$w_i \in \eta(q^{(i-1)}, q^{(i)})$$ [pertenece al lenguaje regular descrito por la expresión regular $\eta(q^{i-1}, q^{i})$]

(3) $q^{(k)} = q_F$

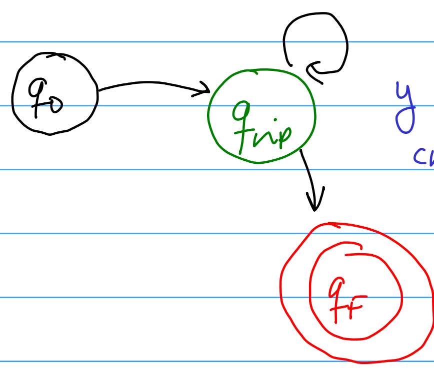
Teorema: El siguiente algoritmo está bien definido y transforma un GNFA M en una expresión regular R que describe el lenguaje aceptado por M.

Algoritmo: INPUT $M = (Q, \Sigma, \eta, q_0, q_F)$

OUTPUT R

Si $|Q| = 2$ $\quad q_0 \xrightarrow{R} q_F \quad$ y M acepta $w \iff w \in R$.

Si $|Q| \geq 3$ escogemos un estado cualquiera $q_{rip} \notin \{q_0, q_F\}$



y queremos eliminarlo, creando un nuevo NFA

$$M' = (Q', \Sigma, \eta', q_0', q_F')$$

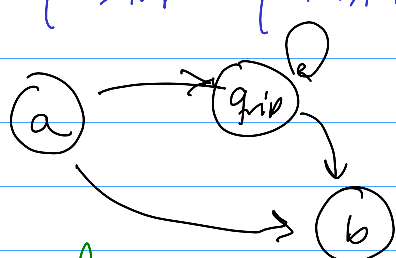
Con:

$$Q' = Q \setminus \{q_{rip}\}$$

$$\eta'(a,b) = \{\eta(a,b) \cup \eta(a, q_{rip}) \circ \eta(q_{rip}, q_{rip})^* \circ \eta(q_{rip}, b)\}$$

$q_0' = q_0$

$q_F' = q_F$



maneras de llegar desde a hasta b.

Lema: $L(Q) = L(Q')$

Dem: $w \in L(Q) \Rightarrow \exists\ w_1, \ldots, w_k$ con

$w = w_1 \cdots w_k$ y $q^{(0)}, \ldots, q^{(k)}$ con

$w_i \in \eta(q^{i-1}, q^{i})$

$(w_1\ w_2\ w_3)\ w_4\ w_5\ (w_6\ w_7\ w_8)$

frip frip · frip frip

A C D a b

Busquemos el primer frip

$$w_j \in \eta(q^{(j-1)}, q^{(j)})$$
$$w_{j+1} \in \eta(q^{(j)}, q^{(j+1)}) \leftarrow \text{frip}$$
$$\vdots$$
$$w_{j+s} \in \eta(q^{(j+s-1)}, q^{(j+s)})$$

ninguno de los dos es frip.

$$\Rightarrow \underbrace{\left(w_j (w_{j+1} \cdots w_{j+s-1}) w_{j+s}\right)}_{\text{asociamos y la ejecución en } M' \text{ nos deja pasar de } q^{(j-1)} \text{ a } q^{(j+s)}} \in \eta'(q^{(j-1)}, q^{(j+s)})$$

Recíprocamente utilizo la expresión regular de aceptación de $w \in L(M')$

$$w_j \in \eta'(a, q_{frip}) \circ [\eta(q_{frip}, q_{frip})]^* \eta(q_{frip}, b)$$

para saber que w_j se puede partir en trozos

$$w_j = a_1^{(j)} a_2^{(j)} a_3^{(j)} \cdots a_s^{(j)} a_{s+1}^{(j)}$$

que nos dan una garantía de aceptación en la máquina original M.

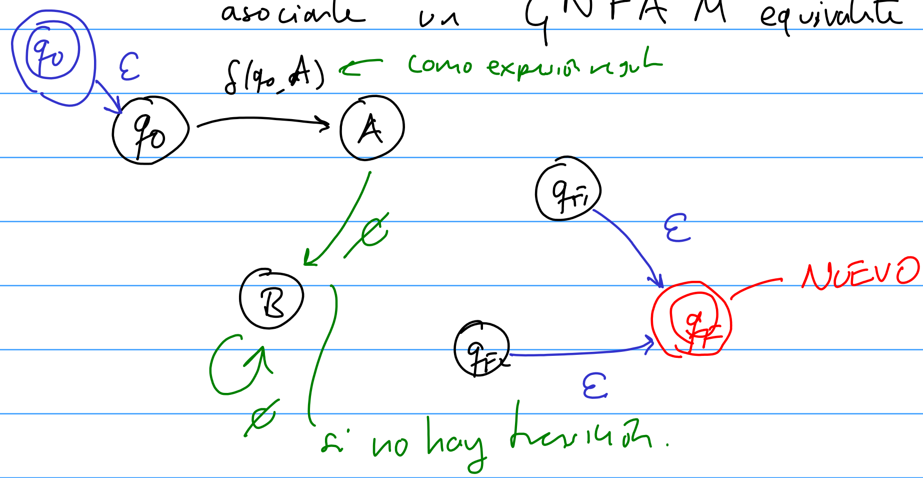
ó uso w_j misma si

$w_j \in \eta(a, b)$ (caso fácil).

Teorema: Todo lenguaje regular admite una expresión regular.

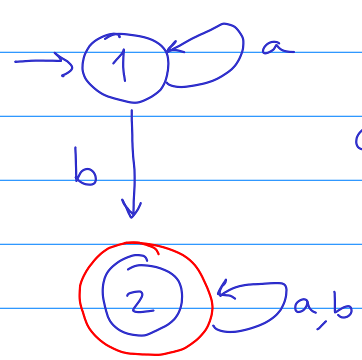
Si L es regular existe M DFA con $L=L(M)$

Dem: Si M es un DFA podemos asociarle un GNFA $\tilde{M}$ equivalente así:



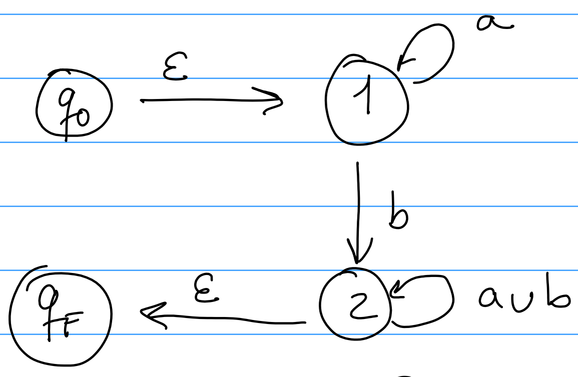
con $L(M)=L(\tilde{M})$. Por el Teorema anterior $L(\tilde{M})$ se puede describir mediante una expresión regular.

Ejemplo: $\Sigma=\{a,b\}$

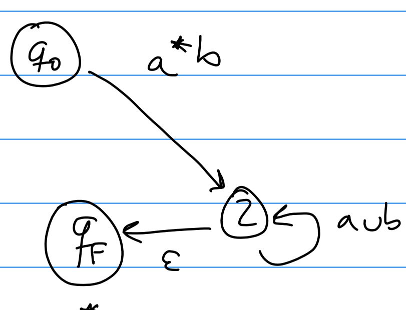


¿Cómo es R?

I.



$q_{rip}=1$



Concluimos que $R = a^*b(a \cup b)^*$

Pumping Lemma:

Qué particularidades tienen los lenguajes regulares dentro de la colección de todos los lenguajes? La clave es la FINITUD de los estados, que tiene consecuencias drásticas...

Pumping Lemma: Si $A \subseteq \Sigma^*$ es un lenguaje regular existe un entero $p \in \mathbb{N}$ (pumping length) con la siguiente propiedad:

"Si $s \in A$ y $|s| \geq p$ entonces podemos dividir s en tres partes $s = xyz$ con:

(1) $|y| > 0$

(2) $|xy| \leq p$

(3) Para cada $i > 0$ $\quad x\underbrace{\overbrace{yyyy\ldots y}^{i}}\,z \in A$

Dice algo solo para lenguajes que contengan infinitas palabras. (d.l.c. $p > \max_{w \in L} |w|$ y no dice nada)

podemos copiar el "centro" tantas veces como queramos de palabras largas.

Ejemplo: En $\Sigma = \{0,1\}$ demuestre que el lenguaje $\{0^n \cdot 1^n : n \in \mathbb{N}\}$ NO ES regular.

Sol. Si fuera regular habría un pumping length finito p. Se sigue que

Como $w = 0^p 1^p \in A$ podíamos escribir $w = xyz$ con $|xy| \leq p$, $|y| > 0$ así que y consiste de una cantidad no nula de ceros

$x = 0^r$, $y = 0^t$, $z = 0^a 1^p$ con $t > 0$, $r+t+a = p$

Luego para todo j

$xy^jz \in A$ lo cual es imposible porque xy^jz tiene más ceros que unos!

¿Porque ocurre el pumping lemma?

$$\overbrace{w_1}^{x} \Big| \overbrace{w_2 \quad w_3}^{y} \Big| \overbrace{w_4 \quad w_p \; \dots}^{z}$$

$$\underbrace{q_0 \quad q^{(1)} \quad q^{(2)} \; \dots \; q^{(j)} \quad q^{(p)}}_{\text{hay alguna repetición}}$$

$$q^{(1)} = a, \qquad q^{(j)} = a$$

Porque la sucesión $q^{(0)}, \dots, q^{(p)}$ tiene $p+1$ elementos y hay sólo p estados distintos

Si copio la y voy del estado a al estado a otra vez porque el automata es determinista y recibe los mismos inputs... así que el estado final de

xyz

xy^2z

xy^3z

$\vdots$

xy^jz siempre es de aceptación

Ejemplo: $F = \{ ww : w \in \Sigma^* \}$ palabras dobles

Sol: Considere $(0^p 1)(0^p 1) = w \in F$ $\forall p$, Por pumping Lemma $w = xyz$ con $|xy| \le p$, $|y| > 0$ fuerza y a ser un conjunto de ceros de longitud positiva

$$(0^a 0^j 1)(0^p 1) \notin F \text{ para } j \gg 0$$

así que F no es regular.

Ejemplo: $F = \{ 1^{n^2} : n \in \mathbb{N} \}$, no es regular pues el gap de longitudes crece cuadráticamente.