Queremos estimar la complejidad de algunos de los algoritmos que hemos visto.

(1) Para todo NFA existe un DFA equivalente.

Recordemos la construcción: Si

$M = (Q, \Sigma, \eta, q_0, F)$ es un NFA

diferencia principal entre DFA y NFA

$\eta : Q \times (\Sigma \cup \{\varepsilon\}) \longrightarrow$ Subconjuntos de Q

$(q, a) \longmapsto$ "Estados a los que es posible llegar a partir de q leyendo símbolo $a \in \Sigma \cup \{\varepsilon\}$."

Otra forma de verlo es pensar que tenemos una relación de transición

$$\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$$

(q, a, r) transiciones admisibles

[Equivalentes si $(q, a, r) \in \Delta \iff r \in \eta(q, a)$]

definimos un DFA así:

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

$Q' = 2^Q$ (subconjuntos de estados de M)

Para $q \in Q$ definimos

Estados alcanzables desde q sin leer ningún input

$$E(q) = \{ p \in Q : (q, \varepsilon) \vdash^* (p, \varepsilon) \}$$

($E(q)$ es la clausura de q bajo la relación de transición con ε's

$\{(p, r) : (p, \varepsilon, r) \in \Delta\}$)

$q_0' := E(q_0)$

$F' := \{ A \subseteq Q : A \cap F \neq \emptyset \}$

$$\delta'(S, a) := \bigcup \{ E(p) : p \in Q \text{ y } (q, a, p) \in \Delta \text{ para algún } q \in S \}$$

$$:= \bigcup_{q \in S} \left(\bigcup_{\substack{p \in Q : \\ (q,a,p) \in \Delta}} E(p) \right)$$

lugares a los que M puede llegar desde algún $q \in S$ leyendo a y haciendo ε-transitions.

Algoritmo: (1) Pre-compute $E(p)$'s. así:

Iniciamos con $E(p) = \{p\}$

Si para algún $q \in E(p)$ $\exists\, r \notin E(p)$ con $(q, \varepsilon, r) \in \Delta$

$E(p) \leftarrow E(p) \cup \{r\}$.

Af: Esto es $O(|Q|^3)$ (Usando depth-first-search $O(\underbrace{|V|}_{|Q|} + \underbrace{|E|}_{\leq |Q|^2})$ para cada q)

(2) Dado $S \in Q'$, $a \in \Sigma$

reunimos los estados p con $(q, a, p) \in \Delta$ para $q \in S$

$$\delta'(S, a) = \bigcup_{\substack{q \in S,\ p \in Q \\ \text{con } (q,a,p) \in \Delta}} E(p) \leftarrow |S| \cdot |\Delta| \cdot |Q|$$

$\approx O(|\Delta| |Q|^2)$ chequeos

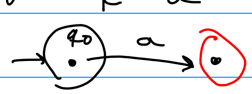
así que el cálculo de δ' requiere

$$O\left(\left(2^{|Q|} \cdot |\Sigma| \cdot |\Delta| \cdot |Q|^2\right) + |Q|^3\right)$$

(2) $\boxed{\text{Convertir una expresión regular en un autómata no determinista equivalente es un proceso eficiente.}}$

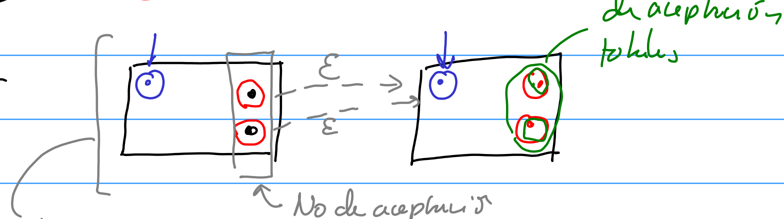
Lema: Si R es una expresión regular entonces existe un NFA que acepte el lenguaje descrito por R con $\leq 2|R|$ estados.

Dem: Si $R = a$, $a \in \Sigma$



tiene ≤ 2 estados

$R = R_1 \circ R_2$



Nuevo autómata tiene $\leq 2|R_1| + 2|R_2|$ estados

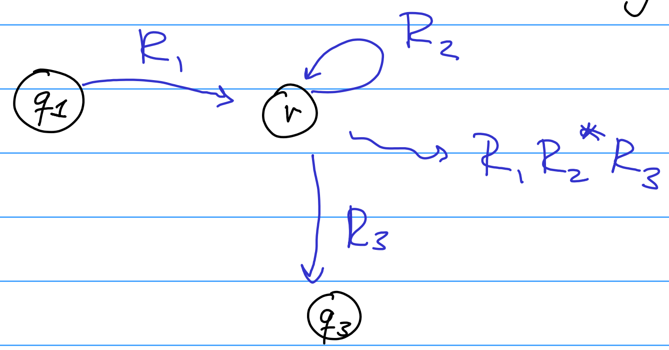
$R = R_1^*$ tiene $\leq 2|R|$ estados

$R = R_1 \cup R_2$ tiene $\leq$ estados M_1 + estados M_2 $\overset{\text{HI}}{\leq} 2|R_1| + 2|R_2|$

Se sigue que la relación de transición tiene

$$\leq (2|R|)\cdot(2|R|)\cdot|\Sigma| \text{, polinomial en la longitud.}$$

(3) Convertir un GFA en una expresión regular equivalente necesita en cada paso $j = 1, .., |Q|$ calcular las expresiones, que son $\leq j^2$ en cada paso



así que calculamos $O(|Q|^3)$ expresiones regulares. La dificultad está en que estas expresiones se hacen MUY largas, exponencial. incluso triplicándose en cada paso. Podría suceder que

$$|\text{Expresión final}| \geq O(3^{|Q|^3})$$

(4) El algoritmo de minimización de estados de la clase anterior es polinomial!

Recordemos su funcionamiento:

Sea $M = (Q, \Sigma, \delta, q_0, F)$ un DFA y queremos disminuir su número de estados.

Def: $p, q \in Q$

$p \equiv q \iff \forall w \in \Sigma^* \big($ si $(p, w) \vdash^* (f, \varepsilon)$ y $(q, w) \vdash^* (f', \varepsilon)$ entonces $\{f, f'\} \subseteq F$ ó $\{f, f'\} \subseteq Q \setminus F \big)$.

Si conocemos las clases de equivalencia de $\equiv$ construimos el autómata óptimo de Myhill-Nerode así:

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

$$Q' = \{ [p]_{\equiv} : p \in Q \}$$

$$\delta'([p]_{\equiv}, a) = [\delta(p,a)]_{\equiv}$$

$$q_0' = [q_0]_{\equiv}$$

$$F' = \{ [f] : f \in F \}.$$

Obs: $[p_1] = [p_2] \Rightarrow \forall a \in \Sigma \; \forall w \in \Sigma^* \left(\begin{array}{l} (p_1, aw) \vdash (f, \varepsilon) \\ (p_2, aw) \vdash (f', \varepsilon) \end{array} \right.$

cumple $\{f, f'\} \subseteq F$ ó $\{f, f'\} \subseteq Q \setminus F$ así que $[\delta(p_1, a)] = [\delta(p_2, a)]$. ($\delta'$ está bien def)

Lema: $\forall w$, si $(q_0, w) \vdash^j_M (r, w_{\geq j+1})$

$\Longrightarrow ([q_0]_{\equiv}, w) \vdash^j_{M'} ([r]_{\equiv}, w_{\geq j+1})$

Dem: Ind en j. demuestra la equivalencia

Así que el problema principal es el cálculo de la relación $\equiv$.

Def: $p, q \in Q$

$p \equiv_n q \iff \forall w \left(|w| \leq n \text{ y } \begin{array}{l} (p, w) \vdash^* (f, \varepsilon) \\ (q, w) \vdash^* (f', \varepsilon) \end{array} \text{ y} \right.$

entonces $\{f, f'\} \subseteq F$ ó $\{f, f'\} \subseteq Q \setminus F$ $\Big)$

Lema: (1) $p \equiv q \iff p \equiv_n q \;\; \forall n \in \mathbb{N}$

(2) $p \equiv_n q \iff p \equiv_{n-1} q$ y $\left(\begin{array}{c} \forall a \in \Sigma \\ \delta(p, a) \equiv_{n-1} \delta(q, a) \end{array} \right)$

(3) $\equiv_0$ tiene dos clases, F y $Q \setminus F$

Del lema $p \equiv q$ puede calcularse en $\leq |Q|$ pasos y cada paso requiere $O(|\Sigma|\,|Q|^2)$ comparaciones, así que en total $O(|\Sigma||Q|^3)$.

Por último, la complejidad exponencial del Algoritmo NFA $\leadsto$ DFA es NECESARIA,

Ejemplo: $\Sigma = \{a_1, \ldots, a_n\}$
$L = \{w \in \Sigma^* : \exists j \; a_j \text{ no aparece en } w\}$

(1) Hay un NFA con $n+1$ estados.

(2) El autómata de Myhill-Nerode necesita 2^n estados

Porque para $A \subseteq \Sigma$ definimos

$L_A = \{w$ que usan todos los símbolos de A y NINGUNO de $\Sigma \setminus A\}$

los L_A son las clases de equivalencia de $\underset{L}{\approx}$ que son los estados del autómata óptimo (en cantidad 2^n!!)