$r^+ = r(r)^*$

ESEMPIO

Se l'espressione regolare $(a+b)^* a$ questa rappresenta il linguaggio:

$$\mathcal{L}\left((a+b)^* \cdot a\right) = \mathcal{L}\left((a+b)^*\right) \circ \mathcal{L}(a) = \left(\mathcal{L}(a+b)\right)^* \circ \mathcal{L}(a) =$$

$$= \left(\mathcal{L}(a) \cup \mathcal{L}(b)\right)^* \circ \mathcal{L}(a) = \left(\{a\} \cup \{b\}\right)^* \circ \{a\} =$$

$$= \left(\{a,b\}\right)^* \circ \{a\} = \{a,b\}^* \circ \{a\} = \{x \mid x \in \{a,b\}^+, x \text{ termina con } a\}$$

$x \in$ a tutte le stringhe che posso fare con a, b

Determinare l'espressione regolare che sull'alfabeto $\{a,b\}$ definisce l'insieme delle stringhe il cui terzultimo carattere è b.
(penultimo)

$\{a,b\}^* \circ \{b\} \circ \{a,b\}^1$ → PENULTIMO oppure $\{a,b\}^* \circ \{b\} \circ (a+b)$

$\{a,b\}^* \circ \{b\} \circ \{a,b\}^2$ → TERZULTIMO

b secondo carattere $\qquad\qquad$ b primo carattere

$$(a+b)\circ\{b\}\circ\{a+b\}^* \qquad\qquad \{b\}\circ\{a+b\}^*$$

non ci interessa cosa ci sia dopo la richiesta quindi metto "cose o caso"

$$C=\{0,1\dots 9\} \rightarrow \underbrace{(1+2+\dots+9)}_{\substack{\uparrow\\ 9 \text{ elementi da combinare}}}(0+1+\dots+9)^*+0,(0+1+\dots+9)^+$$

espressione regolare

ESERCIZIO

Scrivere il linguaggio definito dall'espressione regolare

$$a^*((aa)^*b+(bb)^*a)b^*$$

$$=\mathcal{L}\left(a^*((aa)^*b+(bb)^*a)b^*\right)=$$

$$=\mathcal{L}(\{a\}^*\circ(\{aa\}^*\circ\{b\}+\{bb\}^*\circ\{a\})\circ\{b\}^*)$$

$$=\mathcal{L}(\{a\}^*\circ\{\{aa\}^*\circ\{b\}\cup\{bb\}^*\circ\{a\}\}\circ\{b\}^*)=$$

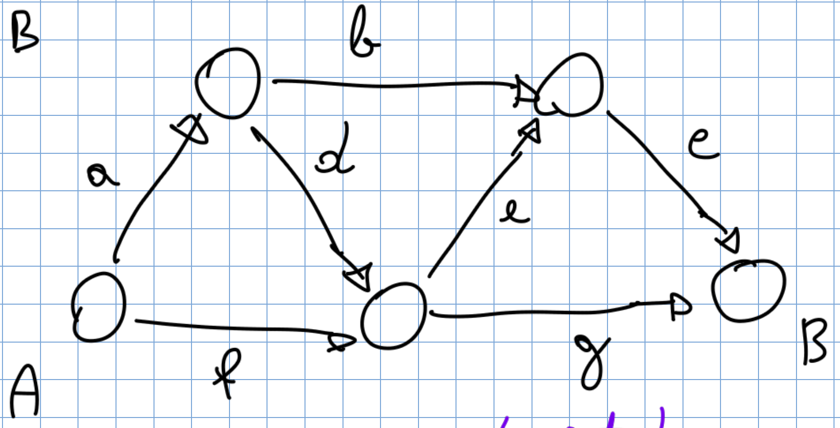
$$=\{x\mid x\in\{\{aa\}^*\circ\{b\}\cup\{bb\}^*\circ\{a\}\}\mid x \text{ inizia con } a$$

e termina con b $\}$

ESERCIZIO

DESCRIVERE UNA MAPPA STRADALE CON TUTTI I TRATTI A SENSO UNICO E CONTRASSEGNATI DAI CARATTERI DELL'ALFABETO

FORNIRE UN'ESPRESSIONE REGOLARE CHE COLLEGHI TUTTI PERCORSI TRA A e B



da intendere come un loop

$$(ad + f) \circ (ebd)^* \circ (ec + g) = \mathcal{L}(ad) \cup \mathcal{L}(f) \ldots\ldots$$

DEFINIZIONE

Dato un alfabeto $\Sigma = \{a_1, \ldots\ldots a_n\}$ si definisce ordinamento lessico-grafico delle stringhe di Σ^* l'ordinamento ($<$) ottenuto stabilendo un (qualunque) ordinamento tra i caratteri di Σ e definendo l'ordinamento di 2 stringhe $x, y \in \Sigma^*$ in modo tale che $x < y \iff$ una delle seguenti condizioni è verificata

simbolo

1) $|x| < |y|$

2) $|x| = |y|$ ed esiste $z \in \Sigma^*$ tale che $x = z\, a_i\, u$ e $y = z\, a_j\, v$ con $u, v \in \Sigma^*$ e $i < j$

ESEMPIO

$\Sigma = \{a, b\}$ dove $a < b$, le stringhe di Σ^* sono enumerate:

$\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, aba, abb \ldots\ldots\ldots$

LINGUAGGIO REGOLARE

Un linguaggio si dice regolare se esiste un ASF che lo riconosca

PROPRIETÀ DI CHIUSURA DEI LINGUAGGI REGOLARI

TEOREMA

Dati 2 linguaggi regolari L_1, L_2 la loro unione $L_1 \cup L_2$ è un linguaggio regolare

DIMOSTRAZIONE

siano dati 2 ASFD $A_1 = \langle \Sigma_1, Q_1, q_{0_1}, \delta_1, F_1 \rangle$

$A_2 = \langle \Sigma_2, Q_2, q_{0_2}, \delta_2, F_2 \rangle$

che accettano i linguaggi $L_1 = L(A_1)$ $L_2 = L(A_2)$ vogliamo dimostrare che partendo da L_1 e L_2 e quindi da A_1 e A_2 costruiamo l'automa $A = < \Sigma, Q, q_0, \Delta, F >$ che riconosca $L(A_1) \cup L(A_2)$

$L = L(A_1) \cup L(A_2)$

$\Sigma = \Sigma_1 \cup \Sigma_2$

$Q = Q_1 \cup Q_2 \cup \{q_0\}$

$F = F_1 \cup F_2$ oppure $F = F_1 \cup F_2 \cup \{q_0\}$
il caso in cui i 2 automi riconoscono anche la stringa vuota.

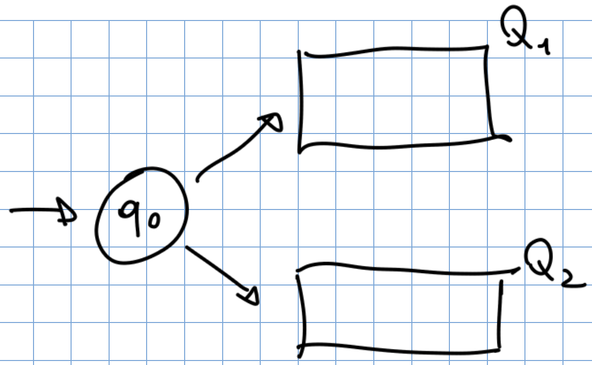
$\Delta(q, a) = \Delta_1(q, a)$ se $q \in Q_1$ e $a \in \Sigma_1$

$\Delta(q, a) = \Delta_2(q, a)$ se $q \in Q_2$ e $a \in \Sigma_2$

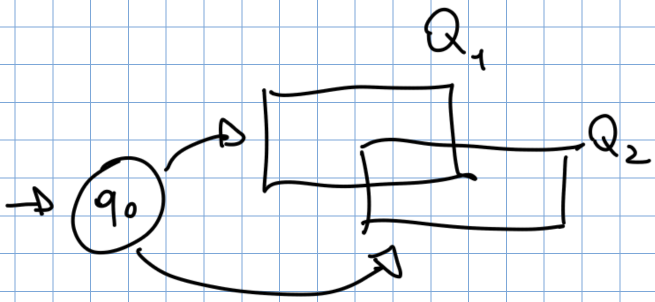
$\Delta(q, a) = \Delta_1(q_0, a) \cup \Delta_2(q_0, a)$ $a \in \Sigma = \Sigma_1 \cup \Sigma_2$

non implica

A_1, A_2 DETERMINISTICI $\not\Rightarrow$ A SIA DETERMINISTICO



in questa configurazione stato finale non in comune



questo è una configurazione con stato finale in comune

COMPLEMENTO → $\overline{L} = \Sigma^* / \{L\}$

TEOREMA

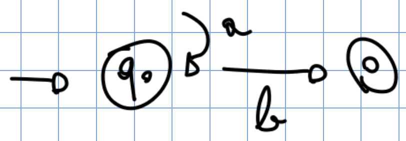
Dato un linguaggio regolare L il suo $\overline{L}$ è un linguaggio regolare

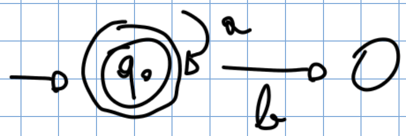
DIMOSTRAZIONE

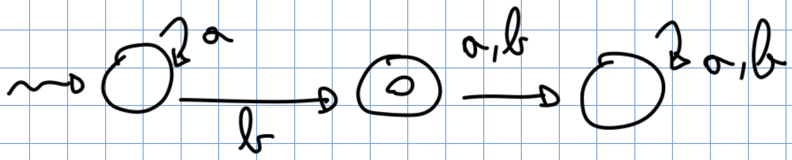
$A = < \Sigma, Q, q_0, \Delta, F >$

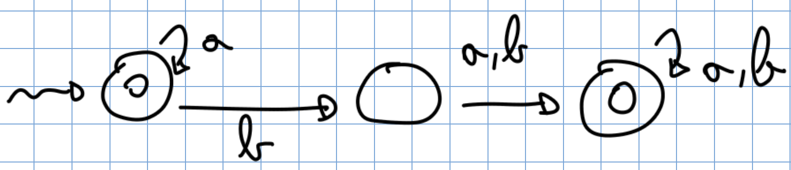
→ definire un linguaggio riconosciuto $L = L(A)$
e possiamo costruire $\overline{A} = < \Sigma, Q, q_0, \Delta, \{Q-F\} >$
che riconosce il linguaggio $\overline{L(A)} = \overline{L}$
ogni stringa che porta l'automa A in uno stato finale F porta l'automa $\overline{A}$ in uno stato non finale, e viceversa

$$L(\bar{A}) = \Sigma^* / L(A) = \overline{L(A)}$$

A :  $aab \in L_1$ $\quad ba \notin L_1$

$\bar{A}$:  $aab \notin L_2$ $\quad ba \notin L_2$

A_1 : 

A_2 : 

Pag 95

TEOREMA

dati 2 linguaggi regolari L_1, L_2 la loro intersezione

$L = L_1 \cap L_2$ è un linguaggio regolare

DIMOSTRAZIONE

$$L = L_1 \cap L_2 = \overline{\bar{L}_1 \cup \bar{L}_2}$$

L_1 è reg. $\Rightarrow$ L_2 è reg

$\bar{L}_1$ è reg. $\Rightarrow$ $\bar{L}_2$ è reg $\quad \Rightarrow \bar{L}_1 \cup \bar{L}_2$ è reg. $\Rightarrow \overline{L_1 \cup L_2}$ è reg.

TEOREMA

Dati 2 linguaggi regolari L_1, L_2 la loro concatenazione è un linguaggio regolare.

DIMOSTRAZIONE

ASFD $A_1 = < \Sigma_1, Q_1, q_{0_1}, \Delta_1, F_1 >$ riconoscono: $L_1 = L(A_1)$

$A_2 = < \Sigma_2, Q_2, q_{0_2}, \Delta_2, F_2 >$ $L_2 = L(A_2)$

Sia $A = < \Sigma, Q, q_0, \Delta, F >$ NON DET.:

- $\Sigma = \Sigma_1 \cup \Sigma_2$
- $Q = Q_1 \cup Q_2$
- $F = \begin{cases} F_2 & \text{se } \varepsilon \notin L(A_2) \\ F_1 \cup F_2 & \text{altrimenti} \end{cases}$
- $q_0 = q_{0_1}$
- $\Delta(q, a) = \Delta_1(q, a) \ \forall q \in Q_1 - F_1 \ a \in \Sigma_1$
- $\Delta(q, a) = \Delta_2(q, a) \ \forall q \in Q_2 - F_1 \ a \in \Sigma_2$
- $\Delta(q, a) = \Delta_1(q, a) \cup \Delta_2(q, a)$

  $\forall q \in Q_1$

  $a \in \Sigma = \Sigma_1 \cup \Sigma_2$

TEOREMA

Dato un linguaggio regolare L, anche L^* è un linguaggio regolare

DIMOSTRAZIONE

ASFD $A = < \Sigma, Q, \delta, q_0, F >$ che riconosce $L(A) = L$

a partire da questo costruiamo

$A' = < \Sigma, Q \cup \{q_0'\}, \delta', q_0', F \cup \{q_0'\} >$

che riconosce $L^* = L(L(A))^*$ ponendo

$\delta'(q, a) = \delta(q, a) \quad \forall q \in Q - F \quad a \in \Sigma$

$\delta'(q, a) = \delta(q, a) \cup \delta(q_0, a) \quad \forall q \in F \quad a \in \Sigma$

$\delta'(q, a) = \delta(q_0, a) \quad a \in \Sigma$

Il secondo automa A' riconosce stringhe in più rispetto al primo