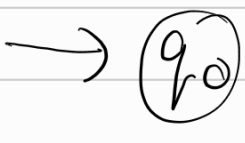
Expresii Regulate (RegEx)

Def: $RegEx(\Sigma)$ = mulțimea cuvintelor peste alfabetul $\Sigma \cup \{(,), *, \cdot, +, \phi, \lambda\}$ definită recursiv astfel: (+ reuniune, · concatenare)

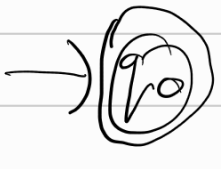
1) $\phi, \lambda \in RegEx$ și $a \in RegEx, \forall a \in \Sigma$
2) dacă $e_1, e_2 \in RegEx \Rightarrow (e_1 + e_2) \in RegEx$
3) dacă $e_1, e_2 \in RegEx \Rightarrow (e_1 \cdot e_2) \in RegEx$
4) dacă $e_1 \in RegEx \Rightarrow (e_1)^* \in RegEx$

Transformare din RegEx în AF

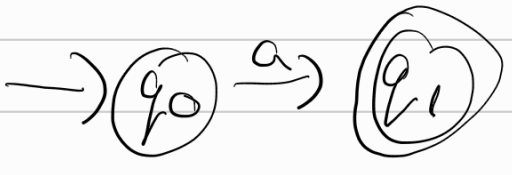
I) $e = \phi$, $L = \phi$



II) $e = \lambda$, $L = \{\lambda\}$

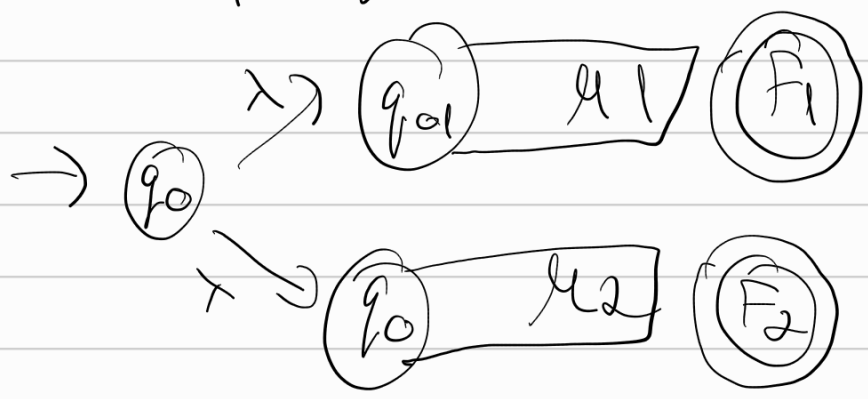


III) $e = a \in \Sigma$, $L = \{a \mid a \in \Sigma\}$



IV) $e = e_1 + e_2$

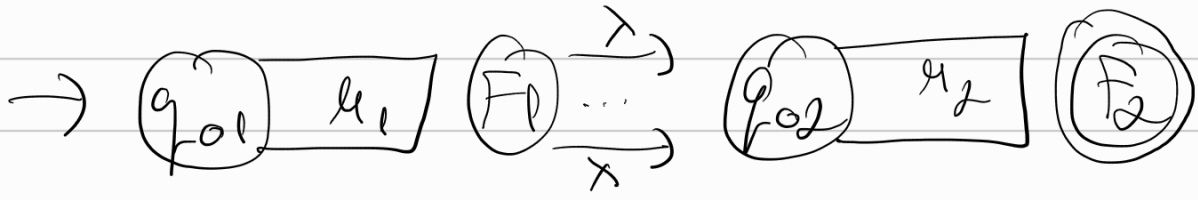
$L = L_1 \cup L_2$



$F = F_1 \cup F_2$

V) $e = e_1 \cdot e_2$

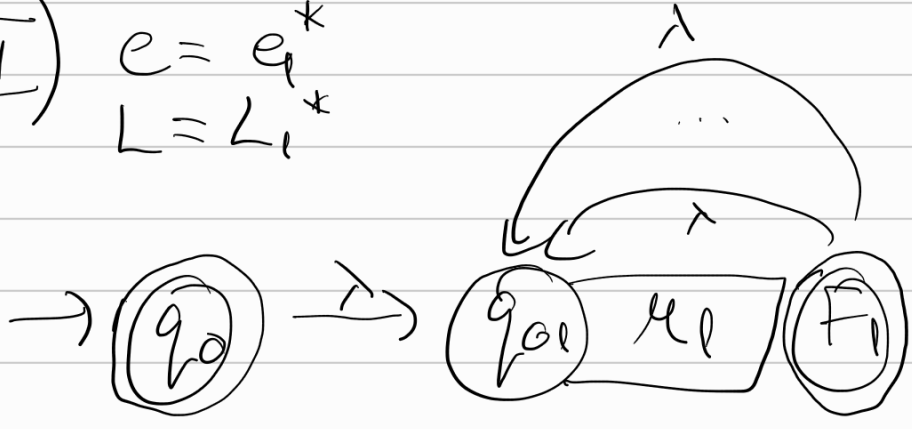
$L = L_1 \cdot L_2$



$f = F_2$

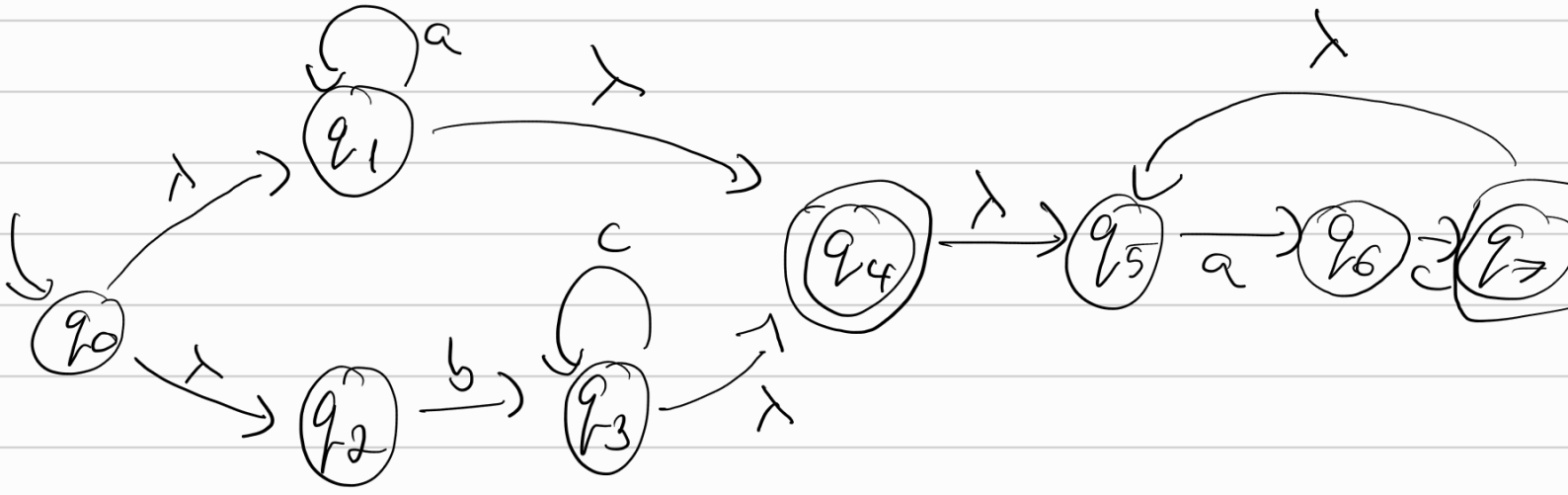
VI) $e = e_1^*$

$L = L_1^*$



$F = F_1 \cup \{q_0\}$

ex: $(a^* + bc^*) \cdot (ac)^*$



$F = \{q_4, q_7\}$

Transformare din AF în RegEx

Def: AFE (Automat Finit Extins) $= (Q, \Sigma, \delta, q_0, F)$

Funcție de etichetare $\delta: Q \times Q \to RegEx(\Sigma)$

Algoritm:

1) Transformăm AF în AFE (Reunim simbolurile aflate pe aceeași săgeată)

2) Dacă starea inițială este și finală sau dacă există vreo săgeată spre starea inițială adăugăm o nouă stare inițială din care mergem cu lambda către fosta stare inițială.

3) Dacă ∃ mai multe stări finale sau dacă ∃ vreo săgeată care pleacă din starea finală atunci adăugăm o nouă unică stare finală spre care mergem cu lambda din fostele stări finale.

4) Pe rând, în orice ordine, eliminăm câte o stare, diferită de cea inițială și cea finală, astfel: