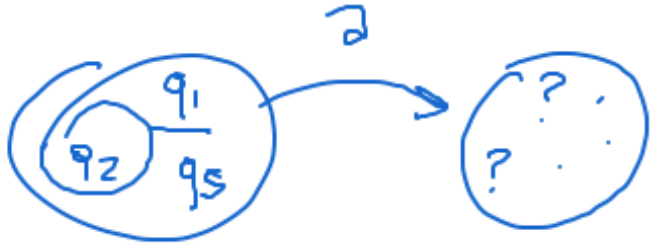
# Reg Lang (not p. lemma)

#1 NFA > DFA : subset construction

input : NFA

out : DFA

- states of DFA are sets :



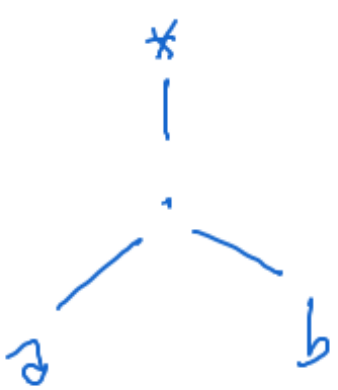

- lazy eval.

#2 : Mutual Induction :

no such exercise in Sept !

#3 : Structural Induction ( RegEx)

$(a \cdot b)^*$ ⇒




$a \cdot b^*$ ⇒

Exercise on S.I. : reversal of RegEx

$E \Rightarrow L(E)$

$\vdots$

$E^R \Rightarrow [L(E)]^R$

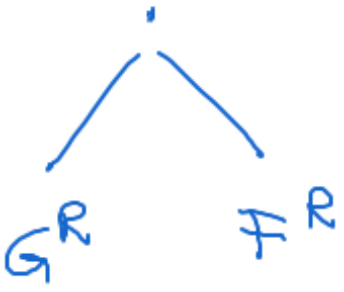
base case : $\exists \phi \in \Sigma$

ind. case : $E = F \cdot G$

$F \dashrightarrow F^R$

$G \dashrightarrow G^R$

$E^R$ :



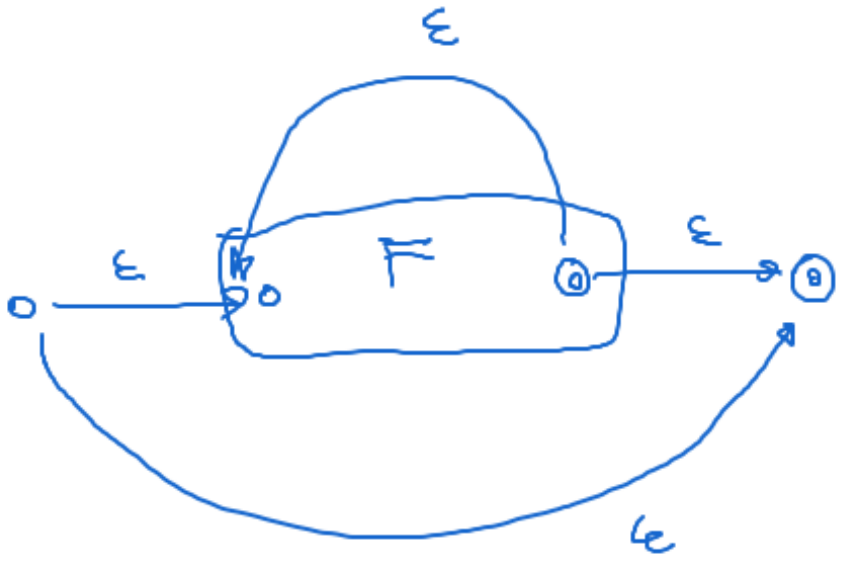
#4 RegEx > ε-NFA (case of S.J.)

base cases : $\emptyset$ ....> start → 0 (0)

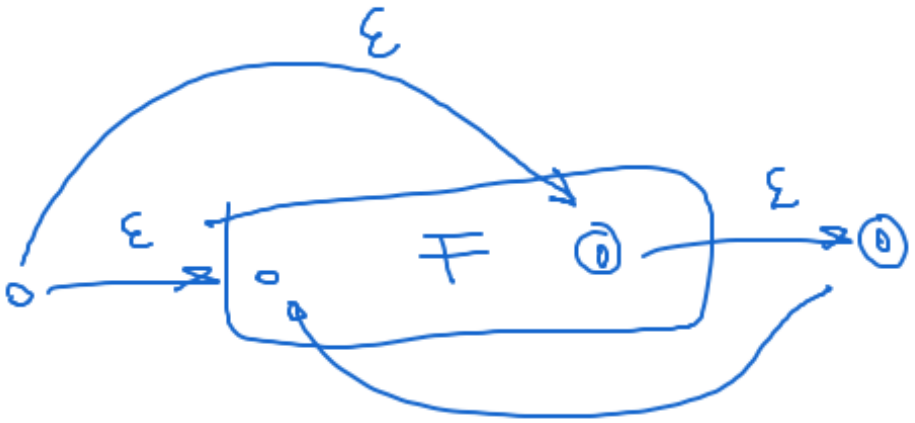
⋮

ind cases :

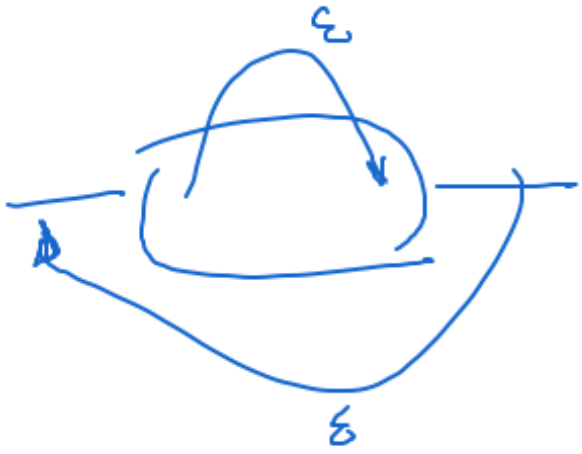
$E = F^*$ ---->



mistakes!



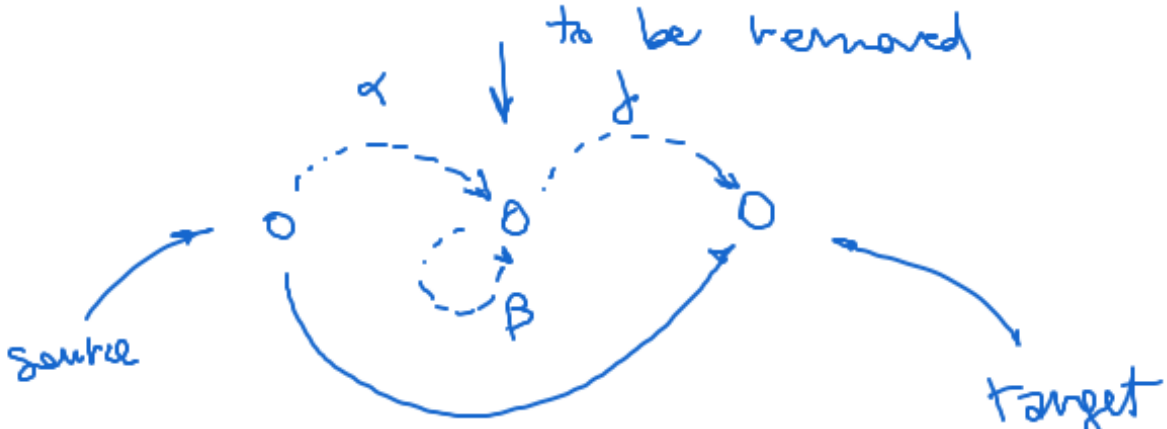

don't do this!




don't!!

#5. DFA > RegEx : state elim. construction

Basic




mistake : inspect DFA and draw RegEx based on intuition | don't !!

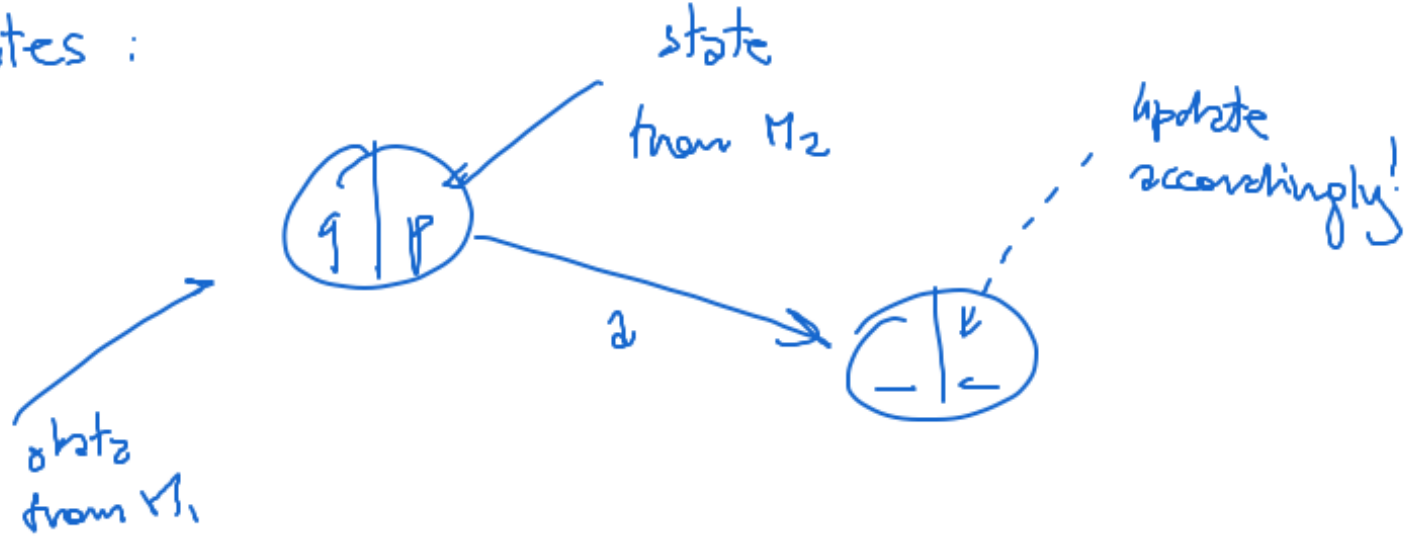
# #6. Intersection of two DFA's

input : $M_1$, $M_2$ DFA

output : $M_{int}$ DFA

simulates simultaneously $M_1$ and $M_2$

$M_{int}$'s states :



#7. Other things:

short yes/no questions

2) it ] take RegLang L,
and change something,
does it become non Reg

pay attention to the quantifiers ($\forall$, $\exists$) in the text ...

possible solution: find counter-example ....