

Jan  
21  
2025

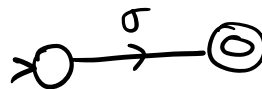
# Equivalence of r.e. and FA.

p. 66

Theorem: A language is regular iff  $\exists$  a r.e. that describes it.

Proof: ( $\Leftarrow$  i.e.  $re \rightarrow NFA$ )

1.  $\sigma, \sigma \in \Sigma$



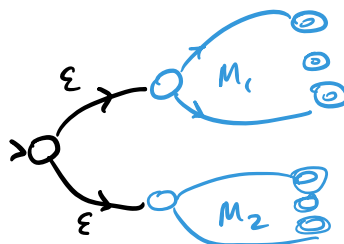
2.  $\epsilon$



3.  $\emptyset$



4.  $(R_1 + R_2)$



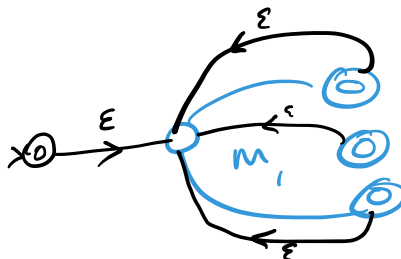
$L(M_1) = L(R_1)$

$L(M_2) = L(R_2)$

5.  $(R_1 \cdot R_2)$



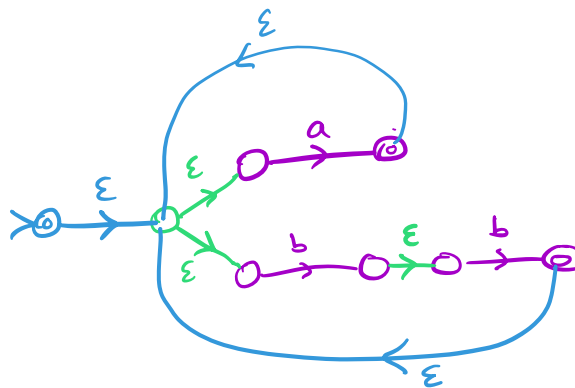
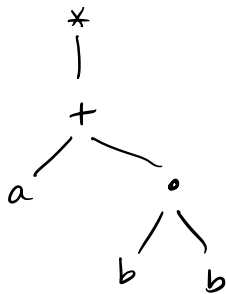
6.  $(R_i^*)$



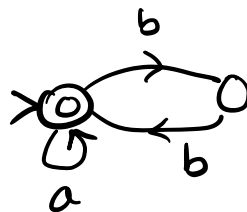
The resulting FA clearly accepts the language described by the r.e.  $\square$

Aside: Let's do an example.  
Using the construction:

$(a+bb)^*$



"Grok + Blurt" method:



can be a NFA without  $\epsilon$ -transitions

II. ( $\Rightarrow$  : FA  $\rightarrow$  r.e.) Construction

Goal: given a FA  $M$ , construct a r.e.  $R$   
such that  $L(M) = L(R)$ .

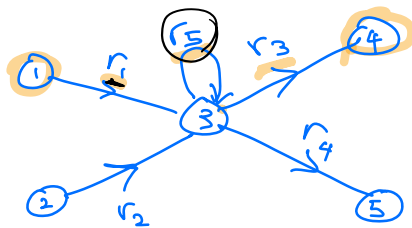
To do this, we invent something we will call  
a "generalized FA", where the transitions are  
labelled with r.e.'s.



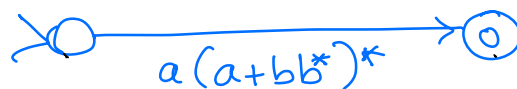
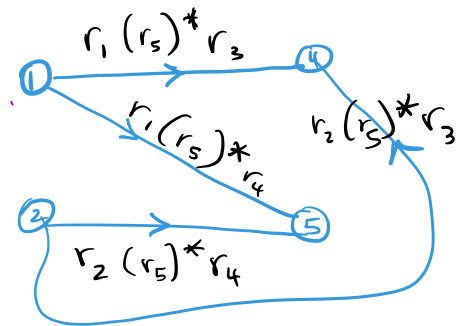
We step-by-step RIP some state and replace the affected paths with new r.e.'s.

FA -to- RE: Given a FA  $M$

1. Ensure start state has no "in" edges.  
- add a new state if necessary... while ensuring the gen-FA accepts the same language
2. Ensure  $\exists$  a single accept state w/ no "out" edges - introduce a new state if necessary... while ensuring etc.
3. While  $\exists > 2$  states:  
RIP a state that is neither start nor accept.



RIP 3  
⇒

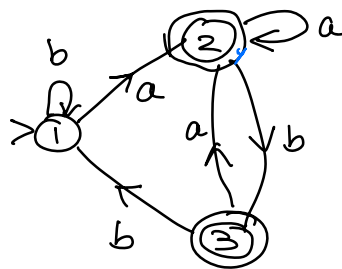


⇐ if we have just one transition, from start state to the final state, we are done.

re. is:  $a(a+bb^*)^*$

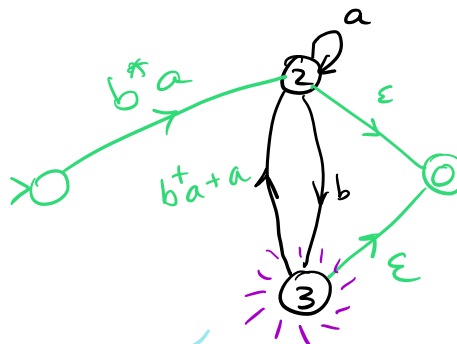
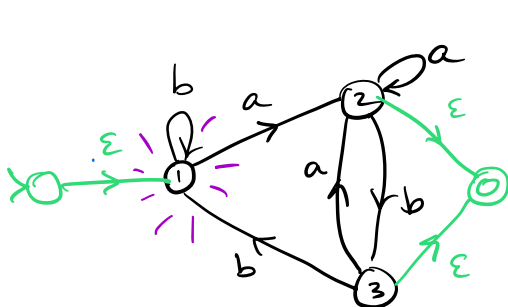


$\Sigma_f$



Grrok + Blurt:

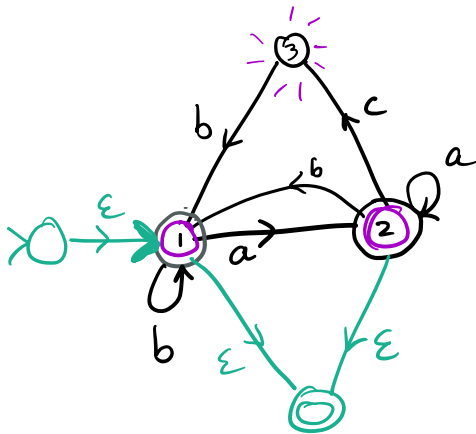
$$(b^*a)^+(b+\epsilon)$$



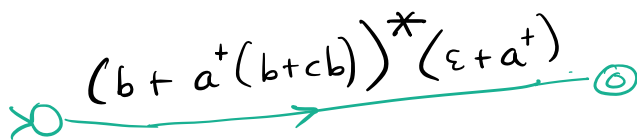
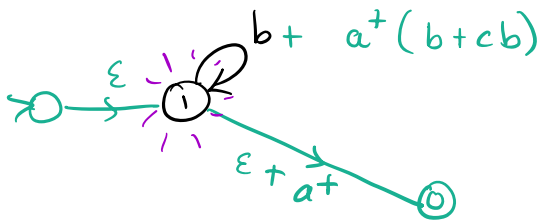
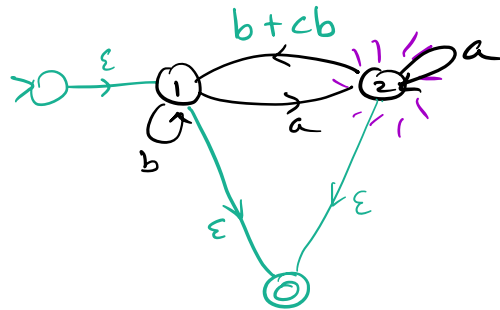
$$\text{start} \xrightarrow{b^*a(a+b(b^+a+a))^*(b+\epsilon)} \text{final}$$

r.e. is  $b^*a(a+b(b^+a+a))^*(b+\epsilon)$





$$(a + b + acb)^*$$



re is  $(b + a^+(b + cb))^* a^*$