COL 352 Lecture 10 Jan 23

Given a r.e. r, how do we construct an NFA N such that

$L(r) = L(N)$

the set of strings represented by r

We will construct an NFA with ε transitions and then show how to construct an NFA w/o ε transitions

<u>General technique</u>: To show that two classes of computing machines M_1, M_2 are equivalent

Given M_1, we will simulate M_1 by M_2 by constructing M_2 s.t.

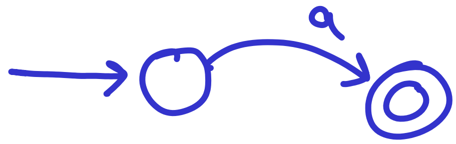
$$L(M_2) = L(M_1)$$

M_2 is at least as powerful as M_1

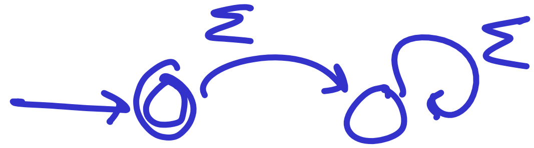
Likewise if we can construct M_1 s.t. $L(M_1) = L(M_2)$ then M_1 are M_2 are equi.

R.e.

(i) $a \in \Sigma$ is r.e.



(ii) ε is a r.e.

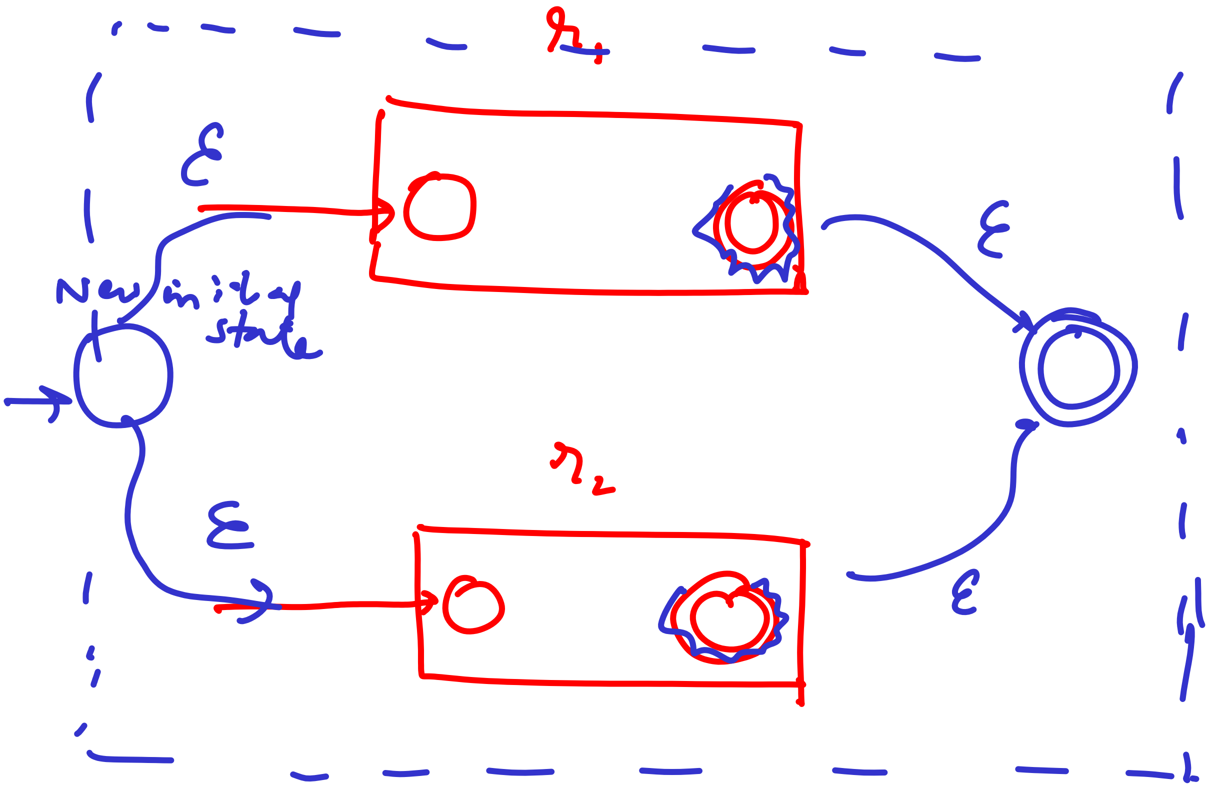


(iii) $\emptyset$ is a r.e.

The formal proof is by induction on the length of the r.e. r

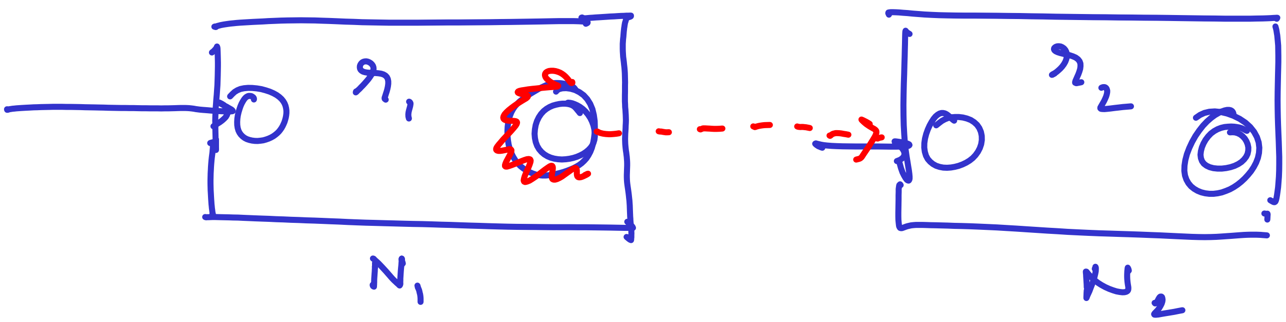
$0 + 1 \cdot 1^*$ has length ~~5~~ etc.

$r_1 + r_2$	$r_1 \cdot r_2$	r^*
By I.11. we have an NFA for r_1, r_2 since their lengths are strictly smaller. We have NFA with ε-Tran with a single finite state to recognise r_1, r_2	→ similarly	Similarly



NFA with ε transition and a single final state that recognises $r_1 + r_2$

For $r_1 \cdot r_2$



$s_1 \in S(r_1)$ $\qquad$ $s_2 \in S(r_2)$

set of strings in r_1

Clearly $s_1 \cdot s_2$ will be accepted

Suppose w is accepted by the composite machine

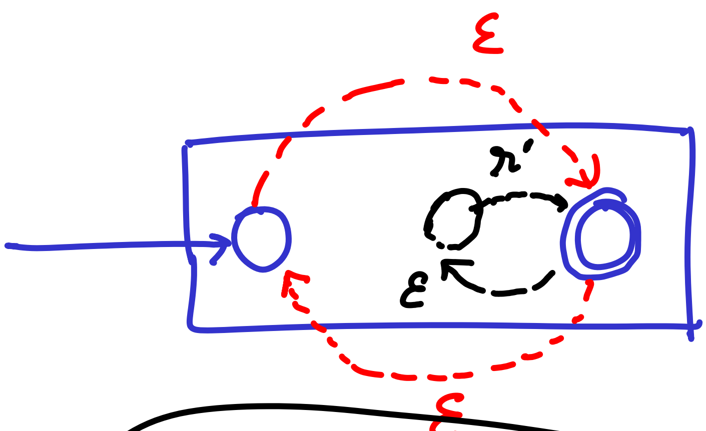
We must argue that

$$w = w_1 \cdot w_2 \quad \Big| \quad w_1 \in S(r_1), \; w_2 \in S(r_2)$$

<u>Obs</u> : A prefix of w, say w' must be such that $w' \in S(r_1)$. The remaining say $w'' \in S(r_2)$

<u>Note</u> $w' w'' = w$ may not be unique

$$(r)^* = \varepsilon + r + r.r + \cdots$$



$r' \notin S(r)$