just for this lecture ⇒ At several places in this lecture, I have referred NFA as Machine, so assume it as NFA. and machine is ~~only~~ meant to use for turing ~~mach~~ machines NTM/DTM

14/2/22 - Monday Class

Regular expressions is another type of model we have seen.

eg. $L = 0^* 1^* 2^*$ $0011222 \in L$

$01012 \notin L$

So we can build a language out of a regular expression

We will now see how much power it does have in terms of computing functions.

- Theorem 1

Let r be a regular expression, then ~~that~~ there exist an NFA M, having one final state and no transistions from the final state } specialised conditions added

such that $\underset{\text{NFA machine}}{L(M)} = \underset{\text{reg. exp.}}{L(r)}$

If we are successful in proving this we would have power of

regular expressions ≤ NFA

- Proof by induction on the number of operations in the regular expression.

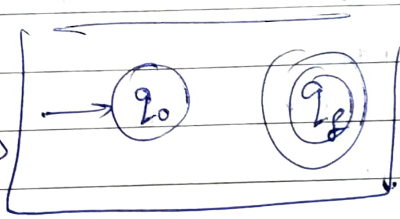
Suppose we have R, S then ~~R~~ R ∪ S, union is defined as ~~one~~ an operator. so such different operators can be used

(union, concatenation, kleene closure, etc)

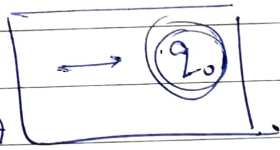
Remember: In DFA, NFA, if machine has no transition (or in halt state) in final state, and input is still not fully read, then it isn't accepted.

- basis:
  no. of operations in a reg. exp = 0
  what are the possibilities of strings
  ϕ
  or ε
  or $\forall \alpha \in \Sigma$ α

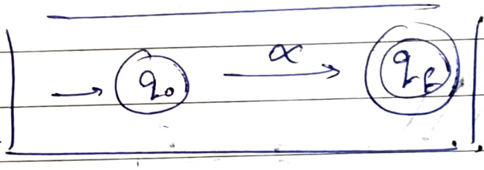
~~We have~~ for ϕ
this is the NFA $\Rightarrow$



for ε
this is the NFA



for α
this is the NFA



I refer to NFA with stated conditions as specialised NFA.

So, we gave NFA, with specialised conditions (mentioned in theorem), for every possible string.

- IH: Let r has less than i operations, the we could have a specialised NFA.

- IS: r' has i no. of operations.
  How could r' be split, just to trace how it was formed.

possibilities
1) $r' = r \cup s$ union
2) $r' = rs$ concatenation
3) $r' = r^*$ Kleene closure
4) $r' = r^+$ Kleene's positive closure

r, s individually have $(\leq i-1)$ operations

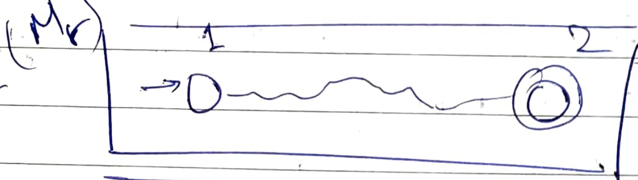
Let us take individual possibilities and analyse.

(1) $r' = r \cup s$.

r, s has $\leq i-1$ operation
by IH we have individual specialised NFA

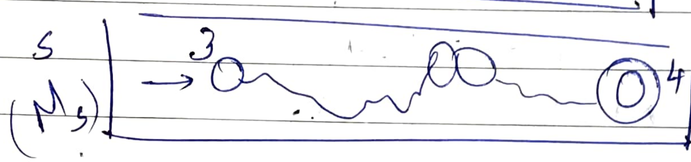
M denote machine.

for r



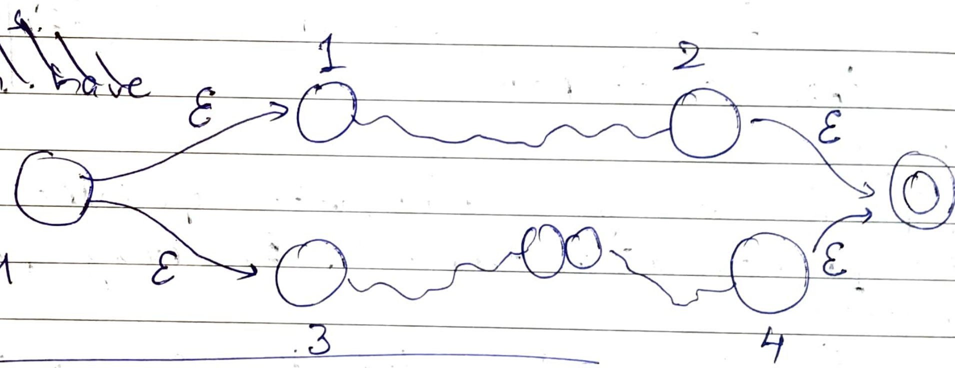
1, 2, 3, 4 denotes states

for s



~~for r U s we can have~~

Let this be Machine M



$x \in L(M_r) \Rightarrow x \in L(M)$
$x \in L(M_s) \Rightarrow x \in L(M)$
(similar proof as above)

Proof

If $x \in L(M_r)$ then, it starts from 1, goes to 2 and ends in accept state.
In M, ~~let~~ by first transistion we reach s, then from 1 to 2 we have same states, transistions as in for M_r, so consuming the input we reach at 2, and lastly by "ε" transistion we reach final state. so M also accepts x.

proof $x \in L(M) \Rightarrow x \in L(M_r)$ or $x \in L(M_s)$

If $x \in L(M)$, then while ~~x~~ processing x, we would have take either of 2 paths.

first: ε, 1 – 2 and last ε transistion
transistion, state to state

second: ε; 3 - 4, and last ε transistion

~~So~~ observing here first path is $[\varepsilon$, path in machine r from 1 to 2, $\varepsilon.]$

~~As ε doesn~~ If we remove the input consumed during first and last ε transistion, we ~~agai~~ get input $= x$.
So ~~a~~ M_r processes x from 1 to 2.
start state, accept state.

Hence $x \in L(M_r)$

If we take second path, and give similar arguments we get
$x \in L(M_s)$

Thus we proved $\boxed{x \in L(M) \Rightarrow x \in L(M_r) \text{ or } x \in L(M_s)}$

And eventually we can say

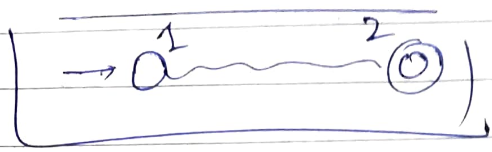
$x \in L(M) \Leftrightarrow x \in L(M_r)$ or $x \in L(M_s)$

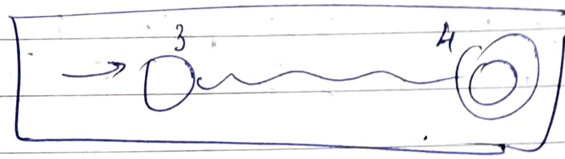
$$\boxed{x \in L(M) \Leftrightarrow x \in L(M_r) \cup L(M_s)}$$

② possibility

$r' = rs$ — similar conditions that we assumed for r, s

$rs \leq i-1$ operations

r  r denotes $L(M_r)$

s  s denotes $L(M_s)$

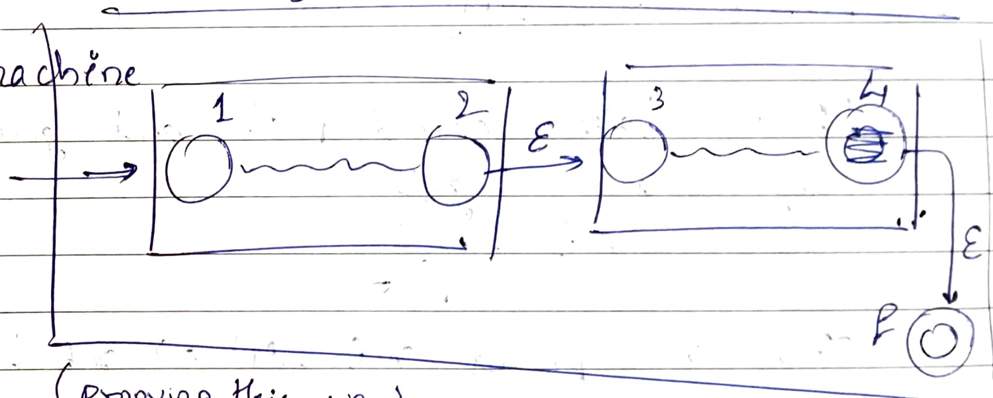
$r' = rs$

$R' \quad R \; S$

$R' = \{xy \mid x \in R, y \in S\}$

$= \{xy \mid x \in L(M_r), y \in L(M_s)\}$

Let machine M be.



(prooving this way)

$x \in L(M) \Leftarrow x' \in L(M_r), y' \in L(M_s), x = x'y'$

proof

~~using~~ consuming x', going from 1 to 2,
the ε transistion, going from 2 to 3,
consuming y', going from 3 to 4,
lastly ε transistion, going from 4 to f.

Thus reaching in an accept state; $x \in L(M)$

short in points. If asked in exam to prove, write this in sentences.

other way round.

$x \in L(M) \Rightarrow x' \in L(M_r),\ y' \in L(M_s),\ x = x'y'$

Suppose we have that machine and $x \in L(M)$. We can see we have only one route between 2, 3 to reach F (ε transaction)

So we must need to reach 2, then 3 and F through 4.

$\exists$ (there exists) a ~~If we~~ split x as per the consumption

$x = x' \varepsilon y' \varepsilon$

such that consuming x' we reached from 1 to 2 and ~~y~~ consuming y' we reached from 3 to 4.

Assuming 1 as a start state, 2 as an end state of some machine M_r, we can say

$x' \in L(M_r)$

Similary $y' \in L(M_s)$

and $x = x' \varepsilon y' \varepsilon = x'y'$

Hence proved.

$$x \in L(M) \Leftrightarrow \underbrace{x' \in L(M_r),\ y' \in L(M_s),\ x = x'y'}$$

This is defn of concatenation

$$x \in L(M) \Leftrightarrow x \in L(M_r) \cdot L(M_s)$$

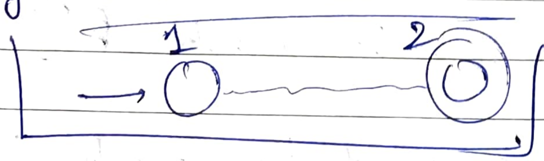
③ possibility.

r^* ∵ $*$ is 1 operation

r has $i-1$ operations

there is an ~~Machine~~ NFA present for r using IH

r



r denotes $L(M_r)$

r^* is language, in set form we have

$$R^* = \bigcup_{i=0}^{\infty} R^i$$

Imp

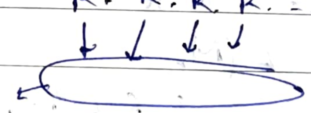
What does R^* mean

$R.R.R.R\ldots$ (zero or more)

r^* mean picking a single element from R and concatenating

like here $R.R.R.R\ldots$

r^*



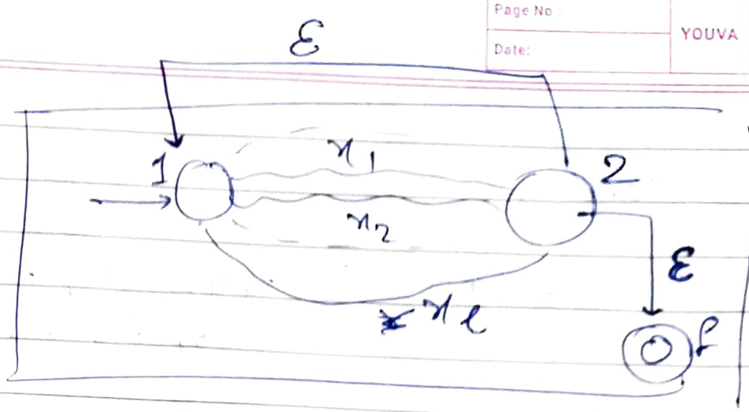
multiple combinations.

This perspective isn't mentioned earlier, but here it is concretely stated. Look ~~for~~ at previous examples, to observe them.

$x \in r^* \Leftarrow x_1 x_2 \ldots x_\ell \in r^*$ and $x = x_1 x_2 \ldots x_\ell$,

$x_1 \in L(M_r)$, $x_2 \in L(M_r)$ ..., $x_\ell \in L(M_r)$

Let Machine M be



Machine M_r has multiple routes from 1 to 2.

we start from 1, and can reach 2 in multiple ways.

~~Lets~~ first we take x_1. We come at 2, take ε come at 1.
again from 1 to 2 via x_2.
then again, ε.
and so on
⋮
lastly we take ε transition from 2 to f

so whole input = $x_1 \varepsilon x_2 \varepsilon \ldots \varepsilon x_\ell \varepsilon$

$$x = x_1 x_2 \ldots x_\ell$$

$x_1 \in L(M_r)$, $x_2 \in L(M_r)$ ..., $x_\ell \in L(M_r)$

this is by property of M_r.

Thus for this machine M.

$$x_1 x_2 \ldots x_\ell \in L(M)$$
$$x \in L(M).$$

Part 2 other way round on next page

Part 2.

$x \in L(M) \Rightarrow x_1 x_2 \ldots x_\ell \in L(M)$ and
$x_1 \in L(M_r)$, $x_2 \in L(M_r)$ ...
$x_\ell \in L(M_r)$

Let that machine accept x.
~~tet~~ So what are the possibilities of reaching from start state to end state.
(1) (2)

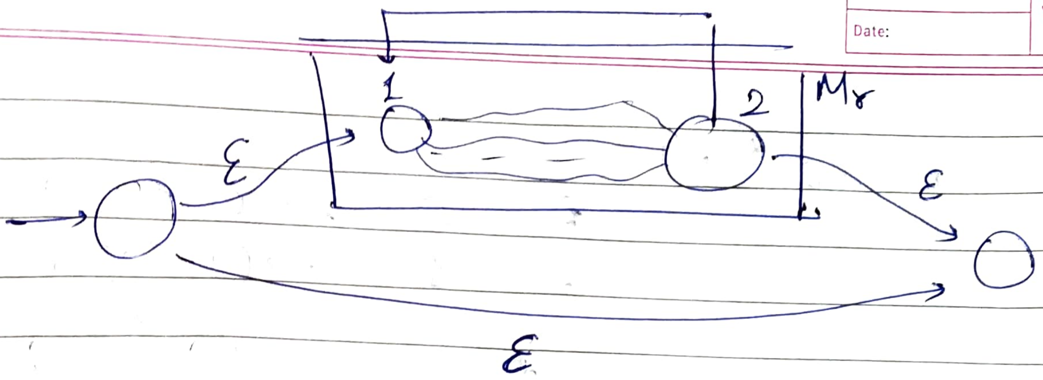
- Let we take x_1 first,
  then back to 1 using ε
  again we take some x_2
  then back to 1
  ⋮
  we take some x_ℓ
  and then to final state by ε transistion

So $x = x_1 \varepsilon x_2 \varepsilon \ldots \varepsilon x_\ell \varepsilon$
$x = x_1 x_2 \ldots x_\ell$

and for every $x_1, x_2, \ldots x_\ell$ we were reaching 2 from 1, we can ~~say~~ assume there to be a machine M_r starting at 1, and ending at 2 (final state).
And $x_1, x_2, \ldots x_\ell \in L(M_r)$

~~Th~~ Thus we proved the second part.

~~Modi~~ ~~Adding few~~ Modifying current machine M, we have



This is whole machine also ~~including~~ accepting ε.

$$\underbrace{x \in L(M) \Leftrightarrow \left(x_1 x_2 \ldots x_e \in L(M) \text{ and } x_1 \in L(M_r),\ x_2 \in L(M_r) \ldots\right) \cup \varepsilon}_{\text{defination of } r^*}$$

~~$x \in L(M)$~~

So we have an NFA for r^* also

(4) Possibility

r^+.

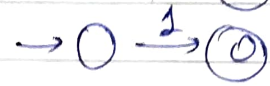
In ~~part~~ possibility (3), ~~just~~ eliminate the ε transistion from start to end/accept state.

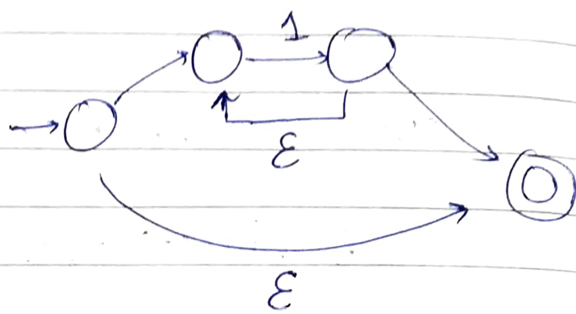
That will give us the NFA for r^+.

PROOF COMPLETED //

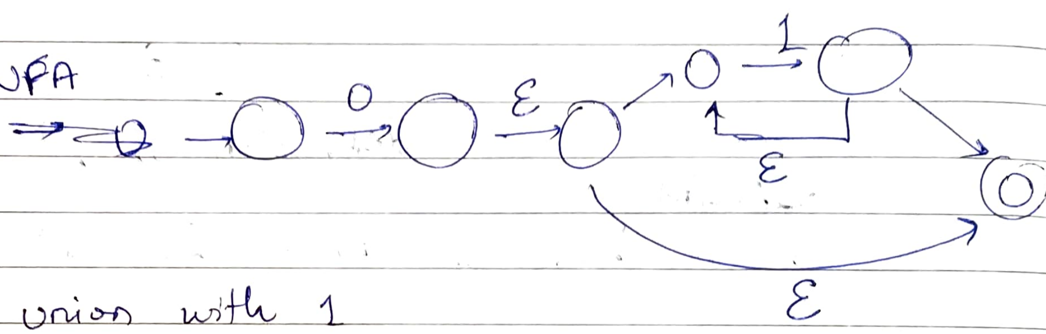
eg $r = (01^* + 1)$

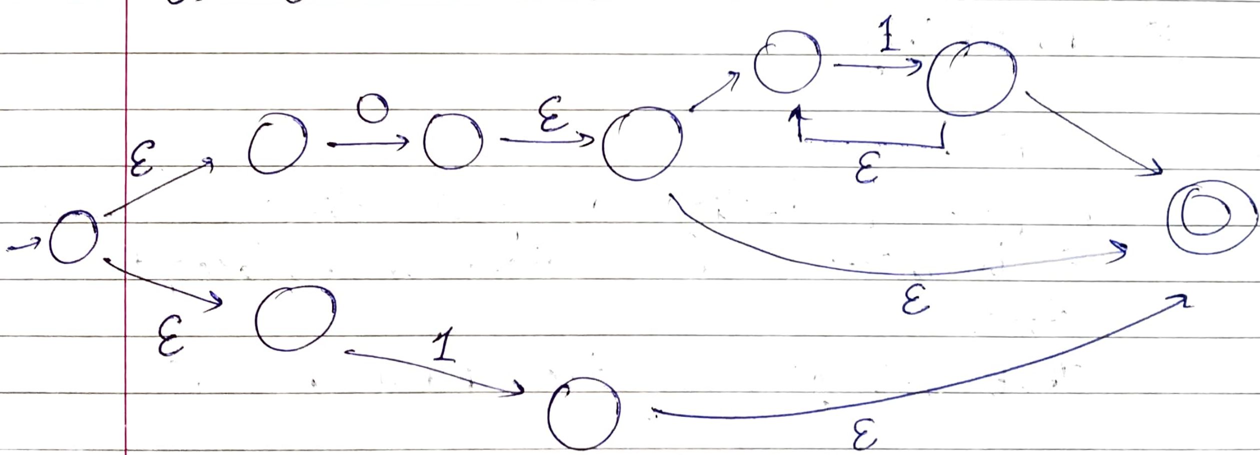
for 0


for 1


for 1^*


0 concatenated with 1^*

01^* NFA


01^* union with 1


NFA for $(01^* + 1)$

By the previous theorem, we can generate NFA for any regular expression

There is a hierarchy mentioned of all the models according to Current status of what we have done. Later some more proofs will come, where a bit of hierarchy will be modified. As it could create confusion i have not included it.