Regular Expressions $\longrightarrow$ Regular languages

$\hookrightarrow$ accepted by finite automata.

Language $\rightarrow$ collection of strings

Eg. $R = \varepsilon$ $\quad L(R) = \{\varepsilon\}$

$R = \phi$ $\quad L(R) = \{\}$

$R = a$ $\quad L(R) = \{a\}$

primitive

minimum lang formed.

$R_1 \cup R_2 \rightarrow$ Regular (union)

$R_1 \cdot R_2 \rightarrow$ Regular (concatenation)

a^* = (any value including 0) Eg. $\{\epsilon, a, aa, aaa \cdots\}$

$(R \cdot E)^* \longrightarrow$ Regular (Kleene closure)

$(a+b)^* = \{a, b, ab, ba, aa, bb, abab \cdots\}$

Language → Finite, Infinite

<u>Finite Lang:</u>

* it must be regular
* F·A can be formed
* R·E can be written

$\Sigma(a,b)$

(i) no string $\quad \{\} \quad \phi$

(ii) length 0 $\quad \{\epsilon\} \quad \varepsilon, \lambda$

(iii) length 1 $\quad \{a, b\} \quad (a+b)$

(iv) length 2 $\quad \{aa, ab, ba, bb\} \quad \boxed{(aa+ab+ba+bb)}$

$a(a+b) + b(a+b)$

$\boxed{(a+b)(a+b)}$

length 3 $(a+b)(a+b)(a+b)$

atmost 1 length (0) or (1)

$\{\epsilon, a, b\}$ $(\epsilon+a+b)$

atmost 2 $(\epsilon+a+b)(\epsilon+a+b)$

not more than 2b's and 1 a.

$\{\epsilon, \cancel{ab}, \cancel{ab}\ a, b, ab, ba, abb, bab, bba\}$.

add + in between for R.E

Infinite Language (*) will be used.

a^* - ϵ, a, aa, aaa.

a^+ - (all ~~contents~~ strings of ~~a~~ * except for ϵ)

$a^+ \Leftrightarrow aa^*$

$aa^\circ = a\cdot\epsilon = a$.

[min string in $a^* = \epsilon$

" " in $a^+ = a$]

(i) all strings having a single b.

$a^* \underline{b}\ a^*$

(ii) all strings having atleast one b.

$(a+b)^* \underline{b}\ (a+b)^*$

we cannot put * for b, else it will become ϵ.

(iii) strings having bbbb as substring

$(a+b)^*\ bbbb\ (a+b)^*$.

(iv) all strings end with ab

$(a+b)^* ab$

(v) all strings start with ba

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

(vi) beginning ~~with~~ and ending with a.

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

(vii) containing a.

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

(viii) starting and ending with diff symbol.

start	end
a	b
b	a

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

$$\Rightarrow \left(a(a+b)^* b\right) + \left(b(a+b)^* a\right)$$

(ix) having 2 bs.

$$a^* b a^* b \cdot a^*$$

$0^* = \varepsilon, 0, 00, 000, 0000 \cdots$ (any no. of zeros)

$(00)^* = \varepsilon, 00, 0000, 000000$ (even no.)

$(00)^* 0 = 0(00)^* = 0, 000, 00000 \cdots$ (odd no.)

Closure properties of regular languages:

(i) union $(L_1 \cup L_2)$

(ii) concatenation $(L_1 \cdot L_2)$

(iii) closure $(*)$ L^*.

(iv) complementation $\overline{L} = \Sigma^* - L$.

(v) intersection $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

(vi) difference $L_1 - L_2 = L_1 \cap \overline{L_2}$

(vii) reversal $(L)^R$.

(viii) homomorphism.

(ix) reverse homomorphism.

(x) quotient operation

(xi) INIT

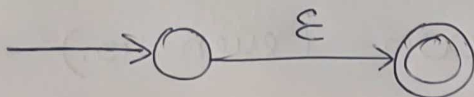
(xii) substitution

(xiii) infinite union

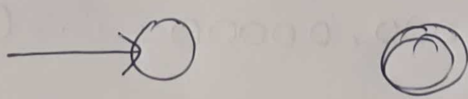
Regular exp are closed for all the operation except for

Basis

$R \cdot E = \varepsilon$

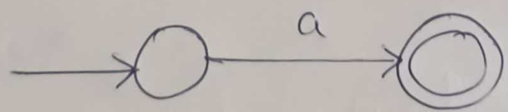


$R \cdot E = \phi$



(no transition)

$R \cdot E = a$



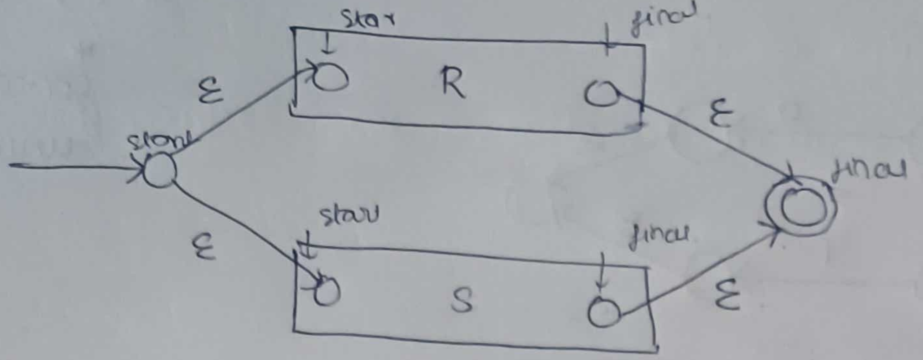
induction (4 steps)

(i) $RE = R + S$

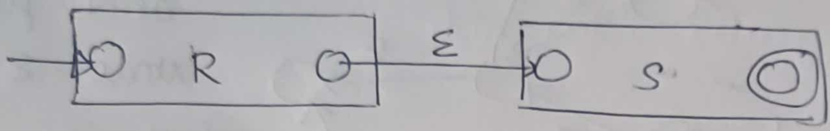
(ii) $RE = R^*$.

(iii) $RE = R^*$.

induction: (i) RE = R+S (R or S)

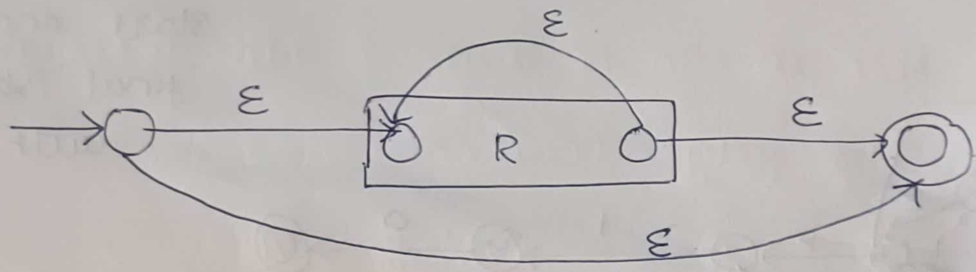


(ii) RE = R·S (R and S)



(iii) RE = R*

{ε, R, RR, RRR...}



(iv) RE = (R).

ε-NFA for R = ε-NFA for (R)

---

convert RE $(0+1)^*10$ to an ε-NFA

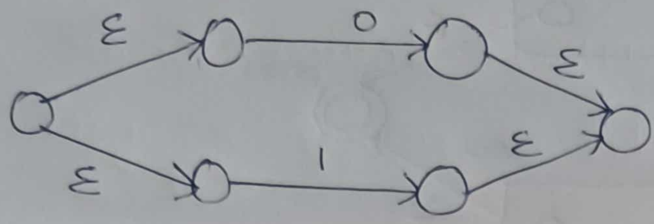
Precedence

1. ()
2. *
3. concat
4. + or | (union)

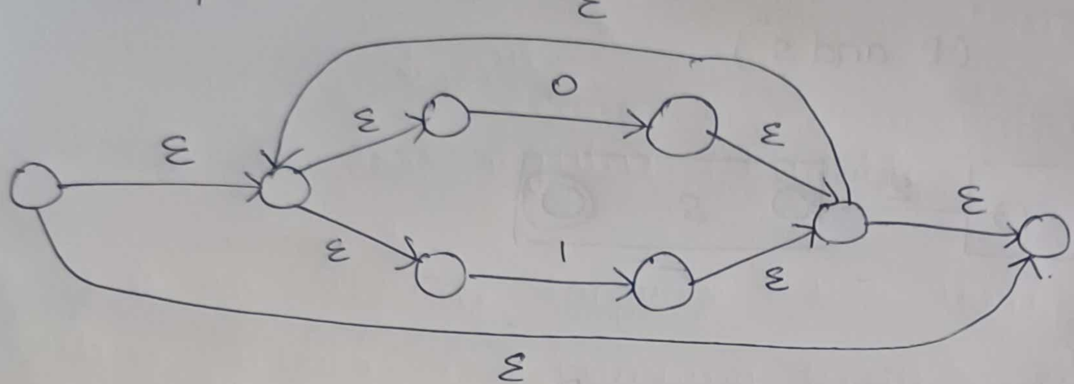
(0+1)*10.

Precedence
()
*
concat
union

step 1:



step 2:

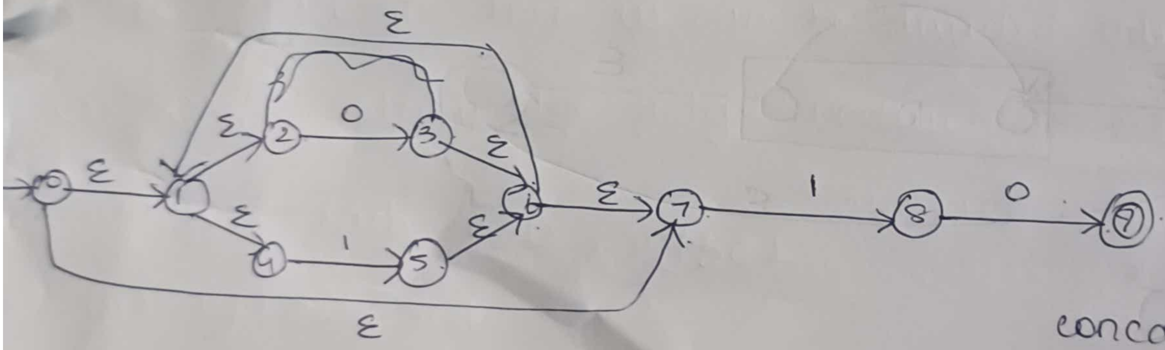


(i) connect start and final state - ε

(ii) new start & final state.

(iii) connect new start and new final state with ε

step 3:



concat na consider final state of R as start state of S.

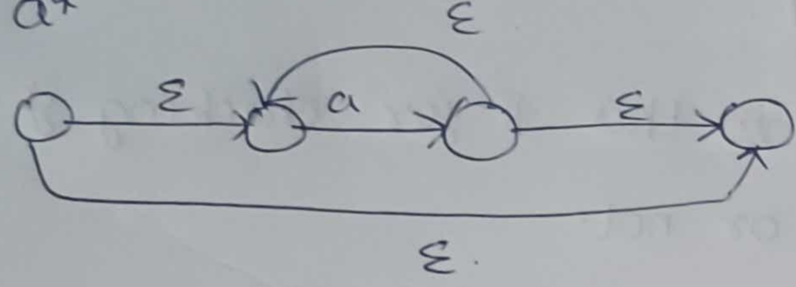
$\boxed{\text{R·E} \rightarrow \varepsilon\text{-NFA} \rightarrow \text{NFA} \rightarrow \text{DFA}}$

<u>pumping</u> <u>lemma</u>. for regular sets

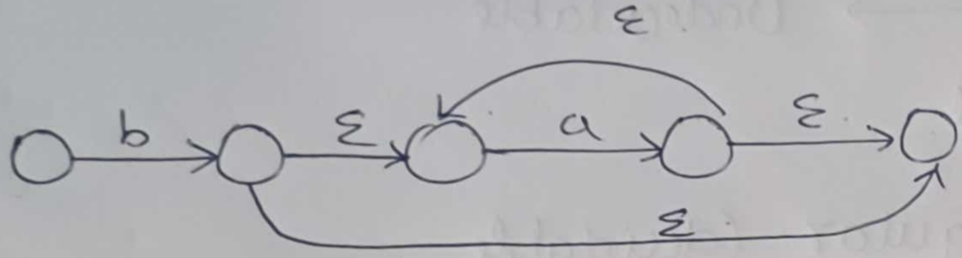
Find R for finite automation given below.

Convert b+ba* to ε-NFA

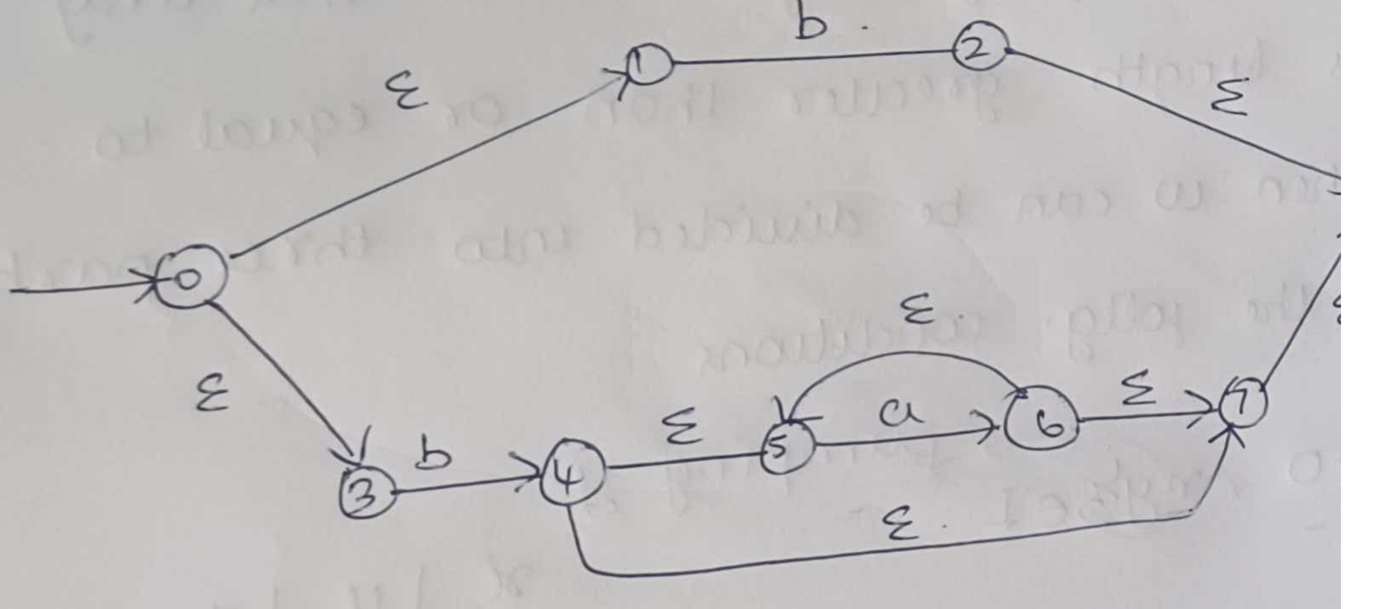
step 1: a*



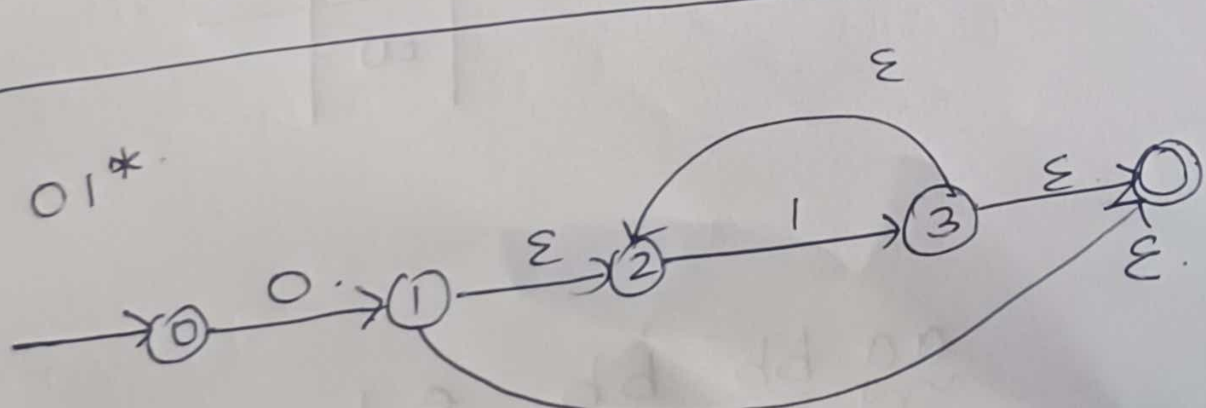
step 2: ba*



step 3: b+ba*



01*



(negative test) ✗

pumping lemma: one of the methods to check if a languag is regulor or not.

finite is always regular so this is for checking if infinite lang is regular or not.

$L \longrightarrow$ (PLT) $\xrightarrow{\text{pass}}$ Undecidable.

$\downarrow$ fail

non-regular. (decidable

If L is an infinite language then there exists some positive integer 'n' (pumping length) such that any string $w \in L$ has length greater than or equal to 'n' ie. $|w| \geq n$ then w can be divided into three parts, $w = xyz$, satisfy the follg. conditions

(i) for each $i \geq 0$, $xy^iz \in L$. → pumping y.

(ii) $|y| > 0$ and.

(iii) $|xy| \leq n$.

x	y	z.
	w	

eg. $a^n b^{2n}$ $n \geq 0$.

aabbbb. $\in L$

$\boxed{w \in L}$

$\underbrace{aa}_{x}\ \underbrace{bb}_{y}\ \underbrace{bb}_{z}$. $\in L$. ✓

$\underbrace{aa}_{x}\ \underbrace{bbbb}_{y}\ \underbrace{bb}_{z.}$ $\cancel{\in L}$ ✗.

not regular.