

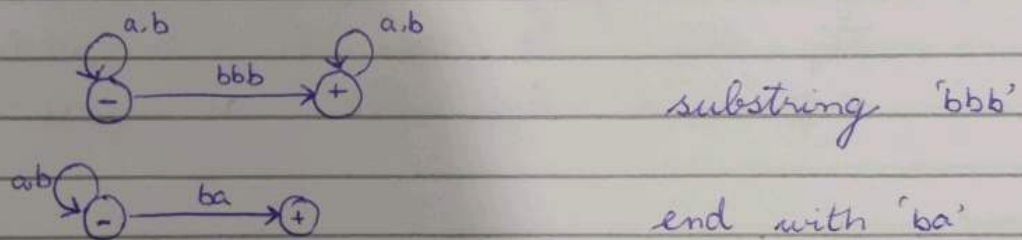
⇒ NFA (Non deterministic finite automata)

- allow missing edges
- allow duplicate edges

NFA can't be DFA but DFA can't be NFA
NFA has different behavior on same input.

⇒ TG (Transition graph)

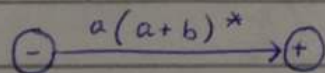
- Null $\ominus \xrightarrow{\wedge} \oplus$
- all conditions of NFA are allowed.
- Can take substring / substrings are allowed.



→ Every DFA can be TG.

⇒ GTG (Generalize Transition Graph)

- Start from 'a'



- all conditions of NFA & TG
- GTG can't be DFA, but every DFA is GTG

Kleen's Theorem

Part 1

If a language can be accepted by DFA then it can be accepted by TG.

Prove

Every DFA itself called TG.

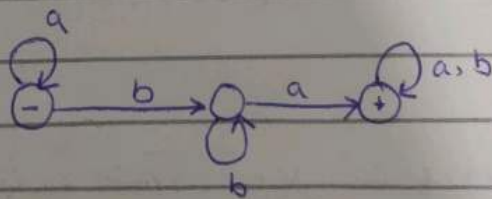
Part 2

If a language can be accepted by TG then it can be accepted by / expressed by RE as well.

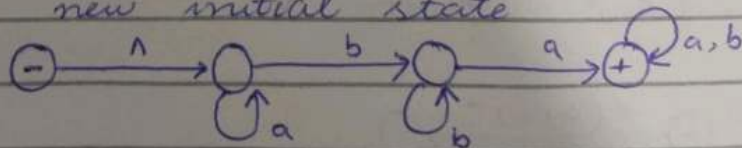
Prove

DFA \rightarrow must have substring ba

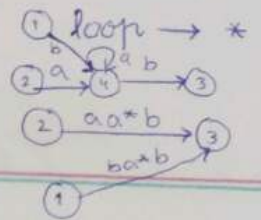
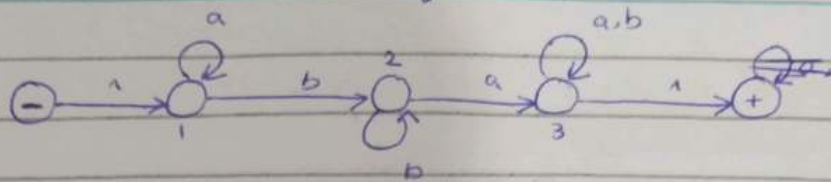
Step 1



Step 2 : add new initial state

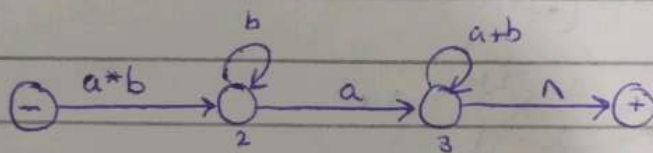


Step 3 : add new final state

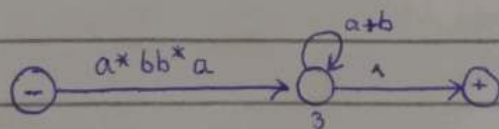


Choose the state which has less no. of edges.

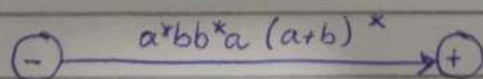
By removing '1' state



Now, removing '2'



Now, By removing '3'



$a(a+b)^*$ \rightarrow union
 \rightarrow Concatenation
 $\rightarrow +/ \times$ operation

DFA \rightarrow TG
 TG \rightarrow RE
 RE \rightarrow DFA

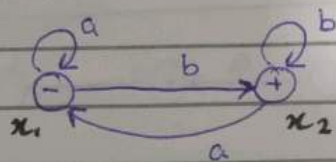
Part 3

If a language is accepted by RE then there must be a DFA for that language.

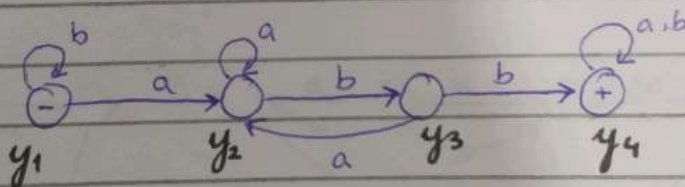
DFA₁ \rightarrow "end with b"
 +

DFA₂ \rightarrow must have "abb"

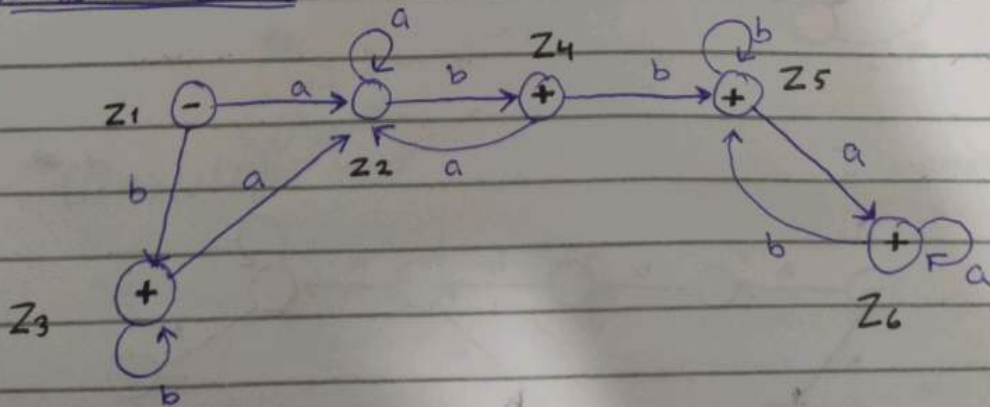
DFA₁



DFA₁



Union DFA



Union = dono mn sy kisi bhi RE ya DFA sy start ya end kr saktay.

Concatination = start from first. end at last.

	a	b	
$Z_1(x_1, y_1)$	$x_1, y_2 = Z_2$	$x_2, y_1 = Z_3$	$Z_1 = x_1, y_1$
$Z_2(x_1, y_2)$	$x_1, y_2 = Z_2$	$x_2, y_3 = Z_4$	$Z_2 = x_1, y_2$
$Z_3(x_2, y_1)$	$x_1, y_2 = Z_2$	$x_2, y_1 = Z_3$	$Z_3 = x_2, y_1$
$Z_4(x_2, y_3)$	$x_1, y_2 = Z_2$	$x_2, y_4 = Z_5$	$Z_4 = x_2, y_3$
$Z_5(x_2, y_4)$	$x_1, y_4 = Z_6$	$x_2, y_4 = Z_5$	$Z_5 = x_2, y_4$
$Z_6(x_1, y_4)$	$x_1, y_4 = Z_6$	$x_2, y_4 = Z_5$	$Z_6 = x_1, y_4$

Final states for Union

x_2, y_4

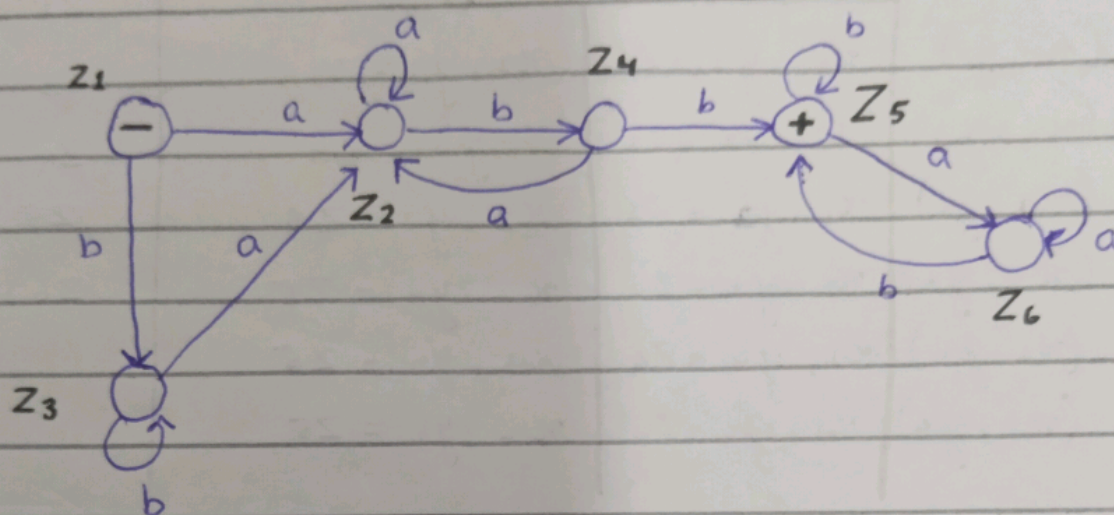
- + $Z_3 = x_2, y_1$
- + $Z_4 = x_2, y_3$
- + $Z_5 = x_2, y_4$
- + $Z_6 = x_1, y_4$

Final states for intersection

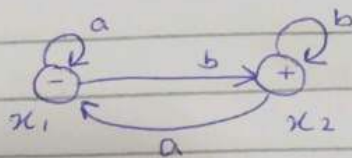
x_2, y_4

- + $Z_5 = x_2, y_4$

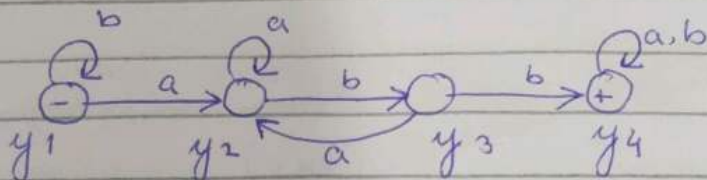
Intersection DFA



DFA₁



DFA₂



Concatination DFA

	a	b
$Z_1(x_1)$	$x_1 = Z_1$	$x_2, y_1 = Z_2$
$Z_2(x_2, y_1)$	$x_1, y_2 = Z_3$	$x_2, y_1 = Z_2$
$Z_3(x_1, y_2)$	$x_1, y_2 = Z_3$	$x_2, y_1, y_3 = Z_4$
$Z_4(x_2, y_1, y_3)$	$x_1, y_2 = Z_3$	$x_2, y_1, y_4 = Z_5$
$Z_5(x_2, y_1, y_4)$	$x_1, y_2, y_4 = Z_6$	$x_2, y_1, y_4 = Z_5$
$Z_6(x_1, y_2, y_4)$	$x_1, y_2, y_4 = Z_6$	$x_2, y_1, y_3, y_4 = Z_7$
$Z_7(x_2, y_1, y_3, y_4)$	$x_1, y_2, y_4 = Z_6$	$x_2, y_1, y_4 = Z_5$

$$Z_1 = x_1$$

$$Z_2 = x_2, y_1$$

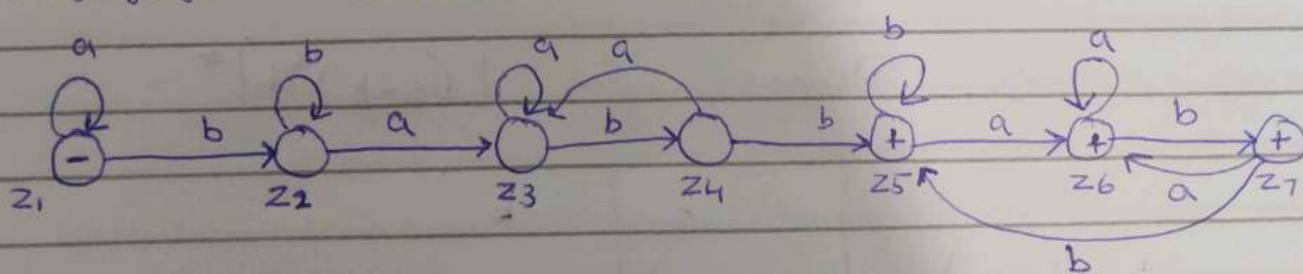
$$Z_3 = x_1, y_2$$

$$Z_4 = x_2, y_1, y_3$$

$$Z_5 = x_2, y_1, y_4$$

$$Z_6 = x_1, y_2, y_4$$

$$Z_7 = x_2, y_1, y_3, y_4$$



\Rightarrow All finite languages are regular (True)

\Rightarrow RE, TG, DFA \rightarrow regular language

\Rightarrow Every finite language has DFA

\rightarrow Every DFA has TG

\rightarrow Every TG has RE

\Rightarrow non regular language \rightarrow No DFA

\rightarrow No TG

\rightarrow No RE

Example

$a^n b^n, n \geq 0$

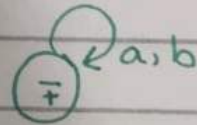
\rightarrow Has PDA

\Rightarrow Subset of regular language is always regular? (False)

Example

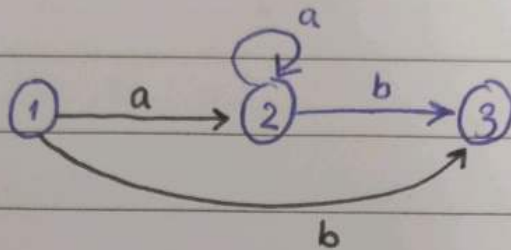
$(a + b)^*$

\rightarrow Regular language

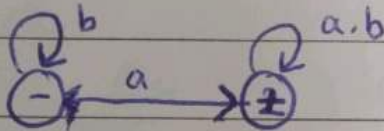


Union of NFA

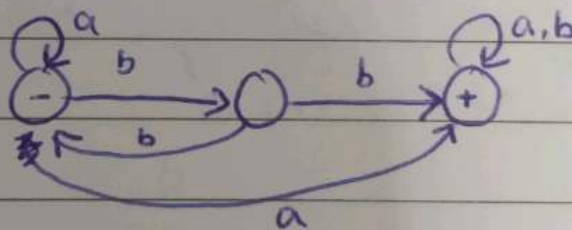
جس state سے جوڑنا ہے اس کی A or B جہاں جہاں جاری ہیں، جس کو جوڑنا ہے اس کو بھی وہاں وہاں جوڑیں گے۔



NFA₁



NFA₂

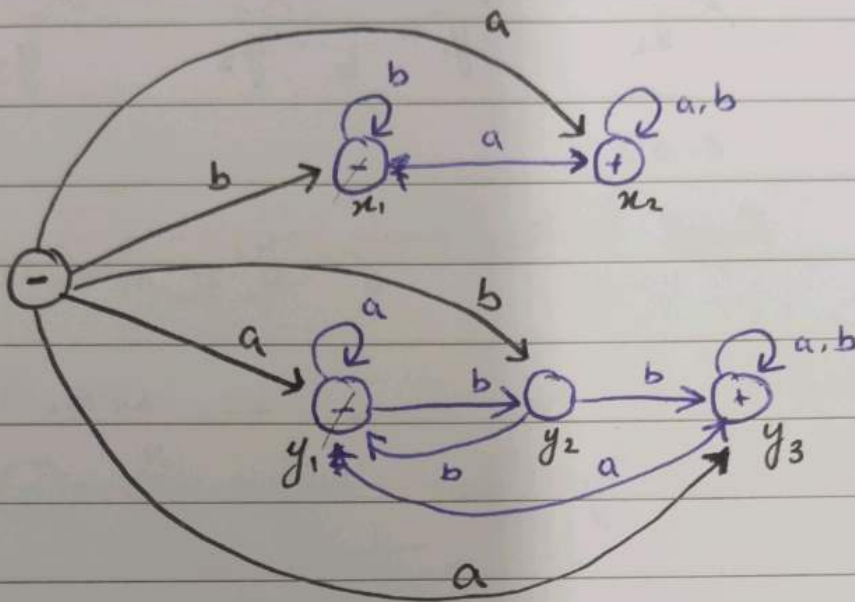


Steps:

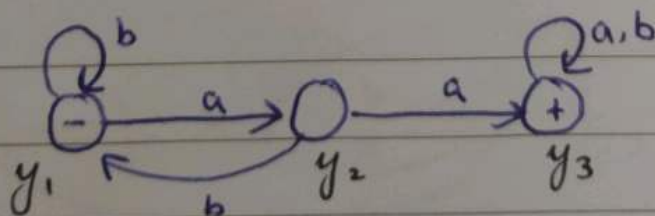
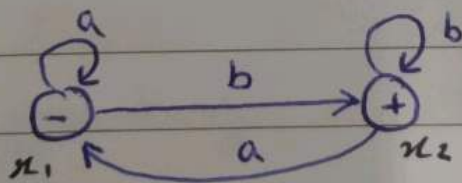
Make new initial

Connect with previous initials.

Union NFA




Concatenation

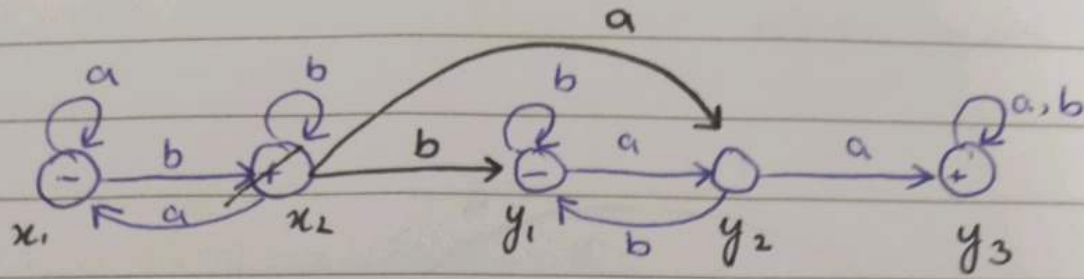


Steps

- 1st ki starting last ki ending
- 1st ki ending join ho jayegi 2nd ki start sy.

→ x_2 k a & b bhi y_1 ki  A or B wali States per jayen gy.

Concatination of NFA



Final state only ' y_3 '

→ NFA₁ ki har Final ko NFA₂ in initial sy join karen gy.

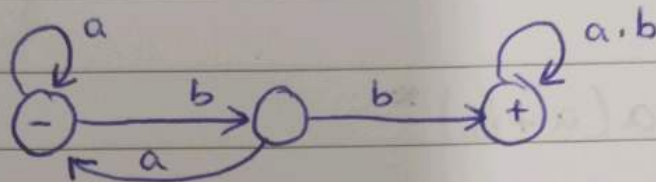
Question

Kis condition mn NFA₁ ki final remove nhi hogi?

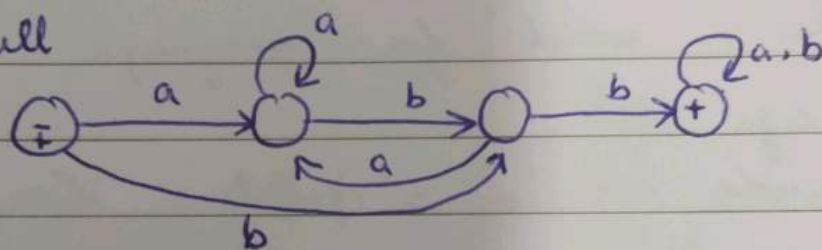
⇒ when NFA₂ is producing null.

⇒ * operation .

new initial & also make it final
for null



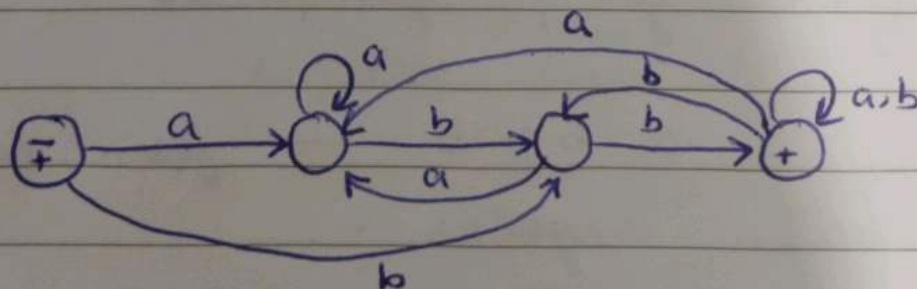
For null



For repetition .

→ Join initial with Final .

→ ~~From~~ Final ki edges us state
pr jayen gi jaha initial
ki edges ja rhi hn .



Pumping Lemma

Identify ~~the~~ whether the language is regular ~~and not~~ or not.

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

Take a word for language

e.g. abb

Divide it in 3 parts

w	Before null
y	in loop
x	After null

$$\frac{a}{w} \quad \frac{b}{y} \quad \frac{b}{x}$$

Take the y & pump it

a b b b \rightarrow from language

so, regular.

$\Rightarrow a^n b^m$ language

e.g. $\frac{aabb}{y}$

'pump the y' $\rightarrow aaaa bb \rightarrow$ not from language

So, not a regular language.

$\Rightarrow a^* b^*$

$\frac{ab}{y}$

$\rightarrow abab$

\rightarrow not from language.

$\frac{ab}{y}$

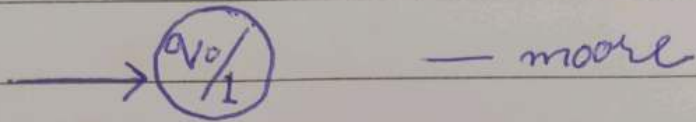
$\rightarrow acb$

\rightarrow from language.

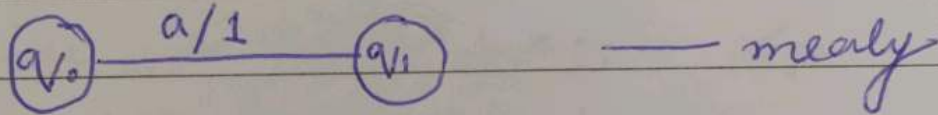
\rightarrow \nexists Agr 1 bhi word language ka aa jye to language regular hai.

⇒ Moore & mealy machine..

- no accepting & rejecting.
- input & sath output kha a rhi hai.
- no final state, jahan rukty hn wohi final ban jati hai



output state k andr.

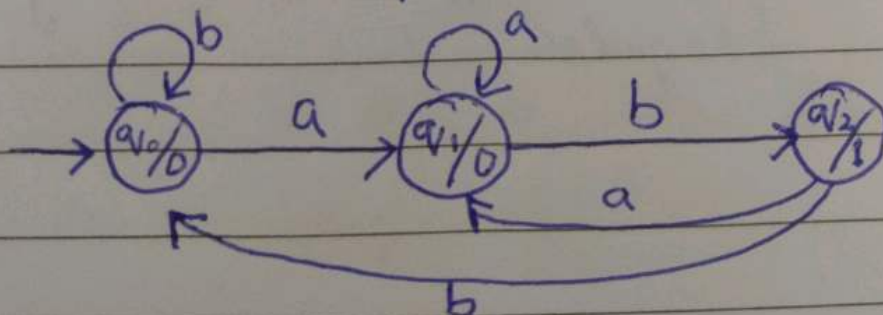


output edge pr.

Output - $I = \{0, 1\}$

Question

machine produces ab (moore)



⇒ 3 a's in input = 3 1's in output.

Question

machine that produces ab (mealy)

