design → automata
~~comp~~ implement → ~~for~~ compiler

DATE: 3 / 1 / 17
PAGE NO.:

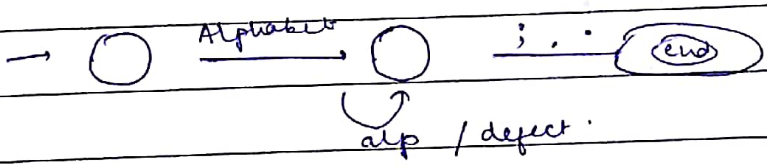
Book:
Formal language and Automata Theory — Peter Linz

Theory of Computation

→ System, while converting HLL to binary, first checks the rules & then converts.

→ Theoretical model of how compiler converts ~~to~~ HLL to LLL is done by Automata

→ Representing in form of diagram is 1 type of automata



Automata : tool which finds whether given i/p is correct or not

① Finite Automata : have finite no. of states (have no memory, processing directly)
② Push Down Automata (FA + stack) → memory in form of stack
③ Linear Bounded Automata (FA + Finite tape) → can process any element
④ Turing Machine (FA + Infinite tape)
↓
can process any type of data.

→ One ~~not p~~ which can't be processed by Turing Machine:
Uncomputable / Undecidable.

→ State remembers on which step I am

Finite Automata

It is defined as 5 tuple Q, Σ, δ, q_0, F

Q → set of states
q_0 → initial state
F → set of final states
δ → Transition funcⁿ
Σ → Input symbol

Finite Automata (FA)

Deterministic (DFA) | Non-Deterministic (NFA)

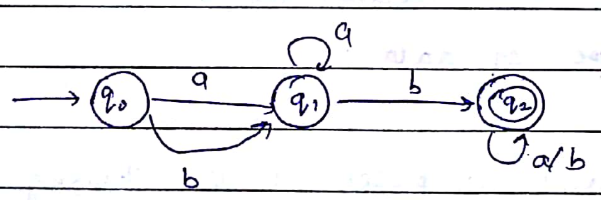
→ are the automata which have move defined for every symbol.
→ has only 1 move / symbol
→ moves not defined are considered ~~said~~ as dead moves

DFA

$\delta : Q \times \Sigma \rightarrow Q$

see state, see funcⁿ → generate new state

ransition iagram



$Q = \{q_0, q_1, q_2\}$
$q_0 = \{q_0\}$
$F = \{q_2\}$
$\Sigma = \{a, b\}$

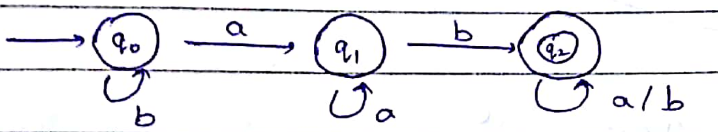
ransition table

	a	b
i.s. → q_0	q_1	q_1
q_1	q_1	q_2
q_2 f.s.	q_2	q_2

Transition func^n :

$\delta(q_0, a) \rightarrow q_1$
$\delta(q_0, b) \rightarrow q_1$
$\delta(q_1, a) \vdash q_1$
$\delta(q_1, b) \mapsto q_2$
$\delta(q_2, a) \rightarrow q_2$
$\delta(q_2, b) \rightarrow q_2$

· Design a finite automata over $\{a, b\}$ that accept string containing ab as substring.



: Tool / Flow chart

need one move for every symbol

→ Now, have to check whether correct or not

Acceptance by final state : acceptable ~~by~~ only if it reaches to Final state

↳ baa

$(q_0, baa) \vdash (q_0, aa) \vdash (q_1, a) \vdash (q_1, \wedge)$

not on F.S.

⇓

not acceptable

↳ ab ab a

$(q_0, ababa) \vdash (q_1, baba) \vdash (q_2, aba)$
$\vdash (q_2, ba) \vdash (q_2, a) \vdash (q_2, \wedge)$

⇓

on F.S. ⇒ acceptable

In DFA : only 1 automata is possible

ii.) contains aba as substring.

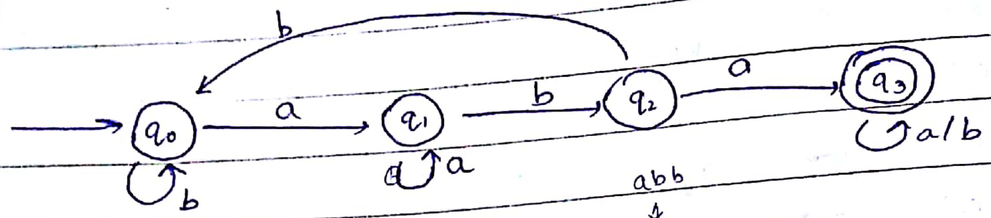
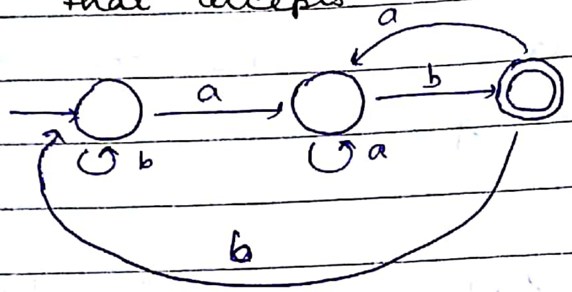


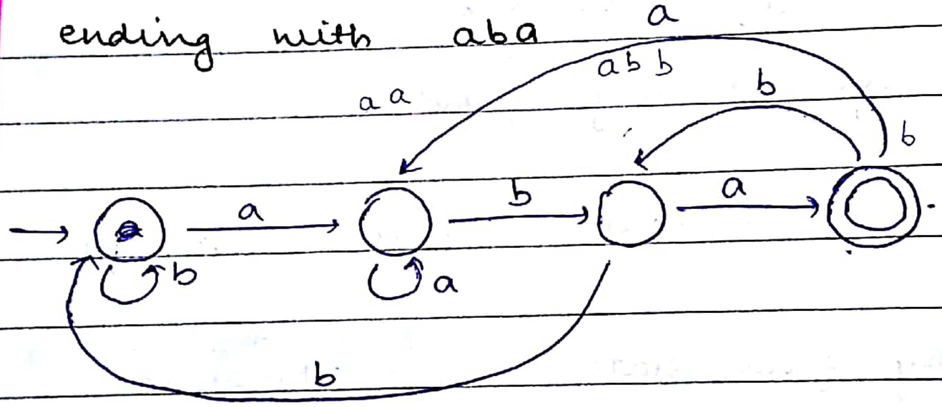
abb
↓
again need a

Q i.) FA that accepts all string ending with ab.



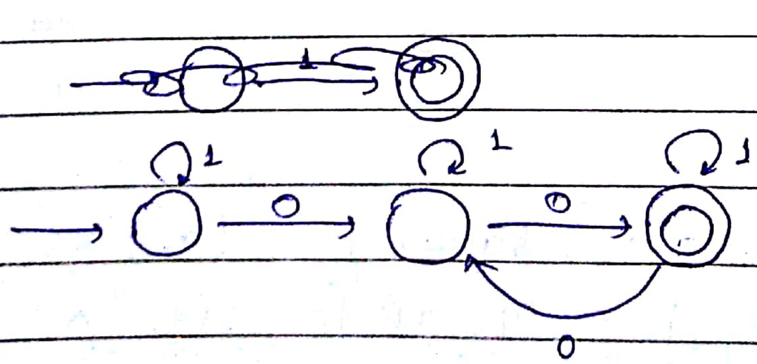
ii.) ending with aba

abab
abaa



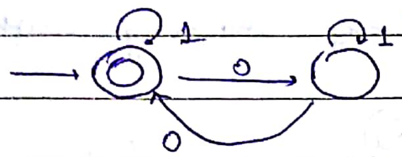
Q. $\Sigma = \{0,1\}$ (no of $0 \geq 1$)

accept all string containing 0 in even no.



→ whenever accepting string of 0 length, initial state is always final state

if |0| ≥ 0



→ i/p : a, b

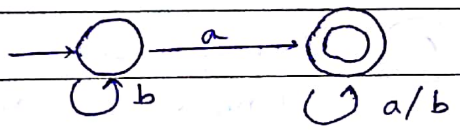
all strings containing

~~exactly 1~~ a

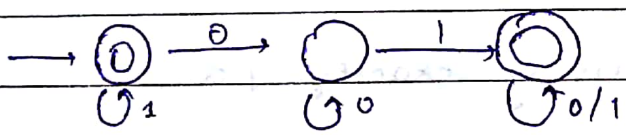
1.) atleast 1 a.

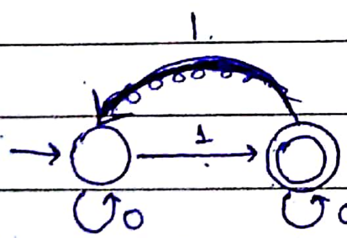
2.) 0 followed by 1.

3.) odd no. of 1's

1.) 

2.) 0-1

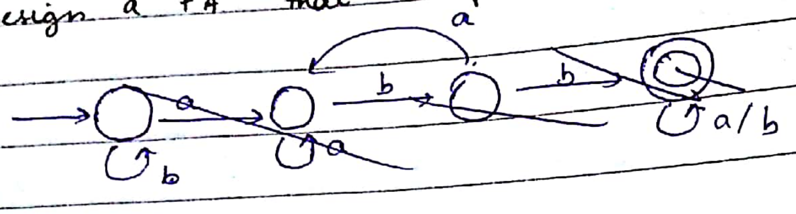


3) 

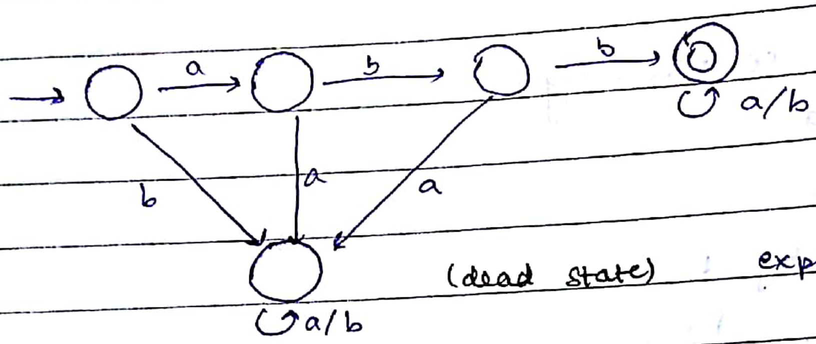
Dead State : It is a state which has all outgoing moves defined to itself, ie, $\forall\ a \in \Sigma$

$\delta(q_d, a) \rightarrow q_d$

Q Design a FA that accept ~~st~~ string starting with abb.

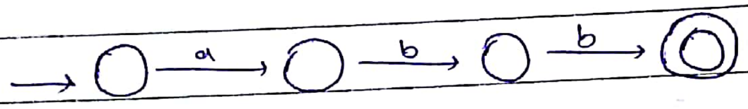


aba



(dead state) explicitely specified dead state

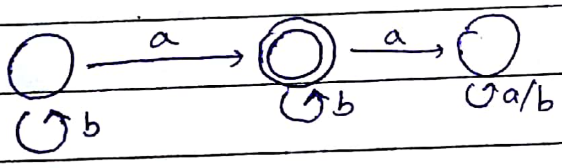
OR



implicite
(moves not defined are considered as dead moves)

→ $\Sigma = \{a, b\}$

Accept all string containing exactly 1 a.



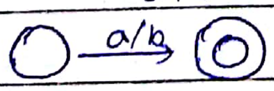
Q. What will be min no. of states required to accept string of length n over $\Sigma = \{a, b\}$

(a) n ✓(b) n+1 (c) ~~n−1~~ (d) None of these

Use principle of induction

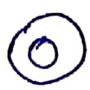
1.) n = 1 2.) n = 2 : 3 states

a/b



Teacher's Signature

3) n = 0



⇒ n+1

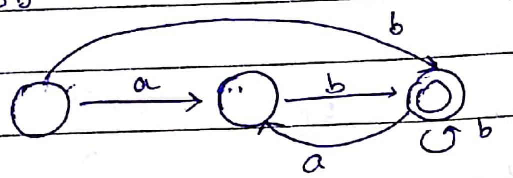
* length : exactly n ⇒ dead move after ⊚ in all cases
  : atleast n ⇒ we can write ⊚ ↺ a/b (length can be more than n)

* ⊚ ↺ a/b : automata accepting ^.
* Automata should not accept anything other than what's mentioned.

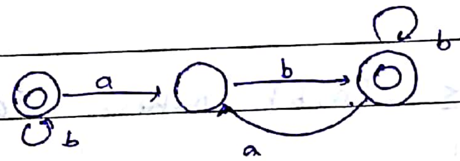
→ accept all strings having a followed by b

bb ✓ ab ✓ aba ✗

b → should accept b



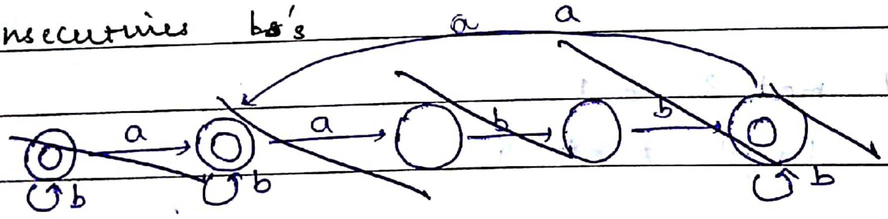
OR



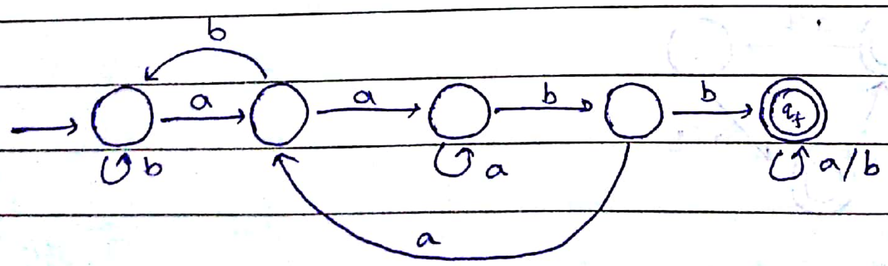
* We can've multiple Final set but only 1 initial set in this type

** → accept all strings containing ~~atleast~~ atleast 2 consecutive 'a' followed by 2 consecutives b's

ab ⇓ ~~1 time rule~~ is followed, it is done



aaabb ✓
aabb ✓

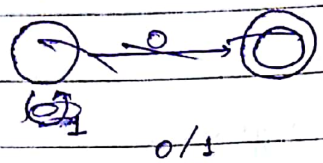


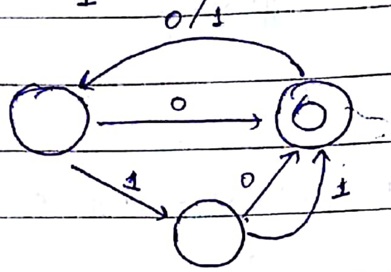
* If rule is ~~not~~ followed with no occurence also, 1st state will be final state.

Teacher's Signature

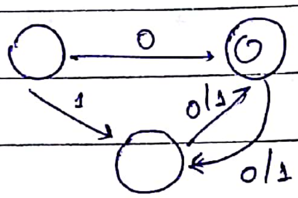
→ all string that start with 0 have odd length & start with 1 have even length

$\Sigma = \{0,1\}$

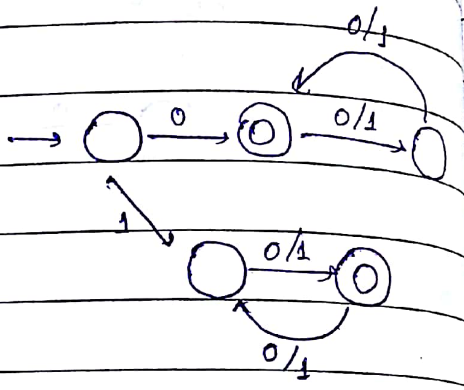




X

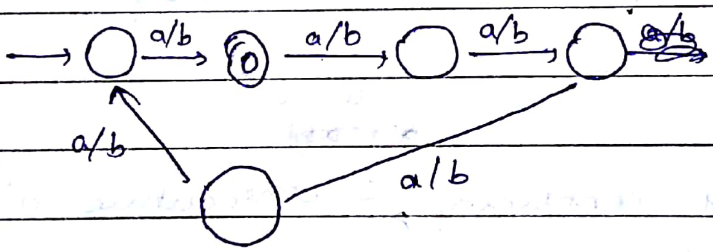
→ 

OR



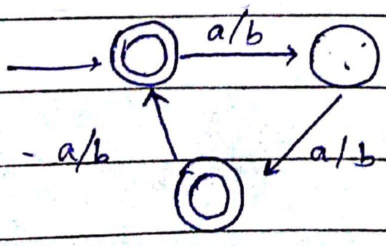
→ Draw a FA over $\Sigma = \{a,b\}$ where $|w| \bmod 5 = 1$

⇒ length : 1, 6, 11, ...



→ $|w| \bmod 3 \neq 1$

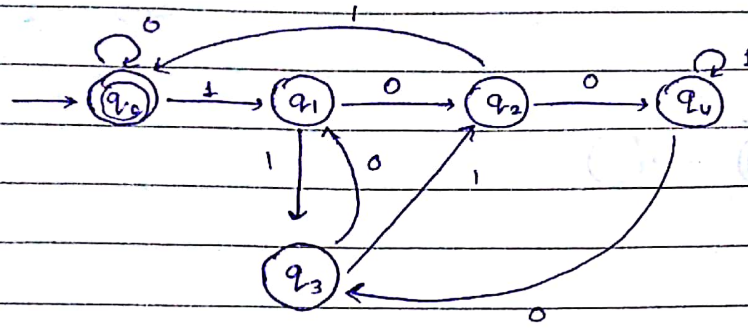
⇒ length ≠ 1, 4,



→ accept all binary string divisible by 5.

Rem
0

0
10
11



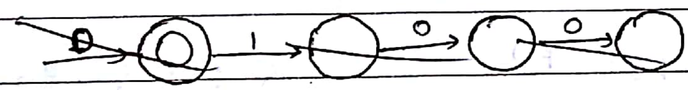
→ all string that don't contain 100 as substring

→ " " where every odd position is 1.

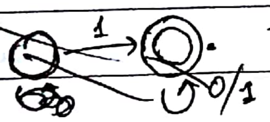
→ all string containing equal occurence of 01 and 10.

→ " " even a's and odd b's.

①



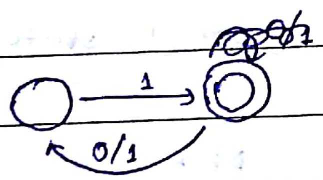
②



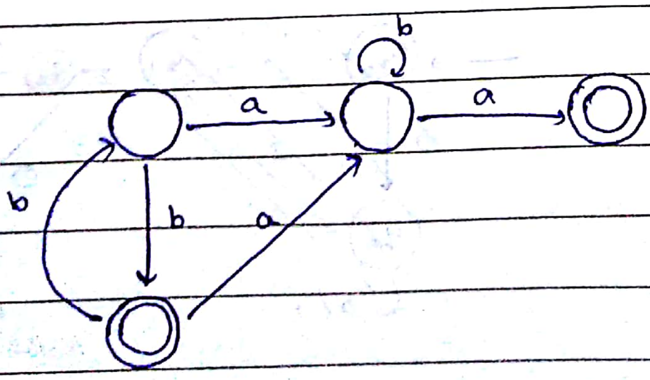
1 ✓
~~010~~

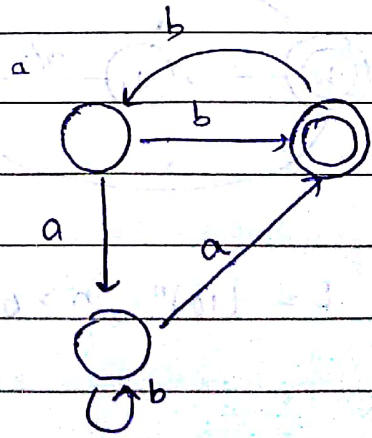
011

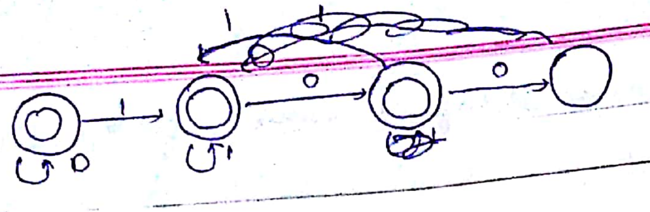
②



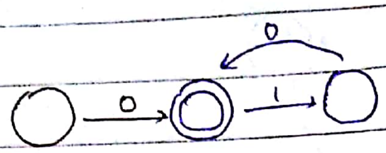
④





① 

010

③ 

<u>Non - Deterministic Finite Automata</u>

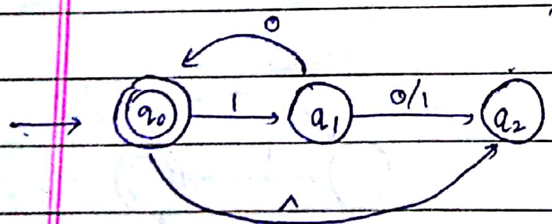
5 tuple $(Q, \Sigma, \delta, q_0, F)$ $Q = \{q_0, q_1, q_2\}$

$\delta : Q \times (\Sigma \cup \Lambda) \longrightarrow 2^Q$, $(q_0, a) \vdash \{q_1, q_2\}$ ⇓ subset of Q

→ Acceptance of string :-
On null move also, we can move from 1 state to another

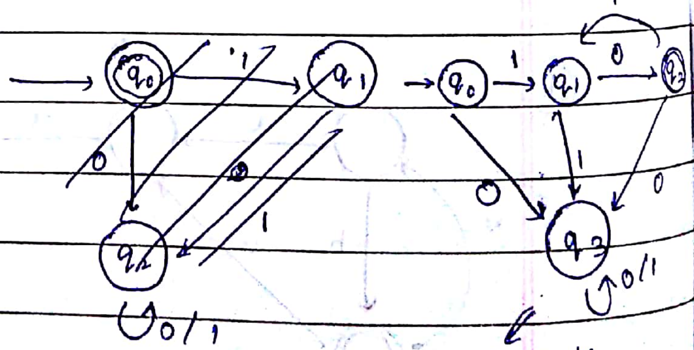
NDFA

dead state



$L = (10)^n, n > 0$

DFA



dead state

$L = (10)^n, n > 0$

Teacher's Signature

not accepting null

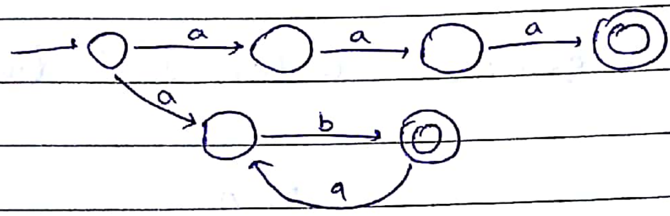
1010

$\rightarrow (q_0, 1010) \vdash (q_1, 010) \vdash (q_2, 10) \vdash (q_2, 0) \vdash (q_2, \Lambda)$

OR

$\rightarrow (q_0, 1010) \vdash (q_1, 010) \vdash (q_0, 10) \vdash (q_1, 0) \vdash (q_0, \Lambda)$

Eg. $L = (a^3 \cup (ab)^n \quad n > 0)$

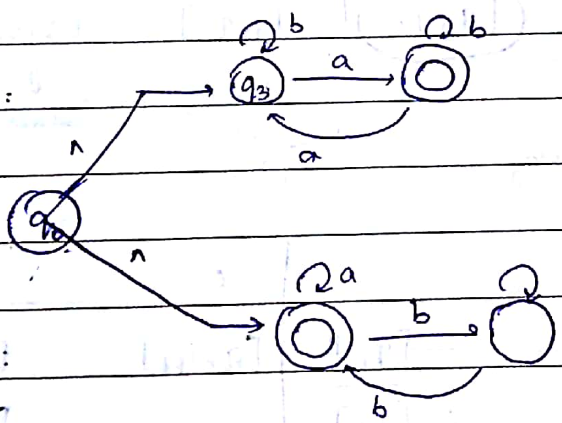


Eg. $L = L_1 \cup L_2$

L_1 : a in odd no.

L_2 : b in even no.

L_1 :



L_2 :

$(q_0, aaab) \vdash (q_3, aaab)$

$\Downarrow$

State has changed but string is the same $\Rightarrow$ using 'Λ'

** NDFA to DFA

$M = (Q, \Sigma, q_0, \delta, F)$ be NDFA

$M' = (Q', \Sigma, q_0', \delta', F')$ be corresponding DFA

$Q' = [q_0, q_1, \ldots q_i]$ where $q_0, q_1, \ldots q_i \in Q$

Eg. : NDFA : $\delta(q_0, a) \vdash q_1, q_2$

DFA : $[q_1 q_2]$

Teacher's Signature

Scanned by CamScanner

$q_0' = [q_0]$

$\Sigma' = \Sigma$

F' = it is state that contains at least 1 final state of NDFA

$[q_0\ q_1\ \ldots\ q_k] \in F'$ if $\exists\ q_i \in F$

$\delta'([q_1\ q_2\ \ldots q_i], a) \vdash (q_1, q_2, \ldots, q_k)$

where $\delta(q_1, a) \cup \delta(q_2, a) \cup \quad \cup (q_i, a) = \{q_1, q_2, \ldots, q_k\}$

Eg

NDFA

	a	b
→ q_0	q_0	q_1
q_1	q_1	$q_0\ q_1$

DFA

	a	b
→ $[q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0 q_1]$
$[q_0 q_1]$	$[q_0 q_1]$	$[q_1 q_0]$

final state all containing q_0.

union of $([q_0], a)$ & $([q_1], a)$

same as $[q_0 q_1]$

Eg.

	a	b
→ q_0	$q_0\ q_1$	q_2
q_1	q_0	q_1
q_2		$q_0\ q_1$

NDFA

	a	b
→ $[q_0]$	$[q_0 q_1]$	$[q_2]$
$[q_2]$		$[q_0 q_1]$
$[q_0 q_1]$	$[q_0 q_1]$	$[q_2 q_1]$
$[q_2 q_1]$	$[q_0]$	$[q_0 q_1]$

DFA

Don't see which states are defined in NDFA

Scanned by CamScanner

eg.

	0	1	2
→ q_0	$q_0 q_4$	q_4	$q_2 q_3$
q_1		q_4	
q_2	q_2		q_3
q_3		$q_4 q_1$	
q_4	q_1	q_2	$q_1 q_2$

DFA:

	0	1	2
→ [q_0]	[$q_0 q_4$]	[q_4]	[$q_2 q_3$]
[$q_0 q_4$]	[$q_0 q_4 q_1$]	[$q_4 q_2$]	[$q_1 q_2 q_3$]
[q_4]	[q_1]	[q_2]	[$q_1 q_2$]
[$q_2 q_3$]	[q_2]	[$q_4 q_1$]	[q_3]
[$q_0 q_4 q_1$]	[$q_0 q_4 q_1$]	[$q_2 q_4$]	[$q_1 q_2 q_3$]
[$q_2 q_4$]	[$q_1 q_2$]	[q_2]	[$q_1 q_2 q_3$]
[$q_1 q_2 q_3$]	[q_2]	[$q_1 q_4$]	[q_3]
[q_1]		[q_4]	
[q_2]	[q_2]		
[$q_1 q_2$]	[q_2]	[q_4]	
[$q_1 q_4$]	[q_1]	[$q_2 q_4$]	[$q_1 q_2$]
[q_3]		[$q_1 q_4$]	

Eg.

	a	b
$\rightarrow q_0$	$q_1 q_2$	q_2
q_1	q_2	$q_1 q_3$
q_2	q_0	$q_1 q_0$
q_3	q_2	$q_1 q_2$

: NDFA

DFA :

	a	b
$\rightarrow [q_0]$	$[q_1, q_2]$	$[q_2]$
$[q_1, q_2]$	$[q_0, q_2]$	$[q_0, q_1, q_3]$
$[q_2]$	$[q_0]$	$[q_1, q_0]$
$[q_0, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_3]$	$[q_1, q_2]$	$[q_1, q_2, q_3]$
$[q_1, q_0]$	$[q_1, q_2]$	$[q_1, q_2, q_3]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$
$[q_1, q_2, q_3]$	$[q_0, q_2]$	$[q_0, q_1, q_2, q_3]$
$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$

17-1-18

NFA with Λ transition

NFA to DFA

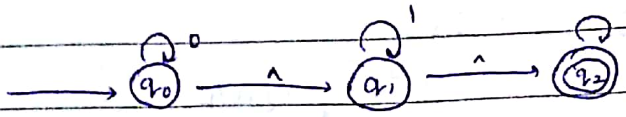
1. Λ - moves (remove)
2. NFA to DFA
3. Minimize it
4. Implement it

(jaha bhi hum q_2 se ja skte h, waha hum q_1 se bhi ja " ")

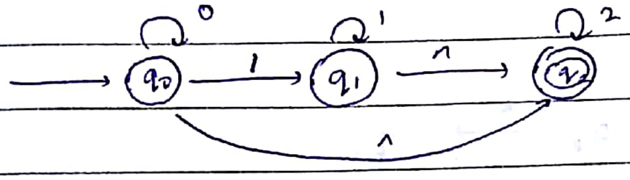
⇒ ③ $q_1 \xrightarrow{\wedge} q_2$

① Duplicate all moves from q_2 to q_1 itself

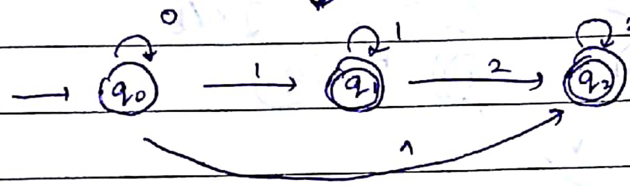
② If q_2 is FS, then make q_1 also as final state

Eg. 

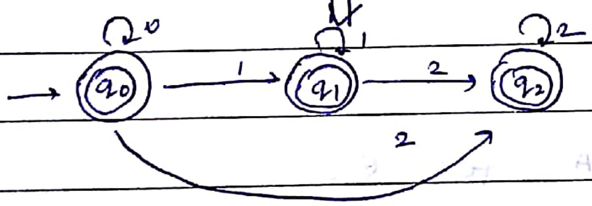
⇓ Remove $q_0 \xrightarrow{\wedge} q_1$



⇓ Remove $q_1 \xrightarrow{\wedge} q_2$



⇓



OR

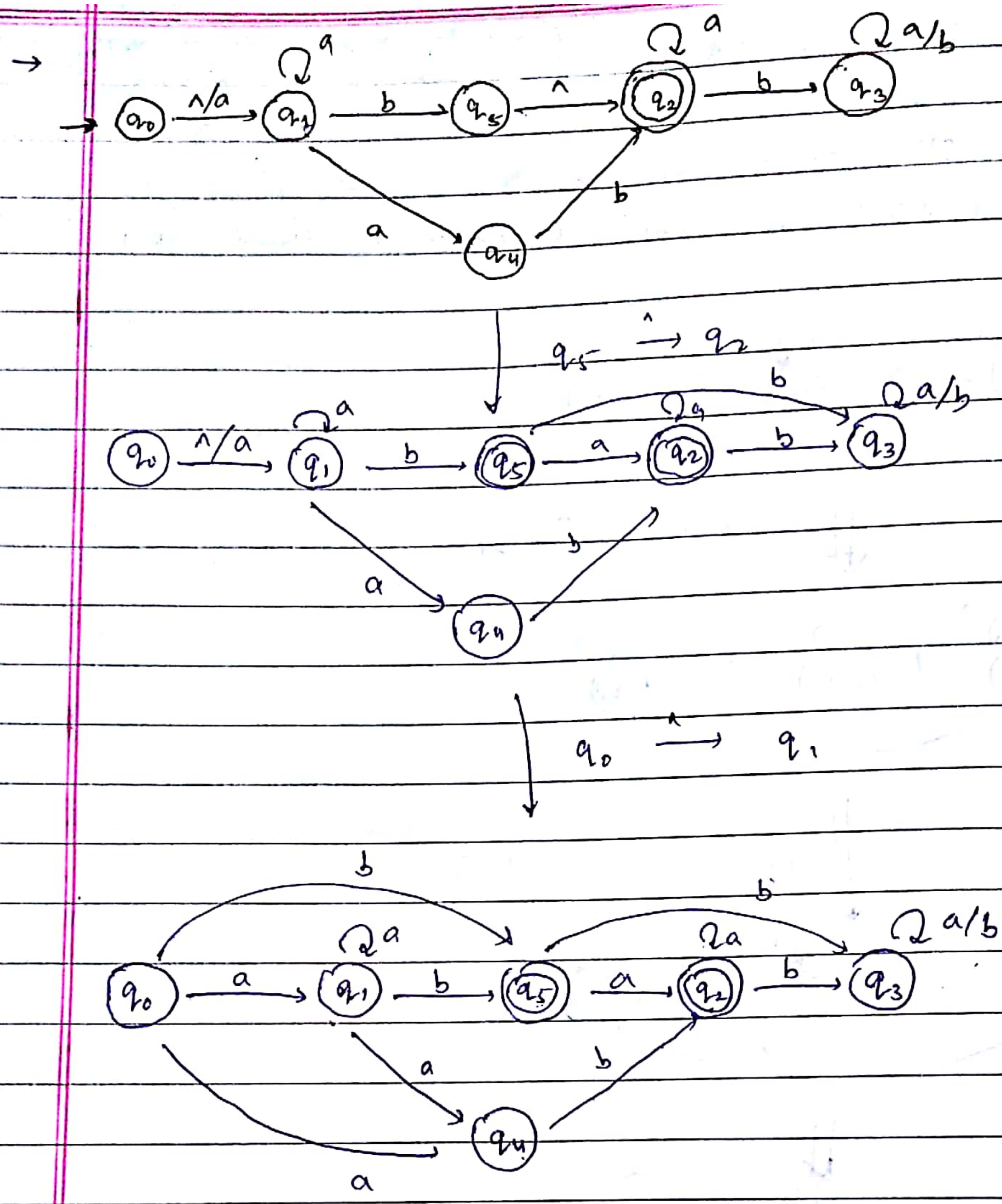
	0	1	2	$\wedge$
→ q_0	q_0	q_1		~~q_1~~ q_2
q_1		q_1		q_2
q_2			q_2	

→

Final table

	0	1	2	$\wedge$
→ q_0	q_0	q_1	q_2	
q_1		q_1	q_2	
q_2			q_2	

Remove all



→ How to check a NFA for ϵ.

ϵ- closure (q)

→ is the set of state possible to reach from q with ^ moves

ϵ - closure (q)

1. $q \in \epsilon\text{-closure}(q)$
2. if $\delta(q, \wedge) \to q_i$, then $q_i \in \epsilon\text{-closure}(q)$
3. $\forall\ p \in \wedge\text{-closure}(q)$ if $\delta(p, \wedge) \to r$ then $r \in \wedge\text{-closure}(q)$

δ(B, 101)

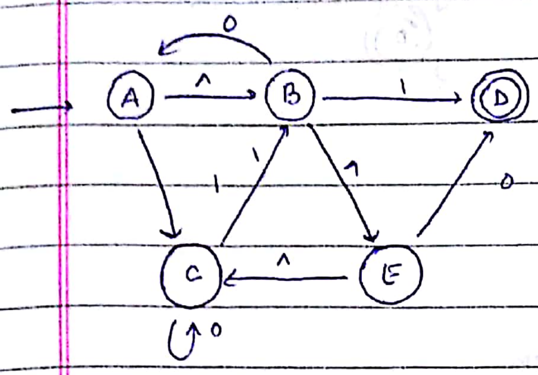
→ ({BEC}, 1) → {DB}

→ ({DBEC}, 0) → {ADC}

→ ({ABCED}, 1) → {CDB}

→ {CBDE}

Repeat 3 till no new state is added in ∧-closure(q)



ε-closure(A) = {A, B, E, C}

ε-closure(C) = {C}

→ δ(q, w)

ε-closure(q) $\xrightarrow{\delta(\varepsilon\text{-closure}(q),\, a)}$ P

↓

ε-closure(P) = R

↓ δ(R, a)

S

↓

ε-closure(S)

Eg.: δ(A, 01).

({A, B, E, C}, 01).

→ {A D C} → ({ABECD}, 1) → {CDB}

closure {ADC}

↓

{CDEB}

Eg. δ(B, 101)

→ {B, E, C} → (~~{...}~~ {DB}, 01) → {DBEC}. → ({ADC}, 1)

→ ~~{ACBECD}~~ {CB} → {BEC}

{BEC}, 1 → {CBDE}

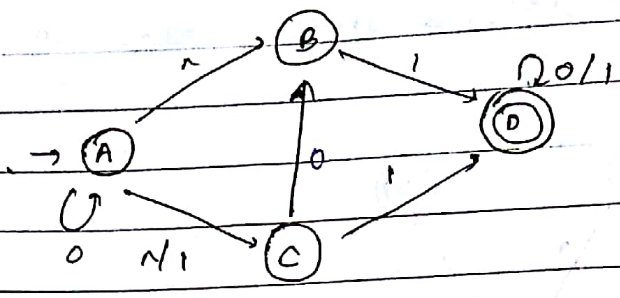
→ ({BBEC}, 0)

→ {AD~~E~~BE}, 1

DATE: / /

PAGE NO.:

eg.



Task 1 δ (A, 01)

Task 2 Remove ^ -move

Task 3 Convert NFA to DFA

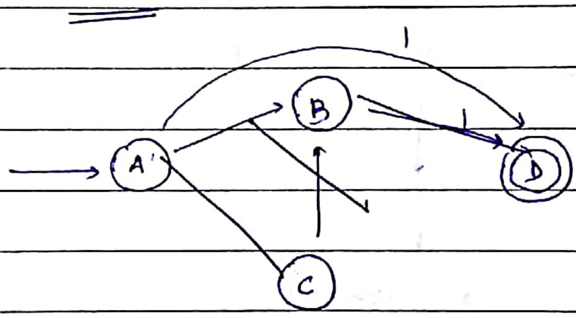
① → δ (A, 01)

→ ({A B, C}, 0) → δ{A B}

→ ({A B C}, 1) → {C D}

→ {C D}

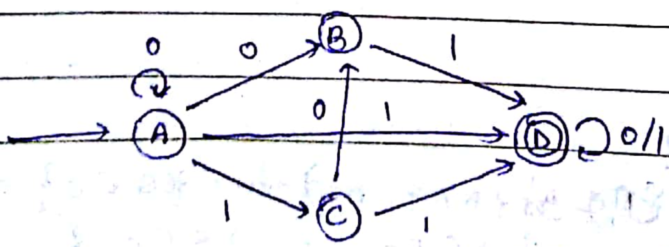
②



	0	1	^
→ A	A	C	BC
B		D	
C	B	D	
D	D	D	

→

	0	1	^
→ A	AB	CD	B.C
B		D	
C	B	D	
D	D	D	



Teacher's Signature

③ NFA to DFA

	0	1
→ A	AB	CD
B		D
C	B	D
D	D	D

: NFA

DFA :

	0	1
→ [A]	[AB]	[CD]
[AB]	[AB]	[CD]
[CD]	[BD]	[D]
[BD]	[D]	[D]
[D]	[D]	[D]

22-01-18

Minimization of Deterministic Finite Automata

✓ State Equivalence : q_1 and q_2 are equivalent iff

1.) $(q_1, x) \rightarrow q_f$ $\qquad q_f \in F$

then

$(q_2, x) \rightarrow q_f$ $\qquad \forall\ x \in \Sigma$

2.) $(q_1, x) \vdash q_j$ $\qquad q_j \notin F$

$(q_2, x) \vdash q_k$ $\qquad q_k \notin F$

	a	b	c
→ q_1	q_1 n.f.	q_2 n.f	q_4 F.S
q_2	q_1 n.f	q_3 n.f	q_4 f
q_3	q_4 f	q_2	q_4
q_4	q_1	q_1	q_4

⇒ q_1 & q_2 : state equivalent

don't consider for final state

Steps :

① Remove all dead and unreachable set

② Divide Q into two sets Q_1 and Q_2 containing set of NF and final state

③ 1.) check equivalence for every set in Q_1 and Q_2 if the state in set are not equivalent split the set. Repeat till all states in set are equivalent.

	0	1
→ q_0	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
~~q_3~~	~~q_2~~	~~q_6~~
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2
~~q_8~~	~~q_8~~	~~q_8~~

Reachable state :

q_0	q_6
q_1	q_7
q_2	
—	
q_4	
q_5	

→ dead state ⇒ remove (q_8)

→ unreachable state ⇒ q_3 : Remove

Scanned by CamScanner

Assume : Automata contain only 2 states :

N.F. Final

$P_1 (q_0, q_1, q_4, q_5, q_6, q_7)$ $P_2 (q_2)$ → single state ⇓ no split

$q_0, a \to P_1$

$q_0, b \to P_1$ $P_3 (q_0, q_4, q_6)$ $P_4 (q_1, q_5, q_7)$

$q_1, a \to P_1$

$q_1, b \to P_2$

$P_5 (q_0, q_4)$ $P_6 (q_6)$ $P_7 (q_1, q_7)$ $P_8 (P_5)$

Now, we have B_2 B_3 B_4

$q_0, a \to P_4$	$q_1, a \to P_6$	$q_0, a \to P_7$	$q_1, a \to P_6$
$q_0, b \to P_4$	$q_1, b \to P_2$	$q_0, b \to P_8$	$q_1, b \to P_2$
$q_4, a) \to P_4$	$q_5, a \to P_2$	$q_4, a \to P_7$	$q_7, a \to P_6$
$q_4, b \to P_4$	$q_5, b \to P_6$	$q_4, b \to P_8$	$q_7, b \to P_2$
$q_6, a \to P_3$	$q_7, a \to P_6$		
$q_6, b \to P_3$	$q_7, b \to P_2$		

q_0 & q_4 → equi q_1 & q_7 : equi.

	a	b
→ P_5	P_7	P_8
P_6	P_6	P_5
P_7	P_6	P_2
P_8	P_2	P_6
P_2	P_5	P_2

→ Final State isn't dead state

Ex.

	a	b
→ q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6
~~q_7~~	~~q_6~~	~~q_3~~

q_0
q_1
q_2
q_3
q_4
q_5
q_6

unreachable : ~~q_7~~

Q

P_1 ($q_0, q_1, q_2, q_4, q_5, q_6$) P_2 (q_3)

q_0 — q_1, P_1

q_1 — P_1, P_1

q_2

q_2 — P_2, P_3

P_3 (q_0, q_1, q_5, q_6) P_4 (q_2, q_4)

q_4 — P_2, P_3

q_2 — P_2, P_6

P_5 P_6

q

q_0 — P_3, P_3

P_5 (q_0, q_6) P_6 (q_1, q_5)

q_1 — P_3, P_4

q_5 — P_3, P_4

q_6 — P_3, P_3

q_0 — P_6, P_5

q_1 — P_5, P_4

q_6 — P_6, P_5

q_5 — P_5, P_1

Teacher's Signature

	a	b
→ P_5	P_6	P_5
P_6	P_5	P_4
P_4	P_2	P_6
P_2	P_2	P_5

→ q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0

Ans.

Ques

	a	b
→ q_0	q_0	$q_0 q_2$
q_1	q_3	$q_2 q_3$
q_2	q_4	–
q_3	q_4	–
q_4	–	–

(NFA → DFA first)

		a	b
A	→ $[q_0]$	$[q_0]$	$[q_0 q_2]$
B	$[q_0 q_2]$	$[q_0 q_4]$	$[q_0 q_2]$
C	$[q_0 q_4]$	$[q_0]$	$[q_0 q_2]$

Unreachable :

Q

	a	b
→ A	A	B
B	C	B
C	A	B

Q

Q_1 {A B} Q_2 {C}

Q_2 {A} Q_3 {B}

A → A, B

B → C, B

	a	b
→ A	A	B
B	C	B
(C)	A	B

29-01-18

Regular Expression :

→ it is an algebric notation for the language accepted by FA

→ it includes 3 symbols : +, ., *

+ : union

. : concatenation

* : Kleene closure

↳ a^* = 0 or more occure of a

$L = a^*$
$= \{\wedge, a, aa, aaa, \ldots\}$

↳ $(a+b)$: either a or b

↳ $a \cdot b$

① L = all string containing abb as substring
$\Sigma = \{a, b\}$

all strings possible from a & b

$(a+b)^* = \{\wedge, a, b, ab, aa, bb, ba, aaa, \ldots\}$

(a+b)(a+b)

$(a+b)^* \; abb \; (a+b)^*$

(2) all string ~~co~~ ending with <u>abb</u>

$(a+b)^* \; abb$

(3) starting with <u>abb</u>

$abb \; (a+b)^*$

(4) even no. of a's.

$(b^* a b^* a b^*)^*$ or $b^* (ab^*a)^* b^*$

$\downarrow$

not interested in b

(5) atleast 3 a's

~~aa~~a

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

(6) exactly 3 a's

$b^* a b^* a b^* a b^*$

Q. $L_1 = a \cdot b + c$

$L_2 : (a \cdot b) + c$

NO $*$ $\Rightarrow$ finite set

L_1 :

$a \cdot b + c$

$a \cdot (b+c)$ $\quad$ $(ab + c)$

$\downarrow$ $\quad$ $\downarrow$

(ab, ac) $\quad$ (ab, c)

Teacher's Signature

* Precedence graph:
  → Star
  → Concatenation
  → Union

* L_1 : $a \cdot b + c \Rightarrow (ab) + c$
  $= (ab, c)$

Eg. $L = a^{2m} b^{2n+1} a\, b \quad m > 0 \quad n > 0$

→ This is not in regular expression

⇒ even no. of a, then odd no. of b followed by a b

R_1 : $(aa)^* (bb)^* b \quad a \cdot b$ ✗ [because ~~m, n > 0~~]
R_2 : $aa\,(aa)^*\, bb\,(bb)^*\, b\, a\, b$ ✓

1st string possible : $a^2 b^3 a\, b$
so, R_2 is correct.

* a^* : { 0 or more } Kleene Closure
  $a a^*$: { 1 or more }
  $a a^* = a^+$: Positive Kleene's Closure

$$\boxed{R \cdot R^* = R^+}$$

Eg. atleast 1 pair of consecutive zero
$\Sigma = 0,1$

$$(0+1)^*(00)^*(0+1)^*$$

*** Eg. has no pair of consecutives zeroes ⇒ whenever 0 is occurring, it should be followed by 1.

$$(1+01)^* \ (0+\wedge)$$

$(1+01)^*$: either 1 or 01 will appear

$(0+\wedge)$: can have single zero or not zero

Kleene Theorem

→ For every R.E., there exists a FA to accept it.

Lang
↓
R.E
↓
ε-NFA
↓
NFA
↓
DFA
↓
Minimize

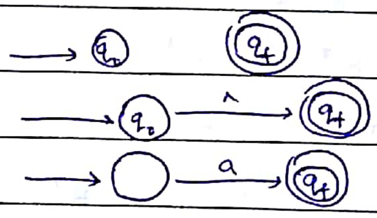
Proof : Based on Induction

Basic:

$R = \phi$

$R = \wedge$

$R = a$

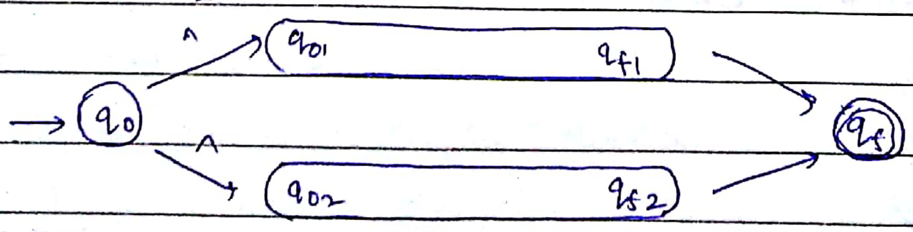


let us assume there exist FA for ~~R.E~~ R.E. R_1 and R_2

R_1 : it is $M_1(Q_1, q_{01}, \Sigma_1, \delta_1, q_{f1})$

R_2 : $M_2(Q_2, q_{02}, \Sigma_2, \delta_2, q_{f2})$

1.) $R = R_1 + R_2$



Teacher's Signature

$M'(Q, \Sigma, \delta, q_0, q_f)$

$Q = Q_1 \cup Q_2 \cup \{q_0, q_f\}$

$\Sigma = \Sigma_1 \cup \Sigma_2$

$\delta = \delta_1 \cup \delta_2 \cup \quad q_0, \wedge \rightarrow \{q_{01}, q_{02}\}$

$\cup \quad q_{f_1}, \wedge \rightarrow q_f$

$\cup \quad q_{f_2}, \wedge \rightarrow q_f$

30-01-18

Finite Automata to Regular Expression

Identities :

symbol

$\wedge$: Empty String

ϕ : Null (No transition)

set [Not even a single $\wedge$]

1. $\phi + R = R$
2. $\phi \cdot R = \phi$
3. $\wedge \cdot R = R$
4. $\wedge^* = \wedge$
5. $R^* R = R R^* = R^+$
6. $R^* R^* = R^*$
7. $R + R = R$
8. $\wedge + R R^* = R^*$ $\quad [R R^* = R^+, \; R^+ + \wedge = R^*]$
9. $(PQ)^* P = PQPQ \quad PQP = P(QP)^*$
10. $(P + Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$
11. $(P+Q) R = PR + QR$

Arden Theorem :

$R = Q + RP$ and $P \neq \wedge$

then $R = QP^*$

8th → 17th Feb

DATE: / /
PAGE NO.:

Proof. 1.) let $R = QP^*$

$R = Q + RP$

$= Q + QP^*P$

$= Q(\Lambda + P^*P) = QP^*$ I_8

2.) $R = Q + RP = Q + (Q + RP)P$

$= Q + QP + RP^2$

$= Q + QP + P^2(Q + RP)$

$= Q + QP + QP^2 + RP^3$

$\vdots$

$= \underbrace{(Q + QP + QP^2 + \dots + QP^i)}_{A} + \underbrace{(RP^{i+1})}_{B}$

let $w \in R$ and $|w| = i$

$\therefore R \in (Q + QP + \dots + QP^i)$

or

$R \in RP^{i+1}$

$P \not\ni \Lambda \Rightarrow |RP^{i+1}| \geq i+1$ [~~each P is atleast of~~ length 1]

Therefore $w \notin RP^{i+1}$ [$\because |w| = i$]

$\Rightarrow w \in (Q + QP + QP^2 + \dots + QP^i)$

$\Rightarrow R = Q(\Lambda + P + P^2 + \dots\dots + P^i)$

$R = QP^*$

Hence Proved

Method : FA to RE

Write eq^n for each state $q_i \in Q$ describing its reachability from other states

$q_1 = \alpha_{11} q_1 + \alpha_{12} q_2 + \alpha_{13} q_3 + \dots + \alpha_{1n} q_n$

$q_2 = \alpha_{21} q_1 + \alpha_{22} q_2 + \alpha_{23} q_3 + \dots + \alpha_{2n} q_n$

$\vdots$

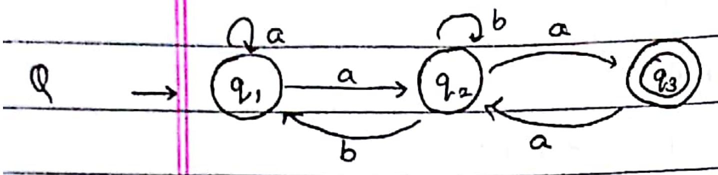
$q_n = \alpha_{n1} q_1 + \alpha_{n2} q_2 + \alpha_{n3} q_3 + \dots + \alpha_{nn} q_n$

Teacher's Signature

Apply identities and Arden Theorem & generate

If $q_f = q_0 \alpha_{ij}$ Remove q_0

This is final ~~automata~~ answer



(How to reach q_i)

$q_1 = aq_1 + q_2 b$ — (1)

$q_2 = q_1 a + q_2 b + q_3 a$ — (2)

$q_3 = q_2 a$ — (3)

finally, get in form of $q_3 = q_1 \alpha_{ij}$

~~$q_3 = q_2 - q_1$~~

(1) & (2) : $q_2 - q_1 = q_3 a$

$\frac{q_3}{a} - q_1 = q_3 a$

q_3

(2) & (3) :

$q_2 = q_1 a + q_2 b + q_2 aa$

$\underbrace{q_2}_{R} = \underbrace{q_1 a}_{Q} + \underbrace{q_2}_{R} \underbrace{(b + aa)}_{P}$

Arden's theorem : $q_2 = q_1 a (b+aa)^*$ — (4)

(1) & (4)

$q_1 = q_1 a + q_1 a (b+aa)^* b$

$\underbrace{q_1}_{R} = \underbrace{q_1}_{R} \underbrace{(a + a(b+aa)^* b)}_{P} + \underbrace{\Lambda}_{Q}$ $(\because R + \Lambda = R)$

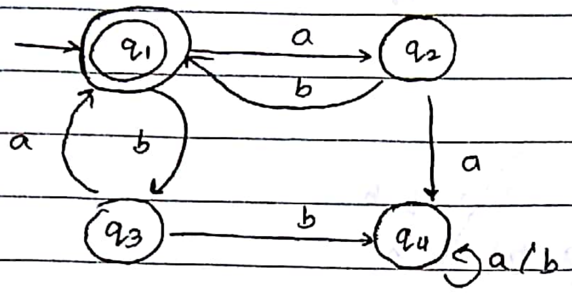
Teacher's Signature

Multiple final set : perform for each, take union

DATE : / /
PAGE NO. :

Arden :
$$q_1 = \wedge\,(a + a(b+aa)^* b)^* \quad \text{— (5)} \qquad [\because R\cdot\wedge = R]$$
$$= (a + a(b+aa)^* b)^*$$

(3) : $q_3 = q_1\, a\,(b+aa)^*a$
$$= (a + a(b+aa)^* b)^*\, a\,(b+aa)^*\, a$$

Q. 

$q_1 = q_2 b + q_3 a$

$q_2 = q_1 a$

$q_3 = q_1 b$

$q_4 = q_4 a + q_4 b$

$q_1 = q_1 ab + q_1 ba$

$q_1 = q_1 (ab)$

$= ab\,[q_1 (ab)]$

$= (ab)^* q_1$

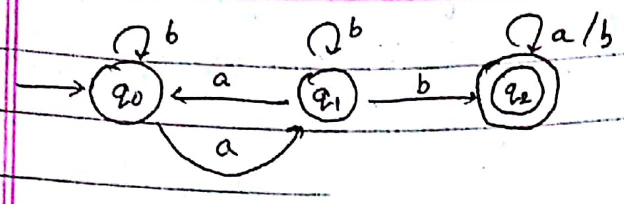
$q_1 = q_1 ab + q_1 ba$

$$\underset{R}{q_1} = \underset{R}{q_1}\,\underset{P}{(ab+ba)} + \underset{Q}{\wedge}$$

Arden : $q_1 = \wedge\,(ab+ba)^*$

$\rightarrow q_1 = (ab+ba)^*$

Q.



$q_0 = q_0 b + q_1 a$ —— ①

$q_1 = q_0 a + q_1 b$ —— ②

$q_2 = q_1 b + q_2 a + q_2 b$ —— ③

$q_2 = (\ )\ q_0$

~~$q_0 = q_0 b + q_1 a$~~

① + ② $\Rightarrow$ $q_0 + q_1 = q_0 b + q_0 a + q_1 a + q_1 b$

$\Rightarrow \underbrace{q_0 + q_1}_{R} = \underbrace{(q_0 + q_1)}_{R}\underbrace{(a+b)}_{P} + \underbrace{\wedge}_{Q}$

$\Rightarrow q_0 + q_1 = \wedge (a+b)^*$

$q_0 + q_1 = (a+b)^*$ —— ④

$\underbrace{q_2}_{R} = \underbrace{q_1 b}_{Q} + \underbrace{q_2}_{R}\underbrace{(a+b)}_{P}$

$\Rightarrow q_2 = q_1 b (a+b)^*$ —— ⑤

$\Rightarrow$ $q_0 = q_0 b + (q_0 a + q_1 b) a$

$q_0 = q_0 b + q_0 a \cdot a + q_1 b a$

$\underbrace{q_0}_{R} = \underbrace{q_0}_{R}\underbrace{(b + a \cdot a)}_{P} + \underbrace{q_1 b a}_{Q}$

$q_0 = q_1 b a (b + aa)^*$ —— ⑥

⑥ & ②

$\underbrace{q_1}_{R} = q_0 a + q_1 b$

$\Rightarrow q_1 = q_1 b a (b+aa)^* a + q_1 b$

$\Rightarrow \underbrace{q_1}_{R} = \underbrace{q_1}_{R}\underbrace{(ba(b+aa)^* a + b)}_{P} + \underbrace{\wedge}_{Q}$

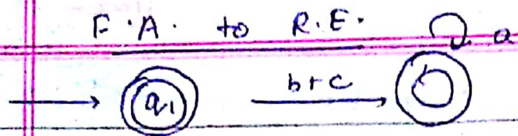
$\Rightarrow q_1 = (ba(b+aa)^* a + b)^*$ —— ⑦

Teacher's Signature

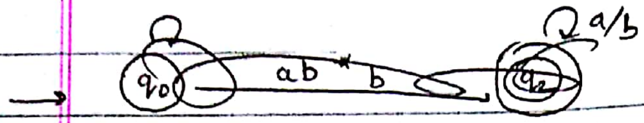
Scanned by CamScanner

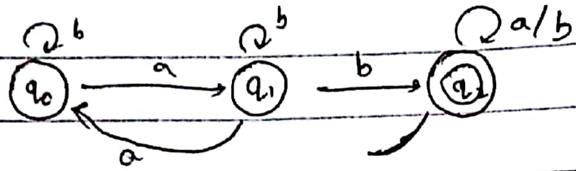
DATE: 31/1/18
PAGE NO.:

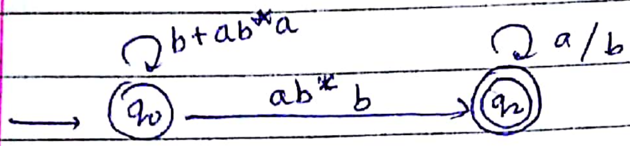
F.A. to R.E.



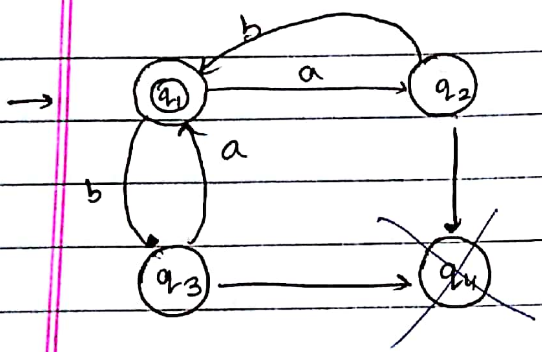
$(a^* + (b+c)a^*)^*$

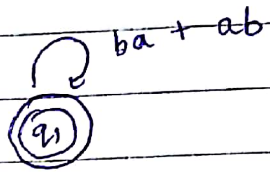






$(b+ab^*a)^* ab^* b (a+b)^*$





$(ab + ba)^*$

Closure Properties on regular language (If L_1 & L_2 : R.L. any opⁿ will give R.L.)

L_1 & L_2 are R.L.

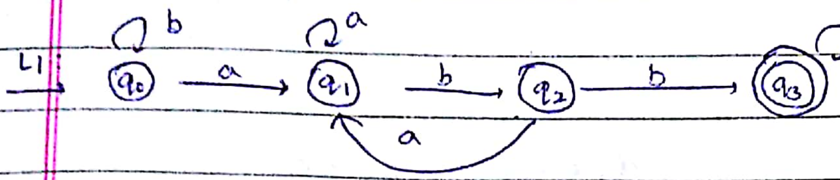
1) Union $L = L_1 \cup L_2$ ✓
2) Concatenation $L = L_1 \cdot L_2$ ✓
3) Kleene closure $L = L_1^*$ ✓
4) Reversal :
$L = L_1^R$

$L_1 = (ab)^*$
$L_1^R = (ba)^*$

Procedure :

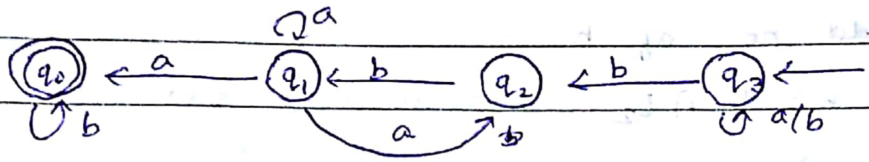
① Reverse the move
② Make IS as FS

③ Make FS as IS, if there are more than 1 FS then a new IS & ∧ - move to all FS

L1:


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

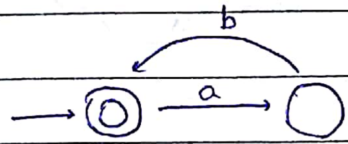
Reversal : L_1^R :

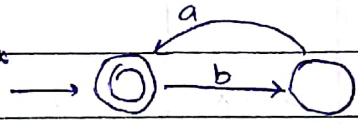


$(a+b)^* bba (a+b)^*$

5.) Complement :

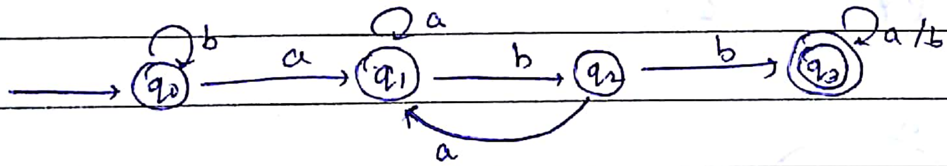
$L = \overline{L_1}$

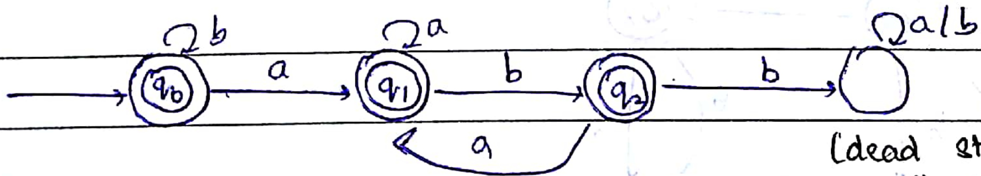
$(ab)^*$


$(ba)^*$


Procedure :

1. Make your F.S. as non-F.S.
2. Make all non-F.S. as F.S.

L1 :


Complement :


(dead state)
⇓
will reject all strings containing 'abb'.

6.) Intersection : (AND)

$L = L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}} \equiv$ Regular language

$L_1 = \{q_{01}, q_{f1}, \Sigma, \delta_1, Q_1\}$
$L_2 = \{q_{02}, q_{f2}, \Sigma, \delta_2, Q_2\}$

Procedure :

→ Make $[q_{01}, q_{02}]$ as initial state for F A L.

→ $\delta[q_{1i}, q_{2i}], a = \{\delta_1(q_{1i}, a), \delta_2(q_{2i}, a)\}$

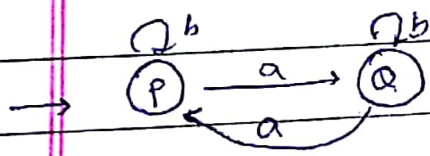
→ $[q_{f1}, q_{f2}] \in FS$

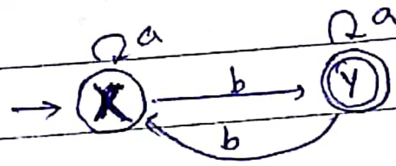
→ $Q = [q_i, q_j]$; $q_i \in Q_1$, $q_j \in Q_2$

Eg. L_1 : even no. of a

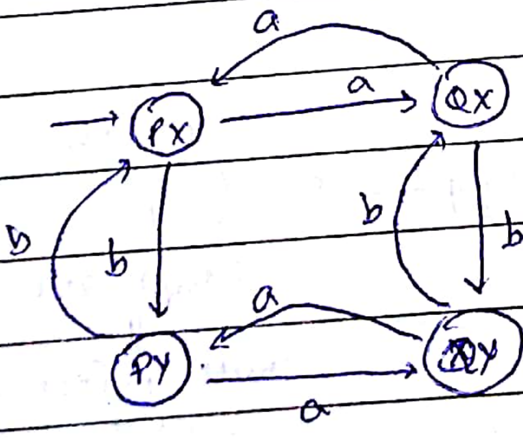
L_2 : odd no. of b

$L = L_1 \cap L_2 \equiv$ even a & odd b





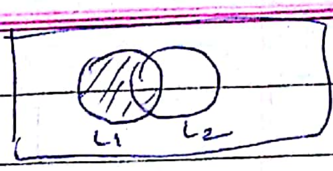
	a	b
→ PX	QX	PY
QX	PX	QY
PY	QY	PX
QY	PY	QX



7) Set Difference :

$L = L_1 - L_2$

$= L_1 \cap \overline{L_2}$



01/02/18

Pumping Lemma :

(Based on Pigeonhole principle)

we don't accept automata based on length of string

we accept automata based on property it follows

If L is Regular then each string $w \in L$ can be written as :

xyz ie, $w = xyz$

where $|y| \geq 1$, $|w| \geq m$

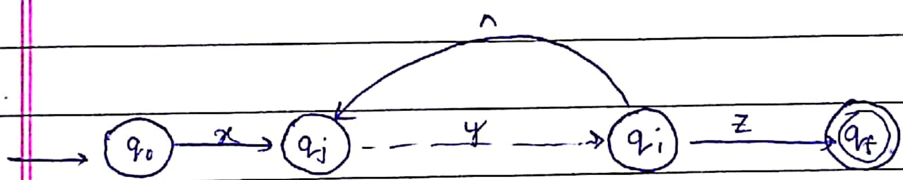
and $|xy| \leq m$

[string length : $\geq m$
no. of states $= m$]

then $\forall\ i \geq 0$ $\underline{xy^iz \in L}$

→ Assuming y is repeated, not x & z.

string length : infinite
⇒ some part is repeated
⇓
y here



If we can find such y ⇒ it is regular
can't " ⇒ not "

Eg $L = 0^n 1^n$

Automata

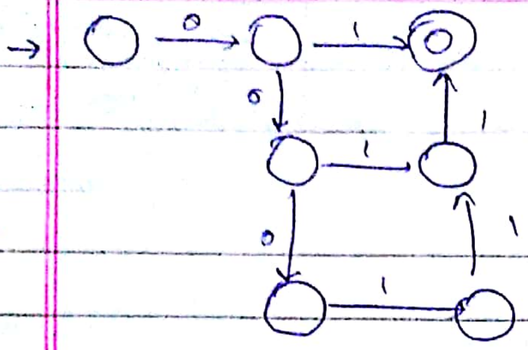
* when not bothered about a symbol ∴ we do self-loop for it

→ F.A. : Every a is remembered by state

$n=1$ 01

$n=2$ 0011

$n > 0$



↳ string infinite

→ no. of states : ∞

→ Not Regular language

$|Q| = K$

$w = xyz = 0^m 1^m = L$

If we find any i when $\notin L \Rightarrow$ Not Regular

let $y = 0$ $\quad |w| = 2m > K$

$K \geq M$

$\Downarrow$

$|y| \geq 1$

$w = \underset{x}{0^{m-1}} \; \underset{y}{0} \; \underset{z}{1^m}$

K: no. of states any string is accepted with max. states K.

$i = 0 \quad w = 0^{m-1} 1^m \notin L$

$i = 1 \quad w = 0^m 1^m$

$i = 2 \quad w = 0^{m-1} 0^2 1^m = 0^{m+1} 1^n \notin L$

→ $y = 1$

$0^m 1 \, 1^{m-1}$

$i = 2 \quad 0^m 1^2 1^{m-1} \notin L$

→ $y = 01$

$i = 2 \quad 0^{m-1} (01)^2 y^{m-1}$

$0^{m-1} 01 \, 01 \, y^{m-1} = 0^m 101^m \notin L$

→ we can't find any such y which makes it L.

* check for all possibilities of y.
If you find any y ~~which~~ for which we can iterate ~~~~ for all i → It is regular language

DATE: / /
PAGE NO.:

Q. $L = 0^{2n}$ $n \geq 0$

$y = 00$

$x = \wedge$ → have to make no. of states finite

$z = 0^{2(n-1)}$

$xy^iz \in L$

$i = 0$ ✓

$i = 1$ ✓

$i = 2$ ✓ → $(00)^2 (0)^{2(n-1)} = 0^{2(n+1)} \in L$

→ $L = 0^p 1^p$ $p > 0$ $|Q| = n$

$|w| = m \geq n$ m : length of string

$w = xyz$ $|y| \geq 1$ $|xy| \leq n$

and xy^iz

5/2/17

Regular Grammar

$G(V, \Sigma, P, S)$

V → is set of NT (Non-terminal) [Represent class]

Σ → set of terminal

P → production

S → start symbol

$S \longrightarrow NVN$ ⇒ $V = \{S, N, V\}$

$N \longrightarrow$ Ram | Apple | Book $\Sigma = \{$Ram, apple, book, ate, read$\}$

$V \longrightarrow$ ate | read

Eg. Ram ate apple

$S \rightarrow N\ V\ N$

$\rightarrow$ Ram V N

$\rightarrow$ Ram ate N

$\rightarrow$ Ram ate apple $\rightarrow$ this belongs to grammer.

Regular gramer :

a grammer G is regular if all the productions are of form

[NT should be 1, either in xtreme left or in xtreme right]

$A \rightarrow a \mid aB$ (Right Linear)

or

$A \rightarrow a \mid Ba$ (Left Linear)

Eg.

$S \rightarrow abA$

$A \rightarrow a \cdot A$

$A \rightarrow a$

$S \rightarrow abA$

$S \rightarrow aba$

$S \rightarrow abA$

$S \rightarrow abaA$

$S \rightarrow abaaA$

$L = abaa^*$

Eg.

$S \rightarrow Aba$

$A \rightarrow a$

$A \rightarrow Ba$

$B \rightarrow Ba$

$B \rightarrow a$

Scanned by CamScanner

S → Aba $(a\cancel{+b})^*$ aba

S → aba —

S → Aba

S → Baba   S → Baba

S → aaba —

Eg. S → aA

A → ~~aB+a~~   A → Bb | b

B →   B → aB | a

S → aA

S → ab   S → aBb

S → aab   S → aaBb

S → aaaBb

~~a*ab~~ a^*ab

Eg. S → aA

A → Bb | b

B → aA

S → aA

S → ab   S → aBb

$\boxed{L = a^n b^n, n > 0}$

↓

Not Regular Language

S → aaAb

S → aabb   S → aaBbb

S → aaaAbb

S → aaabbb

Teacher's Signature

* If some in left linear & some in right linear : grammer may be ~~linear~~ regular or not.

* If all are left linear (or right linear) ∴ grammer is regular

FA to Regular Grammer (NFA / DFA)

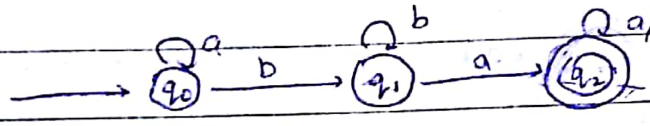
$\delta(q_i, a) \rightarrow q_j$ $\quad$ P : $A_i \rightarrow aA_j$

when $q_j \notin F$

$\delta(q_i, a) \rightarrow q_j$ $\quad$ P : $A_i \rightarrow aA_j \mid a$

when $q_j \in F$

Eg.



P : $A_0 \rightarrow aA_0$

$A_0 \rightarrow bA_1$

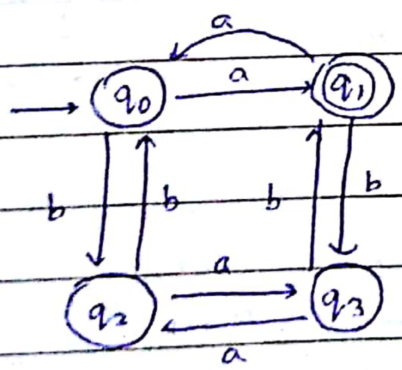
$A_1 \rightarrow bA_1$

$A_1 \rightarrow aA_2 \mid a$

$A_2 \rightarrow aA_2 \mid a$

$A_2 \rightarrow bA_2 \mid b$

Eg.



P : $A_0 \rightarrow aA_1 \mid a$

$A_0 \rightarrow bA_2$

$A_1 \rightarrow aA_0$

$A_1 \rightarrow bA_3$

$A_2 \rightarrow bA_0$

$A_2 \rightarrow aA_3$

$A_3 \rightarrow aA_2$

$A_3 \rightarrow bA_1 \mid b$

<u>Regular Grammar to FA</u>

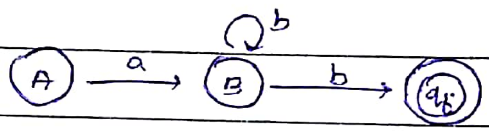
$A_1 \rightarrow aA_2$ then $\delta(q_1, a) \rightarrow q_2$

$A_1 \rightarrow a$ then $\delta(q_1, a) \rightarrow q_f$ (make new state)

$q_f \in F$

Eg. $A \rightarrow aB$

$B \rightarrow bB \mid b$



Eg. $A \rightarrow aB \mid bC$

$B \rightarrow bB \mid b$

$C \rightarrow aA \mid a$