TACD

PAGE No. / DATE / /

* Symbol :- a, Σ, q_0, q_0 . . . .
* Alphabets :- $\{a, b\}$; $\{0, 1\}$
* Strings :- a, ab, ba, b, aaa, - - - -
* Language :- No. of a's is even → aa, aaaa - - -

** Σ °/E (epsilon) → length of string is zero.
* Σ^1 → length of string is one.
* Σ^* → $\Sigma^1 * \Sigma^2 * \Sigma^3 \ldots$ // Kleen closure.

eg In
* Language → Valid strings are part of language.
* Set of valid codes is string.

Finite Automata (FA)

- FA with o/p
  - Moore
  - Mealy
- FA without o/p
  - DFA (Deterministic FA)
  - NFA (Non deterministic FA)
  - ϵ-NFA

* DFA :-

Defⁿ :- $(Q, \Sigma, \delta, q_0, F)$

- Q : set of all states → $\{A, B, C\}$
- Σ : set of input alphabets $\{a, b\}$
- δ : transition fn (δ) representation → $Q \times \Sigma = Q$
- q_0 : starting state eg → a
- F : set of All final state eg → $\{B\}$

eg :- a → (A) → ((B)) a, b

(C) a, b

[Q ≥ F]

In DFA with one symbol it will go to only one state

PAGE No.
DATE / /

Question1 Construct a DFA that accepts set of all strings over a, b of length 2.

Sol^n:- $\Sigma = \{a, b\}$

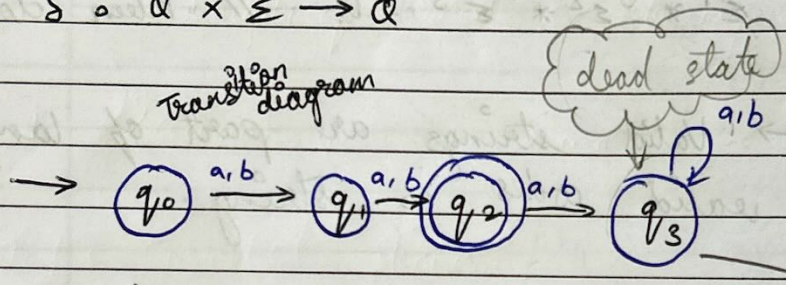
Language = $L = \{aa, ab, ba, bb\}$

$(Q, \Sigma, \delta, q_0, F)$

$\delta : Q \times \Sigma \to Q$

a,b → a or b

Transition diagram



length is given 2 so q_2 is final state.

If anything greater than 2 comes it will be transfered to death state

Transition table

Q \ Σ	a	b	
q_0	q_1	q_1	q_0 goes to q_1
q_1	q_2	q_2	q_1 goes to q_2
* q_2	q_3	q_3	$q_2 \to q_3$
q_3	q_3	q_3	q_3 goes to q_3.

* indicates final state or ⊛.

Final = $F = \{q_2\}$

$Q = \{q_0, q_1, q_2, q_3\}$

PAGE No.
DATE / /

Simulation :-

i) Transition → δ

$$\delta(q_0, ab) = \delta(q_1, b)$$

(q_0 at $a \rightarrow q_1$)

$= q_2$

= Final state → accepted

ii)

$$\delta(q_0, aab) = \delta(q_1, ab)$$
$$= \delta(q_2, b)$$
$$= q_3$$
= dead state
= Rejected

Finite automata is said to accept a language if all the strings in that language are accepted and rejected by the string which are not in language.

Q 2 Design DFA where length of word is greater than and equal to over a, b.

soln :-

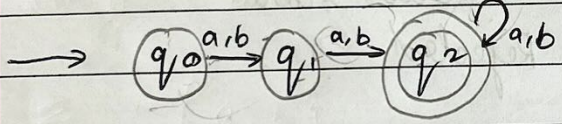
$\Sigma = \{a, b\}$ aa aaa

$|w| \geq 2$

↓ length of word or string.

$L = \{aa, ab, ba, bb, aaa, \ldots\}$

Take smallest string.



less than final is also reject

Q \ Σ	a	b
q_0	q_1	q_1
q_1	q_2	q_2
$*q_2$	q_2	q_2

$F = \{q_2\}$

$Q = \{q_0, q_1, q_2\}$

Simulation :-

Transition → $\delta(q_0, abb) \Rightarrow \delta(q_1, bb) = \delta(q_2, b)$

$= q_2$

= Final state

= accepted

Q3 Design DFA where length of the word is less than 2. over a, b

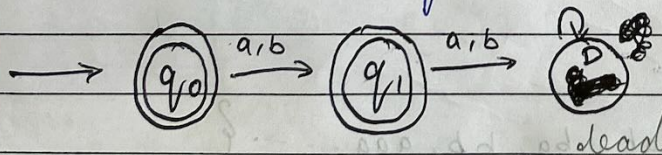
Sol^n :-

$\Sigma = \{a, b\}$

$|w| < 2 \longrightarrow 0, 1$ → 0, 1

$L = \{\epsilon, a, b\}$ E → 0

→ Take smallest string.



Whenever there is E take first state as final state.

δ / Σ	a	b
q_0	q_1	q_1
q_1	D	D
D	D	D

$F = \{q_0, q_1\}$

$Q = \{q_0, q_1, D\}$

PAGE No.
DATE / /

Transition → $\delta(q_0, ab)$

i) $= \delta(q_1, b)$

$=$ ~~D~~ Dead state rejected

ii) $\delta(q_0, \epsilon)$

$= q_0$

$=$ Final state

$=$ Accepted

Design DFA where length of string is mod 2 = 0

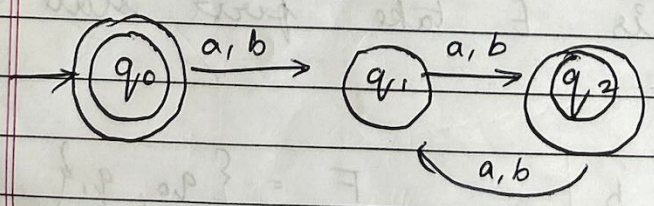
$\Sigma = \{a, b\}$

$|w| \bmod 2 = 0$ $\quad 0 \% 2 = 0$

length of string is divisible by 2.

Sol:-

$L = \{\epsilon, aa, ab, ba, bb, aaa \ldots\}$



Transition :-

→ $\delta(q_0, ab)$

$\delta(q_1, b)$

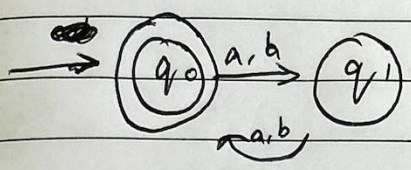
$= q_2$

$=$ Final state

$=$ accepted

PAGE No.
DATE / /

Minimum DFA :-



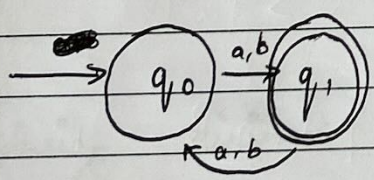
Q \ Σ	a	b
* q_0	q_1	q_1
q_1	q_0	q_0

Design DFA where remainder is one.

$\Sigma = \{a, b\}$ $\quad 1\%2 = 1, 3, 5,$

$|w| \bmod 2 = 1$

$L = \{ a, b, aaa, \ldots \}$



Q \ Σ	a	b
q_0	q_1	q_1
* q_1	q_0	q_0

$F = \{q_1\}$

$Q = \{q_0, q_1\}$

Transition :- $\rightarrow \delta(q_0, abb)$

$\rightarrow \delta(q_1, bb)$

$\rightarrow \delta(q_0, b)$

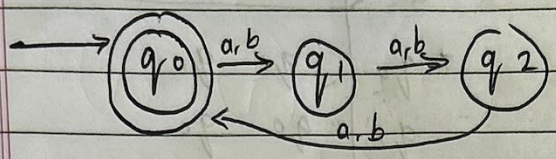
$\rightarrow q_1$

$\rightarrow$ = Final state

$\rightarrow$ = accepted

PAGE No.
DATE / /

|w| mod 3 = 0



$F = \{q_0\}$

$Q = \{q_0, q_1, q_2\}$

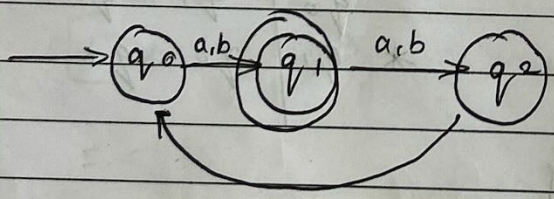
Q \ Σ	a	b
* q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_0	q_0

Transition :- $\delta(q_0, aab) \rightarrow \delta(q_1, ab) \rightarrow \delta(q_2, b)$

$= q_0 \rightarrow$ Final state

= Accepted

|w| mod 3 = 1



0 1 2

3 4 5

$F = \{q_1\}$

$Q = \{q_0, q_1, q_2\}$

Transition :- $\delta(q_0, aaba)$

$\rightarrow \delta(q_1, aba)$

$\rightarrow \delta(q_2, ba)$

$\rightarrow \delta(q_0, a)$

$\rightarrow q_1$

= Final state

= Accepted

Q \ Σ	a	b
q_0	q_1	q_1
* q_1	q_2	q_2
q_2	q_0	q_0

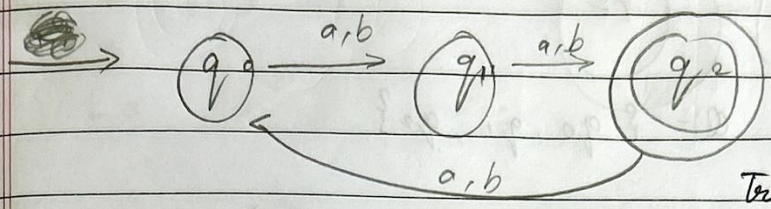
0000

1110

PAGE No.
DATE / /

$|w| \bmod 3 = 2$

Q / Σ	a	b
q_0	q_1	q_1
q_1	q_2	q_2
$*q_2$	q_0	q_0



$F = \{q_2\}$

$Q = \{q_0, q_1, q_2\}$

Transition :- $\delta(q_0, ab)$
$\rightarrow \delta(q_1, b)$
$\rightarrow q_2$
$\rightarrow$ Accepted

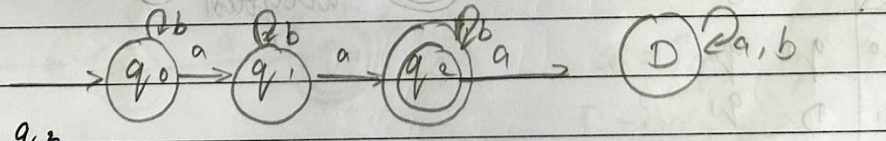
Important Note :-

→ If number is divisible by 2 ; two states are required.

→ If divisible by 3 then three states are required.

Design minimum DFA over input alphabet a, b where $n_a(w) = 2$ → number of a = 2.

$L = \{aa, aab, baa, aba, \ldots\}$



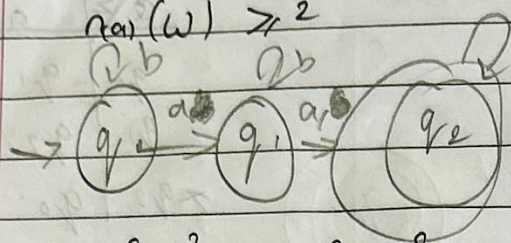
$F = q_2$

$Q = \{q_0, q_1, q_2, D\}$

Q / Σ	a	b
q_0	q_1	q_0
q_1	q_2	q_1
$*q_2$	D	q_2
D	D	D

Transition :-
$= \delta(q_0, abbbbab)$
$= \delta(q_1, bbbbab)$
$= \delta(q_1, ab)$
$= \delta(q_2, b)$
$= q_2$
= Final state

2) $n_a(w) \geq 2$



$F = \{q_2\}$ $\quad Q = \{q_0, q_1, q_2\}$

Q \ Σ	a	b
q_0	q_1	q_0
q_1	q_2	q_1
* q_2	q_2	q_2

Simulation $\delta(q_0, aab)$

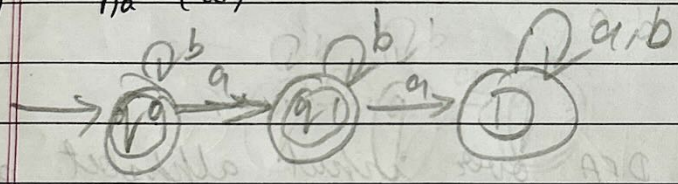
$\rightarrow \delta(q_1, ab)$

$\rightarrow \delta(q_2, b)$

$\rightarrow q_2$

Accepted.

3) $n_a(w) < 2$



$L = \{b, ab, abb, \ldots\}$

$F = q_0, q_1$

$Q = \{q_0, q_1, D\}$

$\delta(q_0, b)$

$\rightarrow q_0$

$\rightarrow$ accepted

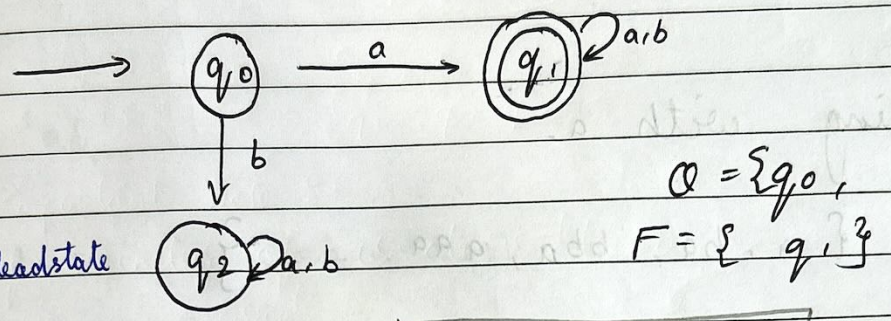
Q \ Σ	a	b
q_0	q_1	q_0
q_1	D	q_1
D	D	D

PAGE No.
DATE / /

Design a DFA that accepts a set of all strings starting with a over input alphabet a, b.

Sol^n:-

$L = \{a, aa, ab, abbb, \ldots\}$



$Q = \{q_0, q_1, D(q_2)\}$
$F = \{q_1\}$

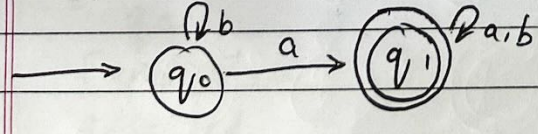
Transition:-
$\delta(q_0, abb)$
$\delta(q_1, bb)$
$= q_1$
= Final state
= accepted

Q / Σ	a	b
q_0	q_1	q_2
* q_1	q_1	q_1
q_2	q_2	q_2

Set of all strings containing a.

Sol^n:-

$L = \{a, aa, ab, abbb, ba, \ldots\}$



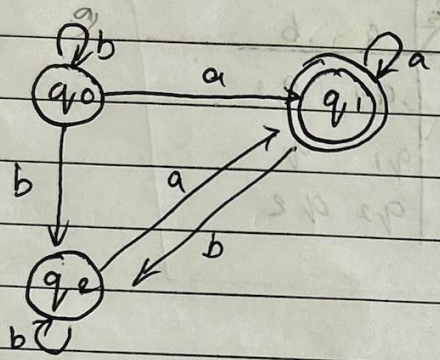
$Q = \{q_0, q_1\}$
$F = \{q_1\}$

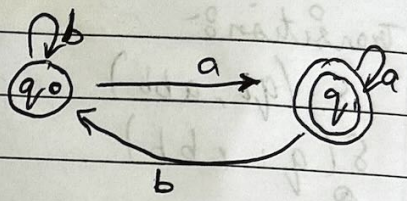
$\delta(q_0, baabb)$
$\delta(q_1, bb)$
$= q_1$
= Final state
= accepted

Q / Σ	a	b
q_0	q_0	q_1
* q_1	q_1	q_1

Ending with a.

$L = \{ a, ba, bba, aaa \ldots \}$





q_2 → deadstate

Q \ Σ	a	b
q_0	q_1	q_0
* q_1	q_1	q_0

Transition :-

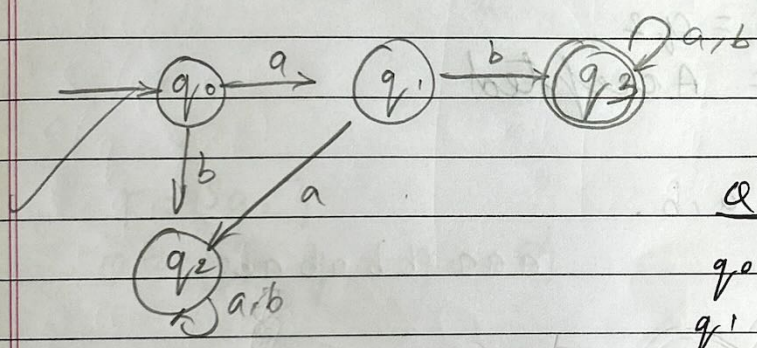
$\delta(q_0, abbaa)$
$\delta(q_1, bbaa)$
$\delta(q_0, aa)$
$\delta(q_1, a)$
$= q_1$
= Final state
= accepted

Construct a DFA which accepts set of all strings over input a, b which.

i) 'ab' → starts
ii) contains 'ab'
iii) ends with 'ab'

i) 'ab'

L = { ab, abaa, abab, abbb, . . . }



Q \ Σ	a	b
q_0	q_1	q_2
q_1	q_2	q_3
q_2	q_2	q_2
*q_3	q_3	q_3

Transition :-

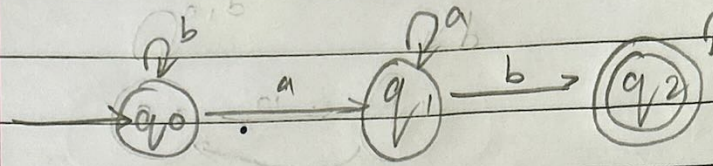
$\delta(q_0, aba)$
$= \delta(q_1, ba)$
$= \delta(q_3, a)$
$= q_3$

Final state accepted

$F = \{q_3\}$

$Q = \{q_0, q_1, q_2, q_3\}$

2)



	Σ	a	b
→	q_0	q_1	q_0
	q_1	q_1	q_2
*	q_2	q_2	q_2

$Q = \{q_0, q_1, q_2\}$

$F = \{q_2\}$

Transition :- $\delta(q_0, aab)$

$\delta(q_1, ab)$

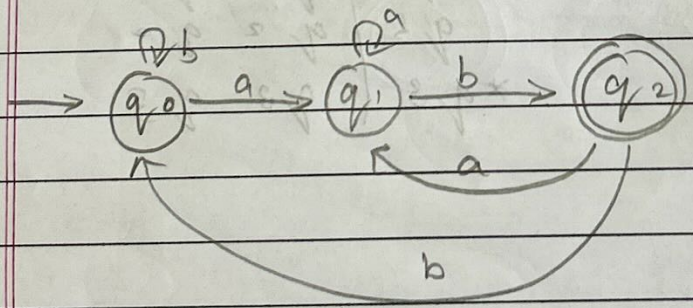
$\delta(q_1, b)$

$\delta = q_2$

= Accepted

3) ends with a, b.

aaa bbabab



$Q = \{q_0, q_1, q_2\}$

$F = \{q_2\}$

Transition :- $\delta(q_0, ab)$

$\delta(q_1, b)$

$\delta(q_2)$

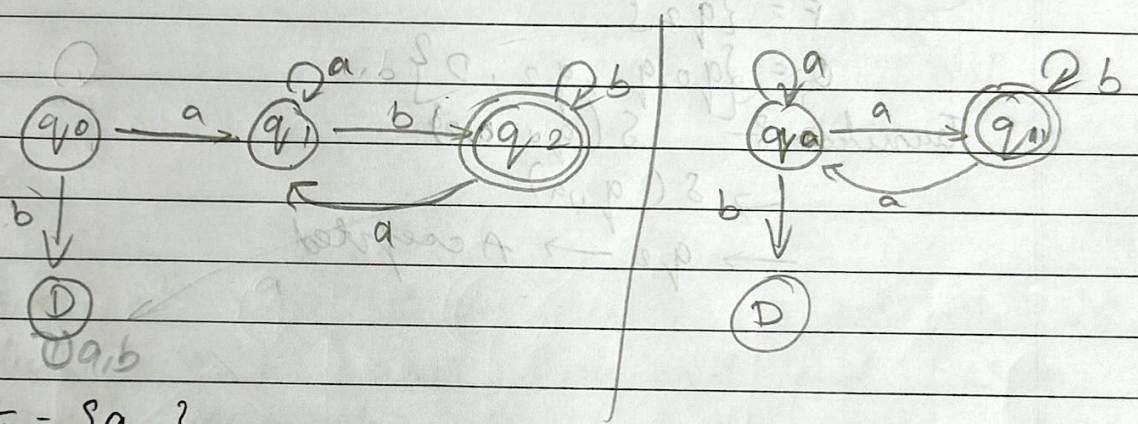
Accepted

PAGE No.
DATE / /

Construct a DFA which accepts set of all strings over input alphabet a,b.

i) starts with a and ends with b.
ii) starts with b and end with a.
iii) starts with a and ends with a.
iv) starts with b and ends with b.

i)



$F = \{q_2\}$

$Q = \{q_0, q_1, q_2, D\}$

Transition :-

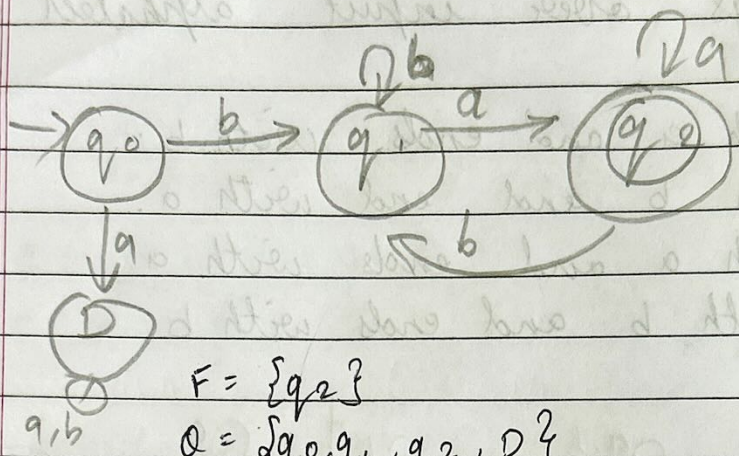
$\rightarrow \delta(q_0, ab)$

$\rightarrow \delta(q_1, b)$

$\rightarrow q_2$

Accepted.

ii) start b → end a



$F = \{q_2\}$

$Q = \{q_0, q_1, q_2, D\}$

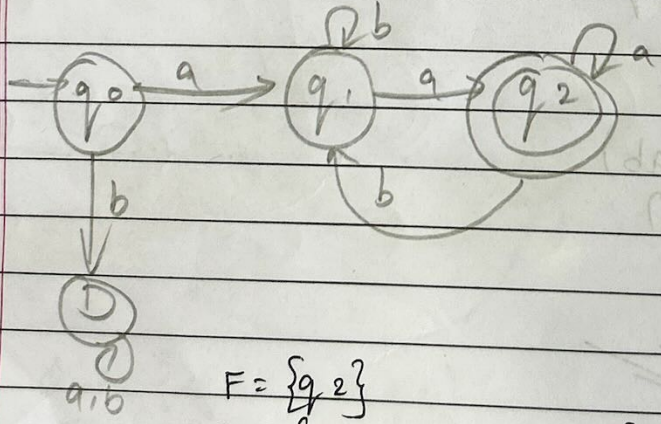
Transition :-

$\rightarrow \delta(q_0, baa)$

$\rightarrow \delta(q_1, a)$

$\rightarrow q_2 \rightarrow$ Accepted

iii) starts a ends a :-



$F = \{q_2\}$

$Q = \{q_0, q_1, q_2, D\}$

Transition :-

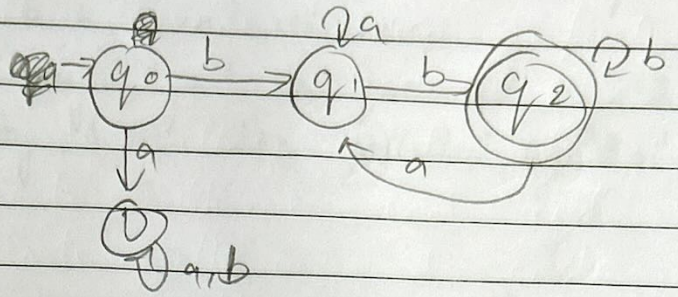
$\rightarrow \delta(q_0, aa)$

$\rightarrow \delta(q_1, a)$

$\rightarrow q_2$

$\rightarrow$ Accepted

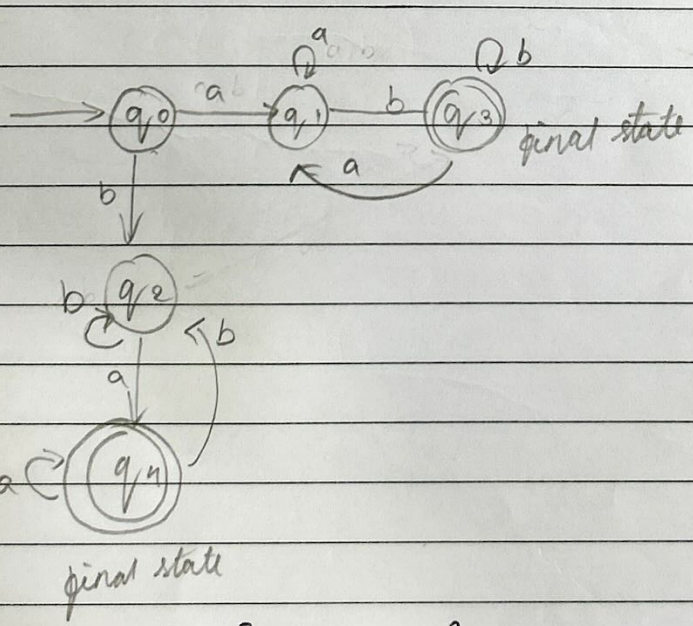
iv) starts b ends with b :-



Construct a DFA which accepts a set of all strings including a, b and starts and end with different symbols.

Ans

$L = \{ abb, ab, aab, abaab, baaba, \ldots \}$



abaaab

Q / Σ	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_4	q_2
* q_3	q_1	q_3
* q_4	q_4	q_2

$F = \{q_3, q_4\}$

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

starts and ends with same ~~##~~ symbol

PAGE No.
DATE / /

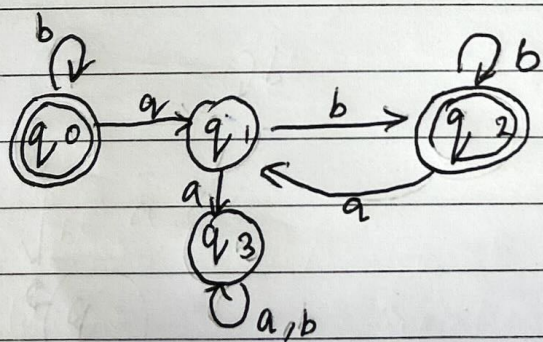
Ff Construct a DFA which accepts set of all strings of a, b in which.

i) Every 'a' is followed by a 'b'

ii) Every 'b' is followed by a 'a'

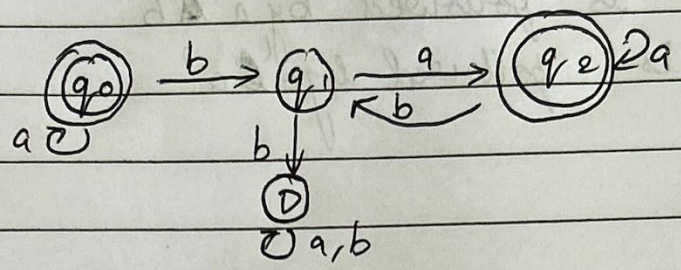
i)

$L = \{b, ab, abbbab, \ldots\}$

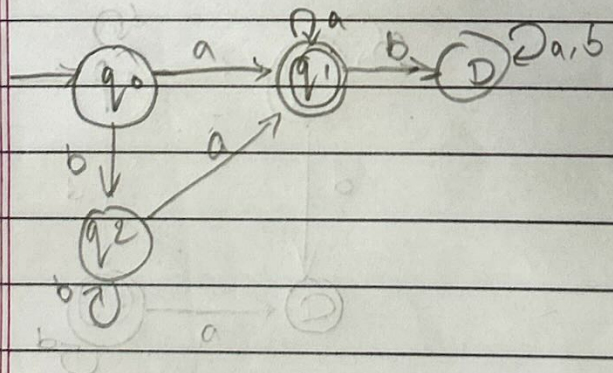


ii)

$L = \{ba, a, baaaba, \ldots\}$



#) construct a MDFA which accepts all strings over 'a,b' in which every 'a' should not be followed by 'b'



ii) 'b' should not be followed by 'a'

PAGE No.
DATE / /

In which every a should be followed by "bb"

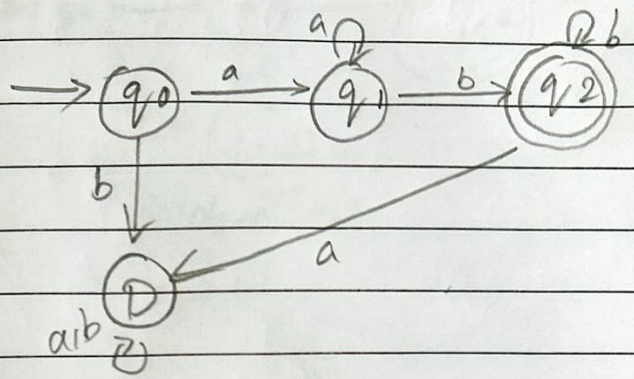
Every a shouldnot be followed by "bb."

PAGE No.
DATE / /

MDFA

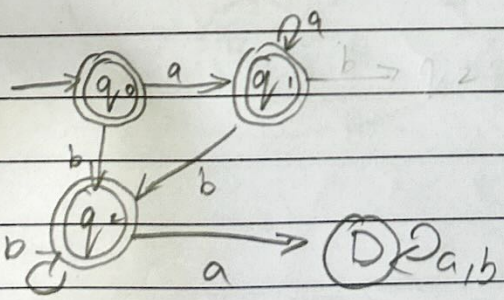
i) $L = \{ a^n b^m \mid n, m >= 1 \}$

$L = \{ ab, aab, abb \ldots \}$



ii) Q) $L = \{ a^n m^n \mid n, m >= 0 \}$

$L = \{ \epsilon, a, ab, aabb, abbb, \ldots \}$



iii) $L = \{ a^i b^j c^k \mid i, j, k \geq 1 \}$

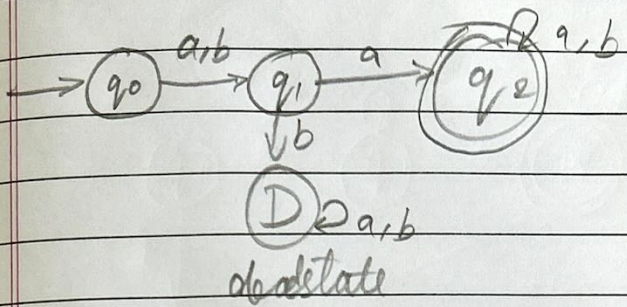
iv) $L = \{ a^l b^m c^n \mid l, m, n \geq 0 \}$

PAGE No.
DATE / /

construct minimum DFA which accepts set of all strings over input{a, b} such that second symbol from LHS is a.

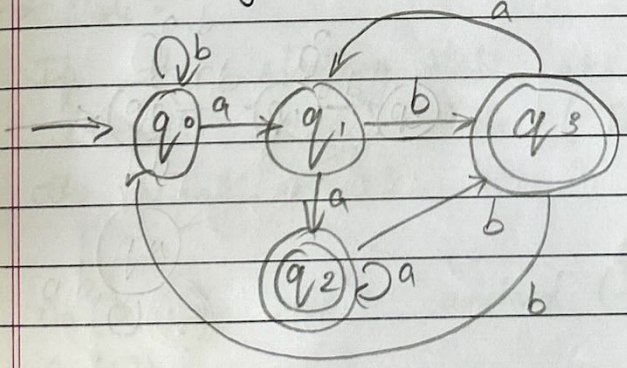
→ L = { a, aa, ba, baa, baab ... }

aabbab



second symbol from RHS is a.

abbbaabab



PAGE No.
DATE / /

3rd from LHS is a

Third from RHS is a.

PAGE No.
DATE / /

Construct a DFA which accepts all string input of {a,b} which accept strings of form $a^3 b_w a^2$

2/2 = ⊙

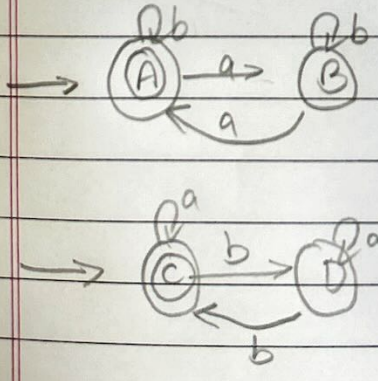
Cross Product Method of DFA :-

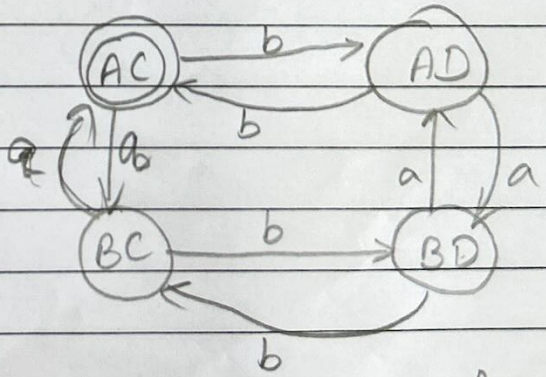
→ Note :-

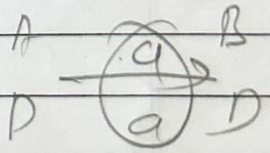
If the starting state are coming together then in the product state (They both together become the final state).

$n_a(w) \cong_0 \text{mod } 2$ andand (ff)

$n_b(w) \cong \text{mod } 2$



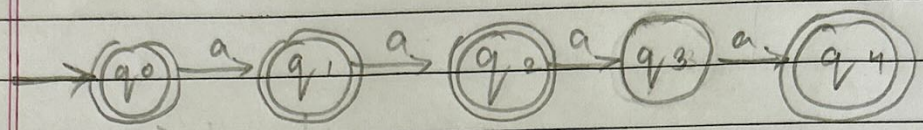




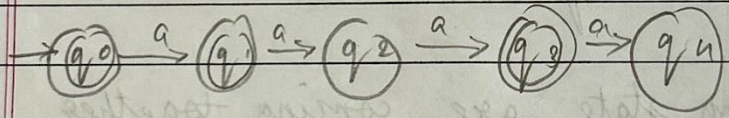
$\{a^n \mid n >= 0, n != 3\}$

$L = \{\epsilon, a, aa, aaaa\}$

Make for $n != 3$; then apply reversal rule and make initial state final and vica versa.



$\{a^n \mid n >= 0, n != 2, n != 4\}$



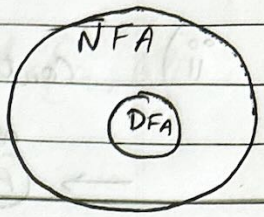
PAGE No.
DATE / /

NFA

$(Q, \Sigma, \delta, q_0, F)$

$\delta : Q \times \Sigma \longrightarrow 2^Q$

Transformation → DFA Q



You can solve DFA for all NFA questions.

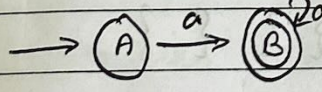
Construct NFA for following language.

i) starts with 'a'
ii) contains 'a'
iii) ends with 'a'

i)

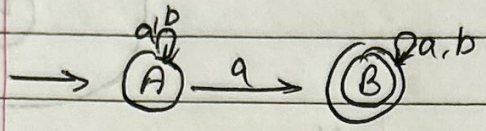
starts with a

$L = \{a, aa, abb, \ldots\}$



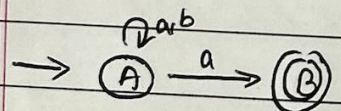
	a	b
A	{B}	{}
B	{B}	{B}

ii) Contains a.



	a	b
A	{A,B}	{A}
B	{B}	{B}

iii) Ends with 'a' :-



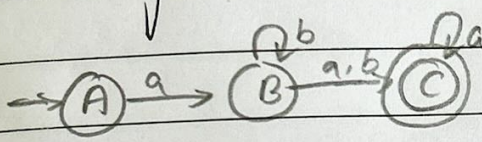
	a	b
A	{A,B}	{A}
B	{ }	{ }

PAGE No.
DATE / /

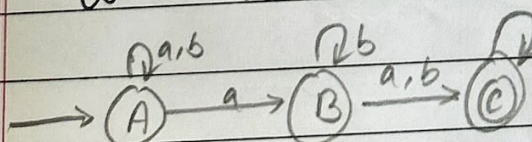
Design NFA over input alphabet {a, b} on following language.

i) string starts with ab
ii) Contains ab
iii) Ends with ab

i) string starts with ab

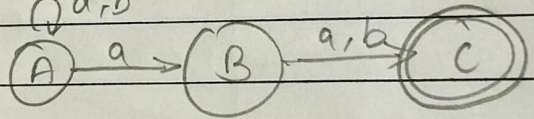


ii) contains ab



second from RHS is a.

$L = \{ aa, ab, aab, [redacted] baa, ---- \}$



PAGE No.
DATE / /

DFA (H.W)

(1) $n_a(w) \cong 0 \bmod 3$ → divide by 3 → 0
$n_b(w) \cong 0 \bmod 2$ → " " 2 → 0

(2) $n_a(w) \bmod 3 == 1$ → by 3 → rem = 1
$n_b(w) \bmod 3 == 2$ → by 3 → rem = 2

(3) $n_a(w) \bmod 3 > n_b(w) \bmod 3$

n_a rem > n_b rem

PAGE No.
DATE / /

(0,1) 0000

Construct a minimal DFA {0,1} when interpreted by binary number is divisible by 2.

Divisible by 3

PAGE No.
DATE / /

Divisible by 4.

Accepts set of all strings for input alphabet {0, 1, 2} when interpreted as ternary number when divisible by 4.

9) can have many end states.

PAGE No.
DATE / /

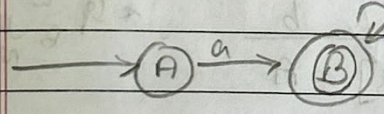
NFA to DFA :-

Rules :-

1) Starts with initial state
2) Find all reachable states from initial state.
3) If there is no state for any specific symbol, create dead state.
4) The states we get in NFA table, we should consider only that states in DFA table.
5) In NFA it goes to {A,B}, write it [AB]
6) Take union in NFA of A and B for new state [AB]
7) If no new state found end table
8) If it contains B and state in combs take combs as end state.

Construct NFA for given language and convert into DFA over input alphabet {a,b} where all strings starts with a.

NFA :-



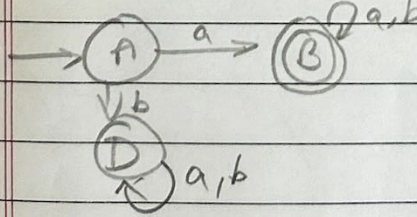
Q	ε	a	b
A		B	ϕ
*B		B	B

Make dead state

→ DFA

Q	ε	a	b
A		B	D
*B		B	B
D		D	D

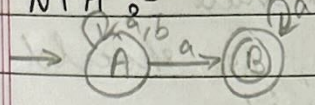
Take B and new state



construct NFA for input {a,b}
NFA to DFA. ⟶ Ending with a.
For given language

$L = \{a, aa, baa, aba, \ldots\}$

NFA :-



NFA

Q	Σ a	b
A	{A,B}	A
*B	∅	∅

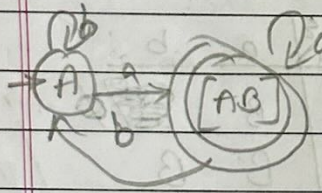
Take union (U)

Take new state instead of ∅

* If we don't get B in RHS table don't take it

* If we get B then create a new state

DFA :-



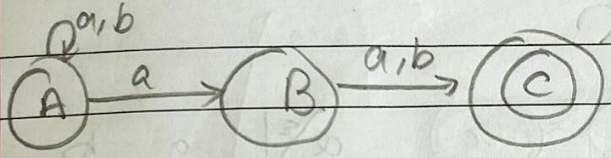
DFA

Q	Σ a	b
A	[AB]	A
*[AB]	[AB]	A

2nd From RHS is a.

$L = \{ ab, aab, \ldots \}$

NFA :-

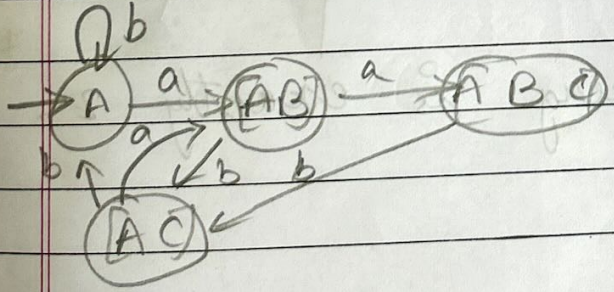


NFA

Q / Σ	a	b
A	{A,B}	A
B	C	C
C	∅	∅

union

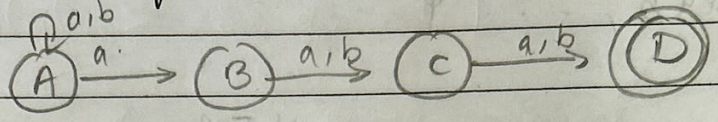
DFA :-



Q / Σ	a	b
A	[AB]	A
[AB]	[ABC]	[AC]
* [ABC]	[ABC]	[AC]
* [AC]	[AB]	A

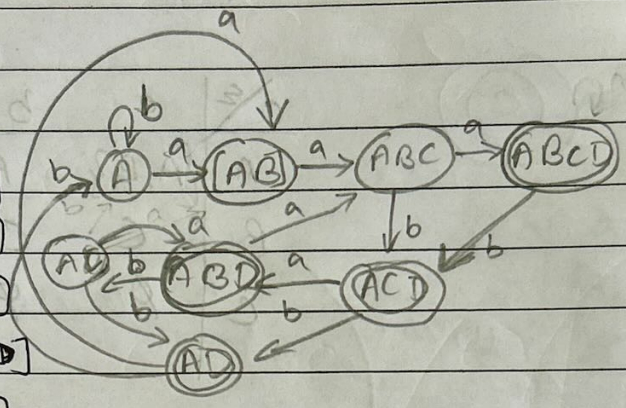
Design NFA → DFA

i) 3rd from RHS is a.



q \ Σ	a	b
A	[A,B]	A
B	C	C
C	D	D
* D	∅	∅

q \ Σ	a	b
A	[AB]	A
[AB]	[ABC]	[AC]
[ABC]	[ABCD]	[ACD]
[AC]	[ABD]	[AD]
*[ABCD]	[ABCD]	[ACD]
*[AD]	[ABD]	[AD]
*[ABD]	[ABC]	[AC]
[AD]	[AB]	[A]



q \ Σ	a	b
A	[A,B]	A
[A,B]	[ABC]	[AC]
[ABC]	[ABCD]	[ACD]
[AC]	[ABD]	[AD]
[ABD]	[ABCD]	[ACD]
[AD]	[ABD]	[AD]

Length of the string is exactly 2.

PAGE No.
DATE / /

length of the string is atmost 2.

atleast 2.

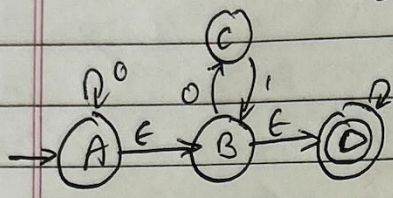
PAGE No.
DATE / /

ϵ NFA :-

ϵ NFA

$(Q, \Sigma, \delta, q_0, F)$

$\delta : Q \times \Sigma \cup \{\epsilon\} \longrightarrow 2^Q$



Q / Σ	ϵ	0	1
A	{A,B,D}	A	∅
B	{B,D}	C	∅
C	C	∅	B
D	D	D	D

0	ϵ*	0	ϵ*
A	A B	A	A → B → D
B	B	C	C
D	D	D	D

	ϵ*	1	ϵ*
A	A	∅	–
B	B	∅	–
D	D	D	D

0	ϵ*	0	ϵ*
→ B	B	C	C
	D	D	D

B	ϵ*	1	ϵ*
	B	∅	∅
	D	D	D

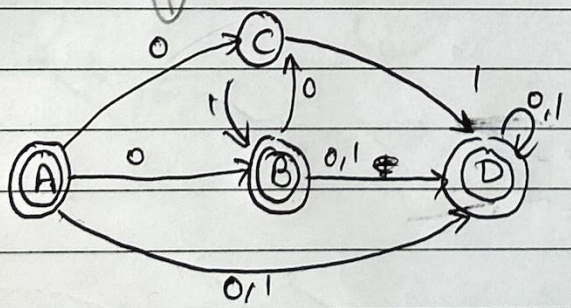
0	ϵ*	0	ϵ*
C	C	∅	∅

	ϵ*	1	ϵ*
C	C	B	B
			D

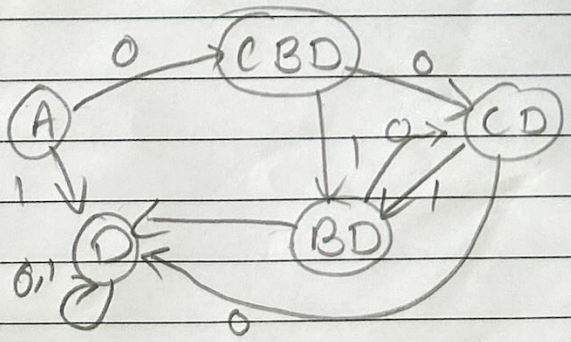
$0E^*0E^*$	$1E^*1E^*$
D D D D	D D D D

Σ / Q	0	1
A	{A, B, C, D}	{D}
→ B	{C, D}	{D}
C	{ }	{B, D}
D	{D}	{D}

Whatever state that can reach the final state from some state by E. Those all states are final states.



	0	1
*A	{C, B, D}	D
*B	{C, D}	D
C	{ }	{B, D}
*D	D	D

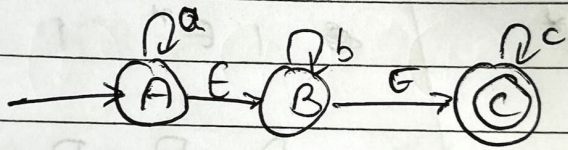


	0	1
*A	[CBD]	D
*CBD	[CD]	[BD]
*CD	D	[BD]
*BD	[CD]	[D]
*D	D	D

PAGE No.

DATE / /

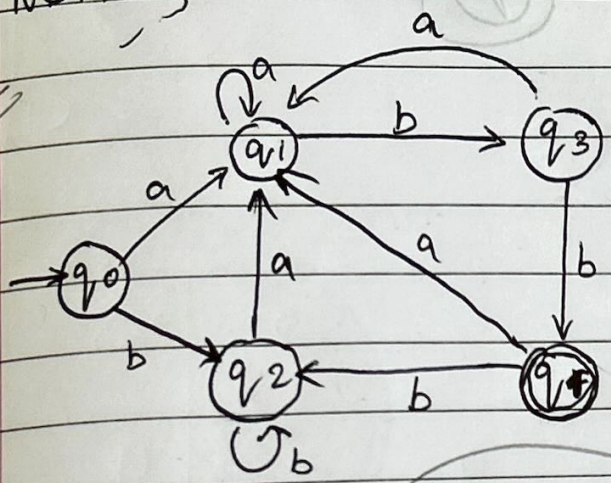
#



continued in 2 pages

PAGE No.
DATE / /

DFA → NDFA



q / Σ	a	b
q0	q1	q2
q1	q1	q3
q2	q1	q2
q3	q1	q4
* q4	q1	q2

equivalent

Check equivalent

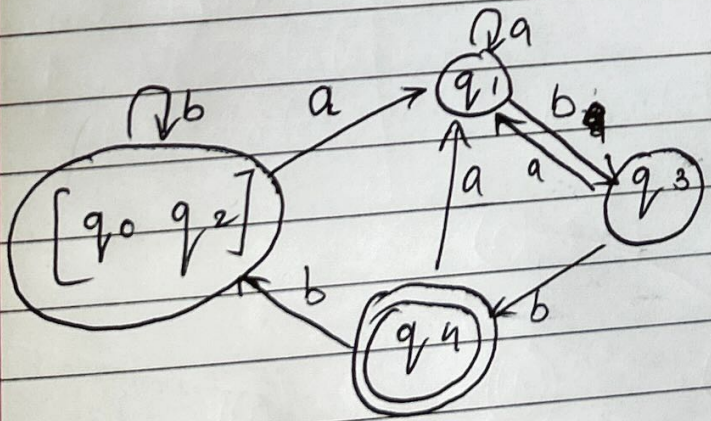
equivalence :-

$\Pi_0 \Rightarrow \{q_0, q_1, q_2, q_3\}\{q_4\}$ (G_1, G_2)

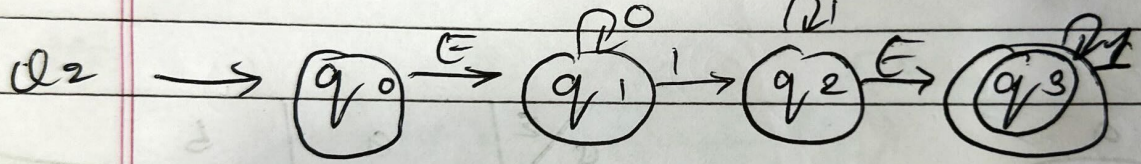
$\Pi_1 \longrightarrow \{q_0, q_1, q_2\}\{q_3\}\{q_4\}$ (G_1, G_2, G_3)

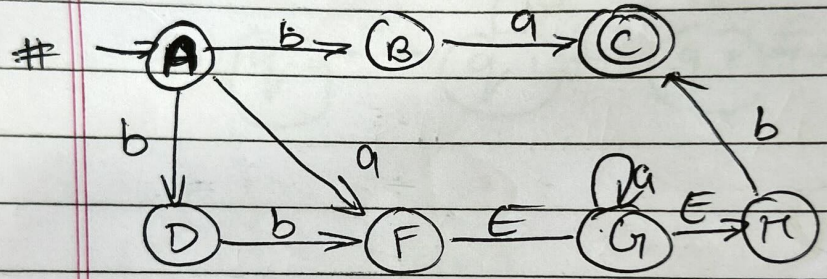
$\Pi_2 \longrightarrow \{q_0, q_2\}\{q_1\}\{q_3\}\{q_4\}$ (G_1, G_2, G_3, G_4)

(If in same group remain
If a, b of them in diff grp then split)

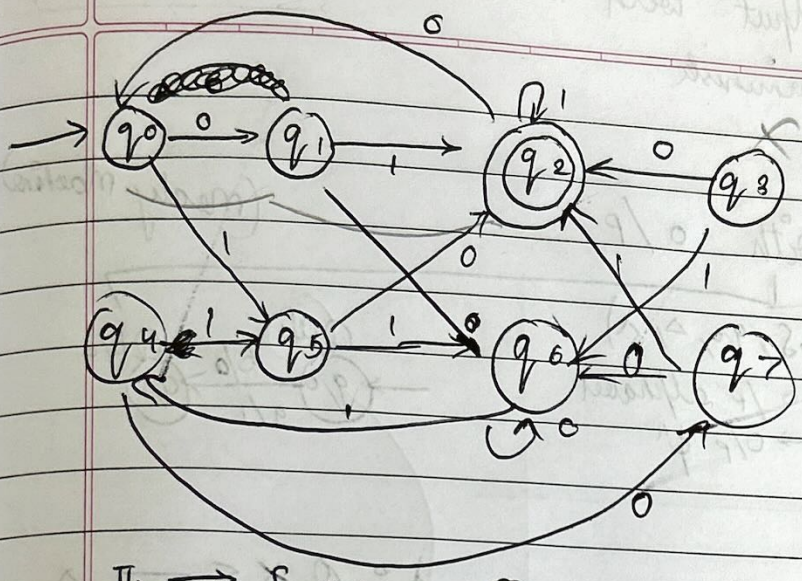


PAGE No.
DATE




#
A
b
B
a
C
b
b
a
a
D
b
F
E
G
E
H

* Delete all nodes where we cannot reach from initial state.

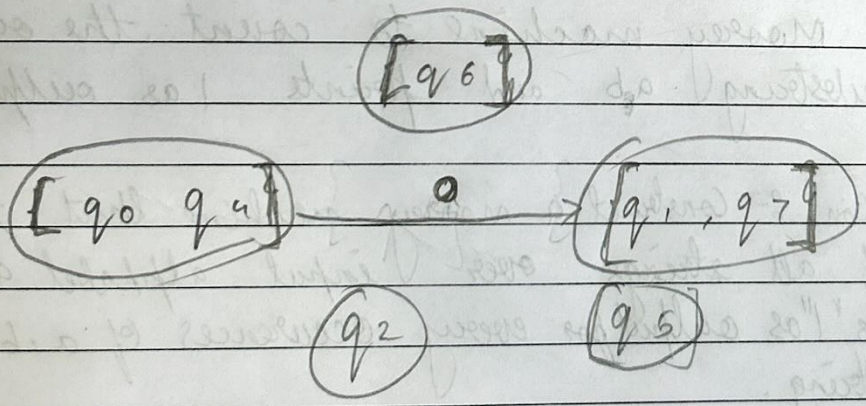


δ	0	1
q0	q1	q5
q1	q6	q2
*q2	q0	q2
×q3	q2	q6 delete
q4	q7	q5
q5	q2	q6
q6	q6	q4
q7	q6	q2

$\Pi_0 \rightarrow \{q_0\ q_1\ q_4\ q_5\ q_6\ q_7\}\ \{q_2\}$

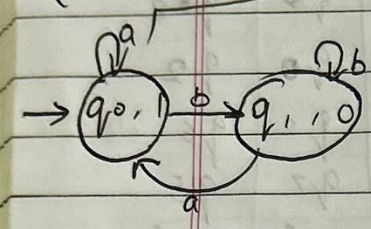
$\Pi_1 \rightarrow \{q_0, q_4, q_6\}\ \{q_1, q_7\}\ \{q_5\}$

$\Pi_2 \rightarrow \{q_0, q_4\}\ \{q_6\}\ \{q_1, q_7\}\ \{q_5\}\ \{q_2\}$

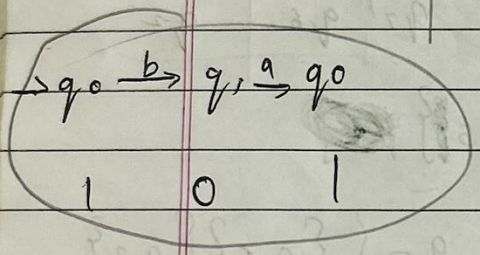


* Moorey Machine has output with q_i.
  → By default without any condition write output of 1st state
* Mealy has output both state (a and b)
* Both are deterministic.

(Moorey Machine)



$\lambda : Q \rightarrow \Delta$



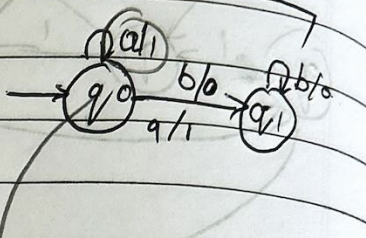
FA with o/p

$(Q, \Sigma, \delta, q_0, \Delta, \lambda)$

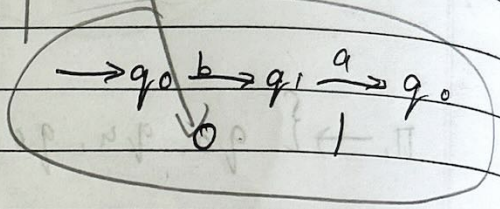
$\Delta \rightarrow$ o/p alphabet

$\lambda \rightarrow$ o/p ϕ

(Mealy Machine)



$\lambda : Q \times \Sigma \rightarrow \Delta$



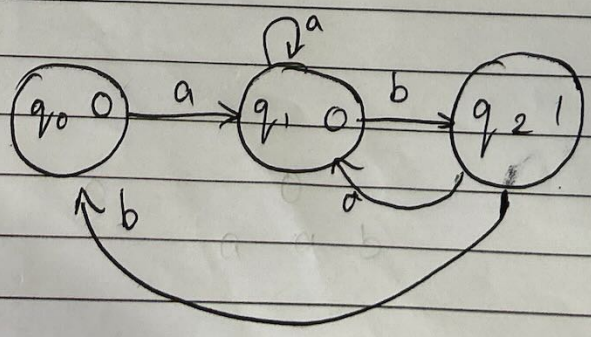
Design Moorey machine to count the occurences of substring ab and prints 1 as output.

Description :- Construct a moorey machine that takes set of all strings over input alphabet a, b and prints "1" as output for every occurences of a, b as a substring.

→ $\Sigma = \{a, b\}$

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

	a	b	Δ
q_0	q_1	q_2	0
q_1	q_1	q_2	0
q_2	q_1	q_0	1

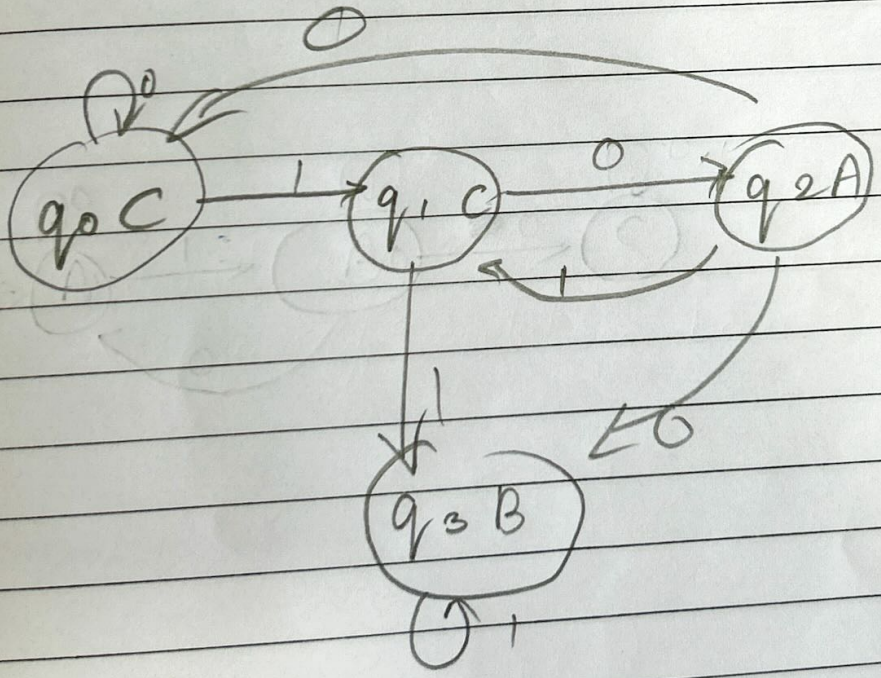


PAGE No.
DATE / /

Simulation :- a a b a b a b

Construct a Moorey machine for counting occurence of substring baa and baba ✓

Q Construct Moore machine that takes set of all string over 0, 1 and produces a as output if input ends with 10 or produces b as output if input ends with 11 otherwise produces c.



PAGE No.
DATE / /

Construct Moorey machine that takes binary number as input and produces residue module 3 as output.

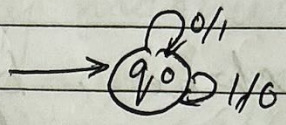
PAGE No.
DATE / /

Mossy machine : base 4 number (0, 1, 2, 3) as input and produces residue modulo 5 as output.

Mealy Machine :-

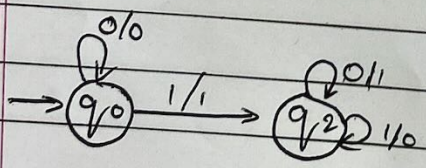
1 → 0

0 → 1

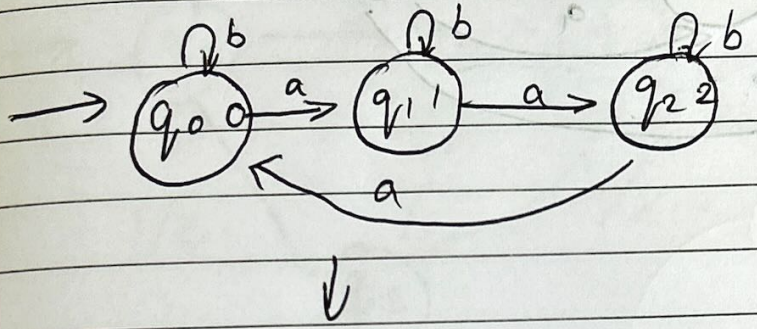


	ε	0	1
→ q0		(q0, 1)	(q, 0)

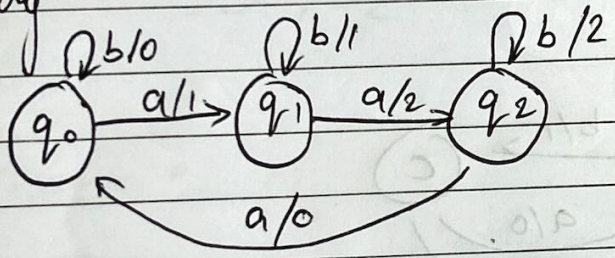
Construct mealy machine for 2's complement Binary No as an input and produces 2's complement as output assume the string is read from LSB to MSB and carry is discarded.



Moore → Mealy M/C



Mealy

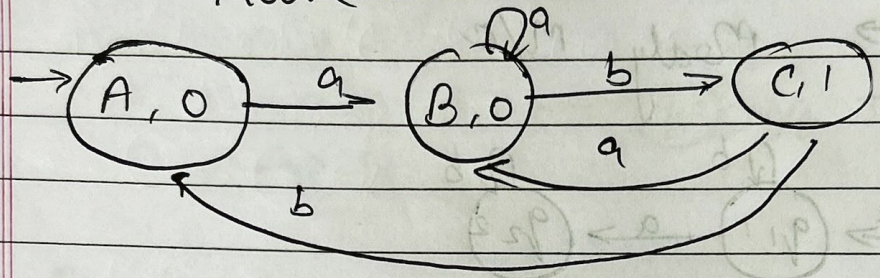


1) Check where the variable or elements (a, b, 1, 0) are going and write with the upcoming output of ~~state~~ state.

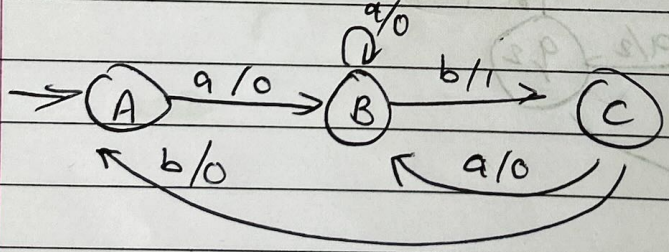
PAGE No.
DATE / /

Q

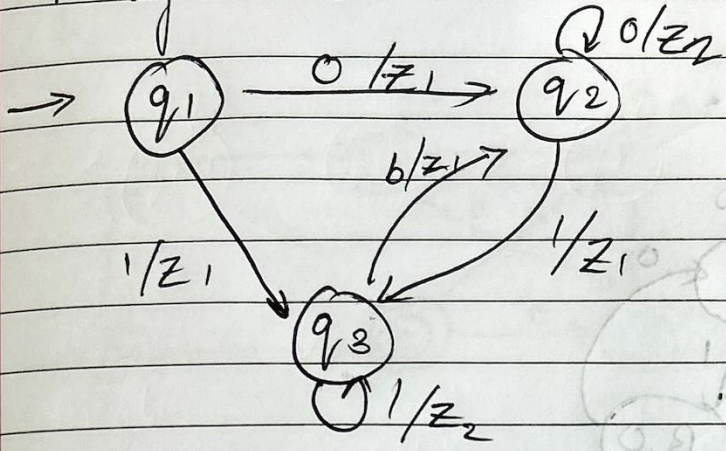
Moore



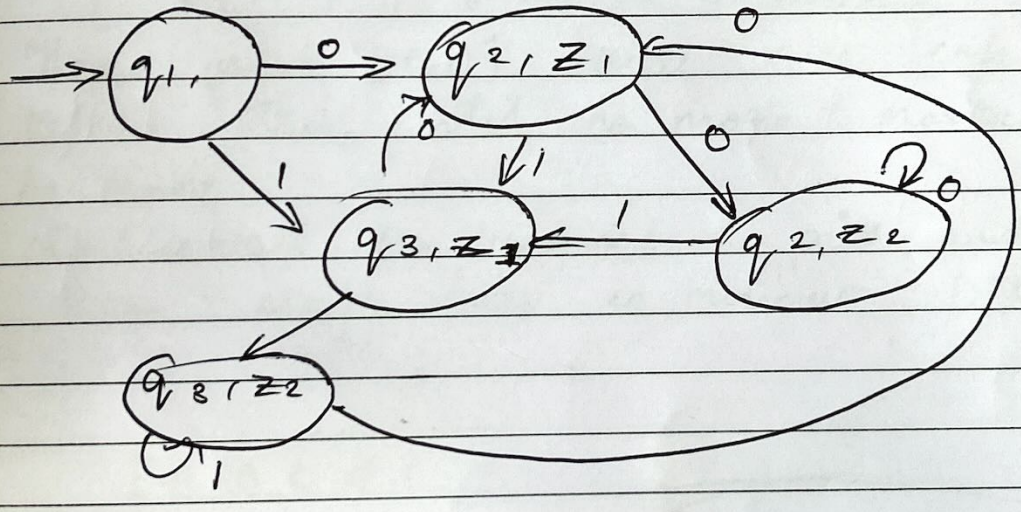
Mealy



Mealy to Moore



Moore :-



step 1:-

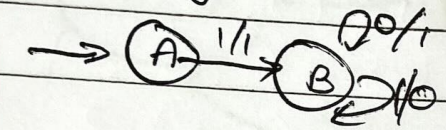
→ check start state For eg (q1) and por 0 which output it has whis here is z1 so the (q1) will go to (q2, z1) for input 0.

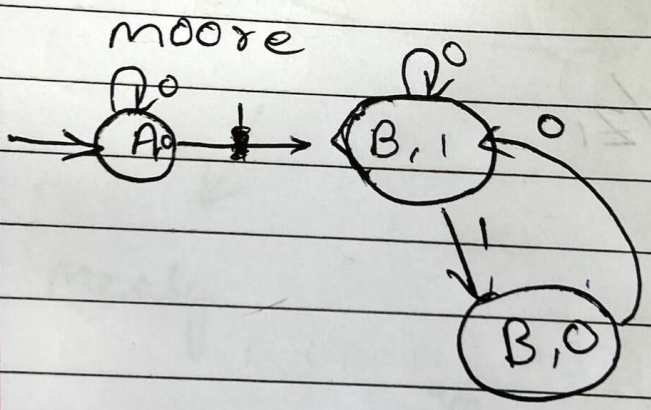
→ For 0 input on q2 it will go to (q2, z2). But this state doesnot exist. So Make a new state.

→ Complete the whole diagram step by step to reduce confusion

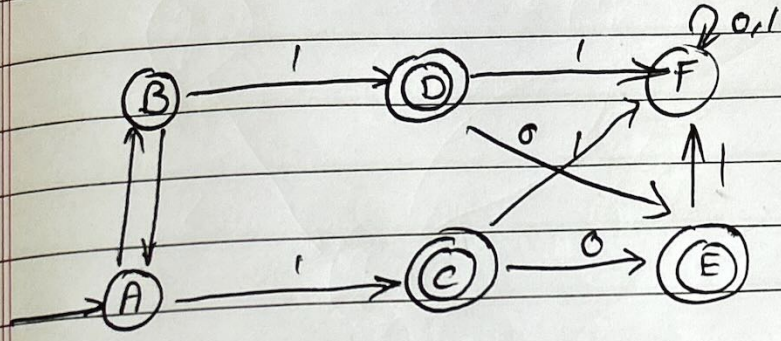
PAGE No.
DATE / /

Mealy M/C to Moore :-





Myhill Nerode



i) Draw table for all pairs of states (P, Q)

ii) Mark all pairs where P ∈ F and Q ∉ F

iii) If there are any unmarked pairs (P, Q) such that $[\delta(P, x), \delta(Q, x)]$ is marked. Then mark [P, Q] where x is input symbol

iv) Repeat this until no more markings can be made.

v) Combine all the unmarked pairs and make them single states in minimum DFA.

	A	B	C	D	E	F
A						
B						
C	×	×				
D	×	×				
E	×	×				
F			×	×		

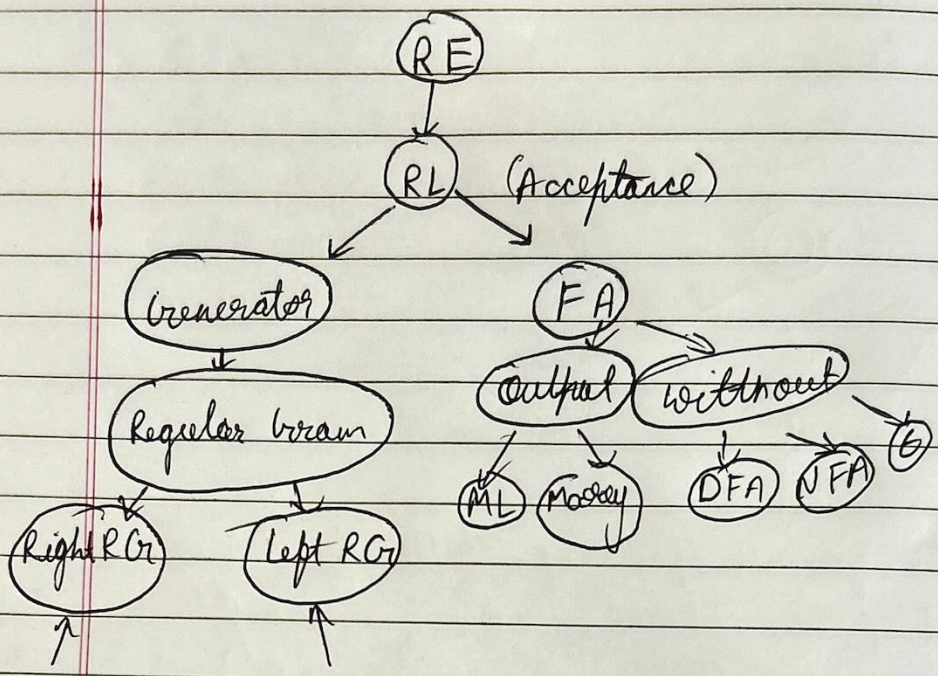
cross when one of 2 states is an end state

$\delta(A, 0) \Rightarrow B$

$\delta(B, 0) \Rightarrow A$

$\delta(A, 1) \Rightarrow C$

$\delta(B, 1) \Rightarrow D$



Provide regular expression for given language
Set of all strings whose length is equal to 2 over input alphabet a, b

$L = \{ aa, ab, ba, bb \}$

$\longrightarrow aa + ab + ba + bb$

$\longrightarrow a(a+b) + b(a+b)$

$\longrightarrow (a+b) \cdot (a+b)$

Exactly length equals to three :-

$L = \{ aaa, bbb, abb, baa, aba, bab \}$

$\longrightarrow aaa + bbb + abb + baa + aba + bab$

$\longrightarrow a(aa + bb + ba) + b(bb + aa + ab)$

$\longrightarrow (a+b)(aa + bb + ba + ab)$

$\longrightarrow (a+b)(a+b)(a+b)$

length equals four

$L = \{ aaaa, bbbb, abbb, baaa, abba, baab, bbaa$
$aabb, abab, baba, bbab, aaba, aaab, bbba, \}$

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

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

Almost 2 :-

$L = \{ \epsilon, a, b, aa, ab, bb, ab \}$

$\epsilon + \epsilon + a + b + aa + ab + bb + ab$

$\rightarrow (\epsilon + a + b)(\epsilon + a + b)$

$\rightarrow \epsilon \times \epsilon + \epsilon \times a + \epsilon \times b + a \times \epsilon + a \times a + b \times a + b \times \epsilon + b \times a + b \times b$

$\rightarrow \epsilon + a + b + aa + ab + bb + ba$

Write a regular expression where length of string is even.

$L = (aa, bb, \epsilon, ab, ba, \ldots$

Length of string is odd :-

* We can get multiple answer to the questions.

PAGE No.
DATE / /

Set of all string divisible by 3.

Set of all string whose length is congruent to $\cong 2 \bmod 3$.

→ R.E for set of strings, where no. of 'a' is 2.

$\Sigma = \{a, b\}$

$L = \{aa, baab, aab, baa, babab, \ldots\}$

$b^* ab^* ab^*$

→ R.E no. of 'a' be atleast 2.

$\Sigma = \{a, b\}$

$L = \{aa, aaa, aab, baa, \ldots\}$

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

→ R.E no of 'a' atmost 2.

$\Sigma = \{a, b\}$

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

→ Number of a's even

$\Sigma = \{a, b\}$

- $L = \{\epsilon, b, aa, aab, aabb, \ldots\}$

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

Or

$(b^* a b^* a b^*)^* \cdot b^*$

PAGE No.

DATE / /

Starting with a, end with a, contains a.

PAGE No.:
DATE / /

→ Starting and ending with different symbol. and same symbol.

PAGE No.
DATE / /

→ 2 as should not be together.

Set of string - where number of 2 as and number of 2 b's together

Properties :-

* $\phi + R = R + \phi = R$

$\phi \cdot R = R \cdot \phi = \phi$

$\epsilon R = R\epsilon = R$

$\epsilon^* = \epsilon$

$\phi^* = \epsilon$

$\epsilon + RR^* = R^*R + \epsilon = R^*$

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

$= (a^* b^*)^*$

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

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

$= a^* (ba^*)^*$

$= b^* (ab^*)^*$

RE to Finite Automata :-

ϕ → ○

a → ○ —a→ ◎

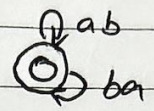
a+b → ○ —a,b→ ◎

ab → ○ —a→ ○ —b→ ◎

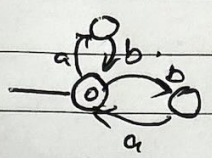
ab^* → ○ —a→ ◎ ↻ b^*

$(ab)^*$ → ◎ ↻ a,b

PAGE No.
DATE / /

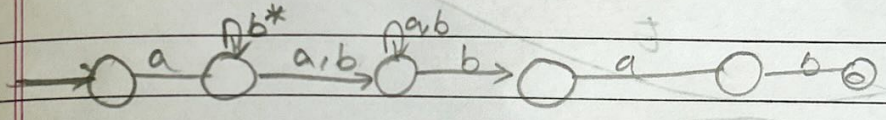
$(ab + ba)^*$ → 

or

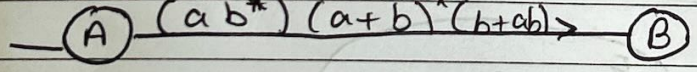


Design transition diagram for regular expression

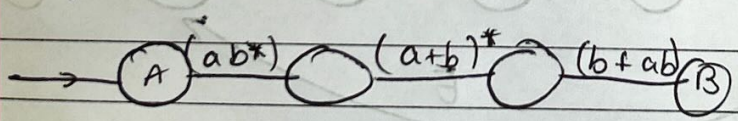
$R = (ab^*)(a+b)^*(b+ab)$



step 1 :-



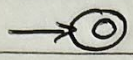
step 2 :-



PAGE No.
DATE / /

RE → FA

ε

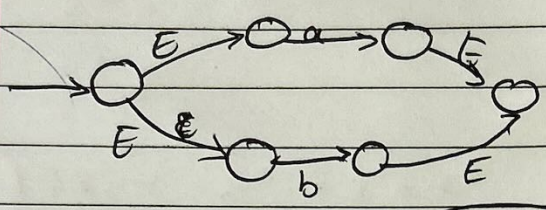


a

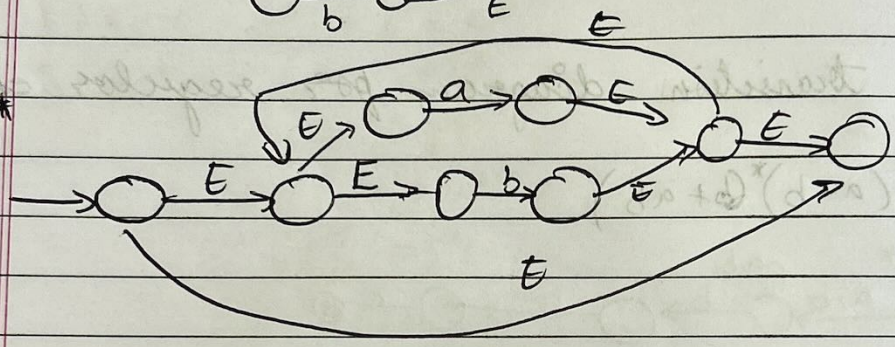
a+b

a+b (a|b)

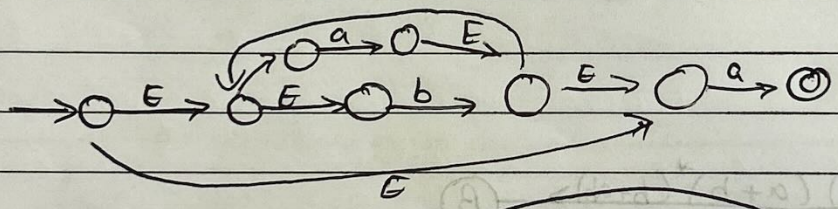
a or b



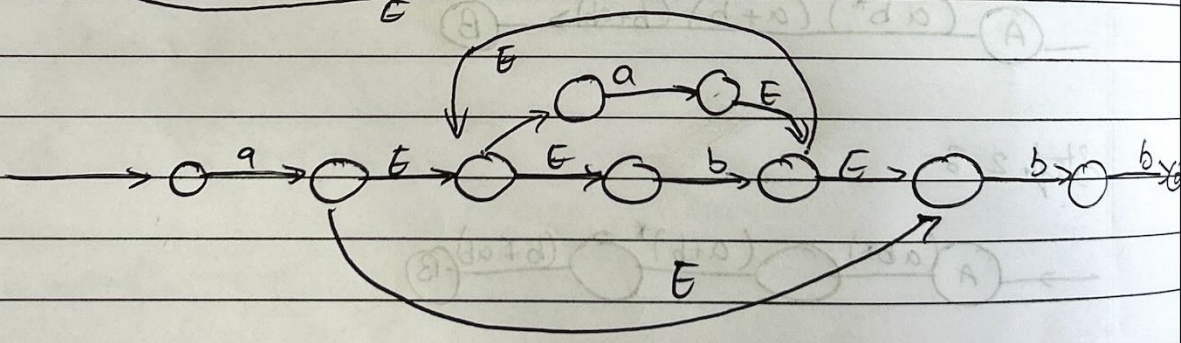
$(a+b)^*$



$(a+b)^*a$



$a(a+b)^*bb$



$10 + (0+11)0^*1$

PAGE No.
DATE / /

Q2 $(0+1)^*(00+11)(0+1)^*$

Q3 $(00+1)^*\ (10)^*$

PAGE No.
DATE / /

Identities for regular expression :-

$I_1 = \phi + R = R$

$I_2 = \phi R = R\phi = \phi$

$I_3 = \Lambda R = R\Lambda = R$

$I_4 = \Lambda^* = \Lambda = \phi^* = \Lambda$ → Epsilon or ϵ

$I_5 = R + R = R$

$I_6 = R^* R^* = R^*$

$I_7 = RR^* = R^*R$

$I_8 = (R^*)^* = R^*$

$I_9 = \Lambda + RR^* = R^*$

$I_{10} = (P\phi)^* P = P(\phi P)^*$

$I_{11} = (P + Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$

$I_{12} = (P + \phi)R = PR + \phi R$

$= R(P + \phi)$

$= RP + R\phi)$

Theorem :-

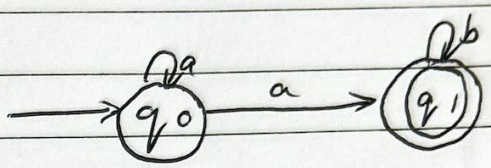
P and Q are Regular expression
R is a unique solution.

$R = Q + RP$

$R = QP^*$

Consider the transition diagram given below

→ Construct the regular expression corresponding to this transition diagram.



$q_0 = \epsilon + q_0 a$

$q_1 = q_0 a + q_1 b$

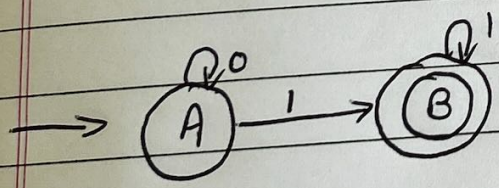
$q_1 = (\epsilon + q_0 a) a + q_1 b$

$q_1 = a^* a a + q_1 b$

$= a^* a b^*$ or $a^+ b^*$

$x^+ \rightarrow$ raise to + is one or many

Q2



$A = \epsilon + A0$ — ①

$B = A1 + B1$ — ②

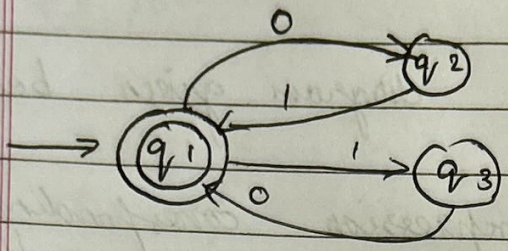
Substituting ① in ②

$B = (\epsilon + A0)1 + B1$

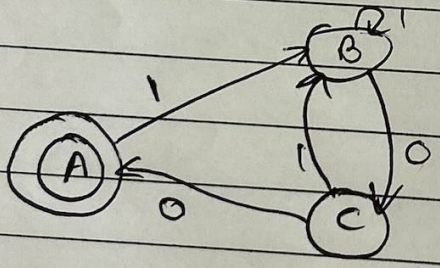
$= 0^* 1 1^*$

PAGE No.
DATE / /

Q3

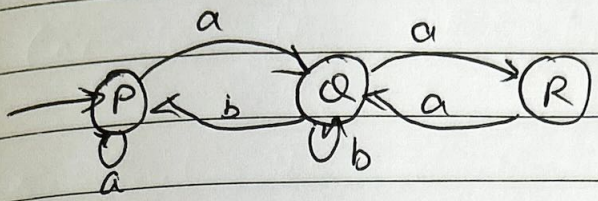


Q4



PAGE No.
DATE / /

Q5



Q6