THEORY OF COMPUTATION

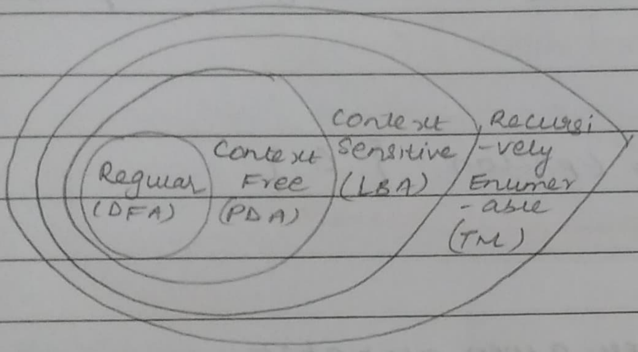
classmate
Date
Page

Venkata Rao M

30-05-2021

TOPICS -

1. Finite Automata and Regular Languages 50%
2. Pushdown Automata and Context Free Languages 30%
3. Turing machine and Recursive Enumerable Languages. } 20%
4. Undecidability.



Chomsky Hierarchy

\# Definition - It is the mathematical study of computing machine and its capability.

or

It is the study of Automata and Formal Languages.

\# Decidable problem - Problems that computers can solve

or

Problems for which an algorithm exists

\# Undecidable Problem - Computers can't solve.

or

Problems for which no algorithm exists

Alphabet -
Finite non-empty set of symbols are known as alphabet

$\Sigma = \{a, b, c\}$ $\Sigma = \{0, 1, 2, 4\}$

$\Sigma = \{$ अ, आ $\}$

String -
Finite sequence of symbols over the given alphabet is known as a string.

* Zero length string = Epsilon $\{\epsilon\}$

Language -
Any set of strings over given alphabet.

Finite Lang. $L_1 = \{ab, ba\}$ valid

Empty Lang $L_2 = \{ \}$ valid → 0 String

Infinite Lang $L_3 = \{a, b, aa \cdots \}$ valid

Finite Lang $L_4 = \{\epsilon\}$ → 1 String

Finite Lang. $L_5 = \{ab\}$

Complete Lang $L_5 = \{a, b, \epsilon, aa, ab \cdots\}$ All possible strings over the given alphabet

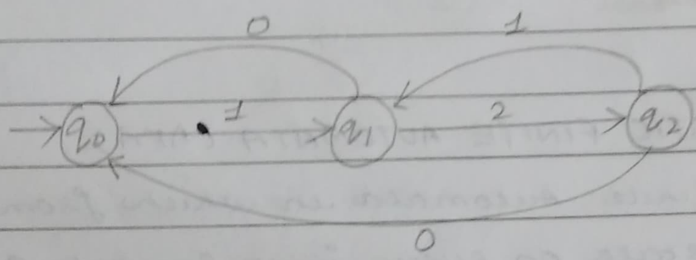
$\Downarrow$ Σ^*

★★ FINITE AUTOMATA

It is a mathematical model which contains finite number of states and transitions defined between them.

Example -

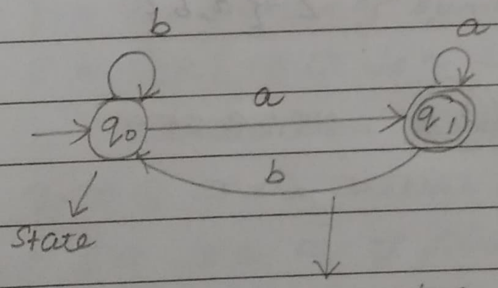
FAN FA



Finite Automata for a FAN

Example -

$L = \{ a, aa, ba, aba, \ldots \}$ = Infinite Language



Set of all strings ending with 'a'.

31-05-2021

Types of Finite Automata

Language recogniser

compilers

i/p → Without Output → YES / NO

1) DFA
2) NFA
3) ε-NFA

i/p → With Output → Output (Symbols)

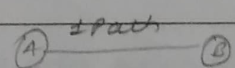
4) Mealey machine
5) Moore machine

1) DETERMINISTIC FINITE AUTOMATA (DFA)

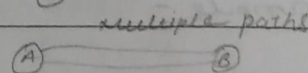
INVALID DEF.

It is a finite Automata in which from every state on every state on every input symbol exactly one transition is possible / should exist.

Deterministic = exact

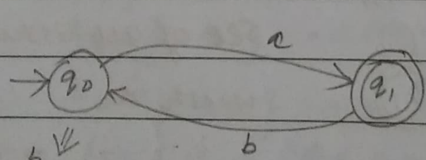


Non-deterministic = multiple



Example -

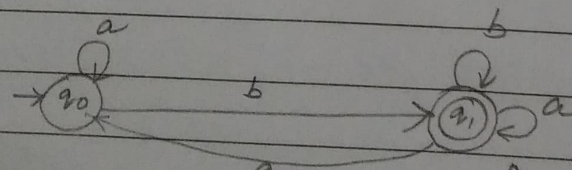
$\Sigma = \{a, b\}$



Not a DFA

कोई Transition नहीं है।

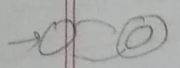
Example -



Not a DFA

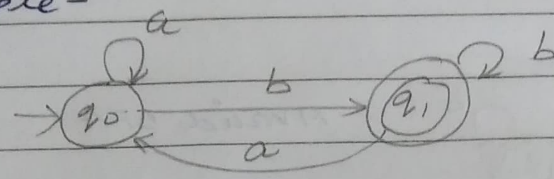
multiple edges on same symbol

DFA = Language = Expression = Grammar



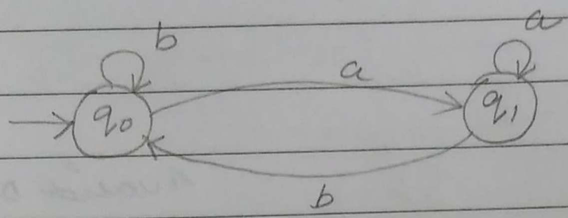
$\{ab, ab\}$ $(a+b)^*$ $S \to aS/bS/\epsilon$

Example -



It is a DFA

Example -



No final state still ~~not~~ a DFA

Formal Definition of DFA -

VALID DEF.

A deterministic finite accepter or DFA is defined by the quintuple

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

where

Q = Finite set of internal states

Σ = Finite set of symbols called the input alphabet

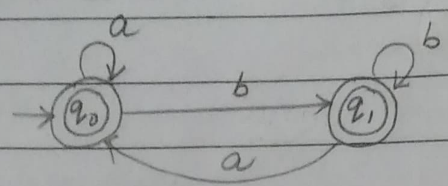
$\delta = Q \times \Sigma \to Q$ is a total function called the transition function.

$q_0 \in Q$ is the initial state [only one]

$F \subseteq Q$ is a set of final states. [≥ 0 allowed]

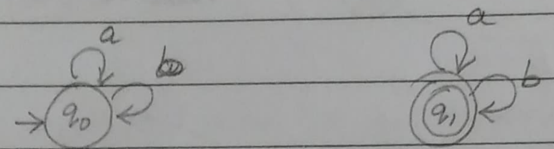
Check if the automata follows formal definition : $[Q, \Sigma, q_0, F, \delta]$

Example -



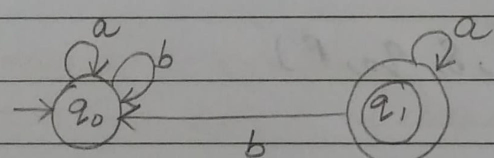
A valid DFA

Example -



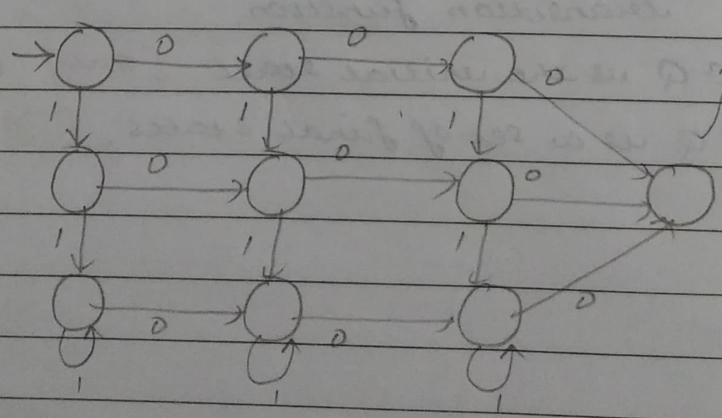
A valid DFA

Example -



A valid DFA

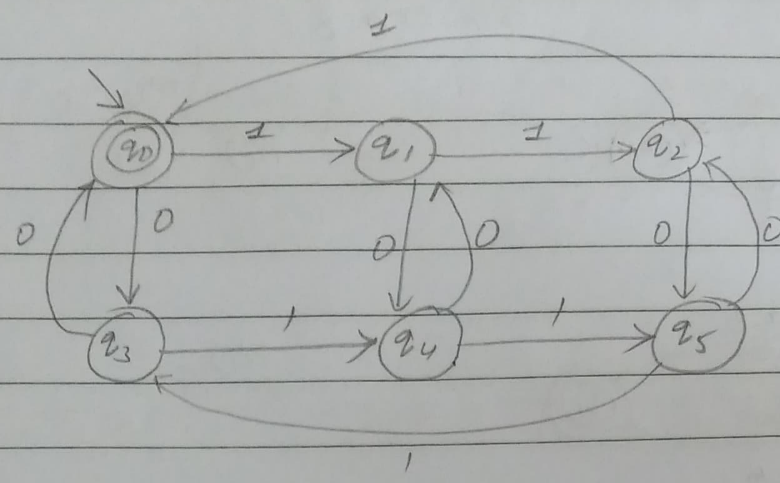
Example -



No 0,1 transition

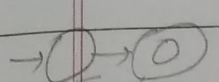
Not a valid DFA

Example -



A valid DFA

DFA = Language = Expression = Grammar

 = {ab, ba} = $(a+b)^*$ = $S \to aS/bS/\epsilon$

TYPES OF QUESTIONS

DFA ⟺ Language

1) DFA ⇒ Language

2) Language ⇒ DFA

★ DFA is a language recogniser.

Languages

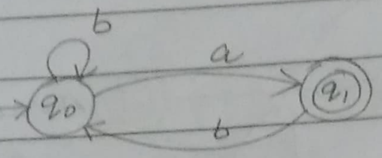
Finite Automata Grammar Expression

01-06-2021

Examples of DFA :-

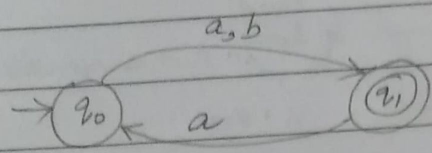
क्या DFA है ?

1.



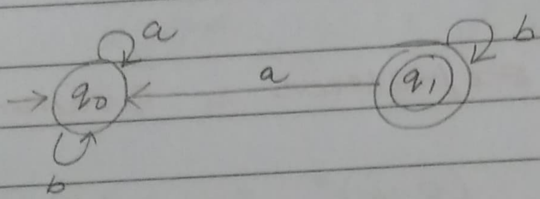
NO

2.



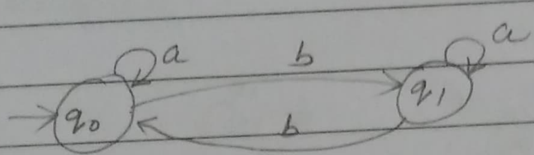
NO

3.



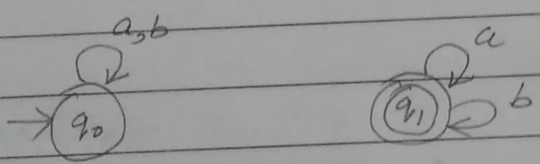
Yes

4.



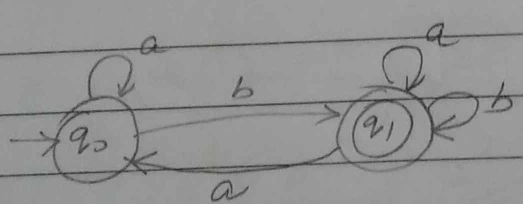
YES

5.



YES

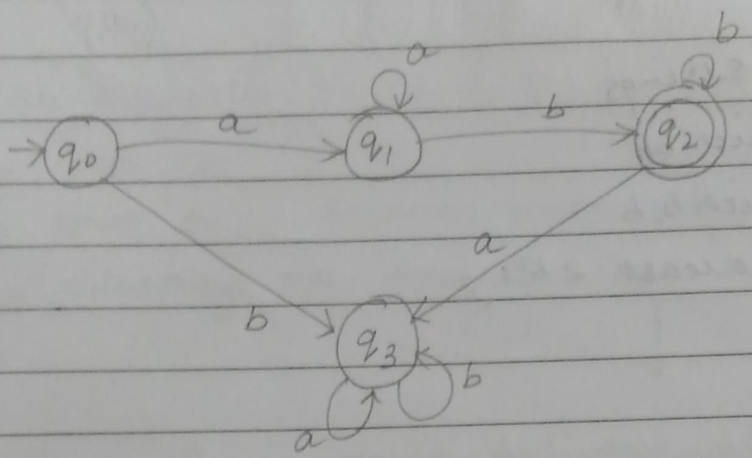
6.



NO

Q. Construct DFA for the following language.

1) $L = \{a^n b^m / n \geq 1, m \geq 1\}$ $\{ab, aab, abb, aabb\}$

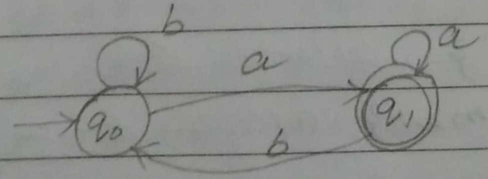


Acceptance method of DFA -

By reading the complete string from left to right end of the string if DFA halts in final state then string is accepted.

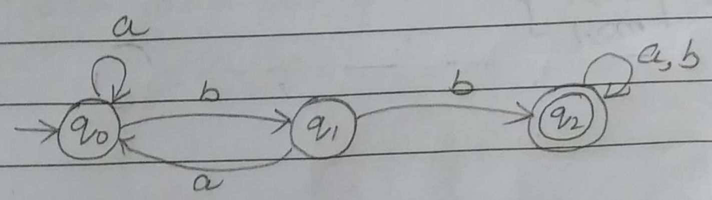
Set of all strings accepted by DFA is known as Language of that DFA.

Example-



Language - Set of all Strings ending with a.

Q. Identify the language accepted by the given DFA



set of all strings

(a) Starting with bb

(b) Ending with b,b

(c) Contains atleast 2 b's

✓(d) None

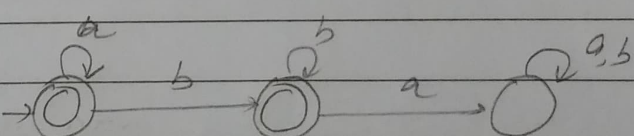
{bb, bba, bab, baba ....}

Not accepted

Hence cannot be the language

Try to disprove

Q. Identify the language accepted by the following DFA.



$= \{ \epsilon, a, b \ldots \}$

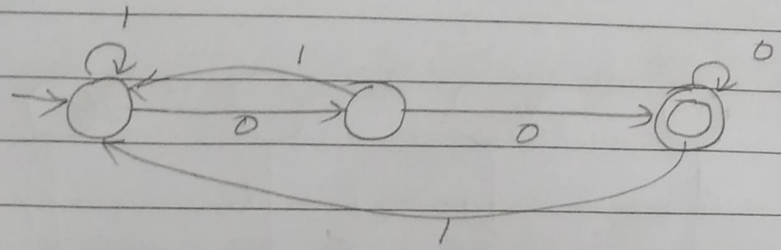
(a) $L = \{ a^n b^m \mid n,m \geq 1 \}$ accepts ϵ

(b) $L = \{ a^n b^m \mid n \geq 0, m \geq 1 \}$ accepts ϵ, a

(c) $L = \{ a^n b^n \mid n \geq 0 \}$ accepts abb

✓(d) None

Q. Identify the language accepted by the following



Set of all strings

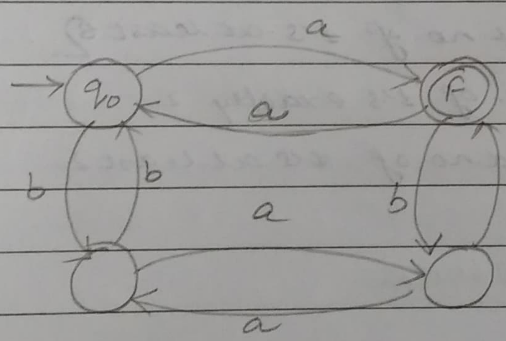
a) Starting with 00

b) ending with 0 Doesnot acept 1010

c) having substring 00 Doesnot accept 1001

✓ d) None.

Q. Identify the language accepted by the given Automata



= { a

Set of all strings having

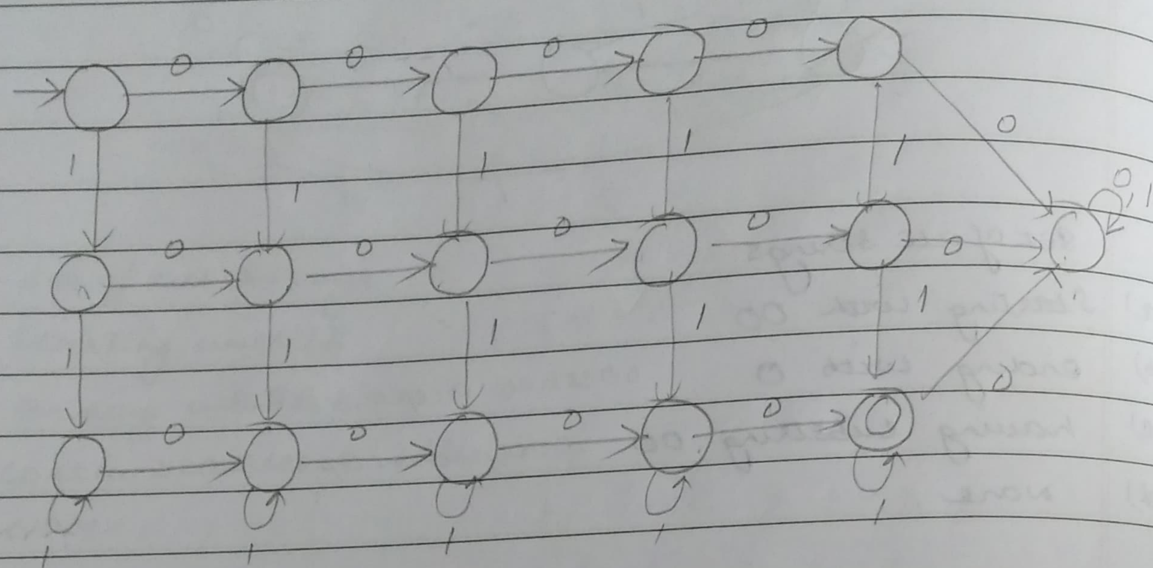
a) even no. of a's and even no. of b's accepts a | ε

b) even no. of a's and odd no.'s of b's accepts abb | b

✓ c) odd no. of a's and even no.'s of b's | a

d) odd no. of a's and odd no.'s of b's accepts aabba | ab, ba

Q. Identify language accepted by given Automata.



Set of all strings having

a) Length of the String at least 6 — does not accept 0000000

X b) No. of 0's at least 4 and no. of 1's at least 2 — 1010000 not accept

c) No. of 0's exactly 4 & no. of 1's exactly 2 — 100010111 accept

✓✓ d) No. of 0's exactly 4 and no. of 1's at least 2

(6-1) X 3

⇒ 15 + 1 ⇒ 16 states

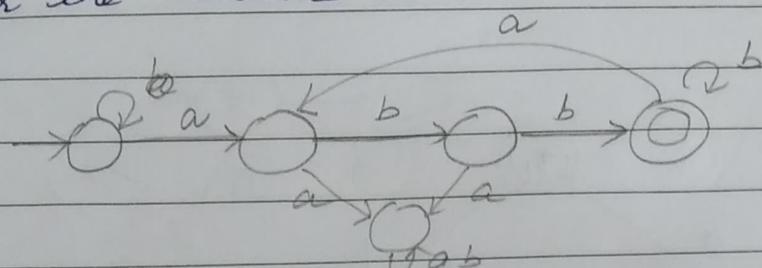
02-06-21

Note -

1) In any DFA if there is no final state, that DFA accepts empty language.

2) In any DFA if all states are final that DFA accepts complete language.

Q. Consider the machine M



The language recognized by M is

(a) $\{ w \in \{a,b\}^* \mid$ Every a in w is followed by exactly two b's $\}$ abbb {accepts} ×

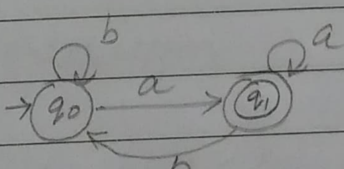
✓(b) $\{ w \in \{a,b\}^*$ Every a in w is followed by at least two b's $\}$

(c) $\{ w \in \{a,b\}^*$ w contains the substring 'abb' abba is not accepted

(d) $\{ w \in \{a,b\}^*$ w does not contain 'aa' as a substring $\}$
abba is not accepted

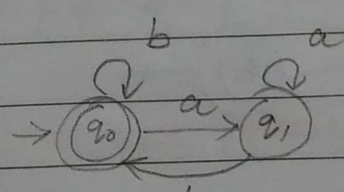
Complement of DFA

By interchanging Final and non-final states we can complement DFA.



Set of all strings ending with a

After complement -



Set of all string not ending with a.

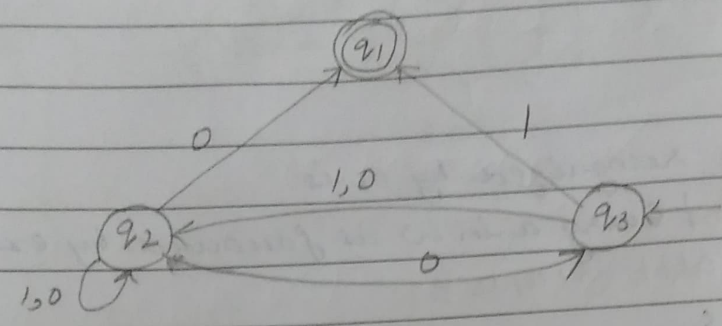
Ending with 'b' mean $\{ b \}$
Should be accepted

$\begin{cases} DFA = L \\ DFA' = \Sigma^* - L \end{cases}$

Note-

Complement is applicable only for DFA.

Q. Consider the NFA M shown below



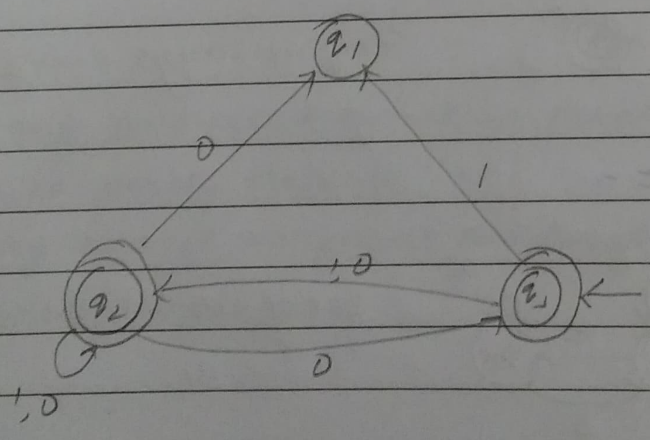
Let the language accepted by M be L. Let L_1 be the language accepted by the NFA M_1, obtained by changing the accepting state of M to be a non-accepting state and by changing the non-accepting state of M to accepting states. Which of the following statements is true?

× (a) $L_1 = \{0,1\}^* - L$

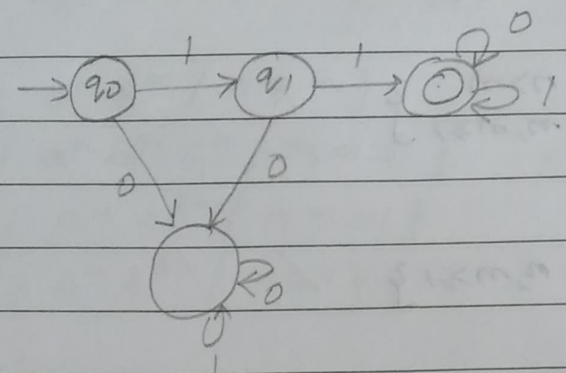
✓ (b) $L_1 = \{0,1\}^*$

× (c) $L_1 \subseteq L$

× (d) $L_1 = L$ $L \Rightarrow$ doesnot accept ϵ, but L_1 does



Q.



$11(0+1)^*$

How many n length strings accepted by given DFA?

a) $\{2^n \mid n \geq 1\}$
b) $\{2^{n-1} \mid n \geq 1\}$
c) $\{2^{n-2} \mid n \geq 2\}$ ✓
d) none

$n = 2$.

$2^{n-2} \Rightarrow 2^0 = 1$. 11 ✓

$2^{3-2} \Rightarrow 2^1 = 2$ 110 111

$2^{4-2} = 2^2 \Rightarrow$ 1100 01 10 11

DFA Construction.

1. Construct DFA for the language

$L = \{a^n b^m \mid n > m\}$

DFA not possible

Note -
DFA fails to accept language in which comparison exists.

E.g $L_1 = \{a^n b^m \mid n, m \geq 1\}$ ✓ DFA can be constructed

$L_2 = \{a^n b^n \mid n \geq 1\}$ ✗ $L_3 = \{a^n b^m \mid n \neq m\}$ ✗

Q. For which of the following DFA is possible.

$L_1 = \{ a^n b^m \mid n > m,\ n, m \geq 1 \}$ ✗

$L_2 = \{ a^n b^m \mid n, m \geq 1 \}$ ✓

$L_3 = \{ a^n b^n \mid n \geq 1 \}$ ✗

$L_4 = \{ a^n b^m \mid n \neq m \}$ ✗

$L_5 = \{ a^n b^{2m} \mid n, m \geq 1 \}$ ✓

$L_6 = \{ a^n b^{2n} \mid n \geq 1 \}$ ✗

03-06-21

DFA construction -
Continued ....

$L = \{ a^n b^n c^n \mid n \geq 1 \}$ ✗

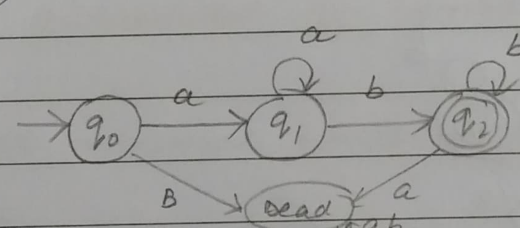
Note -
The languages for which DFA possible are called Regular languages.

Q. which of the following languages is Regular?

a) $L = \{ a^n b^m \mid n \neq m \}$
b) $L = \{ a^n b^m c^n \mid n, m \geq 1 \}$
✓ c) $L = \{ a^n b^{2m} \mid n, m \geq 1 \}$
d) $L = \{ a^n b^{2n} \mid n \geq 1 \}$

Q. construct minimal state DFA for the following language.

$L = \{ a^n b^m \mid n, m \geq 1 \}$



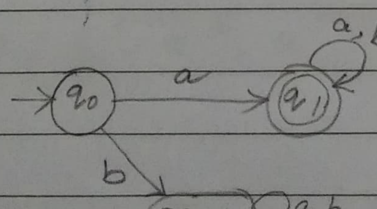
$\{ ab, aab, abb, aabb, abbb \ldots \}$

Steps-
1. Generate strings
2. Start from the initial state

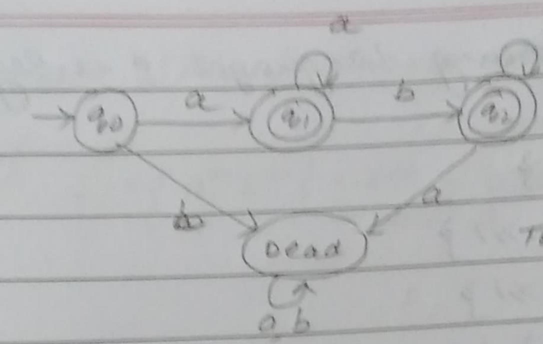
Q. construct a minimal state DFA for the following language.

$L = \{ a^n b^m \mid n \geq 1, m \geq 0 \}$

$\{ a, aa, aa \ldots$
$ab, abb, abbb \ldots \}$



This will accept aba ✗



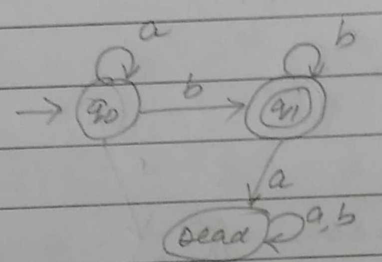
Total states = 4

Q. Construct minimal state DFA for the following language

$$L = \{a^n b^m \mid n \geq 0, m \geq 1\}$$

$\{ b, bb, bb \ \}$

$\{ ab, abb, aabb \ \}$



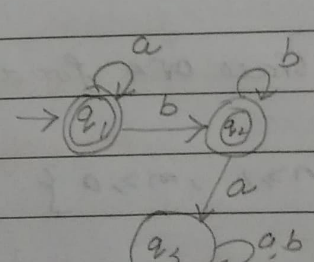
Q. Construct minimal state DFA for.

$$L = \{a^n b^m \mid n, m \geq 0\}$$

$\{ \epsilon, a, aa, aa \ldots$

$b, bb \ldots \}$

$ab, aabb. \}$



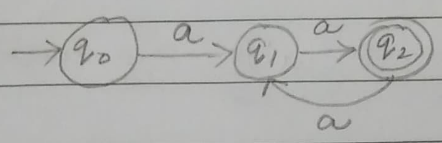
3 states

Q. Construct minimal state DFA for the following

$$L = \{ a^{2n} \mid n \geq 1 \}$$

$$\{ \underbrace{aa,}_{2}\ \underbrace{aaaa,}_{2}\ \underbrace{a^6\ \ a^8}_{2} \ \ldots \}$$

common diff should exist



Note:- For 1 symbol language.
Finite Automata fails to accept language in which there is no common difference between strings generated by symbol.

Q. Construct minimal state DFA for the following language

$$L_1 = \{ a^{2^n} \mid n \geq 1 \}$$

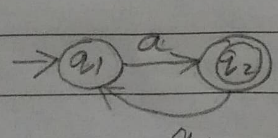
Not a Regular Language

$$L_2 = \{ a^{n^2} \mid n \geq 1 \}$$

Not a Regular Language

$$L_4 = \{ a^k \mid k \text{ is odd number} \}$$

$$\{ a, a^3, a^5, a^7, a^9 \ldots \}$$

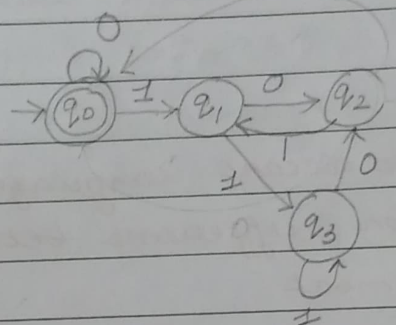


Q. Construct minimal state DFA that accepts set of all binary numbers which are divisible by 4.

Min states = 3

000 0
001 1
010 2
011 3

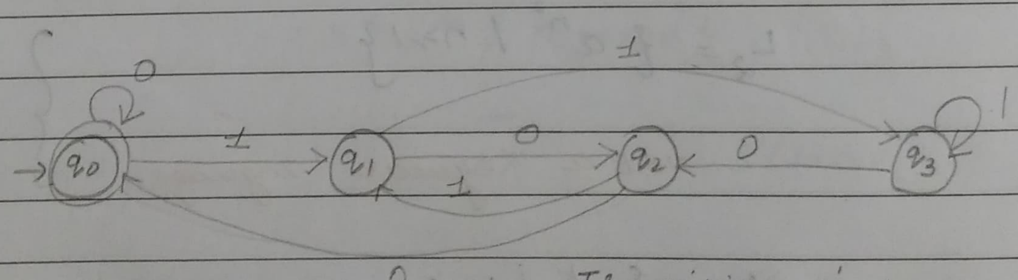
my work.



14

000 → 0 ✓
001 → 1 ×
010 → 2 ×
011 → 3 ×
100 → 4 ✓ /4

Venkatrao Sir's



Transition diagram.

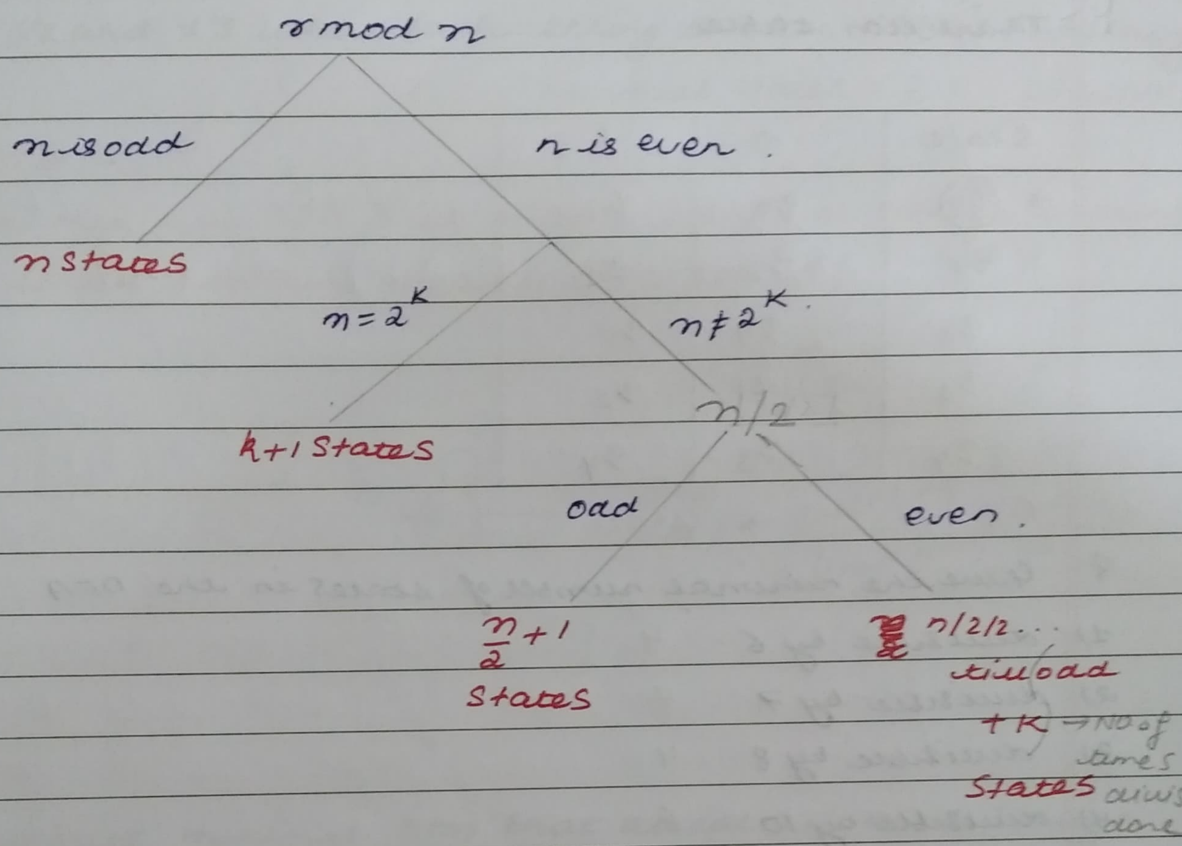
Transition table.

	0	1
→ q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_0	q_1
q_3	q_2	q_3

repeat

→ equal { reduce one }

** Binary Numbers



Minimum No. of States

04-06-21

$L = \{a^n b^m c^{n+m} \mid n, m \geq 1\}$

Not a regular language

aa

Q Construct minimal state DFA that accepts set of all binary numbers which are divisible by 5.

Transition table.

State	0	1
→ q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

repeat

Q Give the minimal number of states in the DFA.

1) divisible by 6 4
2) divisible by 7 7
3) divisible by 8 4
4) divisible by 10 6
5) divisible by 12 5
6) divisible by 16 5

A	1 mod 4	3
B	2 mod 5	5
C	3 mod 6	4
D	4 mod 8	4
E	0 mod 10	6
F	1 mod 12	5
G	3 mod 14	8
H	10 mod 16	5
I	4 mod 32	6
J	5 mod 64	7

Substring DFA

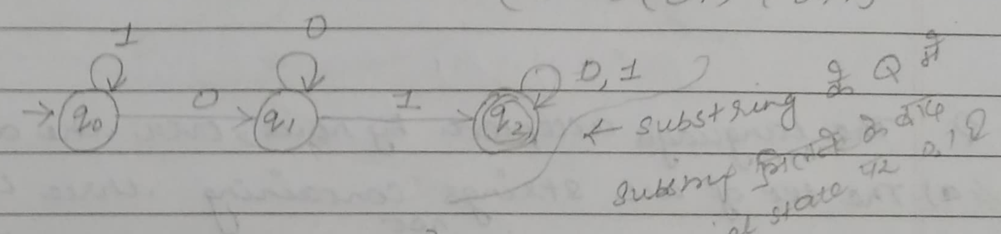
Q. Construct minimal state DFA that accepts all strings of 0's and 1's where each string contain 01 as substring.

Minimal States = 3   NO dead states

NOTE:-

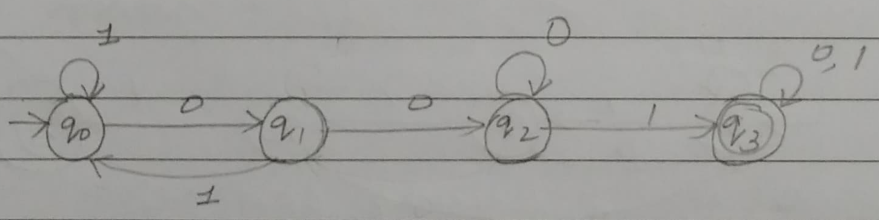
The minimal DFA that accepts set of all strings contains N length substring require N+1 states.

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



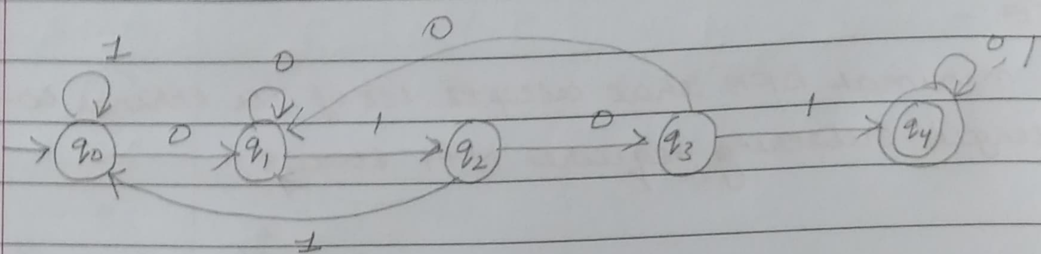
{01, 001, 010, 101, 011}

Q. Construct minimal DFA that accept all ~~alone~~ strings of 0's and 1's where each string contain 001 as substring.



{001, 0001, 0101001 }

Q. Construct minimal state DFA that accept all strings of 0's and 1's where each string contain 0101 as substring.



Q. The language accepted by M is, over the alphabet {0,1}

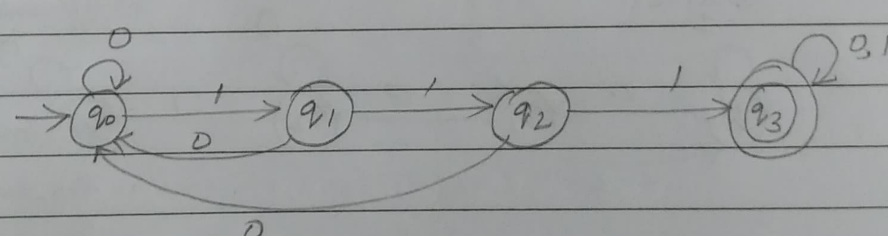
✓ a) The set of all strings containing three consecutive 1's

b) The set of all strings not containing three consecutive 1's

c) The set of all strings beginning with three consecutive 1's

d) The set of all strings ending with three consecutive 1's.

PSU



Options -

X (b) 111 ✓

X (c) 0111 ✓

X (d) 1110 ✓

(a) ✓

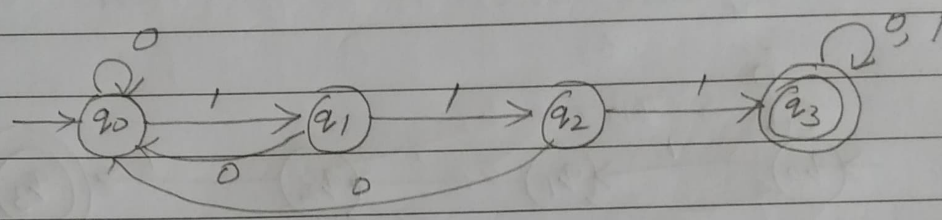
Q. Let S denotes the set of all six bit binary strings in which first and fourth bits are 1. The no. of strings in S that are accepted by M is

(A) 1

(B) 4

(C) 7

✓(D) 8



accepts substring 111

Total possible :-

1 _ _ 1 _ _ } 16

100100 ×	110100 ×
100101 ×	110101 ×
100110 ×	110110 ×
100111 ✓	110111 ✓
101100 ×	111100 ✓
101101 ×	111101 ✓
101110 ✓	111110 ✓
101111 ✓	111111 ✓

classmate

Date

Page

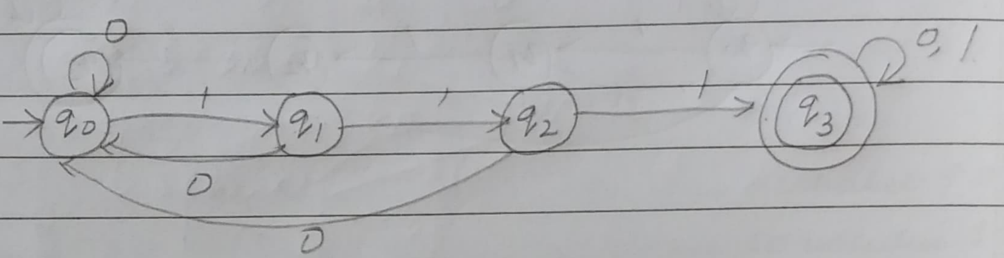
Q. Let S denotes the set of all six bit binary strings in which first and fourth bits are 1 accepted by the machine which is obtained by interchanging final and nonfinal states in M. The no. of strings in S is

(A) 1

(B) 4

(C) 7

(D) 8 ✓/



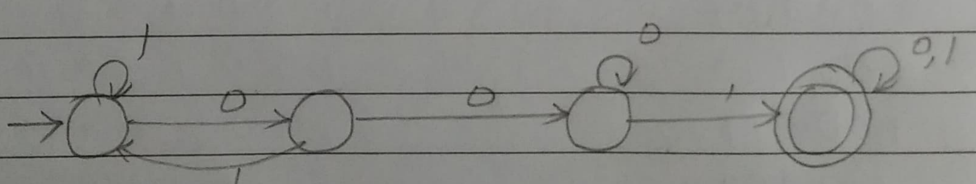
uses ⇒ complement

Total 16 ⇒ Accepted ⇒ 8

Now in complement 8 rejected

∴ 16 - 8 = Rejected

Q. Consider the following DFA M:



Let S denote the set of seven bit binary strings in which the first and the fourth and the last bits are 1. The no. of strings in S that are accepted are

(a) 1 (d) 8

(b) 5

(c) 7 ✓/

Total possible 1 _ _ 1 _ _ 1
⇒ 16

1 00 1 00 1 ✓	1 10 1 00 1 ✓
1 00 1 01 1 ✓	1 10 1 01 1
1 00 1 10 1 ✓	1 10 1 10 1
1 00 1 11 1 ✓	1 10 1 11 1
1 01 1 00 1 ✓	1 11 1 00 1 ✓
1 01 1 01 1	1 11 1 01 1
1 01 1 10 1	1 11 1 10 1
1 01 1 11 1	1 11 1 11 1

Should contain substring 001

05-06-21

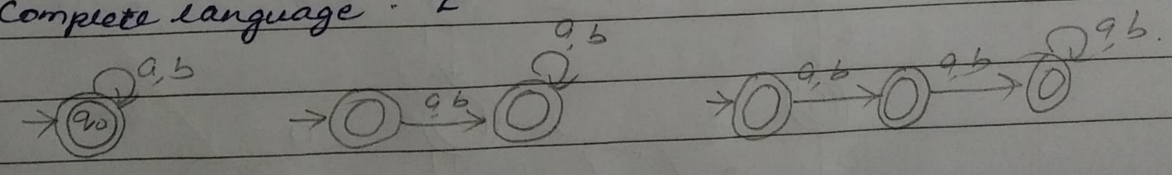
* Binary numbers divisible by 8 = $2^{3+1} = 4$
* Binary numbers divisible by 20 = $20/10 = 10/2 = 5+2 = 7$
* Each string having 'aaba' as substring = 5 states.

* # Minimization of DFA

1. Correct DFA
2. Minimal state DFA.

For a given regular language even though many DFA exist, but minimal states DFA is unique

Ex- Complete language : Σ^*



Minimal DFA

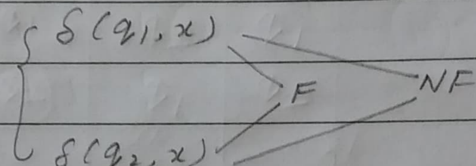
Minimization Algorithm (Only for DFA)

1. State Equivalence Algorithm
2. Table Filling Algorithm.

✱ Equivalent states -

Two states (q_0, q_1) are said tobe equivalent both $\delta(q_0, x)$ and $\delta(q_2, x)$, $x \in \Sigma^*$ should result either final state (or) non-final state

Merge them



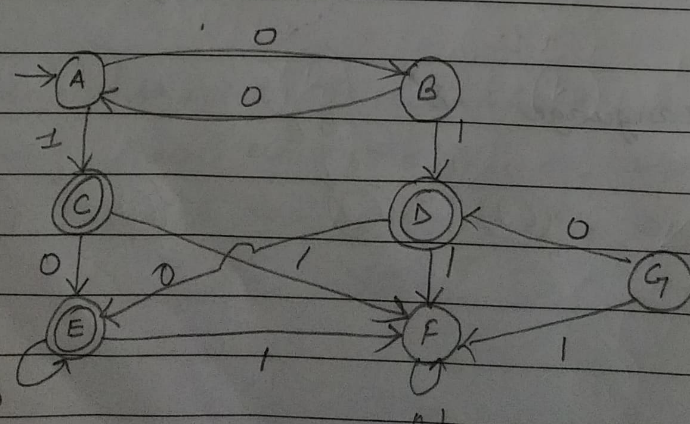
✱ Equal states → (Best friends)

Procedure -

1. Eliminate inaccessible states - Any state which is not reachable from initial state is inaccessible state
2. Apply algorithm steps
3. Merge single group into one state
4. Construct new minimized DFA

Dead state is different from inaccessible state

Example :-



STEP 1: Eliminate inaccessible state – Remove G

STEP 2:

State	0	1
A	B	C
B	A	D
F	F	F
C	E	F
D	E	F
E	E	F

S Algorithm –

Non Final in 1 group   Final in 1 group.

1. { A, B, F }   { C, D, E }

2. Check Equivalence. Splitting

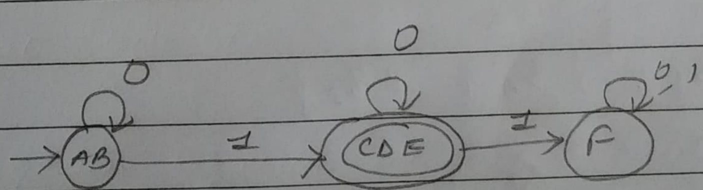
{ A, B } { F } { C, D, E }

same {stop}

3. { A, B } { F } { C, D, E }

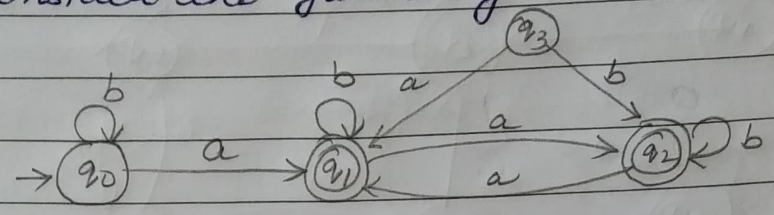
Every group ⇒ One state

STEP 3:



Minimal State DFA

Q. Consider the following Finite State Automation



Minimize Given DFA

STEP2:

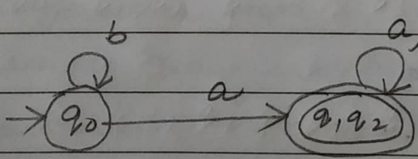
State	a	b
→ q_0	q_1	b
q_1	q_2	q_1
q_2	q_1	q_2

STEP1: Remove q_3

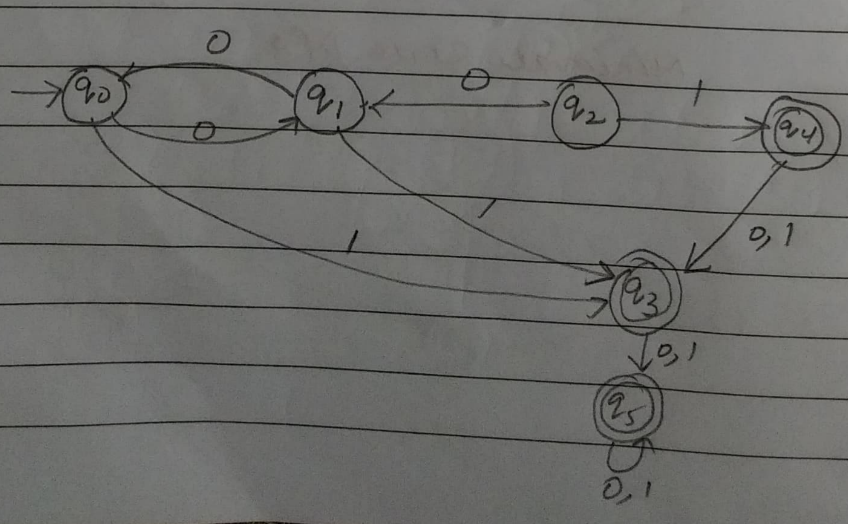
STEP3: $\{q_0\}$ $\{q_1, q_2\}$

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

STEP:4:



Q.



STEP1: Remove q_2, q_4

STEP2:

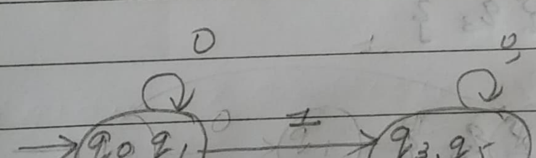
State	0	1
→ q_0	q_1	q_3
q_1	q_0	q_3
q_3	q_5	q_5
q_5	q_5	q_5

STEP3:

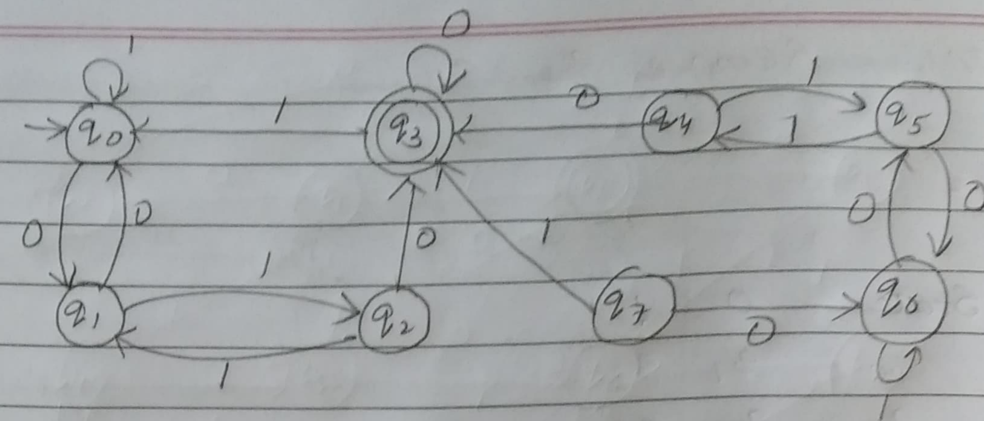
$\{q_0, q_1\}$ $\{q_3, q_5\}$

$\{q_0, q_1\}$ $\{q_3, q_5\}$

STEP4:



Q.



STEP 1: Remove q_4, q_5, q_6, q_7

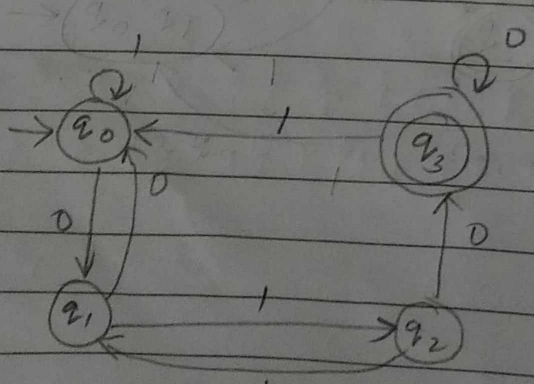
STEP 2:

State	0	1
→ q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0

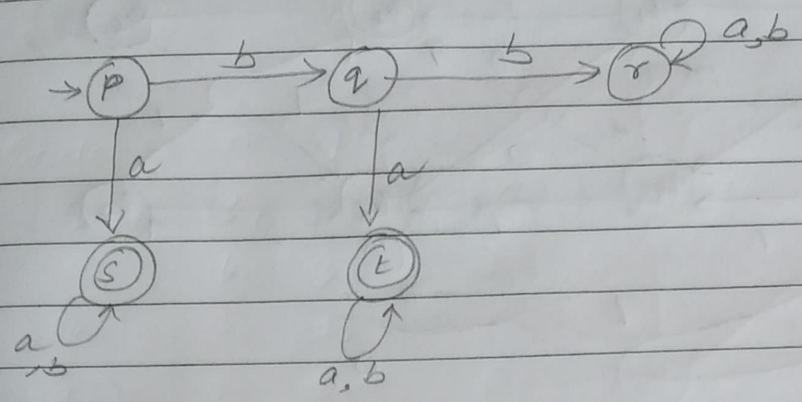
STEP 3: $\{q_0, q_1, q_2\}$ $\{q_3\}$

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

STEP 4: $\{q_0\}$ $\{q_1\}$ $\{q_2\}$ $\{q_3\}$



Q. How many states in minimal DFA?



STEP 1: No unreachable.

STEP 2:

	State	a	b
→	p	s	q
	q	t	r
	r	r	r
	s	s	s
	t	t	t

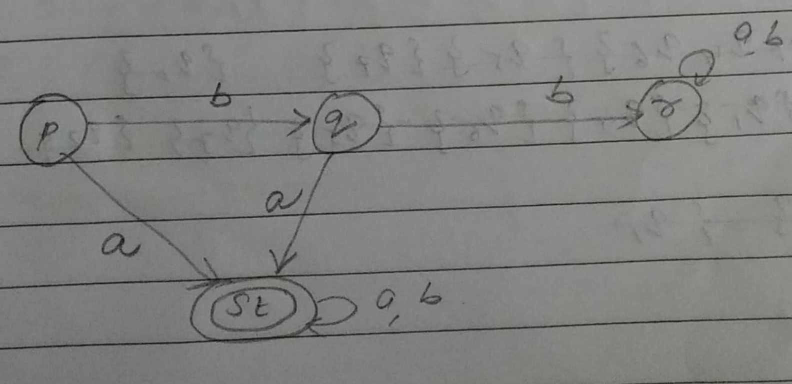
STEP 3:

{pqr} {st}

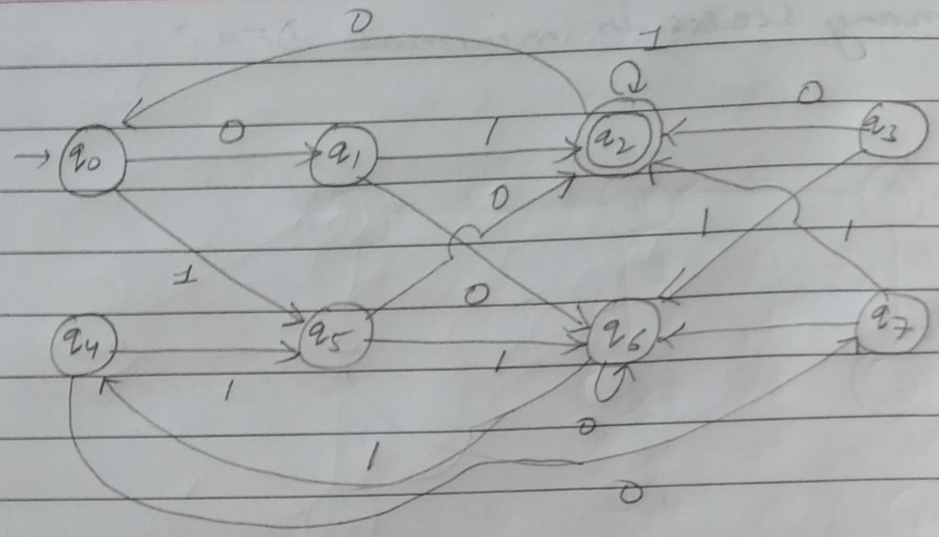
{pq} {r} {st}

{p} {q} {r} {st}

STEP 4:



Q.



STEP1: Remove q_3

STEP2:

State	0	1	
→ q_{04}	q_{17}	q_5	
q_{17}	q_6	q_2	
q_4	q_{17}	q_5	X
q_5	q_2	q_6	
q_6	q_6	q_{04}	
q_7	q_6	q_2	X
q_2	q_0	q_2	

STEP3:

$\{q_0 \; q_1 \; q_4 \; q_5 \; q_6 q_7\}$ $\{q_2\}$

~~$\{q_0 \; q_1 \; q_4 \; q_6\}$ $\{q_5\}$ $\{q_7\}$~~ $\{q_2\}$

~~$\{q_0\}$ $\{q_1\}$ $\{q_4\}$ $\{q_6\}$ $\{q_5\}$ $\{q_7\}$ $\{q_2\}$~~

~~$\{q_0 \; q_4\}$ $\{q_1$~~

$\{q_{04}, q_{17}, q_5, q_6\}$ $\{q_2\}$

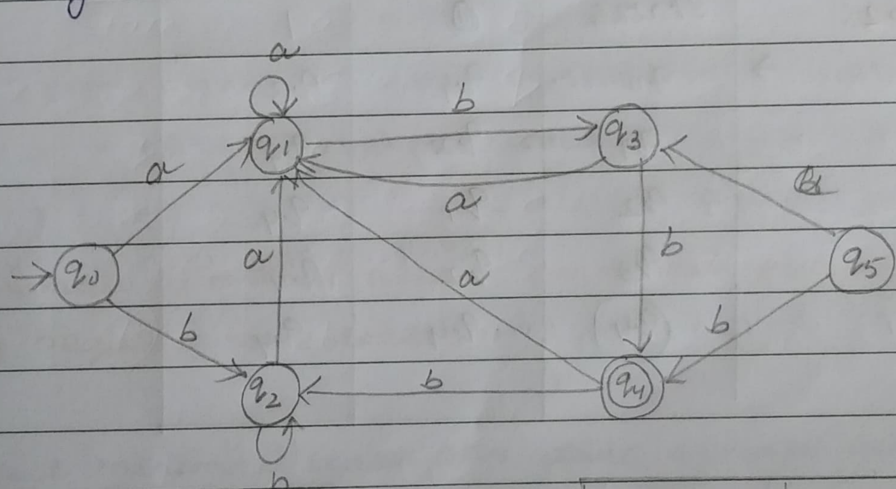
$\{q_{04}, q_6\}$ $\{q_{17}\}$ $\{q_5\}$ $\{q_2\}$

$\{q_{04}\}$ $\{q_6\}$ $\{q_{17}\}$ $\{q_5\}$ $\{q_2\}$

5 States.

07-06-21

Q How many states in minimal DFA for the following DFA?



STEP1: Remove q_5

STEP2: →

State	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	q_4
q_4	q_1	q_2

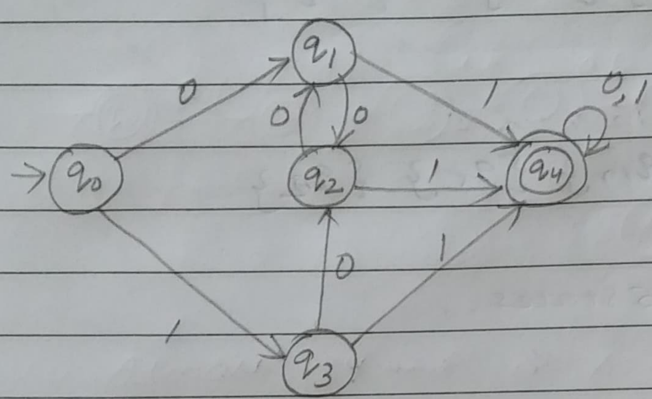
STEP3: $\{q_0\ q_1\ q_2\ q_3\}$ $\{q_4\}$

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

$\{q_0\ q_2\}$ $\{q_1\}$ $\{q_3\}$ $\{q_4\}$

Minimum states → 4

Q. How many states in the minimal DFA?



STEP1: No unreachable state

STEP2:

States	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_4
q_2	q_1	q_4
q_3	q_2	q_4
q_4	q_4	q_4

STEP3:

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

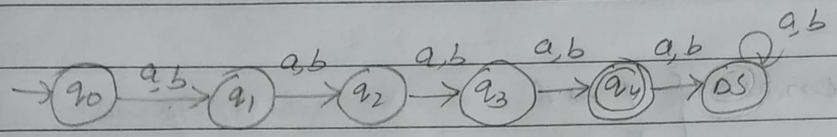
$\{q_0\}\ \{q_1, q_2\ q_3\}\ \{q_4\}$

Minimum states = 3

Q. Construct minimal state DFA that accepts all strings of a's and b's where length of each string exactly 4.

a/b a/b a/b a/b.

16 strings.

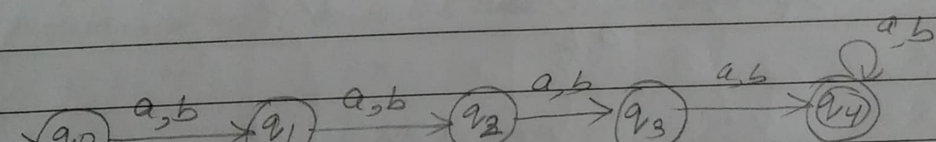


length exactly $\begin{cases} 4 \rightarrow 6 \text{ states} \\ 5 \rightarrow 7 \text{ states} \\ 7 \rightarrow 9 \text{ states} \end{cases}$

Exactly $n \rightarrow n+2$ states

The minimal DFA that accepts set of all exactly n length strings require $n+2$ states.

Q. Construct minimal state DFA that accepts all strings of a's and b's where length of each string is atleast 4.

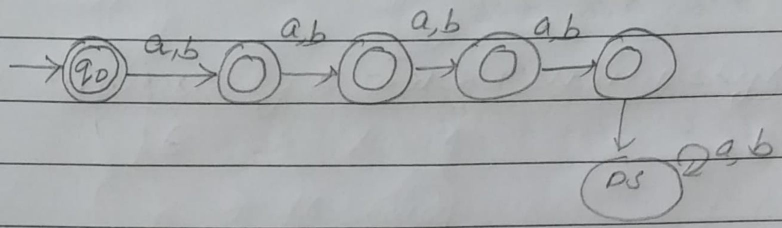


length ~~or~~ atleast $\begin{cases} 4 \rightarrow 5 \text{ states} \\ 5 \rightarrow 6 \text{ states} \end{cases}$

Atleast $n \rightarrow n+1$ states.

The minimal DFA that accepts set of all atleast n length strings require $n+1$ states

Q. Construct minimal DFA that accept all strings of a's and b's where length of each string is atmost 4.



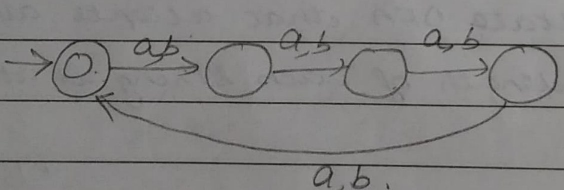
Atmost $\begin{cases} 4 \rightarrow 6 \\ 5 \rightarrow 7 \end{cases}$

Atmost n length $\rightarrow n+2$ states

Q. Construct minimal state DFA that accepts all strings of a's and b's where length of the each string is divisible by 4.

Don't confuse with binary strings / by 4
Both are different

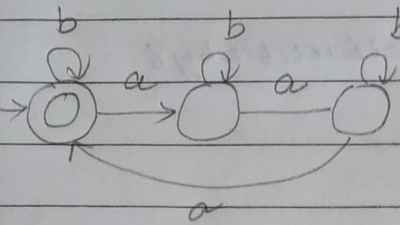
Lengths / by 4 $\rightarrow$ { 0, 4, 8, 12, 16 .



divisible by $\begin{cases} 4 \rightarrow 4 \\ 5 \rightarrow 5 \end{cases}$

divisible by $n \rightarrow n$ states

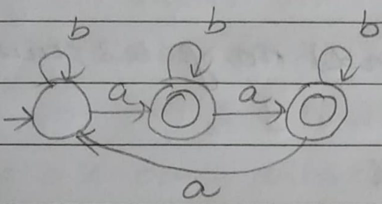
Q. Construct minimum DFA over $\Sigma = \{a, b\}$ where a's divisible by 3.



पहले a's का बनाओ।

b's के लिये सारी states पर selfloop.

not divisible by 3. - Complement



Q. Construct minimal states DFA that accepts all strings of a's and b's where

a's divisible by 3 and b's divisible by 4.

3 × 4

⇒ 12 states

a's divisible by 4 and b's not divisible by 5

4 × 5

⇒ 20 states

a's divisible by 3 and b's atleast 2.

3 × 3

⇒ 9 states

a's exactly 3 and b's atleast 4

(5-1) × 5

⇒ 20 + 1 ⇒ 21 states

Remove deadstate

add dead state

a's exactly 4 and b's atmost 3

$(6-1)=5 \quad \times \quad (5-1)=4$

$= 20+1 \Rightarrow 21$ states

a's divisible by 6 and b's divisible by 8

$6 \quad \times \quad 8$

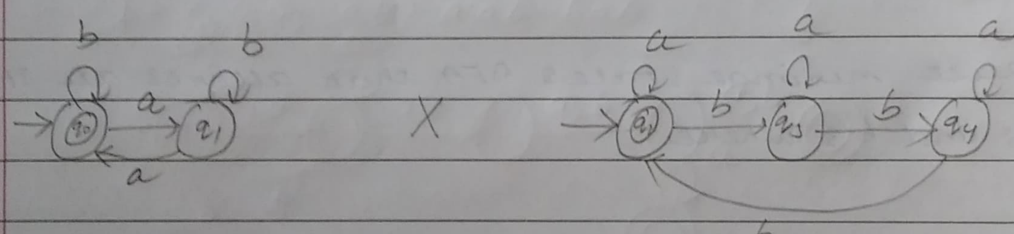
$\Rightarrow 48$ states

wherever you see 'and' $\Rightarrow$ X multiply

Q. (DFA a's divisible by 2) (and) no. of b's divisible by 3

$2 \quad \times \quad 3$

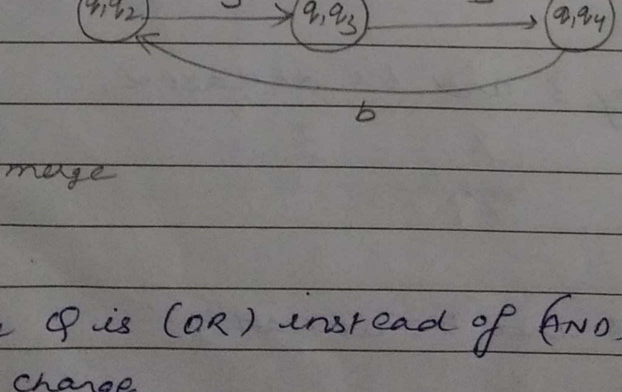
$= 6$



CROSS PRODUCT

$\Rightarrow$

$\{Q, \Sigma\, q_0\, F, S\}$



$\delta(q_0, a) = q_1$

$\delta(q_2, a) = q_2$

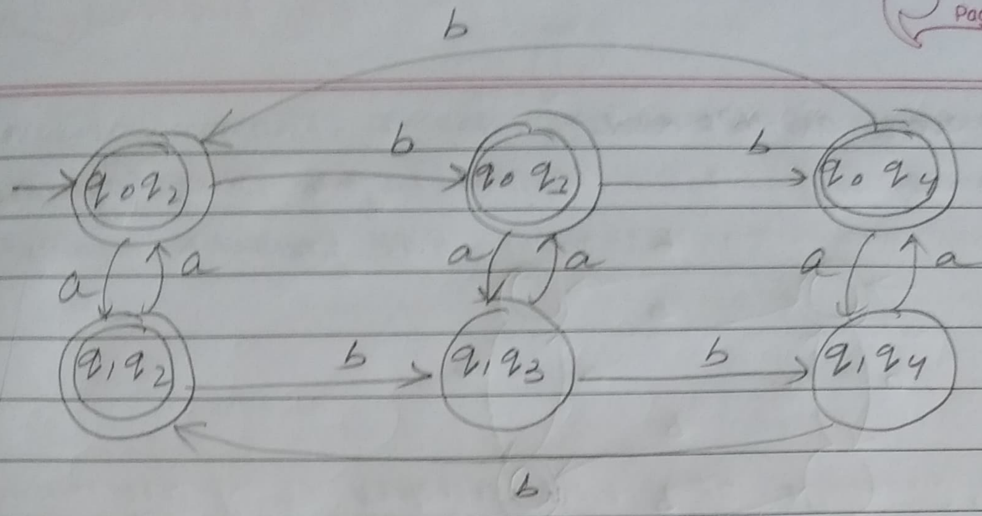
$\delta(q_0, b) = q_0$

$\delta(q_2, b) = q_3$ } merge

If the Q is (OR) instead of (AND) then final states change.

AND $\Rightarrow$ Both final.

OR $\Rightarrow$ Any one final will work.



Q. No. of a's even and no. of b's odd

Q. No. of a's odd (or) no. of b's even

Q. No. of a's odd and no. of b's odd.

10-06-21

Q. No. of a's exactly 4 and b's at least 2.

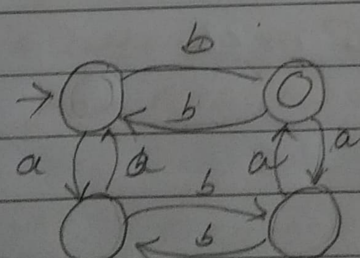
5 × 3

15 + 1 ⇒ 16 states.

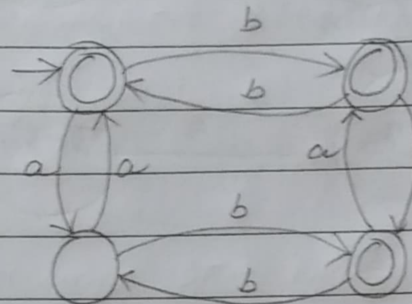
↑ dead

Q. a's even and b's odd

2 × 2 ⇒ 4

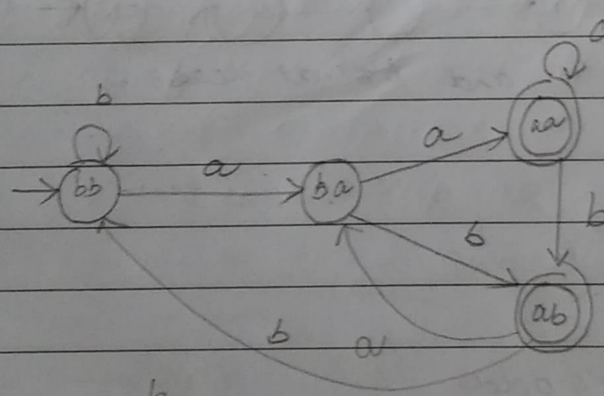


Q a's even or b's odd



Q. Construct minimal state DFA that accept set of all strings of a's and b's where ~~third~~ second input symbol is 'a' while reading string from right hand side.

L = { aa, ab, aab, bab . . . }



bbabab

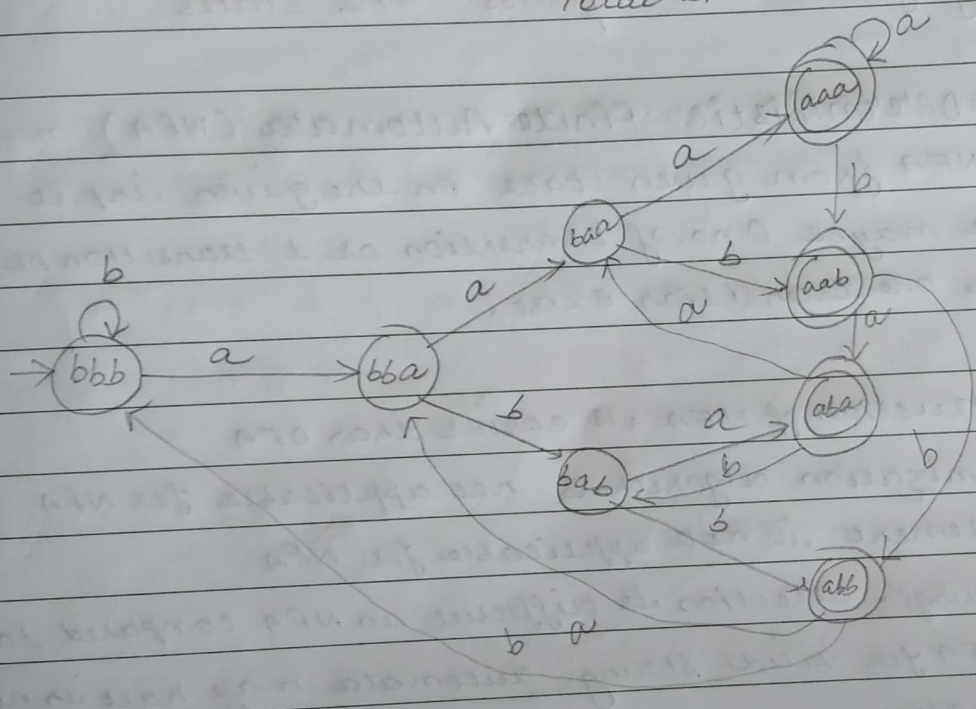
Ending with aa or ab.

Q. Construct minimal state DFA, that accept set of all strings of a's & b's where third input symbol is 'a' while reading string from right

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

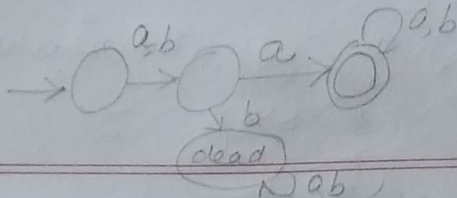
$(a+b)^*$ aaa
aab
aba
abb

Total states = $2^3 = 8$



NOTE:
The minimal DFA that accepts set of all strings of a's and b's where n^{th} i/p symbol is a while reading string from RHS requires $\underline{2^n}$ states.

E.g 10^{th} i/p symbol from RHS is $\Rightarrow$ 1024 states



Second from LHS is a

NOTE :

1. DFA is difficult to construct for some regular languages.
2. Hence ~~DFA~~ NFA is constructed for such type of regular languages.

The minimal DFA that accepts set of all strings of a's and b's where n^{th} i/p symbol is 'a' while reading string from LHS requires $n+2$ states.

2. Non-Deterministic Finite Automata (NFA)

In NFA from given state on the given input symbol there maybe 0 no. of transition or 1 transition or more than one transition exist.

→ Construction of NFA is easier than DFA
→ Minimization algorithms not applicable for NFA
→ Complement is not applicable for NFA.
→ Language detection is difficult in NFA compared to DFA.
→ In NFA for valid string automata may halt in non-final state also.
→ All DFA are NFA
→ All DFA need not be NFA

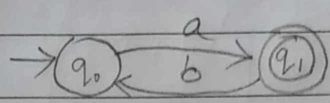
Formal Definition of NFA -

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

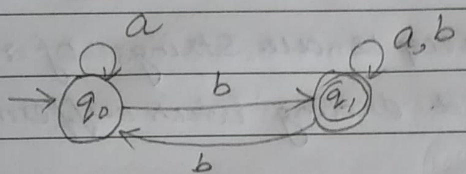
Q : Finite set of states
Σ : input alphabet
q_0 : initial state
F : Set of final states
δ : Transition function.

$\boxed{Q \times \Sigma \to 2^Q}$ → power set of Q

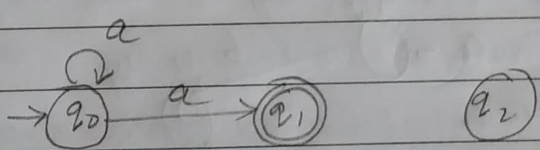
Example -



Example -



why $Q \times \Sigma \to 2^Q$?

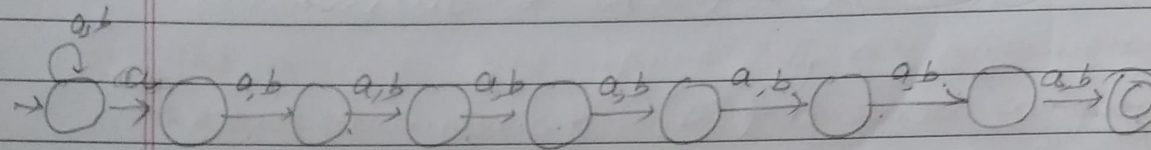


$\delta(q_0, a) = \{\}\ \{q_0\}\ \{q_1\}\ \{q_0, q_1\}$

if introduced

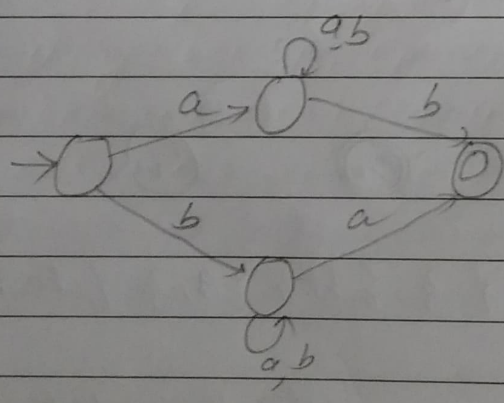
$\delta(q_0, a) = \{\}\ \{q_0\}\ \{q_1\}\ \{q_2\}\ \{q_0, q_1\}\ \{q_1, q_2\}\ \{q_0\ q_2\}$
$\{q_1, q_0, q_2\}$

Q. Construct NFA that accepts all strings of a's and b's where 8th i/p symbol is a while reading String from RHS



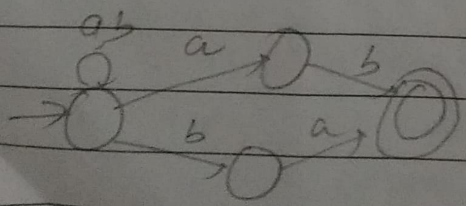
Q. Construct NFA that accepts all Strings of a's and b's each string starting and ending with different symbol.

[a (a+b)* b] or [b (a+b)* a]



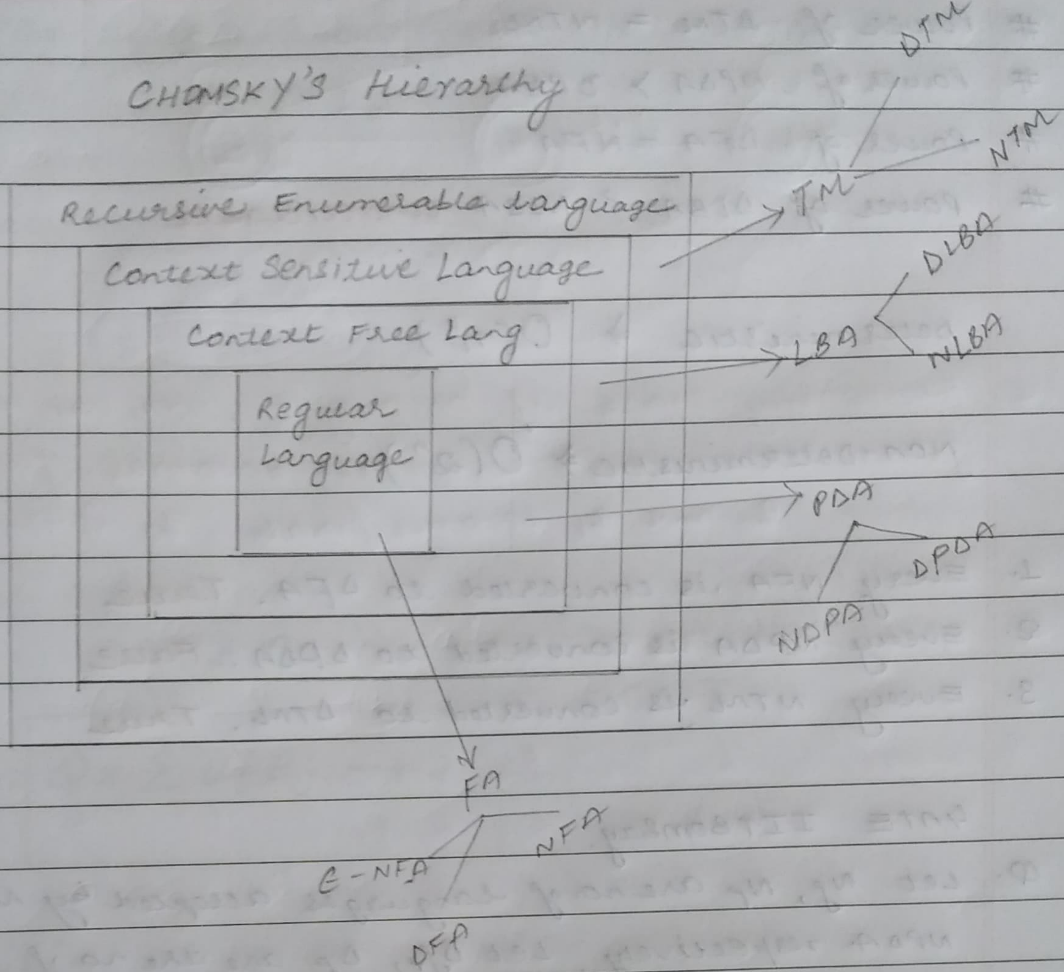
Q. Construct NFA that accepts all strings of a's & b's where last two symbols are different

(a+b)* a/b



11-06-2021

CHOMSKY'S Hierarchy



1. Every contextfree language is Regular language. False
2. Every Regular Language is Recursive Enumerable. True
3. Every Context ~~Free~~ Sensitive Language is Contextfree Language. False

Expressive Power :

No. of languages accepted by a particular automata.

TM	>	LBA	>	PDA	>	F.A
REL + CSL + CFL + Regular		CFL + CSL + Regular		CFL + Regular		Regular

TM is The most powerful machine

Power of DTM = NTM
Power of NPDA > DPDA
Power of DFA = NFA
Power of DLBA ? NLBA undecidable

Deterministic → $O(n^k)$

Non-Deterministic → $O(2^n)$

1. Every NFA is converted to DFA. True
2. Every NPDA is converted to DPDA. False
3. Every NTM is converted to DTM. True

GATE IIT Bombay.

Q. Let N_f, N_p are no. of languages accepted by NFA and NPDA respectively. Let D_f, D_p are the no. of languages accepted by DFA and DPDA respectively. Which of the following is True?

(A) $D_f = N_f$
$D_p = N_p$

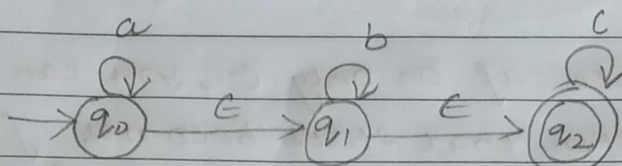
(B) $D_f \subset N_f$
$D_p \subset N_p$

(C) $D_f = N_f$
$D_p \supset N_p$

✓(D) None. $D_f = N_f$ and $D_p \subset N_p$

3. ε-Non-deterministic Finite Automata
(ε-NFA)

Example -



$\{a^n b^m c^k\}$

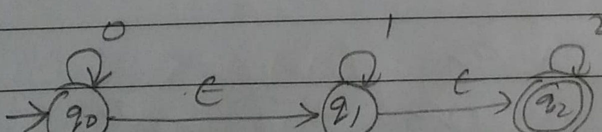
Formal Definition -
$(Q, \Sigma, q_0, F, \delta)$

$$\boxed{Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q}$$

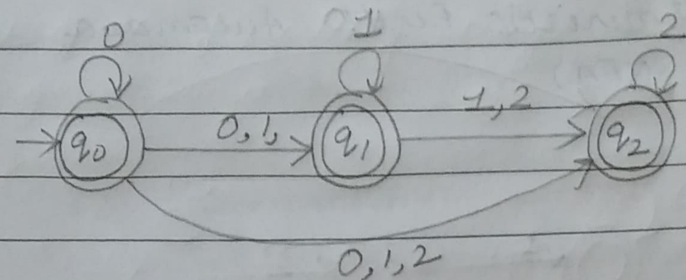
Conversion of ε-NFA to NFA

Note -
while converting from NFA to DFA for the given n states NFA, no. of states possible in DFA is $\{1 \ldots 2^n\}$
i.e max $\Rightarrow 2^n$

Q. Construct an equivalent NFA for the following ε-NFA



1. No. of states are same
2. Initial state is same
3. Final states may increase
4. Transitions may increase



1. From any state if on only ϵ, you can reach the final state, then that state becomes final.

   q_0 and q_1 becomes final : q_2 तो पहले से ही था।

2. Mark the transitions

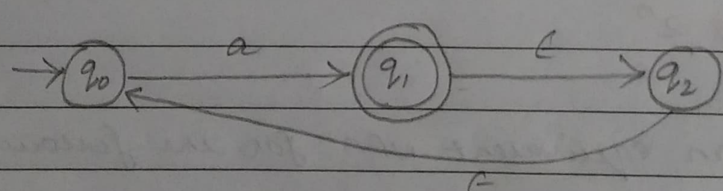
   $\epsilon \cdot 0 = 0$

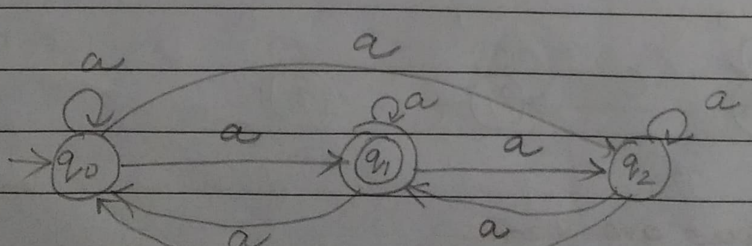
   $\epsilon\ 1 = 1$

   $\epsilon\ 1 \cdot \epsilon = 1$

   $\epsilon\ \epsilon = \epsilon$

Q. Construct an equivalent NFA for the following ϵ-NFA.
GATE

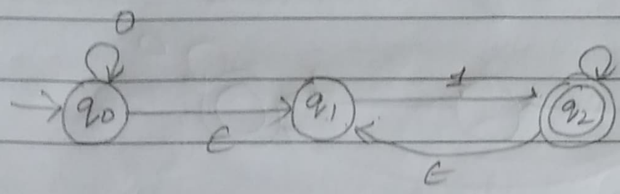


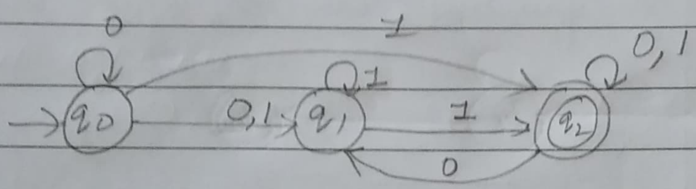


$a\epsilon = a$

$a\epsilon\epsilon = a$

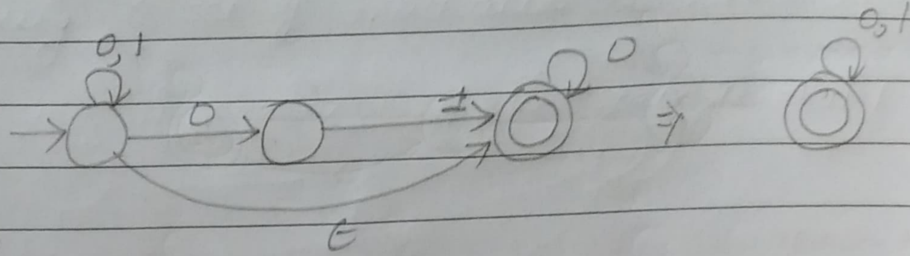
Q. Construct an equivalent NFA for the following ε-NFA?





12-06-2021

Q. Which of the following strings is rejected by the NFA given below:



(a) 0

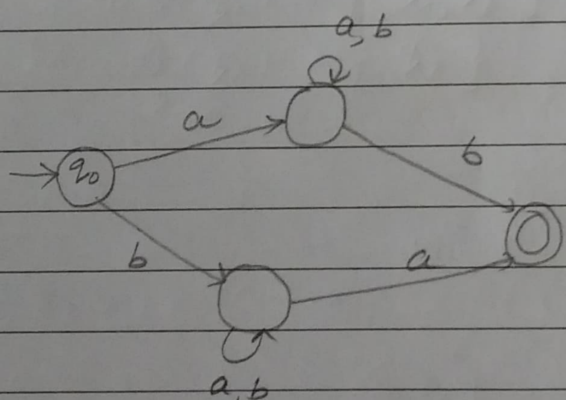
(b) 101000

(c) 0100

(d) 01011

(e) None

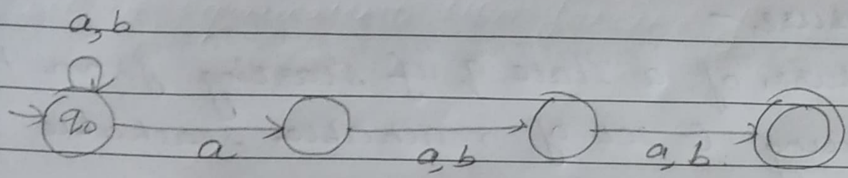
Q. What is the language accepted by following NFA.



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

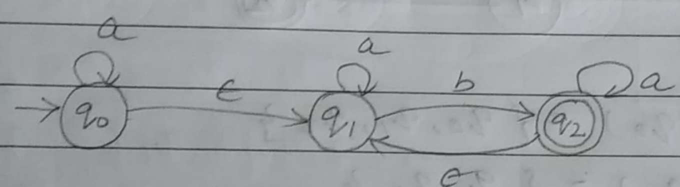
Starting and ending with different symbols.

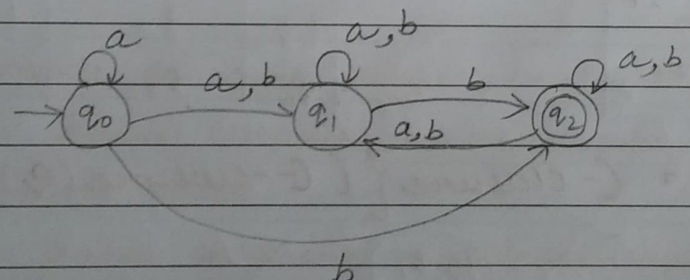
Q.



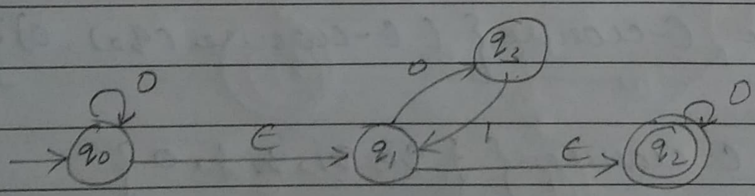
Third symbol from RHS is 'a'

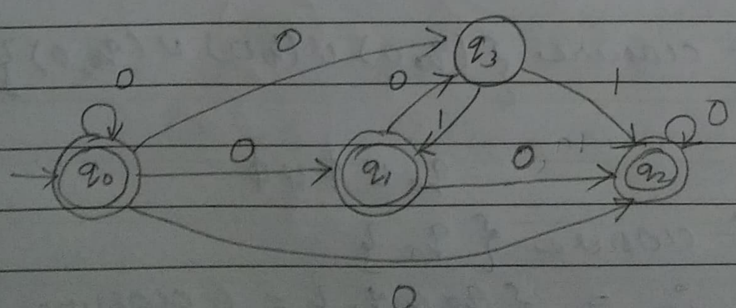
Q. Construct NFA for the following ε-NFA?





Q.



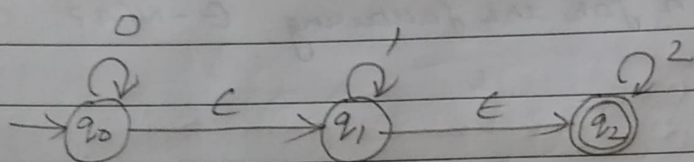


ε closure -

ε closure of a state q if starting from that state by reading ε set of reachable reachable states.

किसी state से ε लेकर किसी और state पर जाना।

Example :-



ϵ - closure $(q_0) = \{q_0, q_1, q_2\}$

ϵ- closure $(q_1) = \{q_1, q_2\}$

ϵ- closure $(q_2) = \{q_2\}$

Theorem :-

$$\delta(q, x) = \epsilon\text{ closure}\{(\epsilon\text{-closure}(q), x)\}$$

Example -

$\delta(q_0, 0) = \epsilon\text{ closure}\{(\epsilon\text{-closure}(q_0), 0)\}$

$\epsilon\text{ closure}\{\{q_0, q_1, q_2\}, 0\}$

$\epsilon\text{ closure}\{(q_0, 0) \cup (q_1, 0) \cup (q_2, 0)\}$

$q_0 \cup \phi \cup \phi$

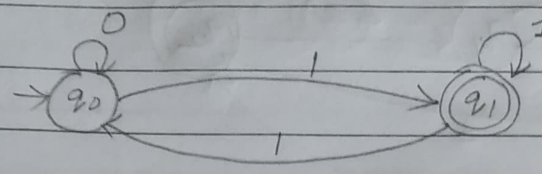
ϵ- closure $\{q_0\}$

$\therefore \quad = \{q_0\ q_1\ q_2\} = \epsilon$ closure

NFA to DFA Conversion

(Subset ~~Conversion~~ Construction Algorithm)

Q. Construct an equivalent DFA for the given NFA.



NFA

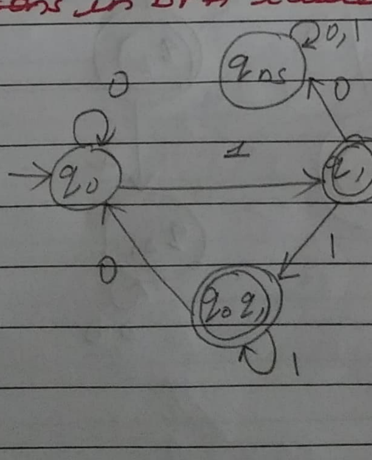
Table

	0	1
$\rightarrow q_0$	q_0	q_1
q_1	ϕ	q_0, q_1

DFA Table

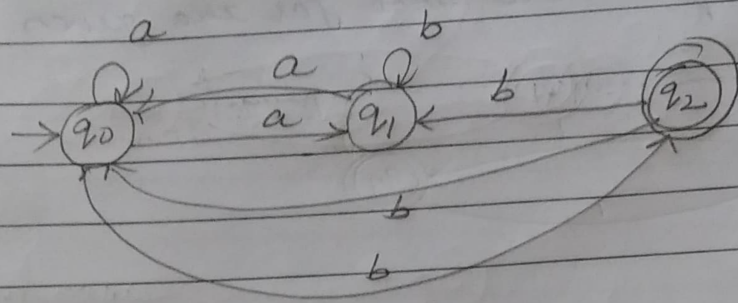
	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_{DS}	$[q_0 q_1]$
q_{DS}	q_{DS}	q_{DS}
$[q_0 q_1]$	q_0	$[q_0 q_1]$

1. If NFA table contains single state then copy same state.
2. If NFA table contains no transition, then add dead state in DFA table.
3. If NFA table contains multiple transitions then add new ~~transitions~~ states in DFA table.



May-may not be minimal

Q. Construct an equivalent DFA for the given NFA

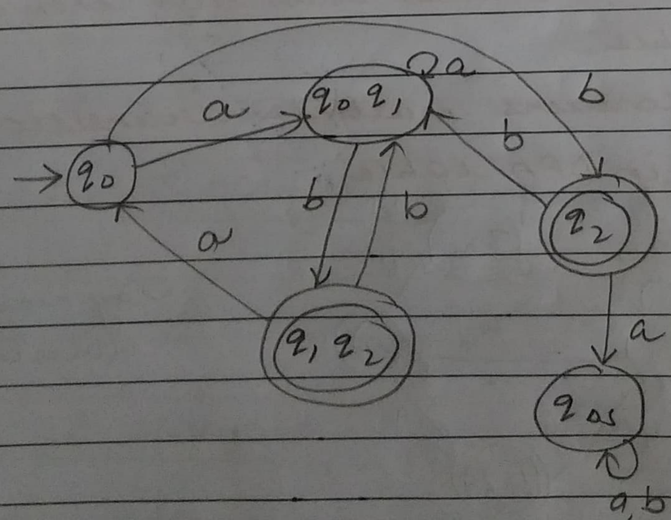


NFA table

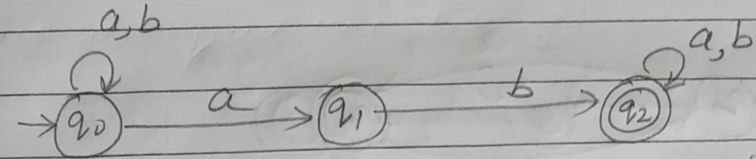
	a	b
$\rightarrow q_0$	q_0, q_1	q_2
q_1	q_0	q_1
q_2	-	q_0, q_1

DFA table

	a	b
$\rightarrow q_0$	$[q_0 q_1]$	q_2
$[q_0 q_1]$	$[q_0, q_1]$	$[q_1 q_2]$
q_2	q_{DS}	$[q_0, q_1]$
$q_1 q_2$	q_0	$[q_0 q_1]$
q_{DS}	q_{DS}	q_{DS}



Q. Construct an equivalent DFA for the given NFA.



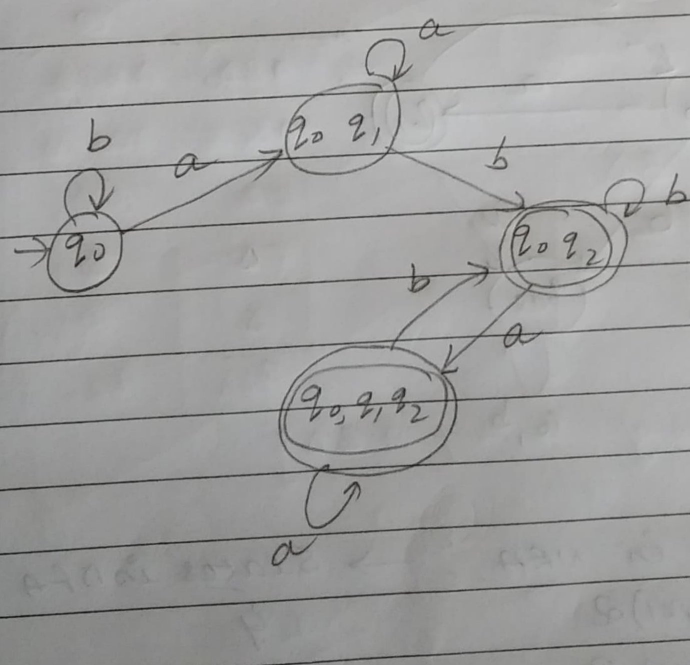
Substring ab problem

NFA table.

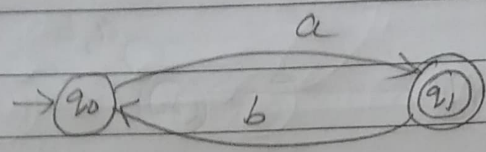
	a	b
→ q_0	$q_0 q_1$	q_0
q_1	–	q_2
q_2	q_2	q_2

DFA table

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



Q. Construct an equivalent DFA for the given NFA

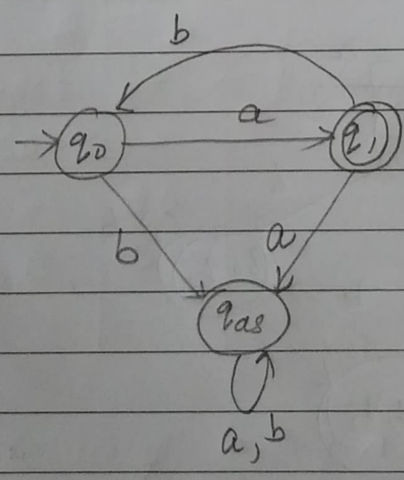


NFA table

	a	b
→ q_0	q_1	—
q_1	—	q_D

DFA table

	a	b
→ q_0	q_1	q_{DS}
q_1	q_{DS}	q_D
q_{DS}	q_{DS}	q_{DS}



States in NFA → States in DFA

Example1) 2   4

Example2) 3   5

Empable3) 3   4

Example4) 2   3

⋮

n . . . . . $\{1 \dots 2^n\}$

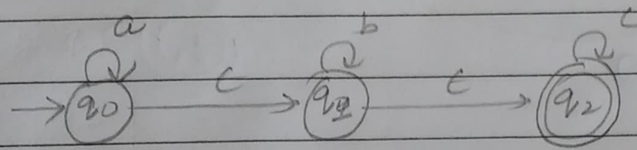
** Time complexity of subset construction Algorithm

Best case ⇒ $O(1)$
Worst case ⇒ $O(2^n)$

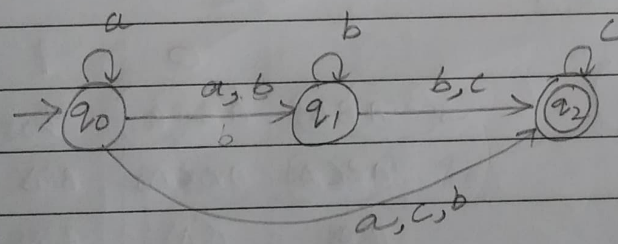
13-06-21

ϵ NFA to DFA Conversion

Q. Construct DFA equal to following ϵ NFA



ϵ NFA



NFA Table

	a	b	c
→ q_0	q_0, q_1, q_2	q_1, q_2	q_2
q_1	—	q_1, q_2	q_2
q_2	—	—	q_2

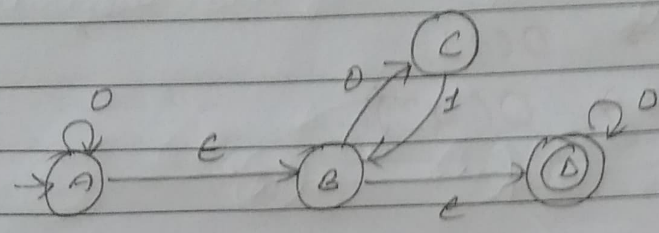
DFA Table

	a	b	c
→ $[q_0 q_1 q_2]$	$[q_0, q_1, q_2]$	$[q_1, q_2]$	q_2
$[q_1, q_2]$	q_{DS}	$[q_1, q_2]$	$[q_2]$
$[q_{DS}]$	q_{DS}	q_{DS}	q_{DS}
$[q_2]$	q_{DS}	q_{DS}	q_2

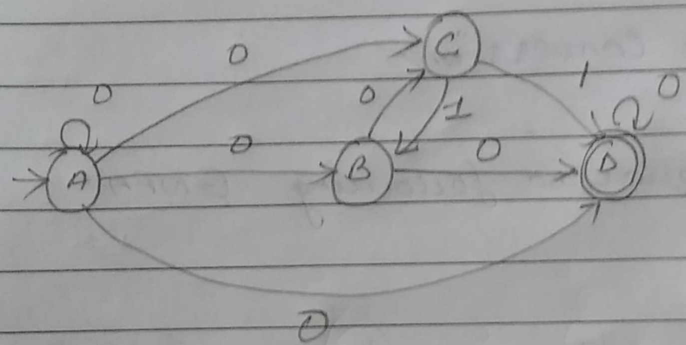
* DFA initial state = ϵ closure (Initial state)

ϵ closure$(q_0) = \{q_0, q_1, q_2\}$

Q. Construct DFA for the following ε NFA?



ε NFA



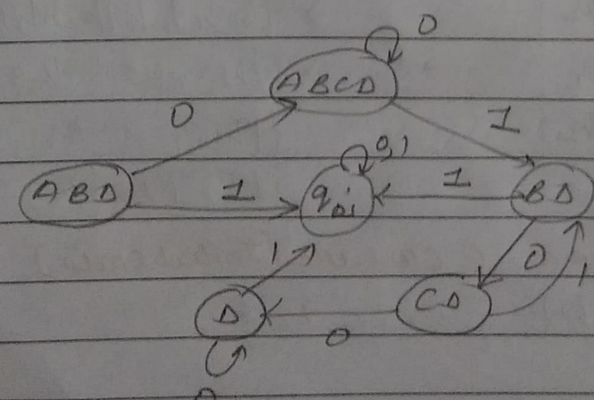
NFA Table

	0	1
→ A	ABCD	–
B	CD	–
C	–	BD
D	D	–

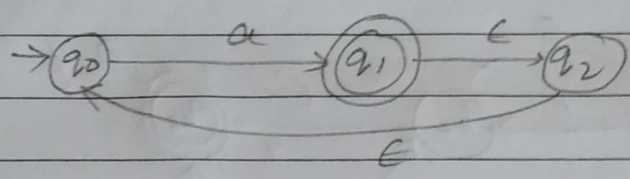
DFA Table.

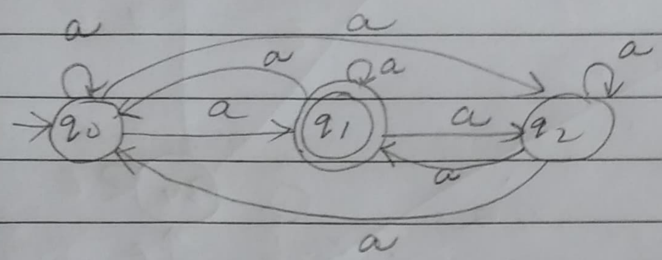
	0	1
→ ABD	ABCD	q_{DS}
ABCD	ABCD	BD
BD	CD	q_{DS}
CD	D	BD
D	D	q_{DS}
q_{DS}	q_{DS}	q_{DS}

ε closure of $A = \{ABD\}$



Q. Construct minimal DFA for the following ε NFA.





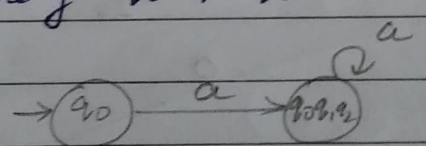
NFA Table

	a	
→ q_0	$q_1 q_2 q_0$	
q_1	$q_1 q_2 q_0$	
q_2	$q_1 q_2 q_0$	

DFA table

	a	
→ q_0	$q_1 q_2 q_0$	
$q_1 q_2 q_0$	$q_1 q_2 q_0$	

ε closure of $q_0 \Rightarrow q_0$

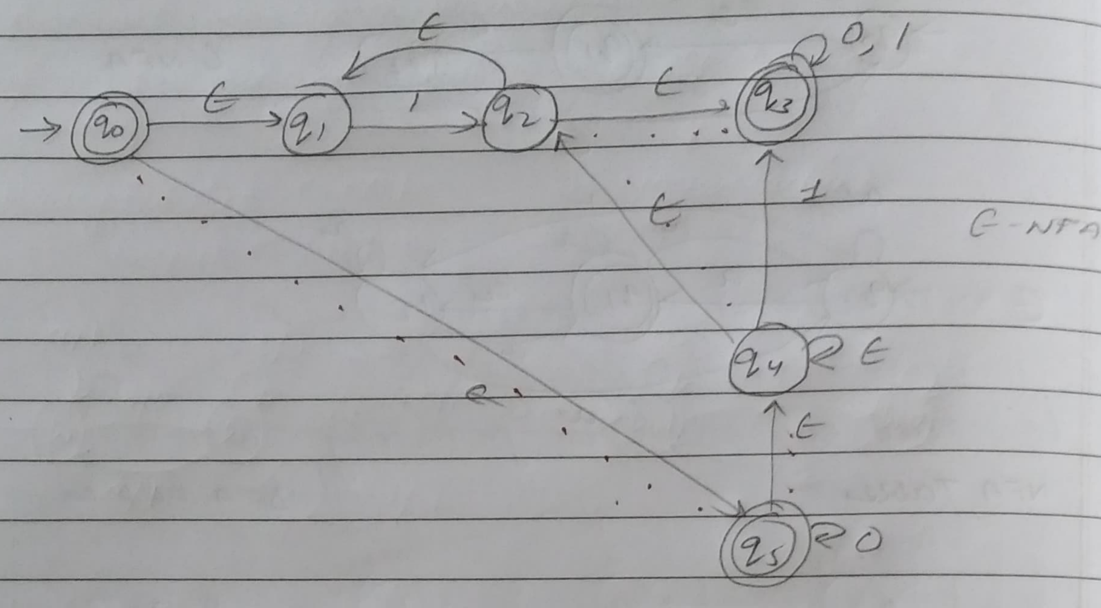


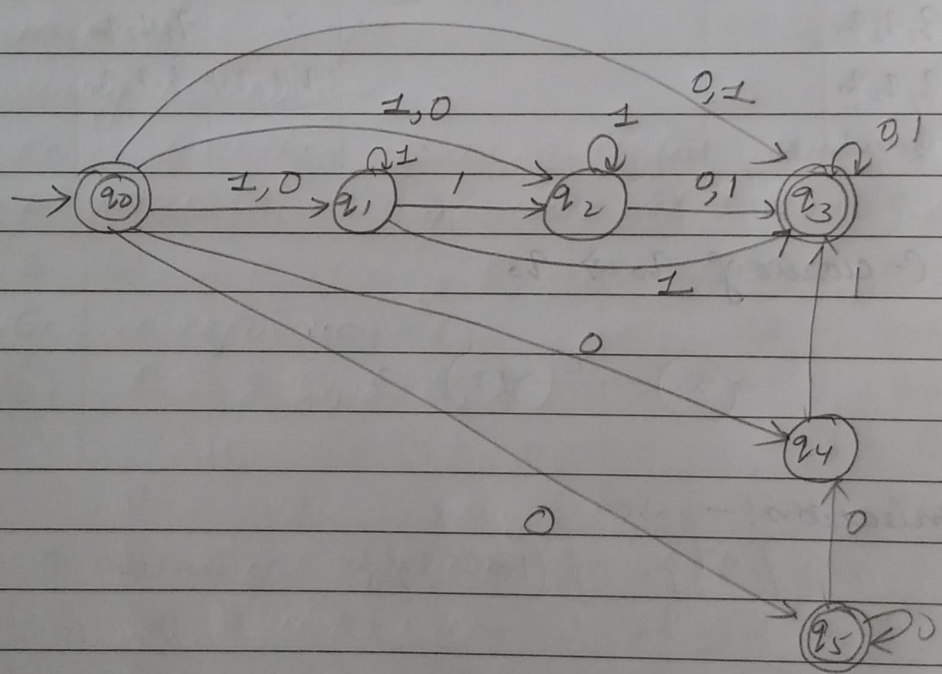
minimisation:-

$\{q_0\}$ $\{(q_0\, q_1\, q_2)\}$

minimal ही है।

Q. How many states required to construct an equivalent minimized DFA for the following ε-NFA

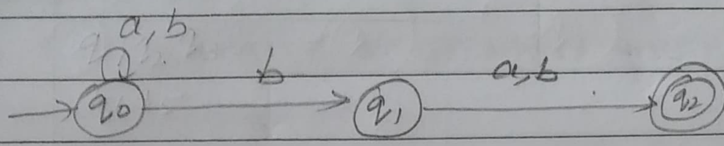




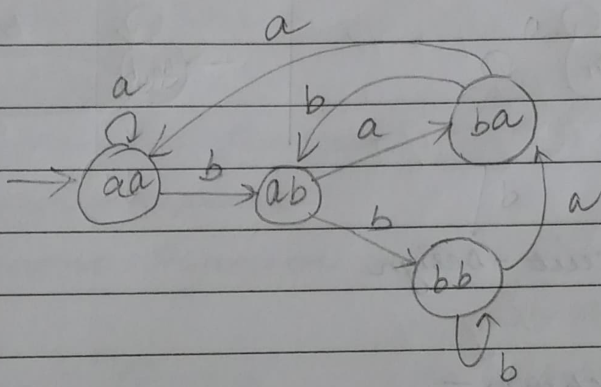
No need to solve ; accepts $(0+1)^*$

$\therefore$ states = 1

Q. How many states in minimal DFA equal to following DFA

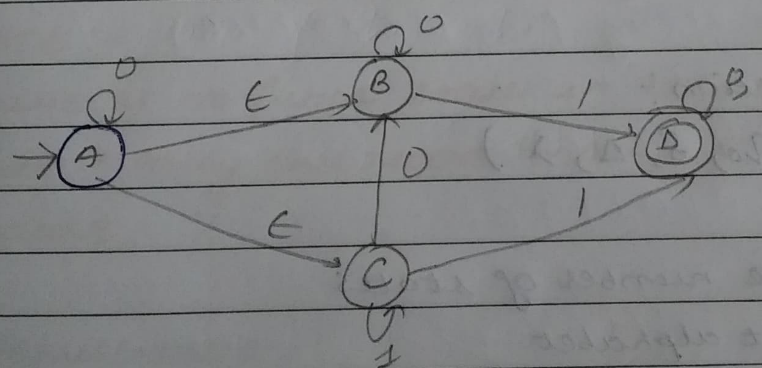


NFA table



Language accepted ⇒ second symbol from RHS = 'b'.
Hence 4 states

Q.



$\delta(A, 01) = CD$

(A) {D}
✓(B) {C, D}
(C) {B, C, D}
(D) {A, B, C, D}

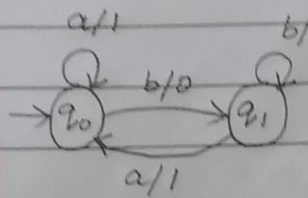
14-06-21

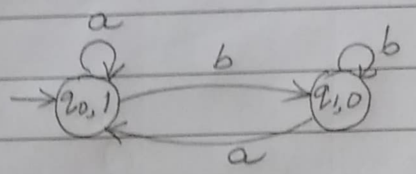
i/p $\{a, b\}$ → FINITE AUTOMATA WITH OUTPUT → Output $\Delta = \{0, 1\}$

(Transducers)

output generators

Mealey Machine | Moore Machine





Used in circuit-Design

Mealey Machine –

It is a mathematical model in which output is associated with transition.

Formal definition :-

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

Q : Finite numbers of states

Σ : Input alphabet

q_0 : Initial state

Δ : Output alphabet

δ : Transition function $\quad Q \times \Sigma \to Q$

λ : Output function

For Mealey (λ) : $Q \times \Sigma \to \Delta$

Moore Machine -

It is a mathematical model in which output is associated with state.

Formal definition :-

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

Q: Finite number of states

Σ: Input alphabet

q_0: Initial state

δ: Transition function $\quad Q \times \Sigma \to Q$

Δ: Output alphabet

λ: Output Function

For Moore: $Q \xrightarrow{} \Delta$
(λ)

NOTE :-

1. There is no final state in Mealey and Moore machines.
2. Both Mealey and Moore machines are Deterministic in nature.

Representation

- → Transition diagrams
- Transition tables.

classmate

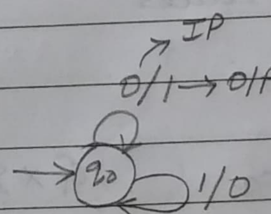
Date
Page

✱ Construction of Mealey Machines

Q. Construct Mealey machine that produces 1's Complement of given binary number as output.

Input = 0101

Output = 1010

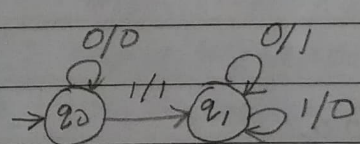


Q. GATE IIT Bombay

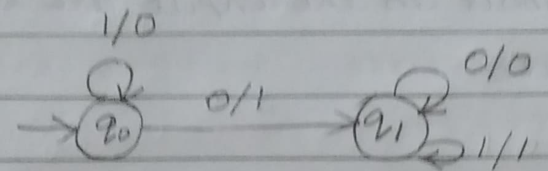
Construct Mealey machine that produces 2's Complement of the given binary number as output (assume we are reading string from LSB to MSB)

Input :- 1010100

Output 0101100



Q. What is the output produced by the following machine



LSB to MSB

Say Input ⇒ 100101 ⇒ 37

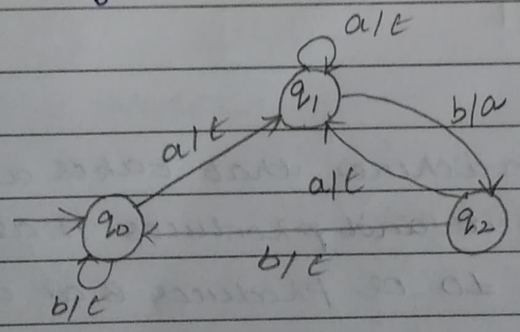
100110 ⇒ 38

100111 ⇒ 39

Output ⇒ Adding 1 to the given input

Binary Incrementor

Q. Consider the following finite state transducer where the label on the edge x/t denotes if the input is x, follow the arrow and emit t.



For the input aabbbaaabbbbabaabb the output is

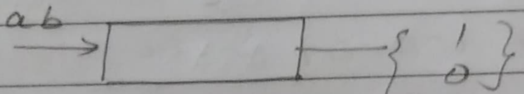
✓(a) aaaa   aaaa

(b) aaaaaaaaa

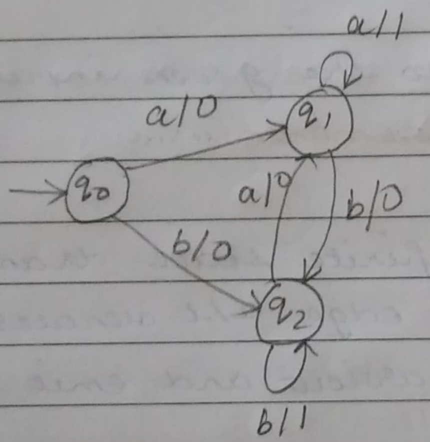
(c) abababab

(d) abbbabbbbababb

Q Construct Mealey machine that takes all strings of a's and b's as input and produces 1 as output if last two symbols in the input are same otherwise produces 0 as output



$\Sigma = \{a, b\}$ $\Delta = \{0, 1\}$

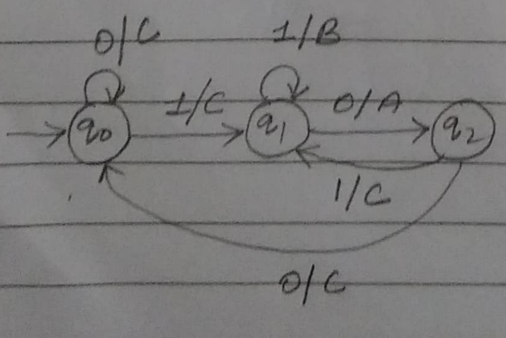


Q. Construct Mealey machine that takes all strings of 0's and 1's as input and produces A as output if input ending with 10 or produces B as output if input ending with 11 otherwise produces output C.

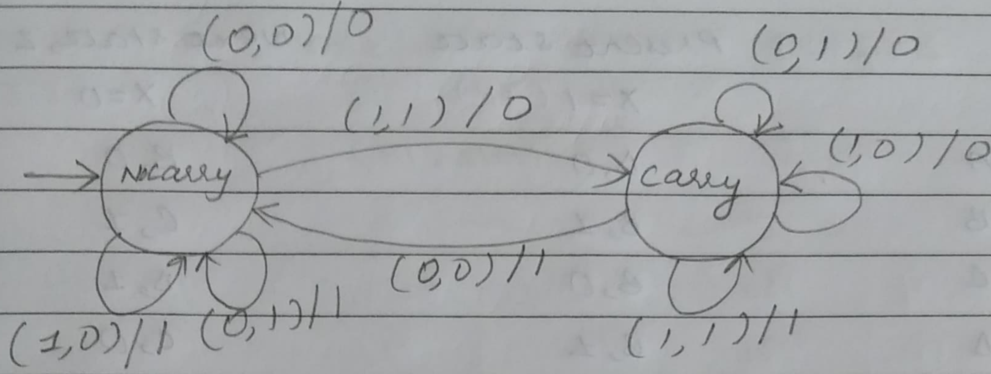
10 – A

11 – B

– C



Q. What type of output does the following Mealey machine produce?



LSB to MSB

Say Input → 0+0=0 output

1+1= 10

↑ carry

1+1 = 11
+
carry ↑ carry

Binary Adder

$$\begin{array}{r} {}^{1}\,{}^{1}\;\;{}^{1} \\ 1011 \\ 1101 \\ \hline \text{carry } 1 \rightarrow 1100 \end{array}$$

Q. A finite state machine with the following state table has a single input x and a single output Z.

	Present state	Next state, Z
	X=1 (IP)	X=0 (IP)
A	D, 0 (State, OP)	B, 0
B	B, 1	C, 1
C	B, 0	D, 1
D	B, 1	C, 0

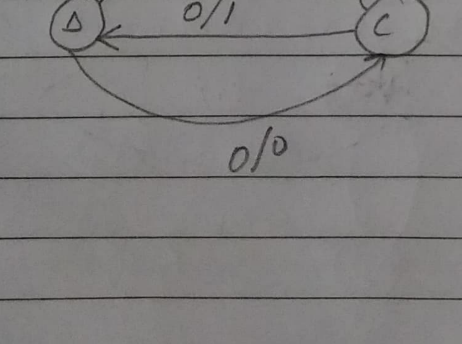
If the initial state is unknown, then the shortest input sequence to reach the final state C?

a) 01

✓ b) 10

c) 101

d) 110



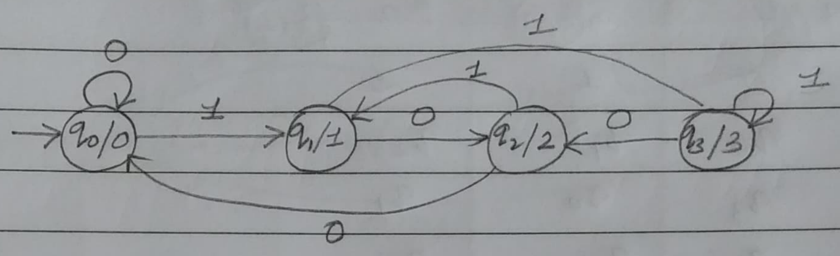
Mealey machine

16-06-21

Q. Construct Moore machine that takes all binary strings as input and produces Residue modulo 4 as output.

$\underbrace{\text{Residue}}_{\text{Remainder}}$

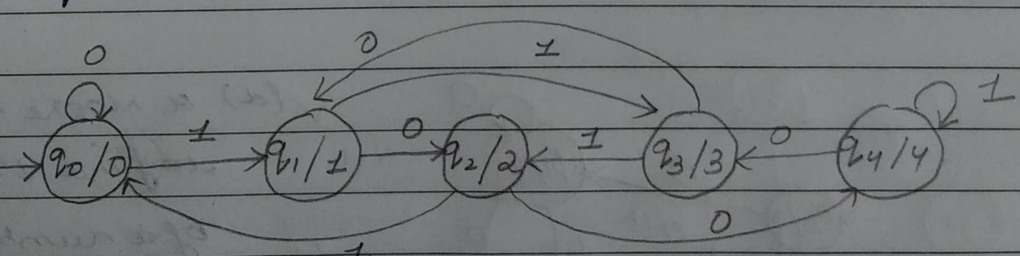
$\Sigma (0,1)$ Remainders after / 4 $\Rightarrow$ $\Delta \{0, 1, 2, 3\}$



Transition table for the above Moore Machine

	0	1
→ [q_0, 0]	q_0	q_1
[q_1, 1]	q_2	q_3
[q_2, 2]	q_0	q_1
[q_3, 3]	q_2	q_3

Q. Construct Moore machine that takes all binary strings as input and produces Residue modulo 5 as output.



	0	1
[q_0, 0]	q_0	q_1
[q_1, 1]	q_2	q_3
[q_2, 2]	q_4	q_0
[q_3, 3]	q_1	q_2
[q_4, 4]	q_3	q_4

classmate

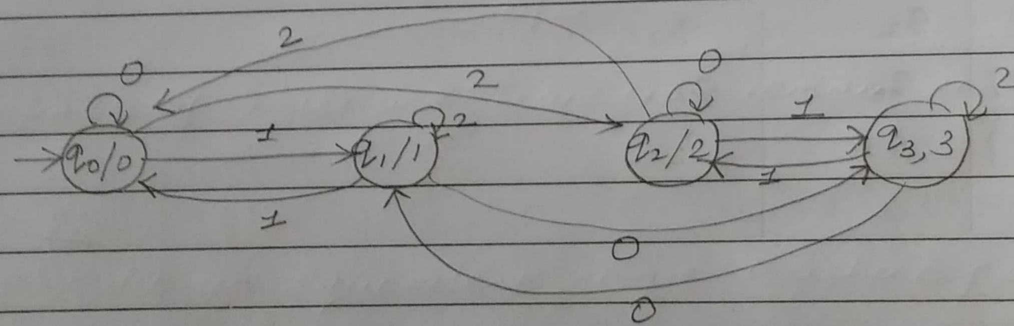
Date
Page

Q. Construct Moore machine that takes all base 3 numbers as input and produces Residue modulo 4 as output.

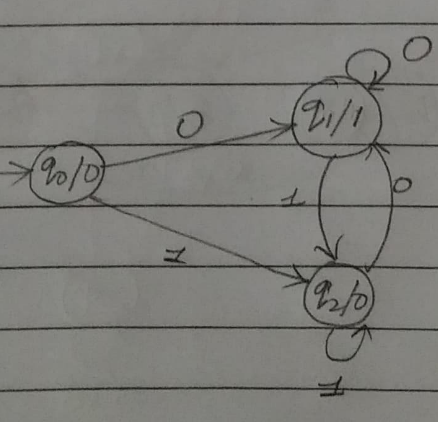
$\Delta = \{0, 1, 2, 3\}$

$\Sigma = (0, 1, 2)$

	0	1	2
$\rightarrow [q_0, 0]$	q_0	q_1	q_2
$[q_1, 1]$	q_3	q_0	q_1
$[q_2, 2]$	q_2	q_3	q_0
$[q_3, 3]$	q_1	q_2	q_3



Q. Below machine is

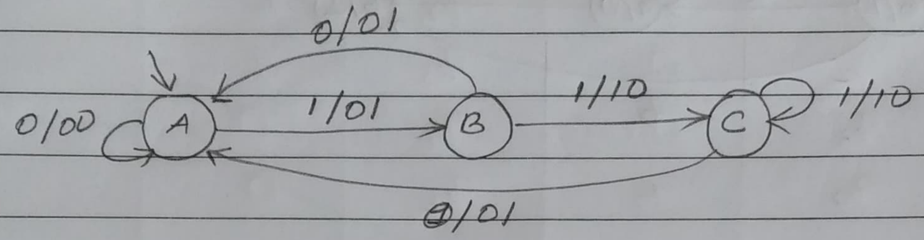


✓(d) a Moore machine to find 1's complement of a number

(a) a mealey machine to find 2's complement of a number
(b) a Moore machine to find 2's complement of a no.
(c) a mealey machine to find 1's complement of a no.

GATE 2002

The finite state machine described by the following state diagram with A as starting state, where an arc label is x/y and x stands for 1-bit input and y stands for 2-bit output.



(a) Outputs the sum of the present and the previous bits of the input ✓

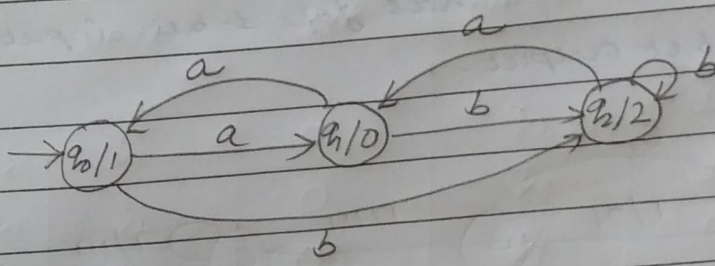
(b) Outputs a "01" whenever the input sequence contains "11".

(c) Outputs a "00" whenever the input sequence contains "10"

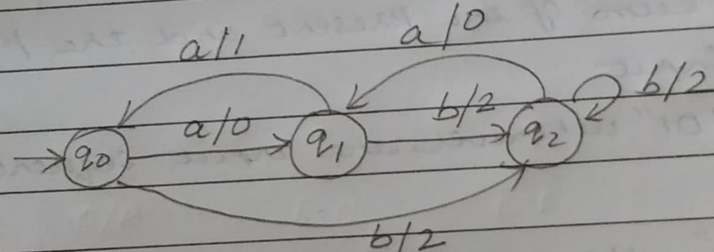
(d) None of the above.

Q. Construct an equivalent Mealey machine for the following Moore machine.

Moore to Mealey



Moore machine

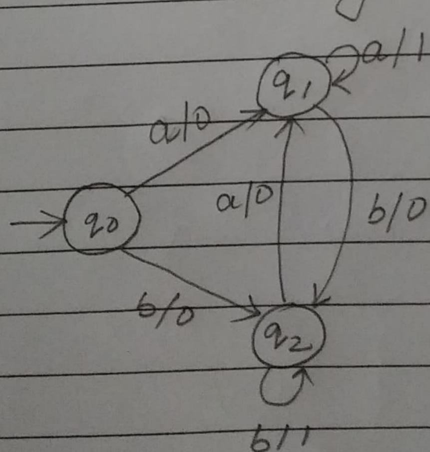


Mealey machine

Number of states and transitions remain the same

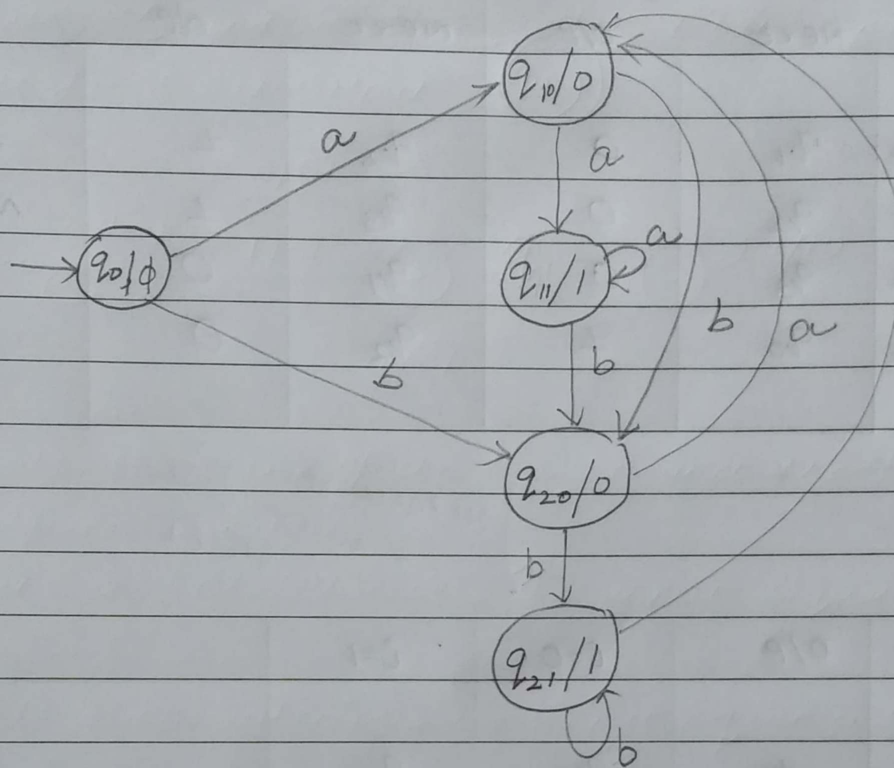
Q. Construct an equivalent Moore machine for the following Mealey machine

Mealey to Moore



Mealey machine

Number of states may increase.



Moore machine

NOTE:

1. While converting from Mealey to Moore, number of states may increase
2. If initial state is split into multiple states, any one of the split states can be taken as initial state in the resultant Moore machine.

Conversion of Mealey machine to Moore machine and vice-versa in form of tables.

	I=0 Next	O/P	I=1 Next	O/P
$\to q_0$	q_1	0	q_2	1
q_1	q_2	0	q_3	1
q_2	q_3	0	q_1	0
q_3	q_0	1	q_2	0

Mealey Machine

इस table में q_2 पर 2 entries है।

इस table से transition देखकर बना है।

State	O/P	i=0	i=1
$\to q_0$	1	q_1	q_{21}
q_1	0	q_{20}	q_{31}
q_{20}	0	q_{30}	q_1
q_{21}	1	q_{30}	q_1
q_{30}	0	q_0	q_{20}
q_{31}	1	q_0	q_{20}

Moore Machine

NOTE:-

Mealey Machine $\Rightarrow$ Maximum states in Moore

{ n states, m outputs Symbols } $\Rightarrow$ $\boxed{n \times m}$

NOTE -

For the given n length input sequence output length produced by Moore is n+1

For the given n length input sequence output length produced by Mealey is n

State	O/P	i=0	i=1
→ A	b	B	A
B	b	B	C
C	a	B	A

Moore Machine

	I=0		I=1	
	Next	O/P	Next	O/P
→ A	B	b	A	b
B	B	b	C	a
C	B	b	A	b

Mealey Machine

O/P associated with B in Moore table