$Q = \{q_0, q_f\}$

$\Sigma = \{a\}$

$q_0 = q_0$

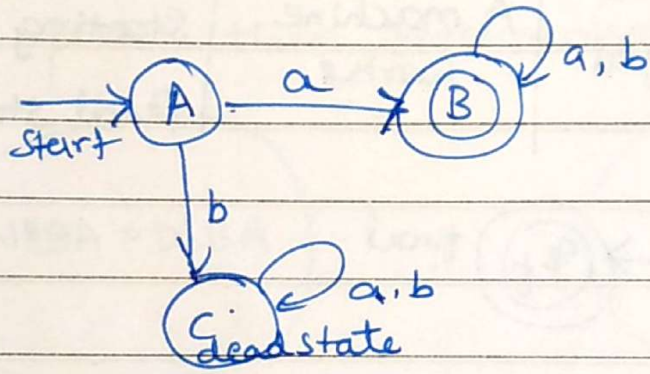
$F = q_f$.

*) Construct a DFA which accepts set of all strings over {a, b} where each string starts with an 'a'.

Solⁿ –

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

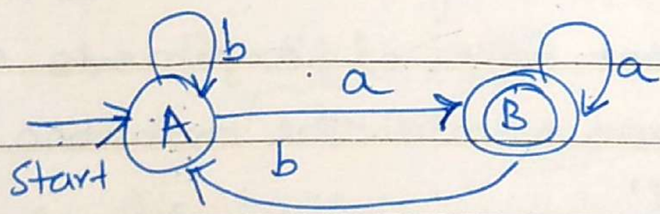
$L = \{a, aa, ab, aaa, ---\}$



*) Construct DFA which accepts set of all strings over {a, b} where each string ends with an 'a'

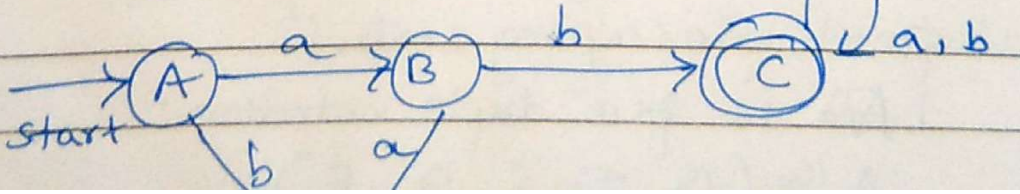
Solⁿ $L = \{a, aa, ba, baa, aaa, aba, ---\}$

Infinite.



*) String starts with ab

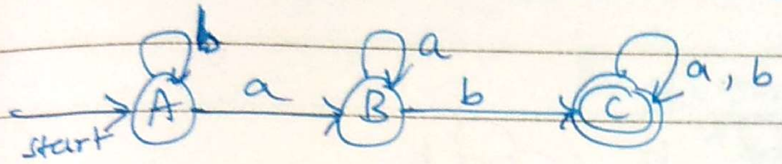
$L = \{ab, aba, abb, ----\}$



Date / /20

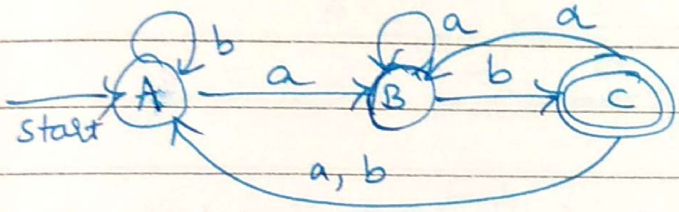
*) "ab" as a Substring -

$L = \{ab, aaba, aabb, \ldots\}$



*) end with 'ab'.

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

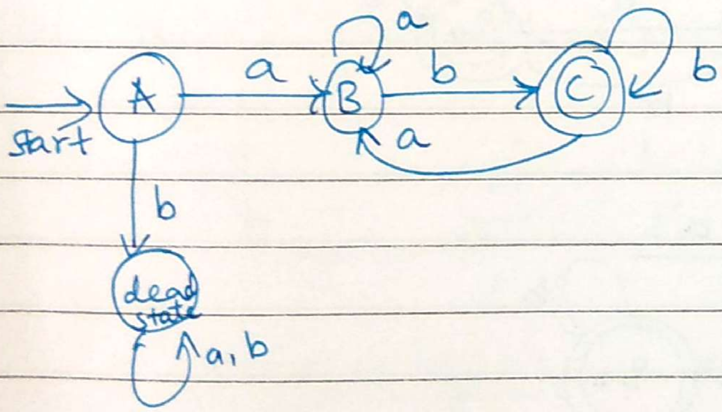


*) Starts and ends with different symbol (a, b)

a - - - b

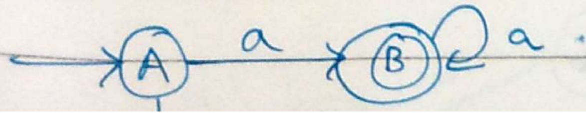
b - - - a

$\{ab, ba, aab, baa, abb, \ldots\}$



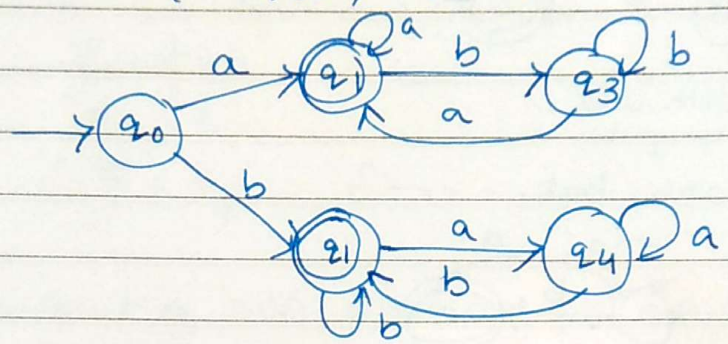
*) Starts and ends with same symbol (a a)

$L = \{\epsilon, a, aa, bb, \ldots\}$



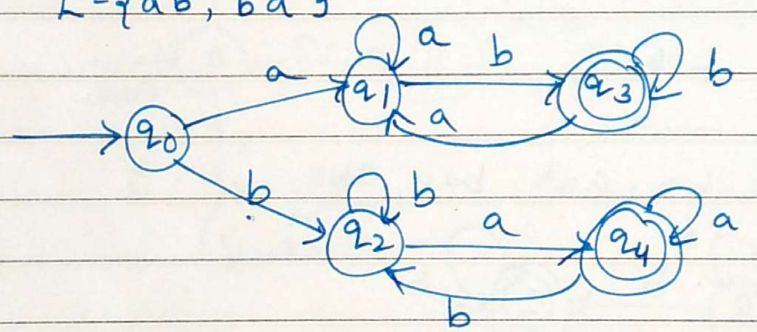
Design a DFA over $\Sigma = (a, b)$ such that every String accepted must start and end with same symbol.

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

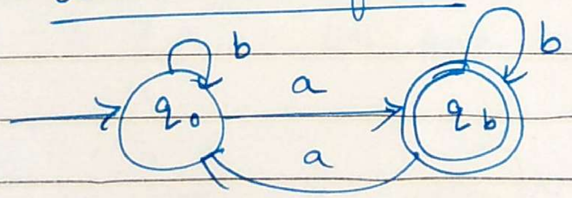


Design a DFA over $\Sigma = (a, b)$ such that every String accepted must start and end with different symbol.

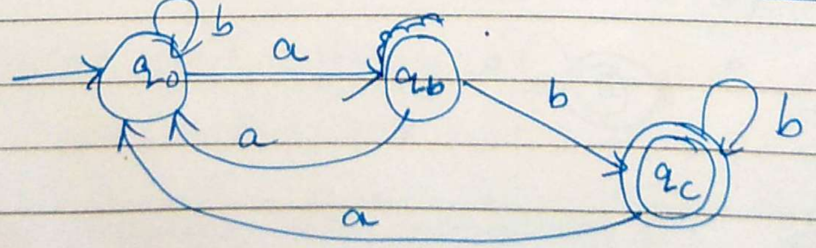
$L = \{ab, ba\}$



odd number of a's

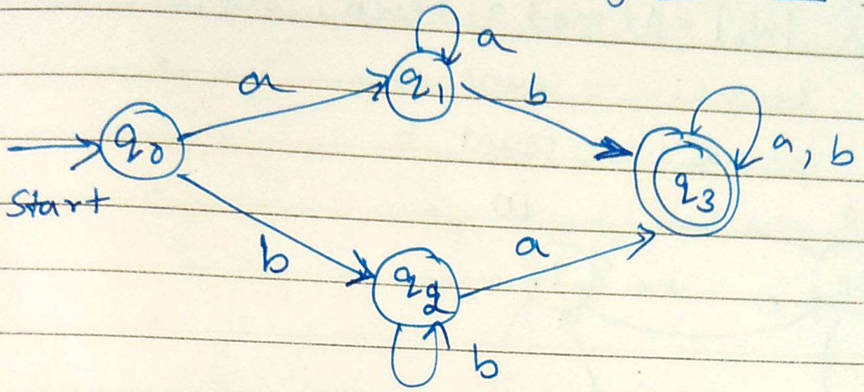


{odd number of a's and end with ab}

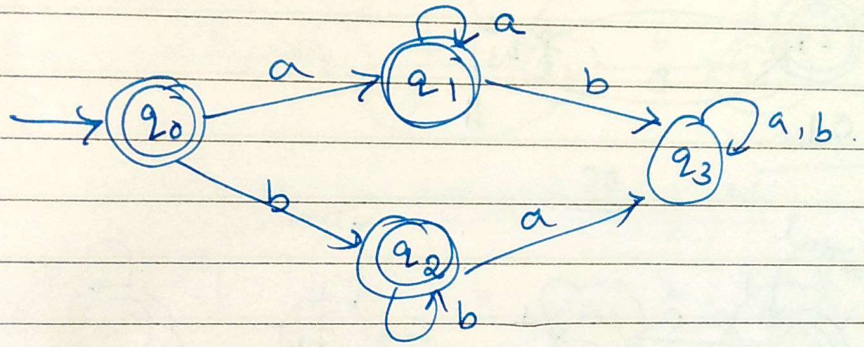


Date / /20

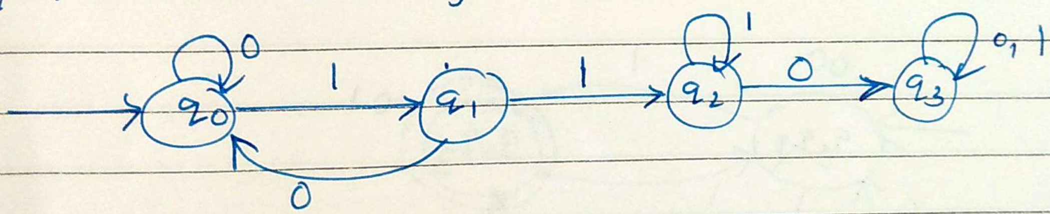
{w/w contains either the substring ab or ba}



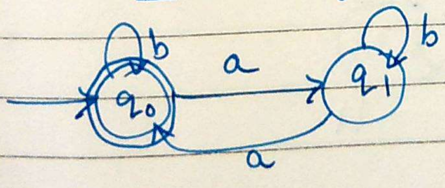
{w/w contains neither, the substring ab or ba}



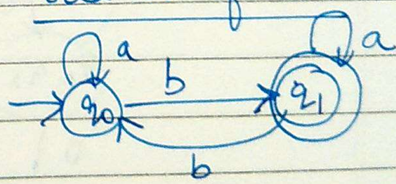
{w/w contain substring 110}



even no. of a's



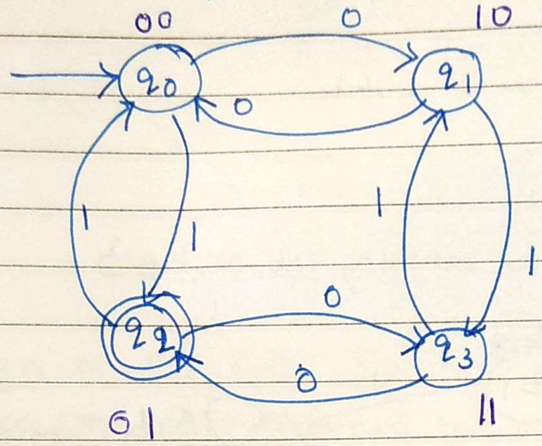
odd no. of b's



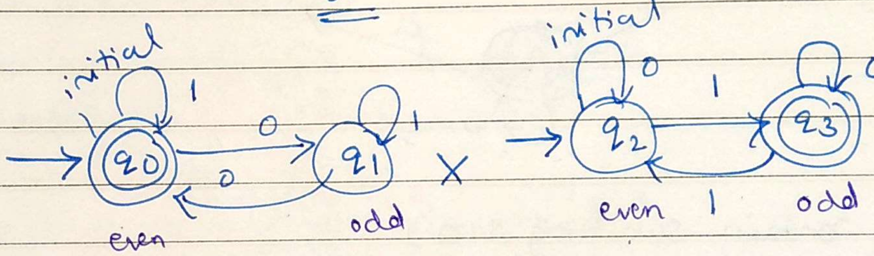
Design a dfa over an alphabet $\Sigma = \{0, 1\}$ such that it accepts all the strings $|w_0| = 0 \pmod 2$ (even) & & $|w_1| = 1 \pmod 2$ (odd)

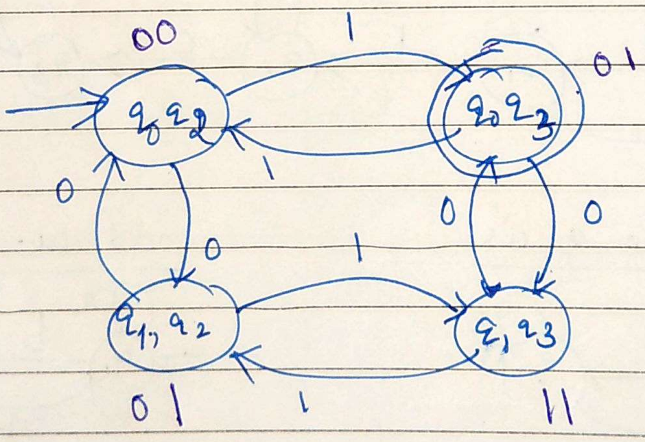
x-axis — count a.

y-axis — count b



or





q_0 on 0 = q_1 $\qquad$ q_0 on 1 = q_0

q_2 on 0 = q_2 $\qquad$ q_2 on 1 = q_3

q_1 on 0 = q_0 $\qquad$ q_1 on 1 = q_1

q_2 on 0 = q_2 $\qquad$ q_2 on 1 = q_3

Scanned by CamScanner

In Case of even no. 0's ~~and~~ or odd no. of 1's

q_0, q_2 — may become final state

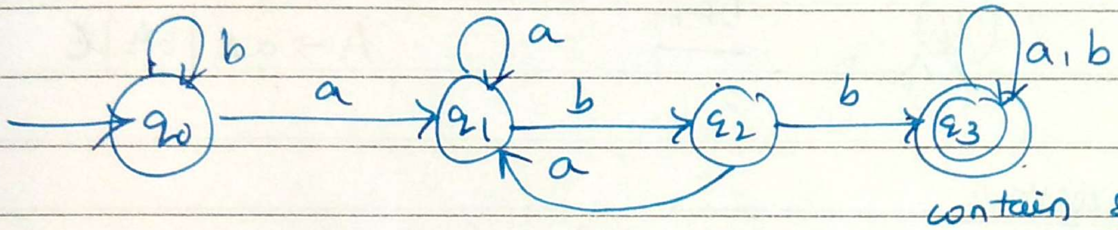
q_0, q_3 - may become final state

q_0, q_2 - may become final state

q_1, q_3 - not a final state

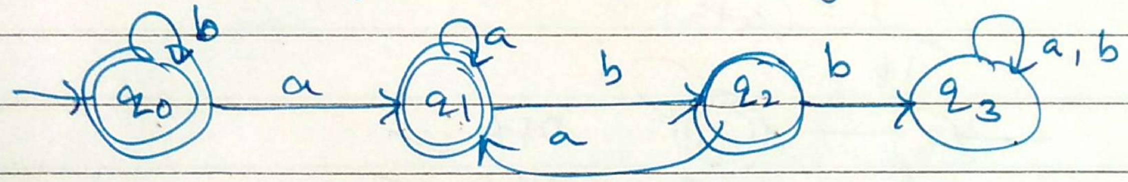
q_0, q_3 are final states.

String does not contain substring 'abb'



contain substring 'abb'

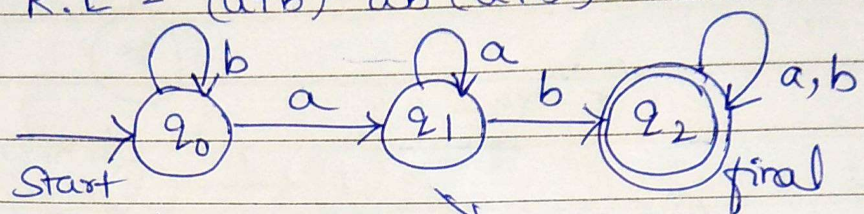
for not contain substring 'abb'



Containing substring 'ab'

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

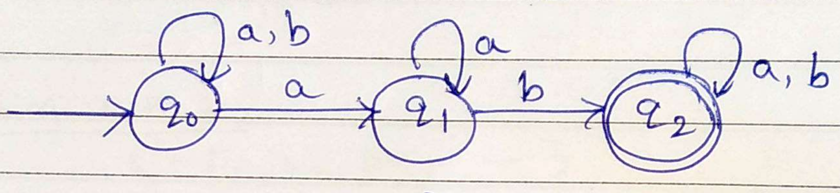
$R.E = (a+b)^* ab (a+b)^*$



$R.E = b^* a a^* b (a+b)^*$

→ DFA ✓

NFA ✓



→ NFA but not DFA

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

One language may contain more than one Regular expression.

All three Regular expressions are equivalent.

Grammar

$S \rightarrow AabA$

$A \rightarrow aA \mid bA \mid \epsilon$

$S \rightarrow AS_1$

$S_1 \rightarrow abA$

$A \rightarrow aA \mid bA \mid \epsilon$

$S \rightarrow bS \mid aA$

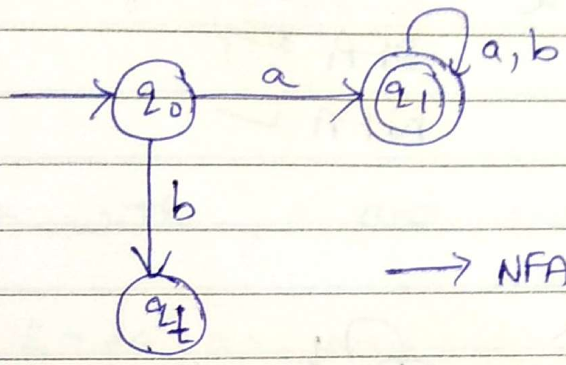
$A \rightarrow aA \mid bB$

$B \rightarrow aB \mid bB \mid \epsilon$

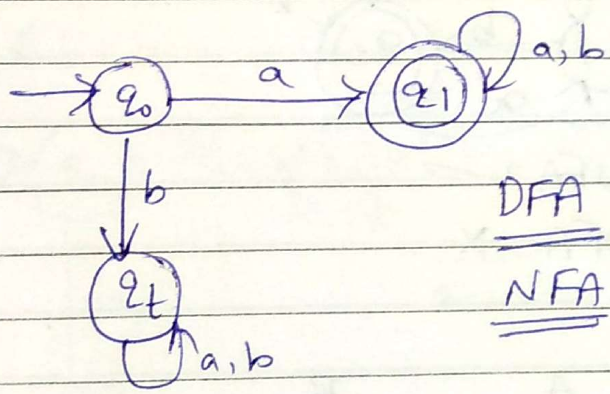
$R.E = L(M_1) = a(a+b)^*$

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

NFA not DFA



→ NFA but not DFA



DFA ✓

NFA ✓

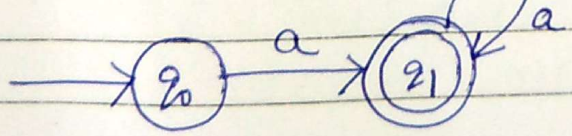
$L(M_1) = a(a+b)^*$

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

3) Ending with 'a'

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

$R.E = (a+b)^* a$



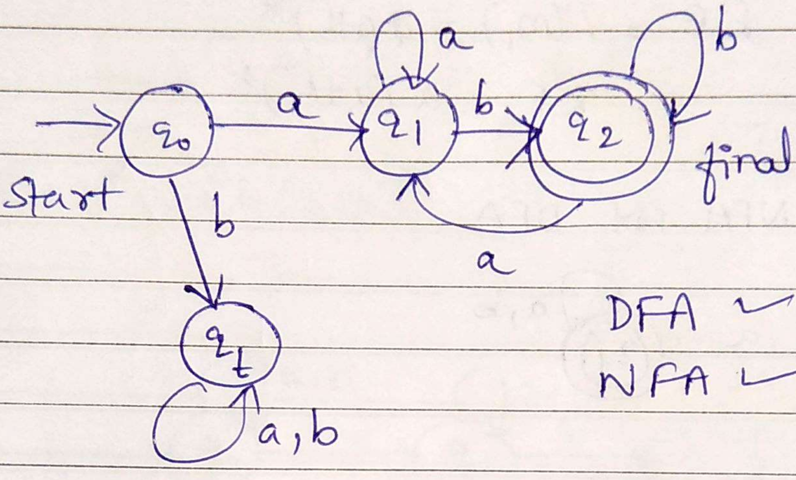
DFA ✓

NFA ✓

Start with 'a' and end with 'b'

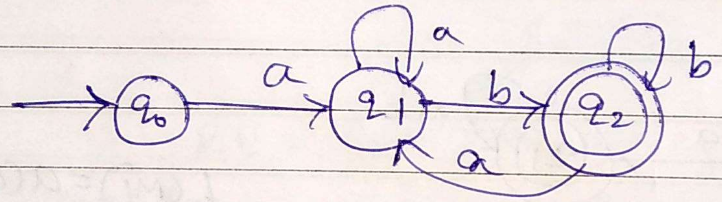
$L = \{ab, aab, abb, aabb, \ldots\}$

$R.E = a(a+b)^* b$



DFA ✓

NFA ✓



NFA ✓

DFA ✗

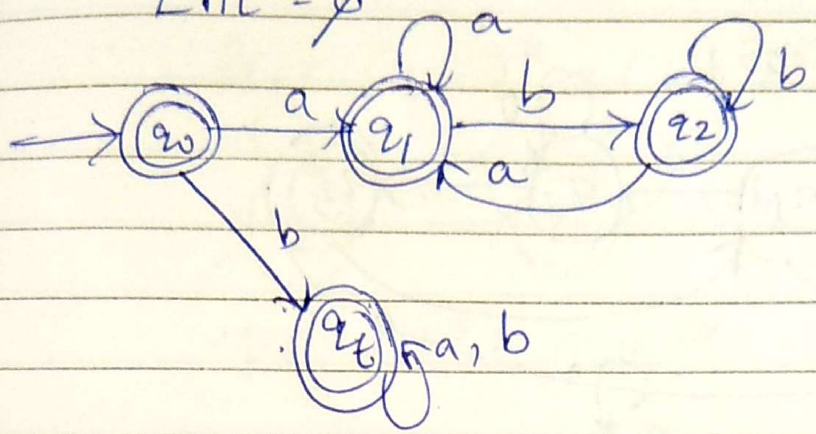
S	A	B
	aa^*	bb^*

$S \rightarrow aA$

$A \rightarrow aA \mid bB$

$B \rightarrow bB \mid aA \mid \epsilon$

$L \cap L^c = \emptyset$

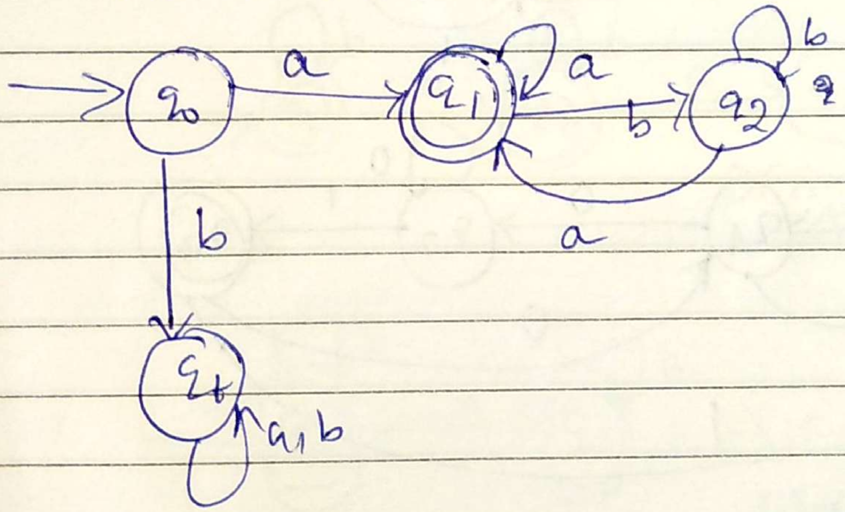


] start with a and end with a

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

$R.E = a + a(a+b)^* a$

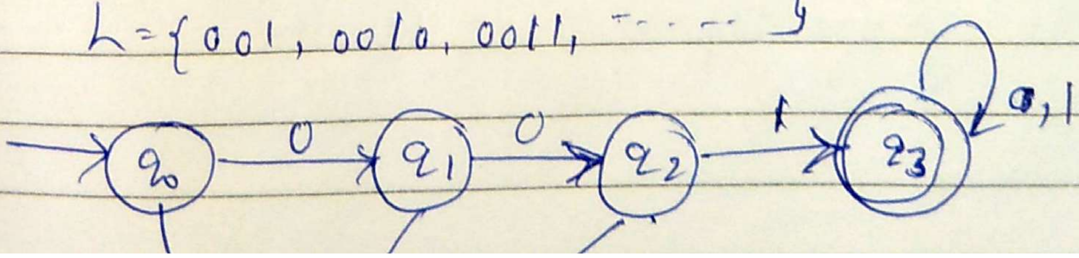
abba



] starting with '001', $\Sigma = \{0, 1\}$

$R.E \; 001(0+1)^*$

$L = \{001, 0010, 0011, \ldots\}$

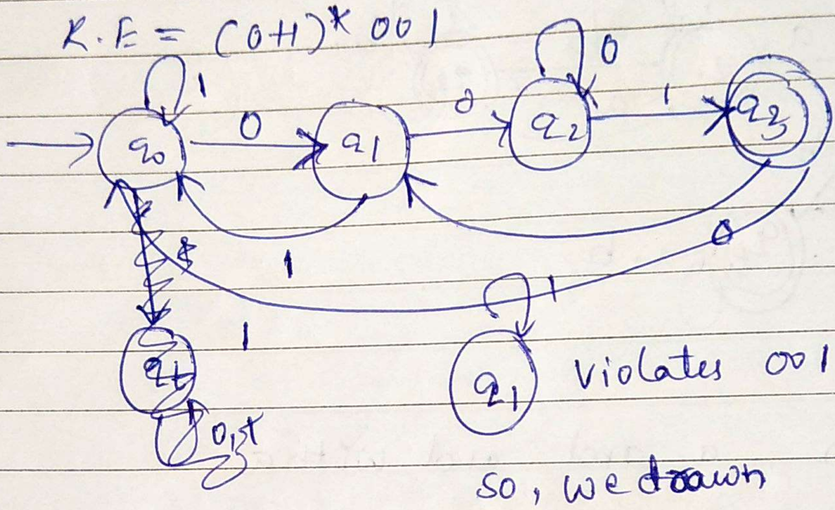


Scanned by CamScanner

ending with 001 , $\Sigma = \{0,1\}$

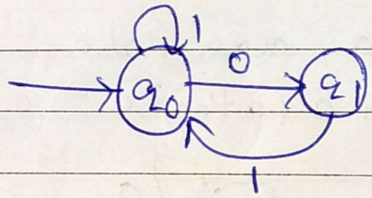
$L = \{001, 0001, 1001, \ldots\}$

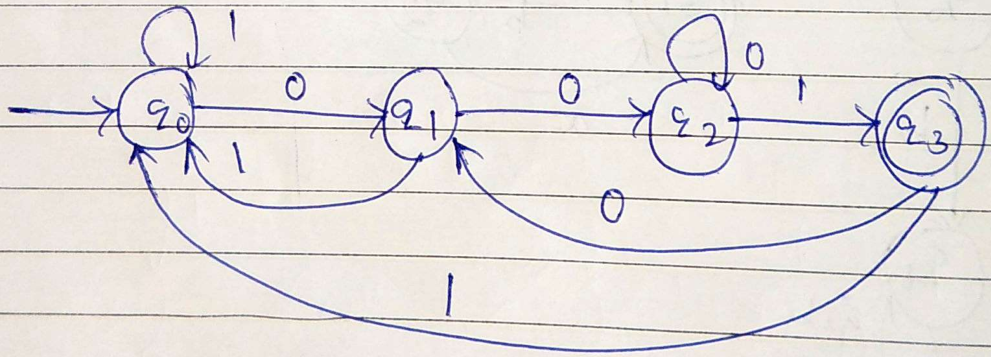
$R.E = (0+1)^* 001$

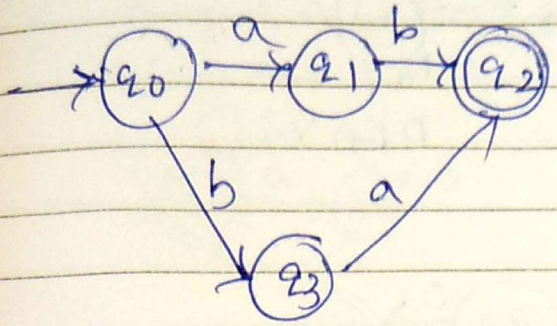


violates 001

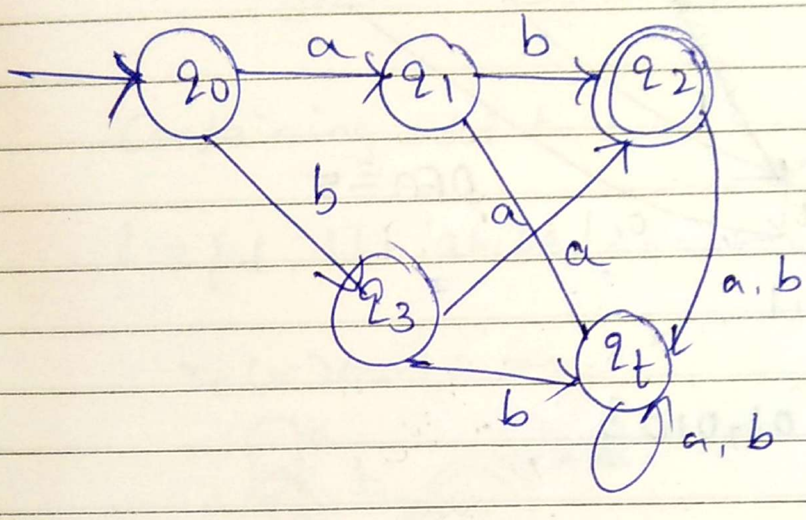
so, we drawn q_1 to q_0 on 1.







NFA ✓
DFA ✗



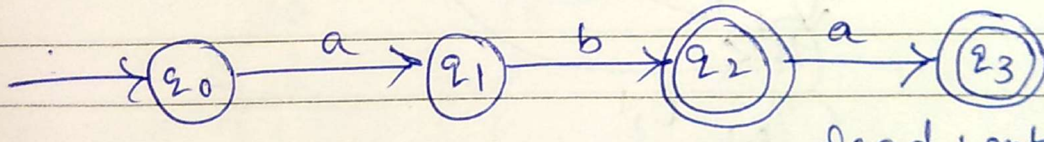
q_t - dead state

minimum no. of states in DFA = 5

minimum no. of states in NFA = 4

Minimum no. of states in FA = min (DFA, NFA) = 4.

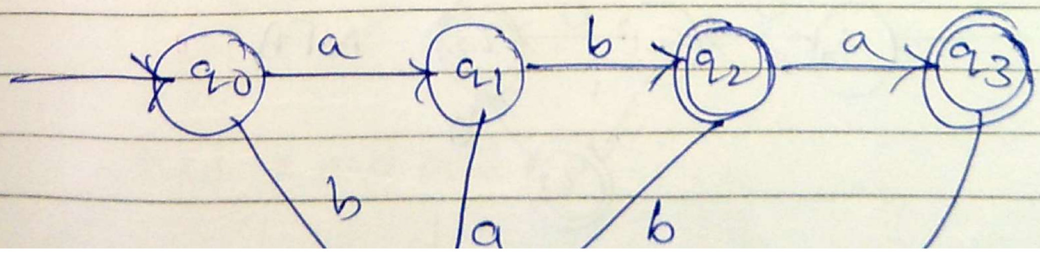
$L = \{ab, aba\}$



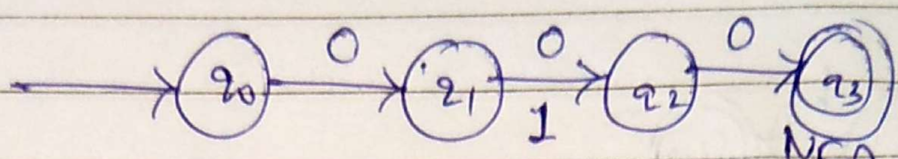
NFA ✓

dead configuration

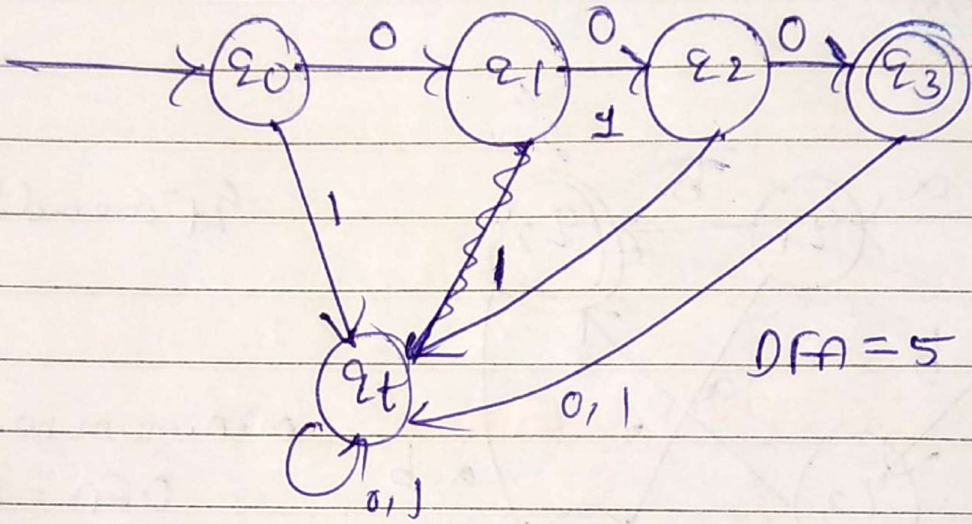
NF = 4 (states)



$L = \{000, 010\}$

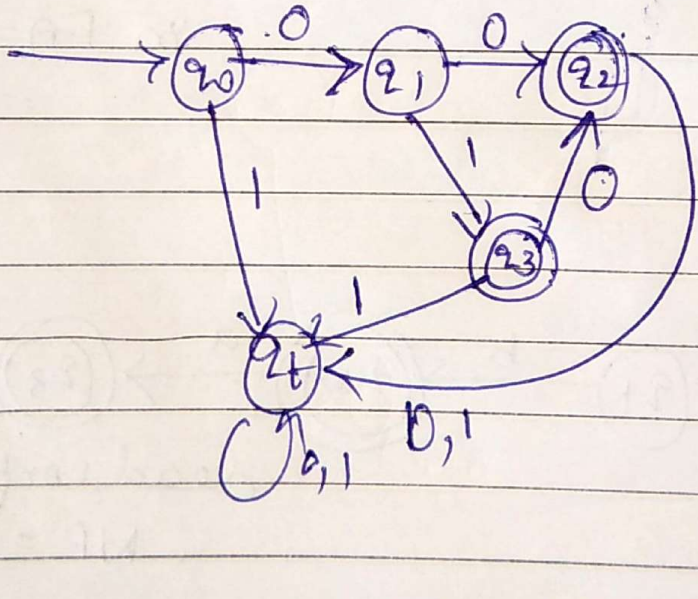


NFA ✓
DFA X

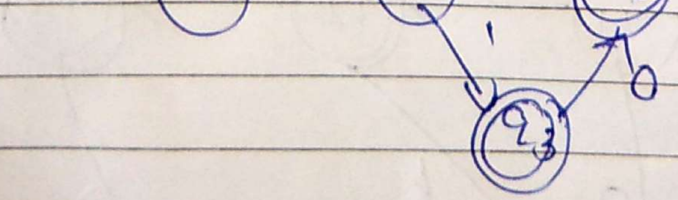


DFA = 5

$L = \{00, 01, 010\}$



DFA



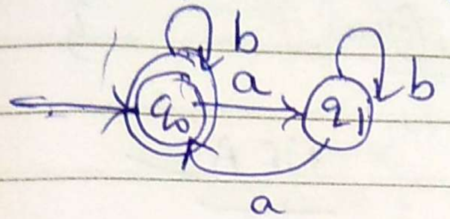
NFA

Date / /20

containing even a.

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

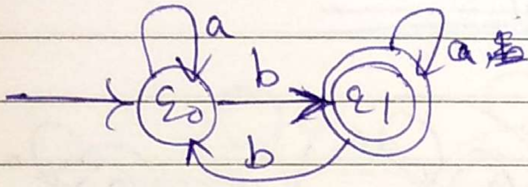
$n_a(w) = \text{even}$ $\quad n_a(w) \bmod 2 = 0$



Containing odd b

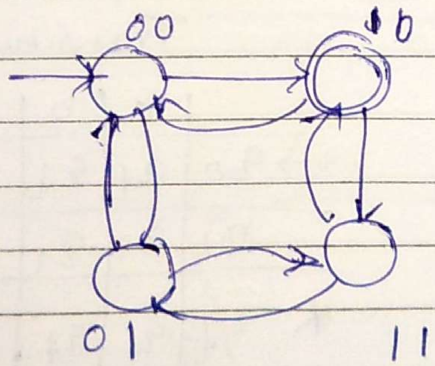
$L = \{b, bbb, ab, ba, \ldots\}$

$n_b(w) \bmod 2 = \phi.$



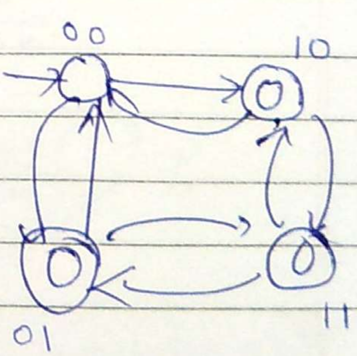
odd a and even b.

1 and 0



odd a or odd b

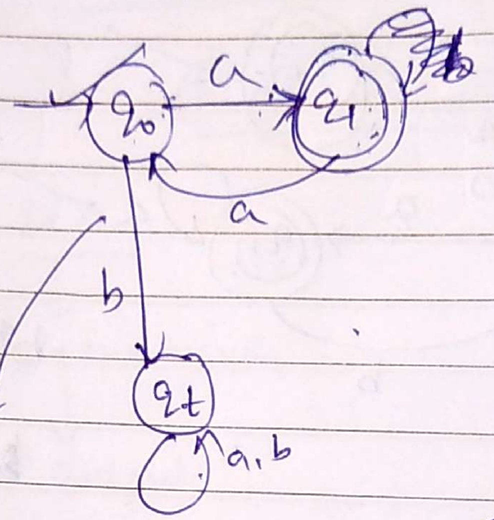
1 or 1

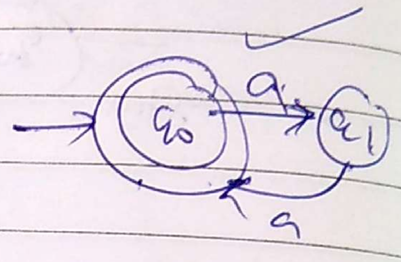


odd no. of a's $\Sigma = \{a\}$

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

DFA





NFA
↓
DFA

odd no. a's and even no. b's

or

~~aaabb~~

~~Start~~

~~end~~

~~substring~~

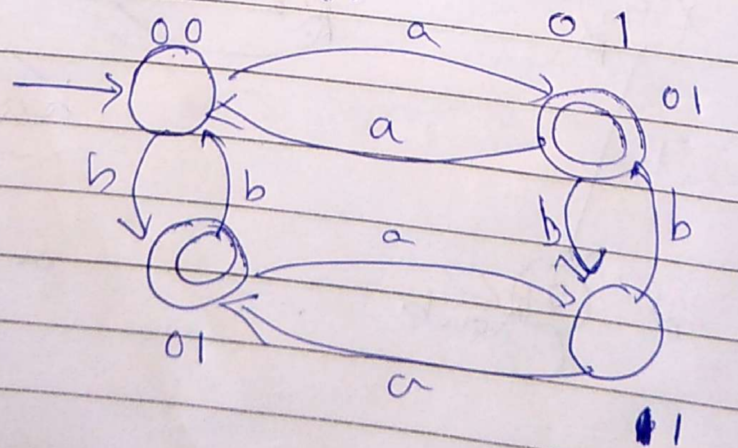
~~nor~~

~~division~~

(Odd a & even b) or (even a and odd b)

10

or

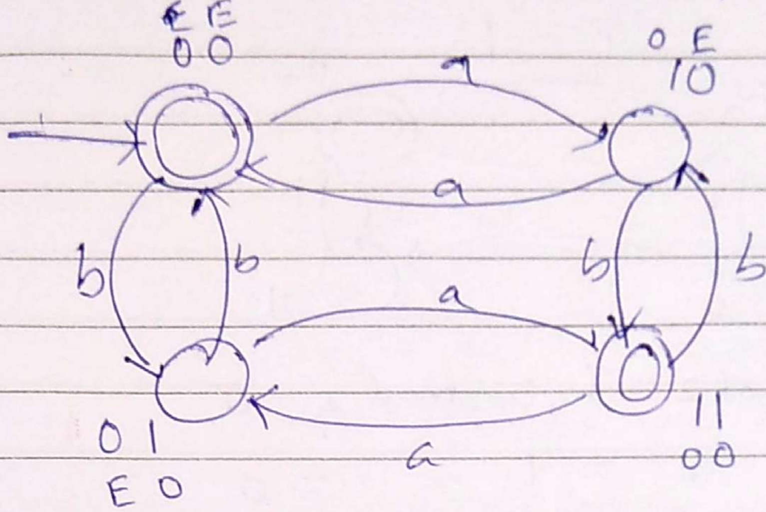


Date / /20

① ②

(odd a or even b) and (even a or odd b)

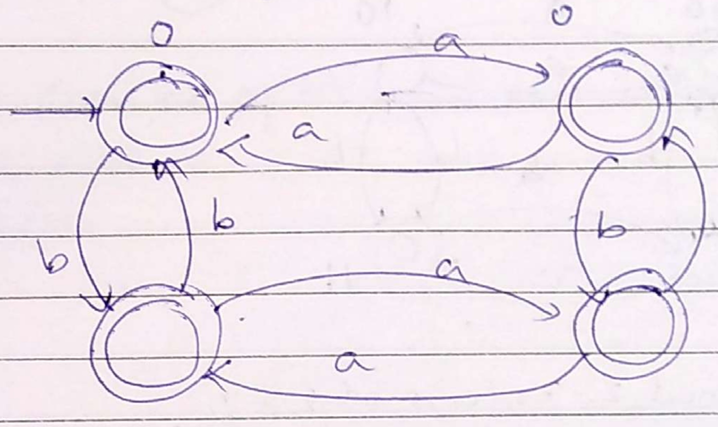
1 0 0 1



10 01 00 01

(odd a or even b) or (even a or odd b)

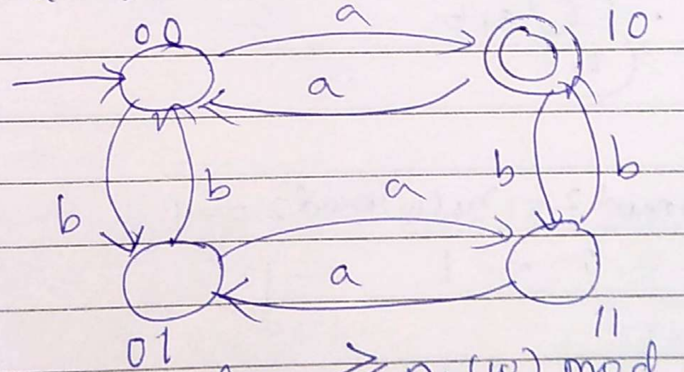
1 0 0 1



10 01

10 01 00 01

$n_a(w) \bmod 2 > n_b(w) \bmod 2$



a's > b's

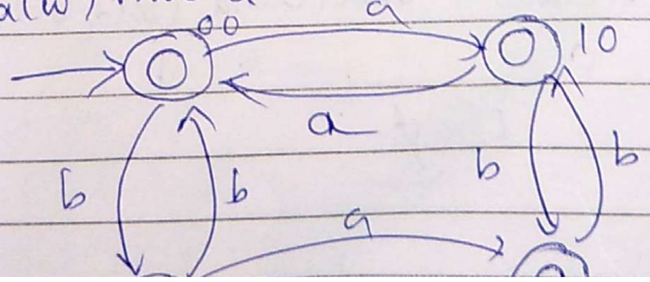
00

01

10 ✓ a's > b's

11

$n_a(w) \bmod 2 \geq n_b(w) \bmod 2$



≥

00 ✓ a's = b's

01

10 ✓ a's > b's

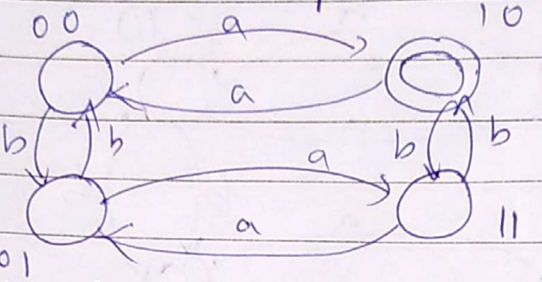
11 ✓ a's = b's

Scanned by CamScanner

$n_a(w) \bmod 2 - n_b(w) \bmod 2 = 1$

$r_a - r_b = 1$

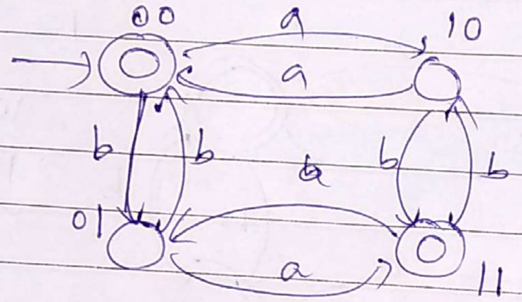
$1 - 0 = 1$



$n_a(w) \bmod 2 - n_b(w) \bmod 2 = 0$

$1 - 1 = 0$

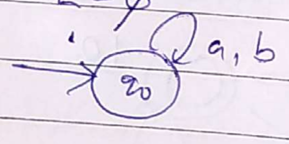
$0 - 0 = 0$



$n_a(w) \bmod 2 - n_b(w) \bmod 2 = 2$

$r_a - r_b = 2$

$L = \phi$



$n_a(w) \bmod 2 - n_b(w) \bmod 2 = -1$

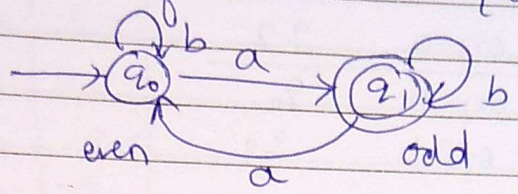
$0 - 1 = -1$

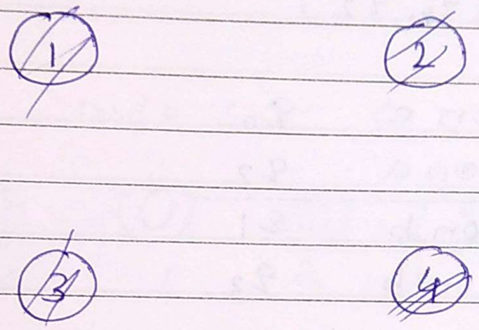
(odd a & even b) and (even a & odd b)

1 0   0 1

$L = \phi$

odd no. of a's $\Sigma = \{a, b\}$



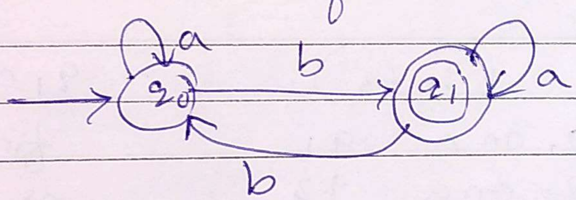


remainder, remainder

state 1 : even no. of a's and even no. of b's (0 0)
state 2 : odd no. of a's and even no. of b's (1 0)
state 3 : even no. of a's and odd numbers of b's. (0 1)
state 4 : odd no. of a's and odd no. of b's. (1 1)

Odd no. of b's $\Sigma = \{a, b\}$

a ✗
ab ✓



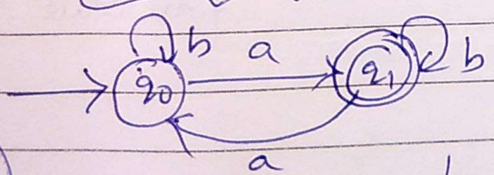
odd no. of a's and an odd no. of b's

$\frac{0}{3} = 0$

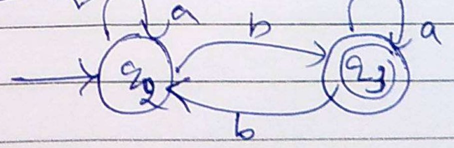
$\frac{1}{3} = 1$

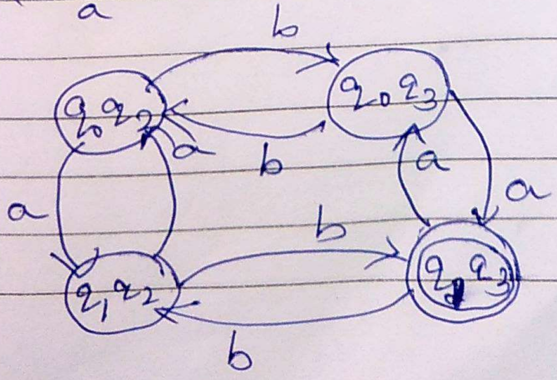
$\frac{2}{3} = 2$

$\frac{3}{3} = 0$



×





(q_0, q_2)

q_0 on a q_1

q_2 on a q_2

q_0 on b q_0

q_2 on b q_3

(q_1, q_2)

q_1 on a q_0

q_2 on a q_2

q_1 on b q_1

q_2 on b q_3

(q_1, q_3)

q_1 on a q_0

q_3 on a q_3

q_1 on b q_1

q_3 on b q_2

(q_0, q_3)

q_0 on a q_1

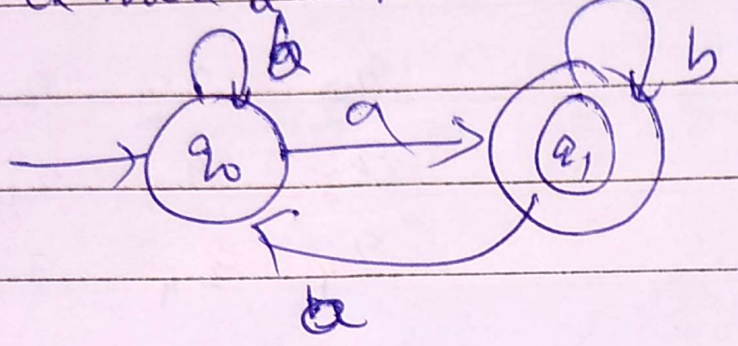
q_3 on a q_3

q_0 on b q_0

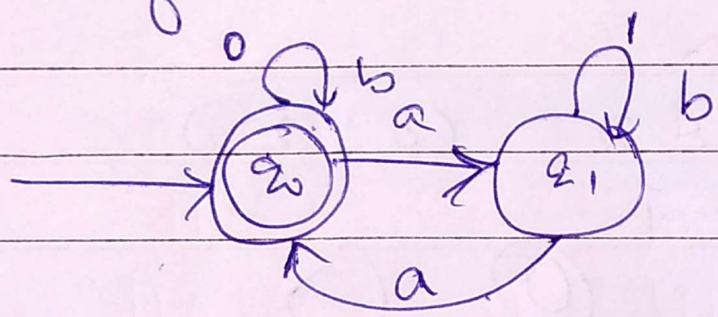
q_3 on b q_2

q_1 and q_3 are final states. so, we taken q_1q_3 state as final state.

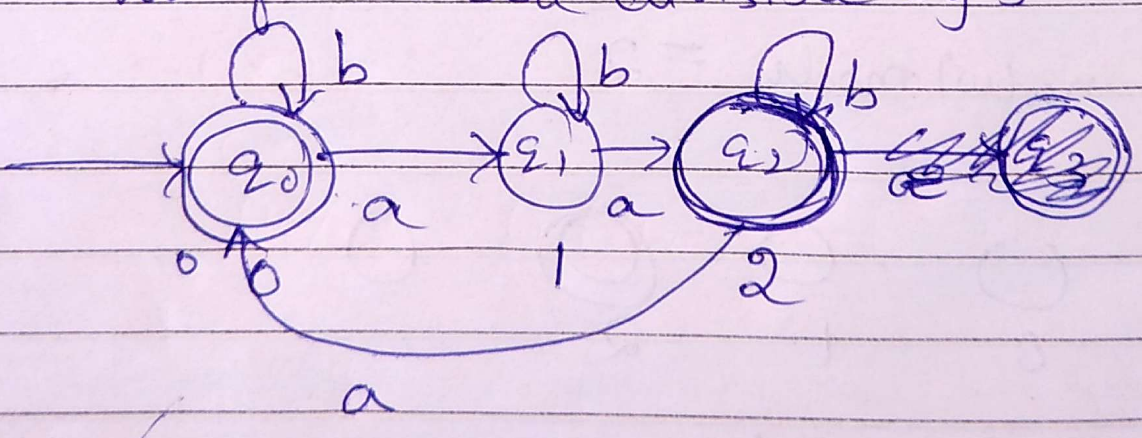
a mod 2 = 1



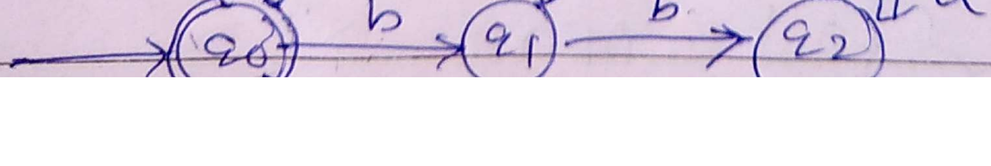
no. of a's are divisible by 2 ✓ 0 or 1



no. of a's are divisible by 3 0 1 2



no. of b's are divisible by 3 ✓



Construct DFA, which accepts set of all strings over {0,1} which when interpreted as binary number is divisible by 3.

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

remainder	remainder	remainder
0	1	2
q_0	q_1	q_2
0 (0)	1 (1)	10
11 (3)	100 (4)	101 (5)
110 (6)	111 (7)	1000
1001 (9)		

$(0)_2 = 0 \times 2^0$
$= 0 \times 1$
$= 0 \% 3 = 0 \; (q_0)$ ($\because$ 0 means q_0 state)

$(1)_2 = 1 \times 2^0$
$= 1 \times 1$
$= 1 \% 3 = 1 \; (q_1)$ ($\because$ 1 means q_1 state)

$(10)_2 = 1 \times 2^1 + 0 \times 2^0$
$= 1 \times 2 + 0 \times 1$
$= 2 + 0 = 2 \; (q_2)$ ($\because$ 2 means q_2 state)
$= 2 \% 3$

$(11)_2 = 1 \times 2^1 + 1 \times 2^0$
$= 1 \times 2 + 1 \times 1$
$= 2 + 1 = 3 \% 3 = 0 \; (q_0)$

$(100)_2 = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$
$= 4 \% 3 = 1 \; (q_1)$

$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
$= 4 + 0 + 1 = 5 \% 3 = 2 \; (q_2)$

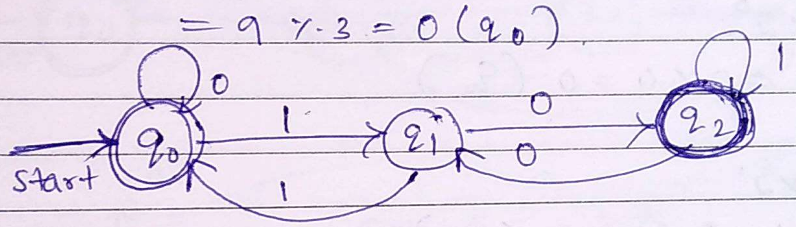
Date / /20

$(110)_2 = 1\times2^2 + 1\times2^1 + 0\times2^0$
$= 4 + 2 + 0$
$= 6 \% 3 = 0 \ (q_0)$

$(111)_2 = 1\times2^2 + 1\times2^1 + 1\times2^0$
$= 4 + 2 + 1 = 7 \% 3 = 1 (q_1)$

$(1000)_2 = 1\times2^3 + 0\times2^2 + 0\times2^1 + 0\times2^0$
$= 8 + 0 + 0 + 0 = 8 \% 3 = 2 (q_2)$

$(1001)_2 = 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0$
$= 8 + 0 + 0 + 1$
$= 9 \% 3 = 0 (q_0)$



q_0 - Start state
q_0 - final state

In question, If I ask, binary number is divisible by 3 and 1 mod 3 then final state is q_1. Because q_1 state represents 1.
If I ask, binary number divisible by 3 and 2 mod 3, then final state is q_2.
2 mod 3 = 2(q_2) (final state).

Transition diagram

	0	1
$* \rightarrow q_0$	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

Divisibility Problems divisible by 3.

logic: 0 remainder q_0
1 remainder q_1
2 remainder q_2

Number divisible by 3

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

number is divisible by 3. not binary, ternary number. Just simple number like (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

		(0,3,6,9)	(1,4,7)	(2,5,8)
*→ 0	q_0	q_0	q_1	q_2
1	q_1	q_1	q_2	0
2	q_2	q_2	q_0	q_1

→ 0 (0, 3, 6, 9)

00 03 06 09

$\frac{00}{3}$ $\frac{03}{3}$ $\frac{06}{3}$ $\frac{09}{3}$

0 , 0 , 0 , 0 (remainder 0) → q_0

→ 0 (1, 4, 7)

$\frac{01}{3}$ $\frac{04}{3}$ $\frac{07}{3}$

1 , 1 , 1 , (remainder 1) → q_1

→ 0 (2, 5, 8)

$\frac{02}{3}$ $\frac{05}{3}$ $\frac{08}{3}$

2 2 2 (remainder 2) → q_2

1 $\;(0, 3, 6, 9)$

$\frac{10}{3}$ $\quad\frac{13}{3}$ $\quad\frac{16}{3}$ $\quad\frac{19}{3}$

1 $\quad$ 1 $\quad$ 1 $\quad$ 1 $\qquad$ (remainder 1) $\rightarrow q_1$

1 $\;(1, 4, 7)$

$\frac{11}{3}$ $\quad\frac{14}{3}$ $\quad\frac{17}{3}$

2 $\quad$ 2 $\quad$ 2 $\qquad$ (remainder 2) $\rightarrow q_2$

1 $\;(2, 5, 8)$

$\frac{12}{3}$ $\quad\frac{15}{3}$ $\quad\frac{18}{3}$

0 $\quad$ 0 $\quad$ 0 $\qquad$ (remainder 0) $\rightarrow q_0$

2 $\;(0, 3, 6, 9)$

$\frac{20}{3}$ $\quad\frac{23}{3}$ $\quad\frac{26}{3}$ $\quad\frac{29}{3}$

2 $\quad$ 2 $\quad$ 2 $\quad$ 2 $\qquad$ (remainder 2) $\rightarrow q_2$

2 $\quad$ 2

2 (1, 4, 7)

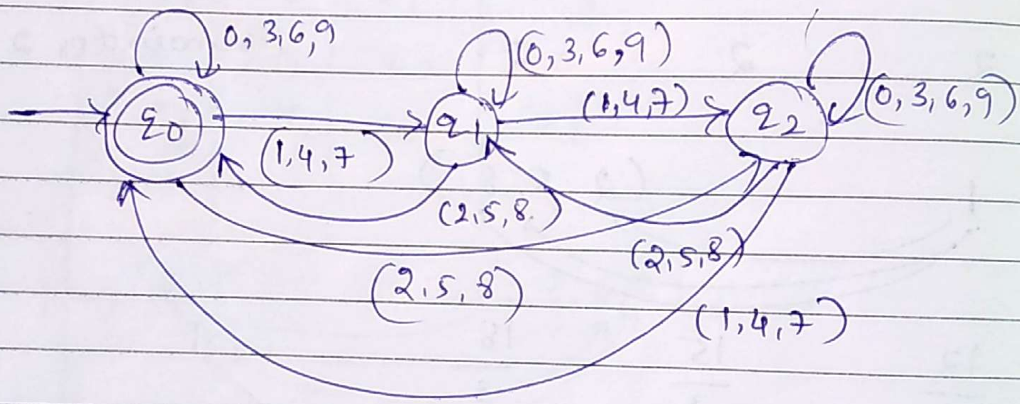
$\frac{21}{3}$ $\frac{24}{3}$ $\frac{27}{3}$

7, 8, 9 $(q_0) \rightarrow$ (remainder 0)

2 (2, 5, 8)

$\frac{22}{3}$, $\frac{25}{3}$, $\frac{28}{3}$

1, 1, 1 $(q_1) \rightarrow$ (remainder 1)



OR