

Date: _____

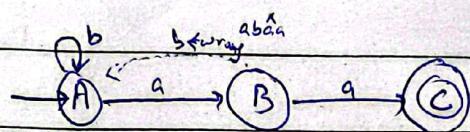
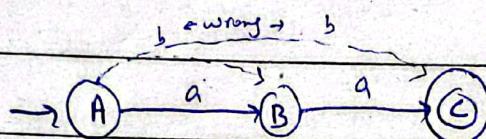
Day: _____

Construct a minimal DFA over $\Sigma = \{a, b\}$ such that
 $n_a(w) \geq 2$ $w \in \{a, b\}^*$

a^* → 0 or more occurrences of a

a^+ → 1 or more occurrences of a

Language = { aa, abab, aab, ..., babab, ... }



✓ baa
✓ bbca

b
→ A

a
→ B

b
→ C

a
→ D

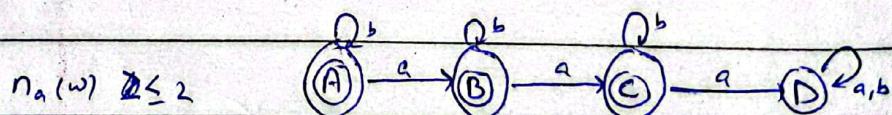
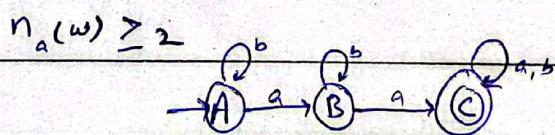
b
→ Dead State

ab²
abb²
bab²

a²b
aab²
bab²

x aab

?

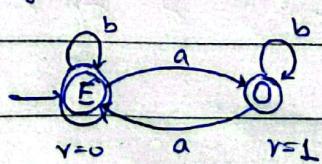


$n_a(w) \geq 2$

$n_a(w) \leq 2$

e.g. = aababab

4 mod 2 = 0
so, even no. of a's

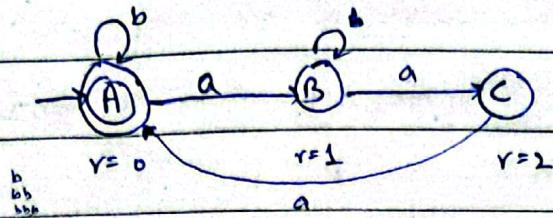


Date: _____

Day: _____

$$n_a(w) \bmod 3 = 0$$

if 1, 2 accepting state
if 2, c accepting state
otherwise no change



$$\text{DFA } w \in \{a,b\}^*$$

if we change 1^c to

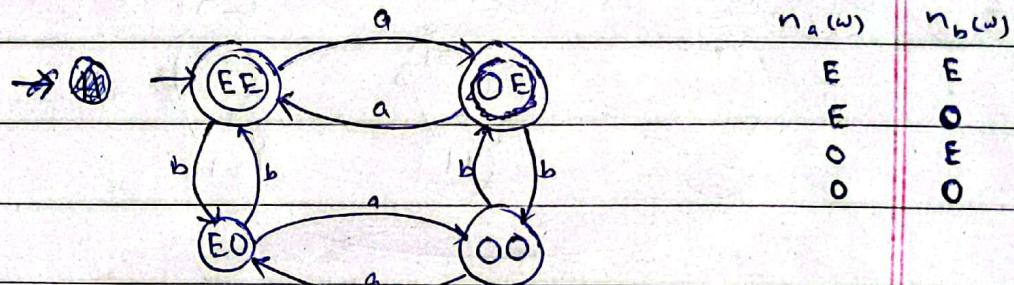
OR, run all except
Ac/EE become
accepting state

if we change it to 1
then accepting state

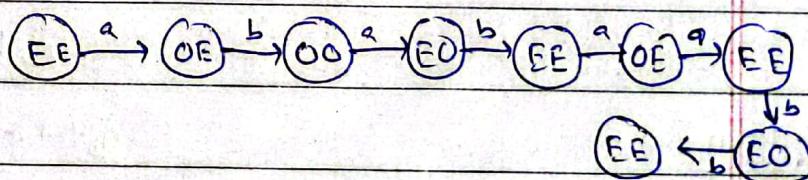
1^c would become BPO

$$n_a(w) \equiv 0 \pmod{2} \quad \& \& \quad n_b(w) \equiv 0 \pmod{2}$$

$$\text{or } n_a(w) \bmod 2 = 0 \quad \& \& \quad n_b(w) \not\equiv 0 \pmod{2} = 0$$



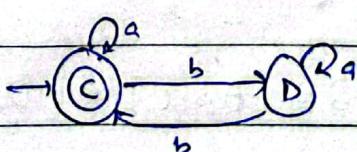
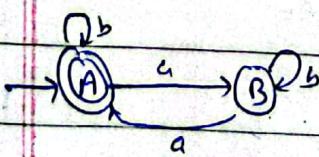
abababb



Cartesian Product

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

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

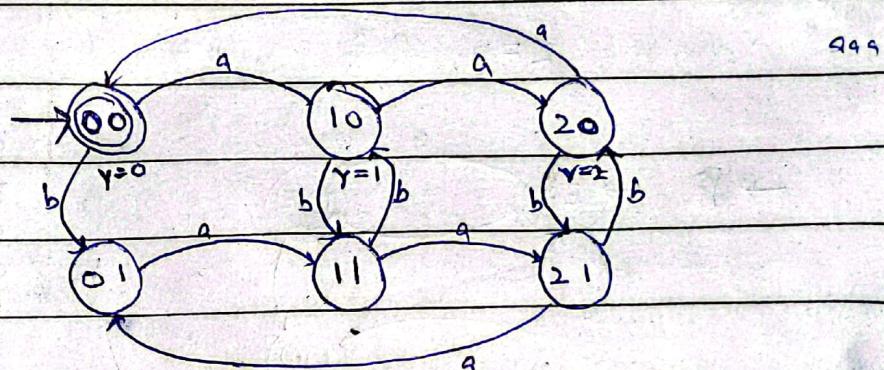
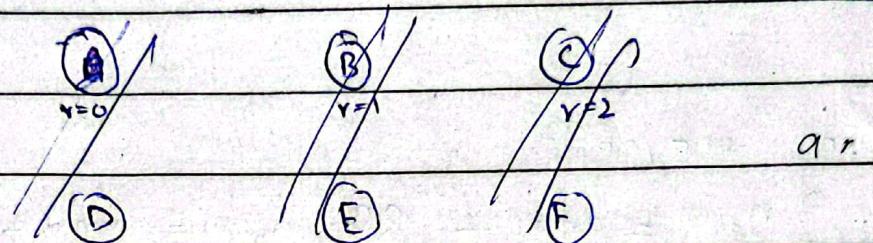


$$\{A, B\} \times \{C, D\} = \{AC, AD, BC, BD\}$$

Date: _____

Day: _____

⇒ Construct a minimal DFA which accepts set of all strings over $\Sigma = \{a, b\}$ where number of a's are divisible by 3 and number of b's are divisible by 2.



0 1 2

0 1

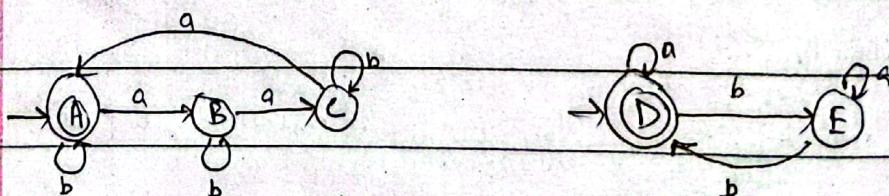
$$a \bmod 3 = 0 \quad \cancel{b \bmod 2}$$

0 0 10 20

0 1 11 21

$$n_a(w) \bmod 3 = 0$$

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

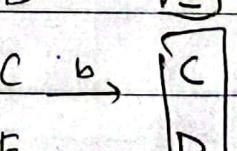
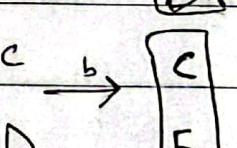
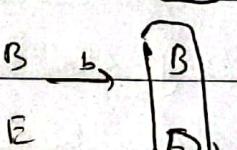
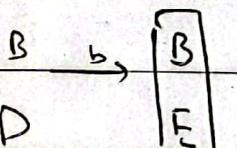
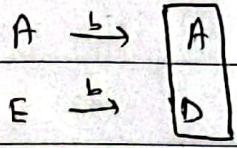
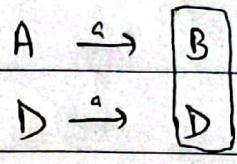
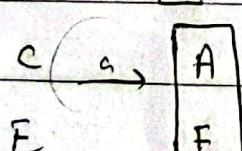
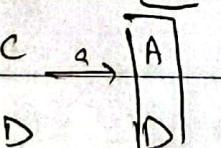
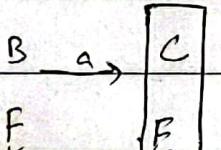
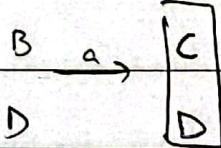
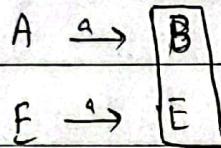
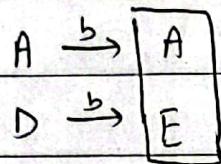
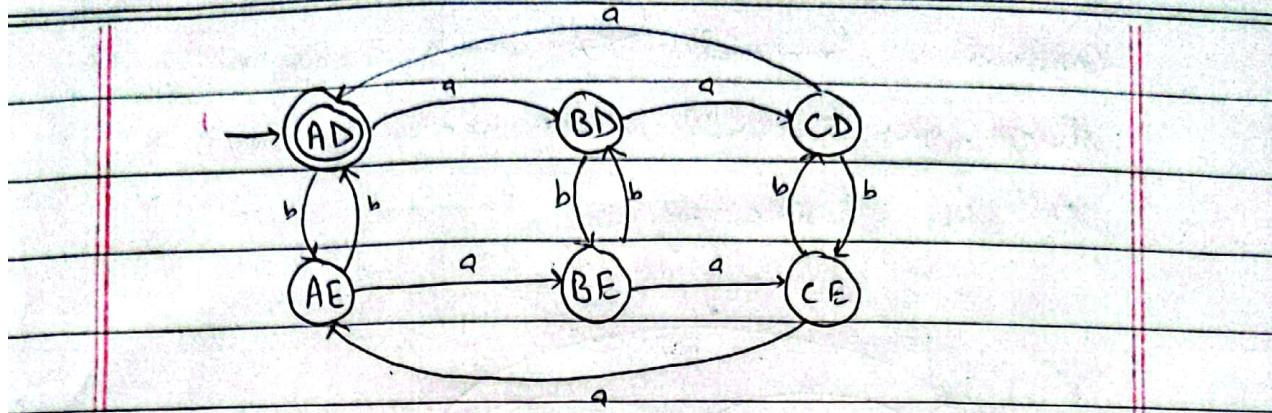


$$Q = \{A, B, C\} \times \{D, E\}$$

$$= \{AD, AE, BD, BE, CD, CE\}$$

Date: _____

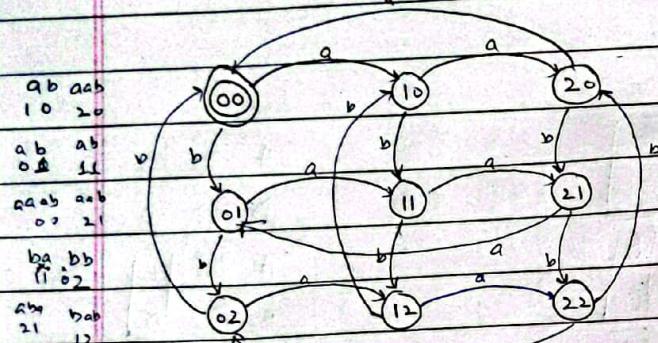
Day: _____



Day: _____

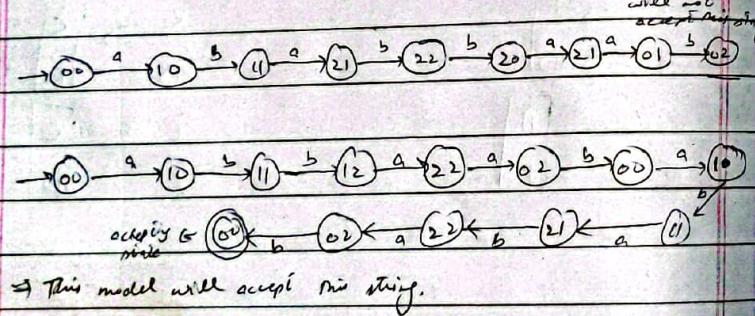
Date: _____

Construct a minimal DFA which accepts set of all strings over $\Sigma = \{a, b\}$ where number of a^3s and b^3s are divisible by 3.



Verify: ababbbaab $\xrightarrow{?} 02$

$\xrightarrow{00} abbaaabbbaabab$ ✓



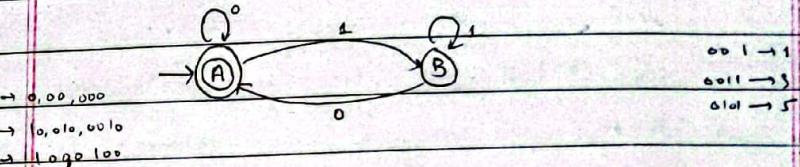
\Rightarrow This model will accept this string.

01101

Day: _____

Date: _____

Construct a minimal DFA which accepts set of all strings over $\Sigma = \{0, 1\}$ which when interpreted as a binary number is divisible by 2.



$0 \rightarrow 0, 00, 000$

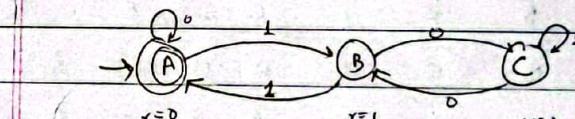
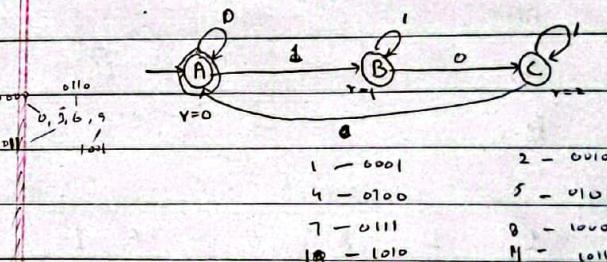
$2 \rightarrow 10, 010, 0010$

$4 \rightarrow 1010, 100$

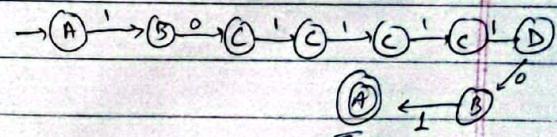
$$\begin{array}{r} 001 \\ 0011 \\ 0101 \end{array} \rightarrow \begin{array}{r} 1 \\ 3 \\ 5 \end{array}$$

$$\begin{array}{r} 3 \\ 13 \\ 3 \end{array} \quad \begin{array}{r} 0 \\ 3 \\ 1 \end{array}$$

divisible by 3



1011101
 $129 \rightarrow 64 + 64 + 1 = 129$ ✓

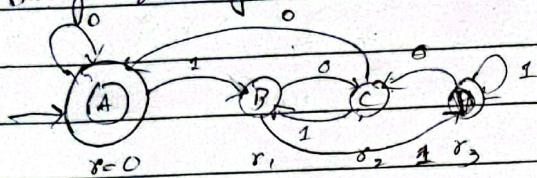


Machine / Model / Automaton / State Transition Graph / Diagram

Date: _____

Day: _____

Binary Div by 4



$$\begin{array}{llll}
 0000 = 0 & 001 = 1 & 010 = 2 & 011 = 3 \\
 0100 = 4 & 0101 = 5 & 0110 = 6 & 0111 = 7 \\
 1000 = 8 & 1001 = 9 & 1010 = 10 & 1011 = 11 \\
 \vdots & \vdots & \vdots & \vdots
 \end{array}$$

State Transition Table

	0	1	
*	A	B	$\frac{8 10}{8 8=0}$
A	C	D	$\frac{8 8=0}{6=2}$
C	A	B	
D	C	D	

	0	1	2	3	4	5	6	7
*	A	B	C	D	A	B	C	D
B	A	B	C	D	A	B	C	D
C	A	B	C	D	A	B	C	D
D	A	B	C	D	A	B	C	D

Sloes
ne go/cd

WAID

pooltolia

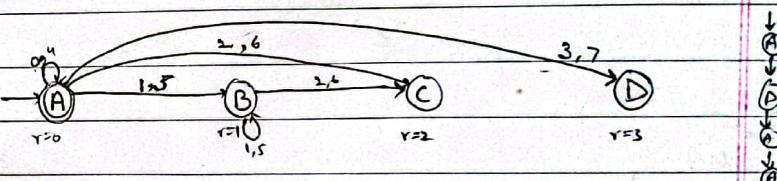
Date: _____

Day: 05-03-2024

Construct a minimal DFA that accepts set of all strings over $\Sigma = \{0, 1\}$ which when interpreted (as) a binary number is divisible by 4.

Construct a minimal DFA that accepts set of all strings over $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7\}$ which when interpreted as an octal number is divisible by 4.

$$\begin{array}{ll}
 (351)_8 & (480)_8 \\
 (351)_8 & (480)_8 \\
 (537)_8 & (740)_8
 \end{array}$$

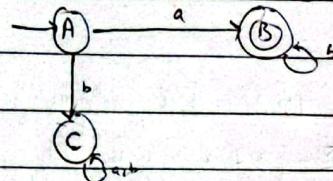


Date: _____

Day: _____

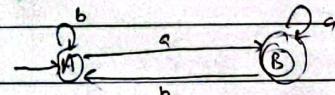
Construct a minimal DFA that accepts set of all strings over $\Sigma = \{a, b\}$ starting with 'a'.

Language = {a, aa, ab, aba}



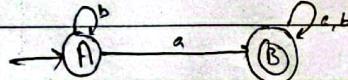
ending with 'a'

Language = {a, aa, ab, aba}

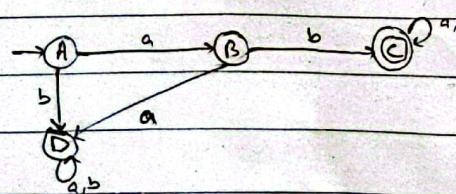


containing 'a'

Language = {a, ab, ba, abc, abb, bab, ...}



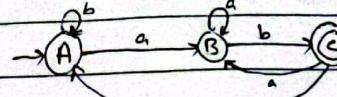
begin with 'ab'



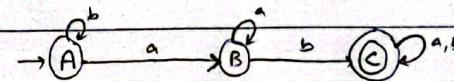
Date: _____

Day: _____

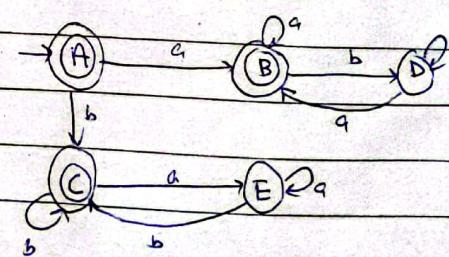
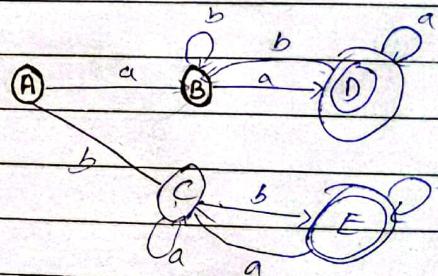
ending with 'ab'



contains 'ab'



beginning and ending with same symbol. $\{a, aa, bb, b, aba, bab, \dots\} \subseteq \Sigma^*$
is empty

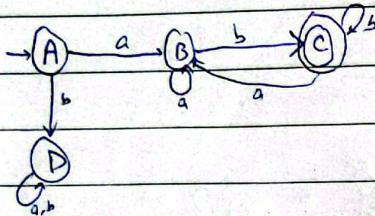


Date: 08-03-2024

Day: _____

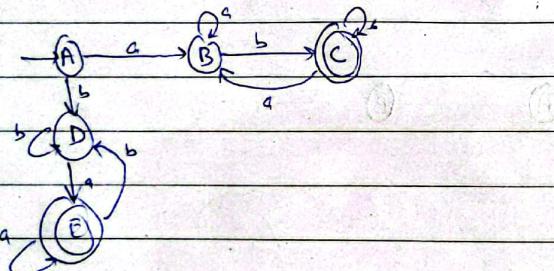
Construct a DFA which accepts set of all strings over $\Sigma = \{a, b\}$ that start with 'a' and ends with 'b'

$$\text{Language} = \{ab, abb, \dots\}$$



Starts and ends with different symbol

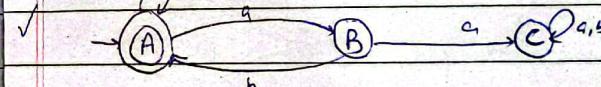
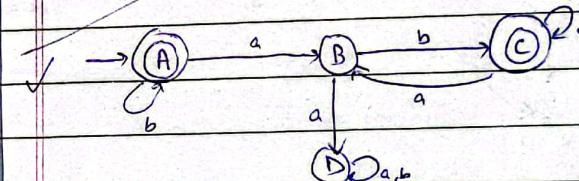
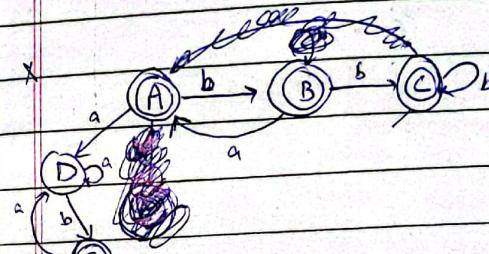
$$\text{Language} = \{ab, ba, aabb, bba, \dots\}$$



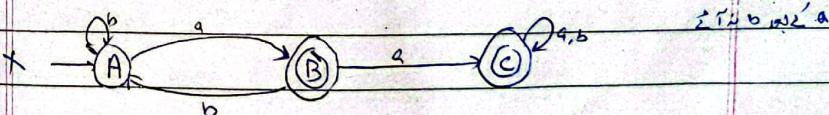
Date: _____

Day: _____

Construct a DFA over $\Sigma = \{a, b\}$ that accepts set of all strings where every 'a' is followed by a 'b'.
Language = $\{ \epsilon, b, bb, ab, abb, abab, \dots \}$



'a' is never followed by a 'b'

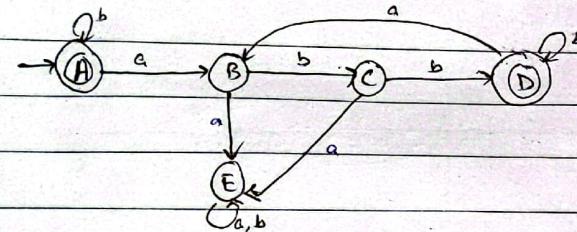
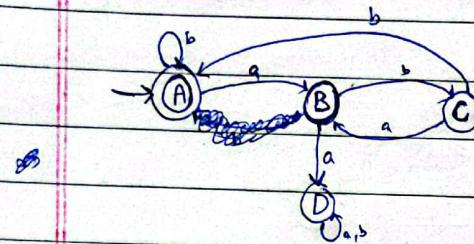


Date: _____

Day: _____

every 'a' is followed by a 'bb'

Language: $\{ \epsilon, bb, abb, abbabb, babb \}$

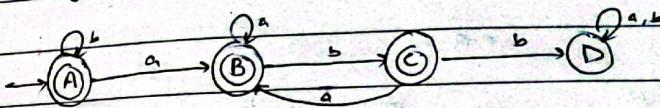


Date: 12-03-2024

Day: _____

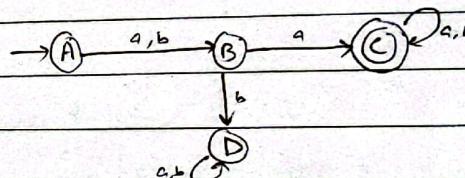
→ 'a' is never followed by a 'bb'

Language =

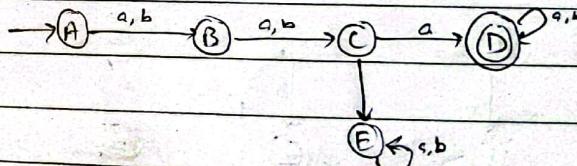


→ set of all strings where second symbol is 'a'

Language = $\{ aa, ba, aab, bab, baaba, \dots \}$

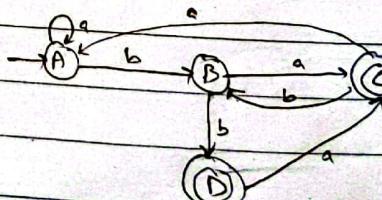


→ third symbol is 'a'



→ second last symbol is 'b'

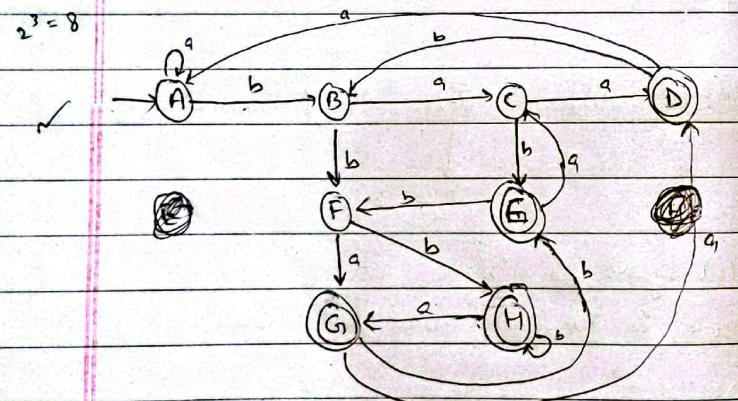
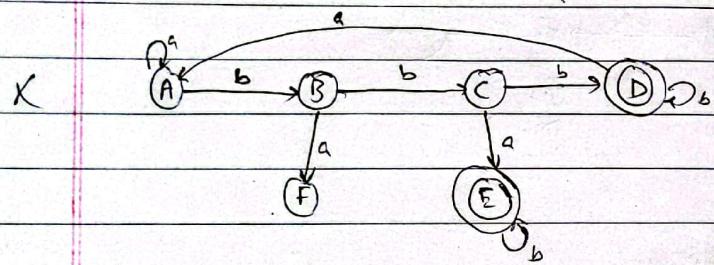
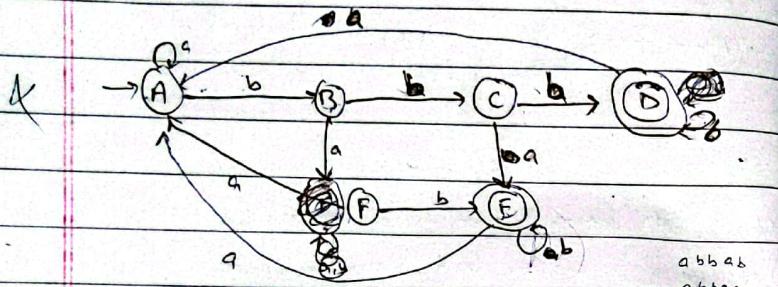
Language = $\{ ba, bb, bba, abb, abbbb, aabb, \dots \}$



Date: _____

Day: _____

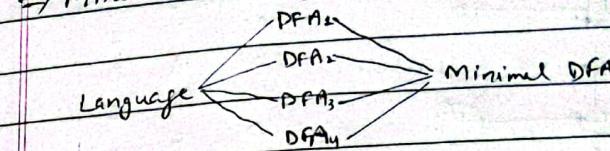
third last symbol is 'b'

Language = { baa, bab, bbb, abba, aabb ... }
bbb.

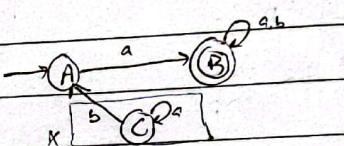
Date: 15-03-2024

Day: _____

⇒ Minimization of DFA



⇒ Eliminate Inaccessible States



⇒ Partitioning Method

- Two states can be combined if both are equivalent.
- Two states are equivalent if $\delta(A, x) \rightarrow F$, or, $\delta(A, x) \rightarrow F$
 $\delta(B, x) \rightarrow F$ $\delta(B, x) \rightarrow F$

• where x is an input stringif $|x|=0$, then A and B are 0-equivalentif $|x|=1$, then A and B are 1-equivalentif $|x|=2$, then A and B are 2-equivalent

⋮

if $|x|=n$, then A and B are n-equivalent

accept state

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

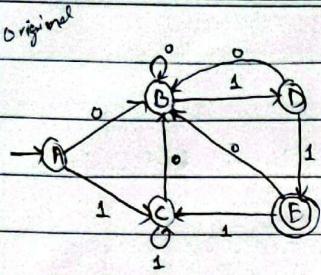
↓

↓

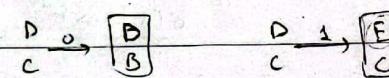
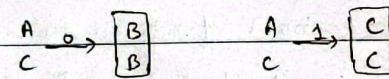
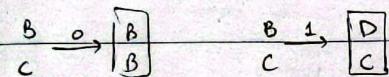
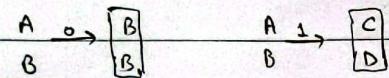
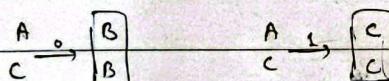
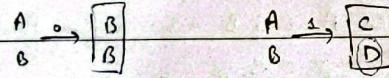
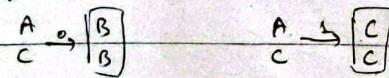
↓

Date: _____

Day: _____

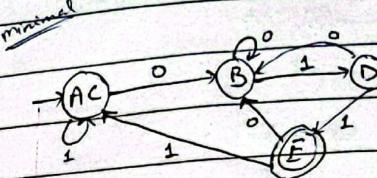


	0	1
$\rightarrow A$	B	C
B	B	D
C	B	G
D	B	E
$*E$	B	C

0-equivalence: $\{A, B, C, D\} \{E\}$ 1-equivalence: $\{A, B, C\} \{D\} \{E\}$ 2-equivalence: $\{A, C\} \{B\} \{D\} \{E\}$ 3-equivalence: $\{A, C\} \{B\} \{D\} \{E\}$ 

Date: _____

Day: _____



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

 $\Rightarrow \text{Verify: } 00101011$

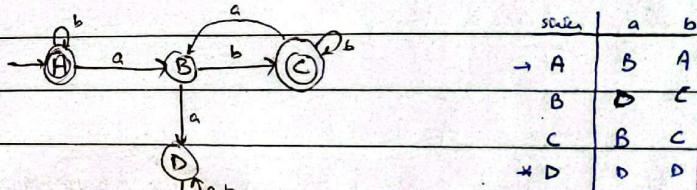
Original:

$$\rightarrow (A) \xrightarrow{0} B \xrightarrow{0} B \xrightarrow{1} D \xrightarrow{0} B \xrightarrow{1} D \xrightarrow{0} B \xrightarrow{1} D \xrightarrow{1} (E)$$

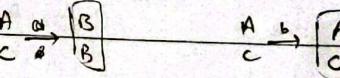
Minimal:

$$\rightarrow AC \xrightarrow{0} B \xrightarrow{0} B \xrightarrow{1} D \xrightarrow{0} B \xrightarrow{1} D \xrightarrow{0} B \xrightarrow{1} D \xrightarrow{1} (E)$$

Every 'a' has to be followed by a 'b'

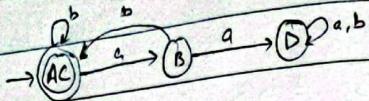


	a	b
$\rightarrow A$	B	A
B	D	C
C	B	C
$*D$	D	D

0-equivalence: $\{A, C\} \{B, D\}$ 1-equivalence: $\{A, C\} \{B\} \{D\}$ 2-equivalence: $\{A, C\} \{B\} \{D\}$ 

Day:

Date:



Verify: ababbbaab

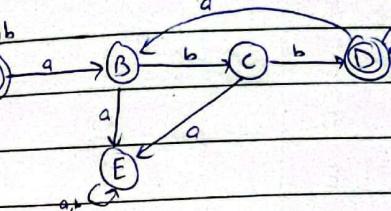
aabbbab

Date: 19-03-2024 Day: Tuesday

⇒ Partitioning Method

a is followed by a 'bb'

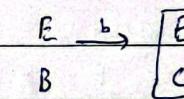
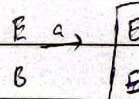
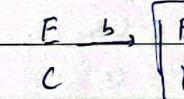
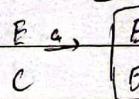
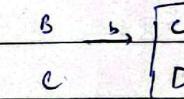
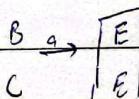
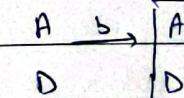
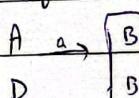
Language { E, a bb, b bb, ... }



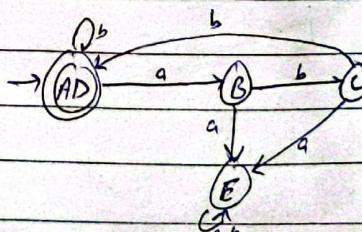
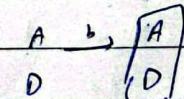
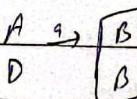
$\rightarrow *A$	a	b
B	B	A
C	F	C
$\times D$	E	D
E	E	E

0-equivalence: {B, C, E} {A, D}

1-equivalence: {A, D} {B} {C} {E}



2-Equivalence: {A, D} {B} {C} {E}



Date:

Day:

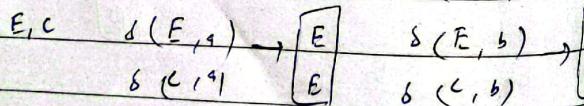
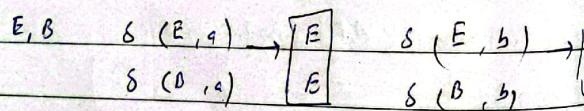
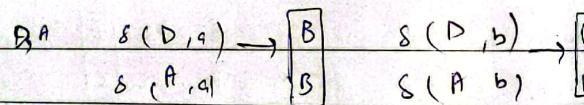
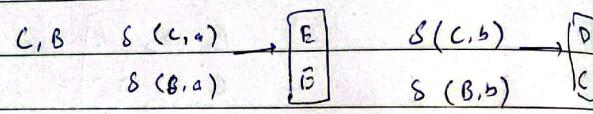
⇒ Table Filling Method

	A	B	C	D	E
A					
B	✓				
C	✓	✓			
D		✓	✓		
E	✓	✓	✓	✓	

⇒ Iteration-01:

Initially choose a pair in which 1 state is accepting and 1 state is non-accepting and mark that as a '✓'

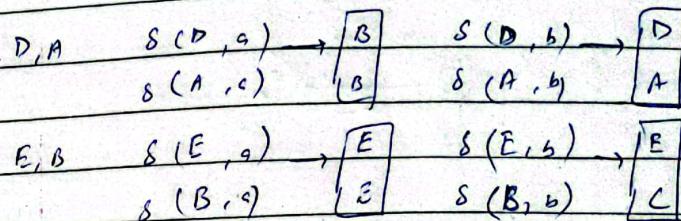
⇒ Iteration-02:



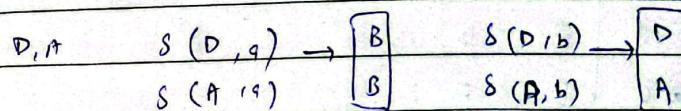
Date:

Day:

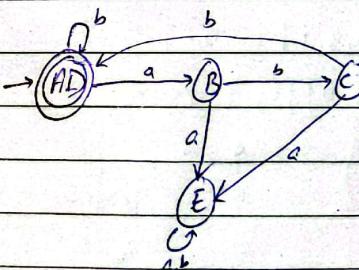
⇒ Iteration-03:



⇒ Iteration-04:

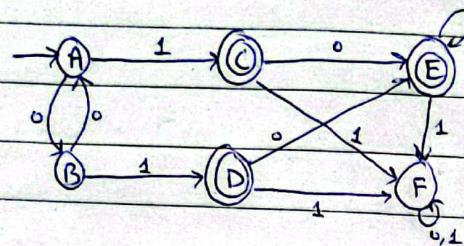


⇒ Merge the unmarked states.



Date: _____

Day: _____

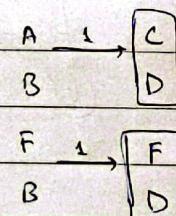
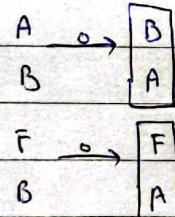


Partitioning Method:

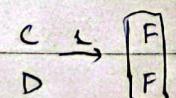
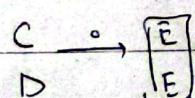
	0	1
→ A	B	C
B	A	D
* C	E	F
* D	E	F
* E	E	F
F	F	F

0-equivalence : {A, B, F} {C, D, E}

1-equivalence : {A, B} {F} {C, D, E}

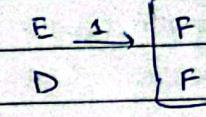
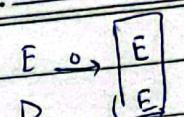


2-equivalence :

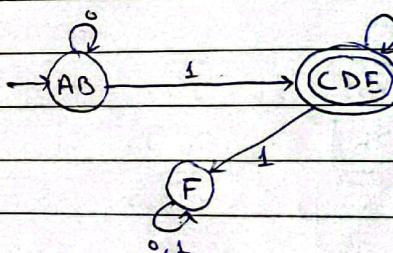
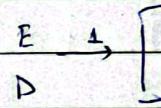
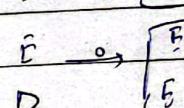
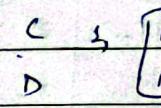
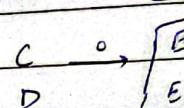
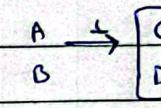
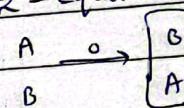


Date: _____

Day: _____



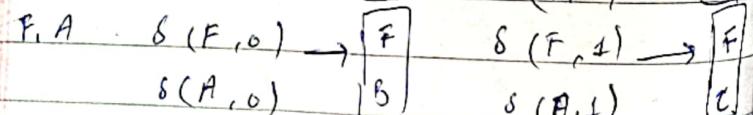
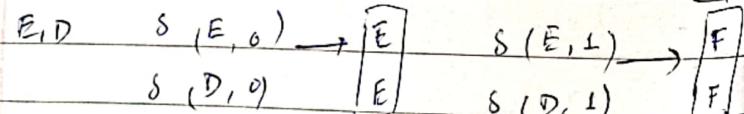
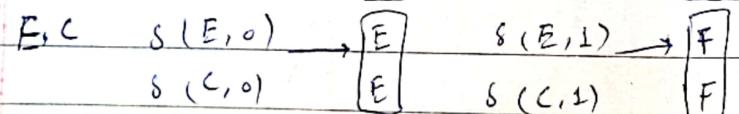
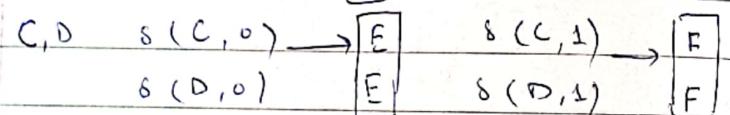
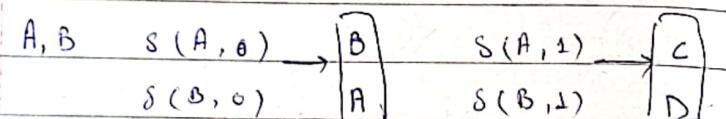
2-Equivalence: {A, B} {F} {C, D, E}



⇒ Table Filling Method

	A	B	C	D	E	F
A						
B						
C	✓	✓				
D	✓	✓	✗			
E	✓	✓	✗	✗		
F	✓	✓	✓	✓	✓	

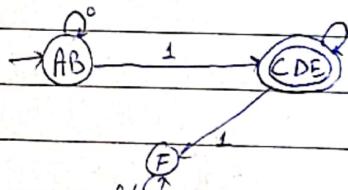
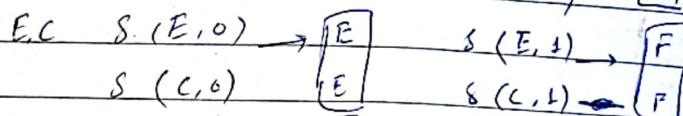
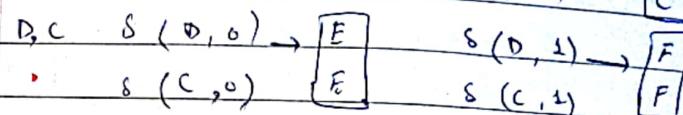
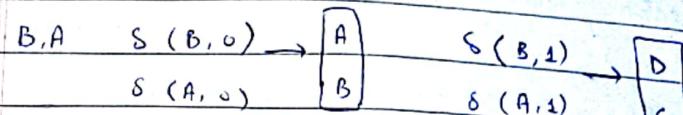
Iteration - 02:



Date: _____

Day: _____

Iteration - 03:



Date: 22-03-2024

Day: Friday

\Rightarrow NFA (Non-deterministic Finite Automata)

NFA is a 5-tuple (Q, Σ, S, q_0, F)

where $Q \rightarrow$ Finite set of states

$\Sigma \rightarrow$ Input alphabet

$q_0 \rightarrow$ initial state

$F \rightarrow$ Accepting / Final state

$\delta : Q * \Sigma \rightarrow Q^Q$

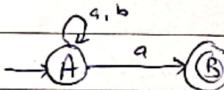
$$Q = \{A, B, C\}$$

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

$$P(Q) = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{B, C\}, \{A, C\}, \{A, B, C\}\}$$

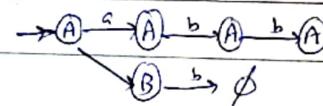
* Every DFA is a NFA but every NFA is not a DFA.

\Rightarrow ends with 'a' :-

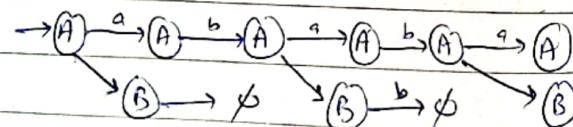


Verify:

$$X \Rightarrow abb$$



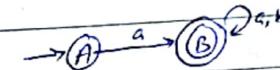
$$✓ \Rightarrow ababa$$



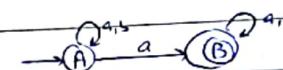
Date: _____

Day: _____

ii) begins with 'a'

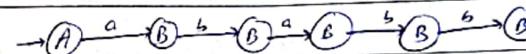


iii) contains 'a'



iv) verify

ababb



ababb

$$76 - 63 = 3824$$

Day: Tuesday

EACD

NFA \Rightarrow Non-Deterministic Finite Automata

NFA \rightarrow DFA \rightarrow minimization

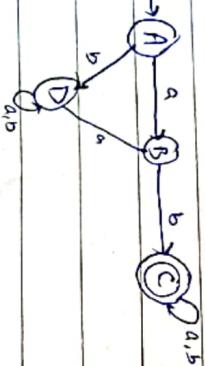
Subnet Configuration Method

\rightarrow Construct a minimal DFA over $\Sigma = \{a, b\}$ that
accepts strings starting with "ab".

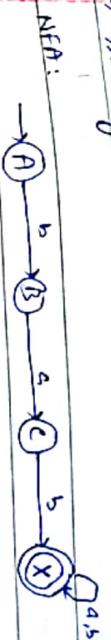
$\text{NPA} \Rightarrow$

→(A) →(B) →(C)
Eots
Staged Transition Task

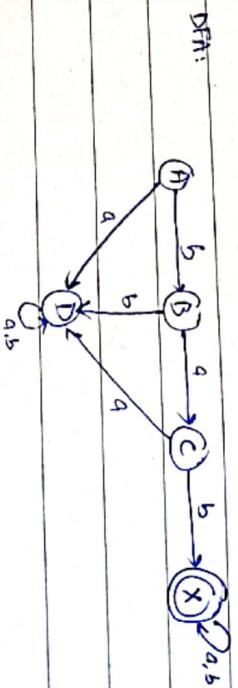
States	a	b	a	b
$\rightarrow \{A\}$	$\{b\}$	\emptyset	$\rightarrow [A]$	$[B]$
$\{B\}$	\emptyset	$\{c\}$	$\{B\}$	$[D]$
$\{C\}$	$\{c\}$	$\{b\}$	$\{D\}$	$[D]$
$\star [C]$	$\{c\}$	$\{c\}$		



\Rightarrow starting with 'ba's'

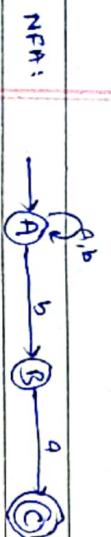


State Transition Table			
	a	b	
*	a → {A}	b → {B}	a → {A}
{B}	{C}	a → {B}	b → {D}
{C}	a → {X}	b → {B}	a → {C}
* {X}	{X}	{X}	a → {D}
	*	[X]	b → [X]

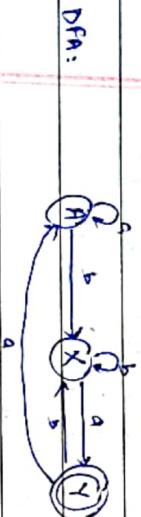


DFA \Rightarrow

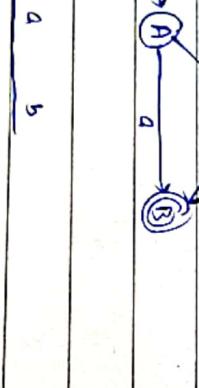
⇒ ends with 'ba'



$\rightarrow \{A\}$	$\{A\}$	$\rightarrow [A]$	$[A]$	$[AB]$
$\{B\}$	$\{B\}$	$\rightarrow [B]$	$[B]$	$[AC]$
$\{C\}$	\emptyset	$\rightarrow [C]$	$[C]$	$[AB]$
$\star \{C\}$	\emptyset	$\star [AC]$	$[A]$	$[AB]$



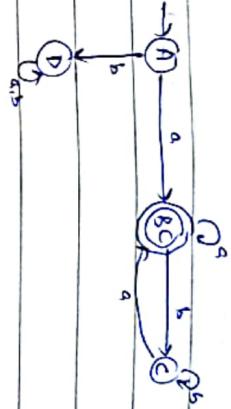
⇒ starting and ending with 'a'



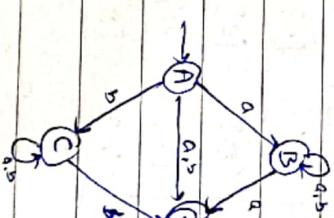
a	b	a	b
$\rightarrow \{A\}$	$\{AB\}$	$\rightarrow [A]$	$[BC]$
$\star \{B\}$	\emptyset	$\star [BC]$	$[C]$
$\{C\}$	$\{B,C\}$	$\{B\}$	$\{B\}$

⇒ beginning and ending with a same symbol.

Verify
abababa
Previous example verification.

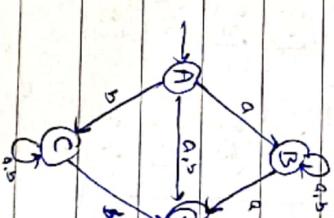


NFA:



a	b	a	b
$\rightarrow \{A\}$	$\{ABC\}$	$\rightarrow [A]$	$[BC]$
$\star \{B\}$	\emptyset	$\star [BC]$	$[C]$
$\{C\}$	$\{B,C\}$	$\{B\}$	$\{B\}$

NFA:

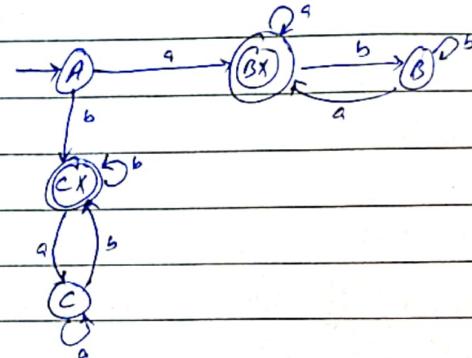
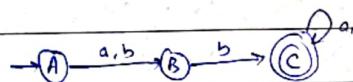


a	b	a	b
$\rightarrow \{A\}$	$\{ABC\}$	$\rightarrow [A]$	$[BC]$
$\star \{B\}$	\emptyset	$\star [BC]$	$[C]$
$\{C\}$	$\{B,C\}$	$\{B\}$	$\{B\}$

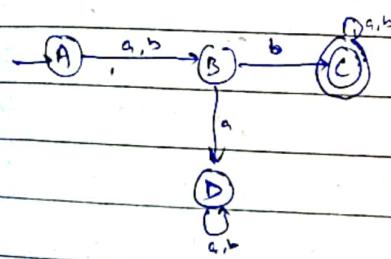
Date: _____

Day: _____

DFA:

 \Rightarrow second symbol is 'b'

	a	b		a	b
$\rightarrow \{A\}$	$\{B\}$	$\{\emptyset\}$	$\rightarrow [A]$	$\{B\}$	$\{\emptyset\}$
$\{\emptyset\}$	\emptyset	$\{C\}$	$[B]$	$\{D\}$	$\{C\}$
$* \{C\}$	$\{C\}$	$\{C\}$	$* [C]$	$\{C\}$	$\{C\}$
			$\{D\}$	$\{D\}$	$\{D\}$

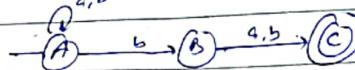


Date: _____ Day: _____

Day: _____

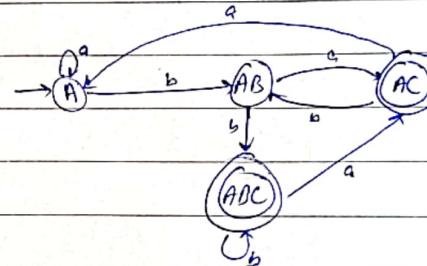
 \Rightarrow second last symbol is 'b'

NFA:



	a	b		a	b
$\rightarrow \{A\}$	$\{A\}$	$\{A,B\}$	$\rightarrow [A]$	$\{A\}$	$\{AB\}$
$\{B\}$	$\{C\}$	$\{C\}$	$[AB]$	$\{AC\}$	$\{ABC\}$
$* \{C\}$	\emptyset	\emptyset	$* [AC]$	$\{A\}$	$\{AB\}$
			$* \{ABC\}$	$\{AC\}$	$\{ABC\}$

DFA:



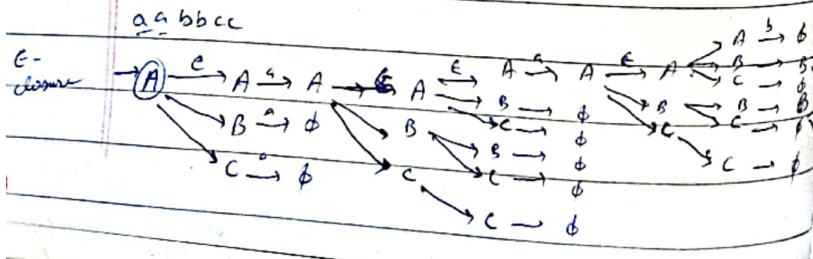
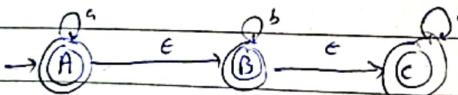
Date: 29-03-2024

Day: Friday

 ϵ -NFA (Epsilon NFA) \rightarrow Extended type of NFA \rightarrow ϵ -NFA is easier to construct ϵ -NFA \rightarrow NFA \rightarrow DFA \rightarrow Minimal DFA \rightarrow can be more than one ϵ -NFA, NFA, DFA \rightarrow There is only one minimal DFA. \rightarrow ϵ -NFA is a 5-tuple set $\{Q, \Sigma, q_0, F, \delta\}$ $Q \rightarrow$ Finite set of states $\Sigma \rightarrow$ Input Alphabet $q_0 \rightarrow$ Initial state $F \rightarrow$ Accepting state/states $\delta \rightarrow Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$ Language = $\{a^m b^n c^p \mid m, n, p \geq 0\}$

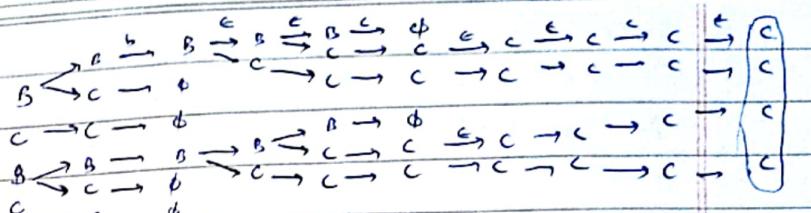
$$a' b' c' = ac \quad | \quad a' b' c' = c$$

one occurrence of a
followed by one occurrence of c one occurrence of c

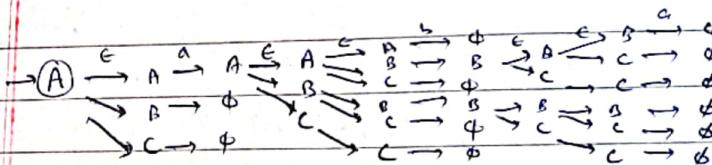
 ϵ -closure ($\delta(\epsilon\text{-closure}(A_i), x)$)

(iii)

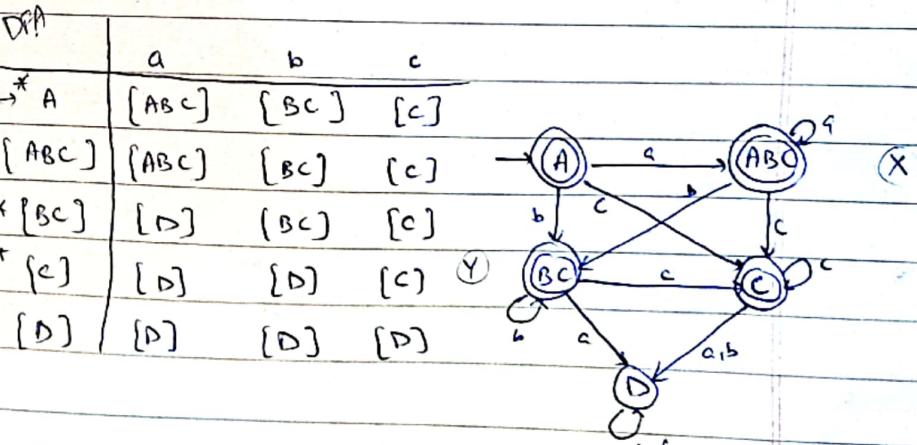
Day:



abac



NFA	a	b	c	ϵ -closure
* A	{A, B, C}	{B, C}	{C}	A {A, B, C}
* B	Ø	{B, C}	{C}	B {B, C}
* C	Ø	Ø	{C}	C {C}



Date:

Day:

Date: 02/04/2024

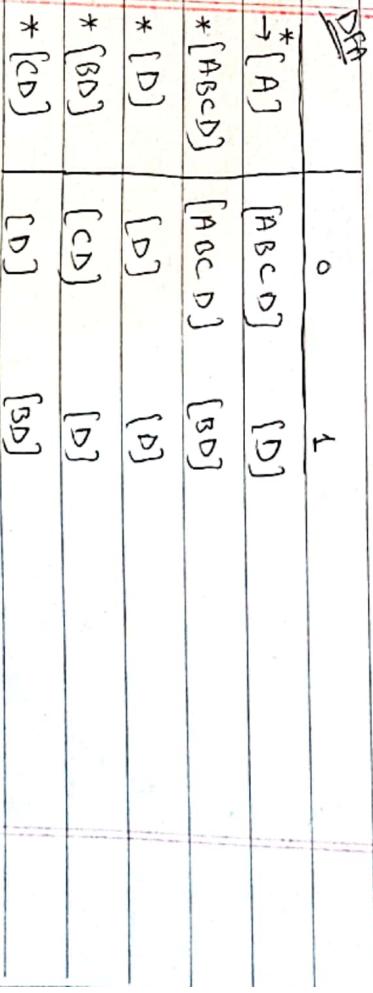
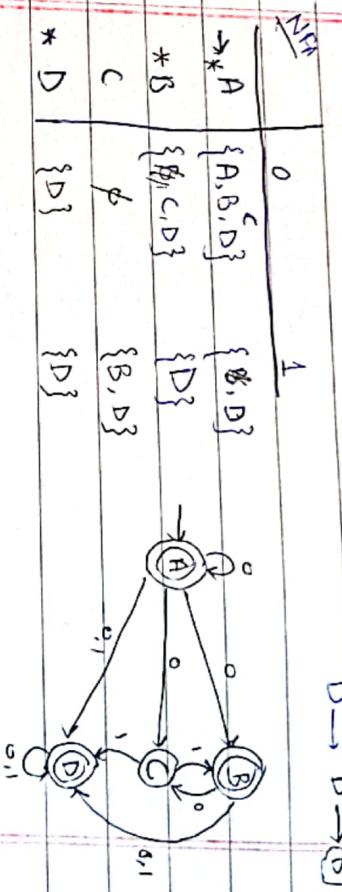
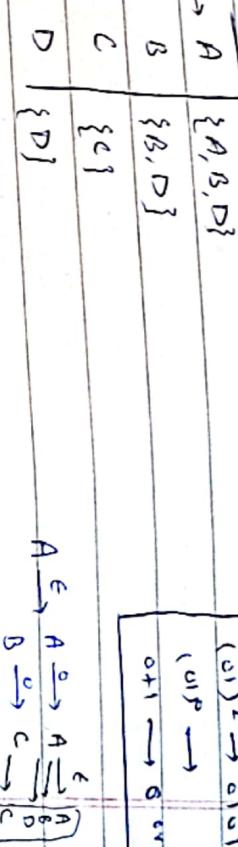
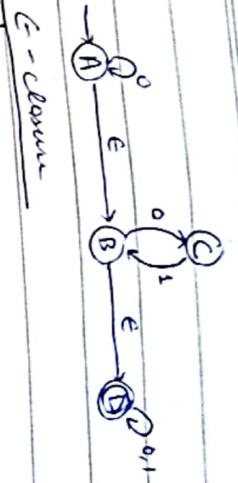
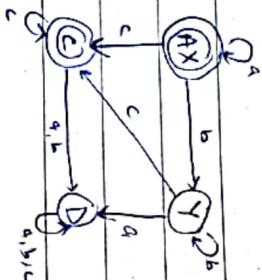
$$\begin{aligned} 101 \\ 0^0 (01)^0 (01+1)^3 &= (0+1)(0+1)(0+1) = 101 \\ \text{Day: } &\uparrow \text{odd/even} \end{aligned}$$

→ ABC

$$A \xrightarrow{a} AB \xrightarrow{b} BC \xrightarrow{a} D \xrightarrow{c} D$$

→ abc

$$A \xrightarrow{a} AB \xrightarrow{b} BC \xrightarrow{a} C \xrightarrow{c} C$$

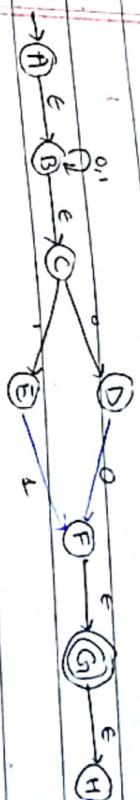
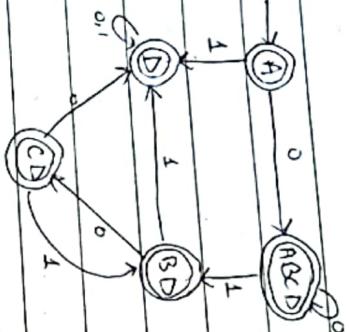
or equivalence : $\{P, X\} \{Y\} \{C\} \{D\}$ 1 - equivalence : $\{A, X\} \{Y\} \{C\} \{D\}$ 2 - equivalence : $\{A, X\} \{Y\} \{C\} \{D\}$ 

Date: _____

Day: _____

Date: _____

Day: _____



\Rightarrow Regular language

A language is said to be regular if it can be described by a Finite Automata.

\times Language = $\{0^n 1^n \mid n \geq 1\}$ - context free grammar
pushdown Automata.

6-tuple set

\Rightarrow Finite Automata with Output

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

$Q \rightarrow$ Finite set of states

$\Sigma \rightarrow$ Input Alphabet

$\delta \rightarrow$ Transition Function

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

$q_0 \rightarrow$ Initial state

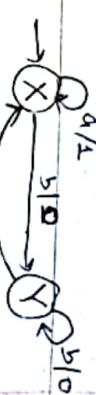
$\Delta \rightarrow$ Output Alphabet

$\gamma \rightarrow$ Output Function.

Every state is associated with an output symbol.

Every state on its input mapping is associated with an output.

$\lambda: Q \rightarrow \Delta$



Date: _____

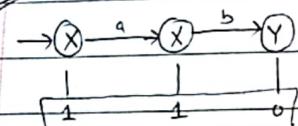
Day: _____

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

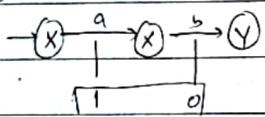
$$\Delta = \{0, 1\}$$

"ab"

Moore's



Mead's



⇒ Finite Automata with Output

generates "1" if "ab" occurs

not ⇒ Moore's Machine:

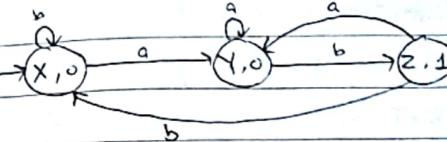
Construct a Moore's Machine

that takes set of all strings over $\Sigma = \{a, b\}$ and prints "1" as output for every occurrence of "ab"

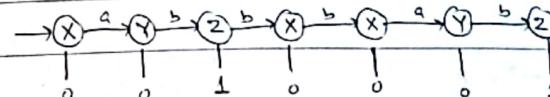
$$\Sigma = \{a, b\}$$

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

Date: _____ Day: _____

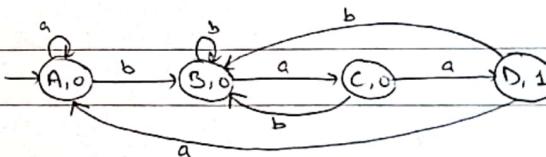


abbbbab

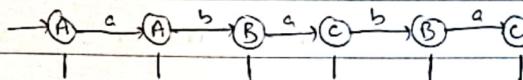


Two '1's show that "ab" has occurred two times in the string.

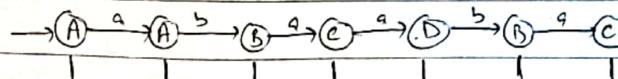
prints '1' as output whenever "baa" occurs.



ababa



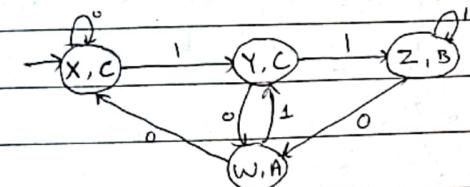
abaaba



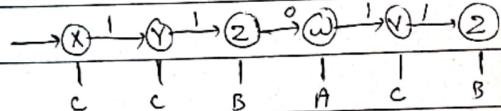
Date: _____

Day: _____

⇒ Construct a Moore's Machine that takes set of strings over $\Sigma = \{0, 1\}$ and prints 'A' as output if the input ends with "10" and produce 'B' if the string ends with "11" otherwise produces "C".



11011



⇒ Mealy's Machine

Construct a Mealy's Machine that takes binary number as input and produces 2's complement of that number as output

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

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

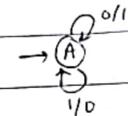
Date: _____

Day: _____

1's complement

$$101011$$

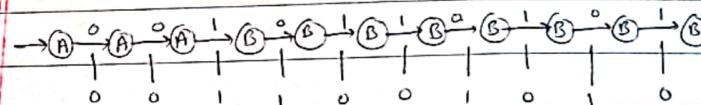
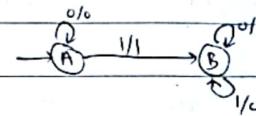
$$\rightarrow 010100$$



2's complement

$$1010110100$$

$$0101001100$$



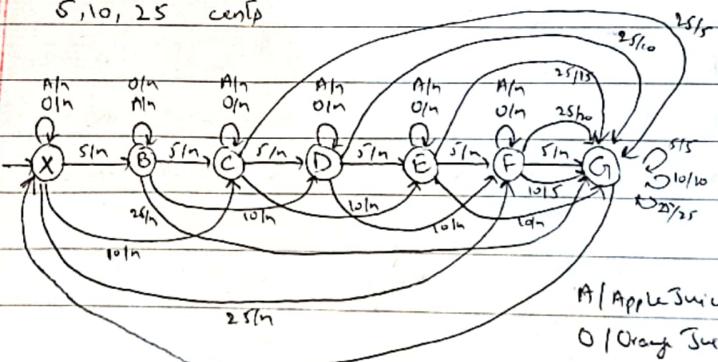
0101001100

⇒ Mealy's Machine

⇒ Vending Machine

A → Apple Juice → 30 cent_p

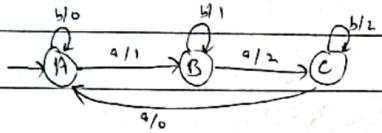
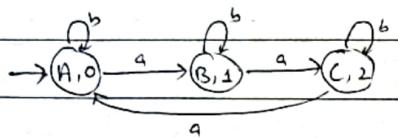
O → Orange Juice

5, 10, 25 cent_p

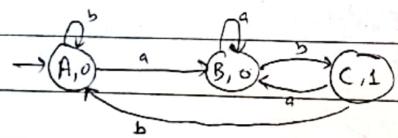
Date: _____

Day: _____

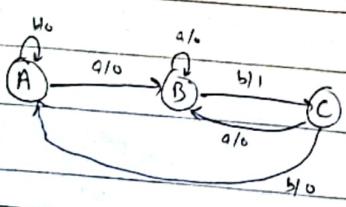
⇒ Conversion → Moore's → to → Mealy's Machine



	a	b	Δ		a	b
→ A	B	A	0	→ A	(B, 1)	(A, 0)
B	C	B	1	B	(C, 2)	(B, 1)
C	A	C	2	C	(A, 0)	(C, 2)



	a	b	Δ		a	b
→ A	B	A	0	→ A	(B, 0)	(A, 0)
B	B	C	0	B	(B, 0)	(C, 1)
C	B	A	1	C	(B, 0)	(A, 0)



Date: _____

Day: _____

⇒ Conversion!

Mealy's to Moore's Machine

