

Theory of Computation

Why Study TOC :-

→ Mathematical models of a Computer?

↳ what cannot a Computer do?

↳ How efficiently can a Computer solve?

Turing Machine is a mathematical model of a Computer

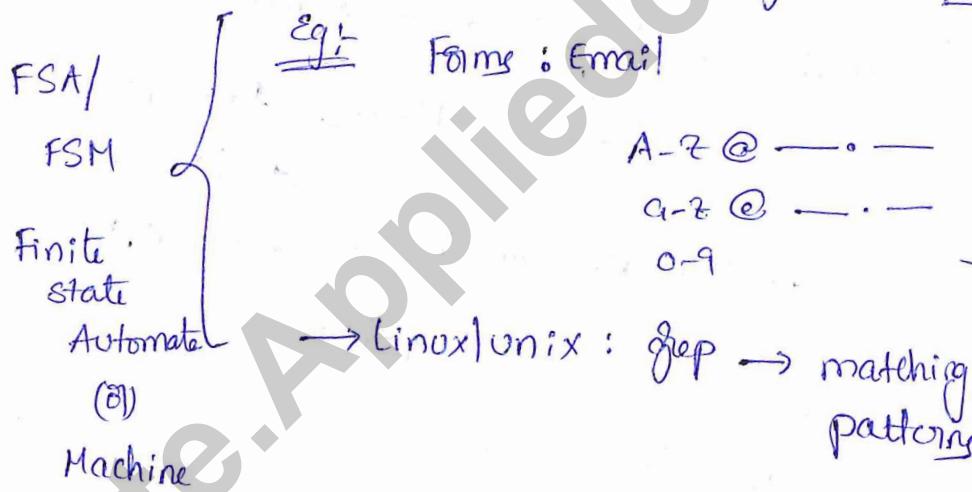


Alan Turing - Father of Computer Science

Turing Prize/Award - Award given for Computer Scientists

TOC has tons of real world applications (TOC, CO)

C-program → [GCC] → [exe]



Automata / Machine / Math-model :-

How to denote a pattern : Grammar ← Linguistics

push-down Automata (PDA) :-

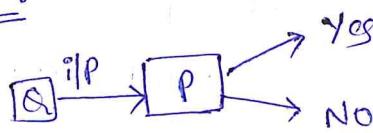
↳ Code → [Compiler] → Verify Syntax

②

③ Turing Machines

Ph: 844-844-0102

Halting problem:



Question is whether Q terminates / halts?

How efficiently can you solve a problem?

(Q)
write a program

Deep Turing Machines (DTM)

↑
Building a TM, which can write the program
Artificial Intelligence

Mathematical preliminaries:-

① set: collection of distinct objects/elements

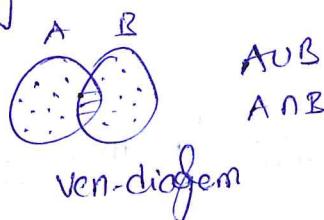
↓
No repetitions $\{1, 1, 2\} \times$

$$S = \{1, 2, 3\} \checkmark$$

size | cardinality $|S| = 3$ $S' = \{A, B, \dots\}$

$$|S'| = 2^6$$

Visually



② Cartesian product:

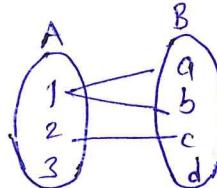
$$A = \{1, 2\}$$

$$\Rightarrow A \times B = \{(1, a), (2, a) \\ (1, b), (2, b) \\ (1, c), (2, c)\}$$

$$B = \{a, b, c\}$$

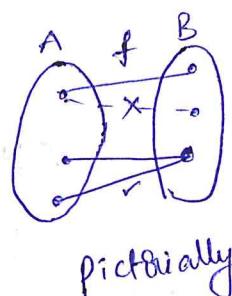
$$A \times B = \{(a, b) \mid a \in A, b \in B\}$$

↳ ordered pairs
Tuples



$$R = \{(1, a), (1, b), (2, c), (2, d)\} \subseteq A \times B$$

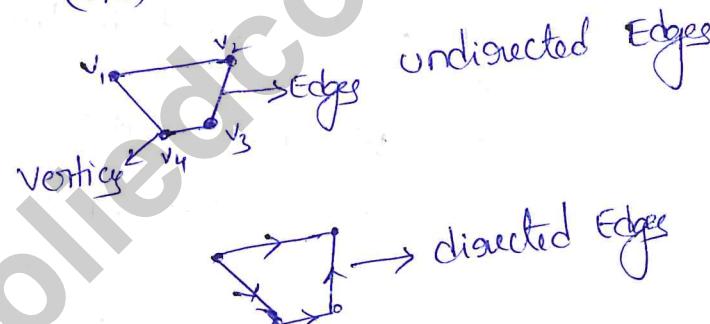
- ④ Function: It is a special type of relation



$$\begin{aligned} (a, b) \in R &\Rightarrow b=c \\ (a, c) \in R & \\ a \in A & \\ \#b, c \in B & \end{aligned}$$

- ⑤ Graphs: It is set of vertices and edges.

(V, E)



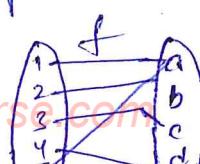
- ⑥ Sequences:

$\{A, B, C, \dots\}$: seq of Alphabets } ordering is important
 $\{1, 1, 2, 3, 5, \dots\}$: seq of Fibonacci Number } ordering is important
 $\{A, B, C, \dots\} \Rightarrow$ set of Alphabets
 $= \{Z, Y, \dots, D, O, B, A\}$ } ordering does not matter

Sequence we can think of a function from natural numbers to objects.

$(1, 1, 2, 3, 5, 8, \dots)$ } specific ordering

→ Repetition are allowed in sequences



Formal Languages :- (Linguistics)Alphabets:

→ Finite Set of Symbols

→ Represented by Σ

English

$$\Sigma_{Eng} = \{A, B, \dots, Z\}$$

$$\Sigma_{bin} = \{0, 1\}$$

$$\Sigma_{dec} = \{0, 1, \dots, 9, .\}$$

$$\Sigma_{Hin} = \{3T, 3T, \bar{3}, \dots\}$$

$$\Sigma_{tel} = \{B, Q, Q, \bar{Q}, \dots\}$$

String/Word:-

→ Finite Sequence of alphabets.

abc, 0101, Applied, GATE, 123.45

Length = # alphabets in the String/word.

$$\text{Eq: } w = abc \Rightarrow |w| = 3$$

$$w = 0101 \Rightarrow |w| = 4$$

Empty String:- $\epsilon, \Lambda, \lambda$

$$\text{Len} = 0 \Rightarrow |\epsilon| = 0$$

$$e.s = s.e = s$$

Σ^* = set of all the strings (including Empty strings)
formed over the alphabet Σ

Language: set of strings from the given alphabet.

→ Subset of Σ^*

Grammar:- set of rules to produce valid strings in

a formal language.

Substring :-

u is a Substring of v if $\exists xuy=v$, $u,x,y,v \in \Sigma^*$

Ph: 844-844-0102

(5)

Eg :-
 $\begin{array}{c} abcde \\ \underline{x} \quad u \quad y \end{array}$
 $\begin{array}{c} e \quad abcde \\ x \quad u \quad y \end{array}$
Prefix :-

u is a prefix of v , if $\exists x$ such that $ux=v$

 $\begin{array}{c} \checkmark \\ \overbrace{01010101}^v \\ u \end{array}$
 $u, x, v \in \Sigma^*$
concatenation of Strings :- uv such that $u, v \in \Sigma^*$

$$\Rightarrow uv = abcxyz$$

 $u=abc$
 $v=xyz$
Finite Automata :-
operations on Alphabets, Strings & Languages :-

① power of an Alphabet :-

$$\text{let } \Sigma = \{0,1\} \Rightarrow \Sigma^k$$

 \uparrow
 Binary Alphabet $\Sigma^0 = \epsilon$ String of Length zero

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

 $\Sigma^2 = \Sigma \cdot \Sigma = \{00, 01, 10, 11\}$ Set of all strings of length 2

 $\Sigma^3 = \Sigma \Sigma \Sigma = \{000, \dots\}$ all strings of length 3

 \vdots
 \vdots
 $\Sigma^K = \{w \mid |w|=K\}$ w are words formed using Σ

② Kleene closure :- Σ^*

$$\downarrow \Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \dots \text{ if } \Sigma = \{0,1\}$$

⑥

$$\Sigma^* = \{ \epsilon, 0, 1, 00, 01, 10, 11, \dots \}$$

$$\Sigma^* = \bigcup_{i \geq 0} \Sigma^i$$

$\Sigma^* = \{ w \mid |w| \geq 0 \}$ w is word formed using Σ

Positive closure :-

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots , \Sigma = \{0, 1\}$$

$$\Sigma^+ = \{0, 1, 00, 01, 10, 11, \dots\} \text{ does not contain } \epsilon$$

$\Sigma^+ = \{ w \mid |w| \geq 1 \}$ w is a word formed using Σ

Basic properties of Σ^+ & Σ^* :

$$\Sigma^* = \{\epsilon\} \cup \Sigma^+$$

$$\Sigma^* \cap \Sigma^+ = \Sigma^+$$

$$\Sigma^* \cup \Sigma^+ = \Sigma^*$$

$$\Sigma^* \cdot \Sigma^* = \Sigma^*$$

$$\Sigma^* \cdot \Sigma^+ = \Sigma^+ = \Sigma^+ \cdot \Sigma^*$$

Examples of Languages:-

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

① $L_1 = \Sigma^* = \text{universal Language} \quad (\text{Contains all the possible strings in the Language})$

$$② L_2 = \Sigma^+ \subseteq \Sigma^*$$

$$③ L_3 = \{ 0^n \mid n \geq 1 \} \quad \Sigma = \{0\} \subseteq \Sigma^*$$

$$= \{0, 00, 000, \dots\}$$

Mail: gatcse@appliedcourse.com consists of strings of one or more zeros.

Empty, Finite and Infinite Languages :-

① L = { } = \emptyset does not even contain ϵ

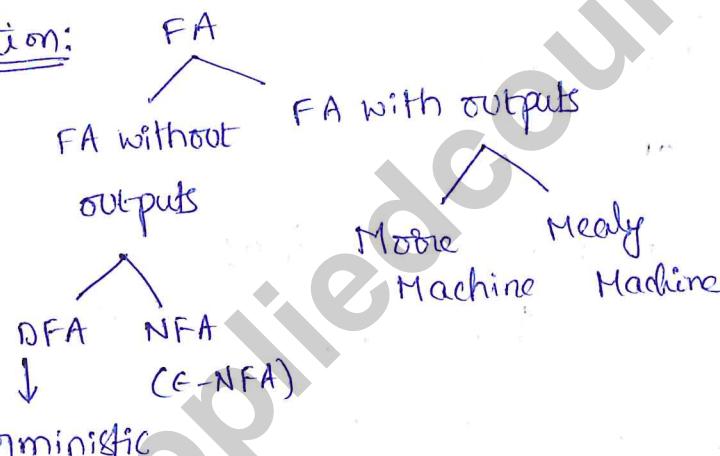
$$|L| = 0$$

② Finite: |L| is finite $\Rightarrow L = \{ w \mid |w| \leq 2 \}$ over $\Sigma = \{0, 1\}$

Infinite: |L| is not finite

$$= \{\epsilon, 0, 1, 01, 10, 00, 11\}$$

$$|L| = 8$$

Finite State Machine/Automata :-Categorization:

What are the applications of FSM | FA (DFA)

- DFA to determine if a given decimal number is even/odd.
- Simpler email-address verification FSM
- Turnstile (wikipedia)

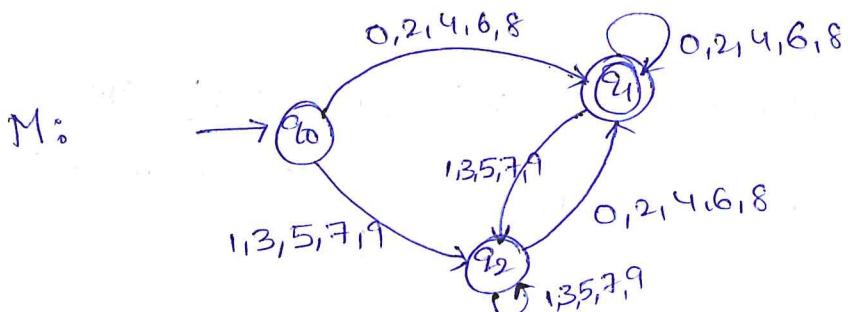
Eg1: Design a FSM to determine if a given decimal number is even/odd

$$\Sigma = \{0, 1, \dots, 9\}$$

⑧

let $w_1 = 123 \xrightarrow{1} 113 | 5 | 7 | 9$ is odd
 $w_2 = 1235670$

input is read from left to right



M
Accepts only
Even numbers

Transition diagram/Graph

- ① $\Sigma = \{0, 1, 2, \dots, 9\}$
 ② $Q = \{q_0, q_1, q_2\}$ states

③ Start state q_0

④ Final state: $\{q_1\}$

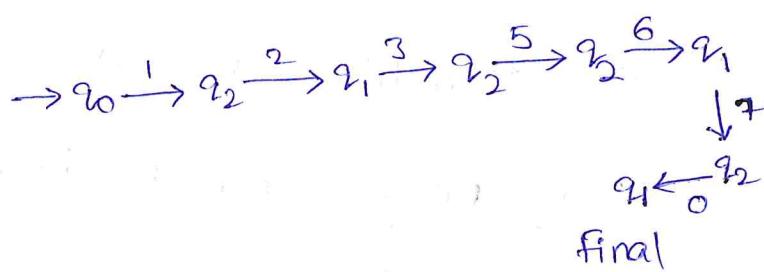
↑ zero or more final states

⑤ Transition junction

$$w_1 = 123$$

$\rightarrow q_0 \xrightarrow{1} q_2 \xrightarrow{2} q_1 \xrightarrow{3} q_2$ does not accept w_1

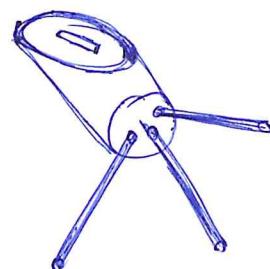
$$w_2 = 1235670$$



w_2 is accepted by DFA

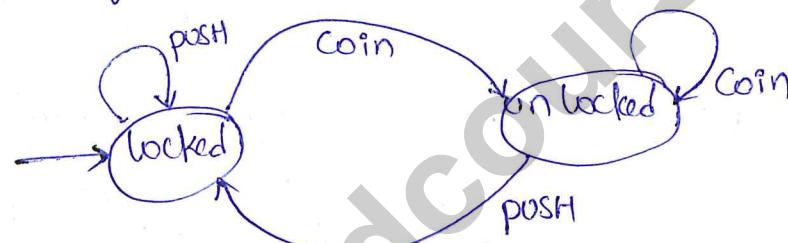
- ① 21 invalid
 ② 12346 valid - Even
 ③ 2 valid - Even

Eg 2:- Turnstile machine: we can find in Metro stations / Airports etc.



After inserting a coin, we are allowed to come out.

State diagram of Turnstile machine



$$Q = \{ \text{locked}, \text{unlocked} \}$$

$$\Sigma = \{ \text{push, coin} \}$$

$$q_0 \text{ (start state)} = \text{locked}$$

$$F = \{ \}$$

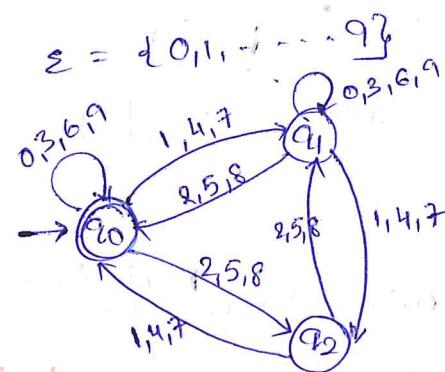
δ : Transition Function

δ	coin	push
locked	unlocked	locked
unlocked	unlocked	locked

Transition Table

Eg 3:

Design a FSM to determine if a decimal number is divisible by 3.



Div by 3

$$\Sigma = \{ 0, 1, \dots, 9 \}$$

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

$$q_0 = q_0$$

$$F = \{ q_0 \}$$

δ : Transition Function

① 99

 $\rightarrow q_0 \xrightarrow{q} q_0 \xrightarrow{q} q_0$
 Final Accepted and divisible by 3.

 ② 141 $\Rightarrow 3 | 141 | 47$

$$\begin{array}{r} 12 \\ \hline 21 \\ 21 \\ \hline 0 \end{array}$$
 $\rightarrow q_0 \xrightarrow{1} q_1 \xrightarrow{4} q_2 \xrightarrow{1} q_0$

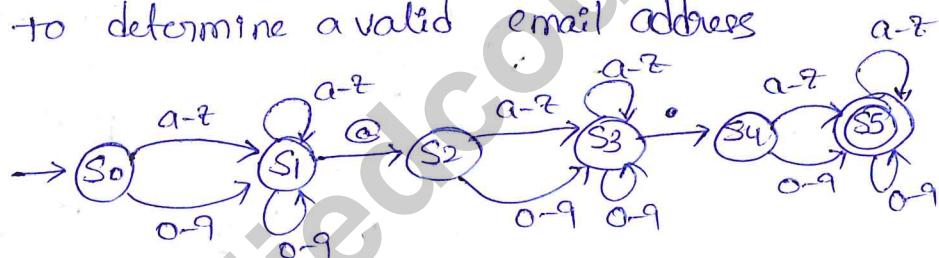
③ 256

 $\rightarrow q_0 \xrightarrow{2} q_2 \xrightarrow{5} q_1 \xrightarrow{6} q_1$

$$\begin{array}{r} 24 \\ \hline 16 \\ 15 \\ \hline 1 \end{array}$$

 Non-final
Eg 4:

DFA to determine a valid email address



[Source: stack overflow.com]

The pattern is

a123 @ gmail.com

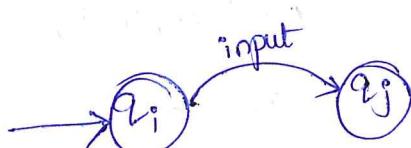
Representation of a DFA:

① Transition Diagram

② Transition Table \Rightarrow

input

		input									
		0 1 2 3 4 5 6 7 8 9									
		q0 q1 q2 q3 q4 q5 q6 q7 q8 q9									
		q0	q1	q2	q3	q4	q5	q6	q7	q8	q9



- * FSM without a final state
- * Valid & invalid inputs
- * $w \in \Sigma^*$

$$(q_i, x) \rightarrow (q_j)$$

Mathematical Definition of DFA:

5-tuple | quintuple

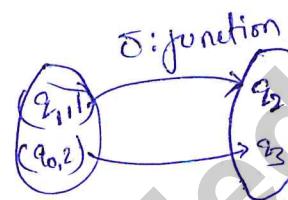
$$M = (\Sigma, Q, q_0, \delta, F)$$

↓ ↓ ↓ ↑
input set initial set of final
alphabet state state
of state
(Finite)

δ : Transition Function

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

$(q_i, 1) \downarrow$ Next State

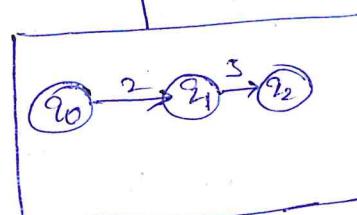


Computation Architecture of FSM:

FSM is less
powerful than
a computer

Input String
 $\rightarrow [2 \mid 3 \mid 1 \mid 4 \mid 6]$ Input Tape

Tape Header (scan/read)

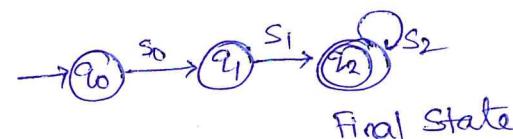


Finite Control
Unit

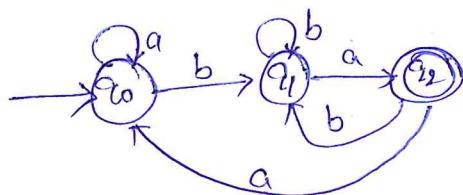
Language of a finite Automata:

$$L(FA) = \{ w \in \Sigma^* \mid w \text{ is accepted by FA} \}$$

$$\text{Let } w = s_0 s_1 s_2$$



(12)



$w_0 = \epsilon$

$w_1 = a^i x$

$L = \{ba, aba, aaba, bba, bbba, \dots\}$

$w_2 = b^i x$

$L = \{w \in \Sigma^* \mid w = xba \text{ where } x \in \Sigma^*\}$

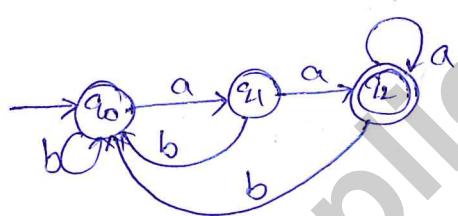
$w_3 = \underline{ba} \checkmark$

$w_4 = \underline{abba} \checkmark$

$w_5 = aabbba \checkmark$

$w_6 = abbabx$

$w_7 = ababa \checkmark$

Eg 2:

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

$w_1 = ab^i x \quad q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0$

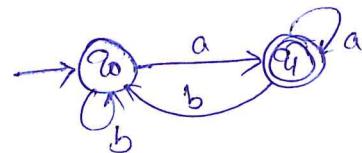
$w_2 = aa \checkmark \quad q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \text{ Final}$

$w_3 = aba \checkmark$

$w_4 = abaa \checkmark \quad L = \{w \in \Sigma^* \mid w = zaa, z \in \Sigma^*\}$

$w_5 = abb \checkmark$

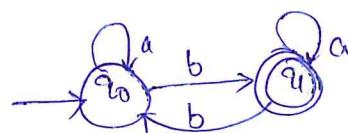
$w_6 = abbaa \checkmark \quad L = \{w = zaa \mid z \in \Sigma^*\}$

Eg: 3

Ph: 844-844-0102

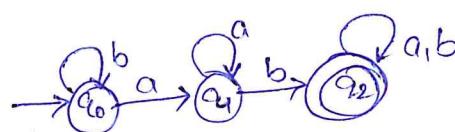
$$L = \{ a, aa, \dots \}$$

$$= \{ xa \mid x \in \Sigma^* \}$$

Eg 4

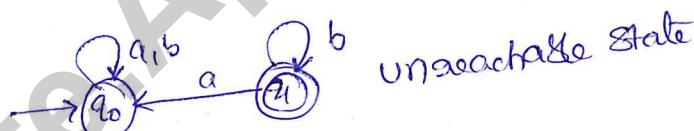
$$L = \{ b, ba, bbb, \dots \}$$

↳ accept words/strings that have odd number of 'b's

Eg 5:

$$L = \{ ab, bab, aab, aba, abb, \dots \}$$

$$= \{ xaby \mid x, y \in \Sigma^* \}$$

Unreachable State:

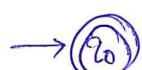
$$L = \{ \} = \emptyset$$



$$L = \{ \} = \emptyset$$

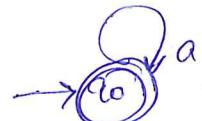


Ques



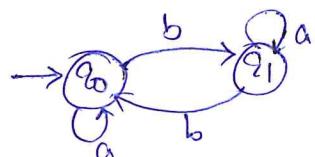
$$L = \{ \epsilon \}$$

Ph: 844-844-0102



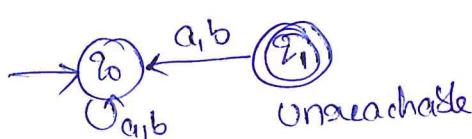
$$L = \{ \epsilon, a, aa, \dots \}$$

Eg 2:-



$$\text{No final staty } L = \{ \} = \emptyset$$

Eg 3:-



$$L = \{ \} = \emptyset$$

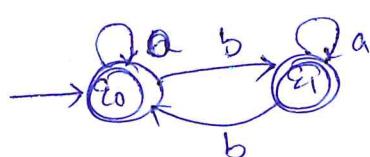
Eg 4:-



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

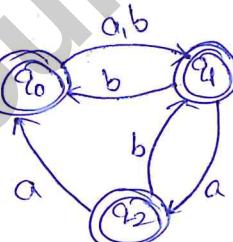
Minimal DFA

$$L = \Sigma^*$$



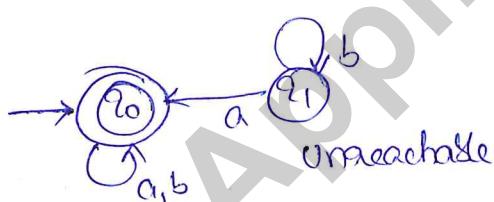
$$L = \Sigma^*$$

Eg 6:-



$$L = \Sigma^*$$

Eg 7:-



Unreachable

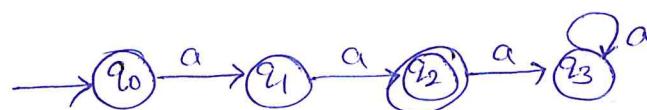
→ Every DFA accepts exactly one language

→ L: Multiple DFA that accepts all words in L

→ Min DFA exactly one

↓
min # staty

Eg 1: $L = \{aay\}$ $\Sigma = \{a\}$



dead state: No way to come out from the state

Ex

ax

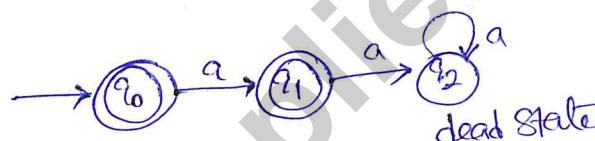
aav

aaax

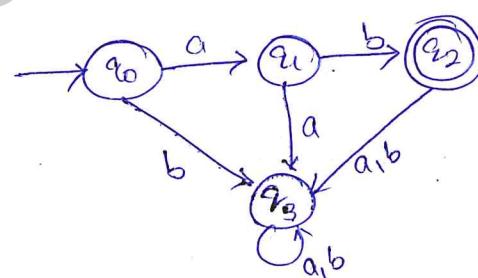
aaaax

DFA: $\delta: Q \times \Sigma \rightarrow Q$ we need to define the transition for each input symbol at every state.

Eg 2: $L = \{\epsilon, a^3\}$ $\Sigma = \{a\}$



Eg 3: $L = \{aby\}$ $\Sigma = \{a, b\}$

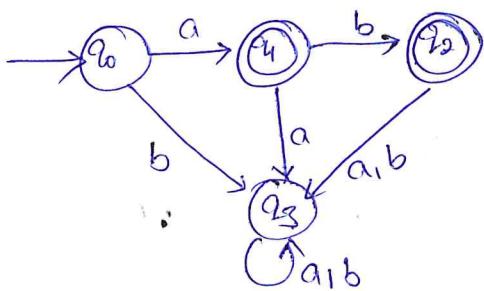


q_3 is a dead state.

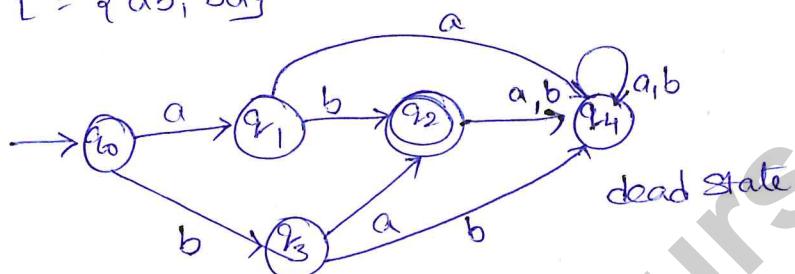
DFA will accept only one string "ab"

let $w = aba$

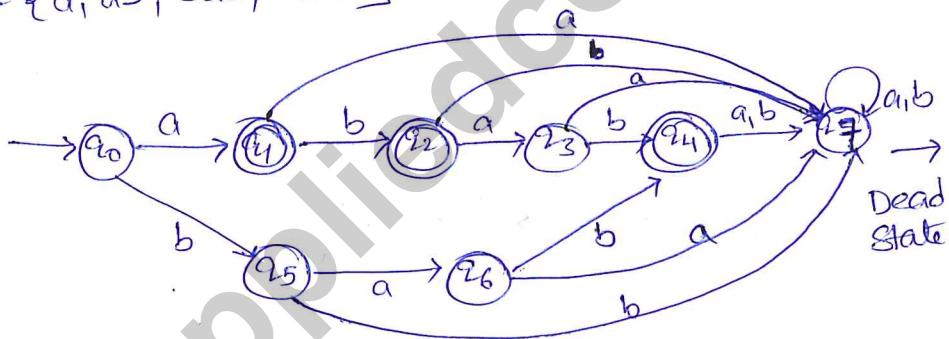
$\Rightarrow q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_3$ invalid | Not accepted.
String

Eg 5:

$$L = \{ab, b\bar{a}\}$$

Eg 6:

$$L = \{a, ab, bab, abab\}$$



Regular Languages & Finite Languages:

$$L = \{a^n b^n \mid n \geq 1\} \rightarrow \text{Non-regular}$$

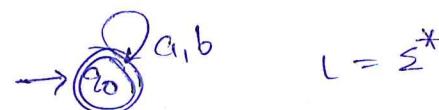
ab, aabb, aaabbb, ...

Regular Language is a Language for which DFA that accepts the Language.

All Finite Languages are Regular Languages, as we can able to construct FA for all Finite Languages.

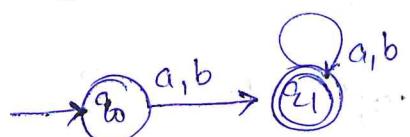
APP COURSE Construction of FA for Non-finite Languages: Ph. 14 = not-finite (17)

Eg 1: All strings of a's & b's including ϵ over $\Sigma = \{a, b\}$



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

Eg 2: All strings of a's & b's excluding ϵ over $\Sigma = \{a, b\}$

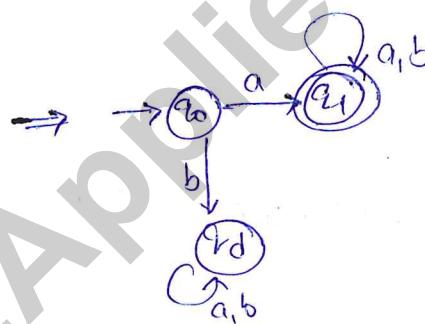


$$L = \{ a, b, aa, ab, ba, bb, \dots \}$$

Eg 3: Strings that start with 'a', $\Sigma = \{a, b\}$

$$L = \{ ax | x \in \Sigma^* \}$$

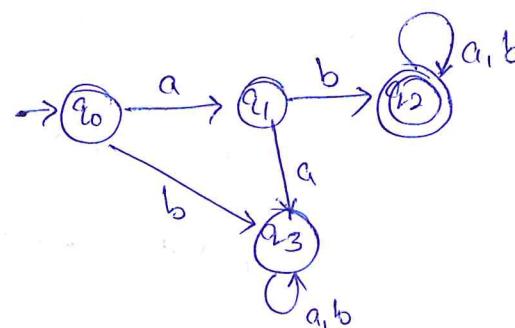
$$L = \{ a, aa, ab, aaa, abb, \dots \}$$



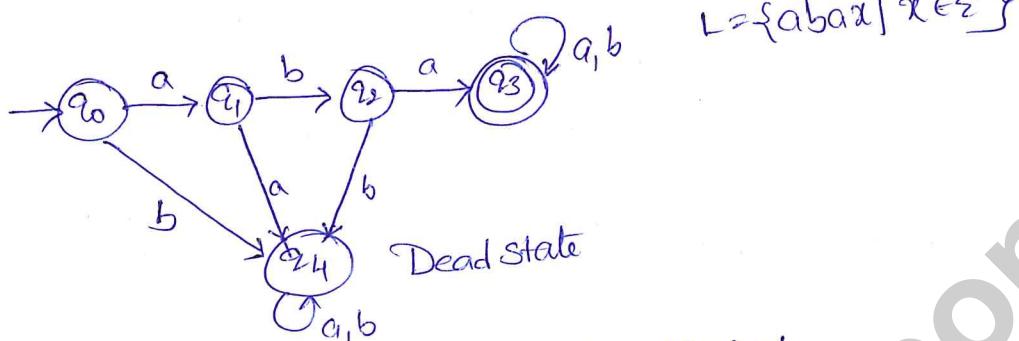
Eg 4: Start with 'ab' & $\Sigma = \{a, b\}$

$$L = \{ abx | x \in \Sigma^* \}$$

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



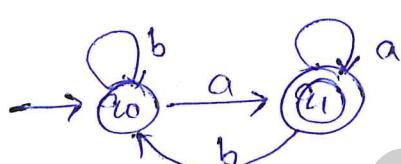
Eg 5: strings that start with 'aba' & $\Sigma = \{a, b\}$



Eg 6: Min FA, that accepts strings ends with 'a'

$$\Rightarrow L = \{xa \mid x \in \Sigma^*\}$$

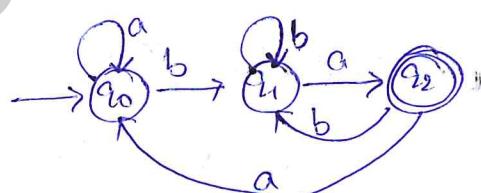
$$= \{a, aa, ba, aaa, aba, \dots\}$$



Eg 7: strings that ends with 'ba'

$$\Rightarrow L = \{xba \mid x \in (a,b)^*\}$$

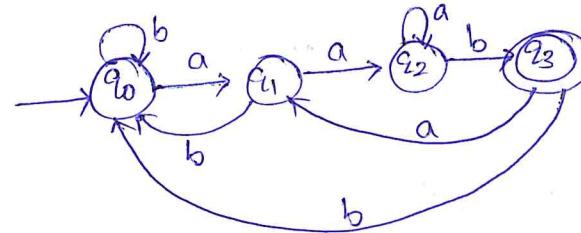
$$= \{ba, aba, bba, aaba, abba, \dots\}$$



Eg 8: strings that ends with 'aab'

$$L = \{xaab \mid x \in \Sigma^*\}$$

$$= \{aab, aaab, baab, \dots\}$$

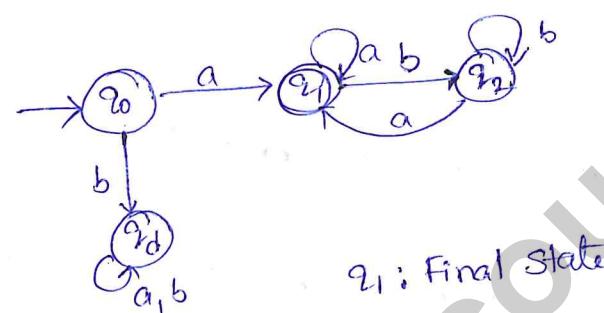


aabbbaab

Eg 9: Strings that start & end in 'a' $\Sigma = \{a, b\}$

$$L = \{ axa | x \in \Sigma^* \}$$

$$= \{ a, aa, aba, aaa, \dots \}$$

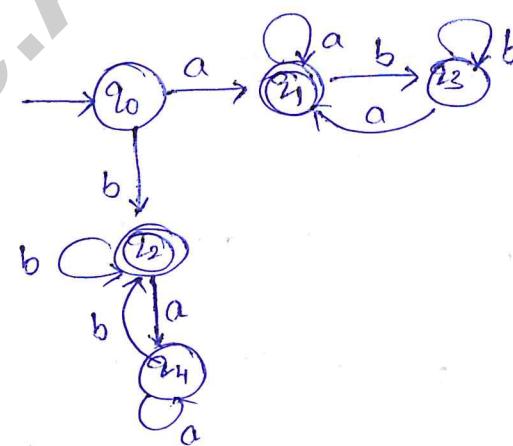


q_1 : Final State

Eg 10: Strings that start & end with same Symbol

$$L = \{ axa, byb | x, y \in \Sigma^* \}$$

$$= \{ a, b, aaa, aba, bbb, bab, \dots \}$$

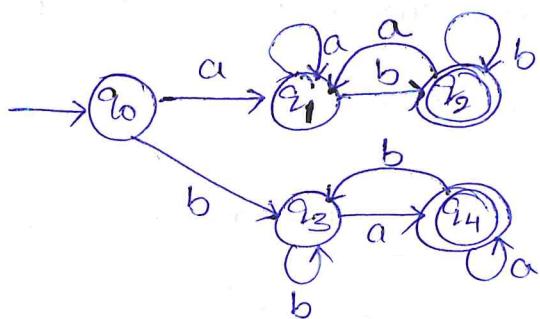


⑩

Eg 11: Strings that start & end in different symbols
P.I.: 844-844-0102

$$L = \{axb, bxa \mid x \in \Sigma^*\}$$

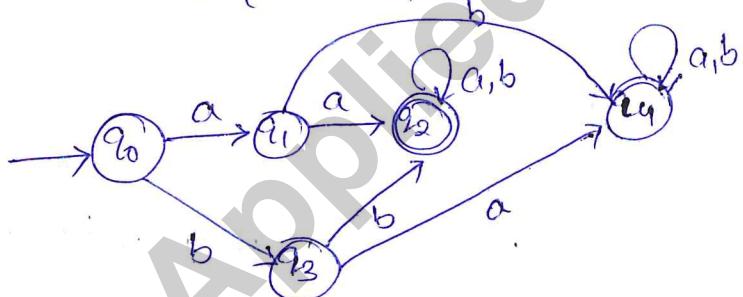
$$= \{ ab, ba, aab, abb, baa, bba, \dots \}$$



Eg 12: Strings that start with aa(8) bb

$$L = \{ aax, bba \mid x \in \Sigma^*\}$$

$$= \{ aa, bb, aab, aaa, bba, bbb, \dots \}$$



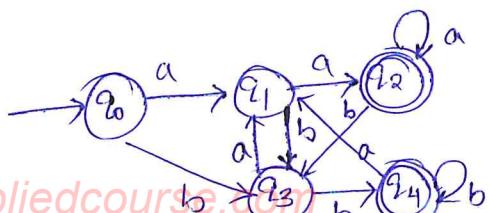
q_2 : Final state

q_4 : Dead state

Eg 13: Strings that end with aa(8) bb

$$L = \{ xaa, xbb \mid x \in \Sigma^*\}$$

$$= \{ aa, bb, aaa, bbb, baa, abb, \dots \}$$



bbaa

$q_0 \xrightarrow{b} q_3 \xrightarrow{a} q_4 \xrightarrow{a} q_1 \xrightarrow{a} q_2$

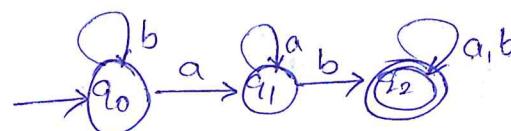
Eg 14:

Strings that contain 'ab' as a Substring.

Ph. 844-844-0102

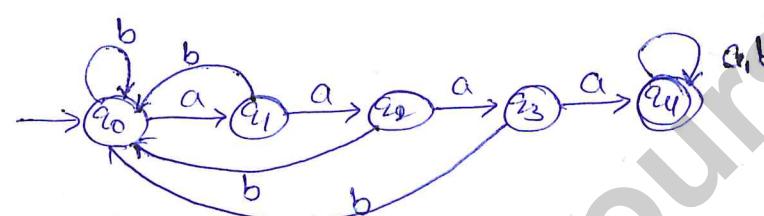
$$L = \{ xaby \mid x, y \in \{a, b\}^* \}$$

$$= \{ ab, aab, aba, \dots \}$$

Eg 15:

strings that contain 'aaaa' as Substring

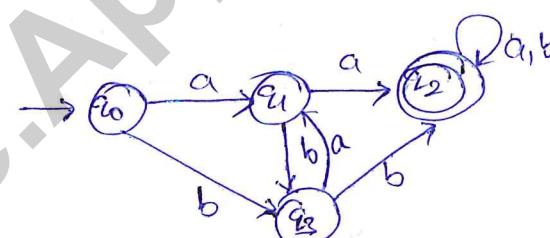
$$L = \{ xaaaaay \mid x, y \in \Sigma^* \}$$

Eg 16:

contains aa or bb as a Substring

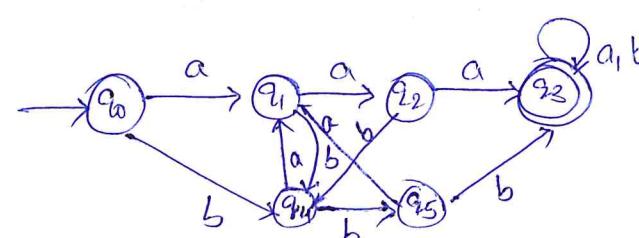
$$L = \{ xaby, xbbby \mid x, y \in \Sigma^* \}$$

$$= \{ aa, bb, aaa, bbb, \dots \}$$

Eg 17:

contains aaa (or) bbb as a Substring

$$L = \{ xaaaay, xbbbby \mid x \in \Sigma^* \}$$

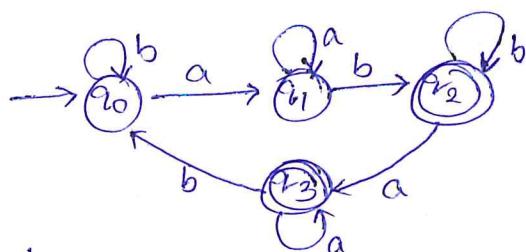


Contains odd occurrence of substring 'ab'

Ph: 844-844-0102



1, 3, 5, 7, ...

 $L = \{ ab, aab, aba, ababab, \dots \}$ let $W = ab$ $\rightarrow q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2$
 $W = aba$
 $\rightarrow q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_3$

F

 $W = abababa$
 $\rightarrow q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_3 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_3$

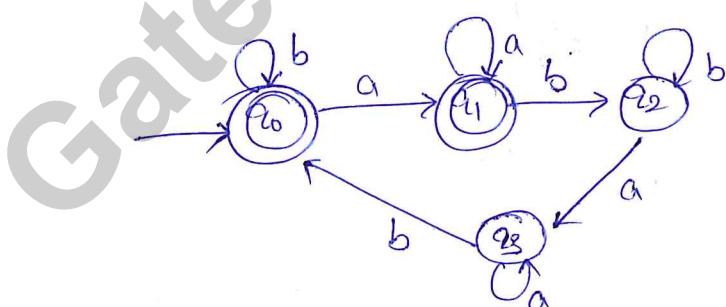
final

Eg 19:

Contains even number of occurrences of substring 'ab'



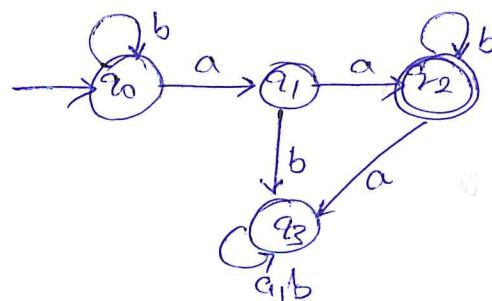
0, 2, 4, 6, ...

 $L = \{ \epsilon, abab, ababab, \dots \}$
aaaa, bbbb, bbbb q_0, q_1, q_3 are final state

Eg 20: $b^* a a b^*$ all strings containing 2 a's, that are exactly consecutive

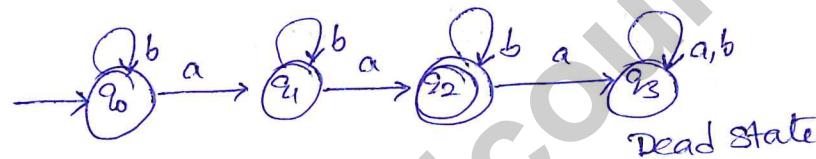
Ph: 844-844-0102

$$L = \{ aa, baa, aab, \dots \}$$



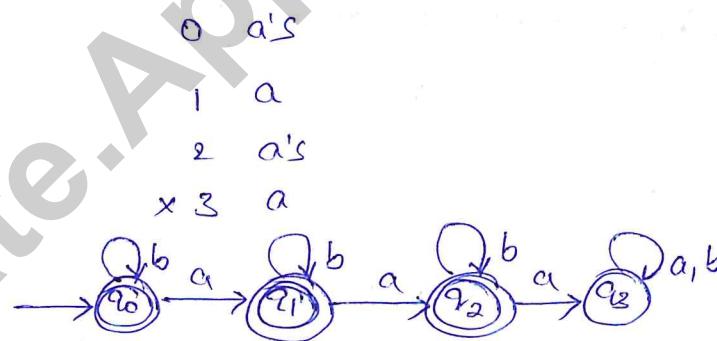
Eg 21: exactly 2 a's $b^* a b^* a b^*$

$$L = \{ aa, baa, aba, \dots \}$$



Eg 22: almost two a's

$$\{\epsilon, a, aa, \dots\}$$

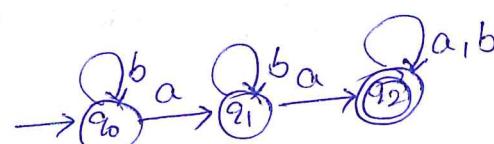


Eg 23: at least 2 a's

$$2 a's$$

$$3 a's$$

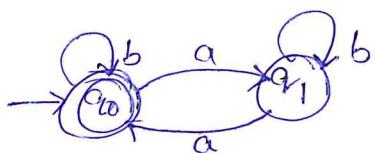
$$4 a's$$



(a) Even number of a's

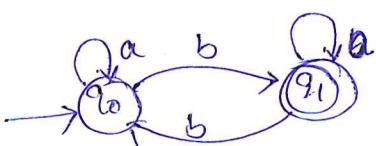
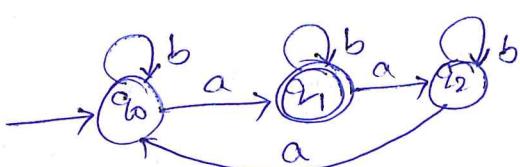
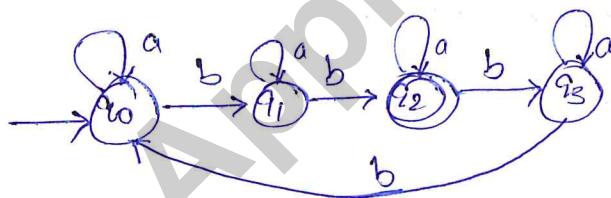
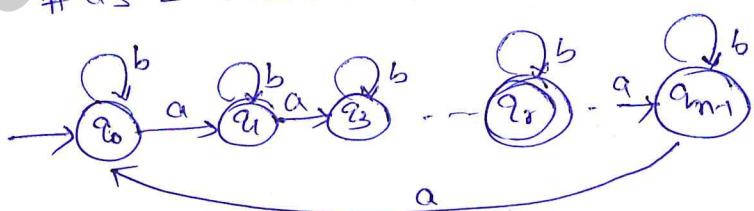


0, 2, 4, 6, ...

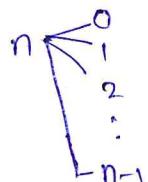


(b) Even number of b's

1, 3, 5, ...

(c) # a's $\equiv 1 \pmod{3}$ (Modulo arithmetic) $= 1, 4, 7, 10, \dots$ (d) # b's $\equiv 2 \pmod{4}$ $= 2, 6, 10, 14, \dots$ Generalization:# a's $\equiv r \pmod{n} = |\text{WA}|$ 

n-states

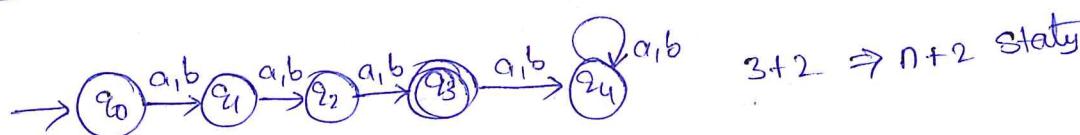


Construct min FA that accepts all strings of $\Sigma = \{a, b\}$

Ph: 844-844-0102

where

(a) $|w| = 3$

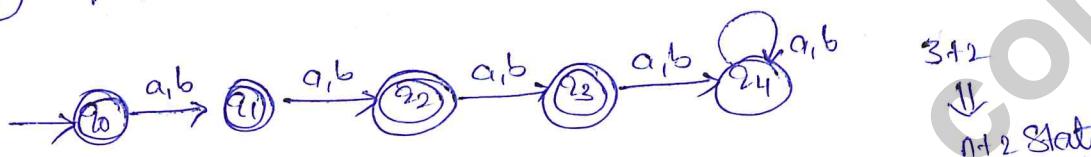


(b) $|w| \leq 3$

(c) $|w| \geq 3$

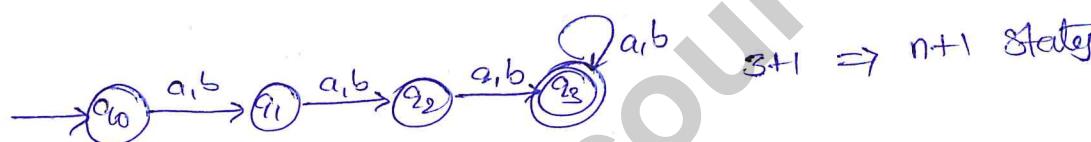
$3+2 \Rightarrow n+2$ states

(b) $|w| \leq 3$



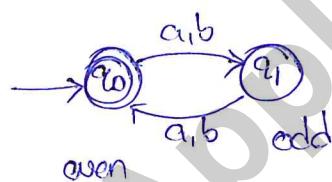
$3+2 \Rightarrow n+2$ States

(c) $|w| \geq 3$

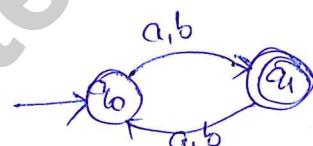


$3+1 \Rightarrow n+1$ states

(d) $|w| = 0 \pmod{2}$ $0, 2, 4, \dots$



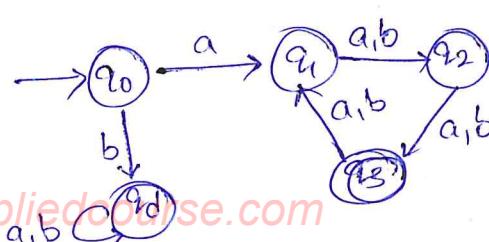
(e) $|w| = 1 \pmod{2}$ odd



Eg 26: Min FA of $a^l b^k$'s

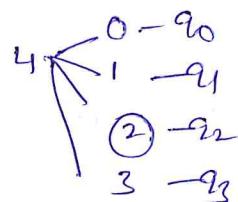
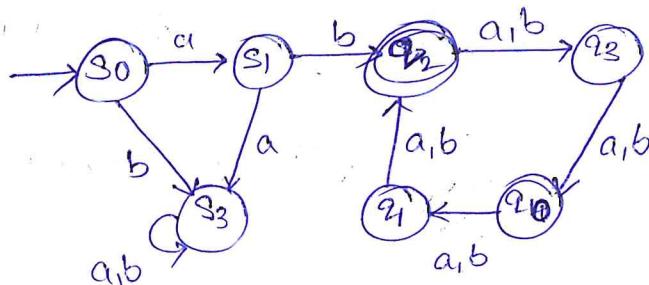
(a) Start with 'a' + $|w| = 0 \pmod{3}$

$$= (0, 3, 6, 9, \dots)$$

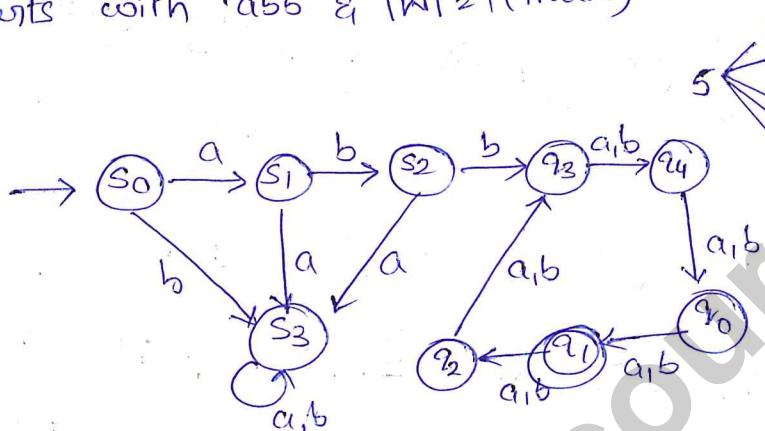


(b) starts with 'ab' & $|w| \equiv 2 \pmod{4}$

Ph: 844-844-0102



(c) starts with 'abb' & $|w| \equiv 1 \pmod{5}$



q_1 is Final State

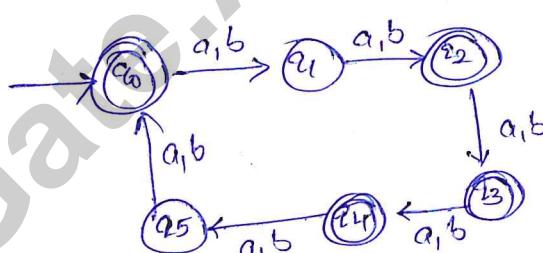
(d) $|w| \equiv 0 \pmod{2}$ & $0 \pmod{3}$

$$0, 3, 6, 9, \dots$$

$$0, 2, 4, 6, \dots$$

$$\Rightarrow |w| \equiv 0, 2, 3, 4, 6, 8, 9, 10, \dots$$

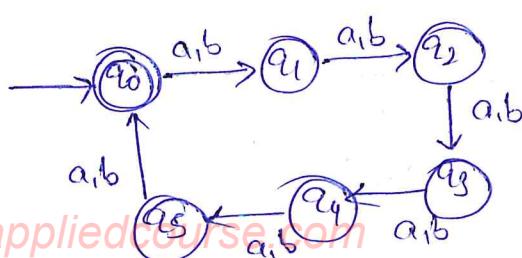
$$\text{LCM}(2, 3) = 6$$



(e) $|w| \equiv 0 \pmod{2}$ and $0 \pmod{3}$

$$\text{LCM}(2, 3) = 6$$

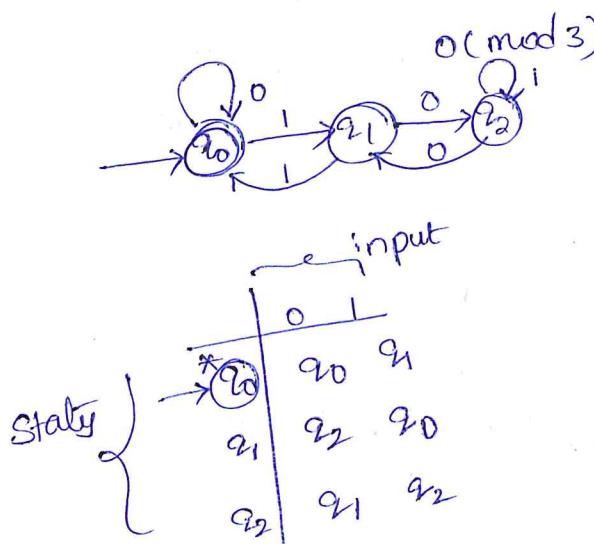
$$6, 12, 18, 24, \dots$$



DFA's on binary input: $\Sigma = \{0,1\}$

Ph: 844-844-0102

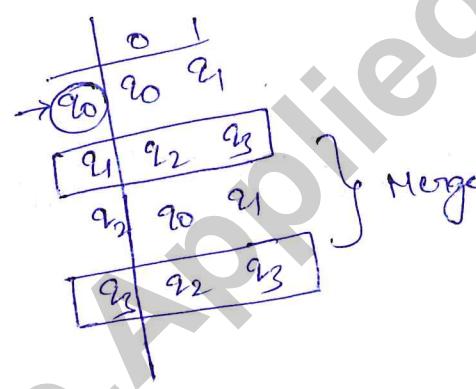
- ① Min DFA that accepts binary strings whose integer value is $0 \pmod 3$



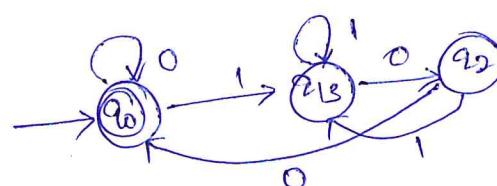
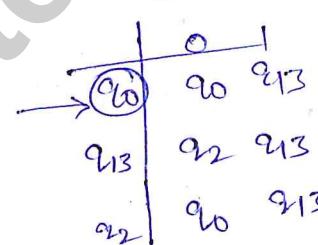
Binary	Decimal	Rem
✓ 000	0	0
x 001	1	1
x 010	2	2
✓ 011	3	0
x 100	4	1
		:

- ② Divisible by $n = 0 \pmod n$

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



bin	dec	rem
0	0	0
1	1	1
10	2	2
11	3	3
100	4	0
101	5	1
110	6	2
111	7	3



Congruent to 1 mod 6 $\equiv 1 \pmod{6}$

Even

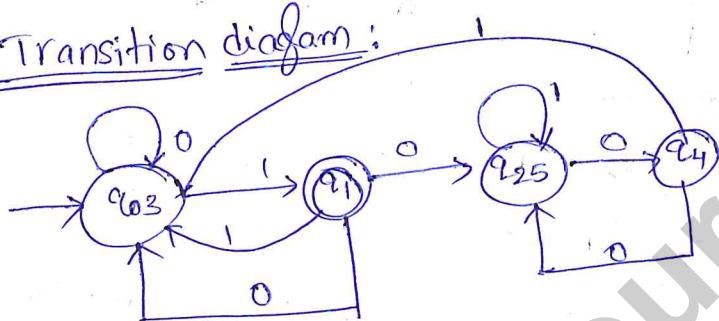
Ph: 844-844-0102

	0	1
0	q ₀	q ₁
1	q ₂	q ₃
2	q ₄	q ₅
3	q ₆	q ₇
4	q ₈	q ₉
5	q ₁₀	q ₁₁

22,5 } \rightarrow { q₀, q₁, q₂, q₃, q₄, q₅, q₆, q₇, q₈, q₉, q₁₀, q₁₁ } \Rightarrow 20,3 \rightarrow { q₀₃, q₁₃, q₂₃, q₃₃, q₄₃, q₅₃, q₆₃, q₇₃, q₈₃, q₉₃, q₁₀₃, q₁₁₃ }

	0	1
0	q ₀₃	q ₁₃
1	q ₂₃	q ₃₃
2	q ₄₃	q ₅₃
3	q ₆₃	q ₇₃
4	q ₈₃	q ₉₃
5	q ₁₀₃	q ₁₁₃

Transition diagram:



$$1 \pmod{6} = \{1, 7, 13, \dots\}$$

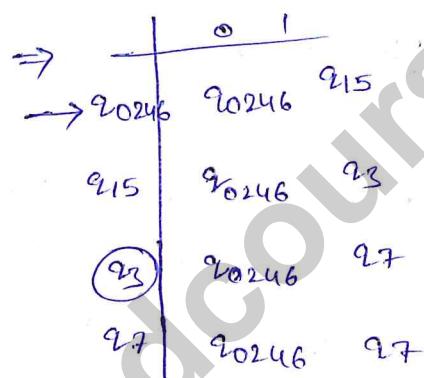
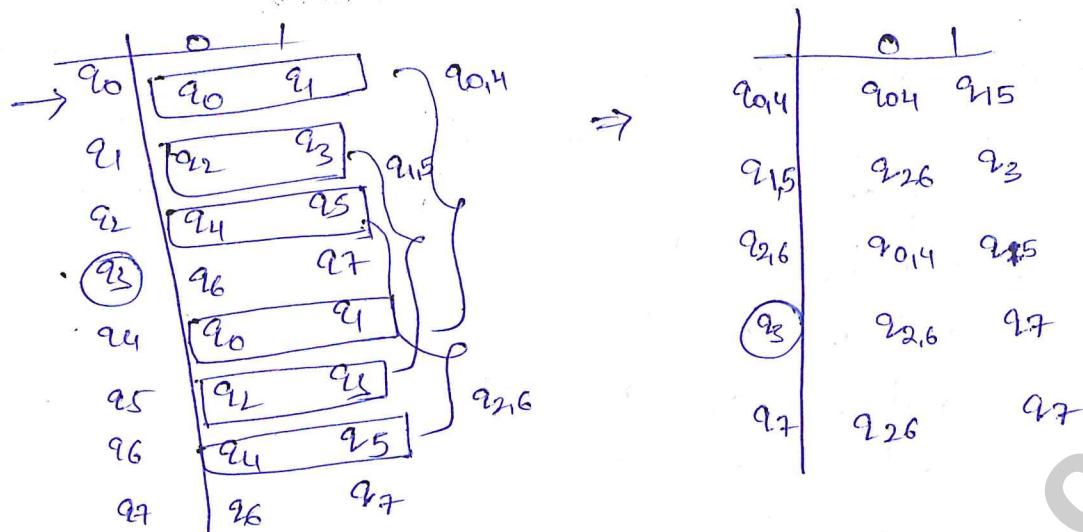
$$\text{Let } n=13 \\ = 1101$$

$\rightarrow q_{03} \xrightarrow{1} q_1 \xrightarrow{1} q_{03} \xrightarrow{0} q_{03} \xrightarrow{1} q_1$ Final state

(4) $\equiv 2 \pmod{7}$ \rightarrow odd

	0	1
0	q ₀	q ₁
1	q ₂	q ₃
2	q ₄	q ₅
3	q ₆	q ₇
4	q ₈	q ₉
5	q ₁₀	q ₁₁
6	q ₁₂	q ₁₃

No two rows are same
 \Rightarrow No merging possible



Generalization:

Bin number $\equiv r \pmod{n}$

a) n is odd \rightarrow minimal FA containing n states

b) $n = 2^k \rightarrow k+1$ states

c) $n = \text{even and } n \neq 2^k$, then $n = (2^k) \times m$,

where m is prime factor of n . Then number of states will be $(k+m)$

$\equiv 1 \pmod{6} \Rightarrow (2^1) \times 3 \Rightarrow k=1, m=3$

number of states $k+m = 1+3=4$ states

(30)

Trinary Number:

$$\Sigma = \{0, 1, 2\}$$

Ph: 844-844-0102

$$(210)_3$$

① Trinary number $\equiv 2 \pmod{4}$

$$\Rightarrow 3^0 \times 0 + 3^1 \times 1 + 3^2 \times 2$$

$$\Rightarrow 0 + 3 + 18$$

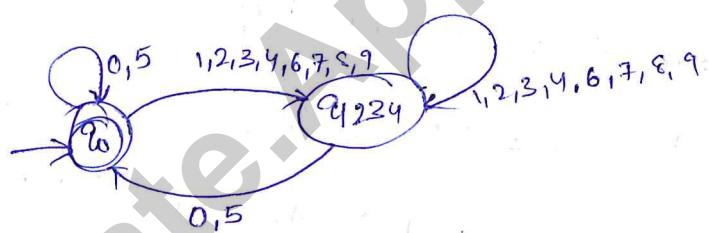
$$\Rightarrow 21$$

	0	1	2
q_0	q_0	q_1	q_2
q_1	q_3	q_0	q_1
q_2	q_2	q_3	q_0
q_3	q_1	q_2	q_3

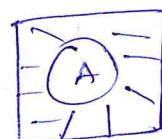
Decimal number (base 10) $\equiv 0 \pmod{5}$

	0	1	2	3	4	5	6	7	8	9
q_0	q_0	q_1	q_2	q_3	q_4	q_0	q_1	q_2	q_3	q_4
q_1	q_0	q_1	q_2	q_3	q_4	q_0	q_1	q_2	q_3	q_4
q_2	:	:	:							
q_3	:	:	:							
q_4	,	,	,							

Same Transitions

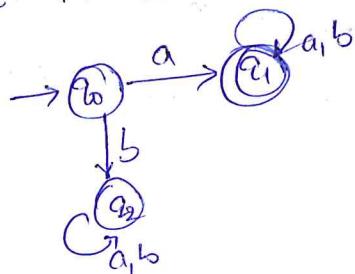


Complement of a FA:



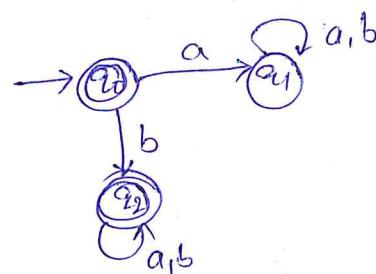
$$A^c = A' = U - A$$

① let M is



$$M^T \cap M^C \Rightarrow L(M^T) = \Sigma^* - L(M)$$

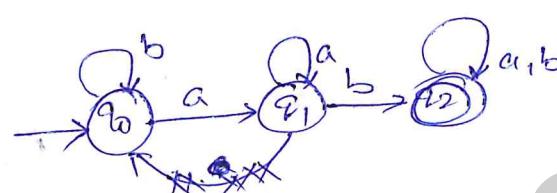
Mail: gatecse@appliedcourse.com



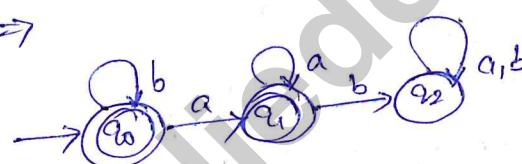
All the non-final states become the final states and final will become Non-final states.

Eg2: Construct a min FA that accepts all the strings that don't contain 'ab' as a substring.

$$L = \{w \mid w = xy \text{ where } x \in \Sigma^*, y \in \Sigma^*\}$$

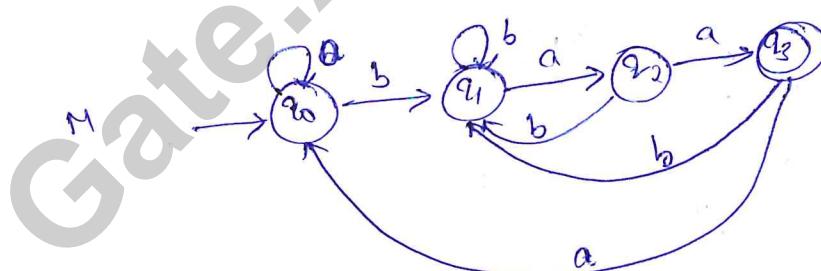


$M_1 \Rightarrow$

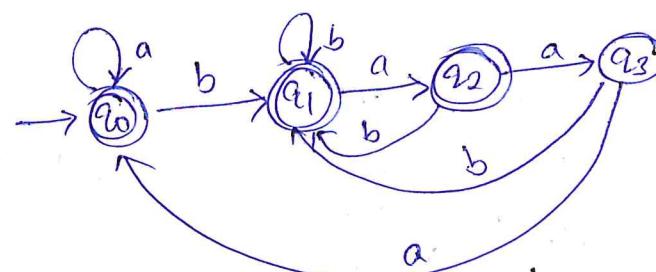


Eg3: Accepts strings that do not end with 'baa'

$$L = \{wbaa \mid w \in \Sigma^*\}$$



M'



3

Compound Automata: Unions & Intersections

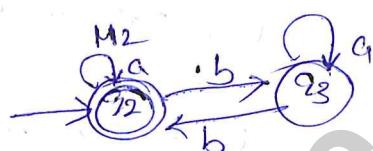
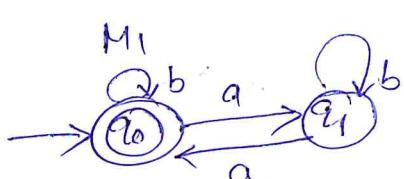
Ph: 844-844-0102

$$M_1, L_1 \quad L_1 \cup L_2 \cup L_3 = L_1 \rightarrow M_1$$

$$M_2, L_2 \quad L_1 \cap L_2 \cap L_3 = L_2 \rightarrow M_2$$

$$M_3, L_3$$

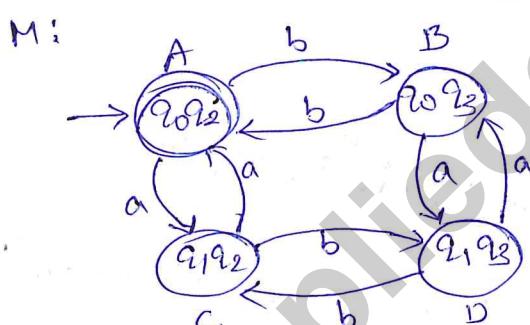
- ① $\Sigma = \{a, b\}$ Number of a's even and number of b's even



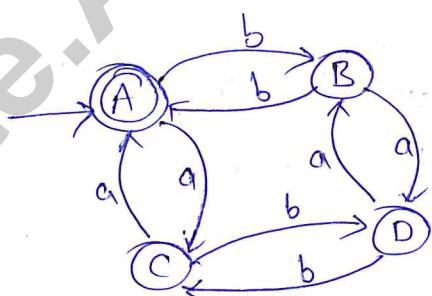
$$\Rightarrow L_1 \cap L_2 = L$$



Apply
Cartesian
product



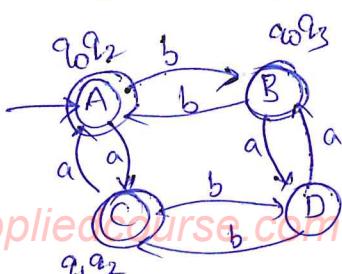
q_0, q_2 are final states in M_1, M_2 respectively



- ② # a's even $\frac{L_1}{L_1} \cup \frac{\# b's even}{L_2}$ Final State

$$L = L_1 \cup L_2$$

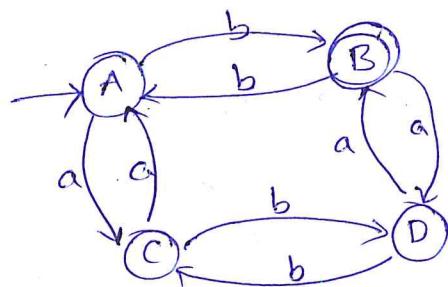
(M)



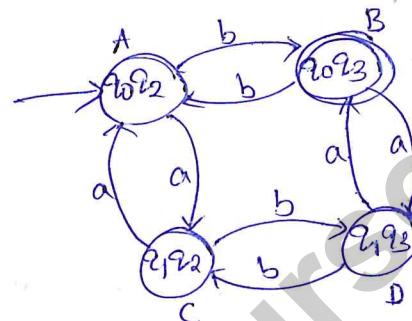
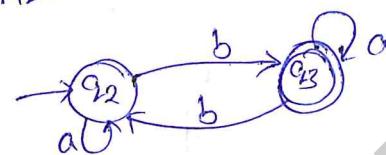
Mail: gatecse@appliedcourse.com

a's even and # b's is odd.

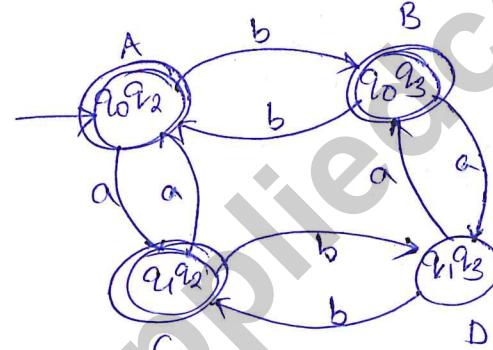
Ph: 844-844-0102



M2

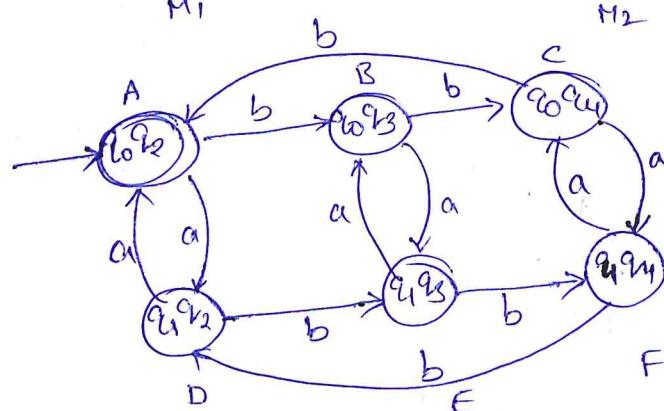
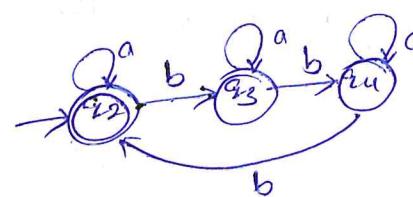
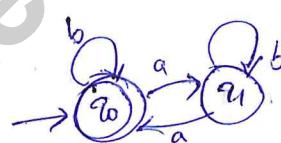


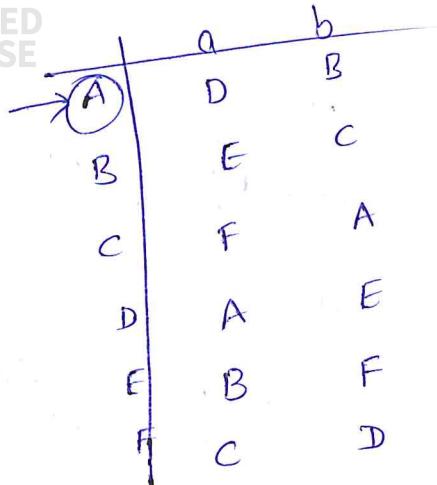
(4) # a's even & # b's is odd



A, B, C are Final States

(5) # a's $\equiv 0 \pmod{2}$ and # b's $\equiv 0 \pmod{3}$



(1) $A \not\in \{B, C, D, E, F, Y\}$ (2) $C, D \not\in X$

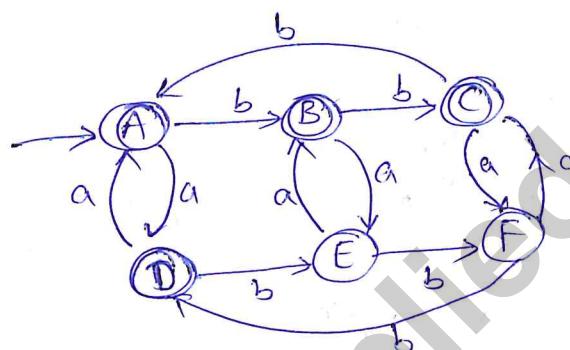
FA

AE

(3) $B, C \not\in X$

EC

FA

(4) $E, F \not\in X$
 $\begin{matrix} B \\ C \\ D \end{matrix}$
(6) # of a 's $\equiv 0 \pmod{2}$ (7) # of b 's $\equiv 0 \pmod{3}$ Counting DFA :-(1) # of 2-state DFA with a designated initial statethat can be constructed using $\Sigma = \{a, b\}$ (1) Final States : 2^2

$0, 1, 2 \rightarrow$ both
 $q_0, q_1 \rightarrow$ one final state

no final state may be final

(2)

	a	b
q_0	2	2
q_1	2	2

 2^4 possibilitiesMail: gatecse@appliedcourse.com $2^2 \times 2^4 = 2^6 = 64$ (possible DFA's)

(55) # of 3-state DFA with a designated initial state that can be constructed using $\Sigma = \{a, b\}$ Ph: 844-844-0102



① Final State: $\{q_0, q_1, q_2, q_0q_1, q_1q_2, q_0q_2, q_0q_1q_2\}$

2^3

Σ	a	b
$\rightarrow q_0$	$\underline{\underline{z}}$	$\underline{\underline{z}}$
q_1	$\underline{\underline{z}}$	$\underline{\underline{z}}$
q_2	$\underline{\underline{z}}$	$\underline{\underline{z}}$

$\Rightarrow 3^6$

Total # of possible DFA's are $2^3 \times 3^6 = 8 \times 729 = 5832$

Generalization:

$$|\Sigma| = m \quad \# \text{State} = |\mathcal{Q}| = n$$

initial state is
designated

① Final State: 2^n

② δ: $m \times n$ cells and each cell can fill with n ways.

$$n^{m \times n}$$

$$\# \text{DFA's} \Rightarrow 2^n \times n^{m \times n}$$

③ # 2 State DFA's over $\Sigma = \{a, b\}$ that accept φ



0 Final State

1 Final State



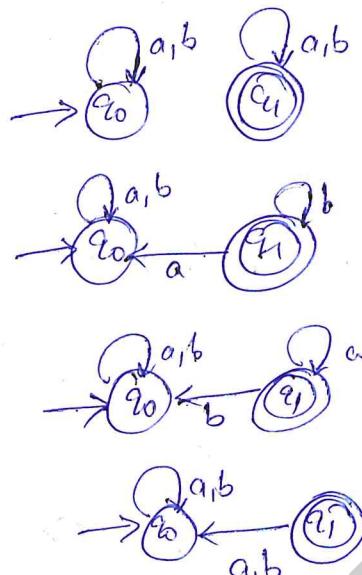
 $\delta:$

q_0	a, b	- -
q_1	- -	- -

$$2^4 = 16$$

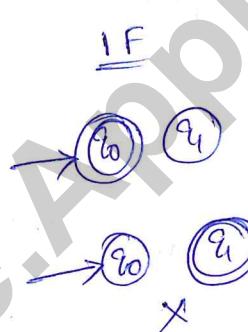
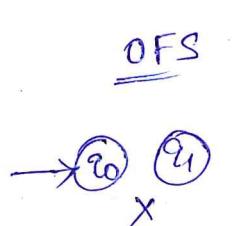
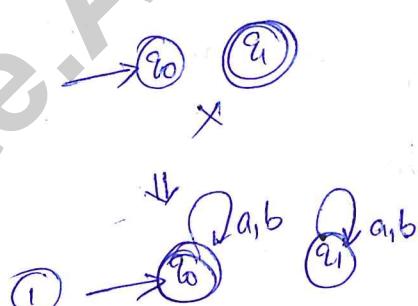


16



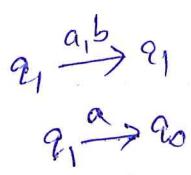
\Rightarrow Total # of DFA's possible
 $= 16 + 4 = 20$

④ # of 2 state DFAs over $\Sigma = \{a, b\}$ that accept ϵ^*

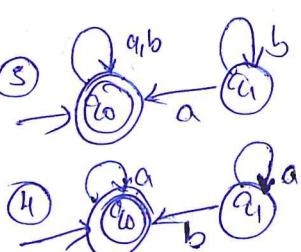
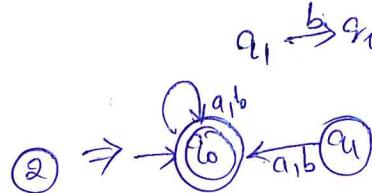
2F

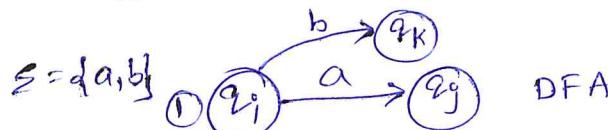
0:	a	b
q_0	$\frac{2}{2}$	$\frac{2}{2}$
q_1	$\frac{2}{2}$	$\frac{2}{2}$

$$2^4 = 16 \text{ DFA's}$$



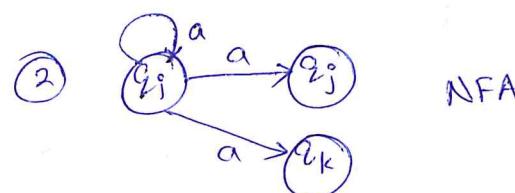
$$\text{Total} = 16 + 4 = 20$$



Non-deterministic FA (NFA)

DFA

$Q \times \Sigma \rightarrow Q$



NFA

5-tuple

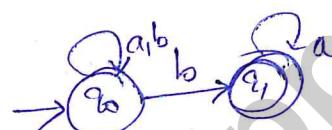
$M = (Q, \Sigma, \delta, q_0, F)$

 Q : set of States Σ : input Alphabet δ : $Q \times \Sigma \rightarrow 2^Q$ (power set of Q)set of all subsets of Q

$Q = \{q_0, q_1\}$

$|P(Q)| = 2^n$

$P(Q) = 2^Q = \{\emptyset, q_0, q_1, \{q_0, q_1\}\}$

Eg 1:-

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

$Q = \{q_0, q_1\}$

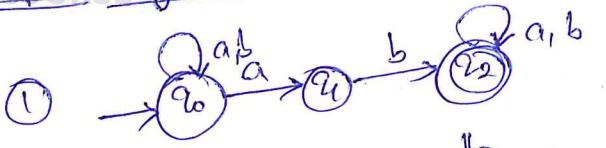
$P(Q) = \{\emptyset, q_0, q_1, \{q_0, q_1\}\}$

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

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

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

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

Acceptance by NFA:

$$W_1 = ab \quad q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_3$$

$$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2$$

If any transition path leads to a final state we accept the word.

$$W_2 = aa$$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0$$

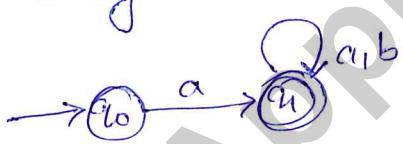
$$q_0 \xrightarrow{a} q_1 \times$$

Try all possible transition paths

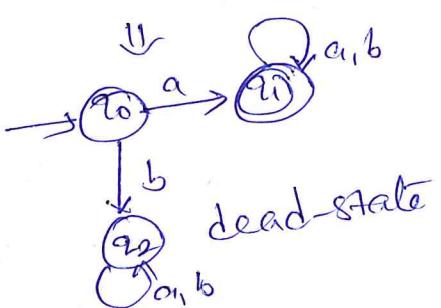
②

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

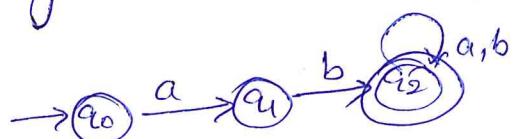
a) strings that start with 'a'



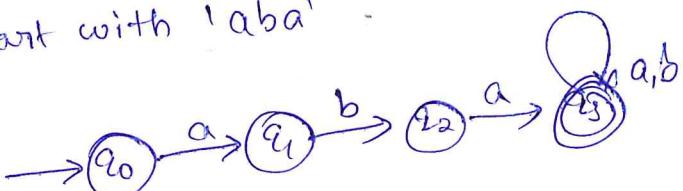
not DFA $q_0 \xrightarrow{b} \emptyset$



b) strings that start with 'ab'

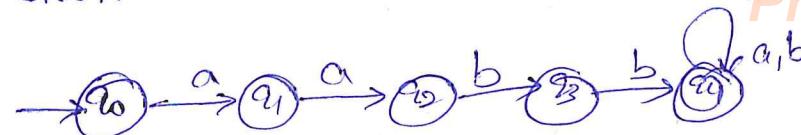


c) start with 'aba'

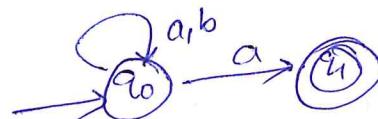


Start with 'aabb'

Ph: 844-844-0102



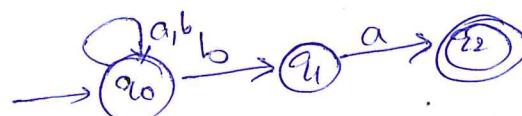
③a) End with 'a'



$w = ba$

$q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1$ Accepted
F

③b) Ends with 'ba'

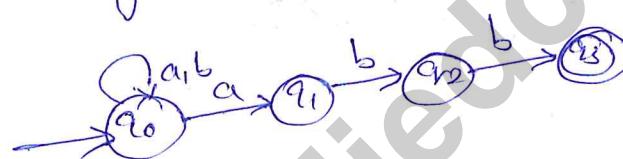


$w = aaba$

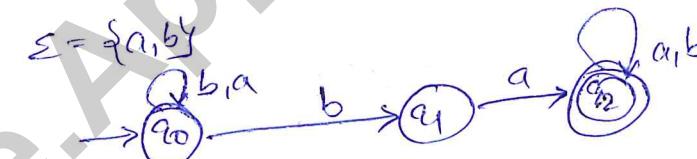
$q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_2$

Every NFA \rightarrow DFA

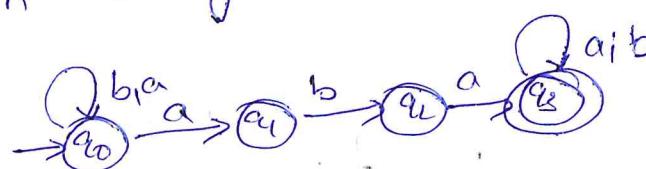
④c) Ending with abb



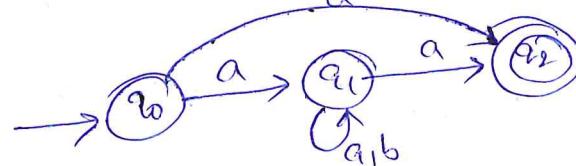
④a) Contain substring 'ba'



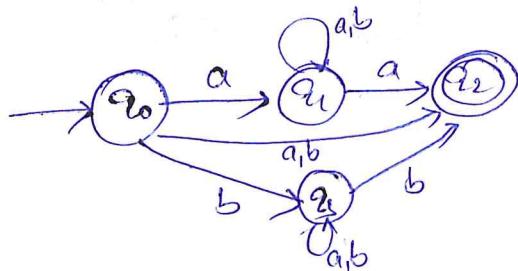
④b) Contain substring 'aba'



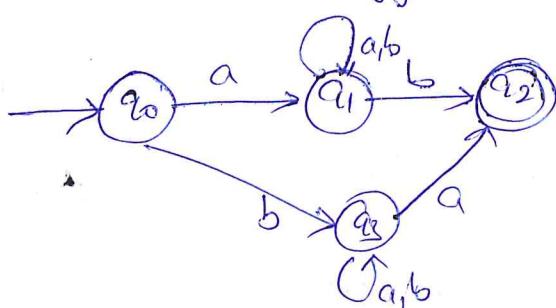
⑤a) Starts & ends with a



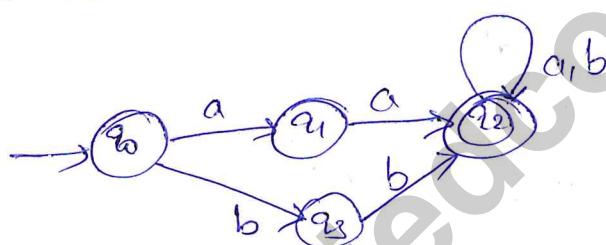
40



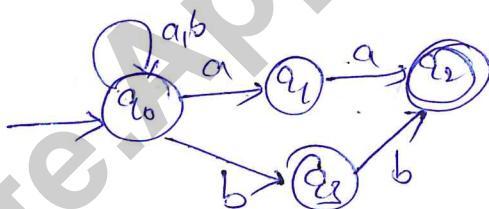
- (5c) Starts & ends with different Symbol



- (5d) Starts with aa & bb



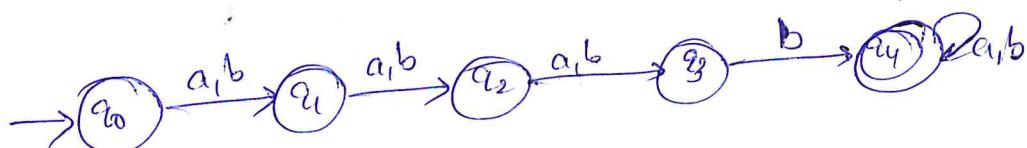
- (5e) Ends with aa & bb



- (6a) 4th symbol from left is 'b'

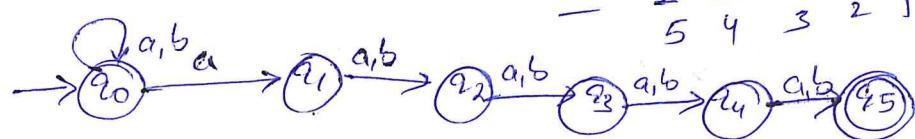
$$W = \underline{a} \ y \ \underline{b} \ z \\ \text{Symbol}$$

$$\begin{aligned} W_1 &= aaab \checkmark \\ W_2 &= abab \underline{abb} \checkmark \\ W_3 &= aaaaabb \checkmark \end{aligned}$$



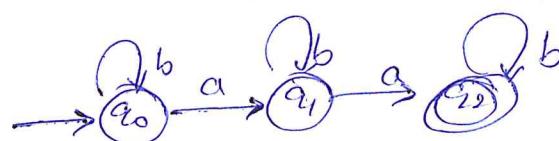
5th symbol from right is 'a'!

Ph: 844-844-0102



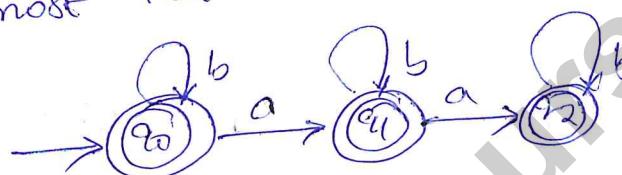
(7a)

contains exactly 2 a's



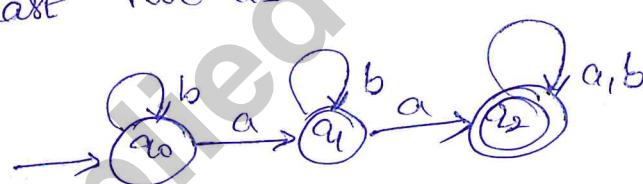
(7b)

at most two a's



(7c)

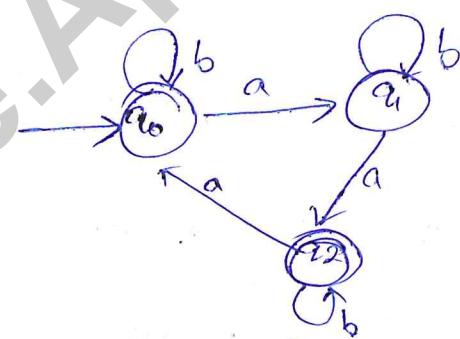
at least two a's



(7d)

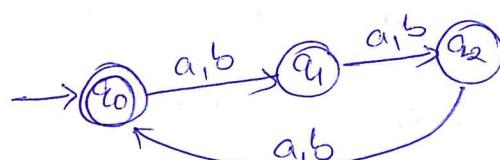
a's $\equiv 2 \pmod{3}$

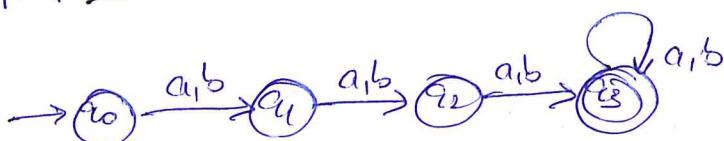
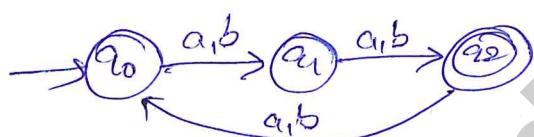
NFA & DFA



(8a)

$|w| \equiv 0 \pmod{3}$



(8b) $|W| \leq 3$ (8c) $|W| \geq 3$ (8d) $|W| \equiv 2 \pmod{3}$. NFAE/DFAObservations:-

- comp: S/W → DFA → easier to implement & efficient
- NFA: Equivalent DFA
- Design of NFA's easier, especially for complex systems
- NFA: parallel computing engine & multi-threaded
- No need of dead states in NFA
- Multiple transition paths & some transitions are not defined in NFA's.

Conversion from NFA to DFA:

Complex Sys \rightarrow NFA \rightarrow DFA
P.T.O. 844-844-0102

<u>NFA</u>	a	b
$\rightarrow q_0$	q_0	q_1
q_1	q_0, q_1	q_2
q_2	q_1	q_0, q_1

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

2^Q

DFA

<u>DFA</u>	a	b
$\rightarrow q_0$	q_0, q_1	
q_1	q_0, q_1	q_2
q_2	q_0, q_1	q_2
q_3	q_1	q_0, q_1
q_4	q_0, q_1	q_0, q_2
q_5	q_0, q_1	q_0, q_2
q_6	q_0, q_1	q_0, q_1

need not be the min DFA

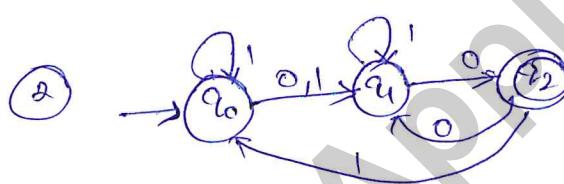
① Input Symbols

② States

③ Final State

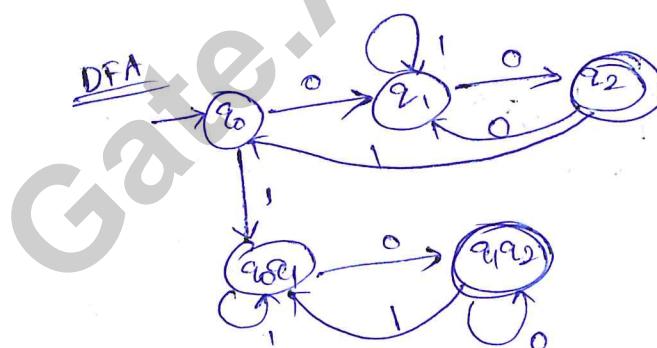
④ Initial State

⑤ δ'



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

At most
2³ states
in DFA



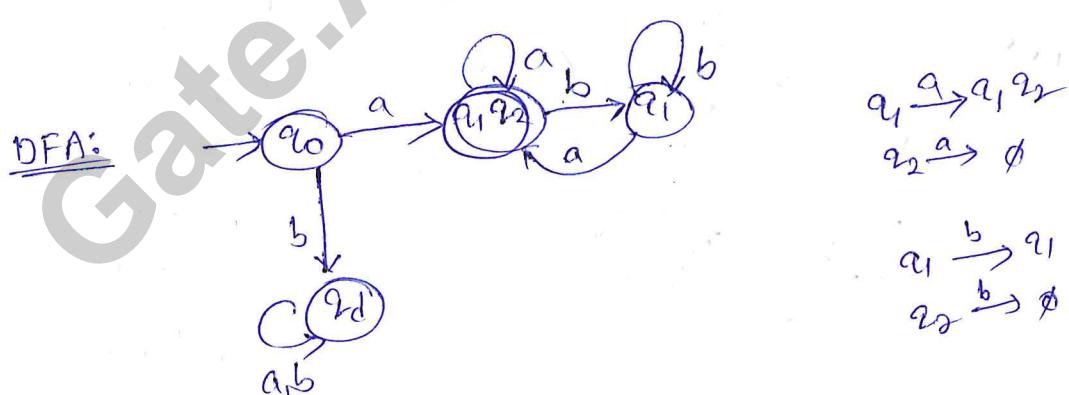
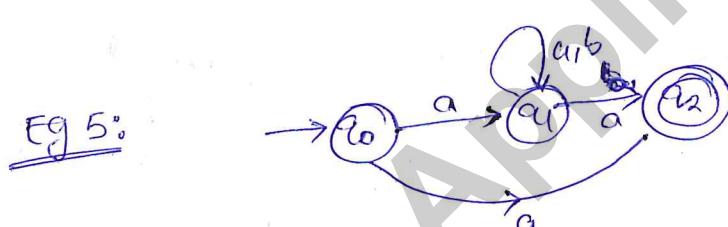
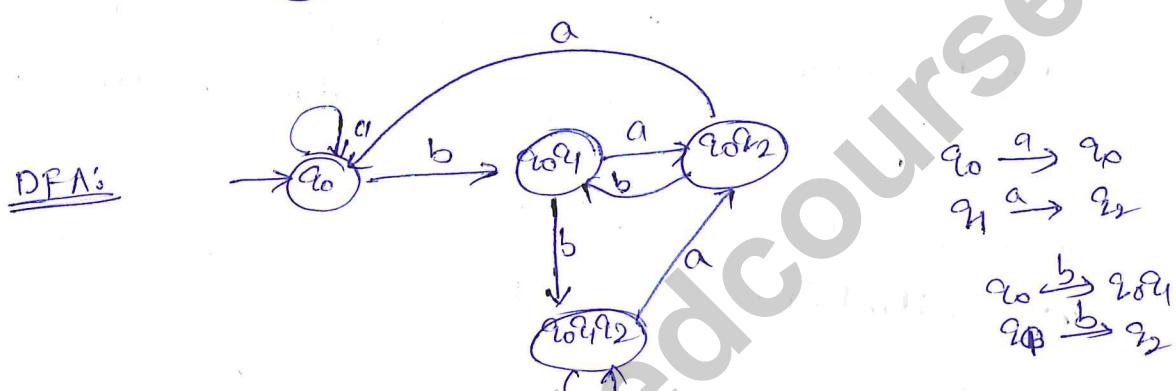
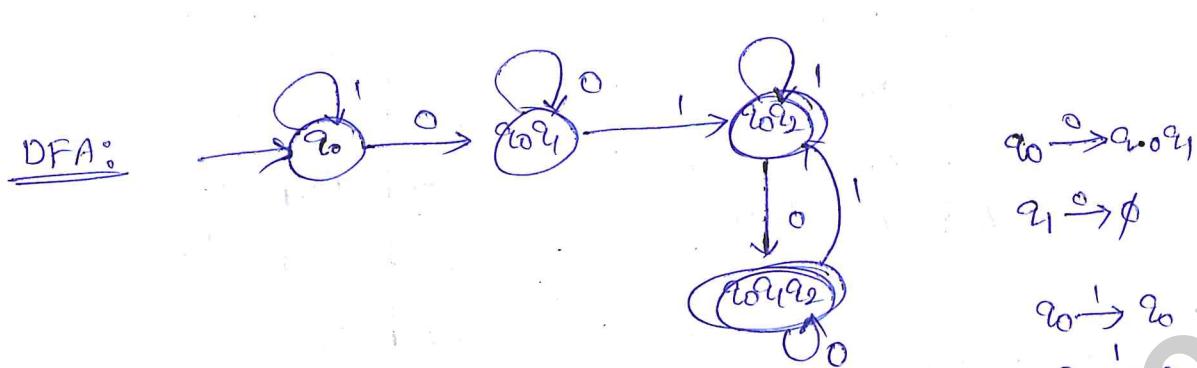
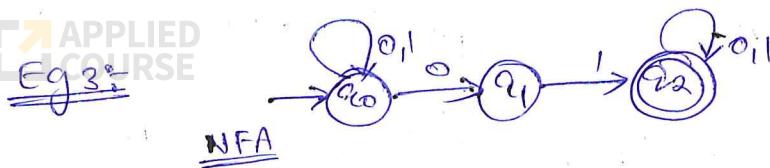
<u>DFA</u>	0	1
q_0	q_1, q_2	q_3
q_1	q_2	q_1

$$q_1 \xrightarrow{0} q_2$$

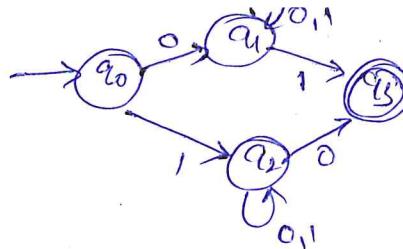
$$q_2 \xrightarrow{0} q_1$$

$$q_1 \xrightarrow{1} q_3$$

$$q_2 \xrightarrow{1} q_3$$



M



Ph. 844-844-0102

- ① $L = L(M)$
- ② change Final to nonfinal vice-versa
 $\rightarrow M' \in L'$

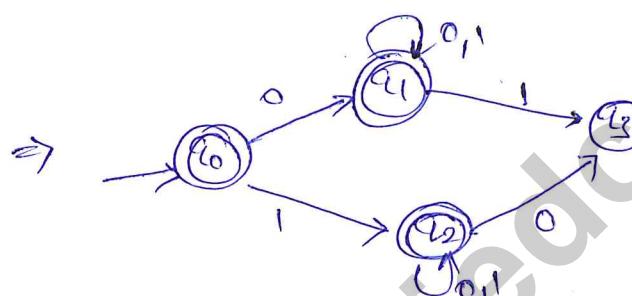
- ③ which options are correct

a) $L = L'$

b) $L \subset L'$

c) $L' = \epsilon^*$

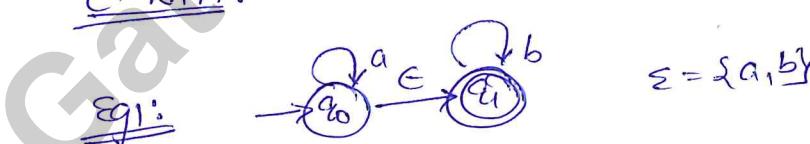
d) None



$L' = \{ \epsilon, 0, 1, \dots \}$

Epsilon-NFA + conversion from E-NFA to NFA & DFA

E-NFA:



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

e transition: By reading empty string we can move from one state to another state.

NFA: easier to model

E-NFA: more easier to model

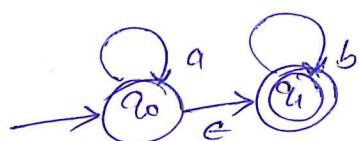
(46)

$$A \cdot E = A = E \cdot A$$

$$W \cdot E = W = E \cdot W$$

$L = \{ \epsilon, a, aa, \dots, b, b, \dots, ab, aab, aabb, \dots, abb \}$

$\overbrace{\hspace{1cm}}$



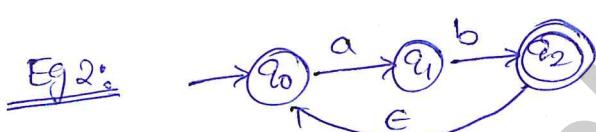
$$L = \{ amb^n \mid m, n \geq 0 \}$$

power: $\epsilon\text{-NFA} \sim \text{NFA} \sim \text{DFA}$

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

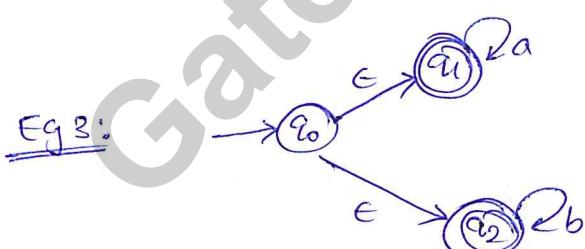
$$\delta: Q \times \Sigma \rightarrow 2^Q \quad \text{NFA}$$

$Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$ set of states
 State Symbol Empty String



$$L = \{ ab, abab, \dots \}$$

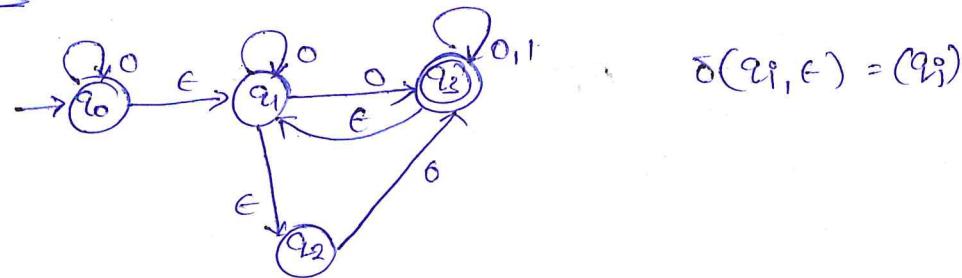
$$= \{ (ab)^n \mid n \geq 1 \}$$



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

$$= \{ \epsilon, b, bb, bbb, \dots \}$$

$$= \{ a^n b^n \mid n \geq 0 \} \quad (\text{Q})$$

ϵ -closure:


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

set of all states that can be reached from the state q_i by reading an empty string ϵ is known as ϵ -closure.

$$\epsilon\text{-closure}(q_1) = \{q_1, q_2\}$$

$$\epsilon\text{-closure}(q_2) = \{q_2\}$$

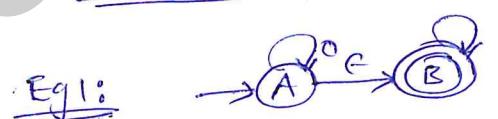
$$\epsilon\text{-closure}(q_3) = \{q_3, q_1, q_2\}$$

Note: ϵ -closure is never an empty set

$$\epsilon\text{-closure}(\emptyset) = \emptyset$$

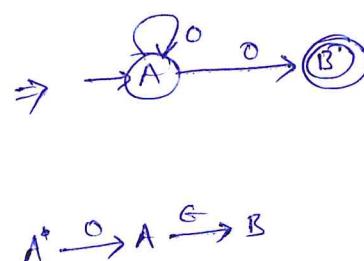


① E-NFA to NFA Conversion:



$$\epsilon(A) = \{A, B\}$$

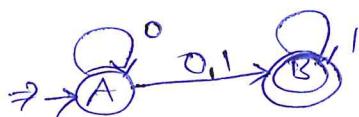
$$\epsilon(B) = \{B\}$$



$$= \epsilon\text{-closure}(A, \emptyset)$$

$$= \epsilon\text{-closure}(A)$$

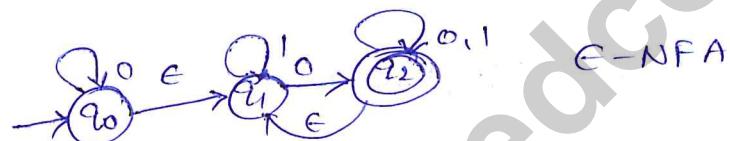
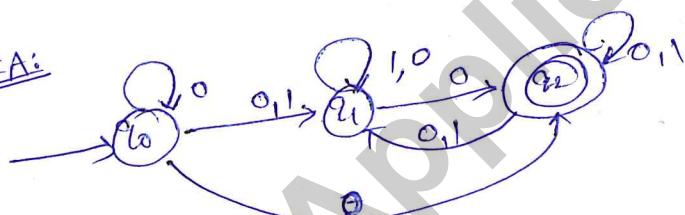
$$= \{A, B\}$$



$$A \xrightarrow{\epsilon} B \xrightarrow{1} B$$

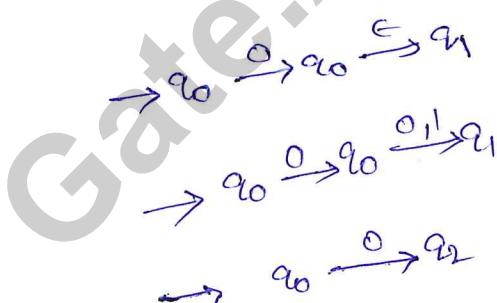
$$A \xrightarrow{\epsilon!} B$$

$$\rightarrow A \xrightarrow{1} B$$

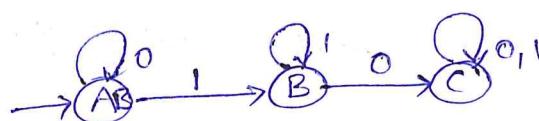
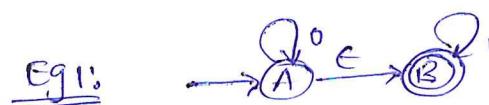
Ex 2:NFA:No change in
set of states ✓

initial state ✓

final state ✓



ϵ -NFA to DFA Conversion:

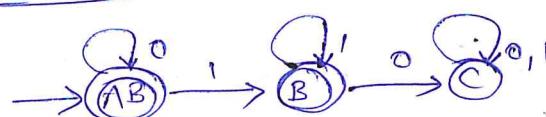


$A \xrightarrow{0} A \xrightarrow{\epsilon} B$

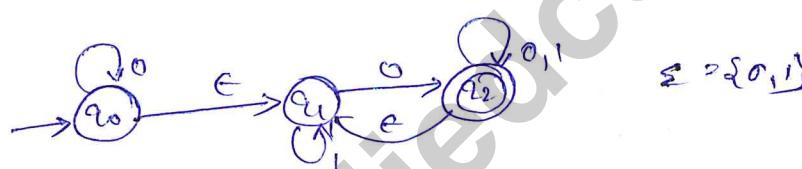
$A \xleftarrow{\epsilon} B \xrightarrow{1} B$

$B \xrightarrow{0} \text{no transition}$

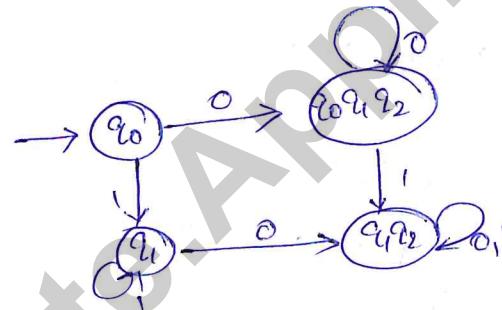
Final State:



Eg 2:



$\epsilon\text{-closure}(q_0) = \{q_0, q_1\}$



$q_0 \xrightarrow{0} q_0 \xrightarrow{\epsilon} q_1$

$q_0 \xrightarrow{\epsilon} q_1 \xrightarrow{0} q_2$

$q_0 \xrightarrow{\epsilon} q_1 \xrightarrow{1} q_1$

$q_0 q_1 q_2 \xrightarrow{0} q_0 q_1 q_2$

$q_0 q_1 q_2 \xrightarrow{1} q_1 q_2$

$q_1 \xrightarrow{0} q_2 \xrightarrow{\epsilon} q_1$

$q_1 q_2 \xrightarrow{0} q_1 q_2$

Ph: 844-844-0102

$\epsilon\text{-NFA} \rightarrow \text{NFA} \rightarrow \text{DFA}$

Thomson

Subset

construction

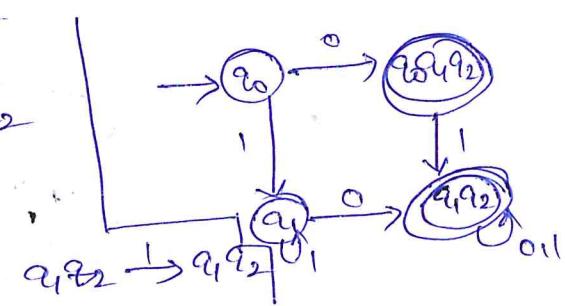
compute ϵ -closure of initial state

$$\epsilon(A) = \{A, B\}$$

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

Final State: q_2

$q_0 q_1 q_2 \}$
 $q_1 q_2 \}$
final state in DFA



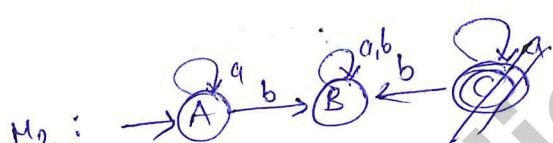
Decision properties of FA:

① Emptyness: Does the FA accept empty or non-empty L.

(a) Delete / ignore all unreachable states

(b) at least one final state \Rightarrow Non-empty L

No final staty \Rightarrow Empty Language.



② Finiteness: Does the FA accept finite language?

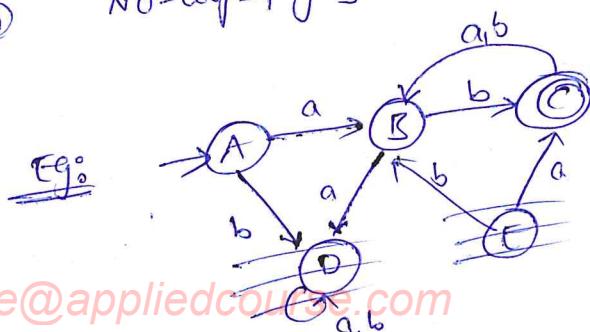
(a) Ignore / delete unreachable state

(b) Ignore / delete dead-state from which all other states

(c) Ignore / delete dead-state from which we cannot reach a final state

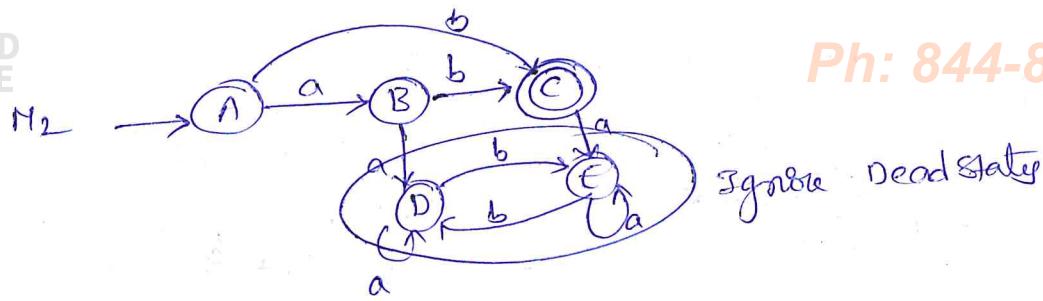
(d) Loops / cycles \Rightarrow infinite - Language

(e) No-loops / cycle \Rightarrow Finite Language.



M1: Infinite Language.

ignore unreachable & dead state



Ignore Dead State

The Language accepted by FA is $\{ b, ab \}$

Finite Language

③ Membership Acceptance:

w M

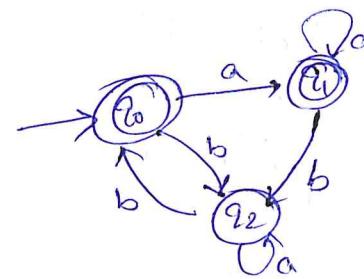
If $q_0 \xrightarrow{w} \text{final state}$ then w is accepted by M

$q_0 \rightarrow q_1$
→ Transition path

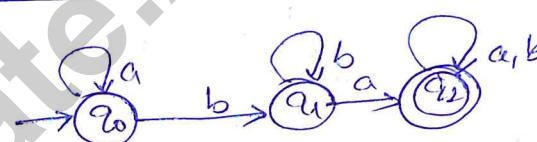
④ Equality of M_1 & M_2 :

$M_1 = M_2$ iff

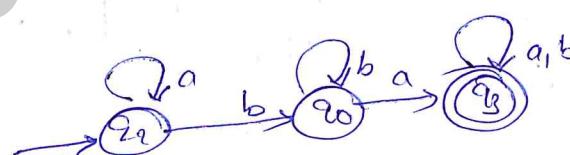
$$L(M_1) = L(M_2)$$



Isomorphism:



M_1 and M_2 are Isomorphic to each other.



$$|Q_1| = |Q_2|$$

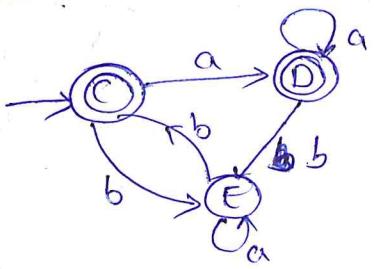
$$L(M_1) = L(M_2)$$

Quintuple

$$\Sigma_1 \quad Q_1 = \{ q_0, q_1, q_2 \}$$

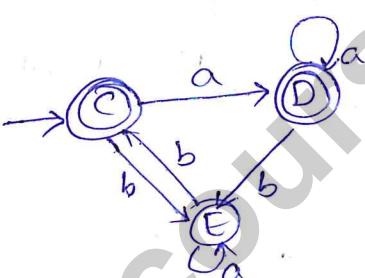
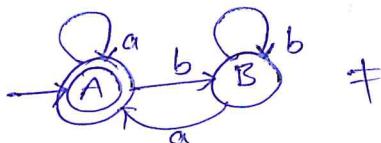
$$\# \quad \Sigma_2 \quad Q_2 = \{ q_1, q_2, q_3 \}$$

By changing the names / labels of the states, we can also get the second machine.

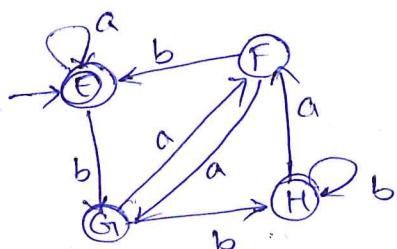
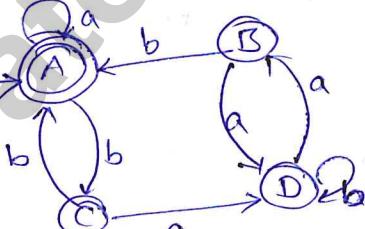
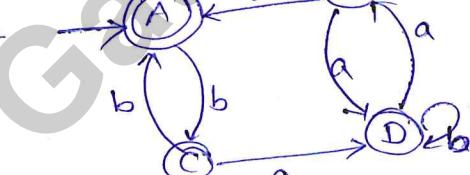
Eg 1:

	a	b
F, F	A, C	AD BE
F, F	A, D	AD BE
NF, NF	B, E	BE AC

FF ✓
NF NF ✓
FNF X
NF FX

Eg 2:

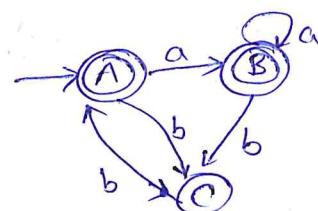
	a	b
AC	AD	BE
AD	AD	BE
BE	AE	BC

Eg 3:

	a	b
AE	AE	CG
CG	DF	AH → NF

 $M_1 \neq M_2$

Eg1:



	a	b	
A	B	C	
B	B	C	
C	-	A	

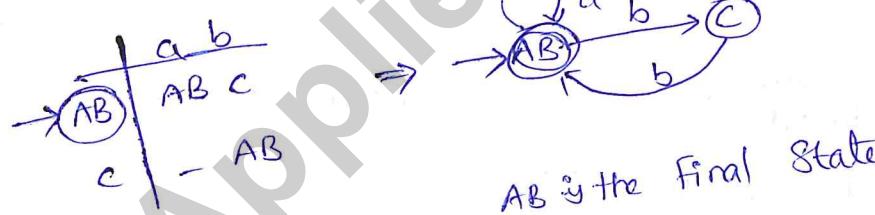
$\Sigma = \{a, b\}$

① Delete all unreachable states

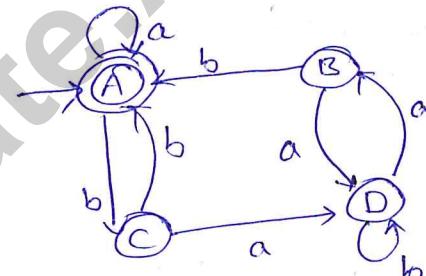
② Merge final state & NF-state

	a	b	
A	B	C	
B	B	C	

$\Rightarrow A \& B$ are Final State and equal State

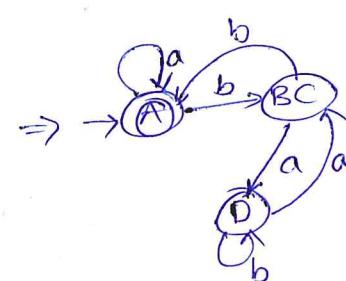


Eg2:



	a	b	
A	A	C	
B	D	A	
C	D	A	
D	B	D	

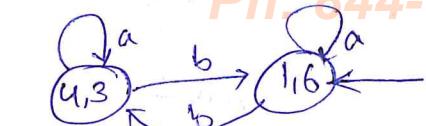
	a	b	
A	A	BC	
BC	D	A	$\rightarrow F$
D	BC	D	$\rightarrow NF$



A is Final State

δ	a	b
→ 1	6	4
2	5	2
③	4	6
④	3	1
5	2	5
6	1	3

M1



M2
↓
2-state

(2,5) unreachable state
 $Q_{a,b}$

Eg4:

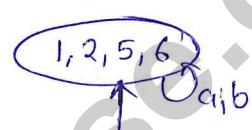
δ	a	b
→ 1	6	6
2	5	2
③	4	4
④	3	3
5	2	5
6	1	1

M1



unreachable state

$$L = \emptyset, Y = \emptyset$$

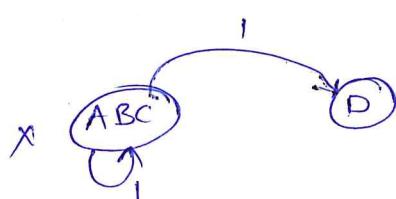
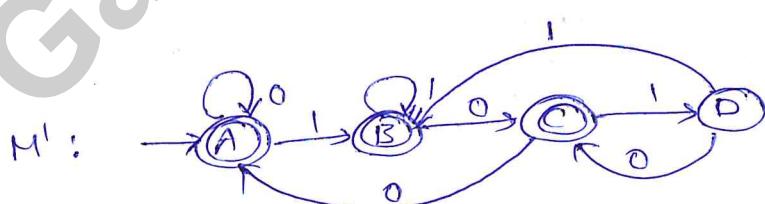
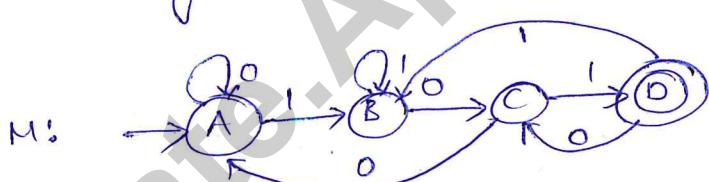


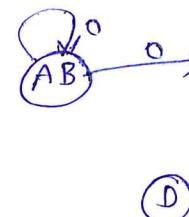
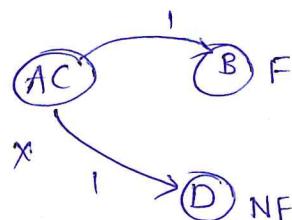
1, 2, 5, 6
 $Q_{a,b}$
M2 containing only one state

$$L = \emptyset, Y = \emptyset$$

Eg5: Build $\min FA$ that does not accept strings

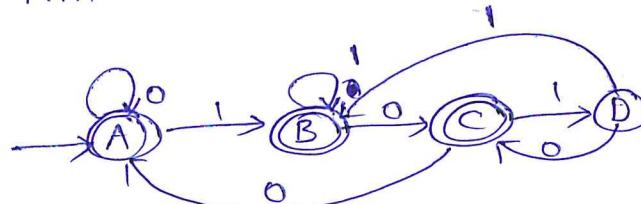
ending in 101 $\Sigma = \{0, 1\}$





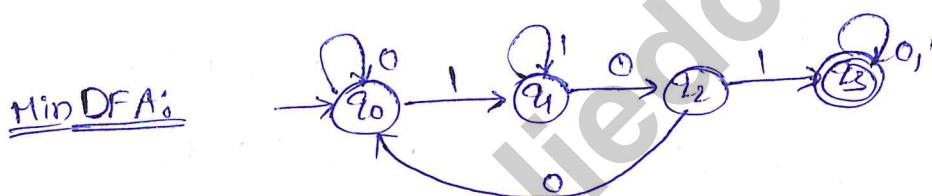
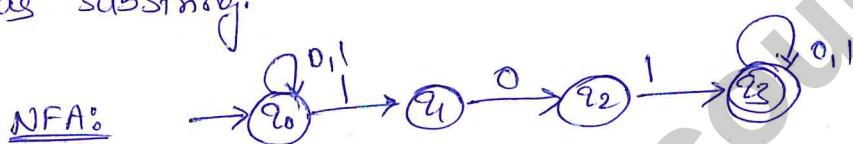
(D)

\Rightarrow Min FA \Downarrow

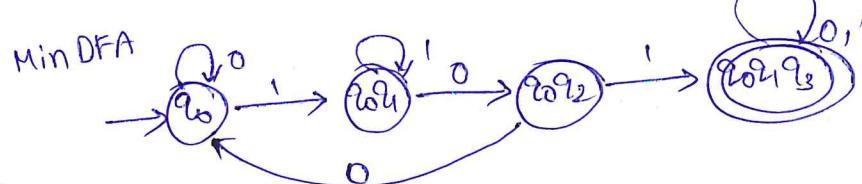
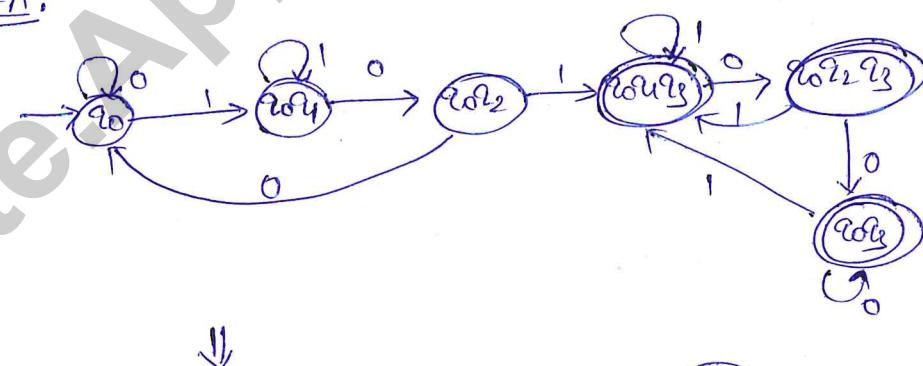


contains 4 staty.

Eg 6: Build min FA that accepts string that contain 101 as substring.



NFA to DFA:



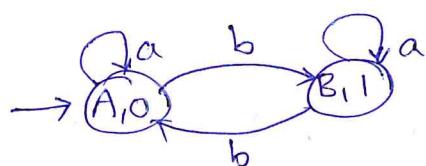
contains 4 staty

FA with outputs:

↳ DFA, NFA, ϵ -NFA \rightarrow acceptance [Final State] \leftarrow without output

- * Moore Machine
- * Mealy Machine

Moore Machine:



$\Sigma = \{a, b\}$ \leftarrow input set

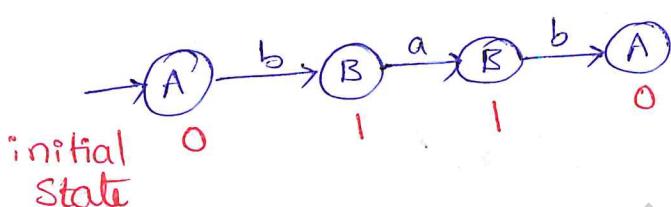
$\Delta = \{0, 1\}$ \leftarrow output set

$Q = \{A, B\}$ \leftarrow State

δ : Transition Function

λ : Output function

no final state



input: n-length word / String

output: (n+1) length

$$M = (Q, \Sigma, \Delta, \delta, \lambda, q_0) \rightarrow 6\text{-tuple}$$

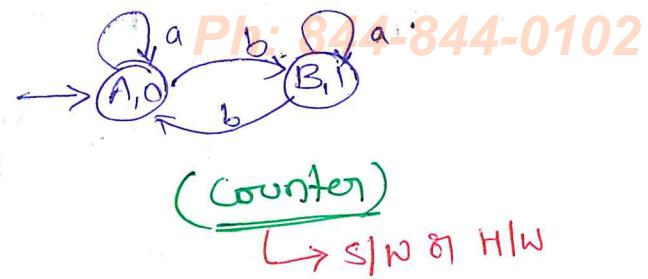
$\delta: Q \times \Sigma \rightarrow Q$ DFA

$\lambda: Q \rightarrow \Delta$

Note:- let $w = \epsilon$ then output = $\lambda(q_0)$

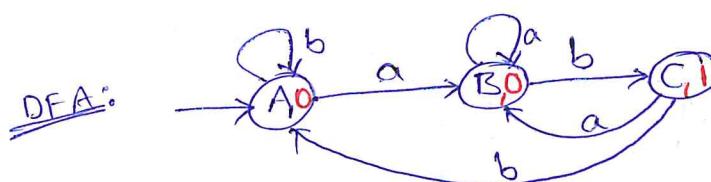
DFA, NFA, ϵ -NFA \rightarrow Transition graph (8) table

δ	a	b	output
A	A	B	0
B	B	A	1



Let $w = abab$
 Not a binary number
 ↑↑↑↑
 0 0 1 0 1 → count the number of 1's
 = 2 → # of b's in the input string.

Eg 2: Construct a moore machine that counts # of occurrence of substring 'ab'. $\Sigma = \{a, b\}$ $\Delta = \{0, 1\}$

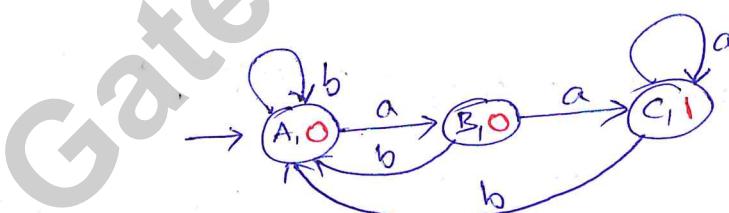


$ab \rightarrow 001$

aab → 0001

abab → 00101

Eg 3: Counts # occurrence of two consecutive a's.



$aa \rightarrow 001$

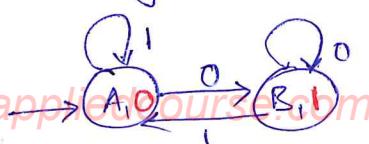
$baa \rightarrow 0001$

$bbbb \rightarrow 00000$

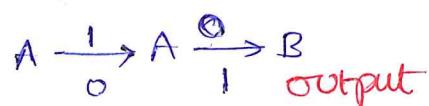
$aaa \rightarrow 0011$

$aab \rightarrow 0010$

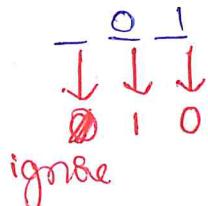
Eg 4: Output is complement of a binary input string. $\Sigma = \{0, 1\}$ $\Delta = \{0, 1\}$



1011
 0100 ← 1's Complement
 H/W ←

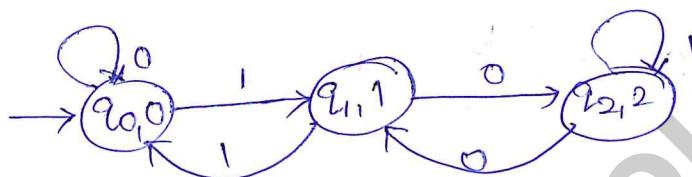


$w = 01$



Eg 5: Input: Binary String, output = (input) mod 3

$$\Sigma = \{0, 1\} \quad \Delta = \{0, 1, 2\}$$



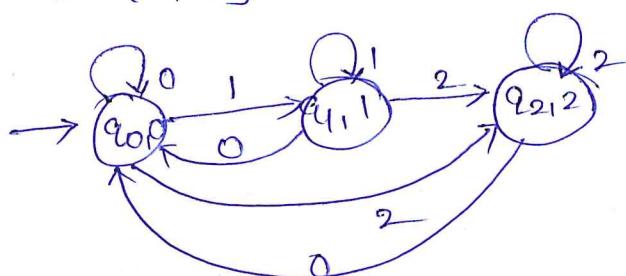
bin dec rem

0000	0	0
0001	1	1
0010	2	2
0011	3	0
0100	4	1
0101	5	2
0110	6	0

Let $w = 101$
 ignore $\overset{\uparrow\uparrow\uparrow\uparrow}{0122} \Rightarrow$ is the output
 $= (101) \text{ mod } 3$
 $= 2$

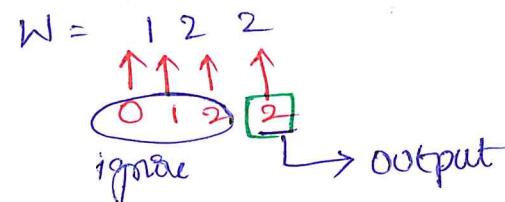
Eg 6: Input: base 3 number; output = (input) mod 3

$$\Sigma = \{0, 1, 2\} \quad \Delta = \{0, 1, 2\}$$

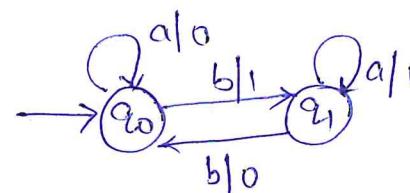


$$(120)_3 = 0 + 6 + 9 = 15$$

<u>base 3</u>	<u>dec</u>	<u>mod 3</u>
0	0	0
1	1	1
2	2	2
10	3	0
11	4	1
12	5	2
20	6	0
21	7	1
22	8	2



Mealy Machine: Output is associated with the transition



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

$$q_0 = q_0$$

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

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

$$\Rightarrow \quad W = E \quad \left. \begin{array}{l} \\ \text{output} = E \end{array} \right\}$$

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

→ Transition diagram

→ Transition Table

		a	b		a	b
		q0	q1	q0	0	1
①	q0	q0	q1	q0	0	1
	q1	q1	q0			

$$W = \overbrace{\begin{matrix} ab & ab & a \\ \uparrow & \uparrow & \uparrow \\ 0 & 1 & 0 & 0 \end{matrix}}^n$$

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

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

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

$$q_0 = q_0$$

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

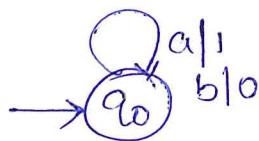
$$\lambda: Q \times \Sigma \rightarrow \Delta$$

		a	b	
		q0	q1	q0
②	q0	q0, 0	q1, 1	q0
	q1	q1, 1	q0, 0	

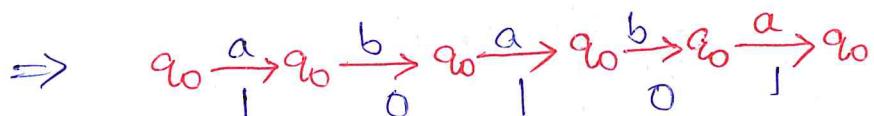
Count # occurrences of 'a's in a String $\Sigma = \{a, b\}$

Ph: 844-844-0102

$$\Delta = \{q_0, 1\}$$

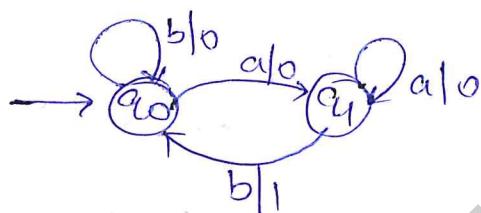


$$\Rightarrow w = ababa$$



Moore \rightarrow Mealy

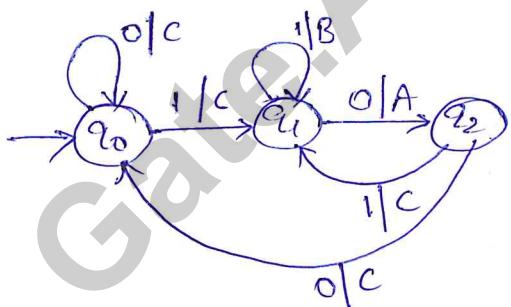
Eg 2: Count the number of occurrence of 'ab' as substring



$$w = abab \\ \begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow \\ 0 & 1 & 0 & 1 \end{matrix}$$

Eg 3:

$$\Sigma = \{0, 1\} \quad \Delta = \{A, B, C\}$$



$010 : A$

$011 : B$

otherwise: C

1:C

0:C

10:CA

11:CB

110:CBA

101:CAC

1011: CACB

Eg 4: 1's Complement Mealy Machine



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

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

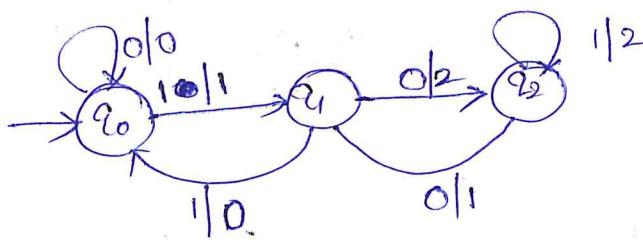
Ph: 844-844-0102

Eg 5:

$$\Sigma = \{0, 1\}, (\text{input})_2 \bmod 3$$

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

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

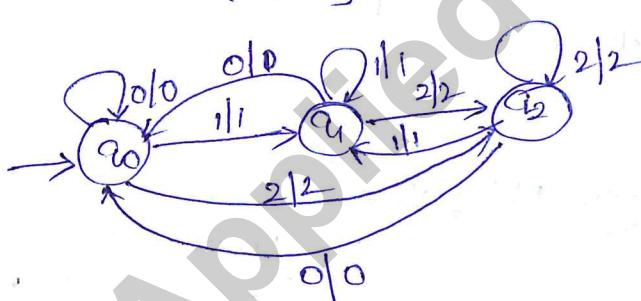


<u>bi</u>	<u>DEC</u>	<u>mod3</u>
0	0	0
1	1	1
10	2	2
11	3	0
100	4	1
101	5	2
110	6	0

Eg 6:

$$\Sigma = \{0, 1, 2\}, \text{ output} = (\text{input})_3 \bmod 3$$

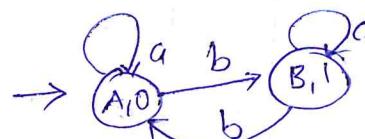
$$\Delta = \{0, 1, 2\}$$



<u>Base3</u>	<u>Dec</u>	<u>Mod3</u>
0	0	0
1	1	1
2	2	2
10	3	0
11	4	1
12	5	2
100	15	0
101	16	1
110	17	2

Conversions: Meille \rightarrow Mealy Machine

Eg 1:



<u>S</u>	<u>a</u>	<u>b</u>	<u>Q</u>
A	A	B	0
B	B	A	1

\Rightarrow

<u>Mealy:</u>			
<u>S</u>	<u>a</u> <u>0</u>	<u>b</u> <u>0</u>	<u>P</u>
A	A 0	B 1	
B	B 1	A 0	

Moore

δ	a	b	λ
$\rightarrow q_0$	$q_1\ q_2$	0	
q_1	$q_0\ q_3$	1	
q_2	$q_3\ q_0$	0	
q_3	$q_2\ q_1$	1	

Mealy

δ, λ	a	b	
$\rightarrow q_0$	$q_1\ 1$	$q_2\ 0$	
q_1	$q_0\ 0$	$q_3\ 1$	
q_2	$q_3\ 1$	$q_0\ 0$	
q_3	$q_2\ 0$	$q_1\ 1$	

Ph: 844-844-0102

Eg 2:

Moore

δ	a	b	λ
$\rightarrow q_0$	$q_2\ q_4$	q_4	0
q_1	$q_0\ q_2$	q_2	1
q_2	$q_3\ q_1$	q_1	2
q_3	$q_4\ q_0$	q_0	0
q_4	$q_1\ q_3$	q_3	1

Mealy

δ	a	b	
$\rightarrow q_0$	$q_2\ 2$	$q_4\ 1$	
q_1	$q_0\ 0$	$q_2\ 2$	
q_2	$q_3\ 0$	$q_1\ 1$	
q_3	$q_4\ 1$	$q_0\ 0$	
q_4	$q_1\ 1$	$q_3\ 0$	

states don't change.

Mealy M/c to Moore Machine:

Eg 1:

δ, λ	a	b
$\rightarrow q_0$	$q_2\ 0$	$q_1\ 0$
q_1	$q_0\ 0$	$q_2\ 1$
q_2	$q_1\ 1$	$q_0\ 0$

Moore M/c

δ	a	b	λ
$\rightarrow q_0$	q_{20}	q_{10}	0
q_10	q_0	q_{10}	0
q_{11}	q_0	q_{10}	1
q_{20}	q_{11}	q_0	0
q_{21}	q_{11}	q_0	1

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

states need not be the same.

<u>Mealy</u>	0	1
δ, λ	0	1
→ q ₀	q ₂ , 1	q ₃ , 1
q ₁	q ₃ , 1	q ₄ , 1
q ₂	q ₀ , 1	q ₁ , 0
q ₃	q ₄ , 0	q ₂ , 0
q ₄	q ₁ , 0	q ₀ , 0

Ph: 844-844-0102

$$q_0 \xleftarrow{0} q_1$$

$$q_1 \rightarrow 0$$

$$q_2 \xleftarrow{0} q_1$$

$$\begin{aligned} q_3 &\rightarrow 1 \\ q_4 &\xleftarrow{0} q_1 \end{aligned}$$

Moore Machine :-

	0	0	1	λ
(δ)	→ q ₀₀	q ₂₁	q ₃	0
	→ q ₀₁	q ₂₁	q ₃	1
	q ₁	q ₃	q ₄₁	0
	q ₂₀	q ₀₁	q ₁₀	0
	q ₂₁	q ₀₁	q ₁	1
	q ₃	q ₄₀	q ₂₀	1
	q ₄₀	q ₁	q ₂₀	0
	q ₄₁	q ₁	q ₀₀	1

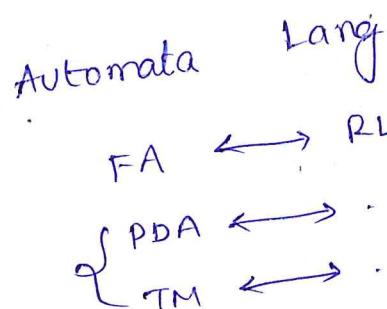
Regular Languages:

FA \leftrightarrow DFA, NFA, C-NFA
 FA \leftrightarrow Moore, Mealy

A Language is regular iff

\exists FA that accepts every word in L.

L is RL if \exists FA, M such that $L(M) = L$



(64)

→ Non-regular Language - no FA exist

Ph: 844-844-0102

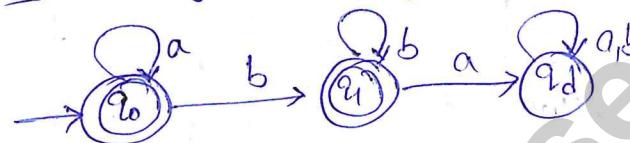
→ Every finite Language is Regular

→ $L = \emptyset = \{\}$ Regular

Example:

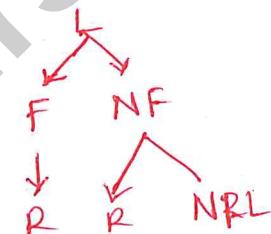
① $L = \{ ab, abb, aab \}$ Finite - Regular

② $L = \{ amb^n \mid m, n \geq 0 \}$ Not finite - Regular



③ $L = \{ ambn \mid m \geq 2, n \geq 3 \}$ - Regular

④ $L = \{ amb^n \mid m \geq n = \text{Const} \}$ - Finite
Finite Regular



⑤ $L = \{ amb^n \mid m = n = \text{Const} \}$ - Finite - Regular

⑥ $L = \{ amb^n \mid 1 \leq m = n \leq \text{Const} \}$ - Finite - Regular
 $\{ ab, aabb \}$

⑦ $L = \{ amb^n \mid 1 \leq m \leq n \leq \text{Const} \}$ - Finite - Regular

⑧ $L = \{ amb^n \mid n \geq 1 \}$: Non-finite = {ab, aabb, ...}
 ↴ Non-regular pumping lemma

* State in FA to

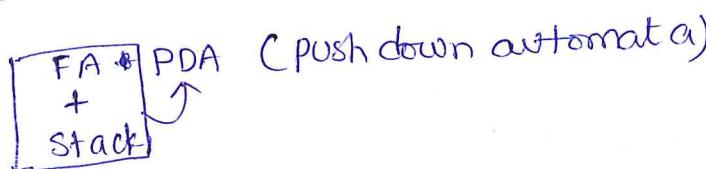
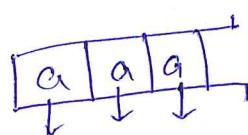
Keep a count $\rightarrow \infty$

\rightarrow Variable: cnt (increment) \rightarrow read/write



→ stack: For all 'a's push operation, for every 'b' apply one pop operation.

Ph: 844-844-0102



FA + read/write capability : TM

- ⑨ $L = \{a^m b^n \mid m=10, 20, 30, \dots \}$ Reminder
 $m \pmod{10} = 0 \quad (0 \dots 9)$
 $n=5, 10, 15, \dots$ $m \pmod{5} = 0 \quad (0 \dots 4)$
- ↓
Regular

- ⑩ $L = \{a^m b^n \mid m < n\} \rightarrow \text{Non-Finite} \rightarrow \text{Non-Regular}$



- ⑪ $L = \{a^m b^n \mid m > n\} \rightarrow \text{Non-Regular}$

- ⑫ $L = \{a^m b^n \mid m \neq n\} \rightarrow \text{Non-Regular}$

- ⑬ $L = \{a^m b^n \mid m = 2n\} \rightarrow \text{Non-Regular}$

- ⑭ $L = \{a^m b^n \mid m = n^2\} \rightarrow \text{Non-Regular}$

- ⑮ $L = \{a^m b^n \mid m+n=10\} \rightarrow \text{Finite} \rightarrow \text{Regular}$

- ⑯ $L = \{a^m b^n \mid \gcd(m,n)=1\} = \text{Solve } \begin{matrix} \text{Reminder} \\ \downarrow \\ \text{N.R.L} \end{matrix}$

- ⑰ $L = \{a^m b^n \mid m+n=\text{even}\} \rightarrow \text{Regular}$

- ⑱ $L = \{wwR \mid w \in \Sigma^*\} \rightarrow \text{Non-Regular}$

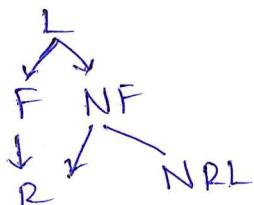
(6)


 $\{w \in \Sigma^* \mid |w_a| = |w_b|\}$ - Not Finite
 $\Rightarrow abab$

Non-Regular.

Properties of RL:① Finite Language \Rightarrow RL

②



③ RL: Finite or Non-Finite

④ Every NRL has to be infinite Lang

⑤ Subset of a RL need not be regular

RL $\nsubseteq \{a^m b^n \mid m, n \geq 0\}$ $L_2 = \{a^m b^n \mid m=n \geq 0\}$

$L_2 \subset L$

⑥ Every finite subset of a RL is a RL

⑦ Every finite subset of a NRL is a RL

⑧ Every subset of a NRL need not be an RL

 $L = \{a^m b^n \mid m=n\}$ NRL $L' = \{ab, acbb, aaabb\}$ RL

⑨ Supersets of a RL need not be RL

" " NRL " " NRL "

is a RL

⑩ Finite union of Regular Languages

$$L = L_1 \cup L_2 \cup L_3 \cup \dots \cup L_n$$

$\downarrow M$ $\downarrow m_1$ $\downarrow m_2$ $\downarrow m_3$ \dots $\downarrow m_n$

Infinite union of RL need not be RL Ph: 844-844-0102

⑪ Finite intersection of RL is a RL

$$L = L_1 \cap L_2 \cap L_3 \cap \dots \cap L_n$$

\uparrow

$M \Rightarrow M_1 \quad M_2 \quad M_3 \quad \dots \quad M_n$

Infinite intersection of RL need not be RL.

Regular Expressions :- (RE)

GREP, Python, Java - string matching

→ One way to represent a RL

$$L = \{ \dots \}$$

FA

→ operators: $\ast, \circ, +$ widely used symbols in RE

↓ ↓ →
 Kleene Concatenation OR
 closure union

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

$$\textcircled{1} \quad r = a^* = \{ \epsilon, a, aa, aaa, \dots \}$$

$$\textcircled{2} \quad r = a + b = \{a, b\}$$

$$\textcircled{3} \quad r = ab = \{ab\}$$

$$\textcircled{4} \quad a^* + ba = \{ \epsilon, a, aa, \dots, ba \}$$

$$\textcircled{5} \quad (ab)^* + b^* a^* =$$

$$\{ \epsilon, ab, abab, \dots \}^0$$

$$\{ \epsilon, b, bb, \dots \} \{ \epsilon, a, aa, \dots \}$$

$$\Rightarrow \{ \epsilon, a, aa, b, ba, baa, bb, bba, \dots \}$$

Order of precedence:

eg: $(a+b)^*$ ^{*}
 $\quad \quad \quad |$ 2
 $\quad \quad \quad 3$

First: *

Second: .

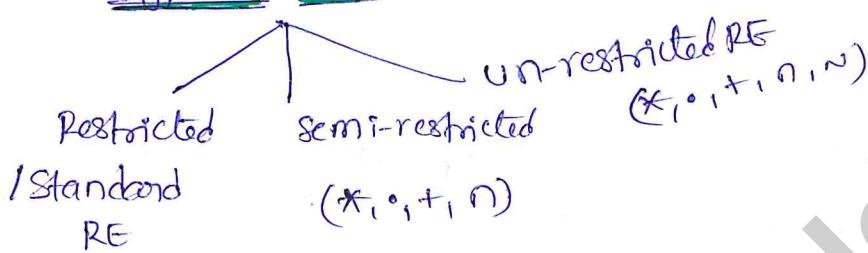
Third: +

$$(a+b)^* = \epsilon \cup (a+b) \cup (a+b)^2 \cup \dots$$

$$(a+b)^2 = (a+b)(a+b)$$

$$= \{aa, aab, baa, bab\}$$

$$r = a^+ = \{a, aa, aaa, \dots\}$$

Types of RE:Observations:

① Every finite Language is regular & we can express it using a regular expression.

$$\hookrightarrow w_1 + w_2 + \dots + w_n$$

$$ab + abab + bab + abb$$

reg expr \leftrightarrow FA

$$r^* = \{\epsilon, r, rr, \dots\}$$

$$r^+ = \{r, rr, rrr, \dots\}$$

$$L(r^*) = L(r)^*$$

$$\Rightarrow r = ab$$

$$\Rightarrow L((ab)^*) = \{\epsilon, ab, abab, \dots\}$$

$$\Rightarrow (L(r))^* \\ \Rightarrow (ab)^* = \{ \epsilon, ab, abab, \dots \}$$

③ $r = \epsilon \Rightarrow r^* = \{ \epsilon, \epsilon\epsilon, \epsilon\epsilon\epsilon, \dots \} \\ = \{ \epsilon \}$

④ $r = \epsilon \Rightarrow r^+ = \{ \epsilon, \epsilon\epsilon, \dots \} = \{ \epsilon \}$

⑤ $r = \phi = \{ \} \Rightarrow r^* = \{ \epsilon \} \\ \Rightarrow r^+ = \{ \} = \phi$

⑥ $r^+ \subset r^*$

$$r^* = r^+ \cup \{ \epsilon \}$$

⑦ $(r^*)^* = r^*$
 $r^* \cdot r = \{ r, rr, r^3, \dots \} = r^+$

Regular Expressions : Solved problems

① Which is correct

a) $r^* = r(r) \times$

b) $(r^*)^+ = r^+ \times = \{ r, rr, \dots \}$

LHS $\{ r^*, r^*r^*, \dots \}$
 $\hookrightarrow \epsilon, r, rr, \dots$

c) $(r^*)^* = r^* \checkmark$

LHS $\{ \epsilon, r^*, r^*r^*, \dots \}$
 $= \{ \epsilon, r^*, r^* \dots \}$
 $= r^*$

$$(r^+)^* = r^+ \quad \times$$



$$\{ \in, r^+, r^+r^+, \dots \}$$

(Q2) which of the following is/are correct.

$$\times @ \quad r^* = r^+ \quad \times$$

$$\checkmark b) \quad r^* = \bigcup_{i \geq 0} r^i = r^0 \cup r^1 \cup r^2 \cup \dots$$

$$\times c) \quad r^+ = \bigcup_{i \geq 1} r^i = r^1 \cup r^2 \cup r^3 \cup \dots$$

$$\checkmark d) \quad r^* = r^+ \text{ iff } r = \emptyset$$

(Q3) which of the following is/are incorrect

$$\times a) \quad r^* + r^+ = r^* \quad LHS = \{ \in, rr, rrr, \dots \} \neq RHS$$

$$\times b) \quad r^* \cdot r^+ = r^+$$

$$\times c) \quad r^* \cdot r^* = r^+ \quad \text{Answer}$$

$$\checkmark d) \quad r^+ (r^*)^+ = r^+$$

(Q4) choose the correct statement

$$\times a) \quad r^*, r^+ always represent \in\text{-lang} \quad r \in \quad r^* = \{ \in \} \quad r^+ = \{ \in \}$$

$$\times b) \quad r^*, r^+ are always finite language \quad r = a \quad r^* = \{ \in, a, aa, \dots \}$$

$$\checkmark c) \quad r^*, r^+ are finite iff r = \emptyset \text{ or } r = \infty$$

$$\times d) \quad \text{None}$$

which of the following are identical

Ph: 844-844-0102

- (a) $a^* = \{ \epsilon, aaa, aaa\ldots \}$ $a \neq b$
- (b) $(aa)^* = \{ \epsilon, aa, aaaa, \ldots \}$ $a \neq c$ $b \neq c$
- (c) $(a\delta)a = \{ aa, \ldots \}$ $a \neq c$
- (d) $(a+\epsilon)a^* = \{ \epsilon, a, aa, aaa, \ldots \}$

$a=d$

$a^* = (a+\epsilon)a^*$ \leftarrow Reg Lang : Multiple RE that can generate it

RE \rightarrow only one RL

Q6 which is correct

- (a) $r_1^* + r_2^* \neq (r_1 + r_2)^*$
- (b) $r_1^* r_2^* \neq (r_1 + r_2)^*$ $\rightarrow r_2 r_1$ $r_1=a$ $r_2=b$ } aba
- (c) $(r_1^* + r_2)^* = (r_1 + r_2)^*$
- (d) $(r_1 r_2)^* \neq (r_1 + r_2)^*$ \downarrow $r_1 r_2$ $r_2 r_1$

$$\begin{array}{ccc} \text{LHS} & & \text{RHS} \\ r_1 r_1 & = & r_1 r_2 \\ r_2 r_2 & & r_1 r_2 r_1 \end{array}$$

Q7 Let $r_1 = 0^* 1$ $r_2 = (0^* 1^*)^*$ which is correct $\Sigma = \{0, 1\}$

- (a) $L(r_1) = L(r_2)$
- (b) $L(r_1) \subset L(r_2)$
- (c) $L(r_1) \supset L(r_2)$
- (d) None

$$\begin{array}{l} 0^* 1 \\ \{ 1, 01, 001, 0001, \ldots \} \\ (0^* 1^*)^* \\ \{ \epsilon, 0, 1, 01, 001, \ldots \} \end{array}$$

- a) $L(r) \supseteq L(s)$
- b) $L(r) \subset L(s)$
- c) $L(r) \supset L(s)$
- d) None

$$L(r) = \{ \epsilon, a, b, ab, aab, \dots \}$$

$$L(s) = \{ \epsilon, a, b, ab, aab, \dots \}$$

$$ba \notin L(r)$$

$$ba \in L(s)$$

Q9) which of the following does not contain the string 1011

- a) $\emptyset^* (10)^* 1 \quad 1010101$
- b) $(\emptyset^* + 1^*)^* \quad 1011$
- c) $(110)^* 0^* 1$
- d) $\emptyset^* (101)^* 1$

Answer: a and c

Q10) which of the following REs does not generate string containing a substring '100'.

- a) $\emptyset^* (10)^* \quad = 000 \dots 101010 \dots$
- b) $\emptyset^* 1^* (10)^* 0 \quad = 10100$
- c) $(01^* + 0)^* (01)^* 1^* 0^* \quad = 0100 \dots$
- d) $(110^* + 1)^* 0 \quad = 11000 \dots$

Q11) which of the following REs does not generate string that contains 101 as substring

- a) $(101^*)^* 01^* (01)^*$ $101101 \dots$
- b) $(0^* + 1^*)^* (11)^*$ $1101 \dots$

- (A) $(0^* 1 + 11)^* 01^*$ 110111
 (B) $(1^* 0)^* (01)^*$ 110110

(C) None

Q12) Which of the following REs does not generate string that contains 110 as substring?

- (A) $(0^* 1^*)^*$ 011011
 (B) $(1^* 01)^0$ 1101
 (C) $0^* \underline{(11)^* 0^* 1^*}$ 1101
 (D) $(01)^* 0^* 1$ 010101000

Q13) Is $\underline{(r_1 r_2)^* r_1} = \underline{r_1 (r_2 r_1)^* ?}$

Shifting Rule of RE

$$\rightarrow 0 \in r_1 = r_1 \epsilon$$

$$1 \quad (r_1 r_2) r_1 = r_1 (r_2 r_1)$$

$$2 \quad r_1 r_2 r_1 r_2 r_1 = r_1 r_2 r_1 r_2 r_1$$

3 → :

Note: $(r_1 + r_2)^* = (r_1^* + r_2^*)^*$

$$\begin{aligned} &= (r_1^* + r_2^*)^* \\ &= (r_1 + r_2^*)^* \\ &= (r_1^* \cdot r_2^*)^* \end{aligned}$$

$$(r_1 + r_2)^* = (r_1^* \cdot r_2^*)^*$$

r₁ r₂ r₁ r₂ r₁ r₁

Mail: gatecse@appliedcourse.com

Powers of Alphabet:

Ph: 844-844-0102

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

$$\Sigma^1 = \Sigma = a+b \leftarrow \text{RE}$$

$$\Sigma^2 = \Sigma \cdot \Sigma = (a+b)(a+b) = (a+b)^2$$

$$\Sigma^3 = \Sigma \cdot \Sigma \cdot \Sigma = (a+b)(a+b)(a+b) = (a+b)^3$$

⋮

$$\Sigma^x = (a+b)^x$$

$$\Sigma^+ = (a+b)^+$$

Constructing RE:

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

- ① (a) RE that generates all strings including ϵ

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

- (b) All strings excluding ϵ

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

- ② Strings that start with 0 $\Sigma = \{0, 1\}$

$$r = 0(0+1)^*$$

- ③ Strings that start with '10'

$$r = 10(0+1)^*$$

- ④ Start with 0110

$$r = 0110(0+1)^*$$

⑤(a) Strings that end in '0'

$$r = (0+1)^* 0$$

(b) Strings that end in '10'

$$r = (0+1)^* 10$$

(c) strings that end in '1001'

$$r = (0+1)^* 1001$$

⑥(a) Contains substring '10'

$$r = (0+1)^* 10 (0+1)^*$$

$\rightarrow 10 \leftarrow$

(b) Contains substring '010'

$$r = (0+1)^* 010 (0+1)^*$$

⑦(a) strings starts & ends in 0

$\overbrace{01110}^{w=5}$

$$w=5$$

$$r = '0 (0+1)^* 0 + 0$$

⑦(b) strings starts & ends in same symbol

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

⑦(c) strings that start & end in different symbol

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

3rd symbol from left is 1

Ph: 844-844-0102

$$\underbrace{(0+1)}_1 \underbrace{(0+1)}_2 \underbrace{_}_3 \underbrace{_}_4 \underbrace{_}_5 \underbrace{_}_6 \dots \Rightarrow (0+1)^2 \ 1 \ (0+1)^*$$

(b) 3rd symbol from the right is 0

$$= \frac{0}{3} \ \frac{1}{2} \ \underline{1}$$

$$(0+1)^* 0 (0+1)^2$$

(9)(a) Strings that contain exactly two 0's. $\Sigma = \{0, 1\}$

$$r = 1^* 0 1^* 0 1^*$$

(b) atmost two 0's

$$r = 1^* (\epsilon + 0) 1^* (0+\epsilon) 1^*$$

(c) atleast two 0's

$$r = \underbrace{(0+1)^*}_0 \underbrace{0}_{(0+1)^*} \underbrace{0}_{(0+1)^*} (0+1)^*$$

(d) # zeros $\equiv 0 \pmod{3}$

$$0, \cancel{3}, \cancel{6}, \cancel{9}, \cancel{12}, \dots$$

$$1^* + \underbrace{(1^* 0 1^* 0 1^*)^*}_3$$

(e) # 1's $\equiv 2 \pmod{3}$

$$2, 5, 8, 11, \dots$$

$$r = \underbrace{0 1^* 0 1^*}_2 \underbrace{(0^* 1^* 0^* 1^*)^*}_3$$

(10)(a) $|w|=2$ $\Sigma = \{0, 1\}$

$$\underbrace{(0+1)}_1 \underbrace{(0+1)}_2 = (0+1)^2$$

(18) b

$$|W| \leq 2$$

$$\underline{(e+0+1)} \underline{(e+0+1)} = (0+1+e)^*$$

(c)

$$|W| \geq 2$$

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

$$\downarrow \\ (0+1)^2 (0+1)^*$$

$$\frac{1}{2} \quad \{0, 1, 2, \dots\}$$

(d)

$$|W| \equiv 0 \pmod{2}$$

$$0, 2, 4, 6, \dots$$

$$(0+1)^2)^*$$

$$\downarrow \\ 2x0 \quad 2x1$$

(e)

$$|W| \equiv 1 \pmod{2}$$

$$1, 3, 5, \dots$$

$$(0+1)((0+1)^2)^*$$

(f)

$$|W| \equiv 2 \pmod{3}$$

$$2, 5, 8, 11, \dots$$

$$\frac{(0+1)^2}{2} \frac{(0+1)^3)^*}{\{0, 3, 6, \dots\}}$$

(ii) a) $L = \{0^n \mid n \geq 0\} \Rightarrow 0^*$ $\Sigma = \{0\}$

b) $L = \{0^n \mid n \geq 1\} \Rightarrow 0^+$ $\Sigma = \{0\}$

c) $L = \{1^n \mid n \geq 3\} \Rightarrow$
 $1111^* \Rightarrow 111^*$

d) $L = \{0^{m,n} \mid m, n \geq 0\} \Rightarrow 0^* 1^*$

e) $L = \{0^{m,n} \mid m \geq 1, n \geq 1\} \Rightarrow [0^+ 1^+] \otimes [0^* 1^*]$

$L = \{0^m 1^n \mid m \geq 0, n \geq 1\}$
 $r = 0^* 1 1^*$
 $\Rightarrow 0^* + \text{PH: } 844-844-0102$

⑩ $L = \{0^m 1^n \mid m \geq 2, n \geq 3\} \quad r = 000^* 1111^*$

⑪ $L = \{0^m 1^n 2^p \mid m \geq 0, n \geq 1, p \geq 2\} \quad 0^* 11^* 222^*$
 \downarrow
 $0^* 1^+ 22^+$

⑫ $L = \{0^m 1^n \mid m+n = \text{even}\}$

$m = \text{even } 2x \quad m = \text{odd } 2x+1 \quad x \geq 0$

$n = \text{even } 2x \quad n = \text{odd } 2x+1$

$0^{2x}, 1^{2x} \quad 0^{2x+1}, 1^{2x+1}$

$(00)^x (11)^x \quad (00)^x 0 (11)^x 1$

 \downarrow

$$(00)^x (11)^x + (00)^x 0 (11)^x 1$$

⑬ $L = \{0^m 1^n \mid m+n = \text{odd}\}$

$m = \text{odd}$

$n = \text{even}$

 \textcircled{B}

$m = \text{even}$

$n = \text{odd}$

$2x+1$

$2x$

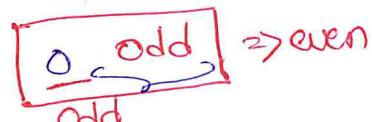
$2x$

$2x+1$

$$(00)^x 0 (11)^x + (00)^x (11)^x 1$$

⑭ @ strings start with '0' & $|w| = \text{even}$

$r = 0 (0+1)(0+1)^2 *$



Starts with '1' & $|w| = \text{odd}$

$$r = 1((0+1)^2)^*$$

Ph: 844-844-0102

$$\left| \begin{array}{c} \text{even} \\ 1 \end{array} \right| = \text{odd}$$

- (15) Starts with 0 & do not contain two consecutive ones.

$$r = 0 \rightsquigarrow$$

$$\times ((00)^* + (01)^* + (10)^*)^*$$

000110

(11)

$$\begin{aligned} & \{ \epsilon, 00, 0000, \dots \}^* \xrightarrow{(00)} \\ & \{ \epsilon, 01, 0101, \dots \}^* \xrightarrow{(01)} \\ & \{ \epsilon, 10, 1010, \dots \}^* \xrightarrow{(10)} \end{aligned}$$

$$0^t = \{0, 00, \dots\}$$

$$(01)^t = \{01, 0101, \dots\}$$

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

0 01

- (16) Strings do not contain two consecutive 0's or 1's.

$$(01)^* = \{ \epsilon, 01, 0101, \dots \}$$

$$(10)^* = \{ \epsilon, 10, 1010, \dots \}$$

$$r = (1+\epsilon)(01)^*(0+\epsilon)$$

$$+ (0+\epsilon)(10)^*(1+\epsilon)$$

- (17) String contains exactly two consecutive 1's.

$$r = 0^* 110^*$$

exactly

2 consecutive 1's.

$$0^* 10^* 10^*$$

exactly 2 1's

(18)

Strings of length ≤ 3 generated by $r = (1+0)^*$

Ph: 844-844-0102

$\text{len} = 0 \rightarrow \{\}$

$\text{len} = 1 \rightarrow \{0\} \rightarrow \textcircled{1}$

$\text{len} = 2 \rightarrow \{10\} \rightarrow \textcircled{1}$

$\text{len} = 3 \rightarrow \{110, 010\} \rightarrow \textcircled{2}$

4 distinct strings.

(19) # distinct strings of length ≤ 3 generated by $r = (0+01)^*$
 $(0+01)^* + (0+1)^*$

$\text{len} = 0 \quad \{\}$

$\text{len} = 1 \quad \{1\} \rightarrow \textcircled{1}$

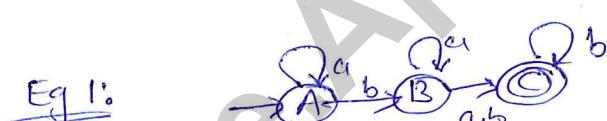
$\text{len} = 2 \quad \{01, 10, 11\} \rightarrow \textcircled{3}$

$\text{len} = 3 \quad \{001, 010, 011, 100, 101, 110, 111\} \rightarrow \textcircled{7}$

11 strings

(20)

Finite Automata - Regular Expression:



FA \leftrightarrow RE
 NFA, DFA

$$A = \epsilon + Aa \rightarrow \textcircled{1}$$

$$B = Ab + Ba \rightarrow \textcircled{2}$$

$$C = B(a+b) + Cb \rightarrow \textcircled{3}$$

Arden's Lemma / Rule

Let $R, P, Q \rightarrow \text{reg expression}$

$$R = Q + RP \rightarrow \text{linear equation in } R$$

$$X = B + XA \Rightarrow AX + B = X$$

Ph: 844-844-0102

Solution :-

$P = QP^*$, if P does not contain ϵ then

QP^* is the unique solution.

else $P = Q + RP$ has ω -many solutions

$$A = \epsilon + Aa$$

$$\Rightarrow A = \epsilon a^*$$

$$B = Ab + Ba$$

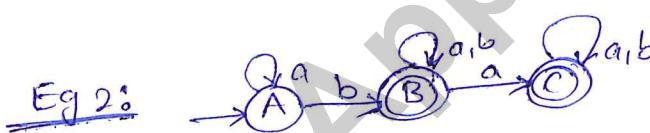
$$= a^*b + Ba$$

$$\Rightarrow B = a^*ba^*$$

$$C = B(a+b) + cb$$

$$C = a^*ba^*(a+b) + cb$$

$$\Rightarrow C = a^*ba^*(a+b)b^*$$



$$A = \epsilon + Aa \Rightarrow A = a^*$$

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

$R = \epsilon + a \quad RP$

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

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

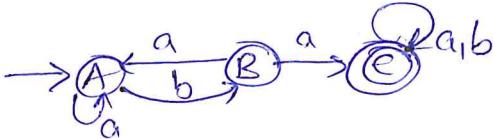
$$C = a^*b(a+b)^*a (a+b)^*$$

B and C are final states.

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

(8)

Eg 3: APPLIED COURSE



Ph: 844-844-0102

$$A = \epsilon + Aa + Ba \quad \text{--- (1)}$$

$$B = Ab \quad \text{--- (2)}$$

Substitute (2) in (1)

$$A = \epsilon + Aa + Aba$$

$$= \epsilon + A(a+ba)$$

$$A = \epsilon \cdot (a+ba)^*$$

$$\Rightarrow A = (a+ba)^*$$

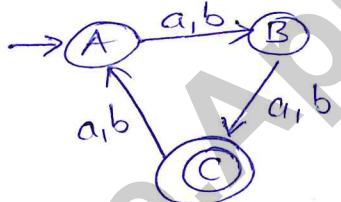
$$\Rightarrow B = (a+ba)^*b$$

$$C = Ba + C(a+b)$$

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

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

Eg 4:



$$A = \epsilon + C(a+b)$$

$$B = A(a+b)$$

$$C = B(a+b) = A(a+b)(a+b)$$

$$\Rightarrow A = \epsilon + C(a+b)$$

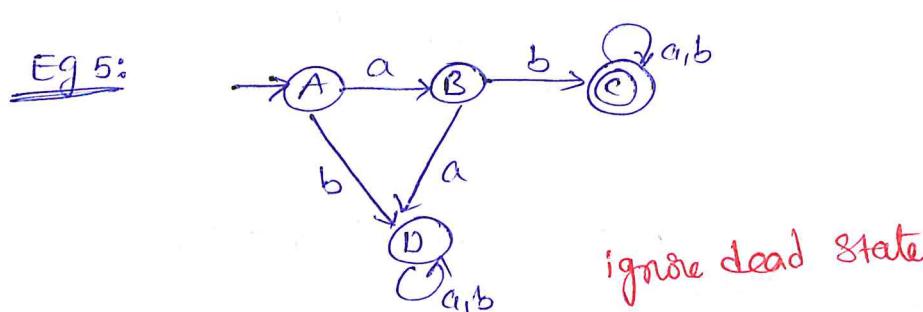
$$= \epsilon + A(a+b)^2(a+b)$$

$$= \epsilon + A(a+b)^3$$

$$\Rightarrow A = ((a+b)^3)^*$$

$$\Rightarrow B = ((a+b)^3)^*(a+b)$$

$c = ((a+b^3)^*)^* (a+b)^*$. len of string $\equiv_2 (mod^3)$



$$A = \epsilon +$$

$$B = Aa$$

$$C = Bb + C(a+b)$$

$$\Rightarrow A = \epsilon$$

$$B = a$$

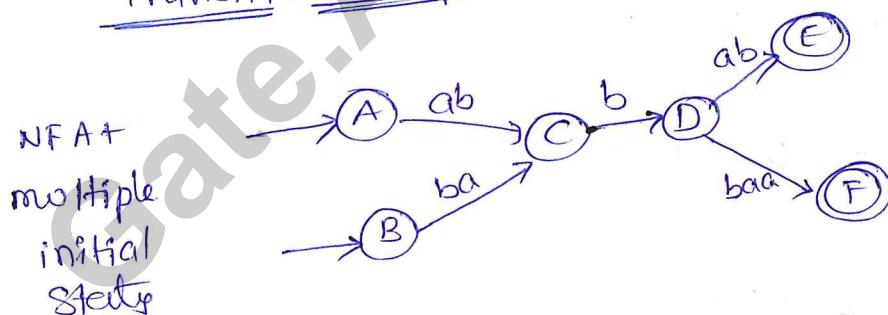
$$C = ab + C(a+b)$$

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

Elimination Method: (State)

$$E\text{-NFA} \rightarrow DE$$

Transition Graph: (TG)



+ transition on words

$$I = \{A, B\}$$

$$F = \{E, F\}$$

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

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

$$\delta: Q \times \Sigma^* \rightarrow 2^Q$$

$\epsilon\text{-NFA} \rightarrow \text{RE}$

DFA / NFA $\rightarrow \text{RE}$

Arden's Rule

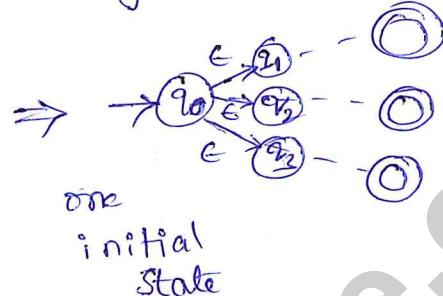
TG $\rightarrow \text{RE}$

Combine initial state using ϵ -transitions

Step 1:



T.G

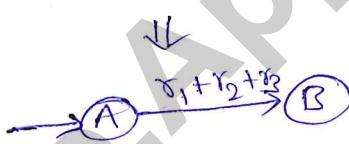
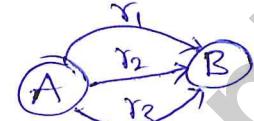


Step 2: separate out initial and final states using ϵ -Transitions

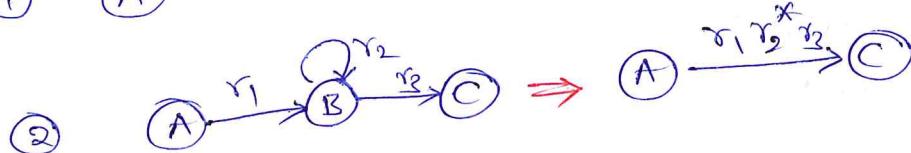
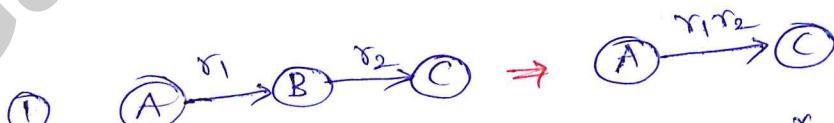


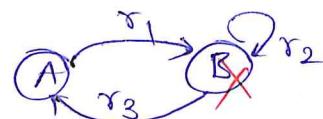
Step 3:

simplifying parallel edges.



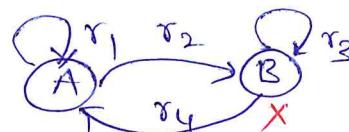
Step 4: Eliminating nodes / states





\Rightarrow $r_1 + r_2 r_3^*$

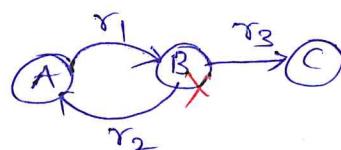
⑤



\Rightarrow

$r_1 + r_2 r_3^* r_4$

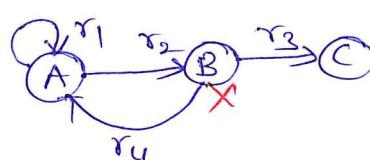
⑥



\Rightarrow

$r_1 r_2$
 $r_1 r_3 \rightarrow C$

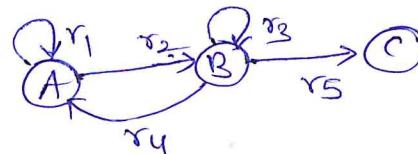
⑦



\Rightarrow

$r_1 + r_2 r_4$
 $r_2 r_3 \rightarrow C$

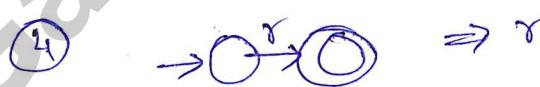
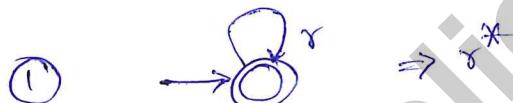
⑧



\Rightarrow

$r_2 r_3 r_5 \rightarrow C$
 $r_1 + r_2 r_3 r_4$

Step 5: TG_i is any of the following RE

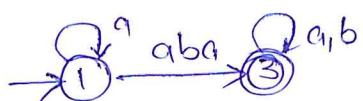


8)

Q1 APPLIED COURSE

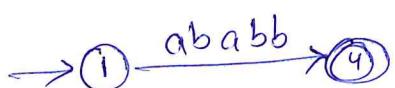
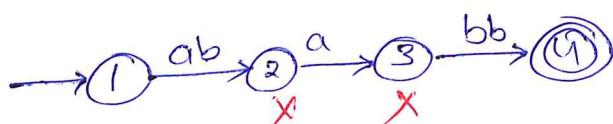


Ph: 844-844-0102



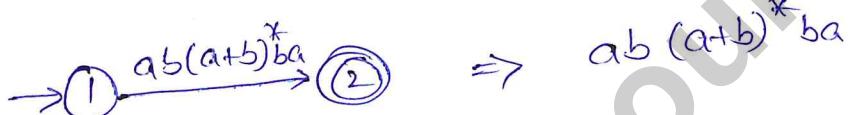
$$\Downarrow \\ a^* aba (a+b)^*$$

Q2

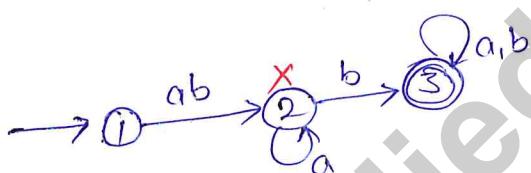


$$RE: ababb$$

Q3

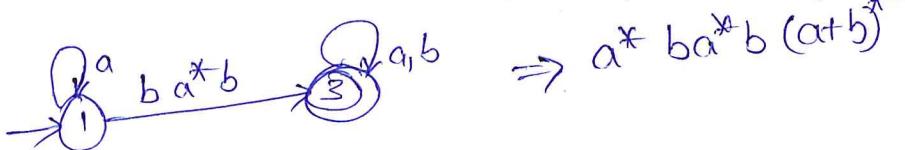
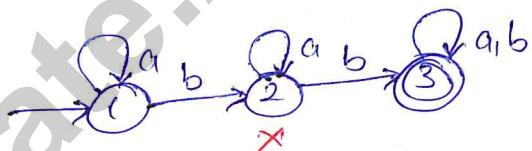


Q4



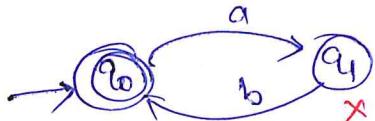
$$aba^*b$$

Q5



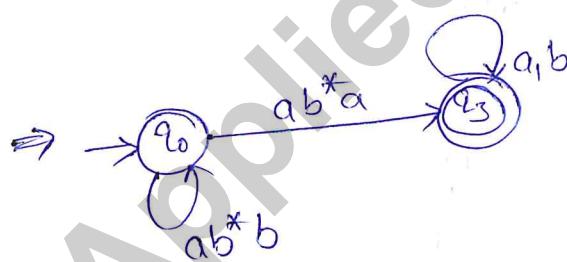
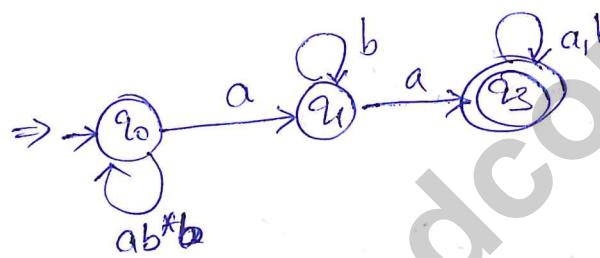
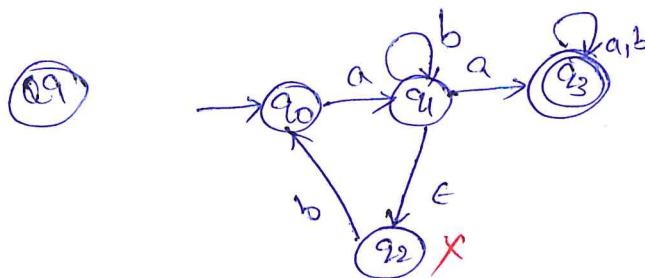
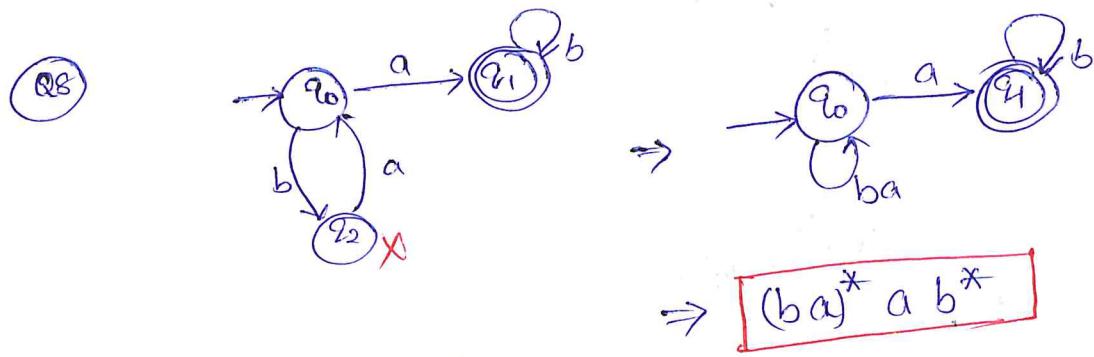
$$\Rightarrow a^* ba^* b (a+b)^*$$

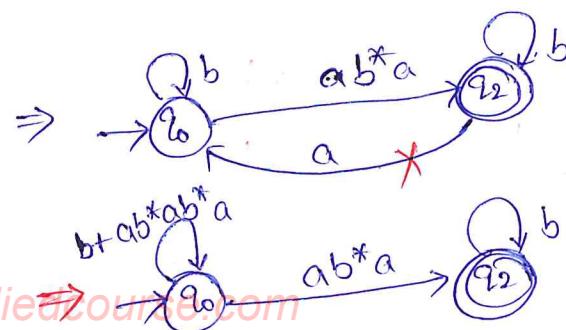
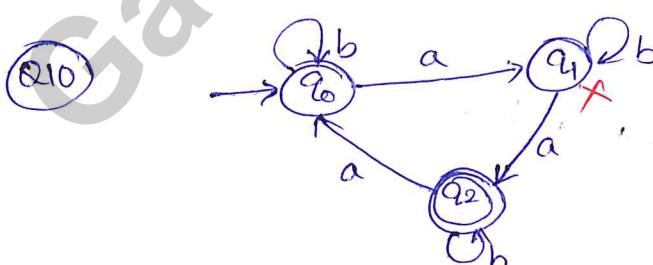
Q6



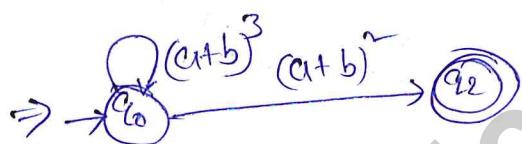
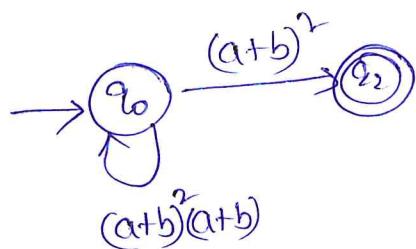
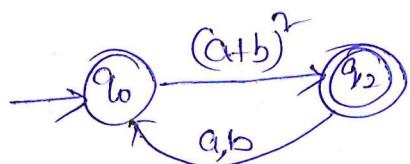
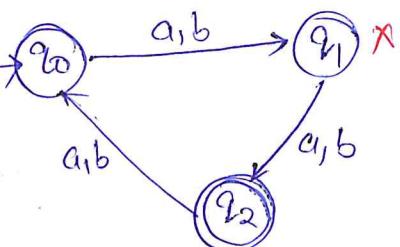
$$\Rightarrow (ab)^*$$

$\rightarrow q_0 \xrightarrow{ab} q_1 \xrightarrow{a} q_2 \Rightarrow (ab)^* a$ Ph: 844-844-0102

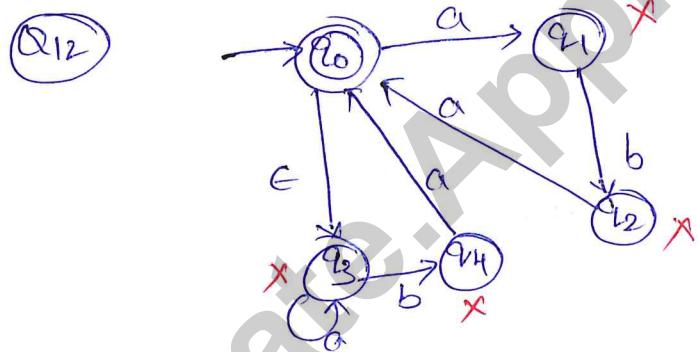


$$(ab^* b)^* ab^* a (a+b)^*$$


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



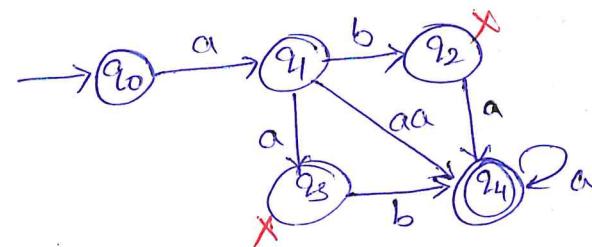
$$\Rightarrow ((a+b)^3)^* (a+b)^2$$



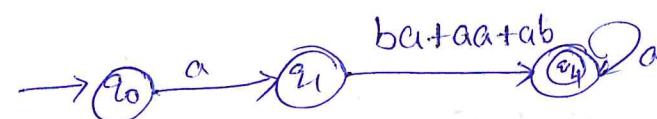
$$\Downarrow \quad aba + a^* b a$$



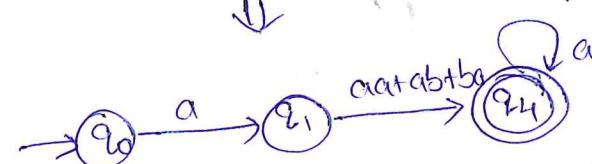
$$\Downarrow \quad \boxed{(aba + a^* ba)^*}$$



↓

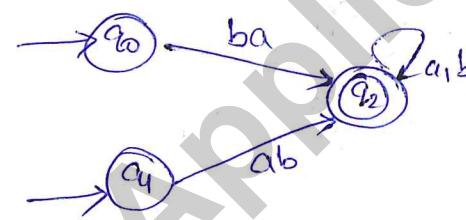


↓

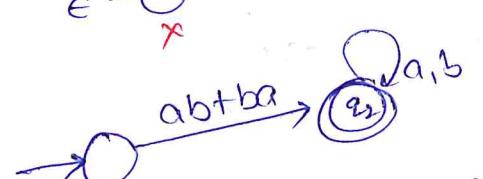
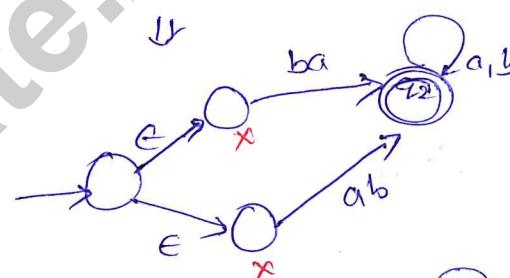


a(aat+bat+ab)*

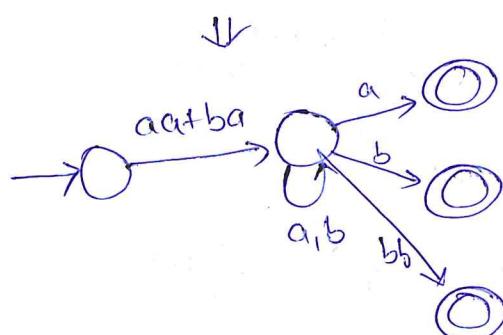
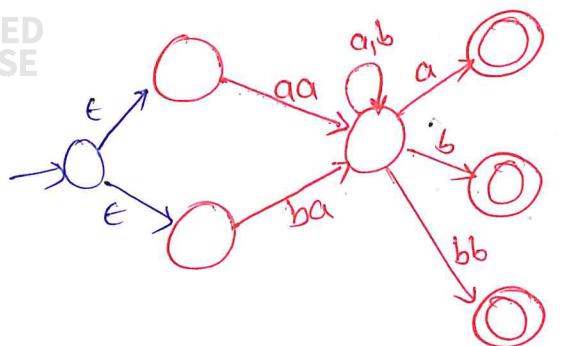
Q14



↓



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



$$(aa+ba)(a+b)^*(a+b+bb)$$

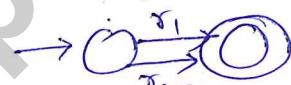
RE → FA

Regular Expression → Finite Automata

① Kleene closure: r^*



② Union: $r_1 + r_2$



③ Concatenation: $r_1 r_2$



RE → ENFA /
NFA /
DFA
← min
DFA

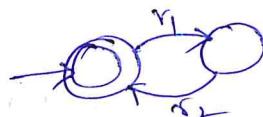
Eg 1:

$$(r_1 + r_2)^*$$



Eg 2:

$$(r_1 r_2)^*$$



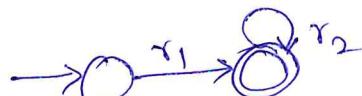
Eg 3:

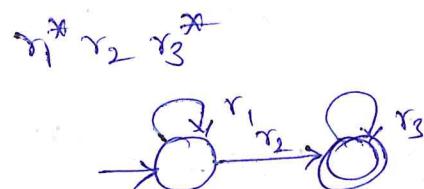
$$r_1^* r_2$$



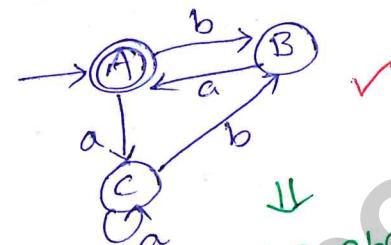
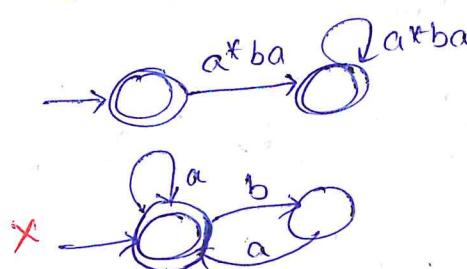
Eg 4:

$$r_1 r_2^*$$



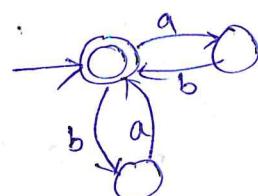


Eg 6:

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

FA accepts a, aa, \dots which are not generated by RE

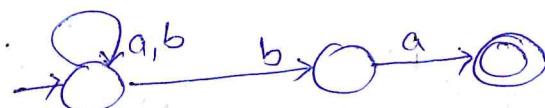
Eg 7:



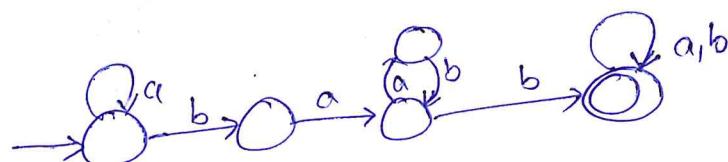
Eg 8:

 $a^*ba(ab)^*$ 

Eg 9:

 $(a+b)^*ba$ 

Eg 10:

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

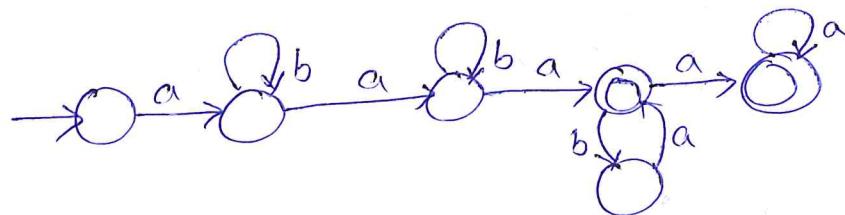
60

Eg 11
APPLIED
COURSE

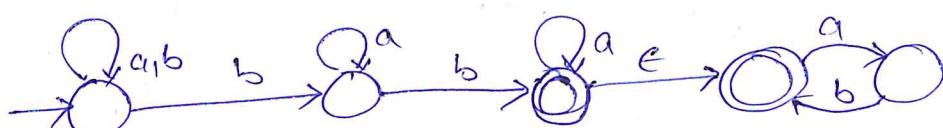
$a b^* a b^* a (ba)^* a^*$

decomposition RE \rightarrow FA

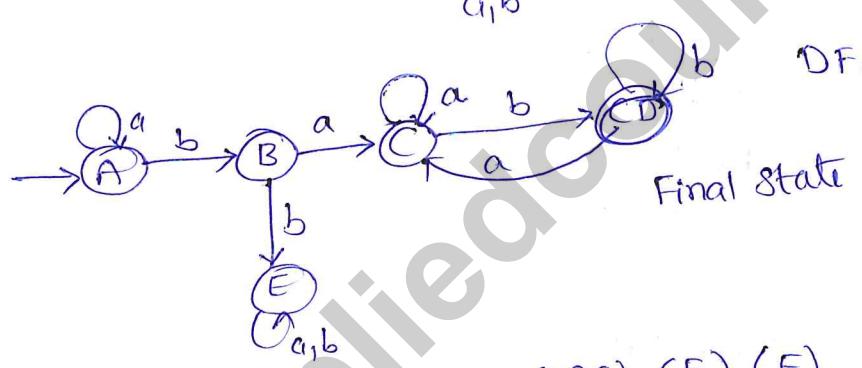
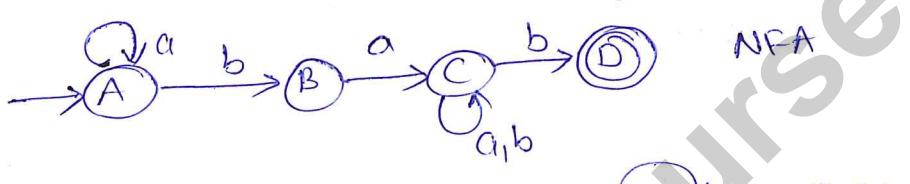
Ph: 844-844-0102



Eg 12: $(a+b)^* ba^* ba^* (ab)^* b$



Eg 13: min DFA for $a^* b a (a+b)^* b$

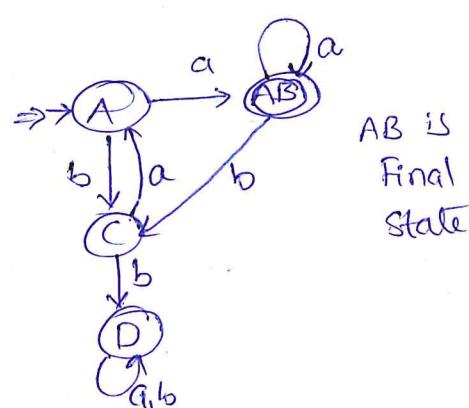
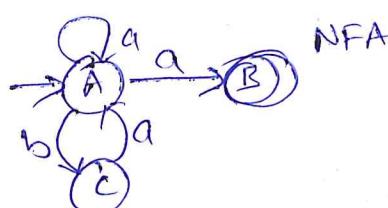


$(ABC) \ X \ (F) \ (E)$

$\frac{BC}{X} \quad \frac{AC}{X}$ say {By {C} } {F} {E}

minimal DFA
contains 5 states

Eg 14: Min DFA for $(a+ba)^* a$

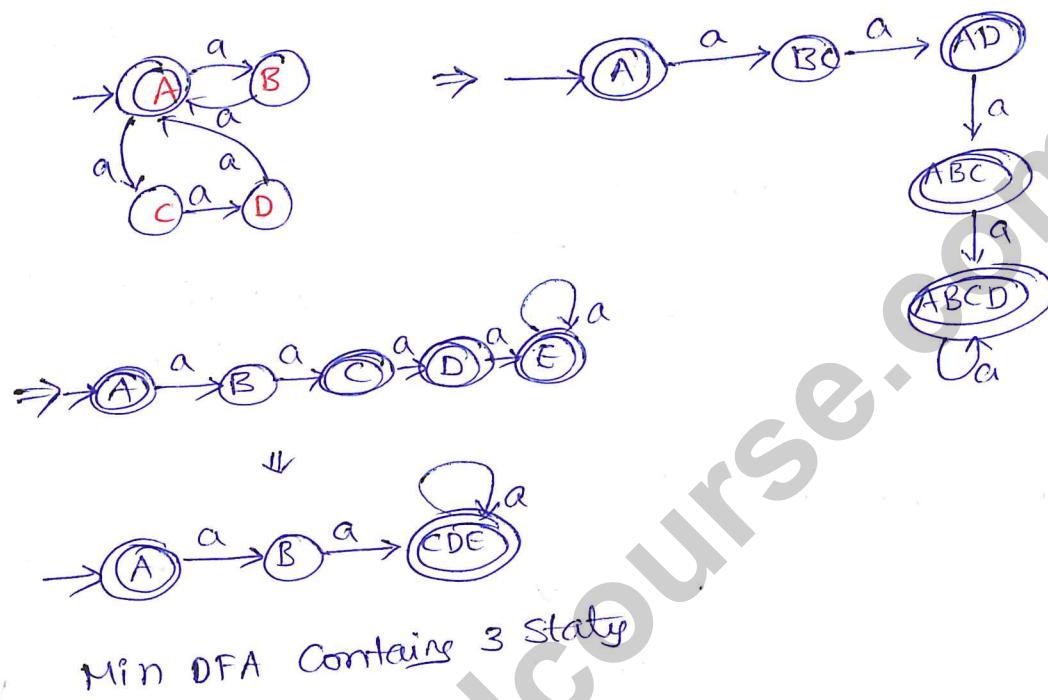


(AC) (D) (B)

Ph: 844-844-0102

(A)(C)(D)(B)

Eg 15: min DFA for $(aa+aae)^*$ $\Sigma = \{a\}$



Algebraic properties of RE operators:

* : Kleene closure unary

+ : union Binary

* : concatenation

DM: Algebraic structures

$\left. \begin{array}{l} \text{SS1, monoids} \\ \text{closure} \\ \text{Associativity} \\ \text{Identity} \end{array} \right\}$ groups
 $\left. \begin{array}{l} \text{abelian} \end{array} \right\}$

① Closure: If r_1, r_2 are RE

r_1^* is a RE

$r_1 + r_2$ is a RE

$r_1 \cdot r_2$ is a RE

② Associativity: If r_1, r_2, r_3 are RE

$$r_1 + (r_2 + r_3) = (r_1 + r_2) + r_3$$

$$r_1(r_2 \cdot r_3) = (r_1 \cdot r_2) r_3 \quad \checkmark$$

\times is a unary operator

③ Identity: $r + x = r \Rightarrow x = \phi$ if $\phi = r$ then ϕ

$$r \cdot x = r$$

is the identity

for +

$$\Rightarrow r \cdot x = r$$

$$x = e$$

$\Rightarrow r \cdot e = r \Rightarrow e$ is the identity element for \cdot

④ Annihilator:

$$r + x = x \Rightarrow r \cdot x = x \text{ No } \Rightarrow \exists \text{ an anti-annihilator}$$

$$r \cdot x = x$$

for +

$$\Rightarrow x = \phi$$

$$\{a, ab, ab\} - \{a\} = \{\}$$

ϕ is the annihilator for \cdot operation

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

⑤ Commutative property:

If r_1, r_2 are RE then

$$r_1 + r_2 = r_2 + r_1 \quad \checkmark \quad a+b=b+a$$

$$r_1 \cdot r_2 = r_2 \cdot r_1 \quad \times$$

$$ab = ba$$

⑥ Distributive property:

$$(r_1 + r_2) \cdot r_3 = r_1 \cdot r_3 + r_2 \cdot r_3 \quad \checkmark$$

Right distributive

$$r_1(r_2 + r_3) = r_1 \cdot r_2 + r_1 \cdot r_3 \quad \text{left distributive}$$

$r_1 = a$

$r_2 = b$

$r_3 = c$

$(ab) + c \neq (a+b)(b+c)$

$r_1 + (r_2 \cdot r_3) \neq (r_1 + r_2)(r_1 + r_3) \times$

⑦ Idempotent property:

$r_1 + r_1 = r_1 \Rightarrow r_1 \cup r_1 = r_1$

$r \cdot r \neq r \Rightarrow$ concatenation does not satisfy
 ↗ the idempotent prop
 Not always

Set

Algebraic structure \rightarrow Set, op \xrightarrow{PE} $\xrightarrow{+, \circ, \times}$ C closure $\leftarrow (R, \times)$

semigroup \rightarrow Associativity

monoid \rightarrow Identity $(R, +) \cdot (R, \circ)$

* Group \rightarrow Inverse does not exist for Regular Expression
 \downarrow
 $(\text{Real}, +)$ \curvearrowright addition

$r + (-r)$

$10 + (-10) = 0$

Abelian Group \rightarrow Commutative property

II

Not exist for PE

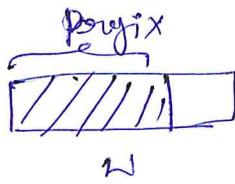
16
prefix operator & closure:

EDUCATIONAL
 COURSE

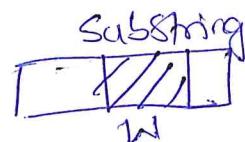
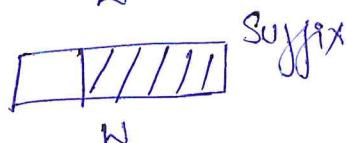
Ph: 844-844-0102

$L : RL$

$L = \{aba, aab, abab\}$

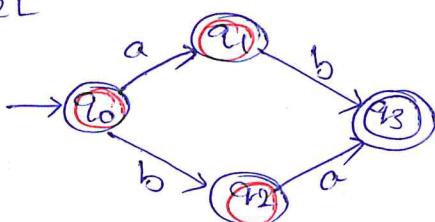


$L' = \{ab, a, aba, aa, aab, ab, aba, abab\}$



If L is a Regular Language, then prefix(L) is also Regular.

$L : RL$



$L' = \text{prefix}(L)$

$L = \{ab, b\bar{a}\} \Rightarrow L' = \{\epsilon, a, ab, b, b\bar{a}\}$

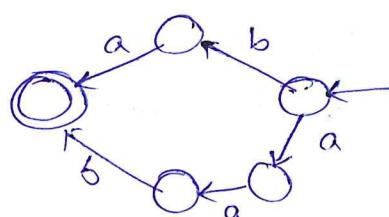
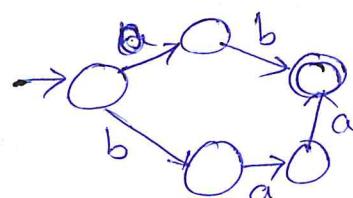
Reversal operator & closure:

w

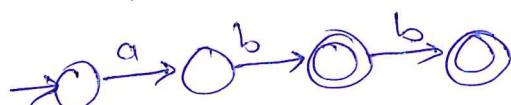
If L is a RL then $\text{rev}(L)$ is also a RL.

$L = \{ab, baa\} \rightarrow \text{FA}$

$\text{rev}(L) = \{ba, aab\}$



$L = \{ab, abb\} \quad \text{rev}(L)$



Quotient operator & closure:
Ph: 844-844-0102

$$\text{Let } L_1 = \{010, 101, 110, 001\}$$

$$L_2 = \{10, 01\}$$

$$\frac{L_1}{L_2} = \{0, 1\}$$

left cancellation

$$\frac{\cancel{010}}{\cancel{01}}$$

$$\frac{L_1}{L_2} = \{x \mid xy \in L_1 \text{ and } y \in L_2\}$$

right cancellation

$$\frac{L_1}{L_2} = \{x \mid xy \in L_1 \text{ and } x \in L_2\}$$

left cancellation.

If L_1, L_2 are RL then $\frac{L_1}{L_2}$ is also regular

Substitution:

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

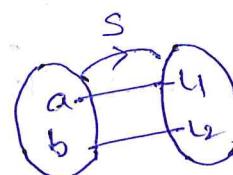
 Δ^* = set of all words using Δ
 2^{Δ^*} = power set of Δ^*

 = set of all languages using Δ

$$\{ \{ \}, \{ \} \}, \{ \{ \} \}, \dots \}$$

$$\{ \{ \epsilon, 0, 1 \}, \{ \epsilon, 00, 11, 10 \}, \dots \}$$

$$\delta: \Sigma \rightarrow 2^{\Delta^*}$$



$$L_1 \in 2^{\Delta^*}$$

$$L_2 \in 2^{\Delta^*}$$

18

$L_1 \& L_2$ are regular on Δ then language obtained

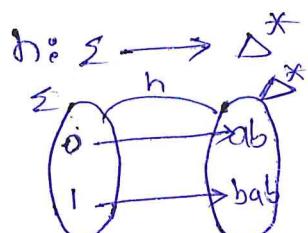
Ph: 844-844-0102

after substitution is also regular.

Homomorphism & closure: L is a RL over Σ & $h: \Sigma \rightarrow \Delta^*$

Special type of Substitution

then $h(L)$ is also a regular over Δ



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

$$h(0) = ab$$

$$h(1) = bab$$

$$L = \{ab\}$$

$$L \xrightarrow{h} h(L)$$

↓

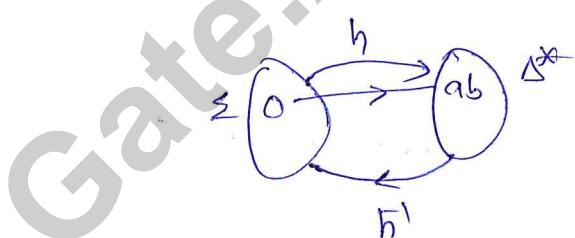
homomorphic image of L



Inverse Homomorphism & closure:

$$\begin{array}{c} \Sigma, \Delta \\ \uparrow \\ L \end{array} \quad h: \Sigma \rightarrow \Delta^*$$

$$h^{-1}: \Delta^* \rightarrow \Sigma$$



$$h^{-1}(L) = \{ x \in \Sigma^* \mid h(x) \in h(L) \}$$

Strings \rightarrow Symbols

If L is a RL then $h^{-1}(L)$ is also RL

RL or Non-RL? [pumping lemma for RL] 844-214-0102

For any RL, L ∃ an integer n → is dependant on L

$$L \subseteq \Sigma^*$$

so ∃ z ∈ L and |z| ≥ n

i) $z = uvw$

ii) $|uv| \leq |z|$

then $uv^iw \in L \forall i \geq 0$

iii) $|v| \geq 1 \& |v| \neq 0$

u, v, w, z are words built
using Σ

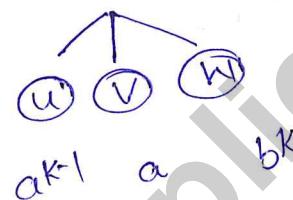
Eg 1: $L = \{a^m b^m \mid m \geq 1\}$ (n)

Let assume L is a RL

Proof

by

Contradiction $z = a^k b^k$ choose a K s.t. $|a^k b^k| = 2k \geq n$



$$|uv| = |a^{k-1}| + 1 = |a^k| = k \leq 2k$$

$$|v| \neq 0 \Rightarrow |v| = 1$$

Hence $uv^iw \in L \forall i \geq 0$

$$\text{for } i=2, a^{k-1} a^2 b^k = a^{k+1} b^k \notin L$$

∴ L is non-regular

$$uv^iw \Rightarrow uv^i w$$

$$uv^i w$$

$$uv^2 w$$

$$uv^3 w$$

:

pumping any number of w 's to

the Finite automata

(P)

Eg 2: APPLIED COURSE $L = \{ amb^p \mid m > p\}$

Ph: 844-844-0102

Let $L \in \text{a RL}$

$z \in L$

$z = a^{k+1} b^k$ choose k s.t. $|z| \geq n$



$$2k+1 \geq n$$

$a^{k+1} \ b \ b^{k-i}$

$$|uv| \leq |z|$$

$$|v| \neq 0$$

$uv^iw \in L \ \forall i \geq 0$

$$i=2 \Rightarrow a^{k+1} b^2 b^{k-1}$$

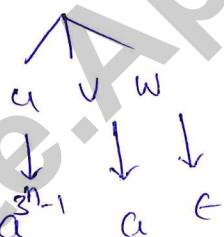
$$\Rightarrow a^{k+1} b^{k+1} \notin L$$

$\Rightarrow \boxed{L \text{ is not a Regular}}$

Eg 3:

$L = \{ a^{3^n} \mid n \geq 1\}$

$z \in L$



$$|uv| \leq |z| \checkmark$$

$$|v| \neq 0 \checkmark$$

$uv^iw \in L \ \forall i \geq 0$

$$i=2$$

$a^{3^{n-1}} \ a^2 \ \epsilon$

$\Rightarrow \boxed{a^{3^n+1} \notin L} \Rightarrow \text{Not Regular}$

Special case of pumping lemma when $\varepsilon = \{ab\}$ Singleton set
Ph: 840-844-0102

Lengths of the string must follow Arithmetic progression
for a L to be a RL

$u v^i w$

$$a(caa)^i a \Rightarrow \underbrace{aa}_2, \underbrace{aaaa}_4, \underbrace{aaaaaaaa}_6 \dots$$

Eg 1: $L = \{a^{3n} \mid n \geq 1\}$ $\Sigma = \{a\}$

RL

$$a^3, a^6, a^9, a^{12} \dots$$

$\swarrow \quad \swarrow \quad \swarrow$

Eg 2: $L = \{a^{2n+1} \mid n \geq 0\}$

$$1, 3, 5, 7, \dots \text{ AP}$$

Eg 3: $L = \{a^{n^2} \mid n \geq 0\}$ Non-regular

$$0, 14, 49 \dots$$

$\swarrow \quad \swarrow \quad \swarrow$

Eg 4: $L = \{a^{n^2+n+1} \mid n \geq 0\}$ Non-regular

$$1, 3, 13, \dots$$

$\swarrow \quad \swarrow \quad \swarrow$

Eg 5: $L = \{a^{2^n} \mid n \geq 0\}$ Non-regular

$$2, 4, 8, 16, \dots$$

$$1, 2, 4, 8, 16, \dots$$

Eg 6: $L = \{a^p \mid p \text{ is a prime}\}$

$$2, 3, 5, 7, 11, 13, 17, \dots \text{ AP}$$

Mail: gatecse@appliedcourse.com Non-regular

English Grammar: Rule to produce/generate correct language.

TG: production rules that generate words in a language.

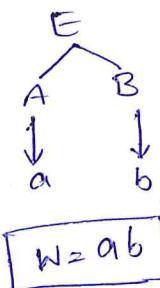
$$G_1: E \rightarrow AB \quad \text{variables: } E, A, B$$

$$A \rightarrow a \quad \text{Terminals: } a, b$$

$$B \rightarrow b$$

Starting Symbol: E

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



Language generated by the Grammar $L(G_1) = \{ab\}$

$$G = (V, T, P, S)$$

V: Variables

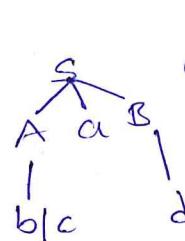
T: Terminals

P: productions

S: Start symbol

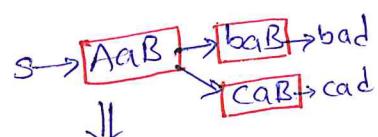
Eg 1:

$$\begin{array}{l} S \rightarrow AaB \\ \text{BNF} \quad A \rightarrow b \mid c \Rightarrow \begin{cases} A \rightarrow b \\ A \rightarrow c \end{cases} \\ \quad B \rightarrow d \end{array}$$



one starting symbol

Derivation Tree



Sentential form

(all intermediate steps)

Eg 2:

$$\begin{array}{l} G_1: \quad S \rightarrow AB \\ \quad \quad A \rightarrow a \\ \quad \quad B \rightarrow b \end{array}$$

G2:

$$\begin{array}{l} S \rightarrow Ab \\ A \rightarrow b \mid a \end{array}$$

$$L(G_2) = \{ab\}$$

$$L(G_1) = \{ab\}$$

$G_1 = G_2 \text{ iff }$

$$S \rightarrow ABC$$

$$A \rightarrow a$$

$$B \rightarrow b$$

$$C \rightarrow c$$

$$S \rightarrow \underset{a}{\downarrow} \underset{b}{\downarrow} \underset{c}{\downarrow} ABC$$

$$\Rightarrow S \rightarrow abc$$

$$L(G) = \{abc\}$$

Ph: 844-844-0102

Backus-Naur Form: (BNF)

$$A \rightarrow \alpha_1$$

$$A \rightarrow \alpha_2$$

$$A \rightarrow \alpha_3$$

$$A \rightarrow \alpha_1 | \alpha_2 | \alpha_3$$

BNF

Eg 1:

$$S \rightarrow aS$$

$$S \rightarrow b$$

 \Downarrow

$$S \rightarrow aS | b$$

Eg 2:

$$S \rightarrow AaB$$

$$A \rightarrow aA$$

$$A \rightarrow b$$

$$B \rightarrow d$$

$$B \rightarrow e$$

$$A \rightarrow aA | b$$

$$B \rightarrow d | e$$

Eg 3:

$$A \rightarrow a$$

$$A \rightarrow b$$

$$A \rightarrow \epsilon$$

 \Downarrow

$$A \rightarrow a | b | \epsilon$$

Recursive production:

$$A \rightarrow aA \text{ (Right)}$$

$$A \rightarrow Aa \text{ (Left)}$$

$$A \rightarrow aAb \text{ (General)}$$

$$A \rightarrow AaA \text{ (L&R)}$$

$$A \rightarrow ab \Rightarrow A \rightarrow abA \text{ (Indirect Recursion)}$$

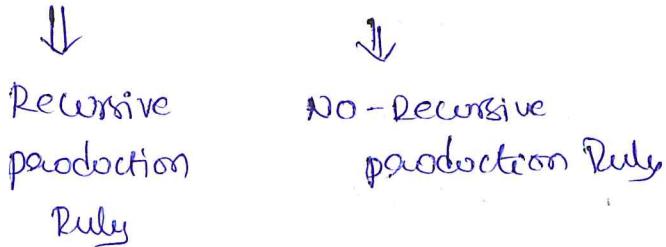
$$B \rightarrow bA$$

Grammar consisting of recursive production only

Recursive Grammar

Recursive & Non-Recursive Grammar:

Ph: 844-844-0102



Recursive Grammar
always generates
a language

Non-Rec Grammar
always generates
a finite language

Eg1: ① $A \rightarrow aA \mid b$

$$A \rightarrow aA \rightarrow aab$$

$$\{ b, ab, aab, \dots \} \boxed{a^*b}$$

② $A \rightarrow Aa \mid b$

③ $A \rightarrow aA \mid \epsilon$

④ $A \rightarrow Aa \mid \epsilon$

⑤ $A \rightarrow aA \mid a$

⑥ $A \rightarrow Aa \mid a$

⑦ $A \rightarrow aA \mid bA \mid \epsilon$

⑧ $A \rightarrow aA \mid bA \mid a \mid b$

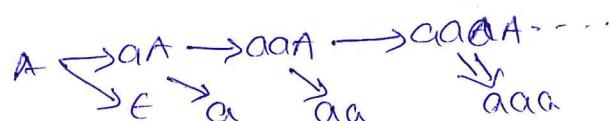
⑨ $A \rightarrow Aa \mid b$

$$\{ b, ba, baa, baaa, \dots \}$$



$L(G) = \boxed{bat}$

⑩ $A \rightarrow aA \mid \epsilon$



$$\{ \epsilon, a, aa, aaa, \dots \} = a^*$$

$$A \rightarrow Aa | \epsilon$$

$$\Rightarrow \boxed{a^*}$$

(5)

$$A \rightarrow Aa | a$$

$$\Rightarrow \boxed{a^+}$$

$$A \rightarrow aa | a$$

$$\Rightarrow \boxed{a^+}$$

(6)

$$A \rightarrow aa | bA | \epsilon$$

$$= \{ \epsilon, a, b, ab, bb, ba, bb, \dots \}$$

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

(7)

$$A \rightarrow aa | bA | a | b$$

$$= \{ a, b, aa, ab, ba, bb, \dots \}$$

$$\Rightarrow (a+b)^+$$

Solved Example / problems :

(Q1)

Find Language generated by

$$S \rightarrow AB$$

$$A \rightarrow a | b$$

$$B \rightarrow c$$

$$S \rightarrow aB$$

$$\rightarrow ac$$

$$S \rightarrow bB$$

$$\rightarrow bc$$

$$L(G) = \{ ac, bc \}$$

(Q2)

Language generated by

$$S \rightarrow AaB$$

$$A \rightarrow b$$

$$B \rightarrow cd$$

$$S \rightarrow \begin{matrix} Aa \\ \downarrow b \\ \wedge \\ B \\ \wedge \\ \downarrow cd \end{matrix}$$

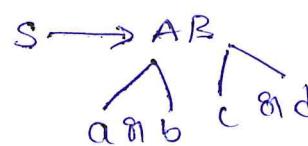
$$L(G) = \{ bac, bcd \}$$

Q3

$$S \rightarrow AB$$

$$A \rightarrow a|b$$

$$B \rightarrow c|d$$



$$\Rightarrow \{ac, ad, bc, bd\}$$

Q4 $S \rightarrow AB | BA$

$$A \rightarrow a|b$$

$$B \rightarrow c|d$$

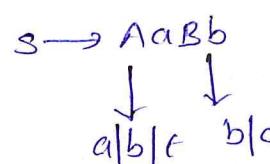


$$L(G) = \{ac, ad, bc, bd, ca, cd, da, db\}$$

Q5 $S \rightarrow AaBb$

$$A \rightarrow a|b|\epsilon$$

$$B \rightarrow b|c$$

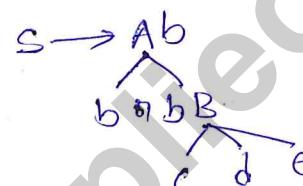


$$L(G) = \{aabc, babb, aacb, abb, abc\}$$

Q6 $S \rightarrow Ab$

$$A \rightarrow bB|b$$

$$B \rightarrow c|d|\epsilon$$



$$= \{bb, bcb, bd\}$$

Q7 $S \rightarrow aSb|\epsilon$

$$L(G) = \{\epsilon, ab, aabb, aaabbb, \dots\}$$

$$aSb$$

$$aasbb$$

$$aaasbbb$$

$$\downarrow \epsilon$$

$$\Rightarrow \{a^n b^n | n \geq 0\}$$

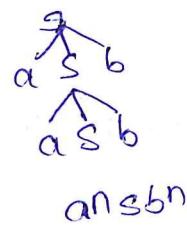
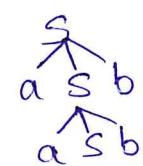
Q8 $S \rightarrow aasb|\epsilon$

$$L(G) = \{\epsilon, a^2b, a^4b^2, a^6b^3, \dots\}$$

$$\Rightarrow \{a^{2n}b^n | n \geq 0\}$$

$s \rightarrow asa | bsb | \epsilon$
Ph: 844-844-0102
 $= \{ \epsilon, aa, bb, abba, baab \dots \}$
 $= \{ w w^R | w \in (a+b)^* \}$

\Downarrow
 Palindrome

(Q10)
 $s \rightarrow asb | aAb$
 $A \rightarrow aA | \epsilon \Rightarrow a^*$
 $s \rightarrow asb | a a^* b$
 $s \rightarrow asb | a^* b$

 $a^n s b^n$
 $a^n a t b b^n$
 $a^n a a^* b b^n$
 $\Rightarrow a^{n+1} a^* b^{n+1}$
 $\Rightarrow \{ a^m b^n | m \geq n \}$
 $n \geq 1$
(Q11)
 $s \rightarrow asb | aBb$
 $B \rightarrow bB | \epsilon$
 $\Rightarrow \boxed{B \rightarrow b^*}$
 $s \rightarrow asb | a b^* b$

 $a^n s b^n$
 $a^n a b^* b b^n$
 $\Rightarrow a^{n+1} b^* b^{n+1}$
 $\Rightarrow \{ a^m b^n | n \geq m \geq 1 \}$

$$S \rightarrow asb | aAb$$

$$A \rightarrow cA | c$$

$$\Rightarrow A \rightarrow c^t$$

$$S \rightarrow asb | ac^{t+b}$$

$$a^n s b^n$$

$$a^n a c^t b b^n$$

$$\Rightarrow \{ a^m c^n b^m | m \geq 1, n \geq 1 \}$$

$$a^{n+t} c^t b^{n+1}$$

$$a^m c^t b^m \quad m \geq 1$$

$$S \rightarrow asa | bsb | c$$

$$asa \quad bsb \quad c$$

$$aca \quad bcb$$

$$\Rightarrow \begin{array}{c} asa \\ absba \\ ab \downarrow cba \end{array} L(G) = \{ w \in (a+b)^* \mid w \in (a+b)^* \}$$

Q13

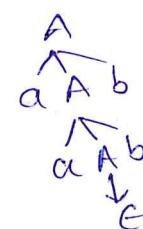
$$S \rightarrow AB$$

$$A \rightarrow aAb | \epsilon$$

$$B \rightarrow cB | \epsilon \Rightarrow \begin{array}{c} aAb \\ \downarrow \\ aAb \\ \downarrow \\ \epsilon \end{array}$$

$$\{ a^n b^n | n \geq 0 \}$$

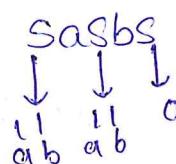
$$L = \{ a^n b^n c^m | m \geq 0, n \geq 0 \}$$



Q15

$$S \rightarrow sasbs | sbsas | \epsilon$$

$$\Rightarrow \{ \epsilon, ab, ba, \dots \} \Rightarrow L(G) = \{ w \in (a, b)^* \mid |wa| = |wb| \}$$



Q16

$$S \rightarrow asb | aAb$$

$$A \rightarrow cAd | cd$$

 \Rightarrow

$$S \rightarrow asb | P \text{ cmd } b^{m+1} 0102$$

$$A \rightarrow cd$$

$$\rightarrow ccd$$

$$\rightarrow cccddd \dots$$

 $a^n b^n$ $a^n a^m d^m b^m$ $a^{n+1} c^m d^m b^{m+1}$

$$A \rightarrow cmd^m | m \geq 1$$

$$\Rightarrow \{ a^k cmd^m b^k | m \geq 1, k \geq 1 \}$$

Designing Grammars for Languages - I

(a) Construct a grammar that generates

(a) all strings using $\Sigma = \{a, b\}$ including $\epsilon \Rightarrow (a+b)^*$

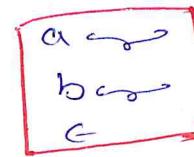
generate()
 ↳ recursive R
 ↳ base case b

(b)

$$S \rightarrow as | bs | \epsilon$$

excluding ϵ

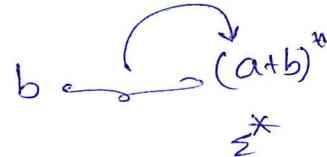
$$S \rightarrow as | bs | a | b$$



(a) @ Strings that start with 'b'

$$S \rightarrow bA$$

$$A \rightarrow aA | bA | \epsilon$$



(b) start with 'ab'

$$S \rightarrow abA$$

$$A \rightarrow \epsilon | aA | bA$$



(110)

③ @ LIED COURSE

Strings that end in 'a'

 $\Sigma^* a$: 844-844-0102

$$S \rightarrow Aa$$

$$A \rightarrow \epsilon | aA | bA$$

④ b. Strings that end in 'ab'

 $\Sigma^* ab$

$$S \rightarrow Aab$$

$$A \rightarrow \epsilon | aA | bA$$

④ a. Strings that contain 'ab' as Substring

$$S \rightarrow AabA$$

$$A \rightarrow \epsilon | aA | bA$$

 $\underline{\epsilon^*} \underline{ab} \underline{\epsilon^*}$

⑤ a. Strings that start & end in 'a'

$$S \rightarrow aAa | a$$

$$A \rightarrow a | bA | \epsilon$$

 $a (atb)^* a + a$

⑤ b. Starts & ends in same symbol

 $a \in^* a$

a

 $b \in^* b$

b

$$S \rightarrow alb | aAa | ab$$

$$A \rightarrow \epsilon | aA | bA$$

⑤ c. Starts & ends in different symbols

 $a \in^* b$
 $b \in^* a$

$$S \rightarrow aAb | bAa$$

$$A \rightarrow aA | bA | \epsilon$$

Third Symbol from left is 'b'

 $\rightarrow \Sigma^* b \leq^*$ 844-844-0102

$$\begin{aligned} S &\rightarrow A b B \\ A &\rightarrow a | a b | b a | b b \\ B &\rightarrow \epsilon | A B | b B \end{aligned}$$

$$\Rightarrow \begin{aligned} S &\rightarrow A A B B \\ A &\rightarrow a | b \\ B &\rightarrow \epsilon | a B | b B \end{aligned}$$

⑥(b) 4th symbol from right is 'a'.

$$\Sigma^* \xrightarrow[5]{\quad} \xrightarrow[4]{\quad} \xrightarrow[3]{\quad} \xrightarrow[2]{\quad} \xrightarrow[1]{\quad} \\ A \qquad \qquad \qquad \overbrace{B \qquad B \qquad B}$$

$$\begin{aligned} S &\rightarrow A a B B B \\ A &\rightarrow \epsilon | a A | b A \\ B &\rightarrow a | b \end{aligned}$$

⑦(a) #a's = 2 $b^* a b^* a b^*$

$$S \rightarrow B a B a B$$

$$B \rightarrow \epsilon | b B$$

⑦(b) #a's ≤ 2

$$b^* (a + \epsilon) \qquad b^* (a + \epsilon) \qquad b^* \\ \uparrow \qquad \qquad \qquad \uparrow \qquad \qquad \qquad \uparrow \\ B \qquad \qquad \qquad A \qquad \qquad \qquad ?$$

$$S \rightarrow B A B A B$$

$$A \rightarrow a | \epsilon$$

$$B \rightarrow \epsilon | b B | a B$$

⑦(c)

#a's ≥ 2

$$\Sigma^* a \Sigma^* a \Sigma^*$$

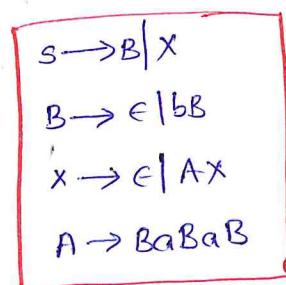
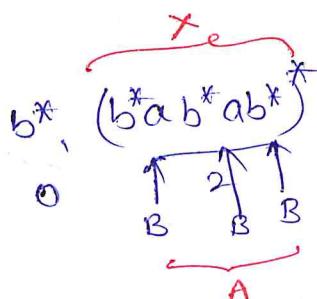
$$S \rightarrow A a A a A$$

$$A \rightarrow \epsilon | a A | b A$$

⑦(d) #a's = even

$$S \rightarrow B | B a B a B S$$

$$B \rightarrow \epsilon | b B$$



(11)

⑦ @APPLIED
COURSE

a's is odd

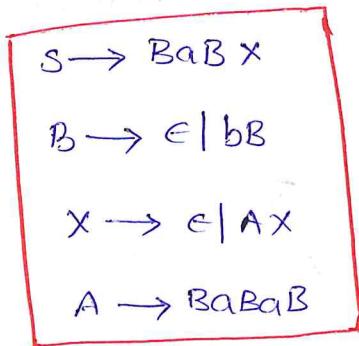
$$b^* ab^* (b^* a b^* a b^*)^*$$

1 0, 2, 4, 6, ...

Ph: 844-844-0102

Let $b^* \leftarrow B$ $x = A^*$ $A_2 = b^* ab^* ab^*$

BABAB

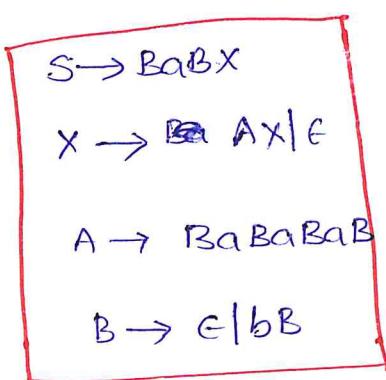


⑧

a's $\equiv 1 \pmod{3}$

1, 4, 7, 10, ...

1 + 0, 3, 6, 9, ...



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

↑ A
B X

⑧ @

|w|=2

$$\begin{array}{l} S \rightarrow AA \\ A \rightarrow a/b \end{array}$$

 $\overline{A} \overline{A}$

⑨

|w| ≤ 2

$$\begin{array}{l} S \rightarrow AA \\ A \rightarrow a/b/\epsilon \end{array}$$

— —
A A
↓
a/b/ε

⑩

|w| ≥ 2

 $\overline{A} \overline{A} \frac{\Sigma^*}{B}$

$$S \rightarrow AAB$$

$$\begin{array}{l} A \rightarrow a/b \\ B \rightarrow aB/bB/\epsilon \end{array}$$

$|W| \equiv 0 \pmod{2}$ $0, 2, 4, 6, \dots$

Ph: 844-844-0102

$$\begin{array}{l} S \rightarrow \epsilon | X \\ X \rightarrow AX | \epsilon \\ A \xrightarrow{\text{a}} BB \\ B \xrightarrow{\text{b}} a | b \end{array}$$

$$\begin{array}{c} [(a+b)^2]^* \\ (- -)^* \\ \uparrow \quad \uparrow \\ a | b \quad a | b \end{array}$$

 $\textcircled{8} \textcircled{c} \quad |W| \equiv 1 \pmod{2}$

$$\begin{array}{l} S \rightarrow AX \\ A \xrightarrow{\text{a}} ab \\ B \xrightarrow{\text{b}} AA \\ X \rightarrow \epsilon | XB \end{array}$$

$$\begin{array}{c} -(-)^* \\ \overbrace{AA}^B \end{array}$$

 $\textcircled{8} \textcircled{d} \quad |W| \equiv 2 \pmod{3}$ $2, 5, 8, \dots \Rightarrow 2 + 0, 3, 6, 9, \dots$

$$\begin{array}{c} (- -) (- -)^* \\ (AA)(AAA)^* \end{array}$$

 $\textcircled{9} \textcircled{a} \quad \{amb^n \mid m, n \geq 0\}$

$$\begin{array}{l} S \rightarrow AB \\ A \xrightarrow{\text{a}} \epsilon | aA \\ B \xrightarrow{\text{b}} \epsilon | bB \end{array}$$

 $\textcircled{9} \textcircled{b} \quad \{amb^n \mid m, n \geq 1\}$

$$\begin{array}{l} S \rightarrow AB \\ A \xrightarrow{\text{a}} a | aA \\ B \xrightarrow{\text{b}} b | bB \end{array}$$

$S \rightarrow AB$ $A \rightarrow a \mid aa$ $B \rightarrow bC$ $C \rightarrow b \mid bc$
 $B \rightarrow bC^*$
 \downarrow
 b^*
 $\{bb, bbb, \dots\}$ Q1d) $\{amb^n \mid m+n \text{ is even}\}$

m: even

n: even

m: odd

n: odd

 $(\underbrace{aa}_A)^* (\underbrace{bb}_B)^*$
 $\underbrace{a(aa)_A^*}_{\sim} (\underbrace{bb}_B)^* b$
 $S \rightarrow XY \mid axby$ $X \rightarrow \epsilon \mid AX \rightarrow A^*$ $A \rightarrow aa$ $B \rightarrow bb$ $Y \rightarrow \epsilon \mid BY \rightarrow B^*$ Q2) $\{amb^n \mid m+n \text{ is odd}\}$

odd+even

even+odd

 $(\underbrace{aa}_A)^* a (\underbrace{bb}_B)^*$
 $(\underbrace{aa}_A)^* (\underbrace{bb}_B)^* b$

$S \rightarrow Xay \mid xby$
 $A \rightarrow aa$
 $B \rightarrow bb$
 $X \rightarrow \epsilon \mid XA$
 $Y \rightarrow \epsilon \mid YB$

Q2a) $\{amb^n \mid m=n\} \quad \{ \in, \frac{ab}{S}, \frac{aabb}{S}, \dots \}$ $S \rightarrow \epsilon \mid aSb$

10(b) $\{amb^n \mid m=2n\}$

Ph: 844-844-0102

$= \{\epsilon, aab, aaaabb, \dots\}$

$$S \rightarrow \epsilon \mid aab$$

10(c)

$\{amb^nc^n \mid m,n \geq 1\}$

$$S \rightarrow AB$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bBc \mid bc$$

$$\begin{array}{l} A \\ \downarrow \\ a^m \mid m \geq 1 \end{array}$$

$$\begin{array}{l} B \\ \swarrow \searrow \\ b^nc^n \mid n \geq 1 \end{array}$$

10(d)

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

$$S \rightarrow aSc \mid abc$$

$$B \rightarrow b \mid bb$$

10(e)

$\{amb^nc^p \mid n=m+p; n, m, p \geq 0\} \quad \subseteq \{a, b, c\}$

$= \{\epsilon, abbc, ab, bc, \dots\}$
 $aabbcc$

$$\boxed{S \rightarrow AB \\ A \rightarrow \epsilon \mid aAb \\ B \rightarrow \epsilon \mid bBc}$$

$amb^m \mid m \geq 0$

$b^p c^p \mid p \geq 0$

11(a)

$L = \{amb^n \mid m > n; m, n \geq 1\}$

$$\boxed{S \rightarrow asb \mid aAb \\ A \rightarrow a \mid aa}$$

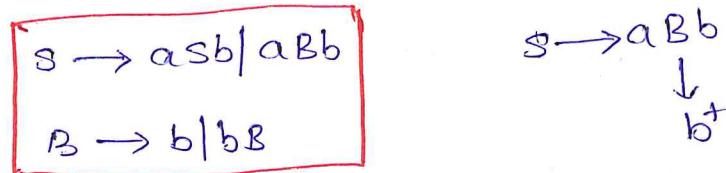
$a^n b^n$
 $a \xrightarrow{s} b$
 \downarrow
 a, aa, \dots

$s \rightarrow asb$
 $aasb$
 $aaaAbbb$

$A \rightarrow at \Rightarrow \boxed{aaaaabbb}$

$\{amb^n \mid m < n, m, n \geq 1\}$

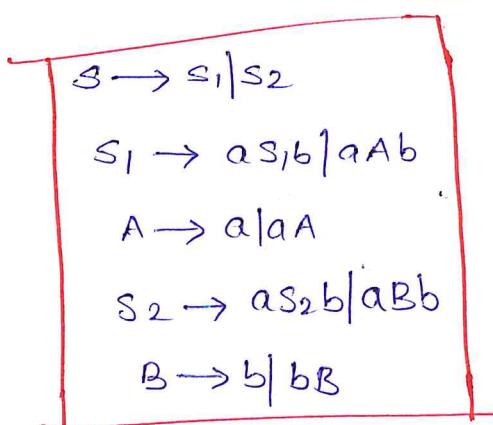
Ph: 844-844-0102



$$S \rightarrow aBb \downarrow b^+$$

⑪ C

$\{amb^n \mid m \neq n, m, n \geq 1\}$



⑫ @

$\{wcyw^R \mid w, c \in \{a, b\}^*\}$

$$\boxed{\begin{array}{l} w \subseteq w^R \\ \downarrow \\ x \end{array}}$$

$$S \rightarrow aSa \mid bSb \mid aAa \mid bAb \mid \epsilon \mid ab$$

$$A \rightarrow aAb \mid bAa$$

b)

$\{wcyw^R \mid w \in \{a, b\}^*\}$

$$S \rightarrow aSa \mid bSb \mid c$$

c)

$\{ww^R \mid w \in \Sigma^*\}$ $\Sigma = \{a, b\}$

$$S \rightarrow aSa \mid bSb \mid \epsilon$$

⑬ @

$\{w \in \Sigma^* \mid |wa| = |wb|\}$ $\Sigma = \{a, b\}$

$$S \rightarrow sasbs \mid sbsas \mid \epsilon$$

$\{ w \in \Sigma^* \mid |w_a| = 2|w_b| \}$

Ph: 844-844-0102

$$S \rightarrow S a S a b S \mid S a S b S a S$$

$$S b S a S a S \mid \epsilon$$

$$\begin{array}{c} \underline{S} \underline{a} \underline{S} \underline{a} . \underline{S} \underline{b} \underline{S} \\ \underline{S} \underline{a} \underline{s} \underline{b} \underline{\quad} \underline{s} \underline{a} \underline{s} \\ \underline{s} \underline{b} \underline{s} \underline{a} \underline{\quad} \underline{s} \underline{a} \underline{s} \end{array}$$

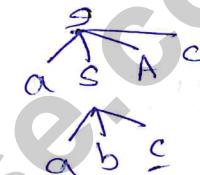
(14) $\{ a^n b^n c^n \mid n \geq 1 \}$

abc, aabbcc, aaabbcc, ...

$$\boxed{S \rightarrow abc \mid a \underline{S} A c \quad ①}$$

$$\underline{c A} \rightarrow \underline{A} c \quad ②$$

$$\underline{b A} \rightarrow \underline{b} b \quad ③$$



Best / hardest example

$$\begin{aligned} S &\rightarrow a S A C \\ &\rightarrow a a b c \underline{A} c \\ &\rightarrow a a b \underline{A} c c \\ &\rightarrow a a b b c c \checkmark \end{aligned}$$

Chomsky hierarchy of Grammars & Languages: Noam Chomsky

Type 0 : Recursively

Enumerated Grammar : PEL \rightarrow TM : Computer

Type 1 : CSG : CSL \leftrightarrow LBA \rightarrow syntax & semantic checks

syntax check \uparrow CD

Type 2 : CFG : CFL \leftrightarrow PushDown Automata \rightarrow FA + stack

Type 3 : RG : Regular Languages \leftrightarrow FA

\downarrow
pattern matching

Type - 0 | Recursively Enumerated | Unrestricted Grammar:

Ph: 844-844-0102

 V : VariablesREL
TM T : Terminals

$\alpha \rightarrow \beta$

$\alpha \in (V+T)^*$

$\beta \in (V+T)^*$

$$\begin{array}{l} \text{REG} \\ S \rightarrow aAB \\ aA \rightarrow aB/b \\ Ba \rightarrow b \end{array}$$

 s : Start Symbol

REL: $L = \{a^n! | n \geq 0\}$

Type - 1 | Context-Sensitive Grammar (CSG)

CSL | LBA

$\forall \alpha \rightarrow \beta \quad \alpha \in (V+T)^*$

$\beta \in (V+T)^*$

① $|\alpha| \leq |\beta|$

② $\beta \neq \epsilon$

Eg:

$$\begin{array}{l} S \rightarrow aAB \\ aA \rightarrow aB/bb \\ Ba \rightarrow bb \\ \boxed{Ba \rightarrow b} \times \end{array}$$

Eg: $L = \{a^n b^n c^n | n \geq 1\}$

$$\begin{array}{l} S \rightarrow abc | aSAc \\ cA \rightarrow Ac \\ bA \rightarrow bb \end{array}$$

$$\alpha \rightarrow \beta$$

$$\Rightarrow \alpha \Rightarrow v$$

$$\beta \rightarrow (v+T)^*$$

β can be ϵ

Eg: $s \rightarrow a s a | \epsilon$

$b b \rightarrow b b$ X

$$CFL : \{a^n b^n | n \geq 1\}$$

Type-3 | Regular Grammar:

$$RG \rightarrow RL \rightarrow FA$$

$$A \rightarrow \alpha B$$

$$A \rightarrow \alpha$$

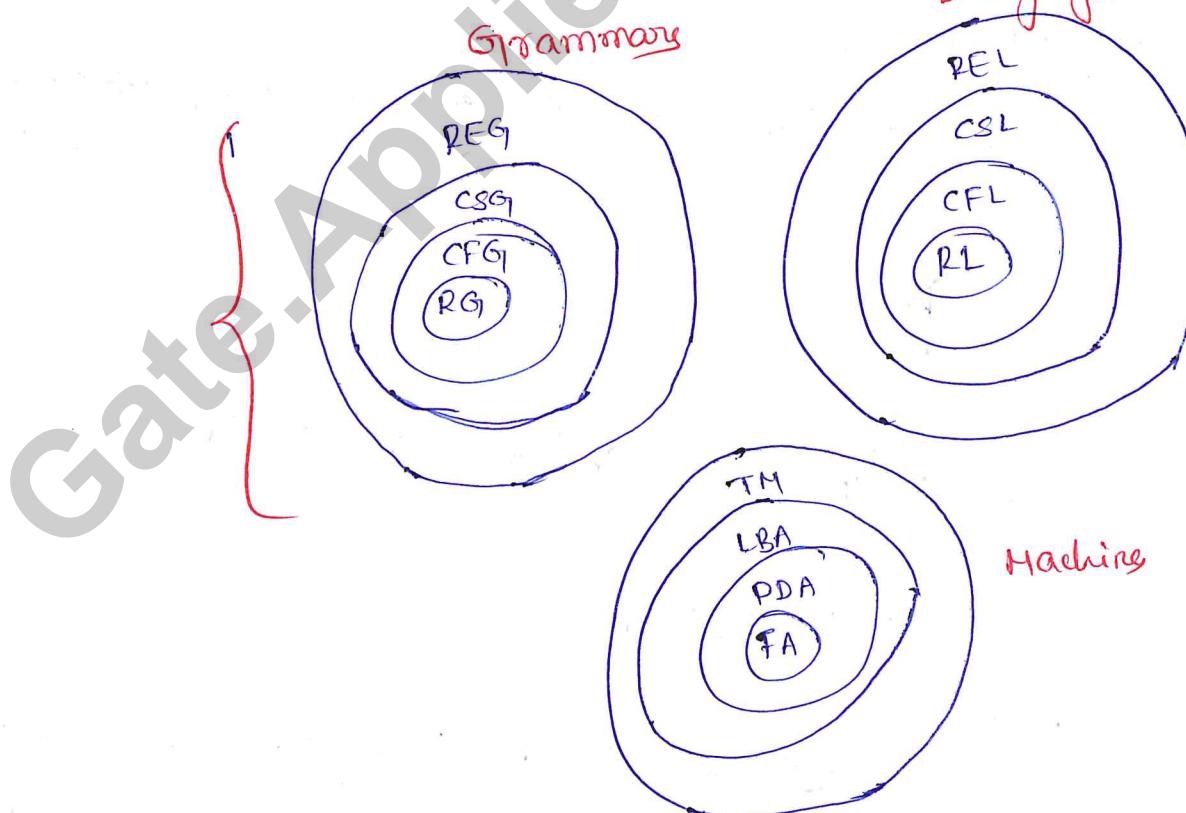
$$B \rightarrow B\alpha$$

$$\alpha, B \in V$$

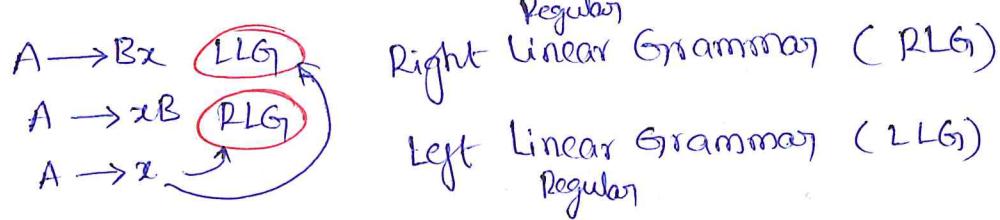
$$\alpha \in T^*$$

\hookrightarrow String of terminals

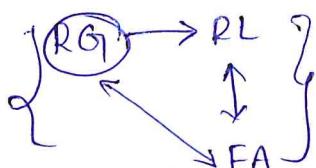
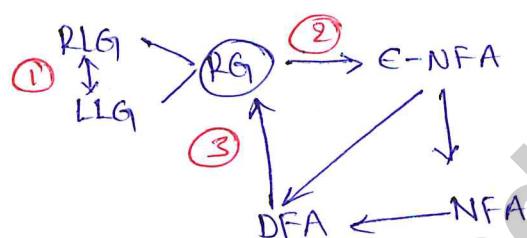
Eg: $s \rightarrow 001s / 10 / \epsilon$



(120)


 $A, B \in V$
 $x \in T^*$

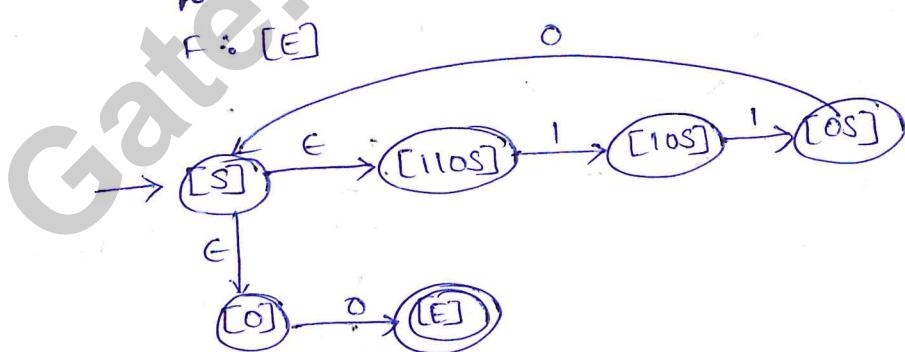
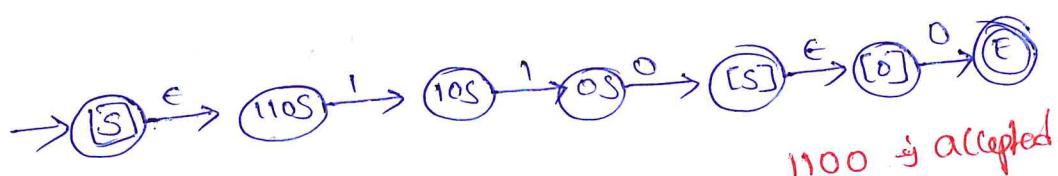
$$RLG \leftrightarrow LLG$$


Equivalence of RG & FA:

① RLG to E-NFA:

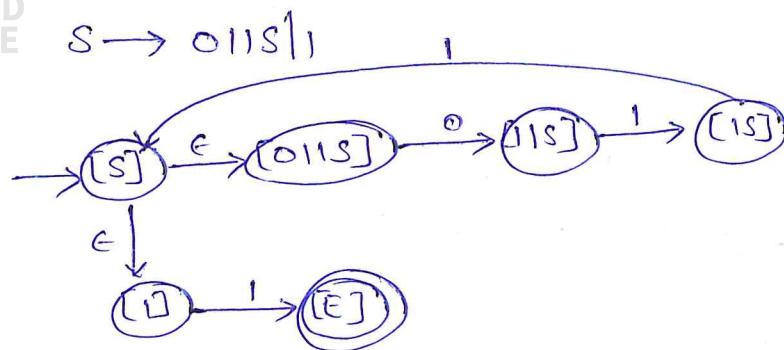
$s \rightarrow 110s \mid 0$

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

$\Omega = \{[S], [110s], [10s], [0s] \mid 0, [E]\}$
states

 $\overset{1}{\text{to}}: [S]$
 $F: [E]$

 1100


$L(G) = \{0, 1100, 1101100, \dots\}$



$$L(G) = \{1, 011, 011011, \dots\}$$

$$L(M) = \{1, 011, 011011, \dots\}$$

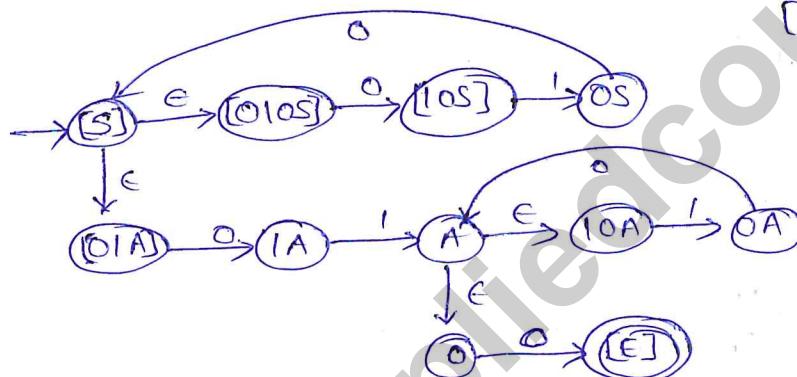
(3) $S \rightarrow 010S|01A$

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

$$A \rightarrow 10A|0$$

$$\Omega = \{[S], [010S], [10S], [0S],$$

$$[01A], [1A], [A], \\ [10A], [0A], [E]\}$$



Equivalences : RLG \leftrightarrow LLG \leftrightarrow FA

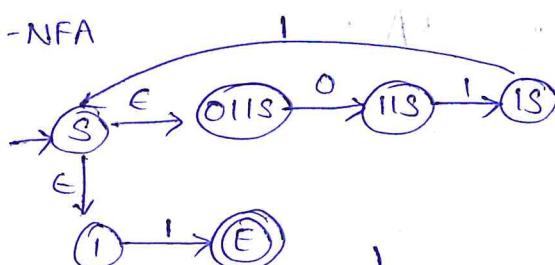
LLG to E-NFA

① $S \rightarrow S110|1$

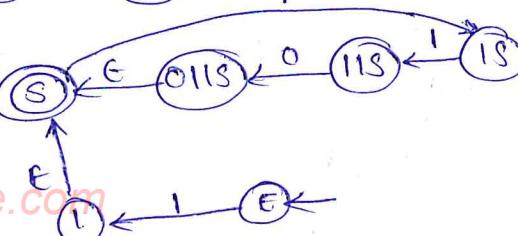
Step ① : Reverse it $\Rightarrow S \rightarrow 011S|1$ RLG

Step ②

E-NFA

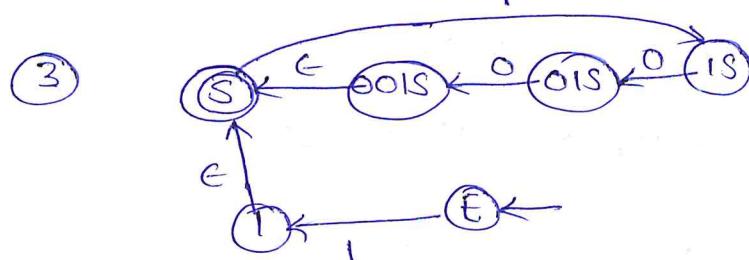
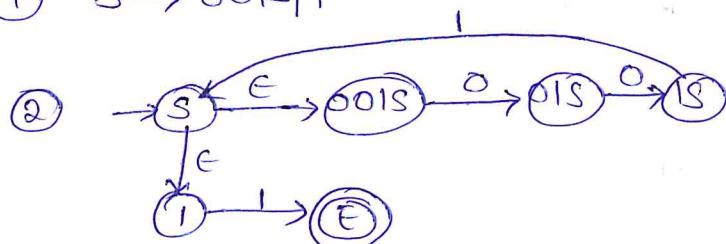


Step ③ :

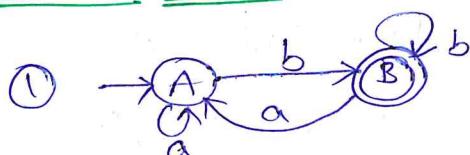


\leftarrow E-NFA
to
LLG

① $S \rightarrow 001S|1$

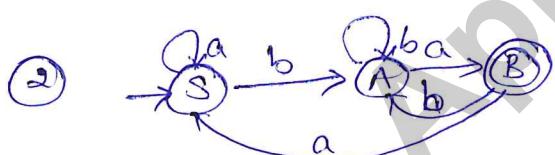


DFA to RLG:



$\Rightarrow A \rightarrow aA|bB$

$B \rightarrow bB|aA$
 $B \rightarrow \epsilon$ (B is the final state)



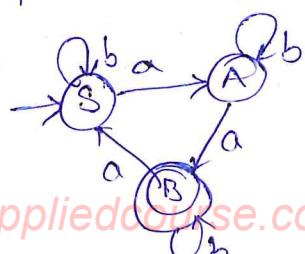
$\Rightarrow S \rightarrow aS|bA$

$A \rightarrow bA|aB$

$B \rightarrow bB|aS$

$B \rightarrow \epsilon$

③ $|w_A| \equiv 2 \pmod{3}$

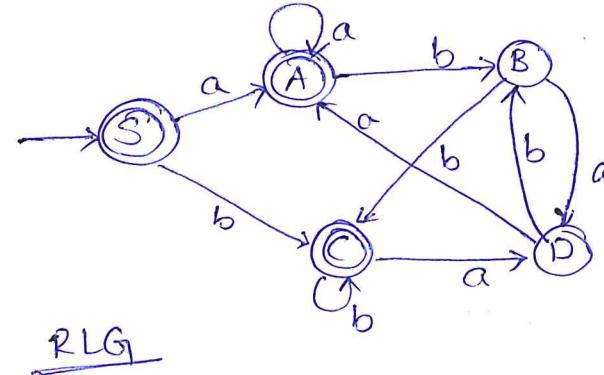


$S \rightarrow aA|bS$

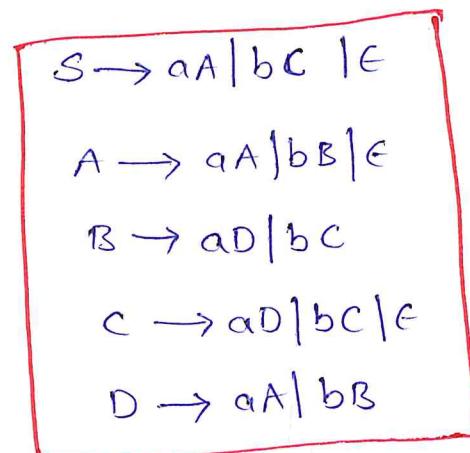
$A \rightarrow aB|bA$

$B \rightarrow aS|bB$

$B \rightarrow \epsilon$

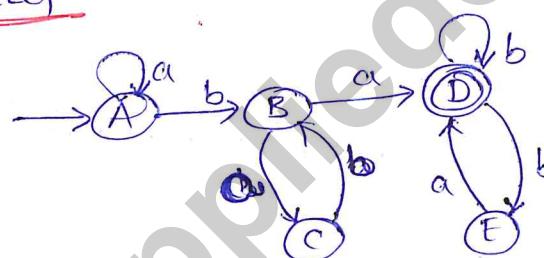


RLG



FA \rightarrow LLG

①

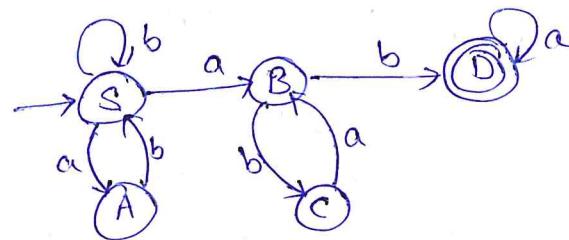


② RE: $a^*b(ab)^*a(b+ba)^*$

③ Reverse RE:

$(ab+b)^*a(ba)^*b\alpha^*$

④ Draw the finite automata for the reversed RE



⑤ Obtain RLG from the FA
 $S \rightarrow bs|ab|aA$

⑤ Reverse RLG to get LLG

$$S \rightarrow Ba \mid Sb \mid Aa$$

$$A \rightarrow Sb$$

$$B \rightarrow Cb \mid Db$$

$$C \rightarrow Ba$$

$$D \rightarrow Da \mid \epsilon$$

DFA \rightarrow LLG (double Revision)

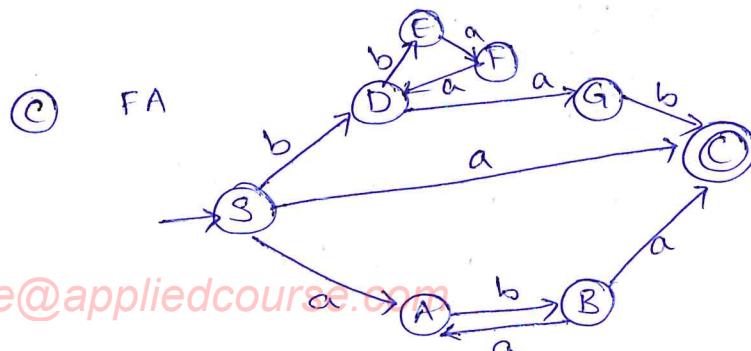
RLG \rightarrow LLG:

① $S \rightarrow aA \mid baB \Rightarrow a(ba)^* + ba(aab)^*b$
 $A \rightarrow baA \mid \epsilon \Rightarrow (ba)^*$
 $B \rightarrow aabB \mid b \Rightarrow (aab)^*b$

a) RE:
 $a(ba)^* + ba(aab)^*b$

b) Reverse

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



$s \rightarrow aA \mid bD \mid ac$
 $A \rightarrow bB$
 $B \rightarrow aA \mid ac$
 $C \rightarrow \epsilon$
 $D \rightarrow bE \mid aG$
 $E \rightarrow aF$
 $F \rightarrow aD$
 $G \rightarrow bC$

(e) Reverse RLG to obtain LLG

 $s \rightarrow Aa \mid Db \mid ca$
 $A \rightarrow Bb$
 $B \rightarrow Aa \mid Ca$
 $C \rightarrow \epsilon$
 $D \rightarrow Eb \mid Ga$
 $E \rightarrow Fa$
 $F \rightarrow Da$
 $G \rightarrow Cb$

LLG to RLG :-

① $A \rightarrow A10 \mid B110 \mid 101$

 $B \rightarrow B011 \mid 01$

(a) REVERSE the RHS of production Rule

 $A \rightarrow 0(A \mid 011B) \mid 101 \quad (01)^*(101 + 011(110)^*10 + 011B)$
 $B \rightarrow 110B \mid 10 \quad (110)^*10$

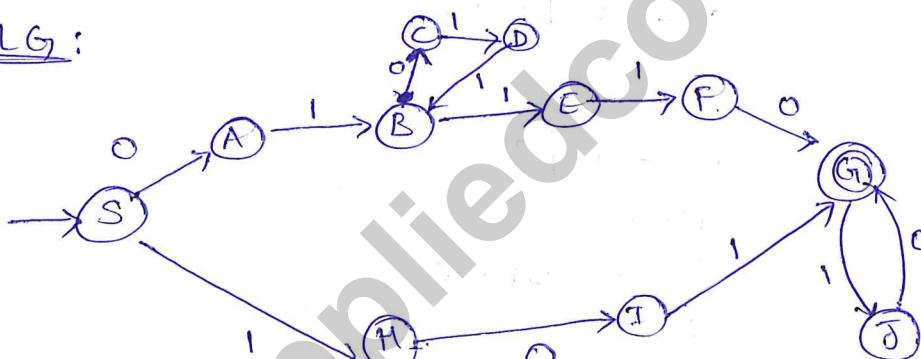
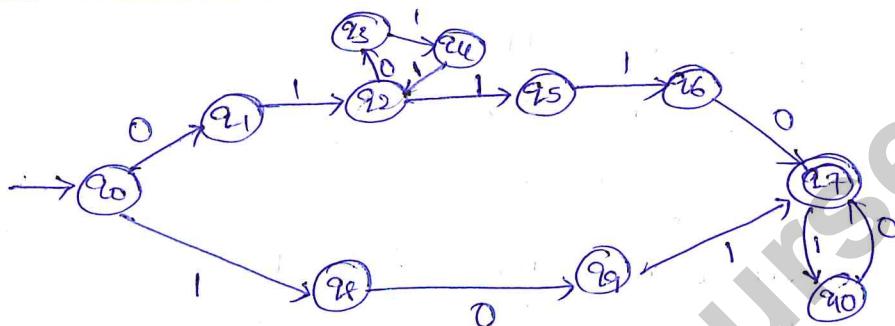
Mail: gatecse@appliedcourse.com [101 + 011(110)^*10] + 011(110)^*10

$$\Rightarrow (01)^*(101 + 011(110)^*10)$$

c) Reverse the RE:

$$(01(011)^*110 + 101)(10)^*$$

d) Finite Automata for the RE:



f)

$$S \rightarrow 0A|1H$$

\Rightarrow Reverse to obtain

$$A \rightarrow 1B$$

$$B \rightarrow 0C|1E$$

$$C \rightarrow 1D$$

$$D \rightarrow 1B$$

$$E \rightarrow 1F$$

$$F \rightarrow 0G$$

$$G \rightarrow 1J|E$$

$$H \rightarrow 0I$$

$$I \rightarrow 1G$$

$$J \rightarrow 0G$$

Context Free Grammars (CFG), CFL, PDA

$$A \rightarrow \alpha, A \in V$$

$$\downarrow$$

$$\alpha \in (V \cup T)^*$$

single var

Eg: $\begin{array}{l} S \rightarrow aSb \\ S \rightarrow e \end{array} \quad \left. \begin{array}{l} S \rightarrow aS \\ S \rightarrow Sa \end{array} \right\} \text{RG}$

$\text{CFG} \quad \text{NOT RG}$

$$\text{CFL: } L(G) = \{ a^n b^n \mid n \geq 0 \}$$

$$L_1: \{ a^m b^n \mid m < n \}$$

$$L_2: \{ a^m b^n \mid m \neq n \}$$

optimization: Elimination of useless symbols / var & Ruleuseless symbols

① $S \rightarrow aA$
 $A \rightarrow b$ $\Rightarrow S \rightarrow aA$
 $B \rightarrow b$

B is not derivable
from S.

② $S \rightarrow aA \mid bB$ X $\Rightarrow S \rightarrow aA$
 $A \rightarrow b$ $A \rightarrow b$

B does not
derive a terminal

symbol

③ $S \rightarrow aAb \mid \boxed{ba}$
 $A \rightarrow \boxed{aB} \mid ba$
 $B \rightarrow \boxed{ABC}$ X

$\Rightarrow S \rightarrow aAb$
 $A \rightarrow ba$

B and C are useless
var

④ $S \rightarrow aAB \mid BC$ X
 $A \rightarrow aB \mid b$
 $B \rightarrow bA \mid BC$ X

$B \rightarrow bA \downarrow b \Rightarrow bb$ (indirectly derives bb)

$S \rightarrow aAB$
 $A \rightarrow aB \mid b$
 $B \rightarrow bA$

$$S \rightarrow aAB \mid \underline{BAC} \mid \underline{BC}$$

$$A \rightarrow aA \mid Ba \mid AC \mid a$$

$$B \rightarrow \underline{BAC} \mid BB \mid b$$

C is a useless var

$$S \rightarrow aAB$$

$$A \rightarrow aA \mid Ba \mid a$$

$$B \rightarrow BB \mid b$$

⑥ Unit production:

$$A \rightarrow B, A, B \in V \quad \{ \text{Simplify} \}$$

$$\begin{aligned} ① \quad S &\rightarrow aA \\ A &\rightarrow B \mid b \\ B &\rightarrow a \end{aligned}$$

$$\begin{aligned} S &\rightarrow aA \\ A &\rightarrow a \mid b \\ B &\rightarrow a \end{aligned} \quad \boxed{\text{useless}}$$

$$\begin{aligned} S &\rightarrow aa \\ A &\rightarrow a \mid b \end{aligned}$$

$$\begin{aligned} ② \quad S &\rightarrow AaB \\ A &\rightarrow B \mid ab \mid a \\ B &\rightarrow C \mid bB \mid b \end{aligned} \Rightarrow \begin{aligned} A &\rightarrow a \\ S &\rightarrow AaB \\ A &\rightarrow B \mid ab \mid a \\ B &\rightarrow bB \mid b \end{aligned} \Rightarrow \begin{aligned} S &\rightarrow AaB \\ A &\rightarrow ab \mid a \mid bB \mid b \\ B &\rightarrow bB \mid b \end{aligned}$$

$$\begin{aligned} ③ \quad S &\rightarrow ABa \mid BC \times \\ A &\rightarrow aA \mid B \mid a \\ B &\rightarrow D \mid b \\ D &\rightarrow b \mid C \end{aligned}$$

$$\begin{aligned} A &\rightarrow B \\ B &\rightarrow D \\ D &\rightarrow C \times \end{aligned}$$

$$\left. \begin{aligned} S &\rightarrow ABA \\ A &\rightarrow aA \mid a \mid b \\ B &\rightarrow b \\ D &\rightarrow b \times \end{aligned} \right\}$$

ϵ -production:

$$A \rightarrow \epsilon ; A \in V$$

$$\textcircled{1} \quad S \rightarrow ABC$$

$$A \rightarrow a$$

$$B \rightarrow \epsilon$$

$$C \rightarrow b$$

$$\Rightarrow \quad S \rightarrow ABC$$

$$A \rightarrow a$$

$$C \rightarrow b$$

$$\textcircled{2} \quad S \rightarrow AaB$$

$$A \rightarrow b/\epsilon$$

$$B \rightarrow C/\epsilon$$

↓
terminal

$$S \rightarrow AaB | aB | Aa | a$$

$$A \rightarrow b$$

$$B \rightarrow C$$

No ϵ -production

$$\textcircled{3} \quad S \rightarrow bAB | BAa$$

$$A \rightarrow aA | \epsilon | \cancel{AC}$$

$$B \rightarrow bB | G | \cancel{BC}$$

$$S \rightarrow bAB | BAa | bB | aA | b | a | \cancel{bBa} | \cancel{ba}$$

$$A \rightarrow aA | a$$

$$B \rightarrow bB | b$$

$$\textcircled{4} \quad S \rightarrow AB \quad \Rightarrow \quad S \rightarrow AB | A | B | \epsilon$$

$$A \rightarrow a | \epsilon$$

$$B \rightarrow b | \epsilon$$

$$A \rightarrow a$$

$$B \rightarrow b$$

Order of optimizations

- ① Remove ϵ -productions
- ② Remove unit productions
- ③ Remove useless symbols.

Normal Forms:

- ① Chomsky Normal Form (CNF)
- ② Greibach Normal Form (GNF)

(130)

Chomsky Normal Form:

$$A \rightarrow BC \quad A, B, C \in V$$

$$A \rightarrow a \quad a \in T$$

Eg 1: $s \rightarrow asb | ab$

$$\begin{array}{ll} A \rightarrow a & s \rightarrow AB | ASB \\ B \rightarrow b & A \rightarrow a \\ & B \rightarrow b \end{array}$$

$$\boxed{\begin{array}{l} s \rightarrow AB | CB \\ C \rightarrow AS \\ A \rightarrow a \\ B \rightarrow b \end{array}}$$

CNF

Eg 2: $s \rightarrow asa | bsb | e$

$$\begin{array}{l} \boxed{s' \rightarrow s' e} \\ \Rightarrow \begin{array}{l} s \rightarrow e \\ s \rightarrow DA | CB \\ D \rightarrow AS \\ C \rightarrow BS \\ A \rightarrow a \\ B \rightarrow b \end{array} \end{array}$$

Eg 3: $s \rightarrow aAb | ab | abA$

$$A \rightarrow a$$

$$B \rightarrow b$$

$$\begin{array}{l} \Rightarrow \begin{array}{l} s \rightarrow AC \\ s \rightarrow AB \\ s \rightarrow CA \\ C \rightarrow AB \\ A \rightarrow a \\ B \rightarrow b \end{array} \end{array}$$

CNF

Eg 4: $S \rightarrow CB | AB$

$C \rightarrow AS$

$B \rightarrow b$

$A \rightarrow a$

Ph: 844-844-0102

Let $w_1 = ab$ $|w_1| = 2$

$w_2 = \overbrace{aab}^4 b$

$S \rightarrow AB$
 $\rightarrow aB$
 $\rightarrow ab$

$S \rightarrow CB$

$\rightarrow ASB$

$\rightarrow AABBB$

$\rightarrow aABB$

$\rightarrow aaBB$

$\rightarrow aabB$

$\rightarrow aabb$

string of len n \xrightarrow{CNF} 2^{n-1} derivations

GNF (Greibach Normal Form):-

$A \rightarrow a\alpha$ $A \in V, \alpha \in T, \alpha \in V^*$

\hookrightarrow string of variables

Eg 1:

$S \rightarrow asb | ab$

\Downarrow

$S \rightarrow asB | aB$
 $B \rightarrow b$

Eg 2: $S \rightarrow asa | bsb | aal | bb$

\Downarrow

$S \rightarrow ASA | BSB | aA | bB$

$B \rightarrow b$

$A \rightarrow a$

Eg 3:

$S \rightarrow aAb | bBa$

$A \rightarrow bAa | b$

$B \rightarrow bBa | a$

$S \rightarrow aAy | bBx$

$A \rightarrow bAx | b$

$B \rightarrow bBx | a$

$X \rightarrow a$

$Y \rightarrow b$

132

$$S \rightarrow Aa \mid bBa$$

$$A \rightarrow bB \mid a$$

$$B \rightarrow bBa \mid b$$

$$S \rightarrow AX \mid bBX$$

$$A \rightarrow bB \mid a$$

$$B \rightarrow bBa \mid b$$

$$X \rightarrow a$$

$$S \rightarrow AX$$

$$A \rightarrow bB \mid a$$

↓

$$S \rightarrow ax \mid bBX$$

$$S \rightarrow ax \mid bBX$$

$$\boxed{A \rightarrow bB \mid a}$$

$$B \rightarrow bBX \mid b$$

$$X \rightarrow a$$

⇒

$$S \rightarrow ax \mid bBX$$

$$B \rightarrow bBX \mid b$$

$$X \rightarrow a$$

Eq 5:

$$S \rightarrow asb \mid ab$$

$$\xrightarrow{N} S \rightarrow aSB \mid aB$$

$$B \rightarrow b$$

$$W_1 = ab$$

$$|W_1| = 2$$

$$W_2 = aabb$$

$$|W_2| = 4$$

$$S \rightarrow aB \} 2$$

$$\rightarrow ab$$

$$\begin{aligned} S &\rightarrow aSB \\ &\rightarrow aasBB \\ &\rightarrow aabB \\ &\rightarrow aabb \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} H$$

String of Length $n \xrightarrow{GNF} n$ productions to generate the string.

Decision properties of CFG:

① Emptyness & Non-emptiness:

$$\text{G}_1: S \rightarrow aA|ab \\ A \rightarrow aB|b$$

$$\text{G}_2: S \rightarrow AB|aB \\ A \rightarrow a|b \\ B \rightarrow AB$$

Decidability TM of Halting problem
Ph. 844-844-002

Given

a problem, if we can write a computer program then the given problem is decidable.

Reduced G:

$$S \rightarrow aA \quad B \text{ is useless} \\ A \rightarrow b \\ \text{Non-Empty}$$

$$\text{G}_2: S \rightarrow AB|aB \\ A \rightarrow a|b \\ B \rightarrow AB$$

B is useless

Reduced Grammar: Empty Language.

$$S \rightarrow \square$$

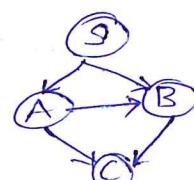
Decidable property of CFL

② Finiteness / Infiniteness: Does the G_i generate finite algo $\leftarrow Y$

variable dependency graph (VDG_i)

\exists algorithm

$$S \rightarrow AB \\ A \rightarrow BC|a \\ B \rightarrow CC|b \\ C \rightarrow d$$



$VDG_i(S)$

directed Graph

Loop/Cycles

{ NO Loop/Cycles \rightarrow Finite
Loops/Cycles \rightarrow Infinite Language }

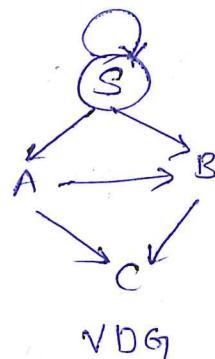
Decidable property

$$S \rightarrow SS \mid AB$$

$$A \rightarrow BC \mid a$$

$$B \rightarrow CC \mid a$$

$$C \rightarrow d$$



\exists loop \Rightarrow Infinite Language 44-0102

Non-Empty Language

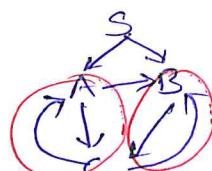
c

$$S \rightarrow AB$$

$$A \rightarrow BC \mid a$$

$$B \rightarrow CC \mid b$$

$$C \rightarrow AB$$



Non-Empty
 \exists Cycle \Rightarrow Infinite Language.

Membership:

$w \in L(G)$?

L Decidable property

G_1 has to be in CNF

CYK
Coke Young Kasami

Dynamic programming

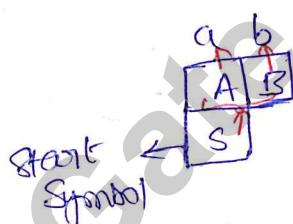
① $S \rightarrow AB$

$$A \rightarrow BA \mid SA \mid a$$

$$B \rightarrow BB \mid BS \mid b$$

CNF

a) $w_1 = ab$



$$w_2 = ba$$

b	a
B	A
A	

c)

Not start symbol

$$w_2 \notin L(G)$$

c) $w_3 = aba$

a	b	a
A	B	A
B	A	
A		

Not the start state

$$w = aba \notin L(G)$$

$S \rightarrow AB$
 $A \rightarrow BA | SA | a$
 $B \rightarrow BB | BS | b$

(d) $w = bba$

b	b	a
B	B	A
B	A	
A		≠ S

$w \notin L(G)$

(e) $w = abb$

a	b	b
A	B	B
A	B	
S		

$abb \in L(G)$

(f) $w = aab$

a	a	b
A	A	B
A	S	
Ø	≠ S	

$w \notin L(G)$

Eg 2: $S \rightarrow AB | AA$

$A \rightarrow BA | SA | a | b$

$B \rightarrow BB | BS | a$

(a) $w = abba$

a	b	b	a
A,B	A	A,B	AB
S,A	A	AB	
A,S	AB		
S,A			

Contains S

$abba \in L(G)$

a b a a

a	b	a	a
A,B	A	A,B	AB
S,A	A,B	A,B	
A,S	A,B		
S,A			

$abba \in L(G)$

Membership of CFL is decidable

Equality of two Context free Grammar: undecidable property

① Grammar must be CNF

② Construct a Triangular table

→ Each row corresponding to one length of Substrings.

- Bottom Row - Strings of Length 1

- Second row from Bottom Row - Strings of Length 2

⋮

Top Row - String w

$x_{i,j}$ is the set of variables such that $A \rightarrow w_i$ is a production of or

Compare at most n pairs of previously computed sets.

$(x_{i,1}, x_{i+1,j}) (x_{i,2}, x_{i+1, j}) \dots (x_{i,j-1}, x_{i,j})$

$x_{1,5}$				
$x_{1,4}$	$x_{2,5}$			
$x_{1,3}$	$x_{2,4}$	$x_{3,5}$		
$x_{1,2}$	$x_{2,3}$	$x_{3,4}$	$x_{4,5}$	
$x_{1,1}$	$x_{2,2}$	$x_{3,3}$	$x_{4,4}$	$x_{5,5}$

$w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5$

Table for string w of length 5

① $G_1: S \rightarrow AB \mid BC$

$A \rightarrow BA \mid a$

$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

$w = baaba \in L(G) ?$

$$S \rightarrow AB \mid BC$$

$$A \rightarrow BA \mid a$$

$$B \rightarrow CC \mid b$$

$$C \rightarrow AB \mid a$$

$\{S, AC\}$	x_{15}	\emptyset	$\{S, A, C\}$	
\emptyset	x_{14}	B	B	
\emptyset	x_{13}	x_{24}	x_{35}	
$\{A, S\}$	$\{B\}$	$\{B\}$	$\{S, C\}$	$\{S, A\}$
x_{12}	x_{23}	x_{34}	x_{45}	x_{55}
$\{B\}$	$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{A, C\}$
x_{11}	x_{22}	x_{33}	x_{44}	x_{55}

$baaba \in L(G)$

$$x_{12} = x_{11} x_{22}$$

$$\{B\} \{A, C\} = BA \mid BC$$

$$x_{23} = x_{22} x_{33}$$

$$\{A, C\} \{A, C\} = \{AA, AC, CA, CC\}$$

$$x_{34} = x_{33} x_{44} = \{A, C\} \{B\} = \underline{\{AB, CB\}}$$

$$S, C$$

$$x_{45} = x_{44} x_{55} = \{B\} \{A, C\} = \{BA, BC\}$$

$$A, S$$

$$x_{13} = x_{11} x_{23} \cup x_{12} x_{33}$$

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

$$= \{AA, AC, SA, SC\} \cup \{BB\}$$

$$= \emptyset$$

$$x_{14} = x_{11} x_{24} +$$

$$x_{12} x_{34} +$$

$$x_{13} x_{44}$$

$$= x_{11} x_{24} \cup x_{12} x_{34} \cup x_{13} x_{44}$$

$$= BB \cup \{AS, AC, BS, SC\} \cup \emptyset$$

$$= \emptyset \cup \{AS, AC, BS, SC\} \cup \emptyset$$

$$= \emptyset$$

$$x_{24} = x_{22} x_{34} + x_{23} x_{44}$$

$$= \{A, C\} \{S, C\} \cup$$

$$= \{AS, AE, CS, CA\} \cup \{B\} \{B\}$$

$$= \{AS, AE, CS, CA\} \cup \underline{BB}$$

$$B \rightarrow CC$$

$$x_{35} = x_{33} x_{45} \cup x_{34} x_{55}$$

$$= \{A, C\} \{S, A\} \cup \{S, C\} \{A, C\}$$

$$= \{AS, AA, CS, CA\} \cup \{SA, SC, CA, CC\}$$

$$\emptyset$$

(138)

$$x_{25} = x_{22}x_{35} + x_{23}x_{45} + x_{24}x_{55}$$

Ph: 844-844-0102

$$= \{A, (YLB) \cup \{BY\} S, A\} + \{B\} \{A\}$$

$$= \{AB, CB, BS, BA, BA, BC\}$$

$$= \{c, A, S\}$$

$$x_{15} = x_{11}x_{25} + x_{12}x_{35} + x_{13}x_{45} + x_{14}x_{55}$$

$$= B \{S, A, C\} + \{A, S\} B + \emptyset + \emptyset$$

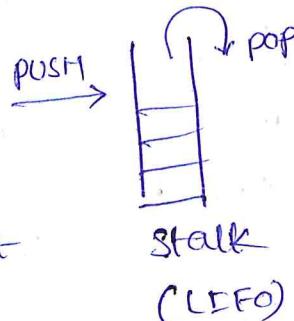
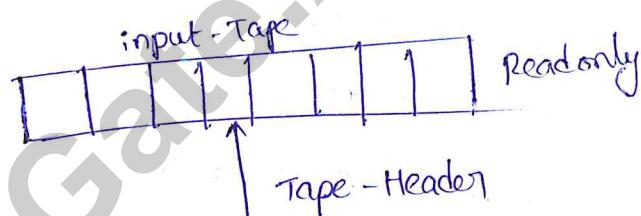
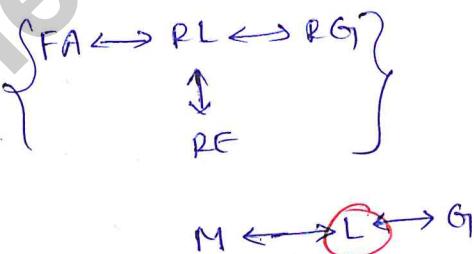
$$= \{BS, BA, BC, AB, SB\}$$

$$= \{S, A, C\}$$

$$x_{i,j}^* = x_{i,i} x_{i+1,j} + x_{i+1} x_{i+2,j} + \dots + x_{i,j-1} x_{j,j}$$

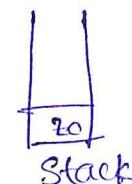
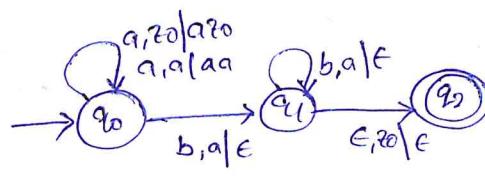
Pushdown Automata:

$$\begin{array}{c} \text{CFG} \leftrightarrow \text{CFL} \\ \uparrow \\ \text{PDA} \end{array}$$



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

Eg 1: $L = \{a^n b^n \mid n \geq 1\}$



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

z_0 : Initial Stack Top

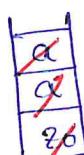
q_0 : Initial State

F : Set of final States

→ Let $w = aabb$

$\uparrow \uparrow \uparrow \uparrow$

$$\delta(q_0, a, z_0) \rightarrow q_0, a z_0$$



$$\delta(q_0, a, a) \rightarrow q_0, aa$$

$$\delta(q_0, b, a) \rightarrow q_1, \epsilon$$

Stack is $\delta(q_1, b, a) \rightarrow q_1, \epsilon$
Empty

$$\delta(q_1, \epsilon, z_0) = \underline{q_2}, \epsilon$$

Final State

Mathematical Model of PDA:

Σ : Set of States

q_0 : initial State

F: set of final States

z_0 : initial stack Top

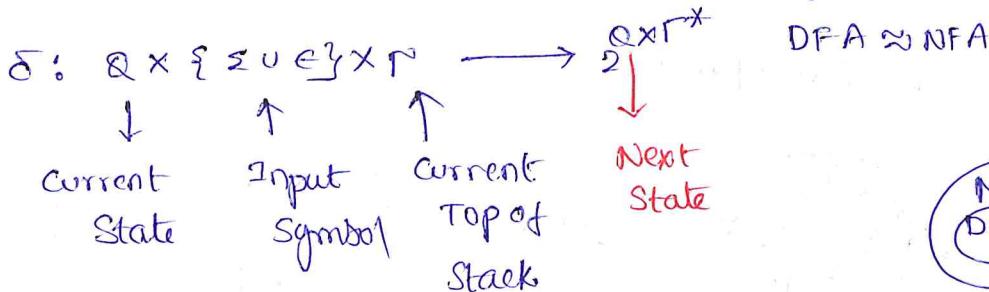
Γ : Set of Stack Symbols

Σ : Set of Input Symbols.

PDA

Ph: 844-844-0102

DPDA \neq NPDA by
less powerful
default



DFA \approx NFA



Acceptance

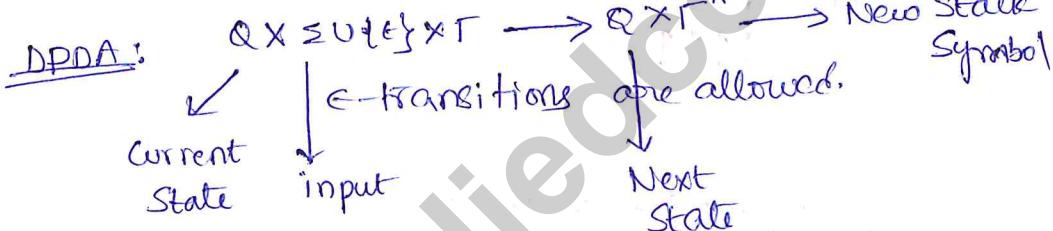
↳ Final State

↳ Empty Stack

Non-deterministic PDA vs Deterministic PDA:

NPDA \neq DPDA

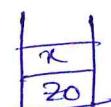
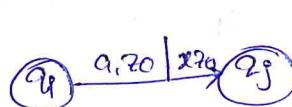
Current Top



NPDA: $Q \times \Sigma \cup \{\epsilon\} \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$

Stack operations in PDA:

① PUSH: $\delta(q_i, a, z_0) = (q_j, xz_0)$

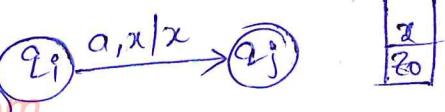


② POP: $\delta(q_i, a, z) = (q_j, \epsilon)$

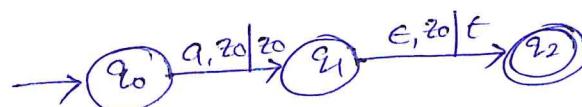


③ SLIP:

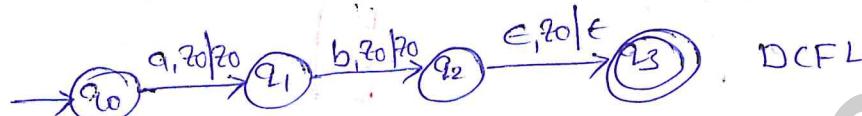
Mail: gatecse@appliedcourse.net



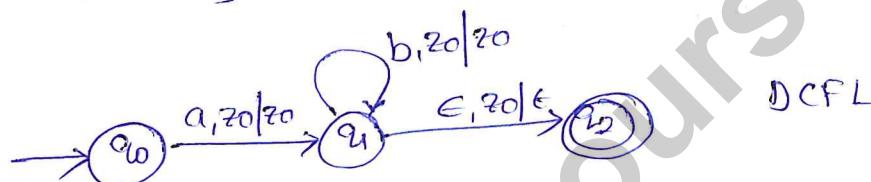
① $L = \{a\}$



② $L = \{aabb\}$

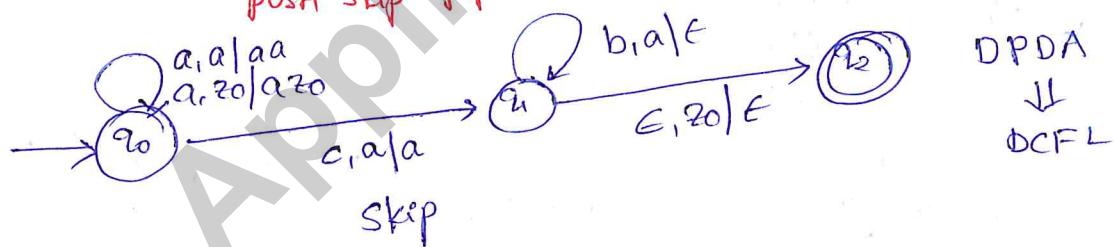


③ $L = \{ab^*\}$



④ $L = \{a^n b^n \mid n \geq 1\}$

\downarrow push \downarrow skip \downarrow pop



DPDA
↓
DCFL

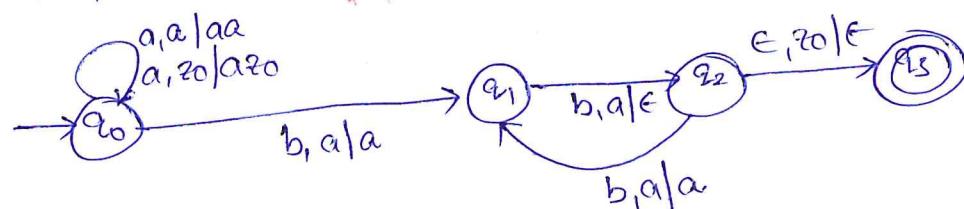
DFA: All transitions must be defined.

DPDA: Need to be defined.
not

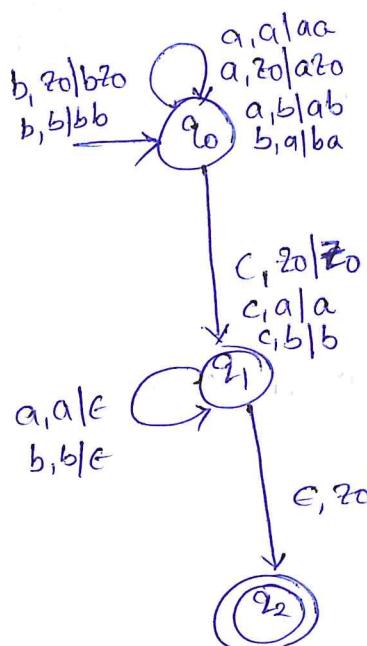
⑤ $L = \{a^n b^n \mid n \geq 1\}$

abb, aabbbb,

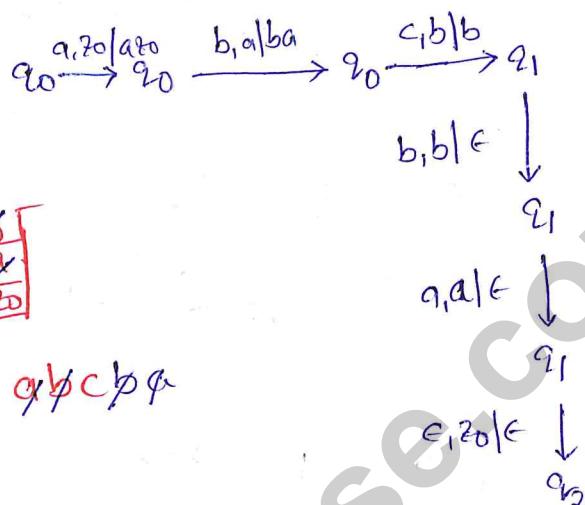
a, a/aa
a, z0/z0



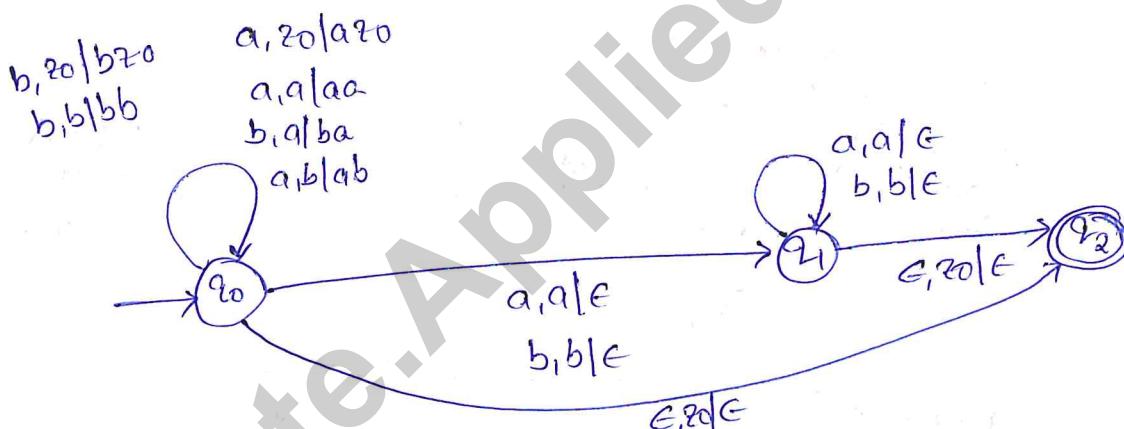
$$L = \{ w \in \{a,b\}^* \mid \text{push} \uparrow \text{pop} \}$$



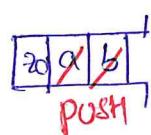
Let $w = abcba$



$$⑦ L = \{ wwr \mid w \in \{a,b\}^* \} \rightarrow \text{NPDA}$$



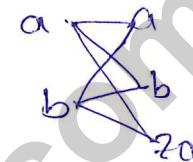
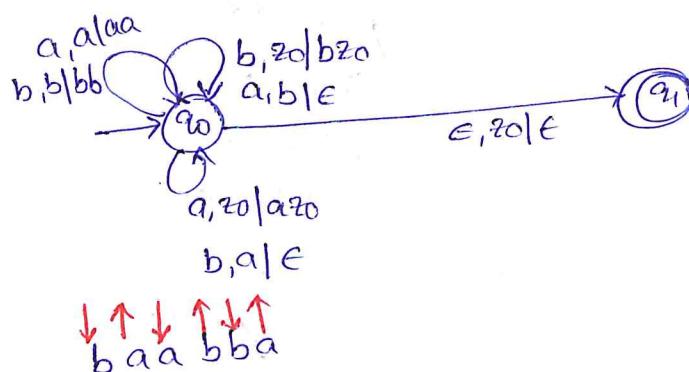
Let $w = \underline{abba}$
push pop



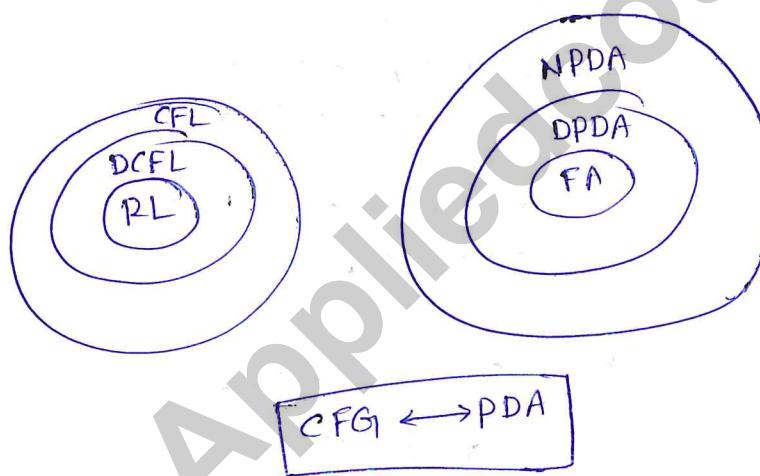
ab/x/a
↑↑ pop

$w = bbbaaa$

$w = baabba$



RL vs DCFL vs CFL*



DCFL: wz_0w^R
 $aabb$
 amb^n/mcn

CFL: wwR
 $xwzx$
 ww^Rx

CFG \rightarrow PDA:

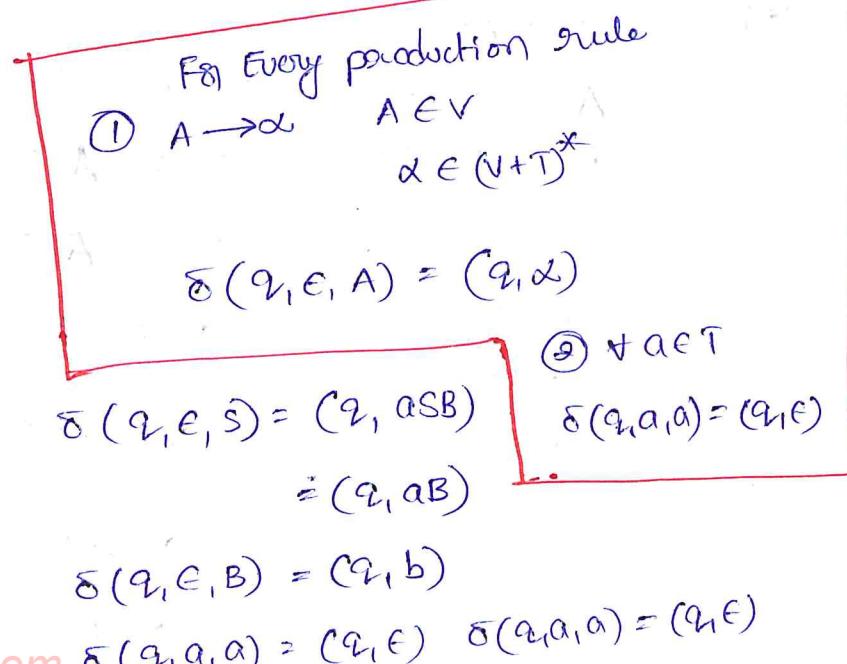
① $S \rightarrow aSB | aB$
 $B \rightarrow b$

Let $z_0 = S$

$q = q$

$q_0 = q$

$F = \emptyset$



$w = aabb \in L(G)$

\Rightarrow is accepted by PDA $\in aabb, S$

~~aabb~~ ~~ASB~~
~~aabb~~ ~~SB~~

~~aabb~~ ~~BB~~

~~bb~~ ~~BB~~

~~bb~~ ~~BB~~

~~b~~ ~~B~~

~~b~~ ~~B~~

~~b~~ ~~B~~

~~b~~ ~~B~~

~~b~~ ~~B~~



$$\textcircled{2} \quad S \rightarrow aSA | bAB$$

$$A \rightarrow aA | a$$

$$B \rightarrow bBA | b$$

$$q_0 = S$$

$$Q = \{q_1\}$$

$$q_f = q_1$$

$$F = \emptyset$$

$$\delta(q_1, \epsilon, S) = (q_1, aSA)$$

$$\delta(q_1, \epsilon, S) = (q_1, bAB)$$

$$\delta(q_1, \epsilon, A) = (q_1, aA)$$

$$\delta(q_1, \epsilon, A) = (q_1, a)$$

$$\delta(q_1, \epsilon, B) = (q_1, bBA)$$

$$\delta(q_1, \epsilon, B) = (q_1, b)$$

$$\delta(q_1, a, a) = (q_1, \emptyset)$$

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

\textcircled{3}

$$S \rightarrow AB$$

$$A \rightarrow AS | a$$

$$B \rightarrow bBa | \epsilon$$

$$q_0 = S$$

$$Q = \{q_1\}$$

$$q_f = q_1$$

$$F = \emptyset$$

$$\delta(q_1, \epsilon, S) = (q_1, AB)$$

$$\delta(q_1, \epsilon, A) = (q_1, AS)$$

$$\delta(q_1, \epsilon, A) = (q_1, a)$$

$$\delta(q_1, \epsilon, B) = (q_1, bBa)$$

$$\delta(q_1, \epsilon, B) = (q_1, \epsilon)$$

$$\delta(q_1, a, a) = (q_1, \epsilon)$$

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

$S \rightarrow aSB \mid aB \quad \} \text{ GNF}$

$B \rightarrow b$

$A \rightarrow a\alpha \quad a \in V^*$

$\alpha \in V^*$

$A \rightarrow V$

If the grammar is in GNF

$$\textcircled{1} \quad \delta(q_1, a, A) = (q_1, \alpha)$$

$$\textcircled{2} \quad \delta(q_1, a, a) = (q_1, \epsilon)$$

$z_0 = S$

$$\delta(q_1, a, S) = (q_1, SB)$$

$Q = \{q_1\}$

$$\delta(q_1, a, S) = (q_1, B)$$

$q_0 = q_1$

$$\delta(q_1, b, B) = (q_1, \epsilon)$$

$F = \emptyset$

$$\delta(q_1, a, a) = (q_1, \epsilon)$$

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

(5)

$S \rightarrow aAB$

$A \rightarrow aBA \mid bAA \mid a \quad \} \text{ GNF}$

$B \rightarrow aBB \mid bAB \mid b$

$z_0 = S; Q = \{q_1\}; q_0 = q_1, F = \emptyset$

$$\delta(q_1, a, S) = (q_1, AB)$$

$$\delta(q_1, a, A) = (q_1, BA)$$

$$\delta(q_1, b, A) = (q_1, AA)$$

$$\delta(q_1, a, A) = (q_1, \epsilon)$$

$$\delta(q_1, a, B) = (q_1, BB)$$

$$\delta(q_1, b, B) = (q_1, AB)$$

$$\delta(q_1, b, B) = (q_1, \epsilon)$$

$$\delta(q_1, a, a) = (q_1, \epsilon)$$

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

For all CFL, $L \exists$ an integer $n \geq 1$
 such that $\forall z \in L$ and $|z| \geq n$

$$(i) z = uvwxy$$

$$(ii) |vwx| \leq n$$

$$(iii) |vx| \neq 0$$

(iv)

$$|vz| \geq 1$$

then $uv^iwx^iy \in L, \forall i \geq 0$

$$\textcircled{1} \quad L = \{a^m b^m c^m \mid m \geq 0\}$$

Let's assume $L \in \text{CFL}$

$$\exists n \geq 1$$

$$z \in L \quad |z| \geq n$$

$$z = a^k b^k c^k$$

$$3k \geq n$$

$$\begin{array}{c} a^k \ b^k \ c^k \\ / \ \backslash \ \backslash \\ u \ \ v \ \ w \ x \ y \\ a^{k-1} \ a \ \ b^k \ \in \ c^k \end{array}$$

$$|vwx| = k+1 \leq n$$

hence

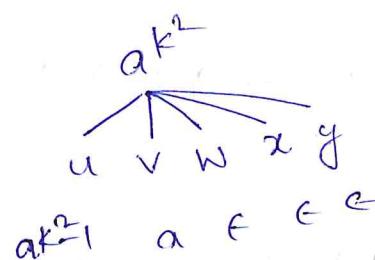
$$uv^iwx^iy \in L \quad \forall i \geq 0$$

$$i=2 \quad a^{k-1} a^2 b^k \notin C^k \Rightarrow a^{k+1} b^k c^k \notin L$$

$\epsilon, a, aaaa, aaaaaaaaaa, \dots$ Let L be a Context-free Language. $\exists n \geq 1$

$$z \in L \quad |z| \geq n$$

$$z = a^{k^2}$$

pick k such that $k^2 \geq n$ 

$$|vwx| \leq n$$

$$i \leq n$$

$$|wx| \geq 1$$

then $uv^iwx^iy \in L, \forall i \geq 0$

$$\stackrel{i=2}{a^{k^2-1} a^2 \epsilon \epsilon \epsilon}$$

$$= a^{k^2+1} \notin L$$

L \neq CFLSpecial case: $\Sigma = \{a^k\}$ & L is defined over Σ then L is CFL, iff the lengths of strings in L are in Arithmetic progression.

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

aa, aaaa, aaaaaa, ...

2 4 6

Ph: 844-844-0102

$$\textcircled{2} \quad L = \{a^{3n+1} \mid n \geq 1\} = \text{CFL}$$

aaaa, aaaaaaa,

4 7 10 ...

$$\textcircled{3} \quad L = \{an^2 \mid n \geq 1\} \Rightarrow \text{Non-CFL}$$

a, aaaa, aaaaaaaaa, ...

1 4 9

$$\textcircled{4} \quad L = \{an! \mid n \geq 0\} \Rightarrow \text{Non-CFL}$$

1, 1, 2, 6, 24, 120, ...

$$\textcircled{5} \quad L = \{ap \mid p \text{ is a prime}\} \Rightarrow \text{NOT in AP} \Rightarrow \text{Non-CFL}$$

Closure properties of CFLs:

\textcircled{1} Union: $L_1: \text{CFL}$ $L_2: \text{CFL}$

$L_1 \cup L_2: \text{CFL?}$

$\downarrow G_1$

$\downarrow G_2$

$A \rightarrow \alpha$

$A \in V$

$\alpha \in (V \cup T)^*$

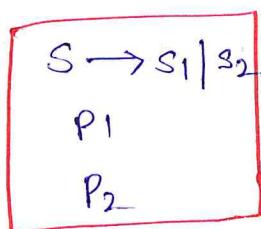
$\hookrightarrow G$

S_1

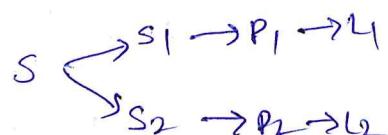
S_2

P_1

P_2

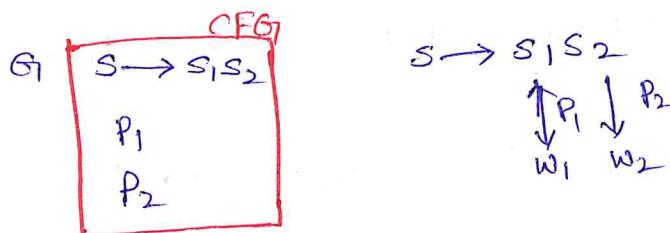


$\rightarrow \text{CFG}$



② Concatenation:

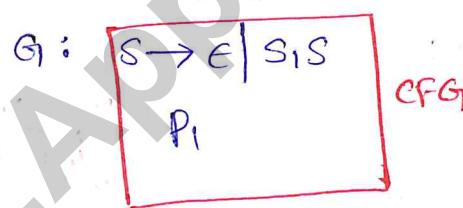
$L_1: CFL$ $L_2: CFL$ w_1, w_2
 ↑ ↑
 $G_1: CFG$ $G_2: CFG$
 s_1, p_1 s_2, p_2



③ Kleene-Closure:

$w_1 \in L_1: CFL$
 ↑
 $G_1: CFG$
 s_1, p_1

$L_1^* = \{\epsilon, L_1, LL \dots\}$
 $G: L_1^*?$
 (CFG)



④ Reversal: CFL

$w_1 \in L_1: CFL$ $w_1^R \rightarrow CFL?$
 ↑ ↑
 G_1 G_2

\downarrow
 $s_1 s_1$
 \downarrow
 $L_1 L_1$

$S \rightarrow 0s1|01$

$\{01, 0011 \dots\}$

$S \rightarrow 1s0|10$

$\{10, 1100, \dots\}$

Homomorphism.

$G_1: S \rightarrow OS1|01$
 $\Sigma = \{0,1\}$

$L: CFL$
 $h(0) = ab$
 $h(1) = e$
 $L' : eFL?$

$G_1': S \rightarrow abS|ab$
 $\Sigma = \{a,b\}$

$G_1' \rightarrow CFG$

Intersection, set-difference: Not closed

$L_1 = \{anb^nc^m \mid n \geq 1, m \geq 1\}$ aabbcc

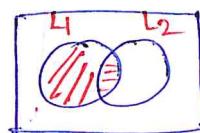
$L_2 = \{amb^nc^m \mid n \geq 1, m \geq 1\}$ abbccc

$L_1 \cap L_2 = \{anb^nc^m \mid n \geq 1\} \rightarrow$ Not CFL

(pumping lemma)

Set difference:

$$L_1 \cap L_2 \\ = L_1 - (L_1 - L_2)$$



$$= L_1 - L_1 \cup L_2$$

Let set difference is closed

if also then
 CFL which is not true.

\Downarrow
 $L_1 \cap L_2$ is also CFL

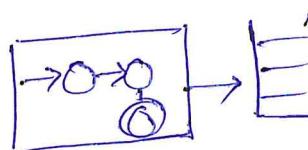
Intersection with Regular Languages:

$L_1: CFL$

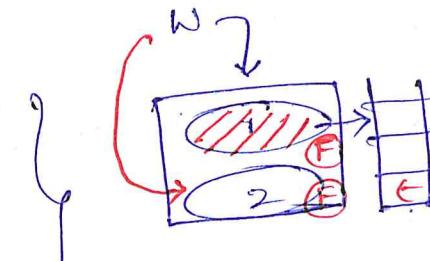
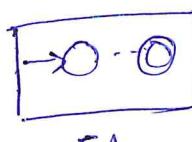
$L_2: RL$

$L_1 \cap L_2 = \text{CFL}?$

$L_1 \in \text{CFL}$



$L_2 \in \text{RL}$



as has to be accepted

by both FA and PDA

then $L_1 \cap L_2 \in \text{CFL}$

$$L_1 = \{a^n b^n \mid n \geq 1\} \rightarrow \text{CFL}$$

$$L_2 = \{a^m b^n \mid m, n \geq 1\} \rightarrow \text{RL}$$

$$L_1 \cap L_2 = \{a^n b^n \mid n \geq 1\} \rightarrow \text{CFL}$$

Note: For GATE: Closure properties:

CFLs are closed under:

- ① union
- ② concatenation
- ③ Kleene closure
- ④ substitution
- ⑤ Homomorphism
- ⑥ Inverse Homomorphism
- ⑦ Intersection with RL
- ⑧ Quotient with RL
- ⑨ Reverse.

CFLs are not closed under

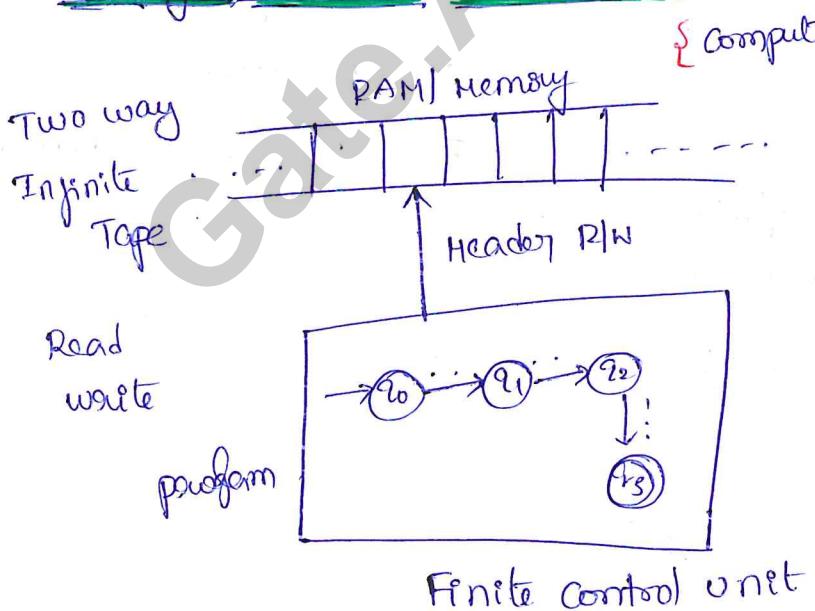
- ① Intersection
- ② Complement
- ③ Set-difference
- ④ Quotient
- ⑤ Inverse Substitution

- ① Complement → DPDA $L_1 L_2$
- ② Inverse Homomorphism
- ③ Intersection with RL
- ④ Quotient with RL

DCFLs are not closed under:

- ① Union
- ② Concatenation
- ③ Kleene closure
- ④ Homomorphism
- ⑤ Reversal
- ⑥ Intersection
- ⑦ Substitution

Turing Machines: An Introduction



Alan Turing

- Father of Computer Science
- Highest Award in Computer Science is Turing Award.

{Most simple model
what is Computable?}

Mathematical Model: 7-tuple

Ph: 844-844-0102

Q: Set of States

Σ : Input Alphabet

Γ : $\{a, b, \text{blank symbol } B\} \cup \Sigma$

- $|B| |B| |a| |a| |b| |b| |B| |B| \dots$

F: Set of Final States

q_0 : Initial State

δ : Set of Tape Symbols

$$\Sigma \subset \Gamma \quad B \in \Gamma$$

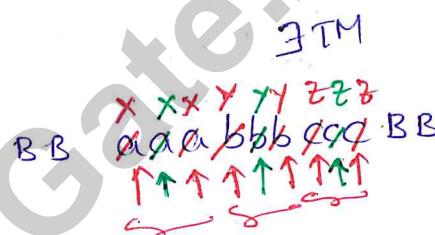
$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

↓ ↓ ↓ ↑ Dissection
Current State Tape input Next State Replacement Symbol

$$x \in \Gamma \quad x \notin \Sigma$$

$$q_0 \xrightarrow{a, x, R} q_1$$

Eg1: $\Sigma = \{a^n b^n c^n \mid n \geq 1\}$: Pumping Lemma of CFL \rightarrow No PDA

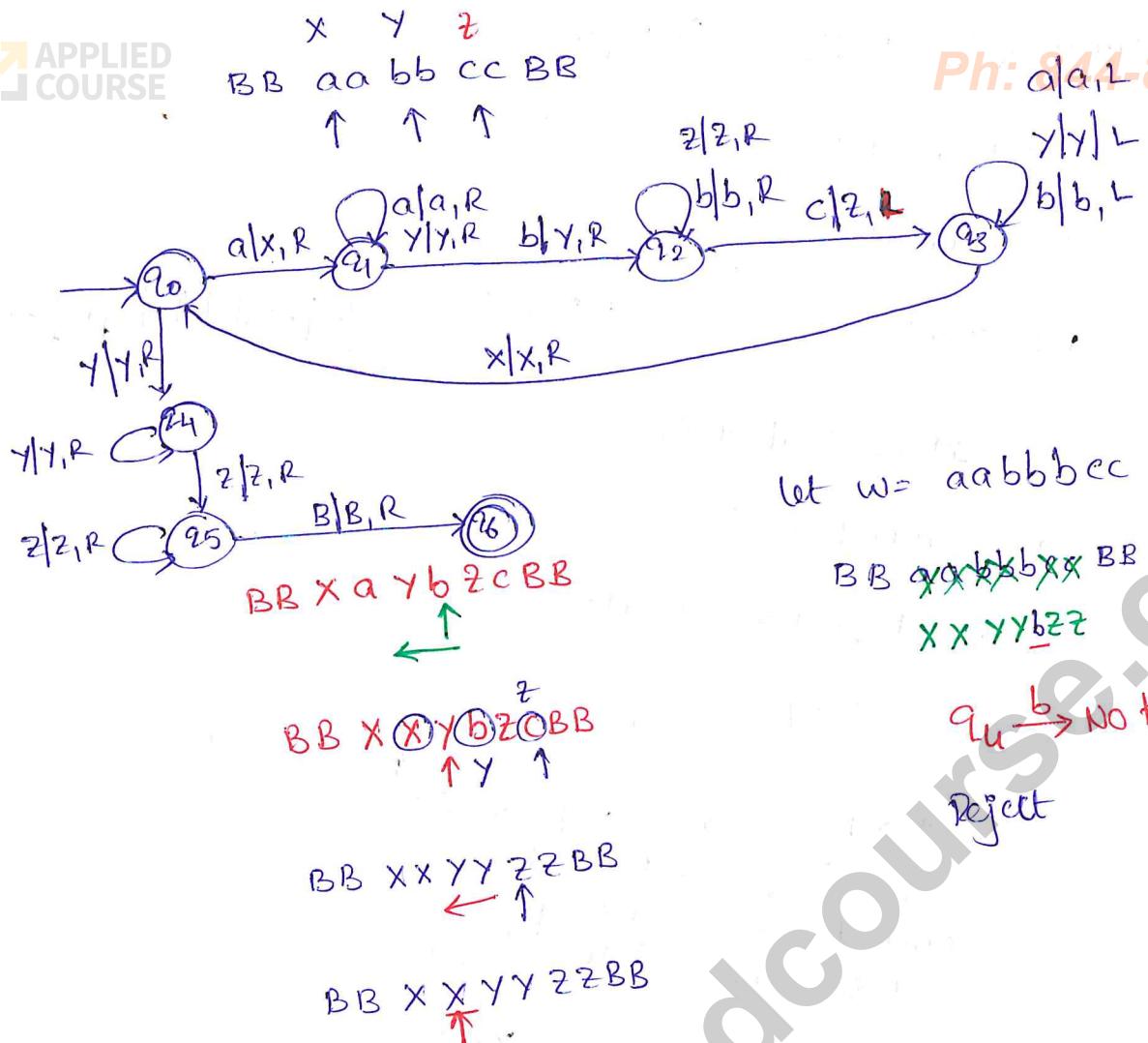


aabbcc
c-program

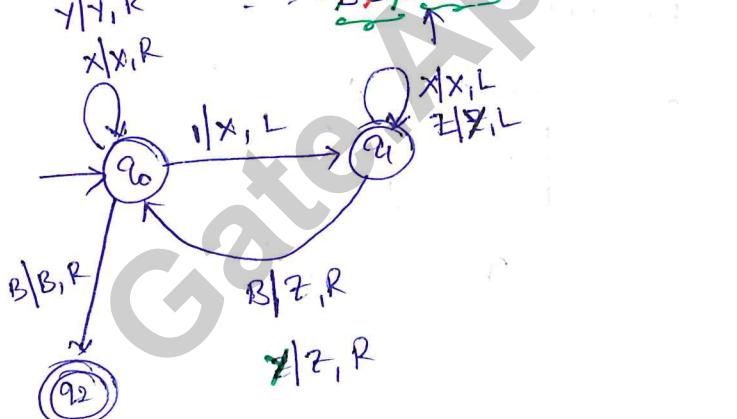
TM \approx Computer

BB $\xrightarrow{\text{red}} a \xrightarrow{\text{green}} b \xrightarrow{\text{green}} b \xrightarrow{\text{red}} c \xrightarrow{\text{red}} c \xrightarrow{\text{green}} c \xrightarrow{\text{green}} BB$
Invalid String

154



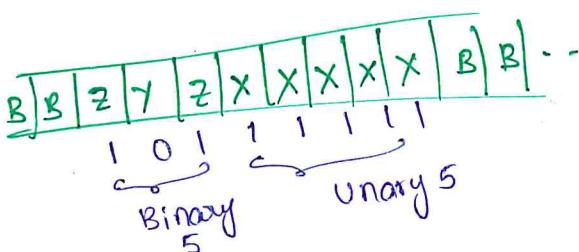
Eg 2: Convert unary to binary

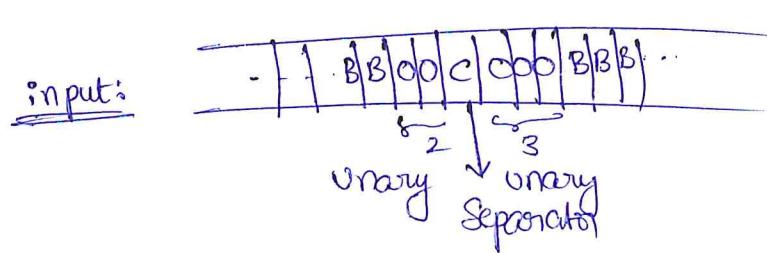


$11111 \rightarrow 101$ Base₂

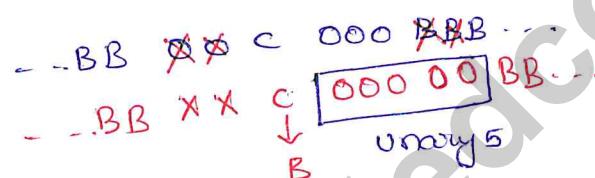
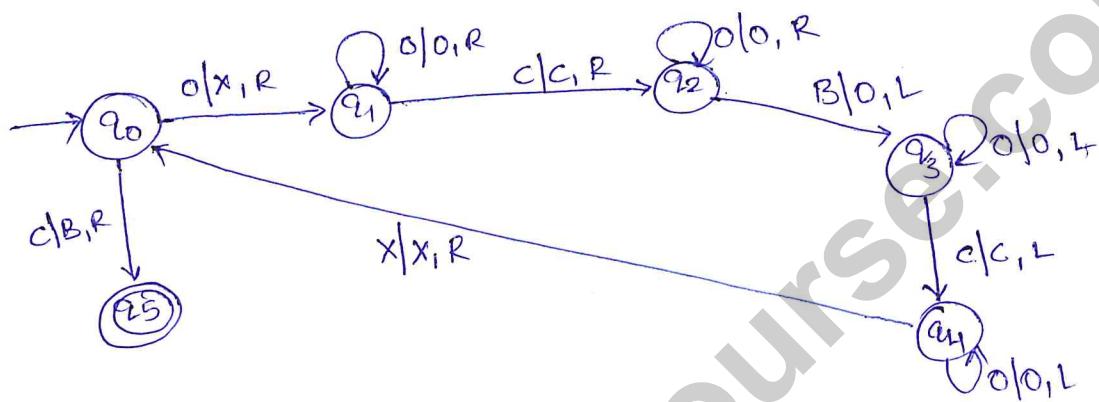
U	B
1	1 : z
11	10 : zz
111	11 : zzz
1111	100 : zzzz
11111	101 : zzzzz

$$\begin{cases} z=1 \\ y=0 \end{cases}$$





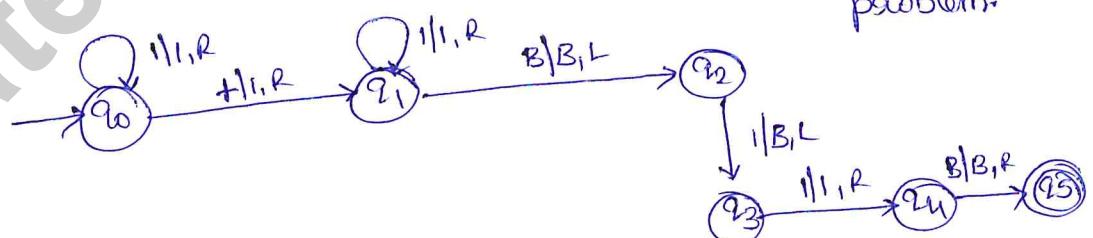
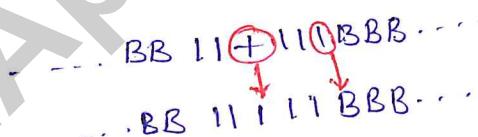
output: ..BB~~0~~0000BB...



Input and output machine.

TM for unary addition

input:



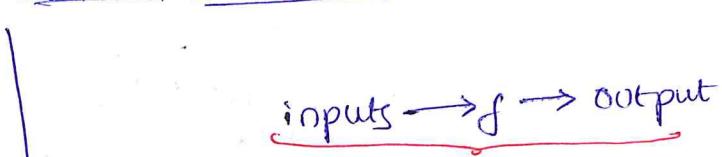
e-programs



different programs
solve the same
problem.

TM as a transducer:

math-juns : addition, subtraction, multiplication



Languages $\{a^n b^n c^n | n \geq 1\}$

simulation (univ TM)

e-program

factorial

ex

recursive func: fib(n)

$$\begin{aligned} & fib(n) \\ & fib(n-1) + \\ & fib(n-2) \end{aligned}$$

Halting & Acceptance:

Stop = Halt

Undecidable

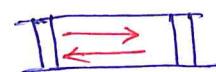
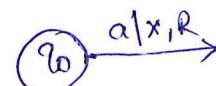
① Final state + Halt \rightarrow Accept

② Non-final + Halt \rightarrow Not-accept

③ α -loop (B) No Halt \rightarrow cannot conclude

undecidable

$w \in L(TM)$
 (B)
 $w \notin L(TM)$



while(1)

?

c:
 |

DTM & NFM:

Default

DTM = NTM (same power)

More steps/transitions

= DFA
NFA

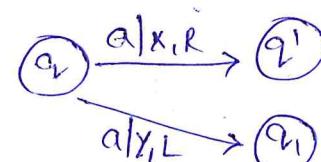
DPDA
NPDA

\downarrow
DCFL
CFL

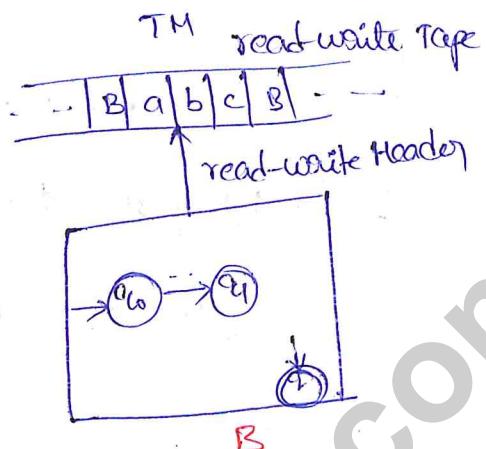
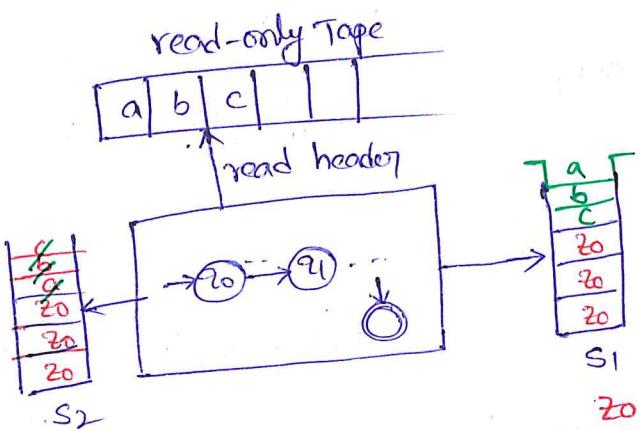
DTM: $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$



NTM: $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$



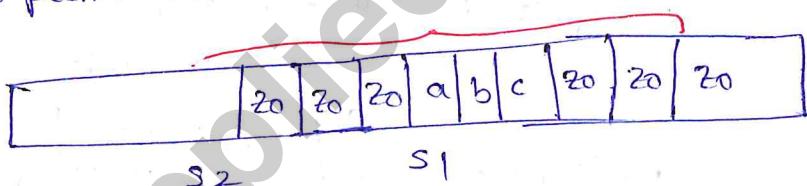
PDA with 2-stacks:



2-stacks

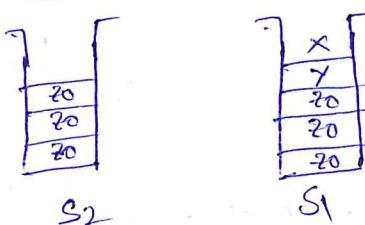
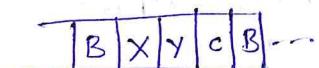
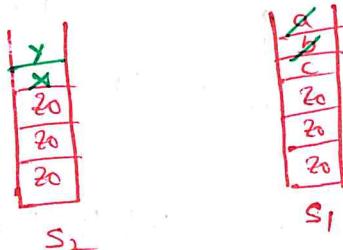
PDA with 2-stacks = TM?

- ① push z0's onto both of the stacks
- ② push input word onto S2
- ③ pop and push w onto S1

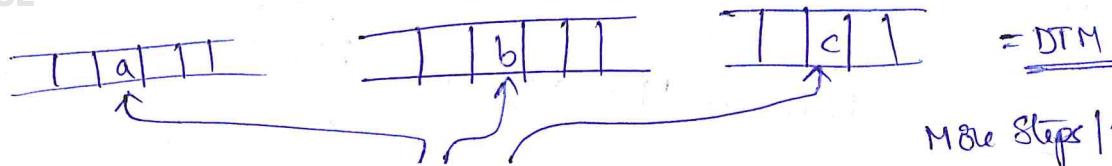


$$TM \quad \delta(q_1, a) = (a|x, R)$$

$$\delta(q_1, b) = (b|y, L)$$



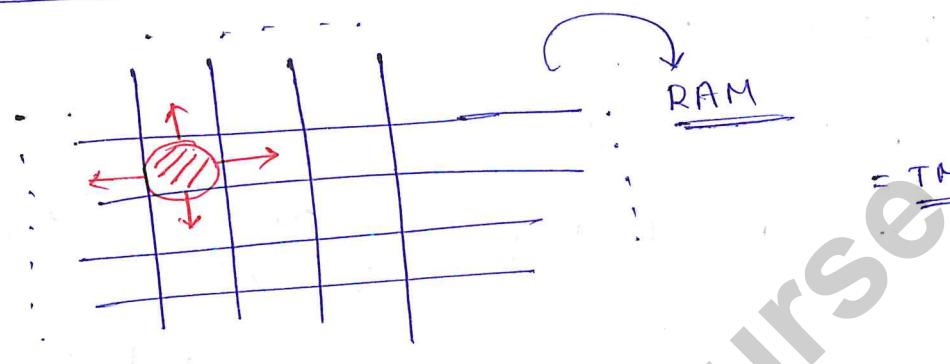
$$\delta: Q \times F_R \times \Gamma_L \rightarrow Q \times \Gamma_P \times \Gamma_L$$



More Steps | Transitions

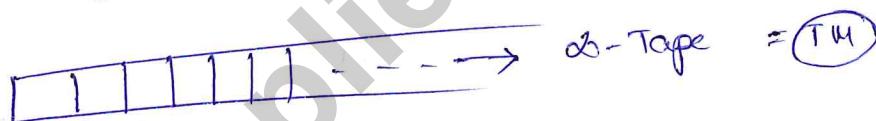
$$\delta(q_i, a, b, c) = (q'_i, x_i, y_i, z, L, R, L)$$

Multi-dimensional TM:

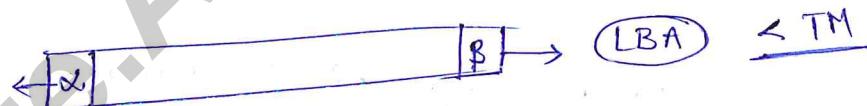


$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, U, D\}$$

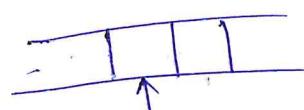
Semi-Infinite TM:



LBA:



TM with stay-option:



$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$$

↓
Stay

\hookrightarrow Recursive Language | Rec Sets (REC)

PDA \leftrightarrow CFLTM \leftrightarrow ?

\hookrightarrow Recursively Enumerable Languages (REL)

\hookrightarrow Chomsky [Unrestricted Grammar]
 $\alpha \rightarrow \beta$

Recursive Language (REC)

L TM

$w \in L$, TM accepts $w \Rightarrow$ TM reaches a final state & halts.

$w \notin L$, TM reaches a non-final state & halts

\downarrow
TM rejects w

Recursively Enumerable Language (REL):

L TM

$w \in L$ then TM reaches a final state & halts
 \downarrow
w accepted

$w \notin L$ then TM reaches a non-final state & halts
 α -loop (A)

does $w \in L$? membership property

If L is a Recursive Language, then we can decide on the membership property.

If L is REL; $w \rightarrow$ TM $\begin{cases} \text{Final + halt} \\ \text{Nonfinal + halt} \end{cases}$
 α -loop

then we cannot

definitely decide if $w \in L$ or not?

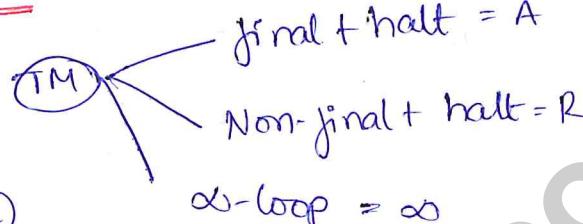
Decidable for Recursive Languages.

CLOSURE properties of REC:

① Union of REC

$$L_1 = L(M_1), L_2 = L(M_2)$$

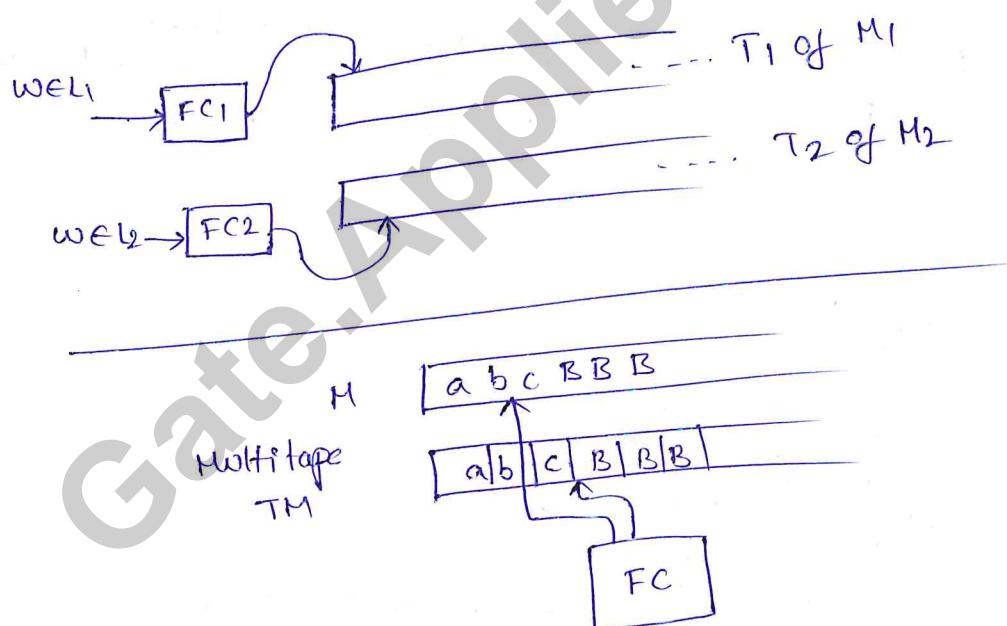
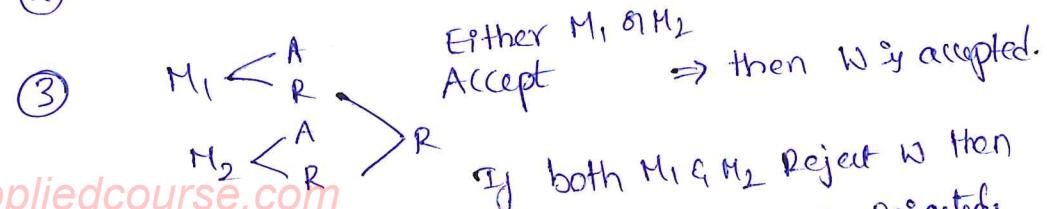
$\swarrow A \quad \searrow R$

Is $L_1 \cup L_2 = ?$ 

$$L(M)$$

$\swarrow A \quad \searrow R$

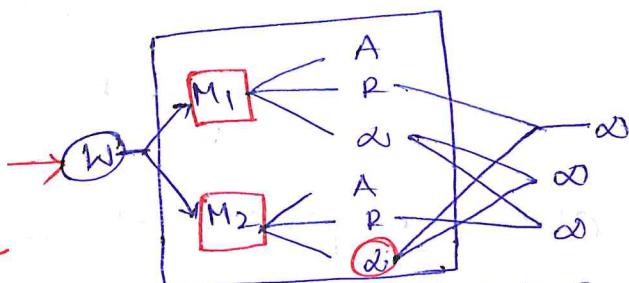
$$\begin{aligned} STM &\approx \text{semi-}\omega\text{-TM} \\ &\approx \text{Multitape TM} \end{aligned}$$

① Copy w from T₁ to T₂② Parallelly Simulate M₁ on T₁, simulate M₂ on T₂

$$\text{REL} \left\{ \begin{array}{l} L_1 = L(M_1) \\ L_2 = L(M_2) \end{array} \right.$$

$w \in L_1 \cup L_2 \rightarrow A \checkmark$

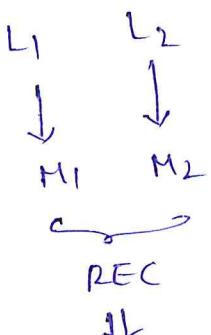
$w \notin L_1 \cup L_2 \rightarrow \begin{cases} R \checkmark \\ A \checkmark \end{cases}$



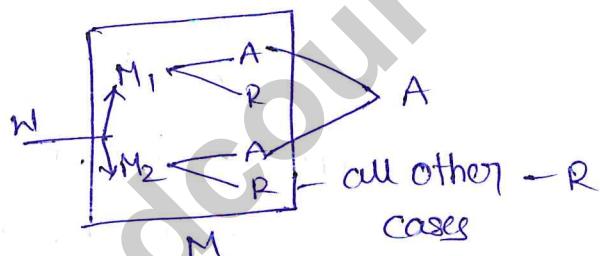
M Block diagram

any one accepts - A(cept)
Both Rejects - Reject

② Intersection:

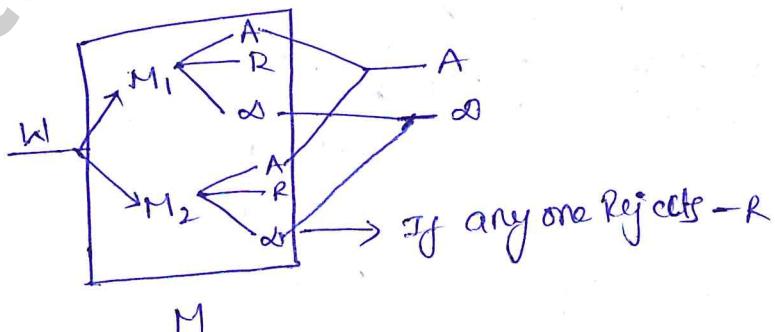


closed under intersection

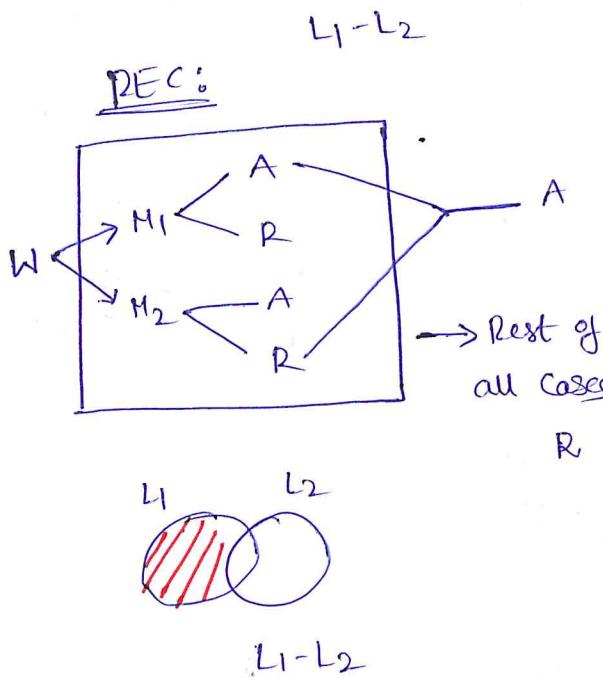


$w - M_1 - R \Rightarrow w \notin L_1$ } Rejected
 $w - M_2 - A \Rightarrow w \notin L_2$ }

Intersection of REL:



RELS are closed under intersection.

Complement :-

$$L^c = \Sigma^* - L \rightarrow \text{Recursive}$$

↳ Reg Lang

\Downarrow

Recursive

$$L^c = \text{REC} - \text{REC}$$

∴ Recursive Languages are also closed under Complement operation.

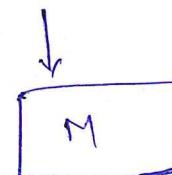
④ Concatenation: (REL)

$$x \in L_1 (M_1)$$

$$y \in L_2 (M_2)$$

$$\text{let } w = xy$$

$$w = 1011110$$

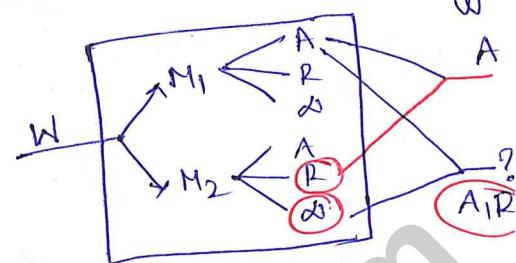


$$x = 1011$$

$$y = 1110$$

break-point

REL: Not closed under Set-difference



$$w \in L_1 - L_2$$

$$\Rightarrow w \in L_1 \text{ & } w \notin L_2$$

If M_1 accepts w , and M_2 stuck for w loops

for few/some words we cannot decide

it is accepted (1)
Rejected

$$L^c = \Sigma^* - L$$

REL REL

\hookrightarrow

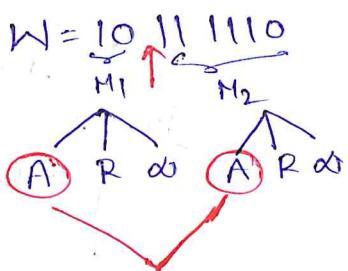
Not closed

undecidable
by RELS

for each possible break point.

M

↑
can we
construct
a TM & not?



A & A → A

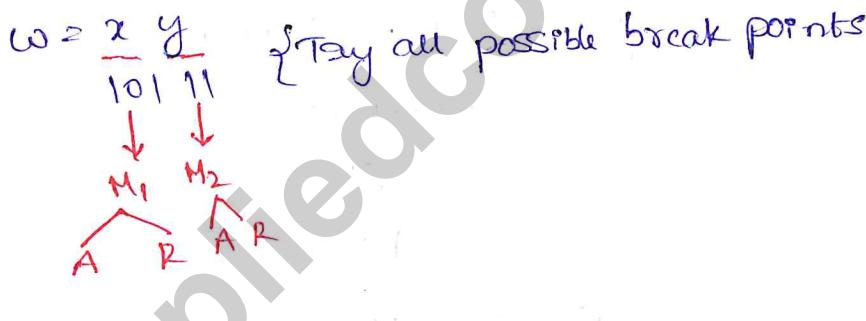
R & R → R one of them

∅ & ∅ → ∅ Reject

$$W = 1011 \underset{\text{break point}}{\uparrow} 1110 \quad x = 1011 \\ y = 1110$$

④(b) Concatenation of REC:

Concatenation is closed for Recursive Languages.



{ Try all possible break points }

$$\begin{array}{l} M_1 \quad M_2 \\ A \& A \rightarrow A \\ R \& - \rightarrow R \\ - \& R \rightarrow R \end{array}$$

⑤ Kleene Closure: → REC, REL are closed

$$L_1^* = \epsilon, W_1, W_2 W_1, W_1 W_2 W_3, \dots$$

try multiple break points & all possible combination of positions

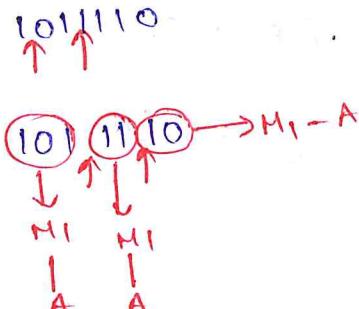
$$L_1 = \{101, 11, 10\}$$

$$W = 101 \underset{\uparrow}{11} 10$$

$$W \in L_1^*$$

if all strings are accepted
→ accept

Any string is rej ⇒ Reject



Mail: gatecse@appliedcourse.com
All strings do loop then do loop

Closed under:

- ① Union
- ② Concatenation
- ③ Intersection
- ④ Complement
- ⑤ Inverse-Homomorphism
- ⑥ Reversse
- ⑦ Intersection with RL
- ⑧ Kleene closure
- ⑨ Set-difference

Not-closed under

- ① Homomorphism
- ② Substitution
- ③ Quotient with RL

Recursively Enumerable LanguagesClosed under:

- ① union
- ② Concatenation
- ③ Intersection
- ④ Substitution
- ⑤ Homomorphism
- ⑥ Inverse-Homomorphism
- ⑦ Intersection with RL
- ⑧ Quotient with RL

Not closed under

- ① set-difference
- ② Complement

Note: If L and L^c are Recursively Enumerable Languages then L must be

Ph: 844-844-0102

Recursive:

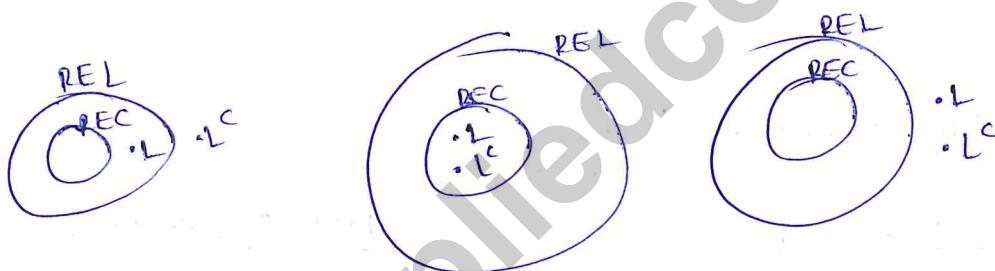
② $L, L^c = \Sigma^* - L$

- valid }
 { ① $L \& L^c$ are REC
 ② L and L^c are not Recursively Enumerable
 ③ L is REL & L^c is not REL

X ④ L is REL & L^c is also REL

↓

NOT True for all Languages.

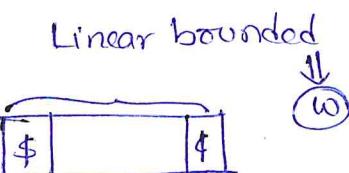


Linear Bounded Automata & Context Sensitive Languages:

LBA \leftrightarrow CSL \leftrightarrow CSG
Chomsky Hierarchy of Languages.

LBA = NTM + Bounded Tape + $\alpha \rightarrow \beta$

Read-write :
Tape
Left end Marker Right end marker



$$|\alpha| \leq |\beta|$$

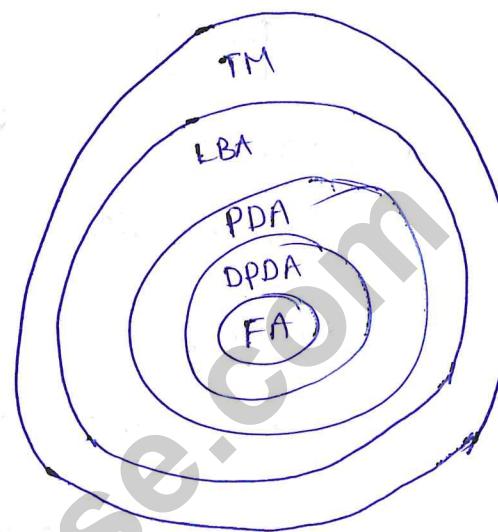
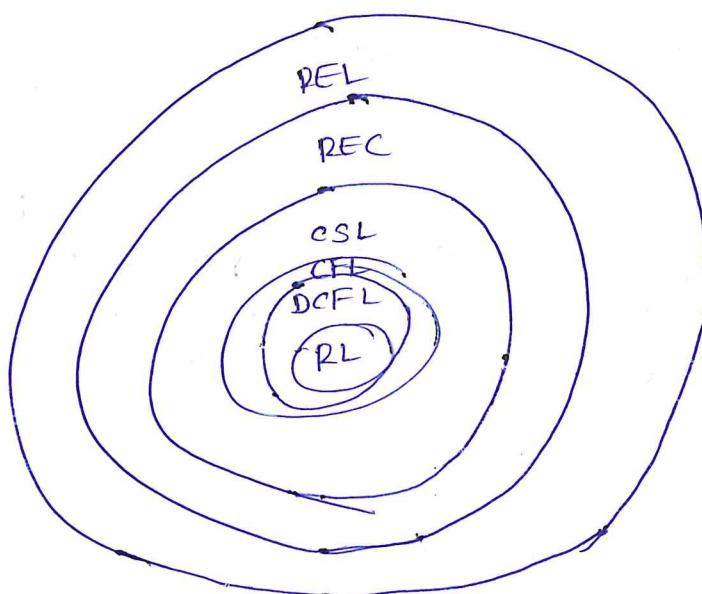
$$\alpha, \beta \in (V+T)^*$$

$$\xleftarrow{\text{Linear in } |w|} \xrightarrow{\text{Linear in } |w|}$$

$$a|w| + b$$

$$a|w|^2 + b|w| + c$$

$$\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R\}}$$

CSG & CSL

$$L = \{a^n b^n c^n \mid n \geq 1\} \rightarrow \text{CSL}$$

$\underbrace{\quad}_{3n}$

LBA

\boxed{aabbc}

Closure properties of CSL:CSLs are closed under

- ① positive closure (L^+)
- ② union
- ③ intersection
- ④ concatenation
- ⑤ reversal
- ⑥ inverse Homomorphism
- ⑦ intersection with PL

not closed under

- ① Homomorphism
- ② substitution

Undecidability & Computational Classes :-

Real-world implications

NP-Complete

NP-Hard

P

↳ Halting problem

T.M

i) L That is non-Recursively Enumerable

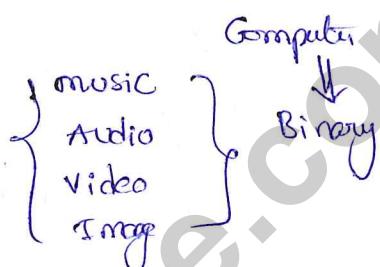
ii) L is REL but not REC

Deep & Tricky

Reducibility

Encoding a TM :- (Using binary strings)

$Q, \Sigma, \Gamma, q_0, F, \{L, R\}, \delta, B$



No q_0 } q_1 : Initial state
} q_2 : Final state
} q_3, q_4, \dots : other states

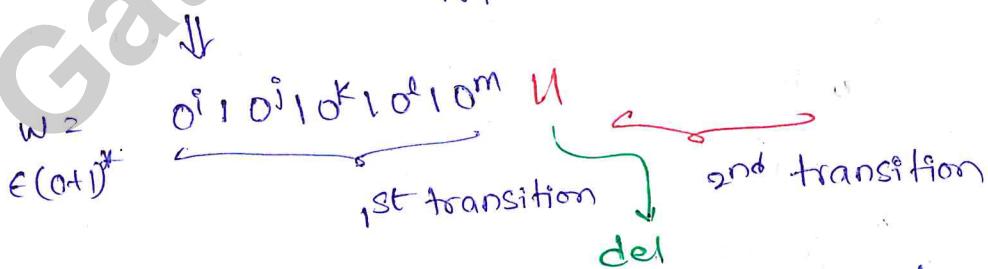
No x_0 } $x_3 : B$
} $x_1, x_2, x_4, x_5, \dots$: other symbols

$no \quad D_0 \rightarrow L : D_1 \quad R : D_2$

$$\delta(q_i, x_j) = (q_k, x_l, D_m)$$

$$(q_i, x_j, q_k, x_l, D_m)$$

$i, j, k, l, m > 0$



TM : binary string = binary number \Rightarrow decimal number
 base_2

0th TM, 1st TM, 2nd TM, ..., TM_i

Not every binary string is a TM

11001 = not a valid TM

Ph: 844-844-0102

$(2)_{10} = (10)_2$ ≠ any valid TM binary encoding



$$L(TM) = \emptyset$$

$$TM_p = \{TMO, TM_1, \dots\}$$

Diagnolization Language : A Non-REL

Ld \Rightarrow No TM to recognize

$$w \in L_d$$

$$L_d = \{ w_i \in \{0,1\}^* \mid w_i \notin LCM_i \} \xleftarrow{\text{Super Grouve}} \xleftarrow{\text{pouf : Genius}}$$

Super Creative

Set of strings which are not accepted by a machine represented by the same string.

IS Ld a REL?

三

\Downarrow if $w \in L$ then \exists TM M such that $w \in L(M)$

$$w \in L_d \rightarrow \text{Halt} + \text{Accept}$$

$w \in L_d \rightarrow$ Halt + Reject

diagonalization

H_3
 TM_2

$$d = \text{diag}(T)$$

$$= \boxed{1 \ 0 \ 1 \ \dots \ 1}$$

$$\begin{cases} 1 & \text{if } w_i \in L(TM^i) \\ 0 & \text{if } w_i \notin L(TM^i) \end{cases}$$

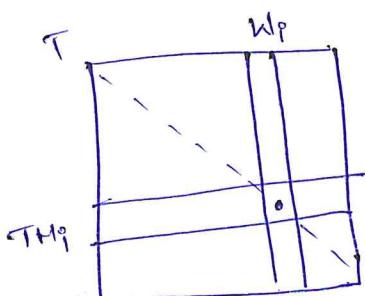
Binary Matrix

Mail: gatecse@appliedcourses.scr

$$d = \text{diag}(T)$$

$$d^1 = \begin{array}{cccccc} & & & & & i \\ \boxed{} & 1 & 1 & 1 & 1 & \end{array}$$

↑ if $w_i \in L(TM^i)$



↳ 0 if $w_i \notin L(TM^i)$

$$d^1[i] = 1 \Leftrightarrow w_i \in L(TM^i)$$

$$L_d = \{w_i \mid d^1[i] = 1\}$$

★ IS L_d in REL? Non-REL

↔ Is there a TM that accepts L_d ?

↑

↔ Is there a row in T , which is equal to d^1 -vector.

↑

Every TM
corresponding
to i TM i

Let i th row be the row in T , which is equal to d^1 -vector
↓ does not exist

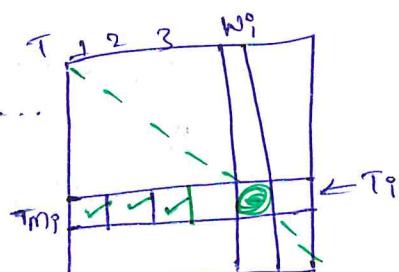
Proof by

contradiction

$$\Rightarrow T^i = d^1$$

$$\Rightarrow T^i[j] = d^1[j] \quad \forall j = 1, 2, 3, \dots$$

$$\Rightarrow T^i[j] = (d[j])'$$



$$\Rightarrow \exists j \quad j=i$$

$$T^i[j] = d[j] = (d[j])'$$

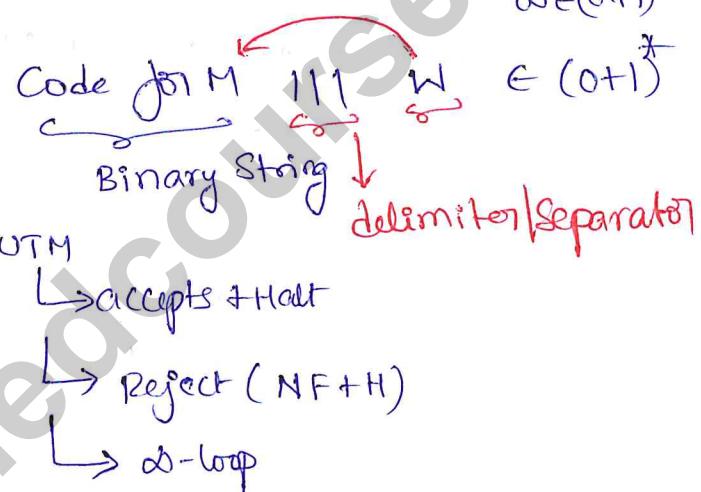
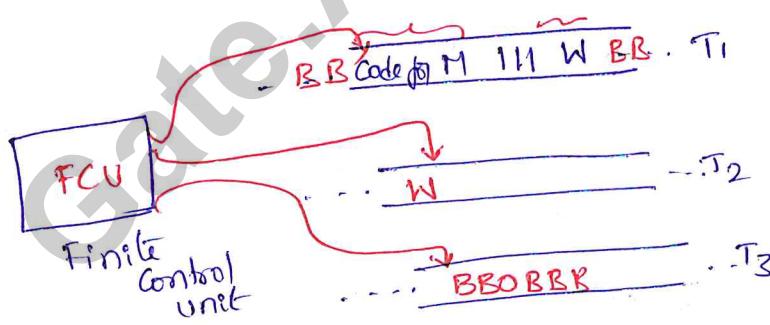
contradiction

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

TM: programUTM: General purpose ComputerNon-REL
• Ld

UTM: TM that takes a TM-Code binary representation of a TM and w as inputs and it accept the inputs (word) iff $w \in L(M)$

Input to a UTM:
on some inputs
 $L_u = \{L(U)\}$
↓
universal language.

Working of UTM U:

to simulate the Tape of M
Current State of M during Simulation

① Check if M is a valid TM

→ If NO $L(M) = \emptyset$

$\Rightarrow w \notin L(M)$

UTM halts + Reject

→ Yes: Continue

- ② Copy w from T_1 to T_2
- ③ Initialize T_3 with initial state of M
- ④ Simulate TM M using $T_1, T_2 \text{ & } T_3$

$$\delta(q_1, 0)$$

↓

$$x_1 : 0$$

$$\delta(q_1, x_1) = (q_2, x_2, R)$$

$$x_2 : 1$$

$$x_3 : B$$

→ Scan the code for M

→ 11 0101 0010010011

$$\delta(q_2, 0)$$

↓

$$\delta(q_2, x_1) = ?$$

→ 11 00101

(F+H)

- ⑤ If M accepts w
then UTM also accepts input

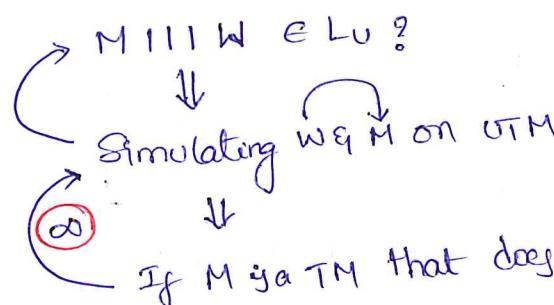
↓
Code for $M \mid \mid w$

- ① Is L_U REC (i) REL (ii) Non-REL?

\downarrow
 $L_U = L(UTM)$ $\exists aTM = UTM$ that
accepts L_U
 $UTM = TM$

(Total TM)
decision

Intuitive:



Reduction: [Most useful ideas], Algorithms (P, NP-...)

Let's assume L_U is REC
 \Rightarrow Decide if $w \in L_U$ using a TTM (M_u) (①) Algorithm (A_w)

Let's take L_d (is Not REL) \rightarrow diag

Instance of Membership in L_d

problem: $w \in L_d ?$

① IS $w \notin L(M_w)$?

Algorithms: ① If M_w is not valid

$$\Rightarrow L(M_w) = \emptyset$$

$$\Rightarrow w \notin L(M_w)$$

$$\Rightarrow w \in L_d$$

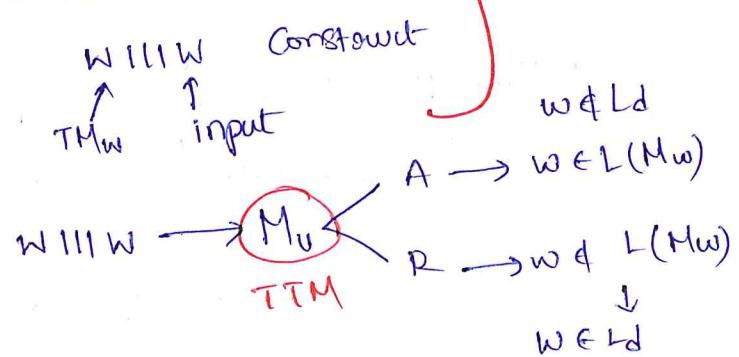
Reduction
Alg

Red-TM

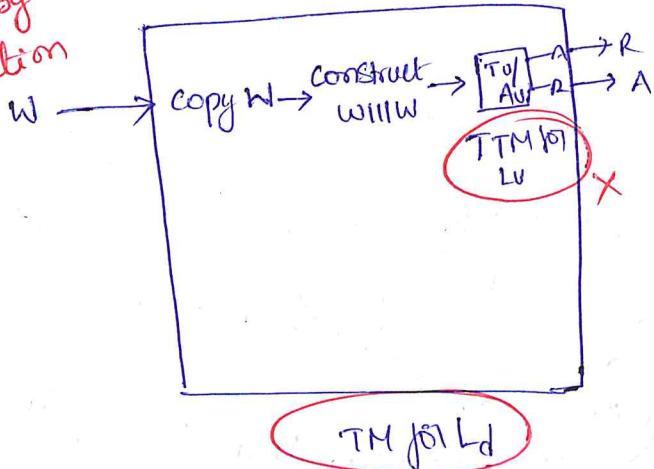
② If M_w valid

$w \parallel\!\!\parallel w$
TM_w input

Construct



*proof by
contradiction*



$A \rightarrow B$ Reduction-diagram
Ph: 844-844-0102

If \exists TTM for L_U then

} I can construct a TM which accepts L_d .

we know that \nexists TM that accepts L_d

Reduction: Instance of $w \in L_d$

↓ Algorithm / TTM Reduction

Instance of $w \in L_U$

assumption \exists Solution

\exists TTM

Non-REL



L_d

REC-decidable

If there is a reduction from P_1 to P_2

instance
of P_1

Then

① P_1 is undecidable $\Rightarrow P_2$ is also undecidable.

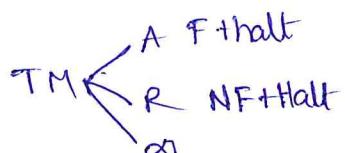
② P_1 is Non-REL $\Rightarrow P_2$ is also Non-REL

$w \in L_d$?

Halting problem: Does a TM, M halt on input w ?

Undecidable

\Downarrow
IS $w \in L(M)$



$w \in L(M) \rightarrow$ halt + Accept

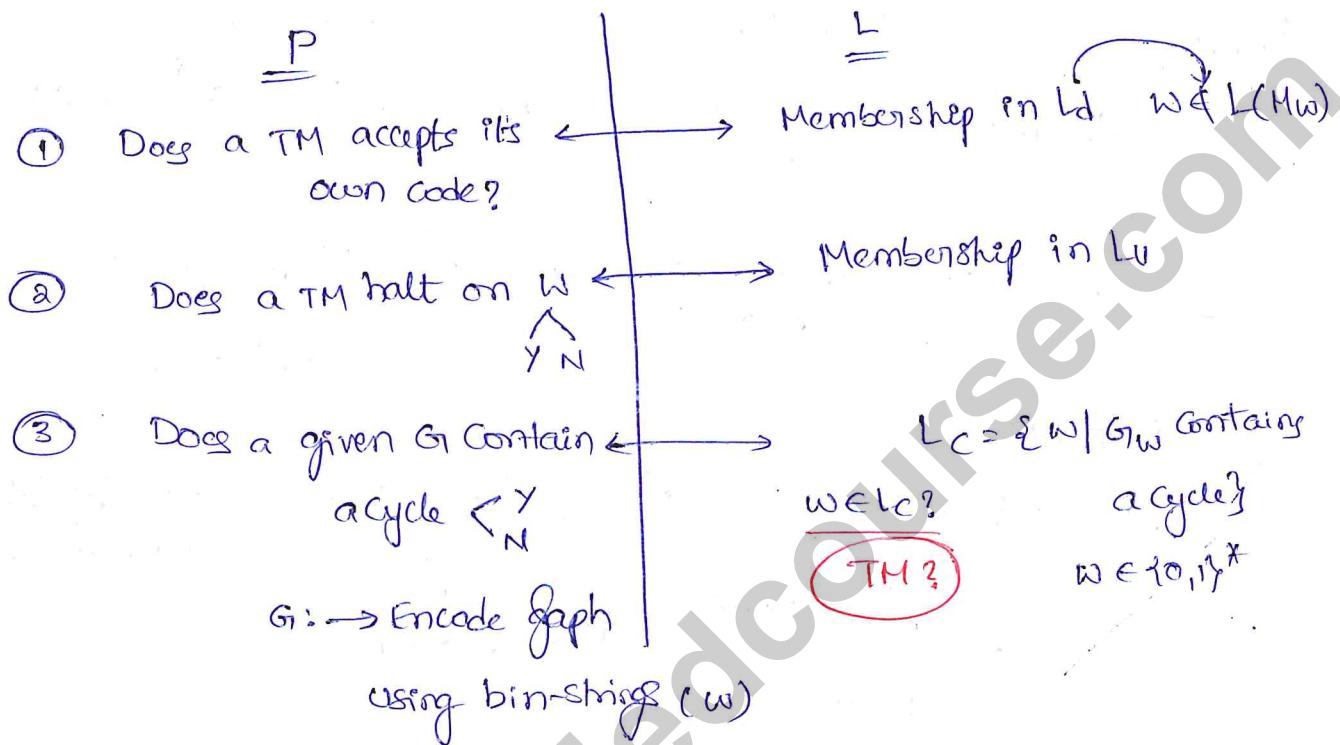
$\neg w \in L(M) \rightarrow R \& \infty$

IS $M \text{ M } W \in L_U$
 \Downarrow
 [Undecidable]

L_U is REL

& not REC

problems ~ Languages



F(GU) $\xrightarrow{\text{TM}}$ Graph

L_e and L_ne:

\uparrow binary-rep of TM.
 $L_e = \{M \mid L(M) = \emptyset\}$

$L_d \downarrow$
 $L_u \downarrow$
 Non-REL REL

$M \in \{0,1\}^*$

L_{ne} \leftarrow Non-REL?
 REL?
 REC?

$L_{ne} = \{M \mid L(M) \neq \emptyset\}$

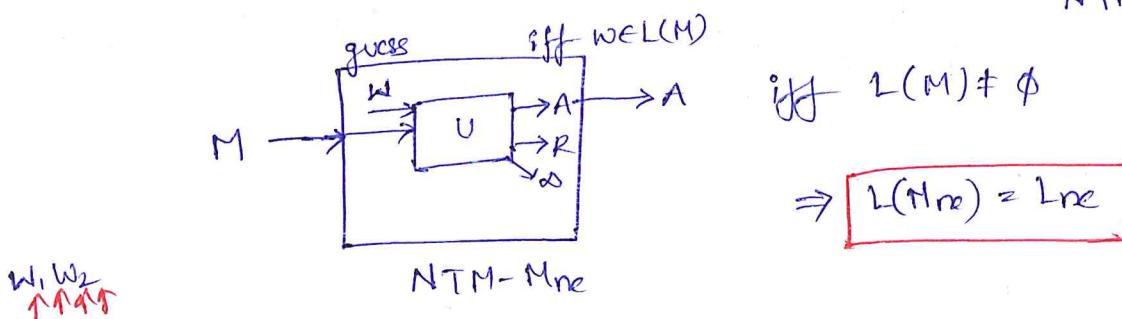
binary string

$$\left\{ \begin{array}{l} L_e = L_{ne}^c = \overline{L_{ne}} \\ L_{ne} = L_e^c = \overline{L_e} \end{array} \right\}$$

\uparrow Closure properties of REL & REC

Is L_{ne} a PFL? \Rightarrow does \exists TM s.t. $L_{ne} = L(\text{TM})$

Ph: 844-844-0102

 $NTM \sim DTM$ 

{ similar strategy to prove closure prop of Concatenation }

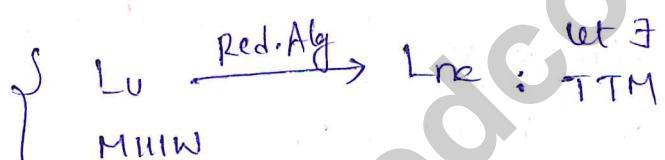
Is L_{ne} a REC-Language? $\Leftrightarrow \exists a \text{ TTM, TM set } L(\text{TM}) = L_{ne}$

Guess: No

→ \nexists a TTM

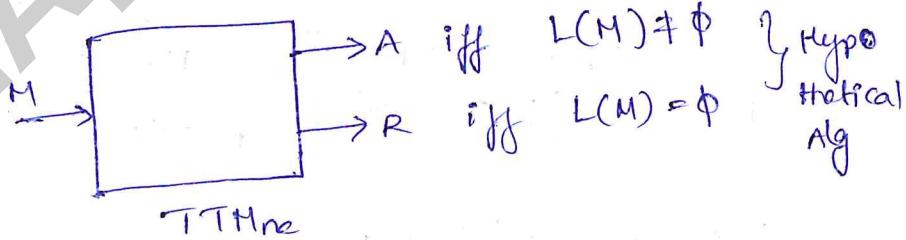
Prove: Reduction

L_U : not REC but PFL: undecidable language.



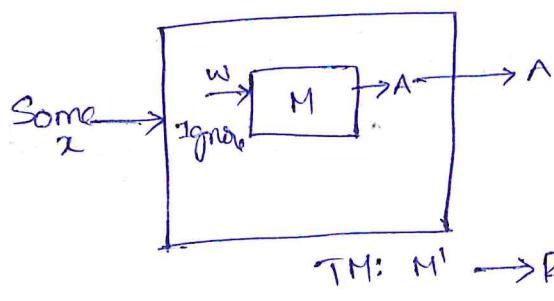
Let's assume L_{ne} is REC Language.

$\Leftrightarrow \exists$ an algo | TTM for L_{ne} to determine if $L(M) \neq \emptyset$

 \Leftrightarrow 

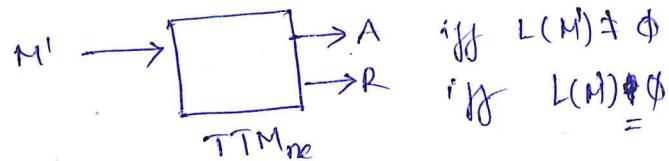
Reduction Algorithm: $1111w \in L_U \rightarrow$ word in L_{ne}

①



M' accepts some x
iff $w \in L(M)$

$TM: M' \rightarrow$ represent M' as a bin-string.



iff $L(M') \neq \emptyset$

iff $L(M) \neq \emptyset$

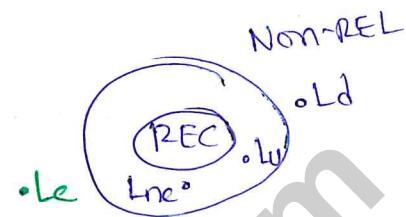
→ w.d $L(M)$

Ph: 844-844-0102

Assumption is incorrect.

$\exists a \in TM_{ne} \Rightarrow L_{ne}$ is not REC

$\exists a \in TM_{ne} \Rightarrow L_{ne}$ is REL



$$L_e = \overline{L_{ne}} = L_{ne}^c$$

if L_e is REL

$$L_e = L_{ne} \text{ is REL}$$

L_{ne} is not REC

RELs are not closed under Complement

$L_e \cap L_{ne}$ are REC

REC are closed under Complement

\Rightarrow L_e is non-REL

$L_e \& L_{ne}$: Special case of Rice Theorem

Note: Reduction:

$L_u \ L_{ne}$

① If \exists a Reduction from P_1 to P_2 , then

ⓐ if P_1 is undecidable, then P_2 is also undecidable.

L_u

L_{ne}

ⓑ if P_1 is non-REL then P_2 is also a non-REL

\downarrow
 L_d

\downarrow
 L_e

Note for GATE:

Ph: 844-844-0102

- ⊗ Covered the most important ways to prove in earlier videos.
- ⊗ Previous Gate Questions: Mostly about properties & applying them.
- ⊗ About the proofs, please refer text books
 - ↗ Hopcroft, Ullman
 - Michael Sipser

Property of a REL:

= Set of REL that satisfy a property of P

$$\left\{ \begin{array}{l} P_{CFL} = \text{Set of REL that are also CFL} \xrightarrow{\text{REL \& P}} P_{CFL} = \{L_1, L_2, \dots\} \\ P_{\text{INF}} = \text{Set of REL that are } \omega\text{-Languages.} \end{array} \right.$$

Trivial property:

$P = \emptyset$ or $P = \text{all recursively enumerable Languages.}$

Rice Theorem: Every non-trivial P of REL is undecidable.

1960's

Undec \Leftrightarrow Not REC

\Leftrightarrow REL & non REL

$L_e, L_{ne} \xrightarrow{\text{REL}}$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 Non REL

L_u
(undec)

$\xrightarrow{\text{Reduce}}$

L_p ?

$P = \{L_1, L_2, \dots, L_f, \dots\}$

$\downarrow \quad \downarrow \quad \downarrow$
 $M_1, M_2, \dots \quad \text{satisfy}$

$\circledcirc M_i$

$L_p = \{M_i \mid L_i \in P\}$

① P: whether a Language accepted by a TM is Empty

$$L_e = \{ M \mid L(M) = \emptyset \} \rightarrow \text{Undecidable}$$

↳ Non-Rel

② P: Lang accepted by TM is Finite

$$L_p = \{ M \mid L(M) \text{ is Finite} \} \rightarrow \text{Undecidable}$$

③ P: Language accepted by a TM is Regular \rightarrow Undecidable

P: Language accepted by a TM is CFL \rightarrow Undecidable.

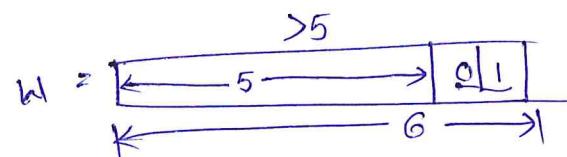
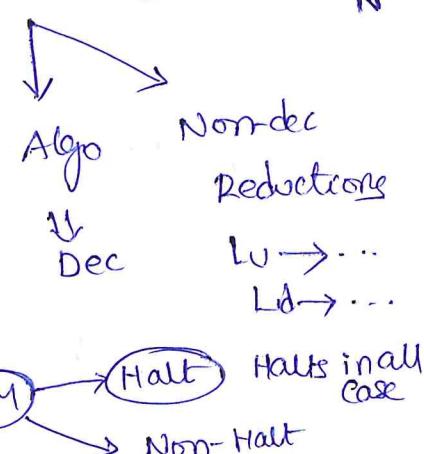
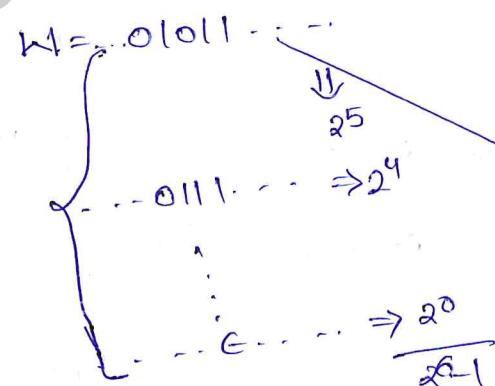
④ P: whether/Does a TM have 5 State? Decidable

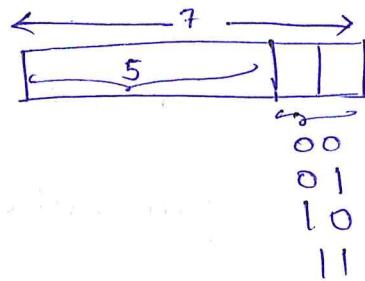
Transition Graph/Table/Encoding

we can write an algorithm to count
the number of State in a TM.

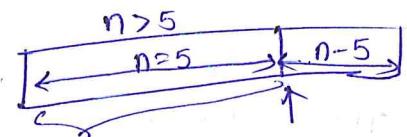
⑤ Does \exists an input w s.t. TM makes atleast 5 moves? ∇

cannot apply RICE Theorem.





Decidable problem

Post's Correspondence Problem: (PCP) : Strings

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

→ Undecidable → TM
Lu

$$|A| = |B|$$

Corresponding pair: (w_i, x_i)

List		A	B
i	w _i	x _i	
1	1 w ₁	11 x ₁	
2	10111 w ₂	10 x ₂	
3	10 w ₃	0 x ₃	

$w_2, w_1, w_3, x_2, x_1, x_3$
 $10111110 \quad 10111110$ Equal nrg of A: List of Strings of Σ
 Strings B: List of Strings of Σ

$$w_2, w_1, w_3 = x_2, x_1, x_3$$

$$10111110 \quad 10111110$$

$$2, 1, 1, 1, 3$$

A: List of words in Σ^* = w_1, w_2, \dots, w_K B: " " " = x_1, x_2, \dots, x_K

PCP has a solution iff ∃ a sequence of indices

$$i_1, i_2, \dots, i_m$$

set

$$w_{i_1}, w_{i_2}, w_{i_3}, \dots, w_{i_m} = x_{i_1} x_{i_2} \dots x_{i_m}$$

	A	B
i	Wi	Xi
1	10	101
2	011	11
3	101	011

No Solution

$$\textcircled{1} \quad i_1 = 1$$

$$10 \quad 101$$

$$\textcircled{2} \quad i_2 = 1 \quad 1010 \times 101101$$

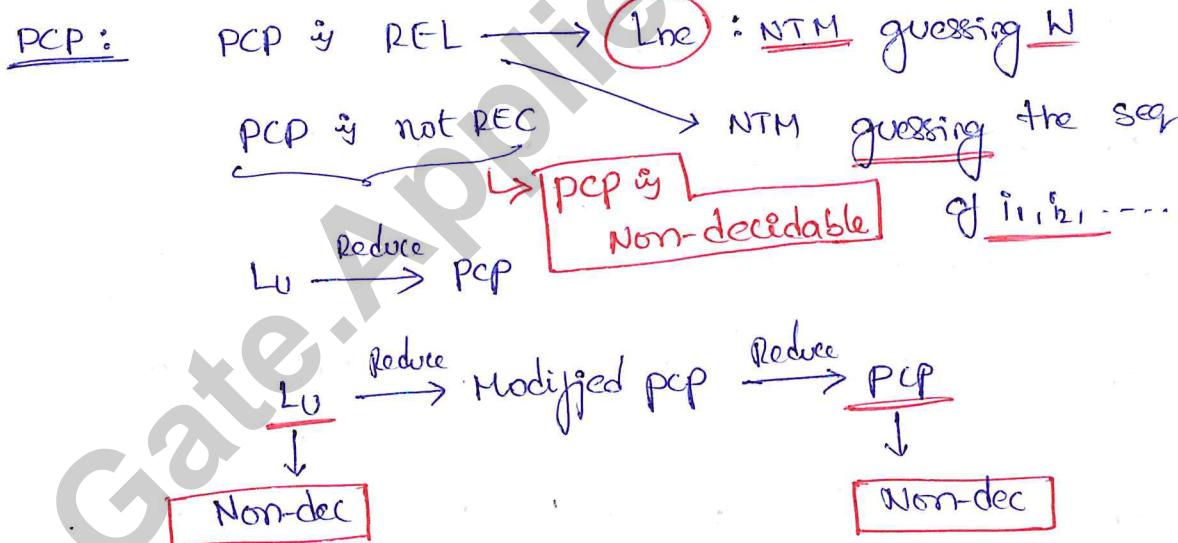
$$i_2 = 2 \quad 10011 \times 10111$$

$$i_2 = 3 \quad \boxed{10101} \quad \boxed{101011}$$

$$(1, 3; 3, \dots)$$

$$\textcircled{3} \quad i_3 = 1 \quad 1010110 \quad 101011101 \quad i_3 = 3$$

$$i_3 \neq 2$$

Logically:

$$\nexists \text{ a Seq } i_1, i_2, \dots, i_m$$

$$\Downarrow$$

$$\infty\text{-loop}$$

① Membership problem? $L \subset N^*$ does $w \in L?$

② Emptiness : Is $L = \emptyset$?

③ Finiteness : Is $|L|$ finite?

④ Equal

Equivalence : Is $L_1 = L_2$?

Is $L(M_1) = L(M_2)$?

Is $L(G_1) = L(G_2)$?

Is $L(r_1) = L(r_2)$?

⑤ Intersection Empty

$L \cap G \sim M$ - regexp

Is $L_1 \cap L_2 = \emptyset$

⑥ Totality / Completeness Is $L = \Sigma^*$?

⑦ subset : Is $L_1 \subseteq L_2$?

⑧ Intersection-Finiteness : Is $|L_1 \cap L_2|$ finite?

⑨ cofiniteness : given L , Is $|L'|$ finite?

⑩ Regularity : Is L a regular lang?

⑪ Ambiguity problem: $L \sim G$

Is L ambiguous?

\Leftrightarrow Is G Ambiguous
 $\exists w \in L$

Then G is ambiguous

$$G: S \rightarrow a | aAb | absb$$

$$A \rightarrow aAb | bs$$

$$w = abab \quad \textcircled{1} \quad S \rightarrow absb \\ \rightarrow abab$$

$$\textcircled{2} \quad S \rightarrow aAb \\ \rightarrow absb \\ \rightarrow abab$$

(12) Complement problem:

Are both L and \bar{L} of the same type?

are $L \cap \bar{L}$: reg?

are " : DCFL?

" : CFL?

" : REC?

" : CSL?

" : REL?

Decision problem	RegLanguages	DCFL	CFL	CSL	REC	REL
① Membership	D	D	D	D	D	UD
② Emptiness	D	D	D	UD	UD	UD
③ Finiteness	D	D	D	UD	UD	UD
④ Equivalence	D	UD	UD	UD	UD	UD
⑤ Totality/Completeness	D	D	UD	UD	UD	UD
⑥ Subset	D	UD	UD	UD	UD	UD
⑦ Intersection-finiteness	D	UD	UD	UD	UD	UD
⑧ co-finiteness	D	D	UD	UD	UD	UD
⑨ Regularity	D	D	UD	UD	UD	UD
⑩ Ambiguity	D	UD	UD	UD	UD	UD
⑪ Complement	D	UD	UD	D	D	UD
⑫ Intersection Empty	D	UD	UD	UD	UD	UD

① Consider the following problems. $L(G)$ denotes the language generated by a grammar G . $L(M)$ denotes the language accepted by a machine M .

① For an unrestricted grammar G and string w , whether $w \in L(G)$ — UD

② Given a Turing Machine M , whether $L(M)$ is regular. — UD

③ Given two grammars G_1 and G_2 , whether $L(G_1) = L(G_2)$ — UD

④ Given a NFA N , whether there is a deterministic PDA P such that N and P accept the same language. — DCFN

such that N and P accept the same language. — Decidable

which one of the following statements is correct?

A only I and II are undecidable

B only II is undecidable

C only II and IV are undecidable

D only I, II and III are undecidable



② Let \underline{A} and \underline{B} be finite alphabets and let $\#$ be a symbol outside both A and B . Let f be a total function from A^* to B^* . we say f is Computable if there exists

a Turing machine M which given an input $x \in A^*$ always halts with $f(x)$ on its tape. Let L_f denote the language

$$\{ x\#f(x) \mid x \in A^* \}$$

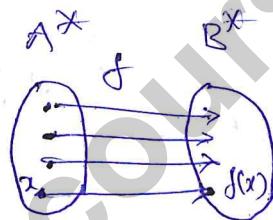
UTM(halts)

which of the following statements is true?

Ph: 844-844-0102

- A: f is Computable if and only if $\{f\}$ is recursive.
- B: f is Computable if and only if $\{f\}$ is Recursively Enumerable \times
- C: If f is Computable then $\{f\}$ is recursive, but not \times
Conversely.
- D: If f is Computable, then $\{f\}$ is Recursively Enumerable, \times
but not Conversely.

always Halts.
 \Downarrow
HALT



- (3) Let $L(R)$ be the Language represented by regular Expression
R. $L(G)$ be the language generated by a Context free Grammar G.
Let $L(M)$ be the language accepted by a Turing Machine M. Which
of the following decision problems are undecidable?
- I Given a regular Expression R and a string w , is
 $w \in L(R)$? —D
- II Given a Context-free Grammar G, $L(G) = \emptyset$? —D
- III Given a Context-free Grammar G, $L(G) = \epsilon^*$ —UD
for some Alphabet ϵ ? Totality / Completeness
- IV Given a TM M and a string w, is $w \in L(M)$? —UD

- (a) I and IV only
 ✗(b) II and III only
 ✗(c) II, III and IV only
 ✓(d) III and IV only.

(4) Which of the following problems are undecidable?

- I. Given NFAs N_1 and N_2 , is $\frac{L(N_1) \cap L(N_2)}{RL} = \emptyset$? D
- ✓II. Given a CFG $G_2 (N, \Sigma, P, S)$ and a string $x \in \Sigma^*$, does $\underline{x \in L(G)}$? D
- ✗III. Given CFG G_1 and G_2 , is $L(G_1) = L(G_2)$? → UD
- ✗IV. Given a TM, M, is $L(M) = \emptyset$? → UD
- M
 E } CFL
 F } Decid-
 able
- (A) I and IV only
 (B) II and III only
 ✓(C) III and IV only
 (D) II and IV only

(5) Consider the following statements

① The complement of every Turing decidable language

\downarrow REC
 is Turing decidable.



② There exist some language which is in NP but not Turing decidable. \exists TTM

③ If L is a language in NP, L is Turing decidable

which of the above statements is/are true?

- A Only 2
 B Only 3
 C Only 1 and 2
 D Only 1 and 3

1 - TRUE

2 - False

3 - True

Computational classes: P and NP, NPC, NP-Hard

TOC - Alg.

problems ~ lang

Class-P: Polynomial : $n, n^2, n^3, n^4, n^5, \dots, n^k$

$\rightarrow \exists \text{DTM}$ that always halts. $\Leftrightarrow \text{DTTM} \Leftrightarrow \exists \text{Det Alg}$

\rightarrow Input : n ; makes 1 halts within $O(T(n))$ moves/step

Big-oh

 $T(n) = \text{polynomial in } n$ DTMstep/Move \leftrightarrow basic operation

$Q_i \xrightarrow{a} Q_j$ (eg) Is w in $L(G)$? $G: \text{CFG} \leftarrow \text{TOC}$
 \Downarrow
 CYK algo $\rightarrow O(n^3)$

(eg) Is \exists a path vertex u to v in Graph

Dijkstra $\rightarrow O(n \log n)$
 $= O(n^2)$

Sorting : $O(n \log n)$ $O(n^2)$ Searching : $O(n)$ $O(\log n)$ $O(1)$

$\rightarrow \exists \text{ NTM that always halts} \Leftrightarrow \exists \text{ NTM}$

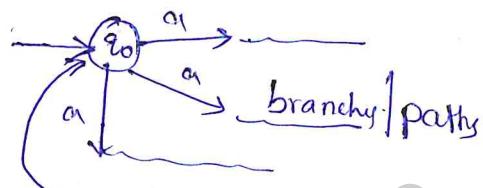
$\rightarrow \text{longest path len is } O(T(n))$

$T(n)$: polynomial in n

DTM: Single threaded program that halts.

\hookrightarrow single path

transition-graph



executed / simulated
parallelly

NTM: Multi-threaded program that halts.

\hookrightarrow OS

$$P \subseteq NP \subseteq \underline{\text{REC}}$$

NTM \sim DTM

Algo

$$\underline{P = NP?}$$

\hookrightarrow Most important

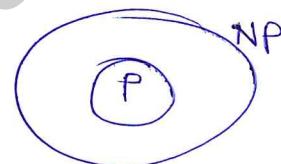
unsolved problems in cs/math.

Many believe $P \neq NP$

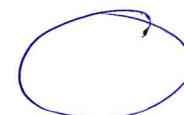
case 1: $P \neq NP$

case 2: $P = NP$

if $P \neq NP$



if $P = NP$



Thousands of known problems in NP

which no one could come up with a det. polynomial time.

Polytime Reductions:

Instance of $P_1 \xrightarrow[\text{(P) poly-time DTM}]{\text{reduction alg}} \text{Instance of } P_2$

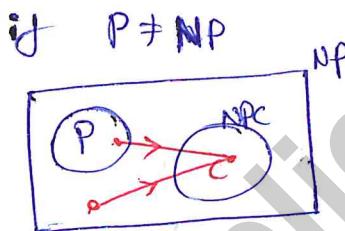
$P_1 \& P_2$: TM, Graphs, arrays, sets, strings

If P_2 is in P then \exists DTM to solve P_2

$$n^2 \cdot n^3 = n^5$$

$$\left. \begin{array}{l} \text{poly} \cdot \text{poly} = \text{poly} \\ \text{poly} + \text{poly} = \text{poly} \end{array} \right\}$$

NP-Complete: ($\text{NPC} / \text{NP-C}$)

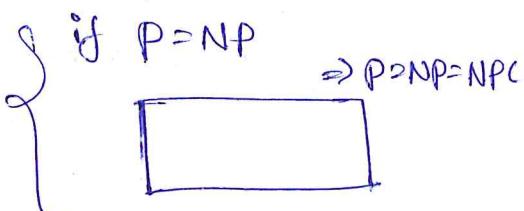
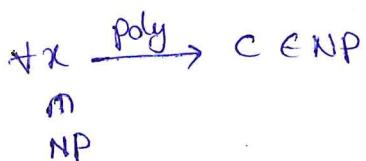


P and NPC are disjoint sets

A problem C is in NPC if

i) $C \in \text{NP}$

ii) Every problem in NP is reducible to C in poly-time.



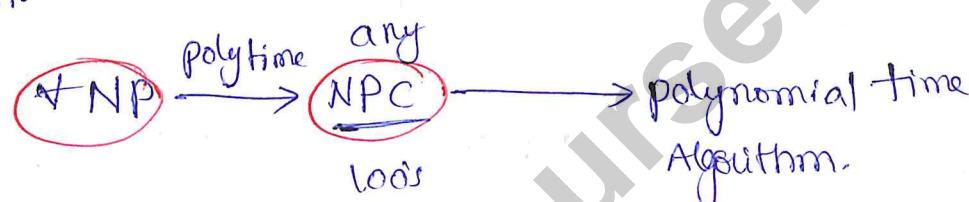
NPC: Today: We mostly believe $P \neq NP$

↳ we don't have a poly-time deti-Algorithm to any problem in NPC

↳ we do have exp time algorithm (2^n)

$O(n^n)$ $O(n!)$

If we show one problem in NPC to have a poly-time algorithm.



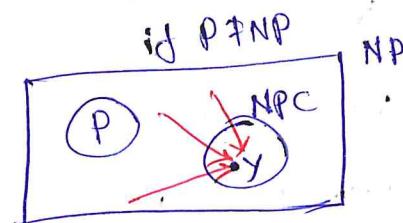
Then $NP = P$

How to prove a problem is NPC:

is $x \in NPC?$ $\xleftarrow{x \in NP}$

take $y \in NPC$

$y \xrightarrow{\text{poly}} x$



$\xleftarrow{NP} \xrightarrow{\text{poly}} y \xrightarrow{\text{poly}} x$

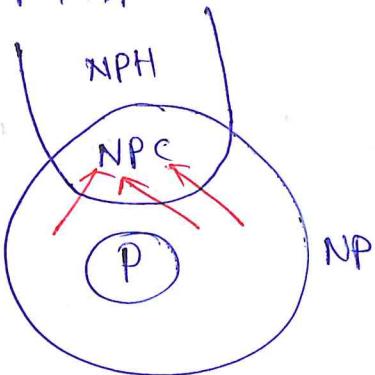
$\xleftarrow{NP} \xrightarrow{\text{Poly+poly}} x$

NP-Hard:

{ x is in NPH if

{ \exists a polytime reduction from every problem in NP

x need not belongs to NP.



NPC \subset NPH

NPH & NP \Rightarrow NPC

if $P = NP$

NPH X

$P=NP=NP\bar{C}$

$P=NP=NP\bar{C}$

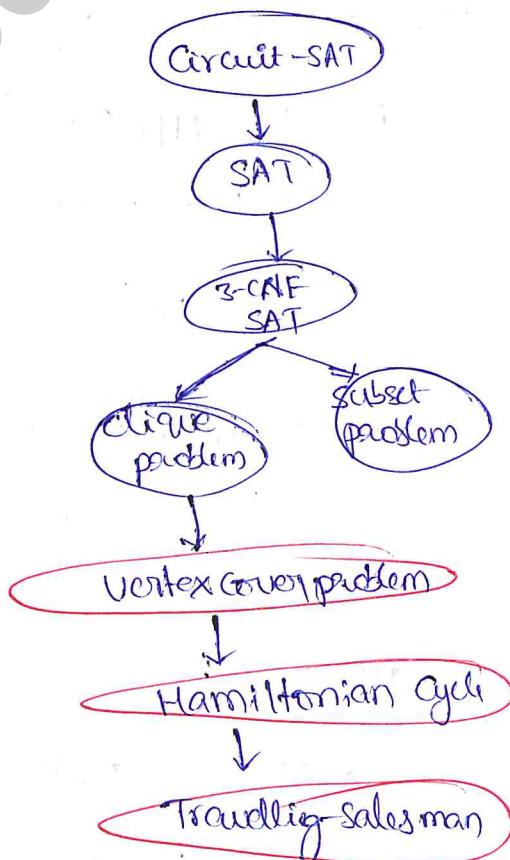
Hard \Rightarrow Non-polynomial time.

List of NP-Complete problems:

P: $O(n)$, $O(1)$, $O(\log n)$, $O(n \log n)$

$O(n^2)$, $O(n^2 \log n)$, $O(n^3)$

- ① 1-planarity
- ② Graph coloring
- ③ Hamiltonian
- ④ Vertex Cover
- ⑤ Knapsack problem
- ⑥ Subset Sum problem
- ⑦ Closest string
- ⑧ Longest common
- ⑨ Boolean sat problem



- ① Halting problem \rightarrow NPH, but not NP Complete
- ② Subset sum problem
- ③ Travelling Salesman problem

[NPC \subset NPH]

— o —