CS & IT ENGINEERING

THEORY OF COMPUTATION

Turing Machine

Lecture No.- 01

By- Venkat sir

Recap of Previous Lecture

Topic

?? CFG

PDA

CFL → Detection

CFL → Closure properties

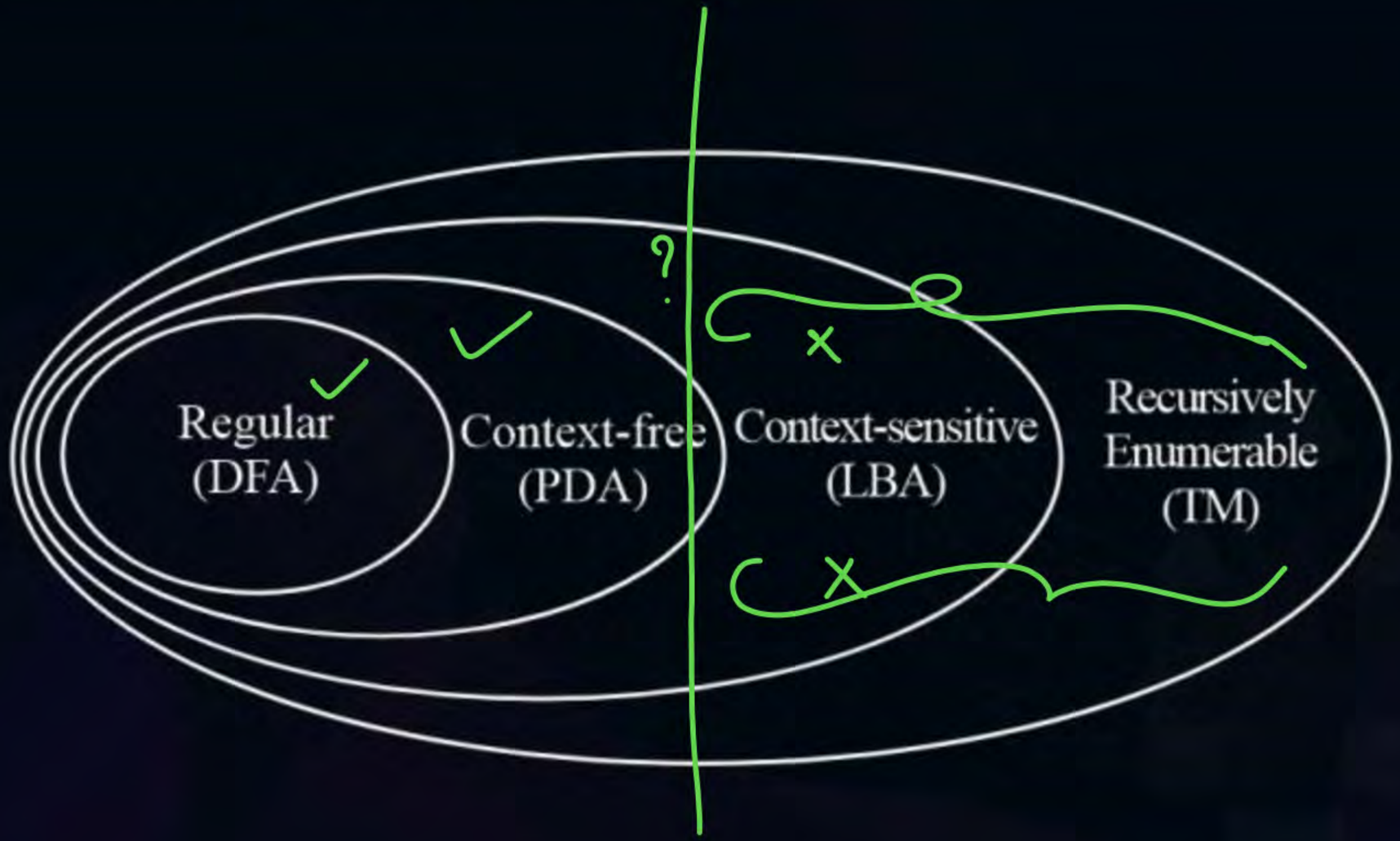
Topics to be Covered

Topic Turing Machine

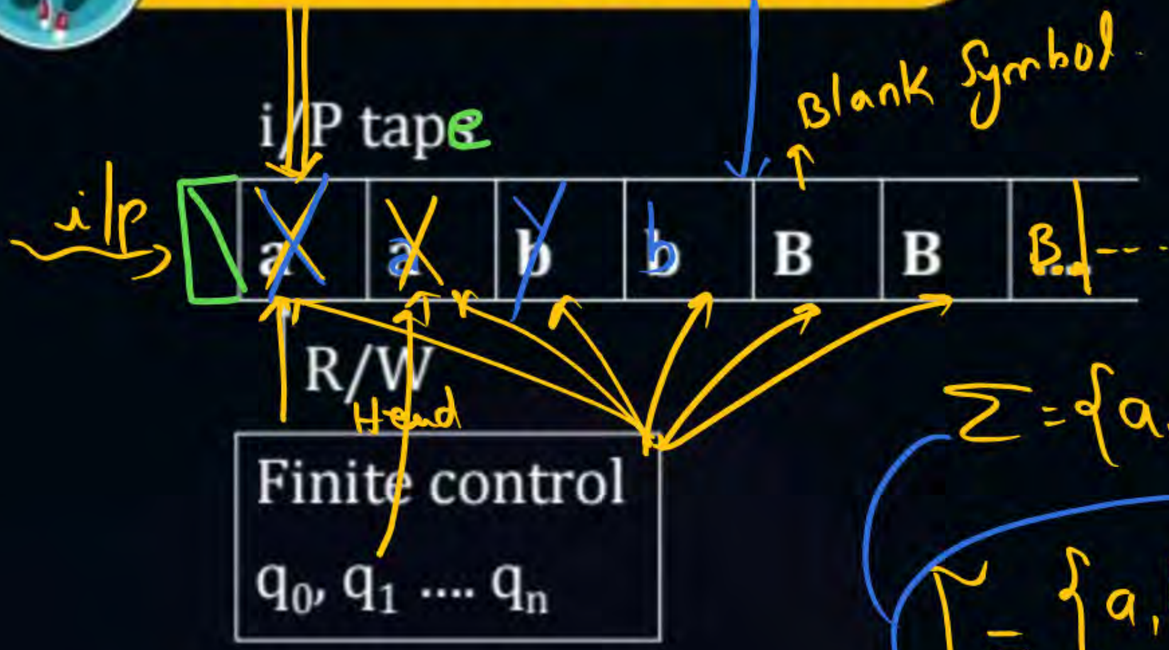
Topic ?? T·M Construction

Topic ?? Types of T·M.

Topic : Theory of Computation



Topic : Turing Machine



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

$\Gamma = \{a, b, x, y, B\}$

Recursive Enumerable Language

1. Infinite length tape
2. Turn around capability
3. Read/write capability

Decidable Problems :- Algorithm exist

Undecidable Problems : No Algorithm not exist

P class : $O(n^k)$

Np class : $O(2^n)$

Np Complete

Np hard $> O(2^n)$

Topic : Turing Machine

→ Turing machine is a mathematical model that represents general purpose computer.

→ The problem, not solved by Turing machine or not soluble by computer also.

→ Hence Turing machine are used to study power of a ~~computer~~ Computer.

NOTE:

Computer to finite automata, PDA, Turing having additional property they are

1. **Infinite Length tape:** Turing machine is one side closed and one side infinite.
2. **Turnaround capability:** Turing machine (Capable) to turn left as well as right side. (direction)
3. **Read-Write capability:** Turing machine can replace reading symbol by other (or) same symbol.

Turing Machine = $(Q, \Sigma, q_0, F, B, \Gamma, S)$

Q : Finite number of state

Σ : I/o alphabet

q_0 : Initial state → only one

F : Set of final states → any

B : Blank symbol

Γ : Tape alphabet

S : Transition function.

Q	×	Γ	→	Q	×	Γ	×{L,R}

S: transition function

$$Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

$(q_1, X) = (q_2, Y, L)$

$(q_1, a) = (q_2, X, R)$

Acceptance by final state only

Topic : Turing Machine

$$|Q| \times |\tau| \rightarrow |Q| \times |\tau| \times \{L, R\}$$

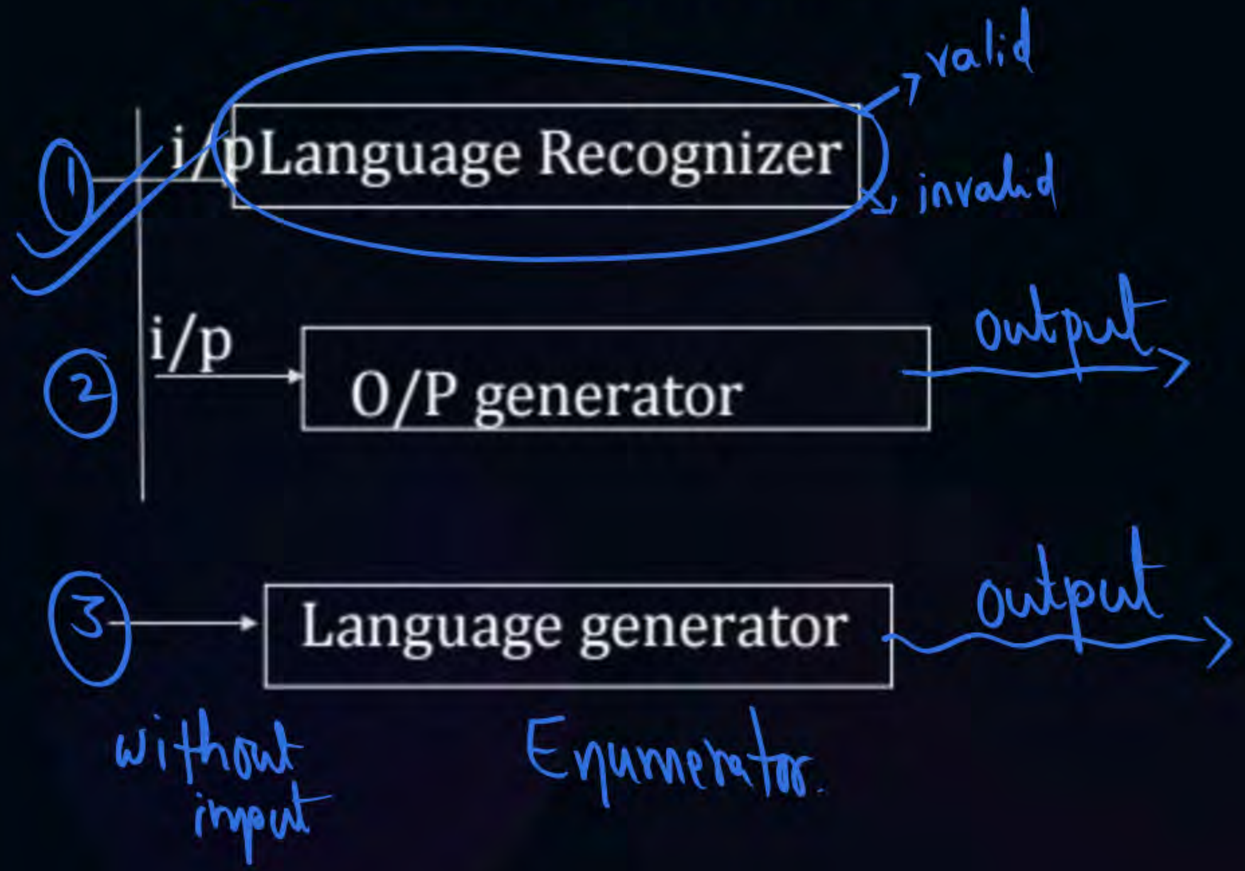
Notaulus :

⇒ Transition diagram

⇒ Transition Table

Topic : Turing Machine

Type of Turing Machine



Topic : Turing Machine

Notations

- Transition diagram
- Transition table

Type of Turing Machine

- i/p → Language Recognizer
- i/p → O/P generator
- Language generator

Topic : Turing Machine

Acceptance by final state.

Turing machine as a language recognizer-

→ By reading the string Turing machine may halt may not halt (go to infinite loop)

→ By reading string 'X' Turing machine halts in final state then X is accepted.

→ By reading string 'X' Turing machine halts non-final state then string is ~~rejected~~ rejected.

→ By reading string 'X' if Turing machine enters into infinite loop then don't knows about the i/p. (don't know accepted (or) rejected)

(We can not say anything about whether it is accepted or not.)

Construct a Turing machine

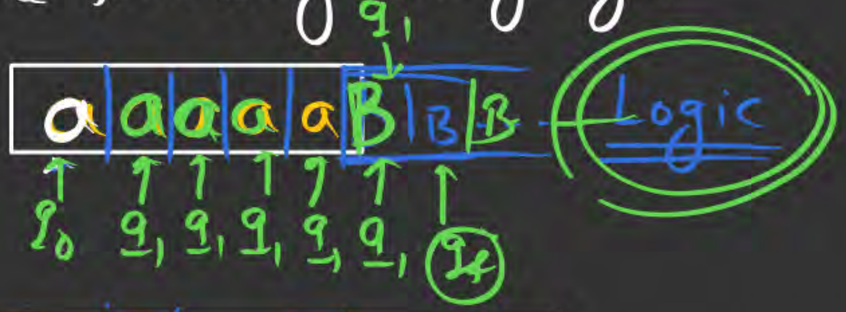
$L = \{a^n / n \geq 1\}$

(Q) Construct Turing Machine for the following Language

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

Diagram (∈) | B | B

T·M



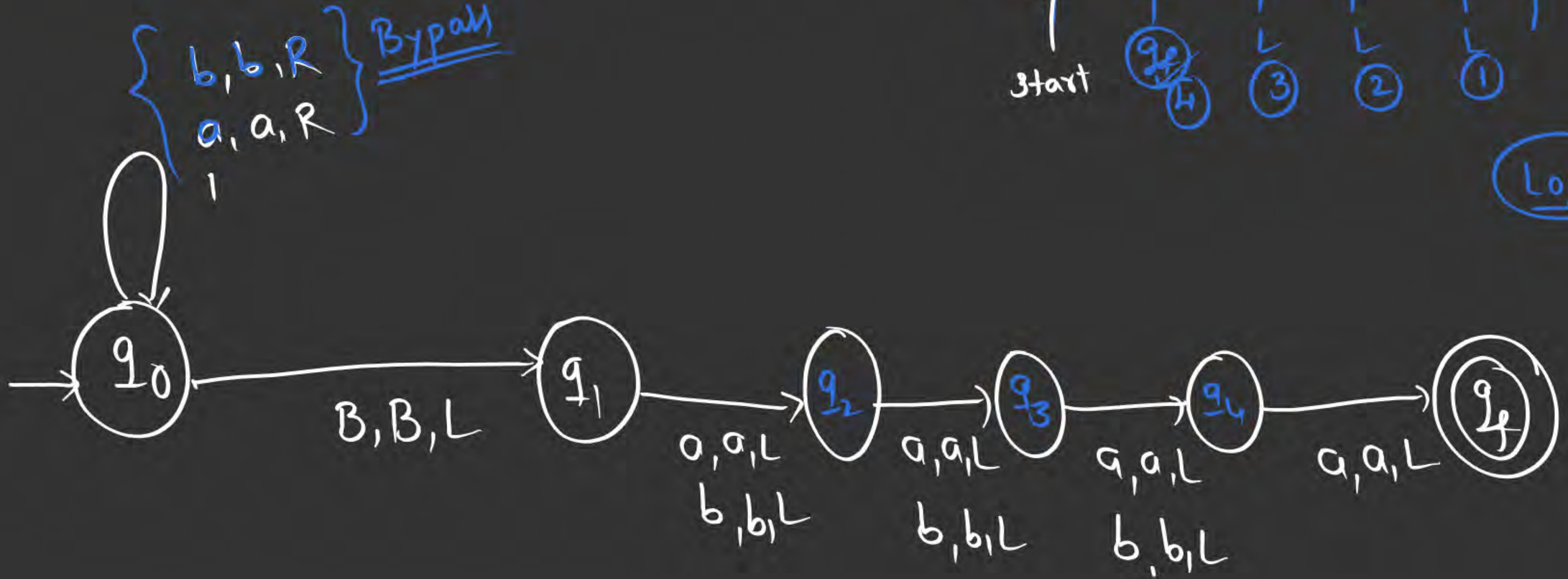
Table

	a	B
→ q_0	(q_1, a, R)	
q_1	(q_1, a, R)	(q_f, B, R)
q_f		

(Q) Construct T.M for $(a+b)^* a (a+b)(a+b)(a+b)$. [Deterministic]

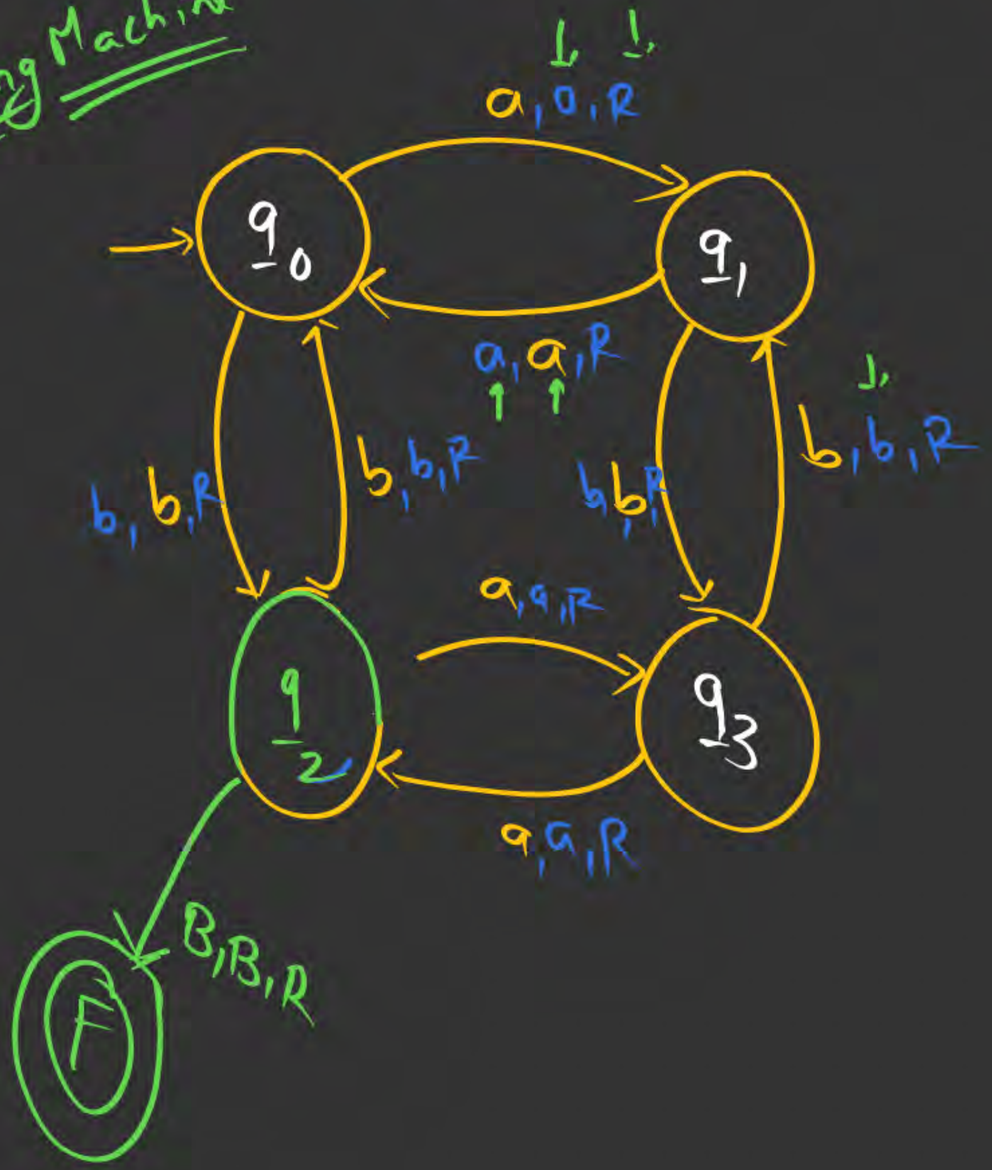
$(a+b)^*$	a	(a+b)	(a+b)	(a+b	B	B	B
	↑	↑	↑	↑	↑	↑	
Start	q_4	L	L	L			
	4	3	2	1			

Logic



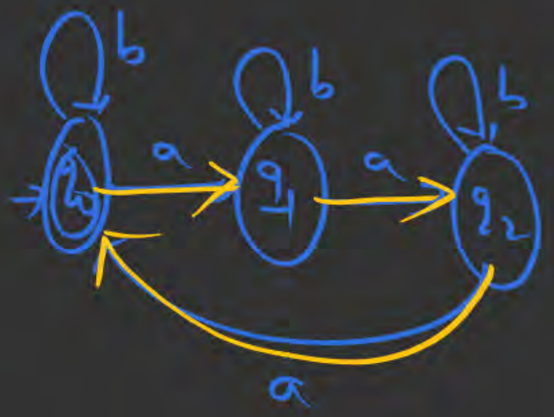
(Q) Construct T·M for # a's even and # b's odd.

T·M Turing Machine



(Q) Identify language accepted by following Turing Machine.

TM



Regular

	a	b	B
→ q_0	(q_1, a, R)	(q_0, b, R)	(q_f, B, R)
q_1	(q_2, a, R)	(q_1, b, R)	
q_2	(q_0, a, R)	(q_2, b, R)	
q_f	HALT		

(a) length exactly 3

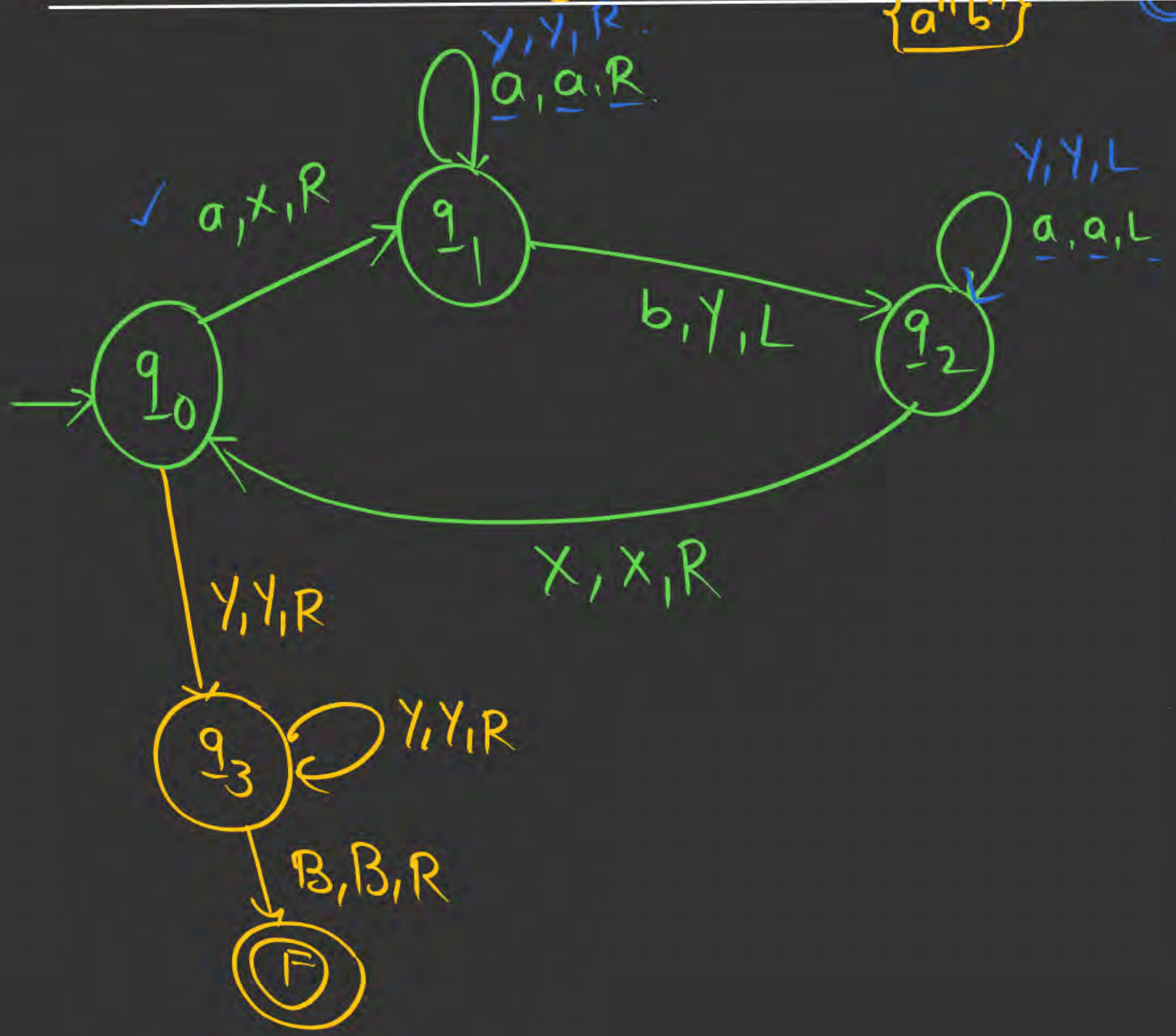
(b) length div by 3

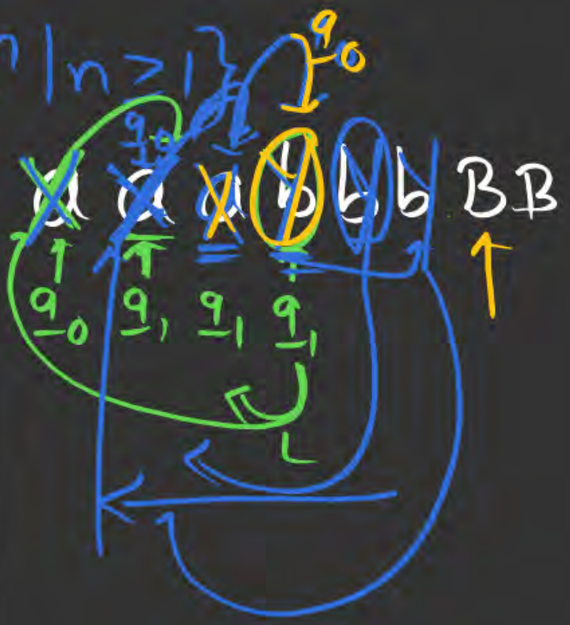
(c) a's div by 3

(d) b's div by 3

(Q) Construct Turing Machine for $L = \{a^n b^n \mid n \geq 1\}$

$\{a^n b^n\}$





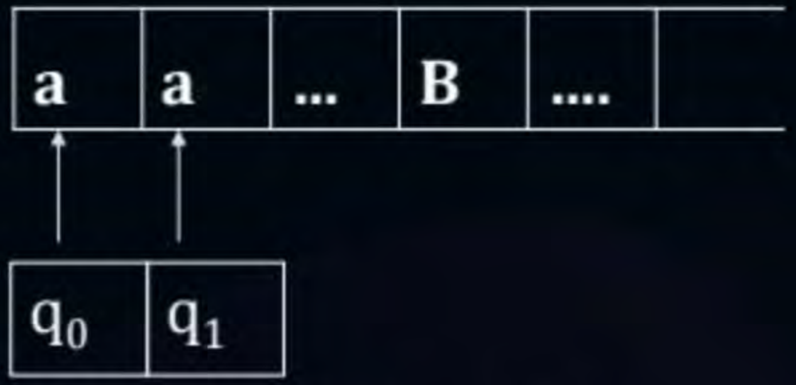
(Q) Construct T.M for $L = \{a^n b^n c^n / n \geq 1\}$

Home Work


A A A A B B B B B B

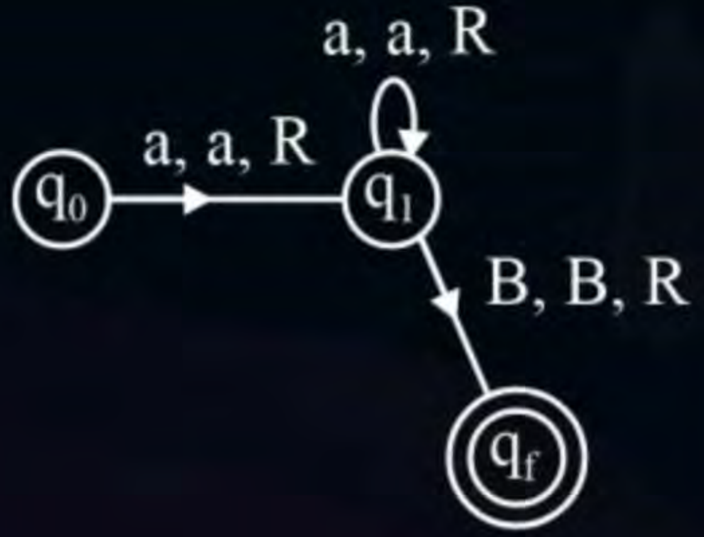
Topic : Turing Machine

{a, aa, aaa ..... }



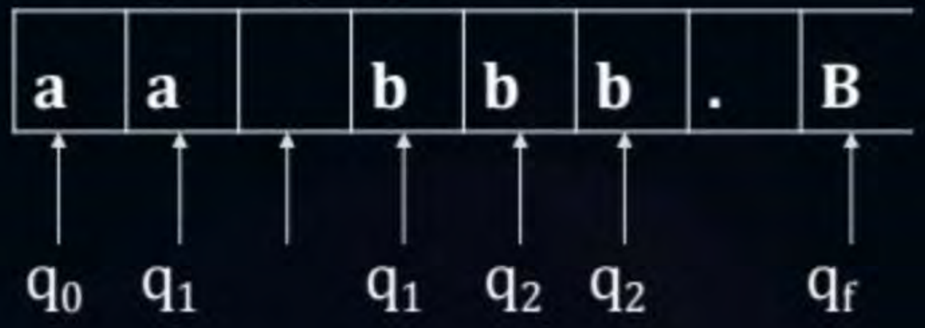
S: $\theta \times \Gamma \rightarrow \theta \times \Gamma \times (L, R)$

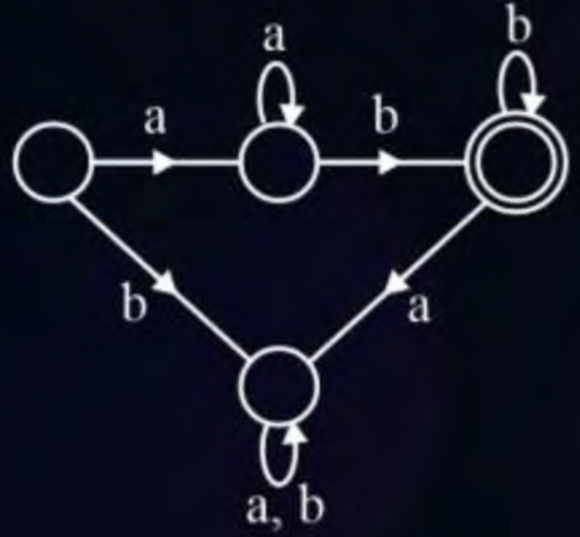
State	a	B
$\rightarrow q_0$	(q, a, R)	B
q_1	(q, a, R)	(q_f, B, R)
q_f	(HALT)	T



Topic : Turing Machine

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





[MCQ]

#Q. Which of the following statements is false?

A. The halting problem for Turing machine is undecidable

B. Determining whether a context free grammar is ambiguous is undecidable

C. Given two arbitrary context free grammars G_1 and G_2, it is undecidable whether $L(G_1) = L(G_2)$

D. Given two regular grammars G_1 and G_2, it is undecidable whether $L(G_1) = L(G_2)$

2 mins Summary

Topic One

Topic Two

Topic Three

Topic Four

Topic Five

THANK - YOU