CS & IT ENGINEERING

THEORY OF COMPUTATION

Turing Machine

Lecture No.- 02

By- Venkat sir

Recap of Previous Lecture

Topic → ?? Turing machine Construction

→ T.M for Regular Language

→ Non Regular Language T.M

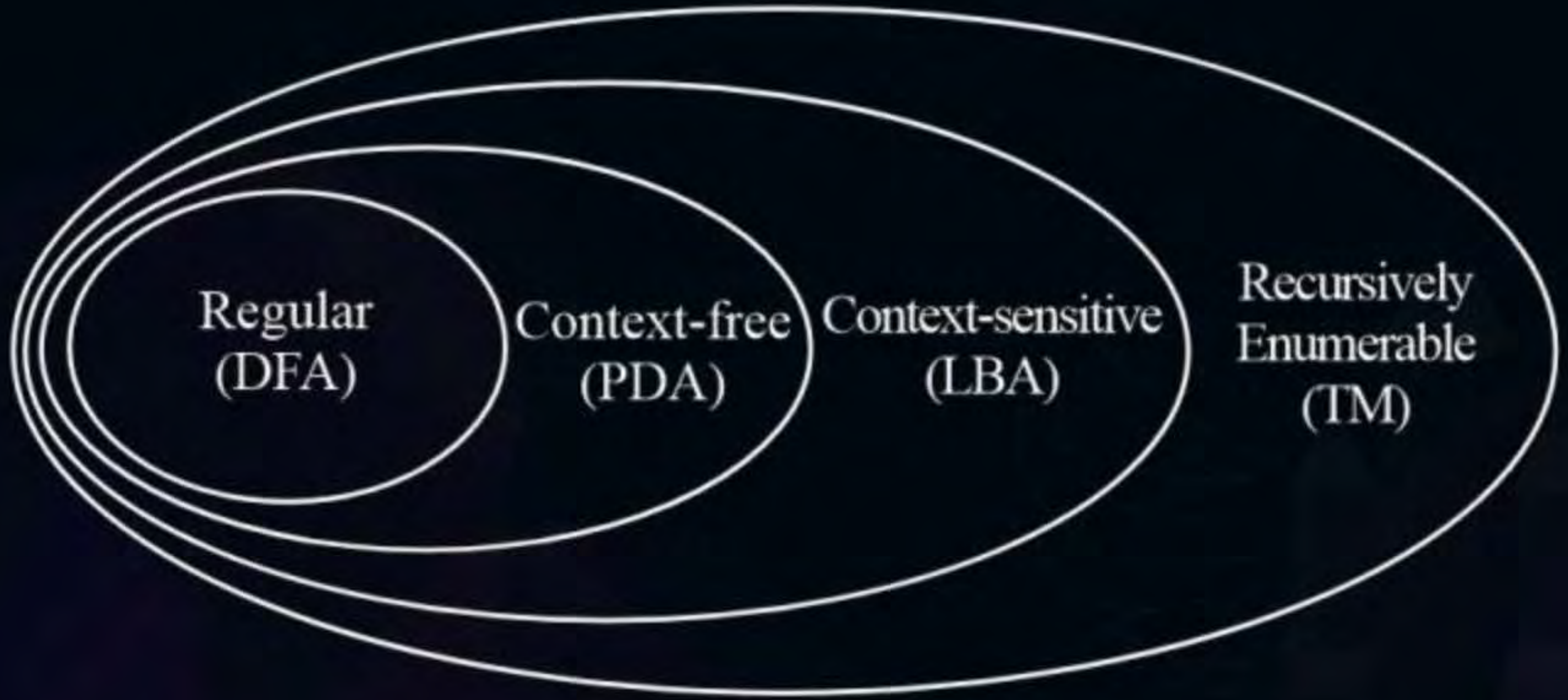
Topics to be Covered

Topic Turing Machine for Non Regular

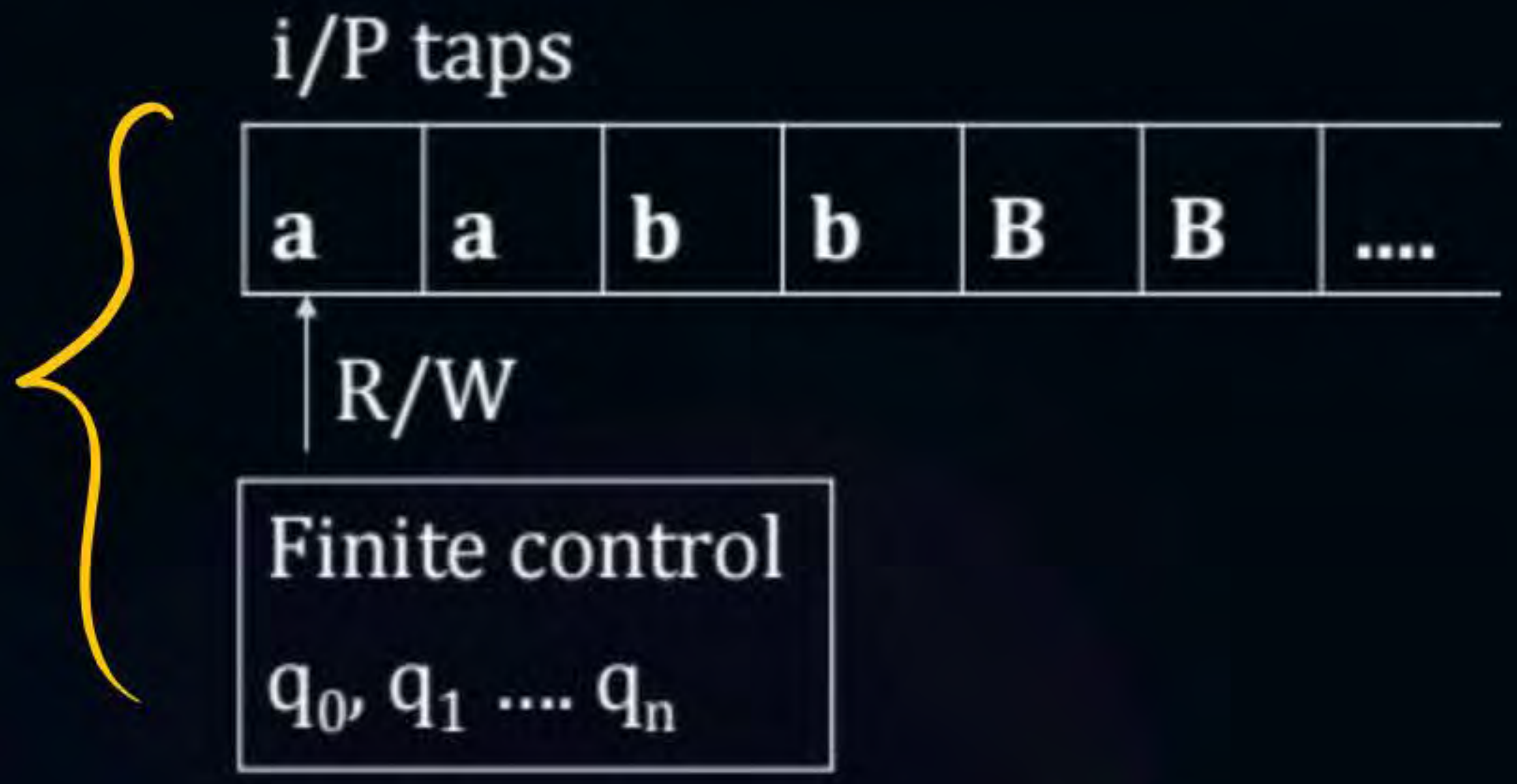
Topic ?? Recursive Enumerable Language

Topic ?? Recursive Language.

Topic : Theory of Computation



Topic : Turing Machine



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

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 compiler.

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 to turn left as well as right side.
3. **Read-Write capability:** Turing machine can replace reading symbol by other or same symbol.

Turing Machine = (Q, Σ, q_0, F, B, Γ, S)

Q : Finite number of state

Σ : I/o alphabet

q_0 : Initial state

F : Set of final states

B : Blank symbol

Γ : Tape alphabet

S : Transition function.

Q	**×**	**Γ**	**→**	**Q**	**×**	**Γ**	**×{L,R}**

Topic : Turing Machine

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

Notaulus :

$\Rightarrow$ Transition diagram

$\Rightarrow$ Transition Table

Type of TM

(ii) $\rightarrow$ Language Recognizer $\rightarrow$ yes / $\rightarrow$ no

i/p $\rightarrow$ O/P generator $\Rightarrow$ OP

$\rightarrow$ Language Generator $\Rightarrow$ Lans

Topic : Turing Machine

Type of Turing Machine

1 i/p $\rightarrow$ Language Recognizer

2 i/p $\rightarrow$ O/P generator

3 $\rightarrow$ Language generator

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

Turing machine as a language recognizer-

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

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

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

→ By reading string 'X' if Turing machine enters into infinite loop then don't knows about the i/p.

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

Construct a Turing machine

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

Topic : Turing Machine

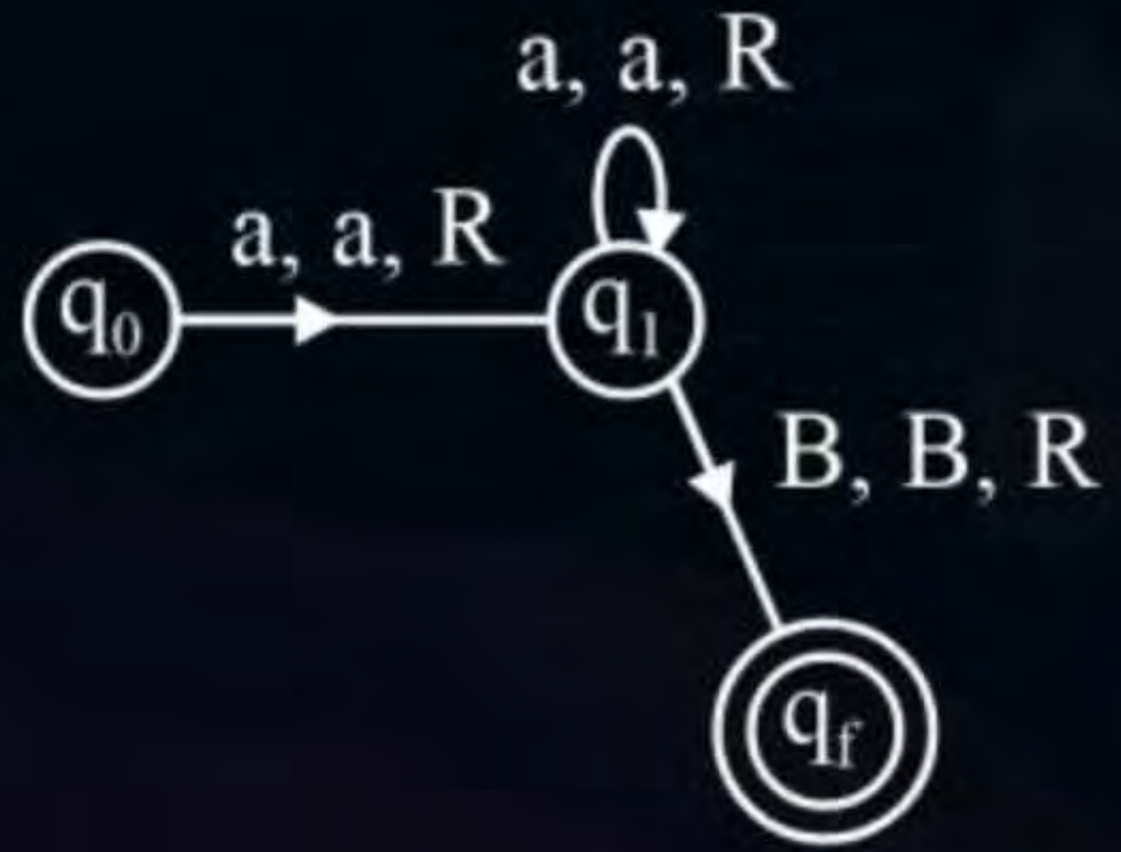
{a, aa, aaa ..... }

a	**a**	**...**	**B**	**....**	

q_0	q_1

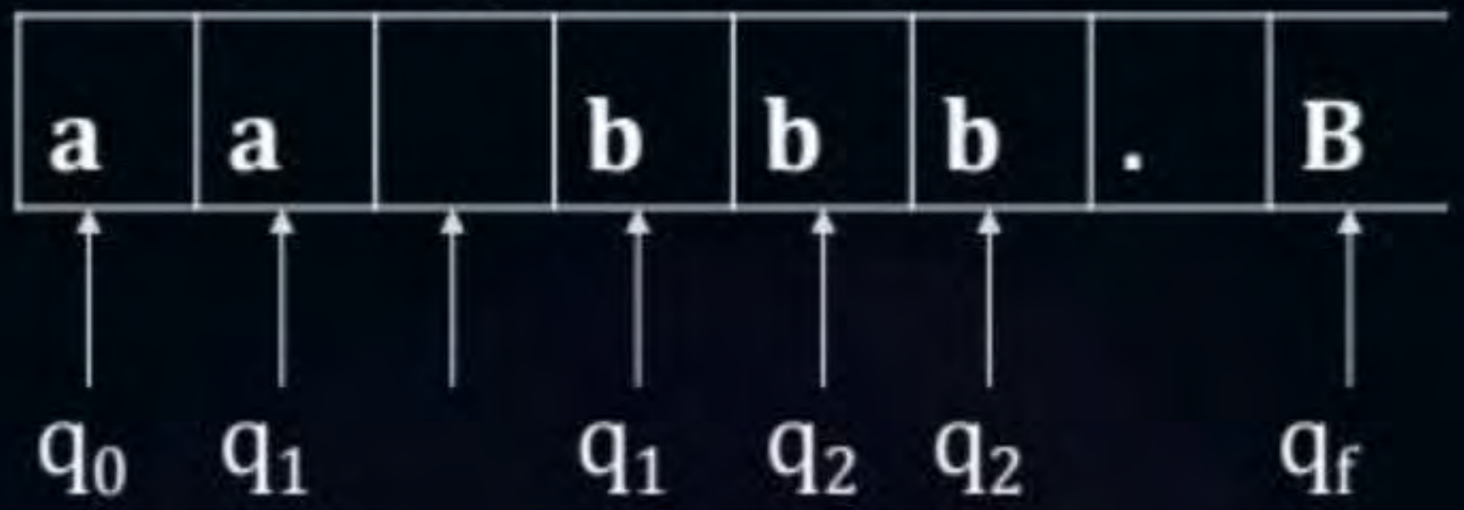
$S: \theta \times \Gamma \rightarrow \theta \times \Gamma \times (L, R)$

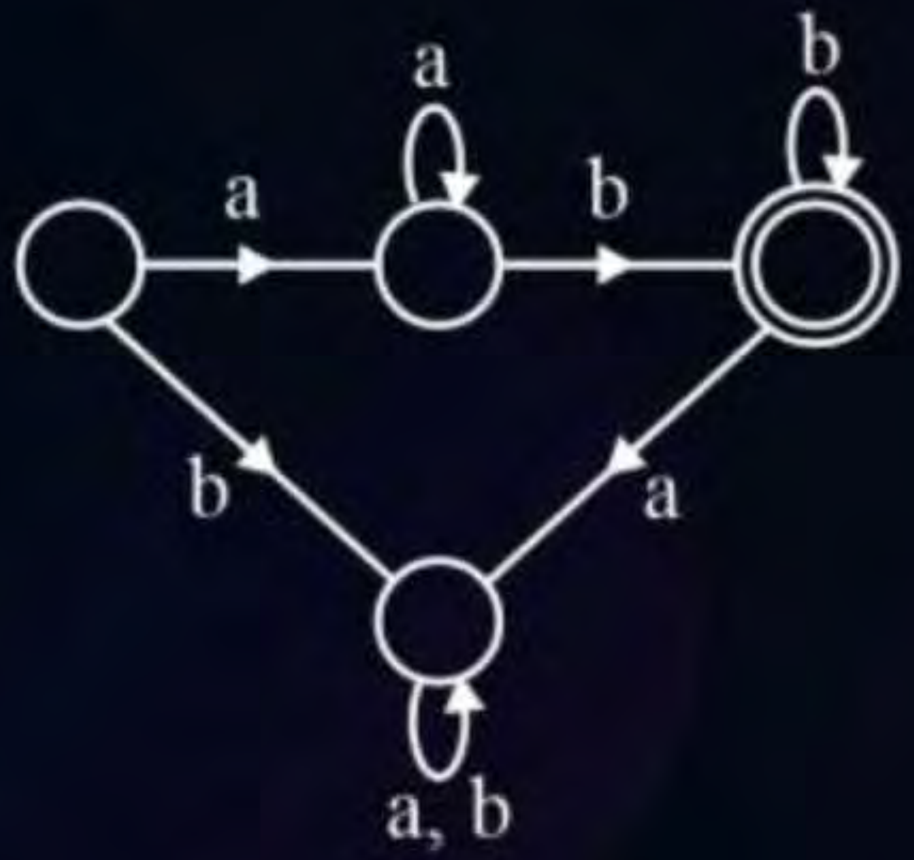
State	**a**	**B**
$\rightarrow q_0$	(q, a, R)	B
q_1	(q, a, R)	(q_f, B, R)
q_f	(HALT)	T



Topic : Turing Machine

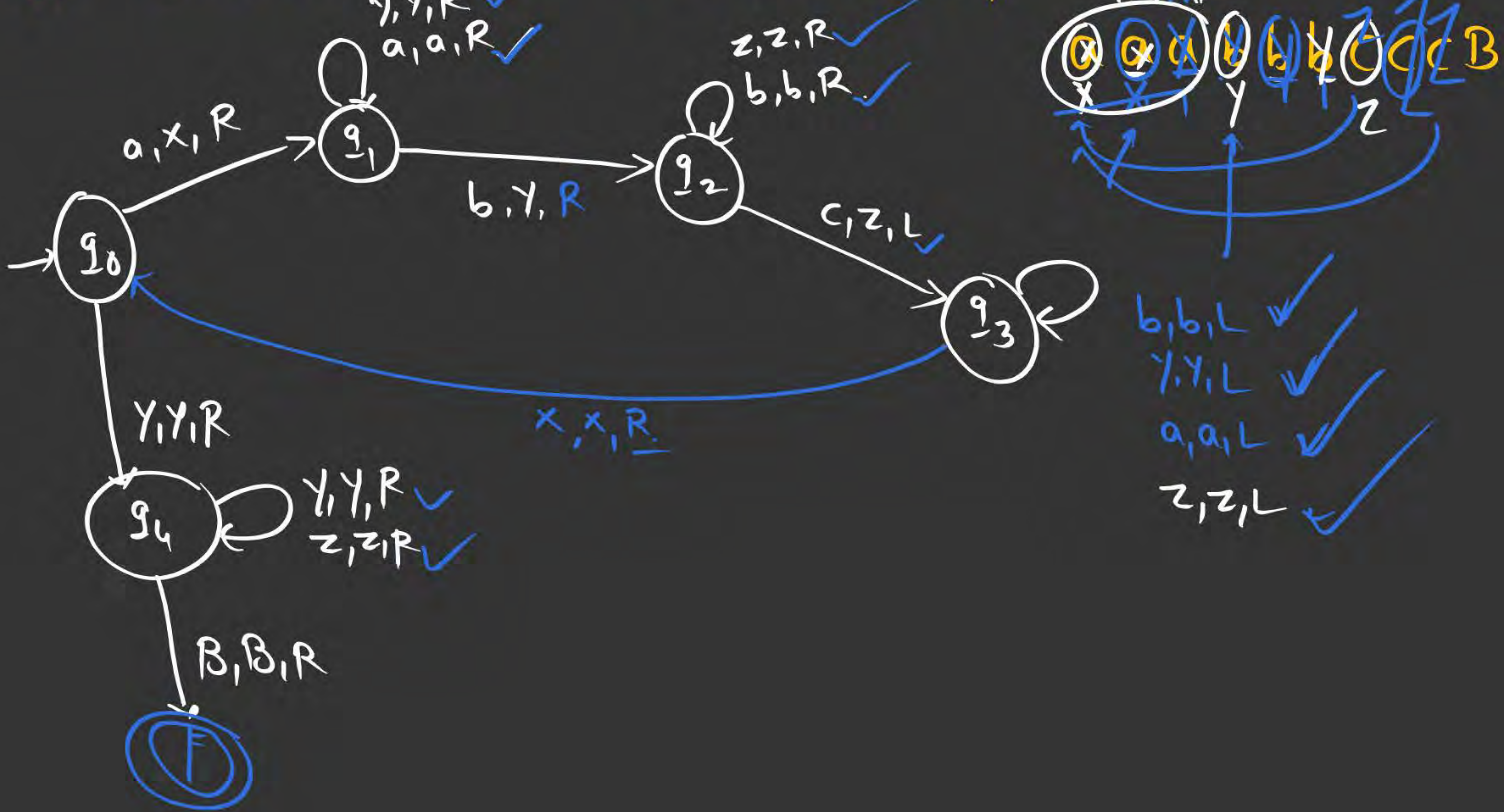
$L = \{a^n b^m / \ m, n \geq 1\}$





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

Logic:



(Q) Construct turing machine for

$L = \{ W C W \mid W \in (a+b)^* \}$ non CFL

Logic :- { a b a b c a b a b B B }

Home Work

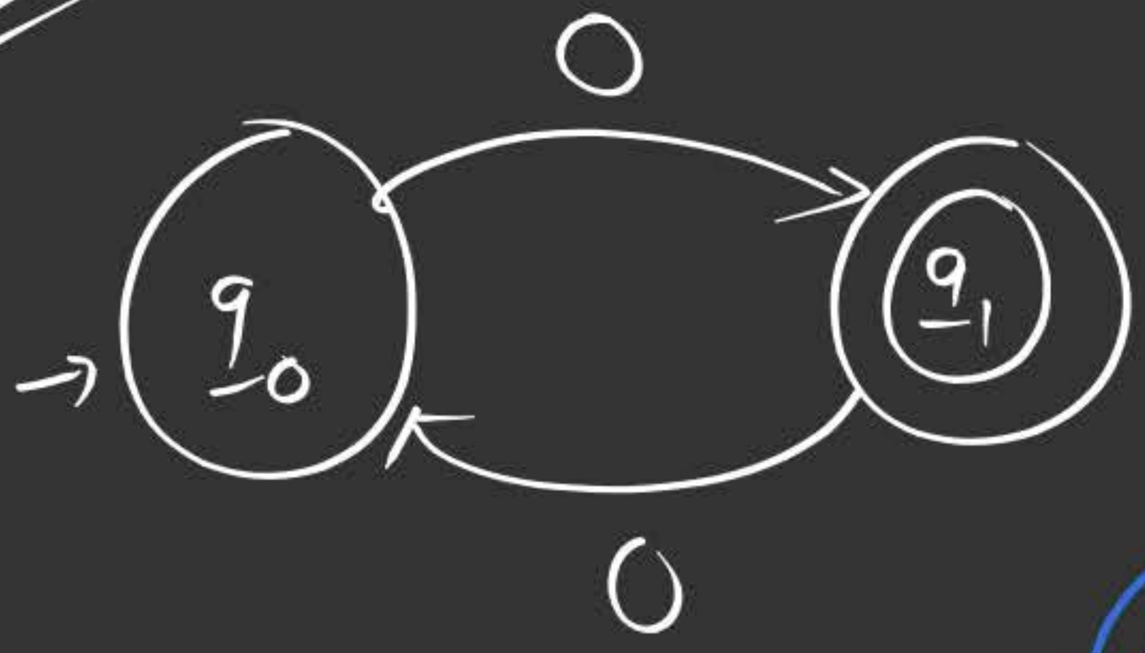
C, C, R

SKIP

Bypass

DFA

odd 0's



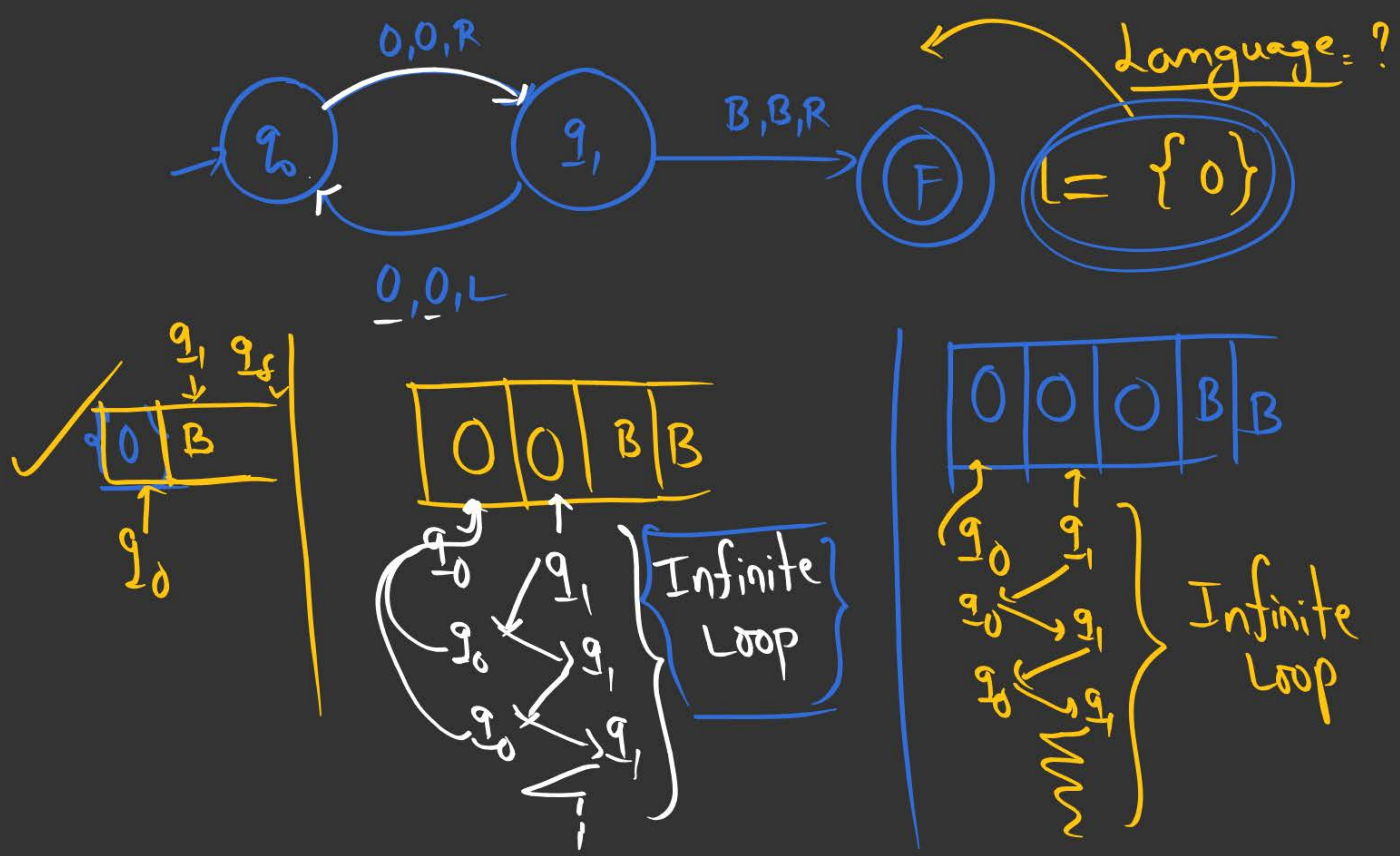
odd → Accept (Yes)

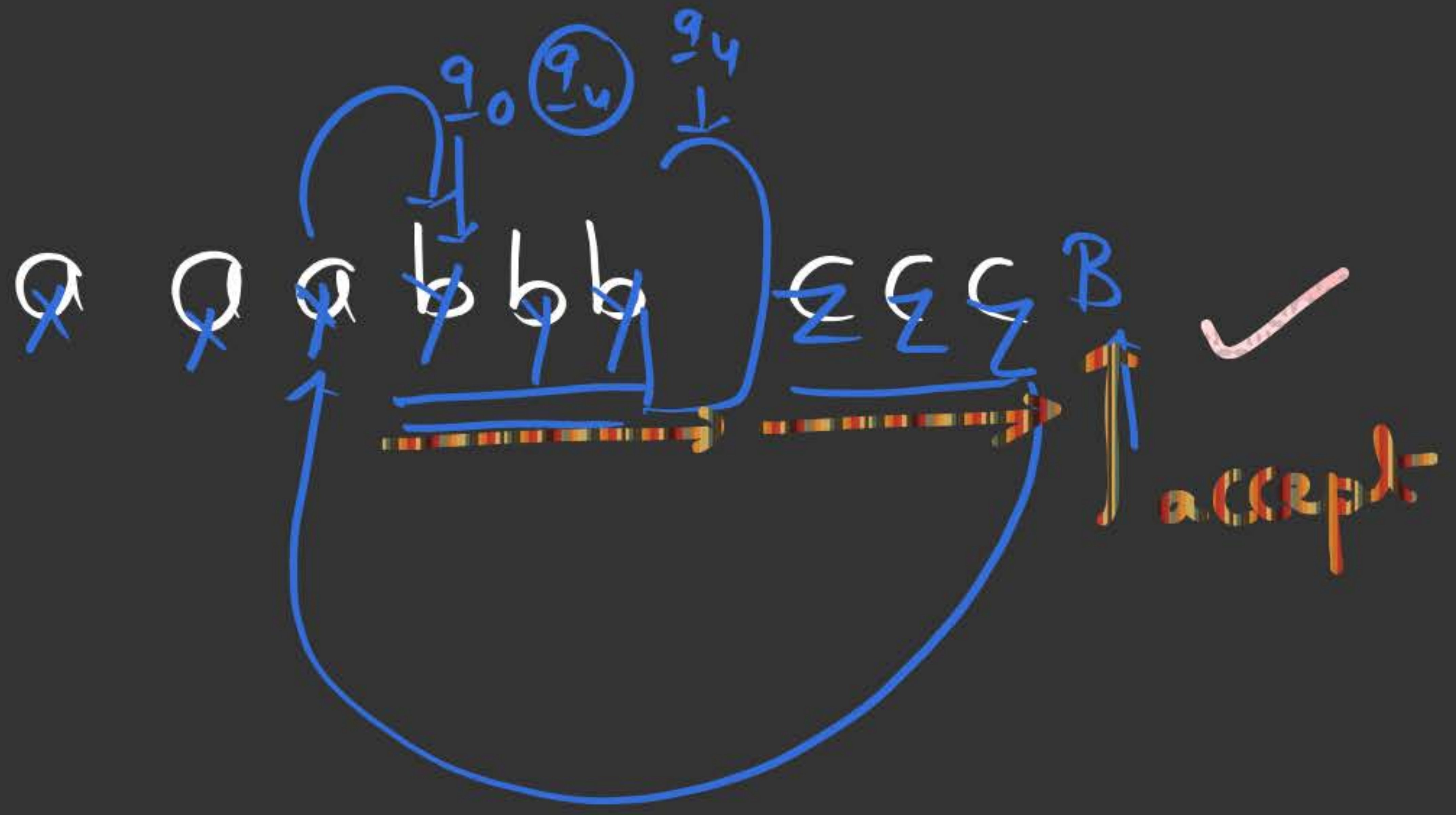
even → not accept (No)

odd

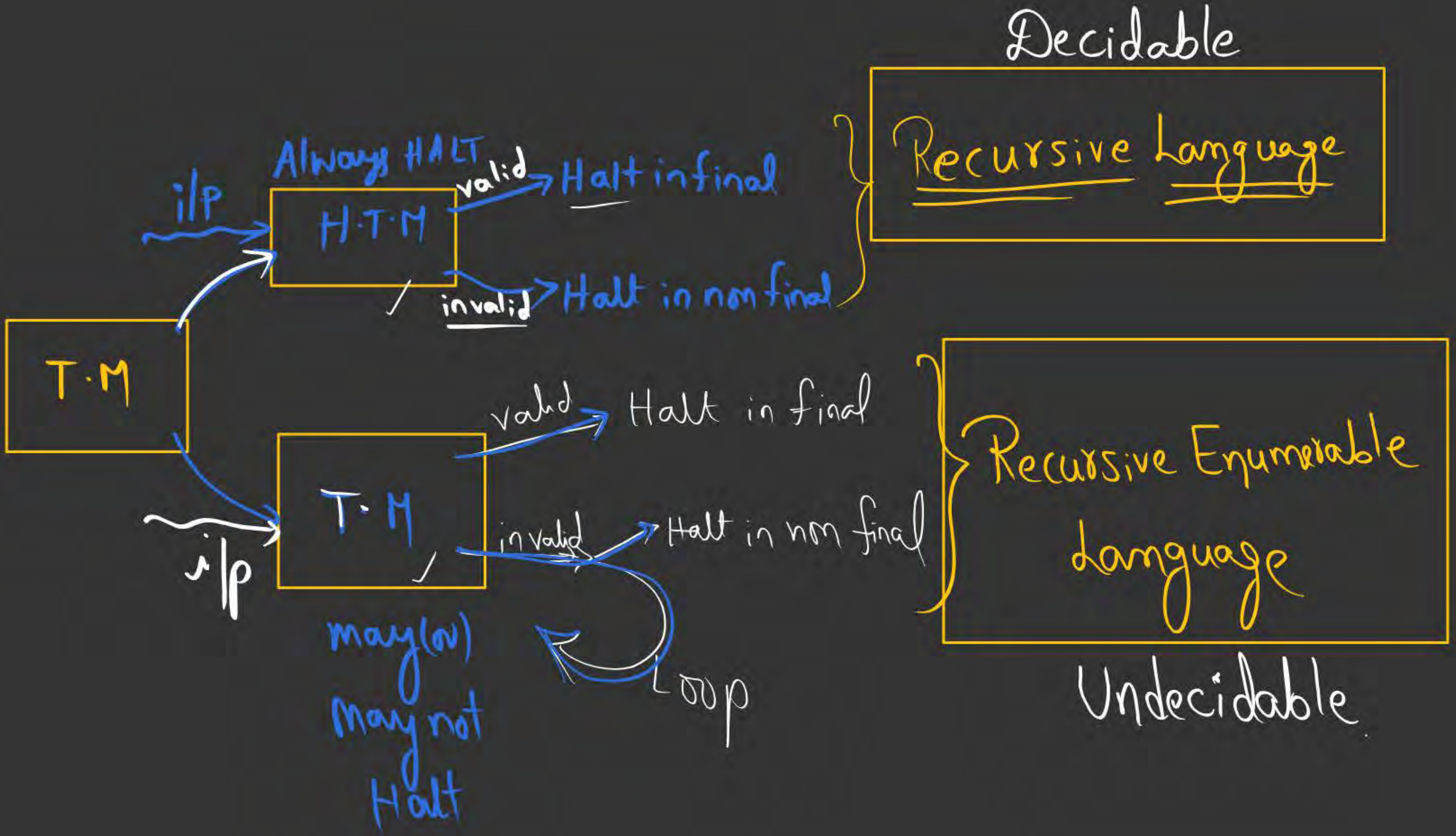
$\{0, 0^3, 0^5 \ldots\}$

$0^2, 0^4,$


0,0,R
B,B,R
0,0,L
q0
q1
F
Language = ?
L = {0}
0 B
0 0 B B
Infinite Loop
0 0 0 B B
Infinite Loop



q_4



Decidable
T·M
i/p
Always HALT
H·T·M
valid
Halt in final
invalid
Halt in non final
Recursive Language
T·M
i/p
may (or) may not Halt
valid
Halt in final
invalid
Halt in non final
Loop
Recursive Enumerable language
Undecidable

Recursive Enumerable (RE) or Type-0 Language

(Undecidable)

RE languages or type-0 languages are generated by type-0 grammars.

An REL language can be accepted or recognized by Turing machine which means it will enter into final state for the strings of language and may or may not enter into rejecting state for the strings which are not part of the language.

It means TM can loop forever for the strings which are not a part of the language. REL languages are also called as Turing recognizable languages.

H.T.M.

$\boxed{\text{(Decidable}}$ → Yes
→ NO

- **Recursive Language (REC)**

- A recursive language (subset of RE) can be decided by Turing machine which means it will enter into final state for the strings of language and rejecting state for the strings which are not part of the Language.

- e.g.; L= {a^nb^nc^n]n>=1}

- is recursive because we can construct a turing machine which will move to final state if the string is of the form a b c else move to non-final state.

- So the TM will always halt in this case. REC Languages are also called as Turing decidable languages.

No H·T·M

Ambiguity Problem = { G_1, G_2, G_3 - - - }

Amb L = { G_1 | G_2 | G_3 | G_4 - - }

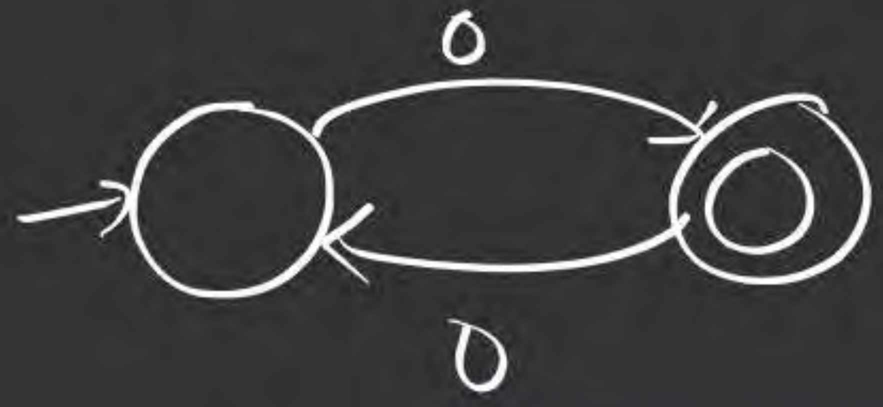
Yes } REL } Undecidable

NO } REL

G_1 : $\boxed{S \to SS \mid a}$ aaa T_1, T_2 } Ambiguous

G_2 : $S \to AA$
$A \to aA \mid a$ aaa

G_4 : $S \to aSa \mid bSb \mid c$



H·T·M

Membership Problem

Deciable for F·A

$L = \boxed{\{x_1, x_2, x_3, x_4, \ldots\}}$ → Yes / → NO } RECURSIVE

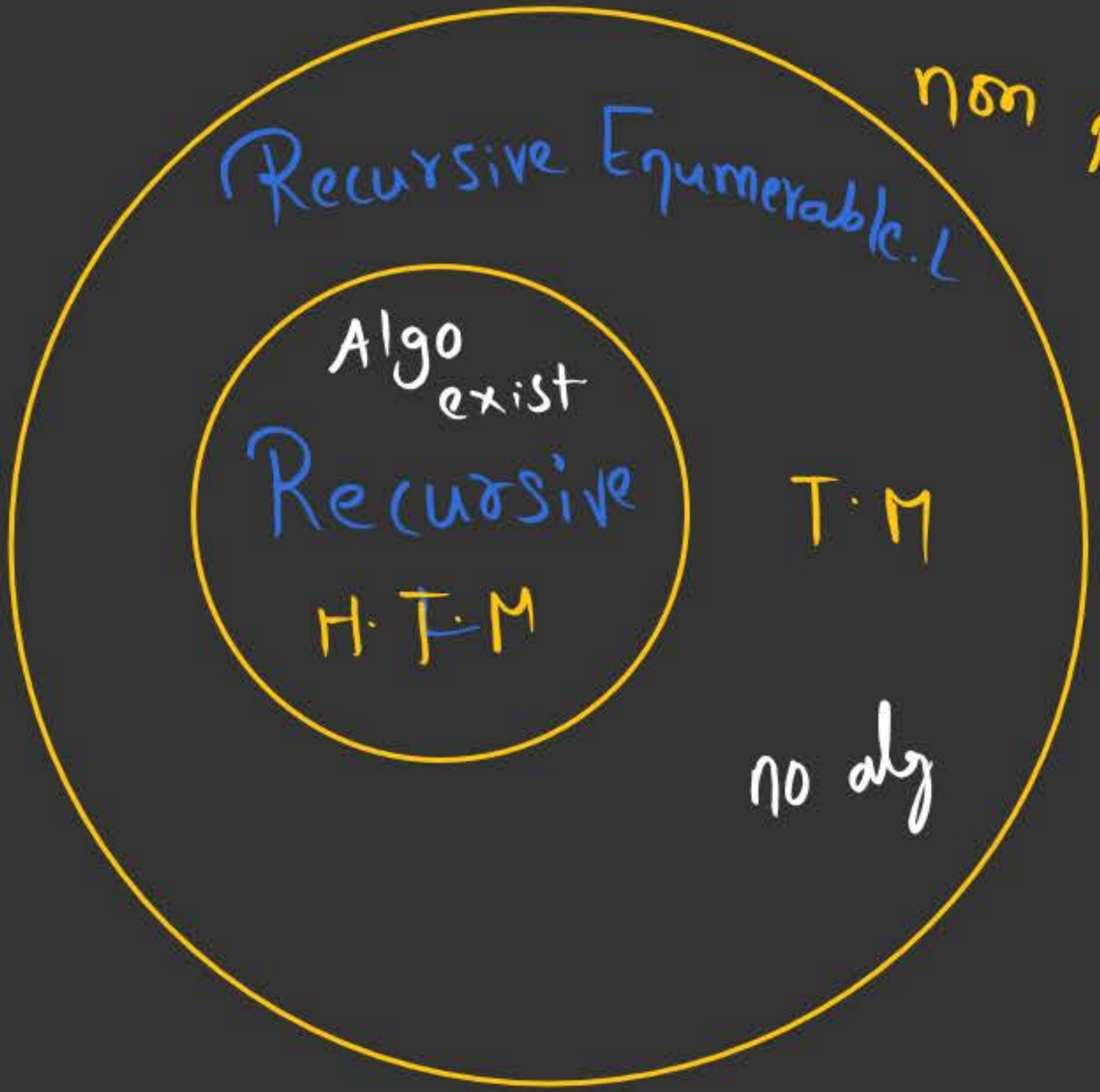
T·M

00

000

0000000

Every REC is REL but REL need not be REC.



non REL

no T·M

Computers can't solve these problems.

T.M → Yes, → NO } Recursive } Decidable

T.M → Yes, NO (loop) } Recursive Enumerable (Partially Decidable) Undecidable but

NO T.M } non REL } Undecidable

$$\left\{ \begin{matrix} \text{finite no. of} \\ \text{steps to solve} \\ \text{a problem} \end{matrix} \right\} \left\{ \text{Algorithm} = \text{H.T.M} \right\}$$

No H.T.M = No algorithm

Topic : Turing Machine

R.E.L

A language 'L' is said to be REL if there exist a Turing machine for that language, that Turing machine may halft on same i/p or may not halt on same i/p

→ I.e if the string is valid string of the languages then Turing Machine halts in final state and it says string is accepted.

→ If the string is not belongs to the language in the enter into infinite loop or halt in non final state

→ REL are called as Turing recognizable language

→ If any languages REL then it is undecidable

(number halting Turing machine exits)

NOTE:

All recursive language are R.E.L., but R.E.L. need not be recursive languages.

Hence recursive language are subclass of R.E.L.

→ By Default Turing Machine is may or may not halting Turing Machine.

→ By default Turing recognizable language are recursive enumerable language.

Chomsky Hierarchy

Topic : Turing Machine

Undecidable

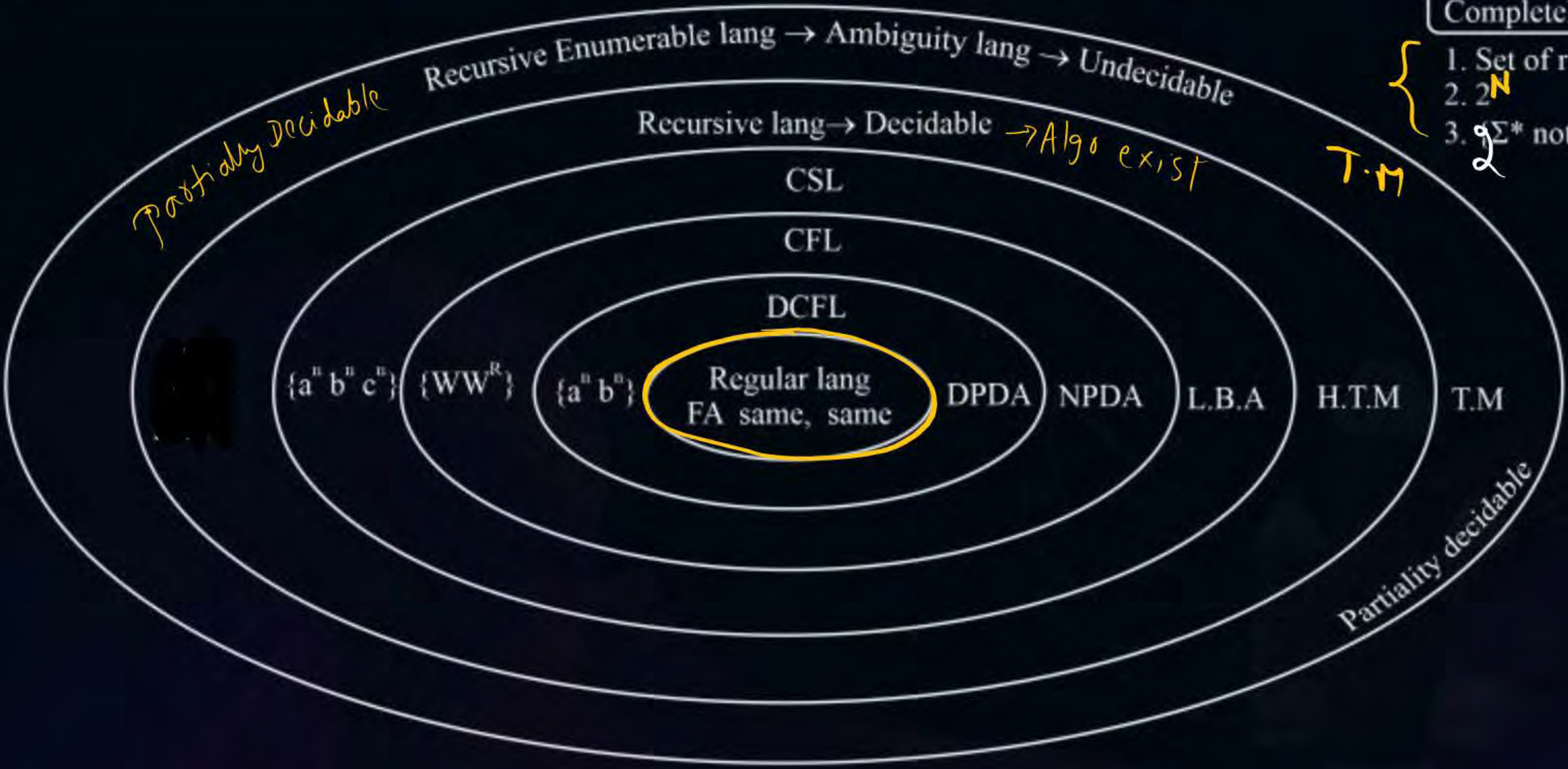
non REL

non R.EL

Completely un-decidable

1. Set of real number
2. 2^N
3. $\{\Sigma^*$ not R.E.L$\}$

No T.M



Topic : Turing Machine

$\Sigma^* = \{\epsilon, a, b, aa - - -\}$

Set

- Countable
  - Countable finite

    Ex:

    L= {a, b,c, d}
  - Countable Infinite

    Ex:

    (1) L= {1, 2,3,4 ---}

    (2) L = $\in$ k
- Uncountable

  NO T.M

  (1) Set of real numbers $\}$ NO T.M

Topic : Turing Machine

$N = \{1, 2, 3 - - -\}$

Countable Set :

~~Is~~ A set said to be countable of there exist 1 to 1 correspondence with nature number set to the given set.

Following are Countable set

→ All finite sets

→ Set of natural Number

→ Union of two countable sets

→ Product of two countable sets

→ Complete Lang.

→ Total population on the world.

T.M ✓

$N = \{1, 2, 3 \ldots\}$

0.000000 1

0.001

Ex :-Uncountable Set :

A Set said to be uncountable if there is no one to one correspondence with natural set to the given set.

$R = \{0.1, \quad 0.2$

$\Sigma^* = \{\epsilon, a, b, aa \ldots\}$

- → Set of real number.
- → Power set of natural set.
- → No of Points in a line
- → Set of all language over the given alphabet.

Points

→ 2^{Σ^*}

No TM

powerset(Σ^*)

2^{Σ^*} } Uncountable

There is no turning M/C exist for uncountable sets uncountable sets are not recursive enumerable.

→ Total No of uncountable sets is uncountable

→ Total No of language for which we can constructed is countable.

→ Total No of language for which we can construct finite Automata of TDA is countable

→ Not recursive enumerable problem are undecidable :

→ Recursive enumervioable long are undecidable (Partially Deciable.)

Topic : Turing Machine

i/P taps

a	a	b	b	B	B	--------

↑ R/W

Finite Control
q_0, q_1-------q_n

(i) Infinite length tape.

(ii) Turn around capacity.

(iii) Read write capability.

T. M = $(Q, \in, q0, f, B, \tau, S)$

Q :- finite no of state

$\in$:- i/p alphabet

q_0 :- initial state

f :- Set of final states

B:- Blank symbol

τ:- Tape alphabet.

S:- veansition function

Topic : Turing Machine

Closure Properties of Recursive Lang. & R .E Lang.

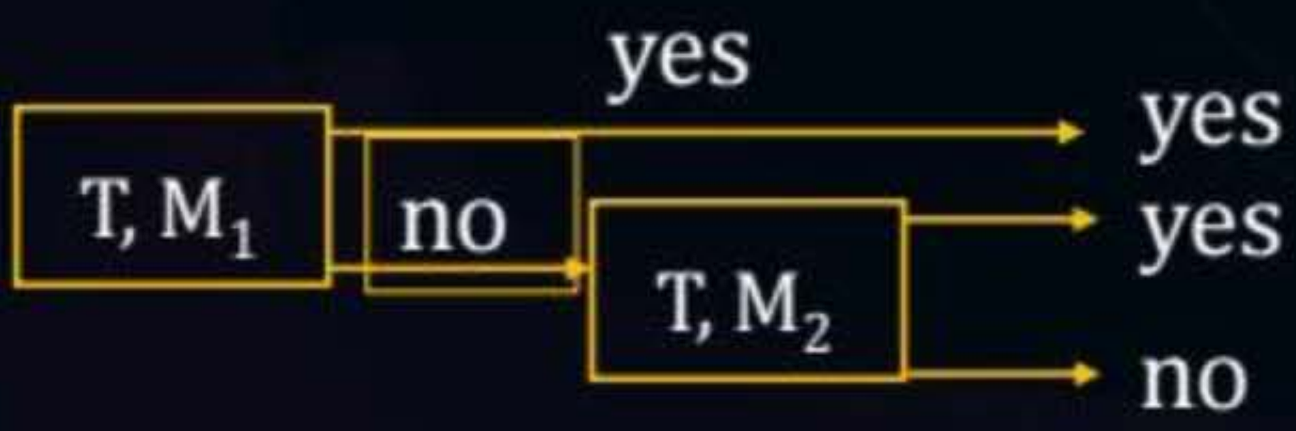
1. Union Operation :

The union of two Recursive lang is always Recursive Hence. Recursive lang are closed under union operation.

L_1 : Recu Lang →

L_2 : Recu Lang →

$L_1 \cup L_2$



Topic : Turing Machine

1. $S \rightarrow SS/a$ $\rightarrow$ useful $\rightarrow X_1$
2. $S \rightarrow SSS/b$ $\rightarrow$ useful $\rightarrow X_2$
3. $S \rightarrow aSb/as$ $\rightarrow$ empty lan $\rightarrow X_3$
4. $S \rightarrow AB/a$ $\rightarrow$ empty $\rightarrow \propto_1$

Empty less

Recursive lang
Decidable

$L = \{X_1, X_2, X_3, X_4 \ldots\ldots\}$

Topic : Recursive Language

A lang: L is set to be recursive if there existed truing M/C for that always halts for on all I/P strings.

i.e. if the string is valid string of a lang that the T.H. Halts in a final state and says string is accepted.

- If the string is not belong to the lang T.M. Halts in Non- final state and it says string is rejected.
- For any lang halting T.M. exits then it is "decidable".
- Halting TM is exactly to Algorithm.
- Hence, recursive lang also known as truing decidable language.

Topic : Turing Machine

Modifications of Turning machine :

The following are modified versions of T M .

1. Two Way infinite tape T.M → In this i/p tape is infinite in both direction.
2. Multitap Turing M/C :-

   In this turning M/C Multiple tape exist where each tape is infinite in both direction.
3. Non – Deterministic TM :-

   It is a T. M in which given Tape symbol / state finite No of choices exist for next to move.
4. Universal T.M :

   Universal TM simulates behaviors of other T. M by Taking them as I/P Hence universal I.P t can takes T.M , PDA, FA as C/P.

Topic : Turing Machine

Note :

After Modification, the Expressive power of T.M Remains same.
(computing speed may increases).

- DECIDABLE PROBLEM:::

A problem is set to be decidable if there exist halting I.M. solve the problem.

(or)

There exist Algorithm to solve this problem.

UNDECIDABLE PROBLEM::

- A problem is said to be undecidable if there is NO halting M/e (or) no turtling M/C for that problem (or) No Algorithm exist for that problem.
- To prove a problem 'X' is undecidable, we can use truing machine technique (or) reduction technique.

Topic : Undecidability

- A problem is set to be decidable if there exist halting I.M. solve the problem.
  (or)
  There exist Algorithm to solve this problem.
- A problem is said to be undecidable if there is halting M/e (or) no turtling M/C for that problem (or) No. Algorithm exist for that problem.
- To prove a problem 'X' is undecidable, we can use truing machine technique (or) reduction technique.

Topic : Undecidability

Reduction: A problem a is reducible to B. means we can canceled the problem B with the help of problem A.

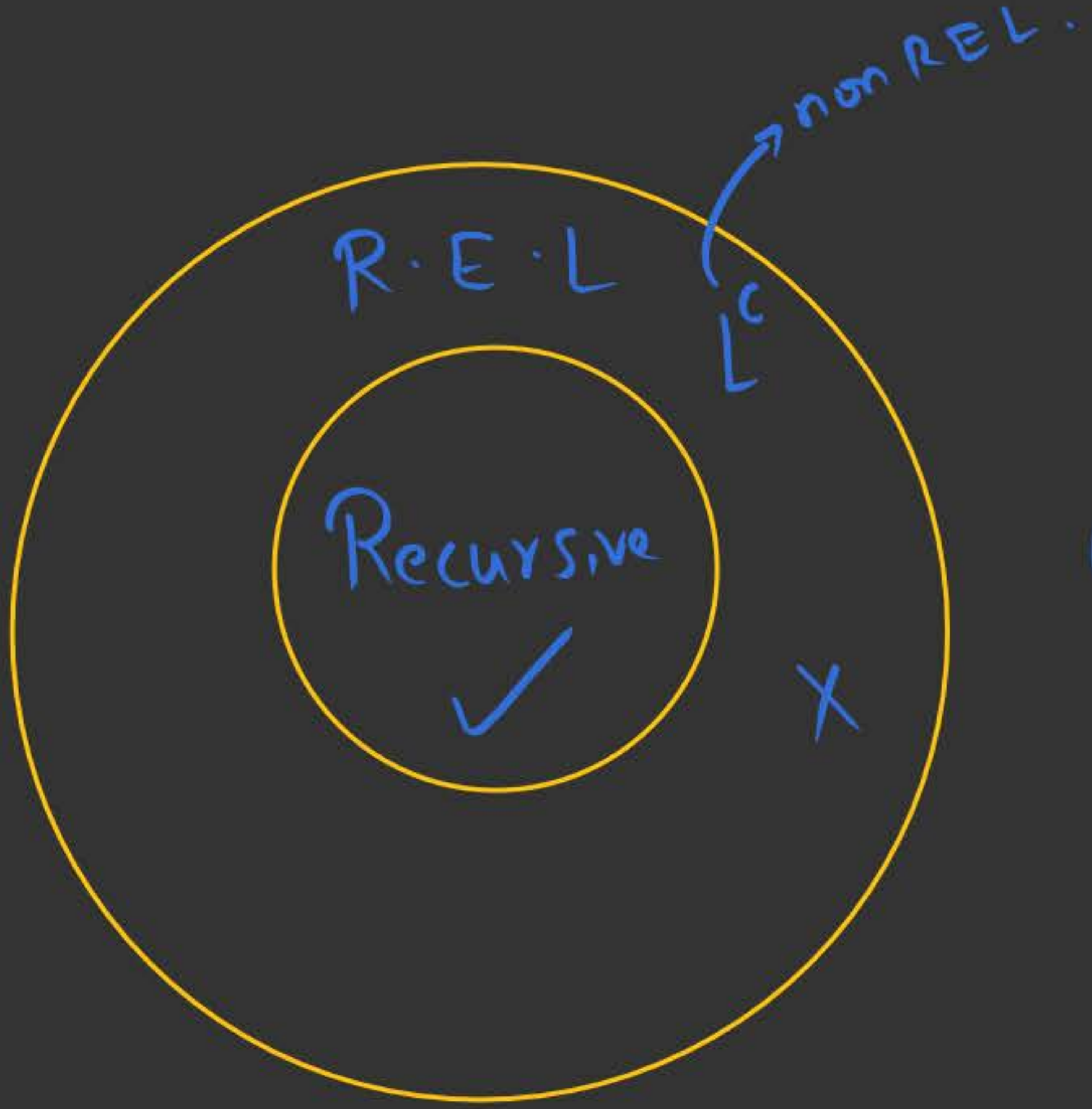
- Whenever A is rescuable to B, then B is as based as A.
- If A is reducible to B, the following of the possibility.

1. B is decidable then A is decidable.
2. If A is undecidable then B is also undecidable
3. If B is recursive lang then A is also recursive lang.
4. If B is REL then A is also REL.

closure Properties

10 am

	Regular	DCFL	CFL	CSL	Rec-Lang	REL
1. UNION	✓	X	✓	✓	✓	✓
2. Concatenation	✓	X	✓	✓	✓	✓
3. Intersection	✓	X	X	✓	✓	✓
4. Compliment	✓	✓	X	✓	✓	X ✓
5. Difference	✓	X	X	✓	✓	X ✓
6. $L \wedge$ Reg.	✓	✓	✓	✓	✓	✓
7. L – Reg.	✓	✓	✓	✓	✓	✓
8. Kleene closure	✓	X	✓	X	✓	✓
9. Positive closure	✓	X	✓	✓	✓	✓
10. Substitution	✓	X	✓	✓	X	✓
11. Homeomorphism	✓	X	✓	X	X	✓
12. I.H.M. (Inverse Homomorphism)	✓	✓	✓	✓	✓	✓
13. Reverse	✓	X	✓	✓	✓	✓



$\text{Recursive}^c \rightarrow \text{Recursive}.$

$(\text{REL but not Rec})^c \rightarrow \text{non REL}$

Recursive Enumerable (RE) or Type-0 Language

RE languages or type-0 languages are generated by type-0 grammars.

An RE language can be accepted or recognized by Turing machine which means it will enter into final state for the strings of language and may or may not enter into rejecting state for the strings which are not part of the language.

It means TM can loop forever for the strings which are not a part of the language. RE languages are also called as Turing recognizable languages.

- **Recursive Language (REC)**

- A recursive language (subset of RE) can be decided by Turing machine which means it will enter into final state for the strings of language and rejecting state for the strings which are not part of the Language.

- e.g.; L= {a^b^c^]n>=1}

- is recursive because we can construct a turing machine which will move to final state if the string is of the form a b c else move to non-final state.

- So the TM will always halt in this case. REC Languages are also called as Turing decidable languages.

[MCQ]

#Q. Context-free languages are

A closed under union

B closed under complementation

C closed under intersection

D closed under Kleene closure

[MCQ]

#Q. If L_1 and L_2 are context free languages and R a regular set, one of the languages below is not necessarily a context free language. Which one?

A L_1L_2

B $L_1 \cap L_2$

C $L_1 \cap R$

D $L_1 \cup L_2$

[MCQ]

#Q. Let R_1 and R_2 be regular sets defined over the alphabet then

A $R_1 \cap R_2$ is not regular

B $R_1 \cup R_2$ is not regular

C $\Sigma^* - R_1$ is regular

D $R_1{}^*$ is not regular

[MCQ]

#Q. If $L_1 = \{a^n \mid n \geq 0\}$ and $L_2 = \{b^n \mid n \geq 0\}$, consider

I. $L_1 \cdot L_2$ is a regular language

II. $L_1 \cdot L_2 = \{a^n b^n \mid n \geq 0\}$

Which one of the following is CORRECT?

A. Only I

B. Only II

C. Both I and II

D. Neither I nor II

[MCQ]

#Q. Let L_1, L_2 be any two context-free languages and R be any regular language. Then which of the following is/are CORRECT?

I. $L_1 \cup L_2$ is context-free.

I L_1 is context-free.

III. $L_1 - R$ is context-free.

IV. $L_1 \cap L_2$ is context-free.

A I, II and IV only

B I and III only

C II and IV only

D I only

[MCQ]

#Q. Let L_1 be a recursive language. Let L_2 and L_3 be language that are recursively enumerable but not recursive. Which of the following statements is not necessarily true?

$L_2 - L_1 = L_2 \cap L_1^c$

A. $L_2 - L_1$ is reclusively enumerable — true

REL ∧ REC

REL ∧ REL = REL

not REL

B. $L_1 - L_3$ is reclusively enumerable → false

C. $L_2 \cap L_3$ is reclusively enumerable → true

$L_1 - L_3 = L_1 \cap L_3^c$

= REC ∧ not REL

D. $L_2 \cup L_3$ is reclusively enumerable → true

#Q. Consider the following types of languages L_1: Regular, L_2: Context-free, L_3: Recursive, L_4: Recursively enumerable.

Which of the following is/are TRUE?

$L_3^c \wedge L_4 \Rightarrow REL$

$REC \wedge REL$

$REL \wedge REL$

(2) $L_3^c \cup L_2 \cup L_4 \Rightarrow REL$

$REC \cup CFL \cup REL$

$REL \cup REL \cup REL$

[redacted]

(3) $\left\{ L_3^c \wedge L_4^c \Rightarrow REL \right\} \rightarrow$ false

$\left. REC \wedge \text{not } REL \right\} \text{nonREL}$

A. I Only

B. I and III only

C. I and IV only

D. I, II and III only

Topic : Undecidability Question

PROBLEMS	RL	DCFL	CFL	CSL	REC.L	REL
Does 'w' belongs to language L?	D	D	D	D	D	UD
Is L= null? (i.e, emptiness problem)	D	D	D	UD	UD	UD
Is L= E*? (i.e, completeness problem)	D	D	UD	UD	UD	UD
Is L1 L2? (i.e, equality problem)	D	D	UD	UD	UD	UD
ds L1 subset of L2? (i.e, subset problem	D	D	UD	UD	UD	UD
Is L1 intersection of L2= null?	D	UD	UD	UD	UD	UD
Is 'L' finite or not? (i.e, finiteness problem)	D	UD	D	UD	UD	UD
Is compliment of 'L' a language of same type or not?	D	D	UD	D	D	UD
Is intersection of two languages of same type or not?	D	UD	UD	D	D	D
Is 'L' regular language or not? ('L' is any language.)	D	D	UD	UD	UD	UD

[MCQ]

#Q. Which of the following problems are undecidable

A. Membership problem in context-free languages

B. Whether a given context-free language is regular

C. Whether a finite state automation halts on all inputs

D. Membership problem for type 0 languages

[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 $L(G_1) = L(G_2)$

[MCQ]

#Q. Consider the following problems L(G) denotes the language generated by a grammar G. L(M) denotes the language accepted by a machine M.

I. For an unrestricted grammar G and a string w, whether $w \in L(G.)$
II. Given a Turing Machine M, whether L(M) is regular.
III. Given two grammars G_1 and G_2, whether $L(G_1) = L(G_2)$.
IV. Given an NFA N, whether there is a deterministic PDA P such that N and P accept the same language.

Which one of the following statements is correct?

A. Only I and II are undecidable

B. Only III is undecidable

C. Only II and IV are undecidable

D. Only I, II and III are undecidable

2 mins Summary

Topic One

Topic Two

Topic Three

Topic Four

Topic Five

THANK - YOU