THEORY OF AUTOMATA AND FORMAL LANGUAGES

BCSC0011

-By Nishtha Parashar

Assistant Professor, Department of CEA, GLA University, Mathura

VARIANTS OF TURING MACHINE

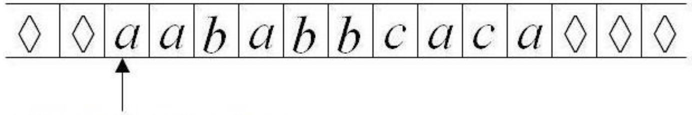
1. Turing Machine with Stay Option

- In standard turing machine, the R/W head must move either Left or Right.

- We can define TM with Stay-option

$$\delta: Q \times T \rightarrow Q \times T \times \{L, R, S\}$$

The head can stay in the same position



Left, Right, Stay

L,R,S: moves

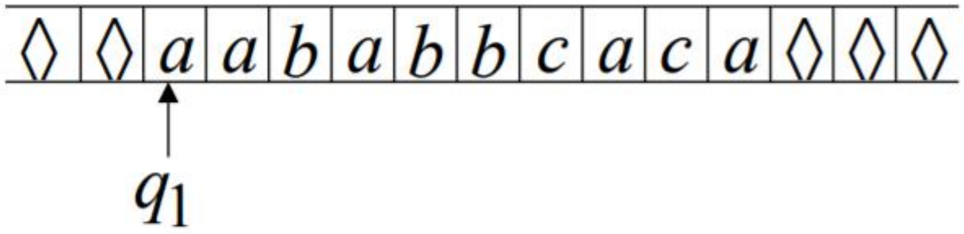
GLA UNIVERSITY
MATHURA

Accredited with A Grade by NAAC

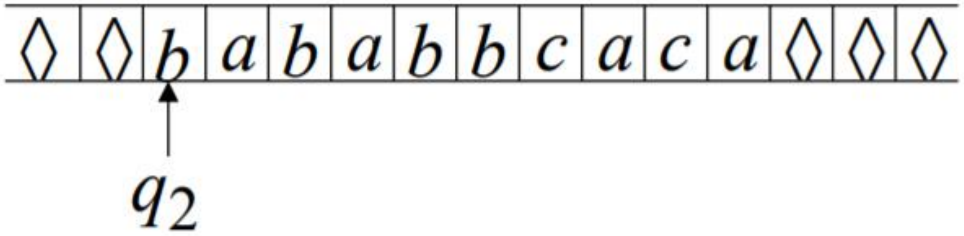
12-B Status from UGC

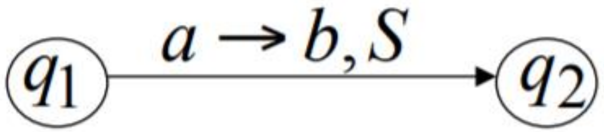
Example:

Time 1



Time 2

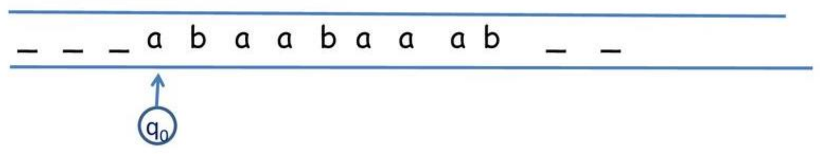




2. Two-way infinite Tape Turing Machine

- Infinite tape of two-way infinite TM is unbounded in both directions left and right.
- Let M0 with one-way infinite tape TM imitating one-way infinite tape TM, M
- The first thing that M0 needs to do is to mark the start of its tape.
- It can move the whole string to the right every time M moves left to the marker.

Two-way Infinite Tape

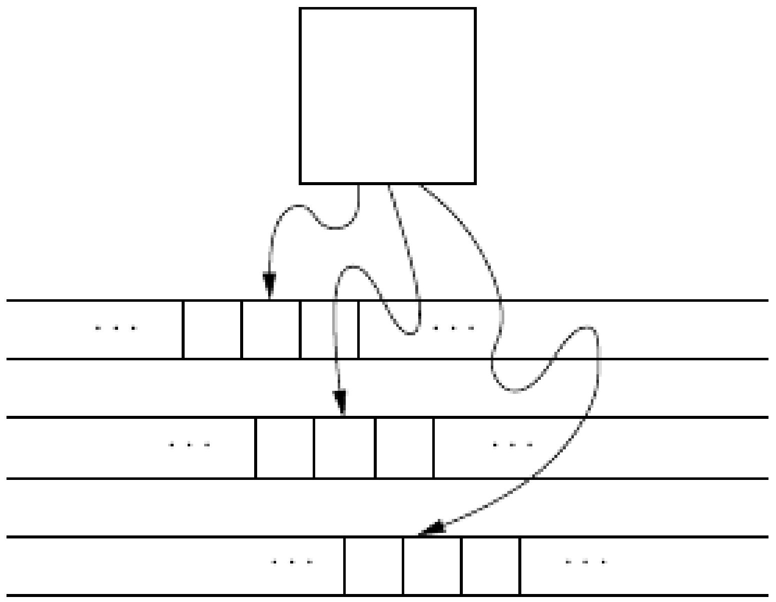


3. Multi-Tape Turing Machine

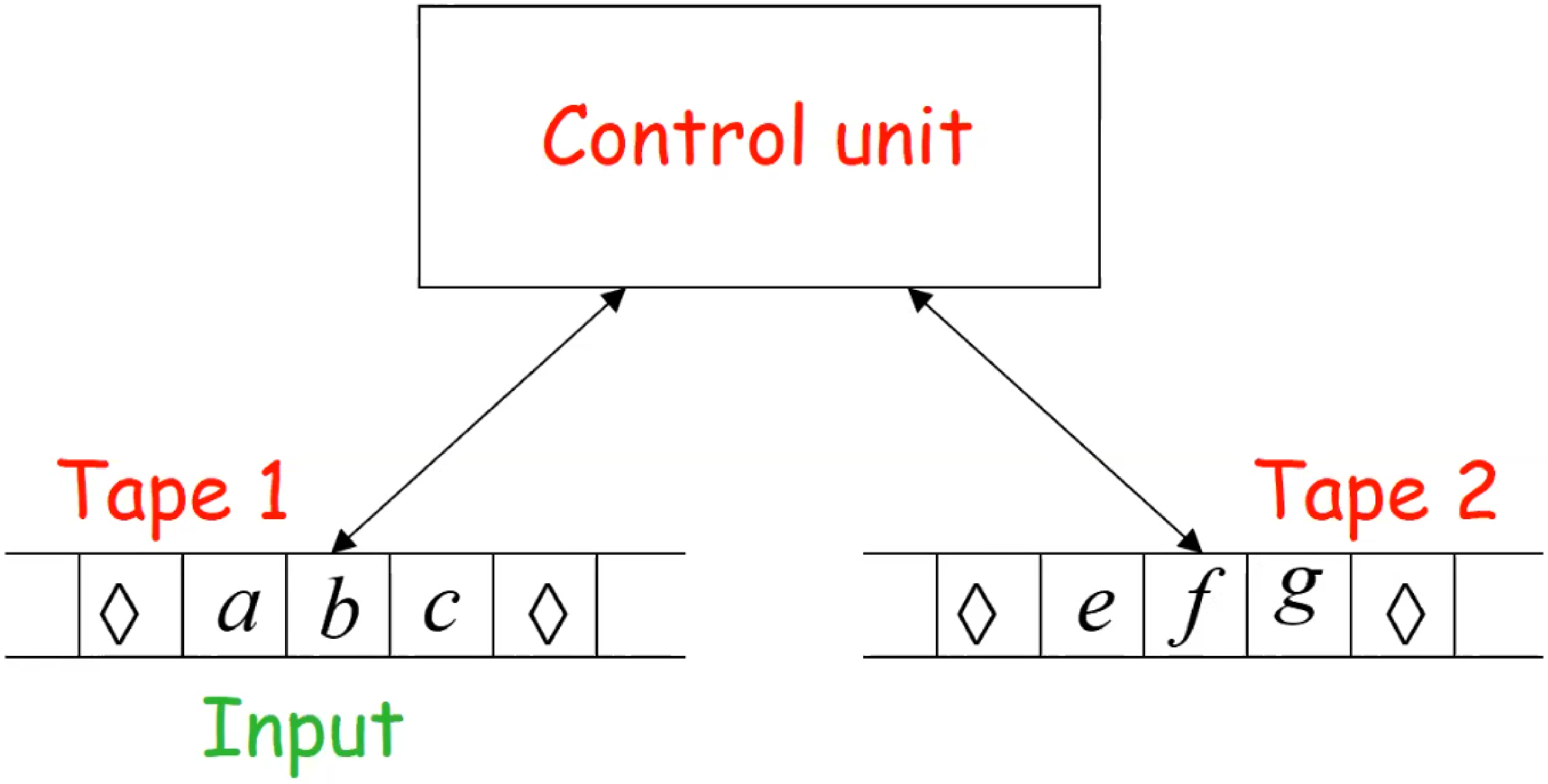
- TM with k tapes and k heads.
- Each tape has its own head for reading and writing.
- Initially the input appears on tape 1, and the others start out blank.

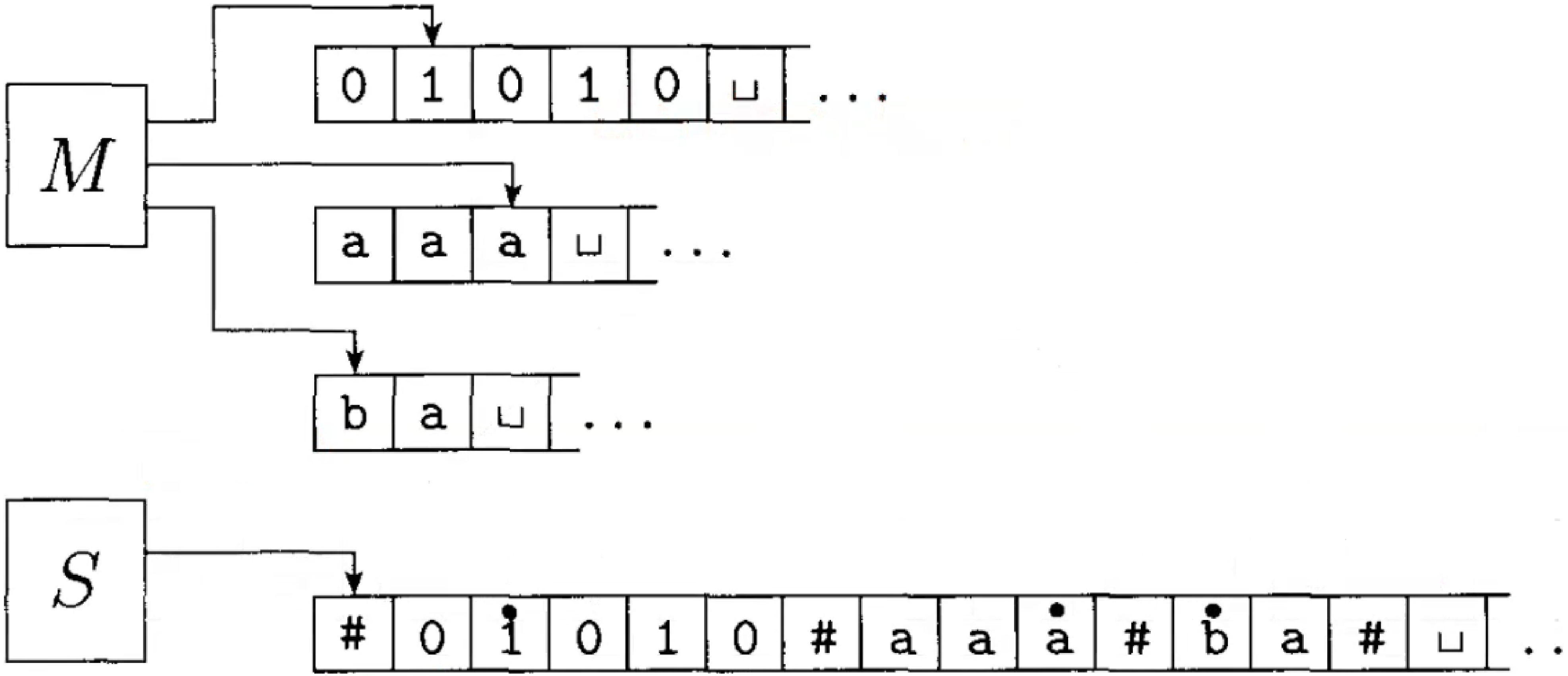
$$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L,R\}^k$$

$$\delta(q_i, a_1, \ldots, a_k) = (q_j, b_1, \ldots, b_k, L, R, \ldots, L)$$



Model of Multi-Tape Turing Machine



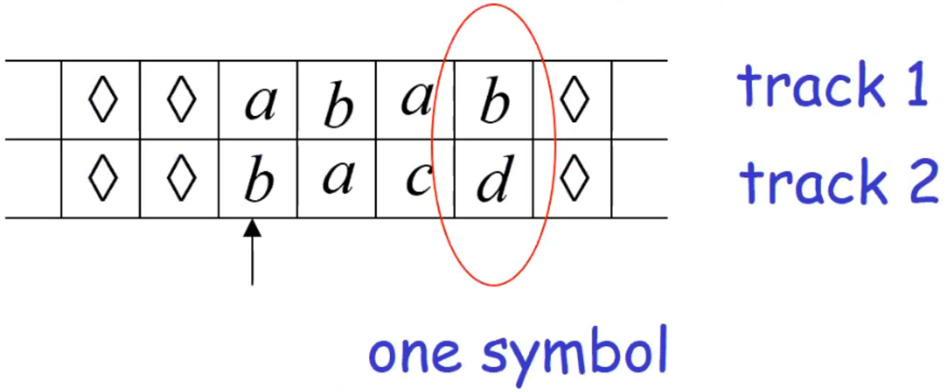


Representing three tapes with one

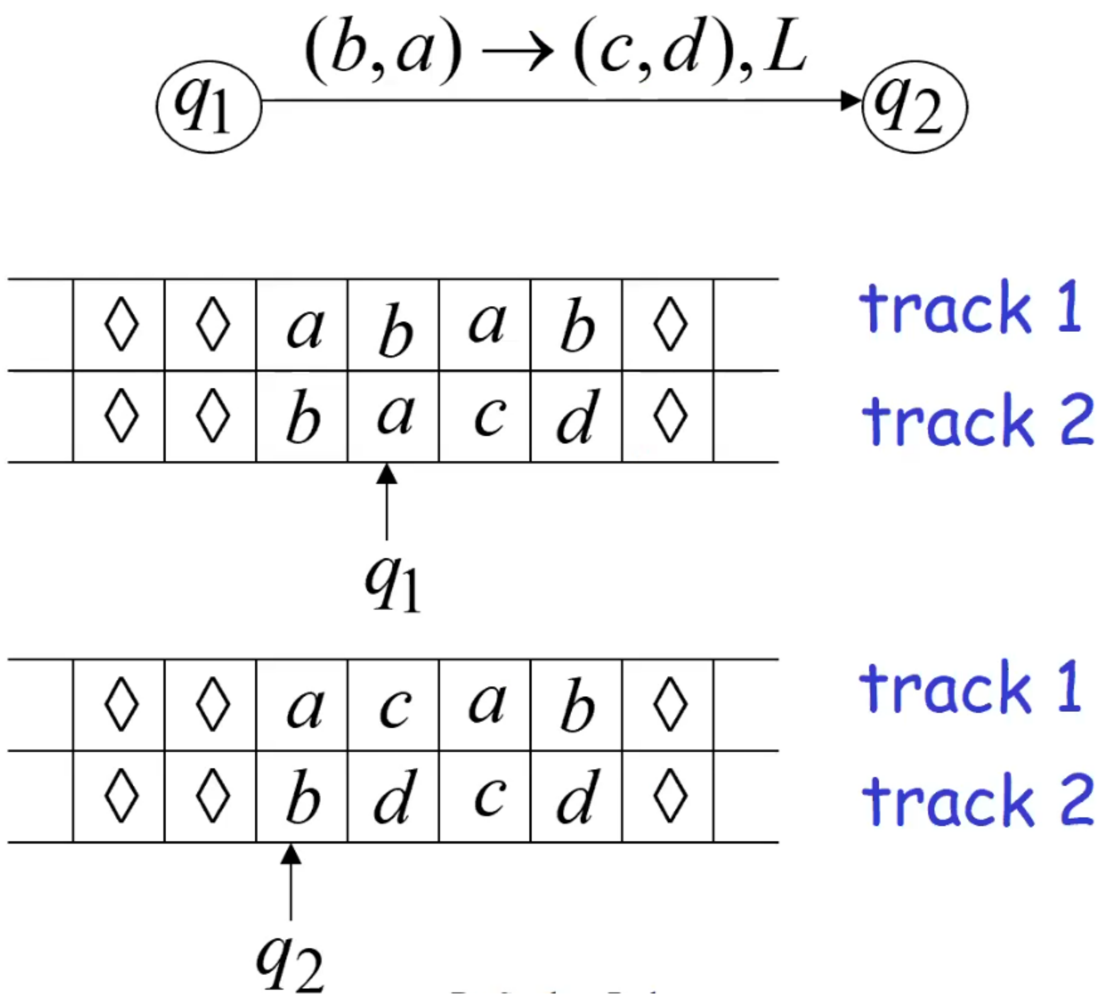
4. Multi-Track Turing Machine

- Multi-track Turing machines, a specific type of Multi-tape Turing machine, contain multiple tracks but just one tape head reads and writes on all tracks.
- Here, a single tape head reads n symbols from n tracks at one step.

$$\delta(Q_i, [a_1, a_2, a_3, \ldots]) = (Q_j, [b_1, b_2, b_3, \ldots], \text{Left\_shift or Right\_shift})$$



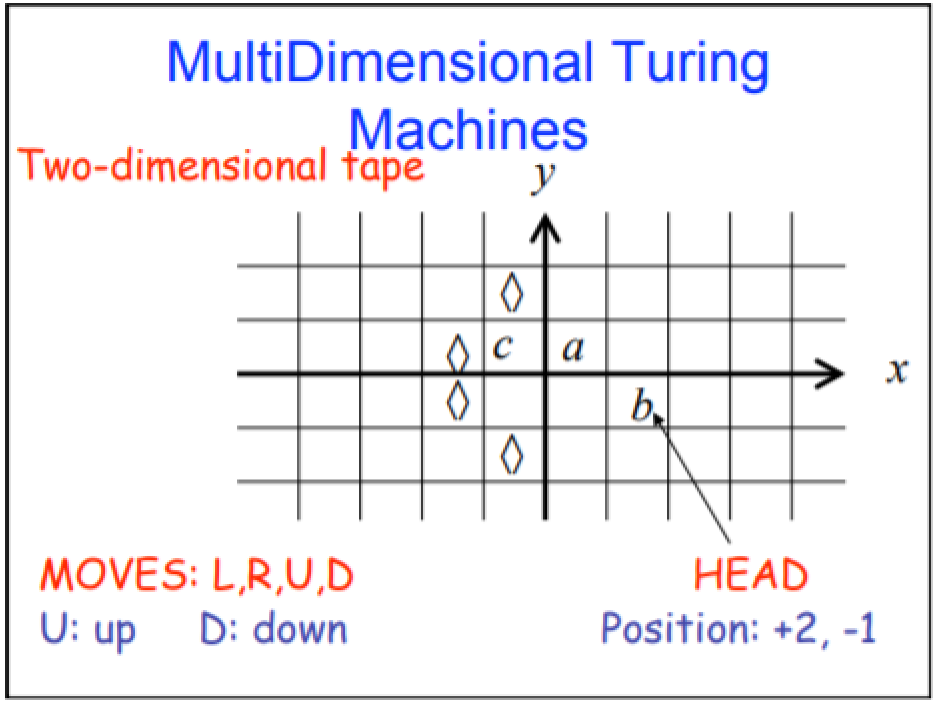
Transition on Multi-Track Turing Machine



5. Multidimensional Turing Machine

- It has multi-dimensional tape where head can move any direction that is left, right, up or down.

$\delta : Q \times \Gamma \quad Q \times \Gamma \times \{L, R, U, D\}$



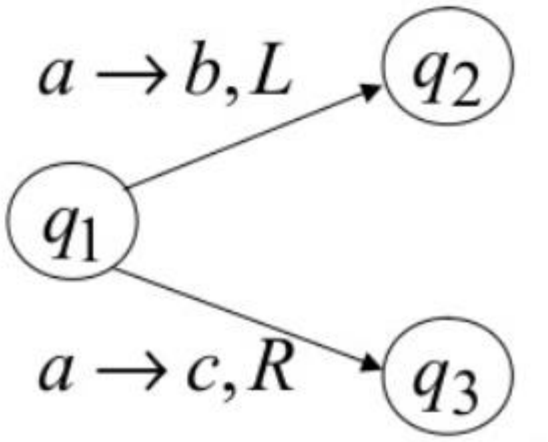
GLA UNIVERSITY MATHURA
Accredited with A Grade by NAAC
12-B Status from UGC

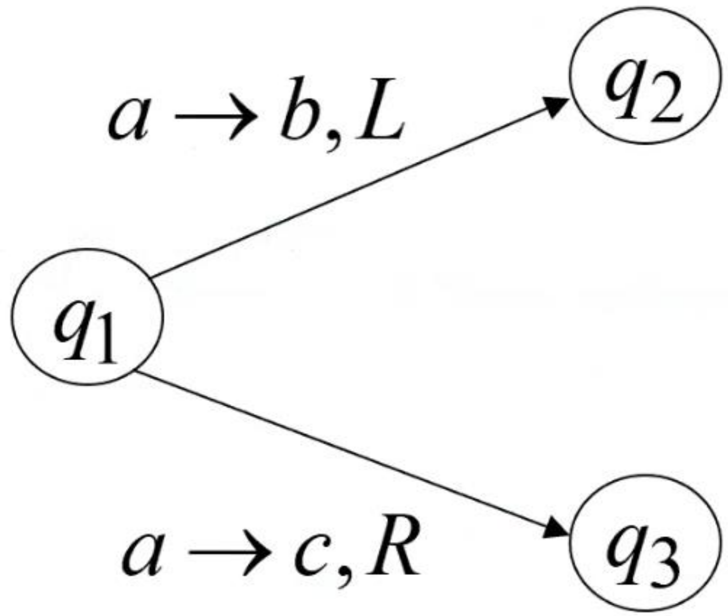
6. Non-deterministic Turing Machine

- At any point in a computation the machine may proceed according to several possibilities.

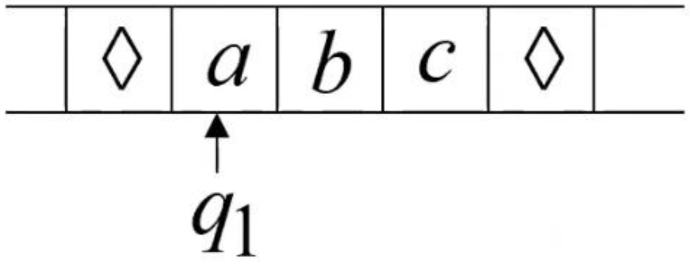
$$\delta: Q \times \Gamma \rightarrow 2^{(Q \times \Gamma \times \{L,R\})}$$

$$\delta(q_i, a) = \{(q_1, b, R), (q_2, c, L), \ldots, (q_m, d, R)\}$$



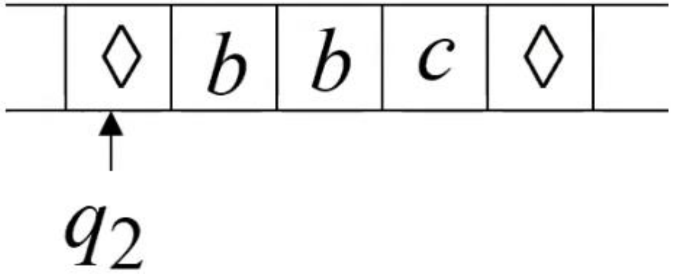


Time 0

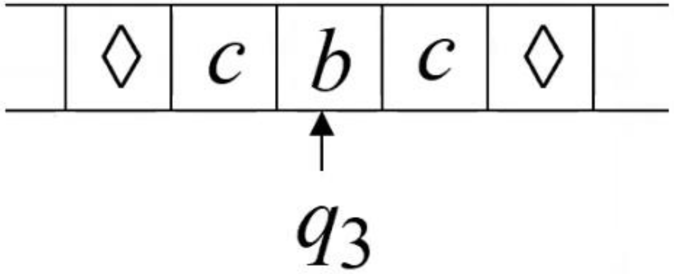


Time 1

Choice 1

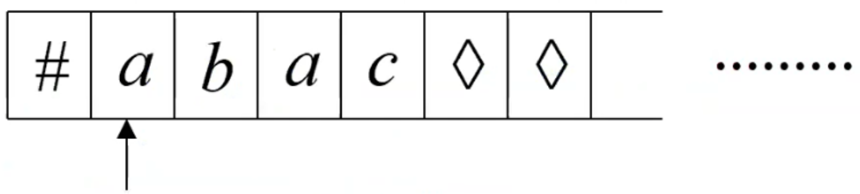


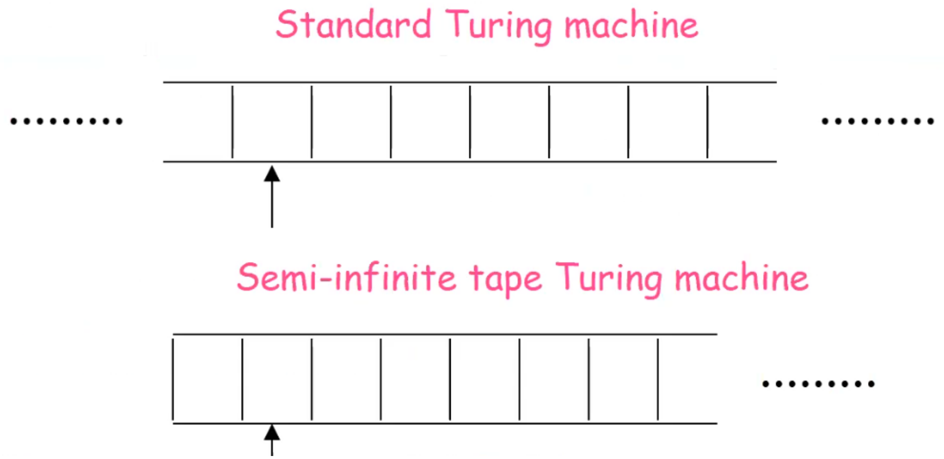
Choice 2



7. Semi-Infinite Tape Turing Machine

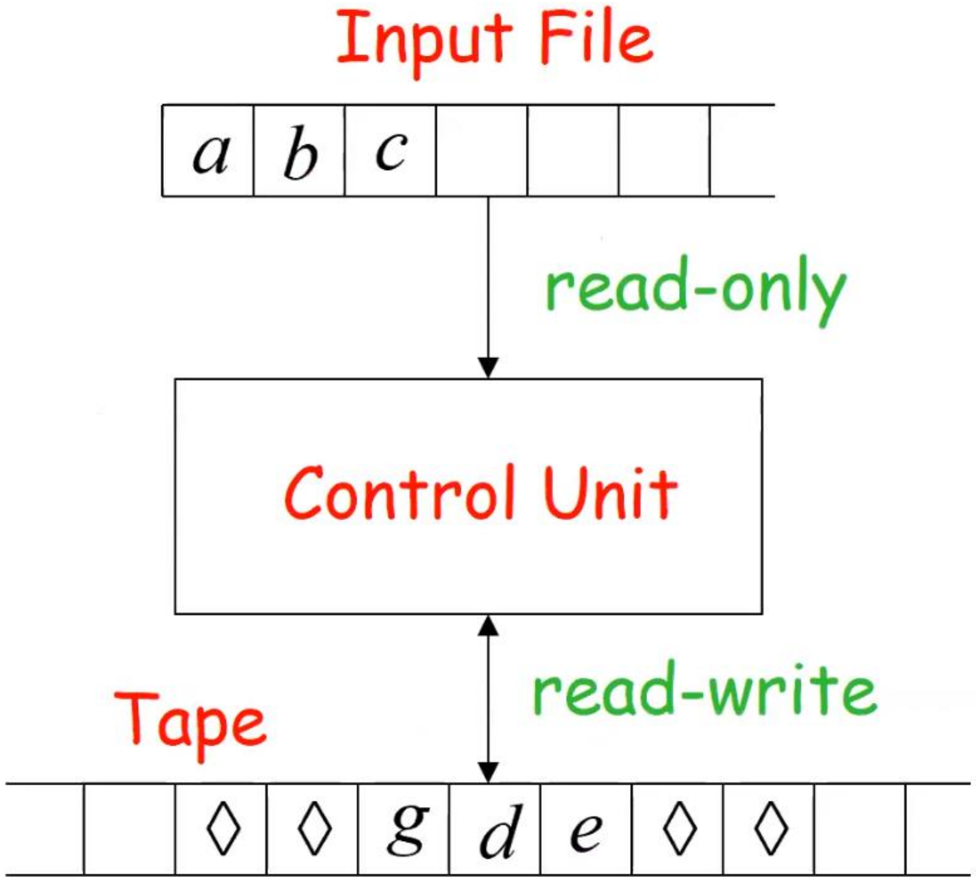
- A Turing Machine with a semi-infinite tape has a left end but no right end. The left end is limited with an end marker.





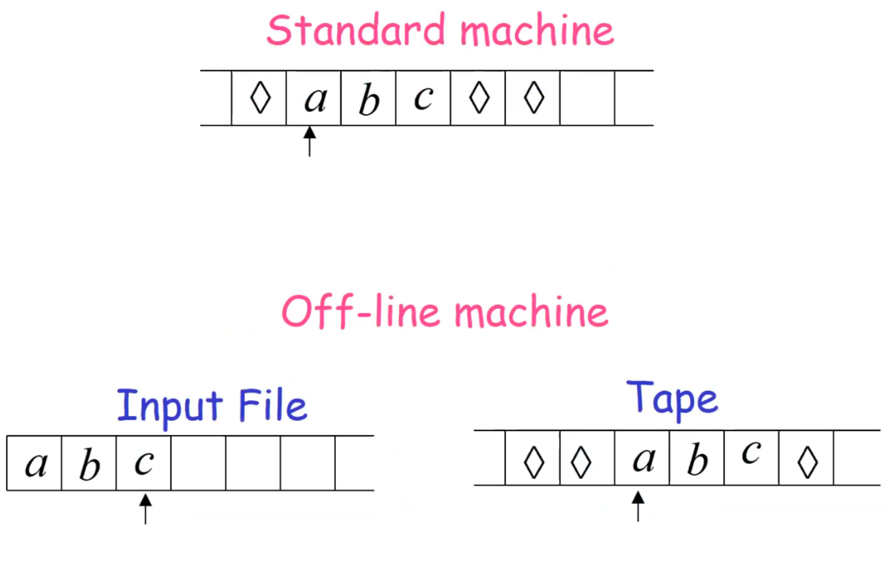
8. Offline Turing Machine

- An offline turing machine has two tapes
  1. One tape is read-only and contains the input.
  2. The other is read-write and is initially blank.

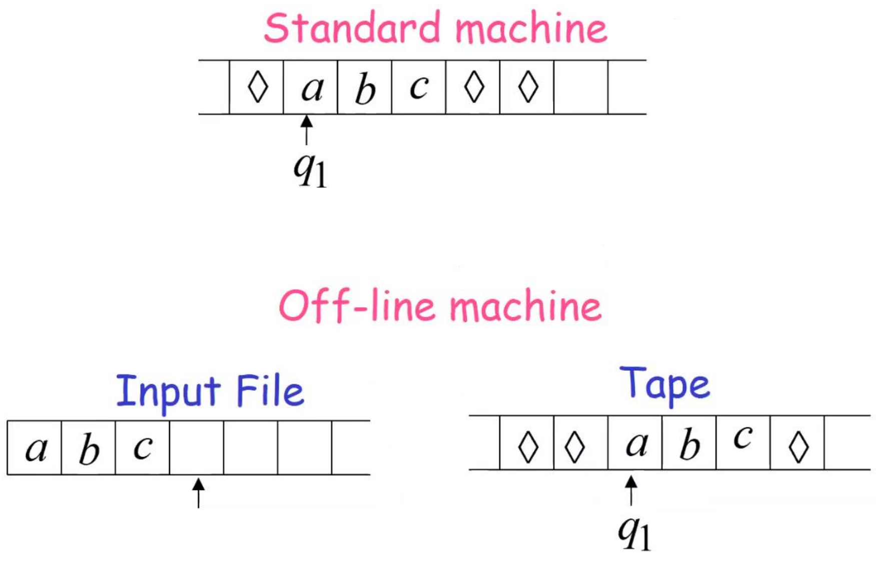


Offline machines simulate Standard Turing Machine

- Offline Machine:
  1. Copy input file to tape.
  2. Continue computation as in Standard Turing Machine.



1. Copy input file to tape



2. Do computations as in Turing machine

CHURCH-TURING THESIS

Church-Turing Thesis

Alonzo Church
(1903-1995)

Alan Turing
(1912-1954)

Some common interpretations of the thesis are-

- **Every algorithmically computable function is TM-computable.**
- **A function is computable iff it can be solved by a Turing Machine.**

Thesis not Theorem: because we cannot prove this.. With a counter example we can try to disprove it (but this has not been done yet).

Church-Turing Thesis

TM is a general model of computation, is simply to say that any algorithmic procedure that can be carried out by any mechanical means can be carried by a TM. This statement was first formulated by Alonzo Church in 1930. It was usually refer to as Church Thesis or Church Turing Thesis.

Some arguments why Turing Thesis is accepted as definition of mechanical computation or computer.

a) Anything that can be done by existing digital computer can also be done by ‘TM’.
b) No one has yet been able to suggest a problem, solvable by what we intuitively consider an algorithm, for which a turing machine program cannot be written.
c) Alternative models have been proposed for mechanical computations, but none of them are more powerful than the Turing machine model.

- It is a mathematically precise statement because we do not have a precise definition of the term “Algorithmic Procedure”. Therefore, it is not something that we can prove. (that’s why it is not a theorem)

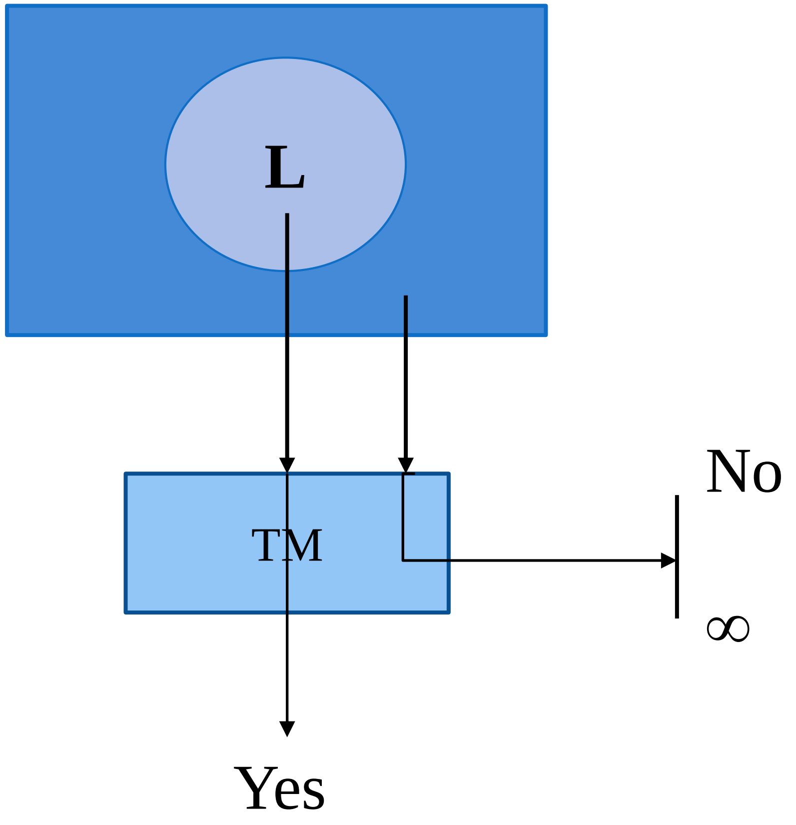
- Since invention of TM however enough evidence has accumulated to cause the church-turing thesis to be generally accepted.

RECURSIVELY ENUMERABLE
&
RECURSIVE LANGUAGES

Definition:

A language is **recursively enumerable** if some Turing Machine accepts it.

- Let L be a recursively enumerable language and M be the Turing Machine that accepts it.



For string w:

- If $w \in L$ then M halts in a final state.
- If $w \notin L$ then M halts in a non-final state or loops forever.

Definition:

- A language is **recursive**, if some Turing Machine accepts it and halts on any input string.

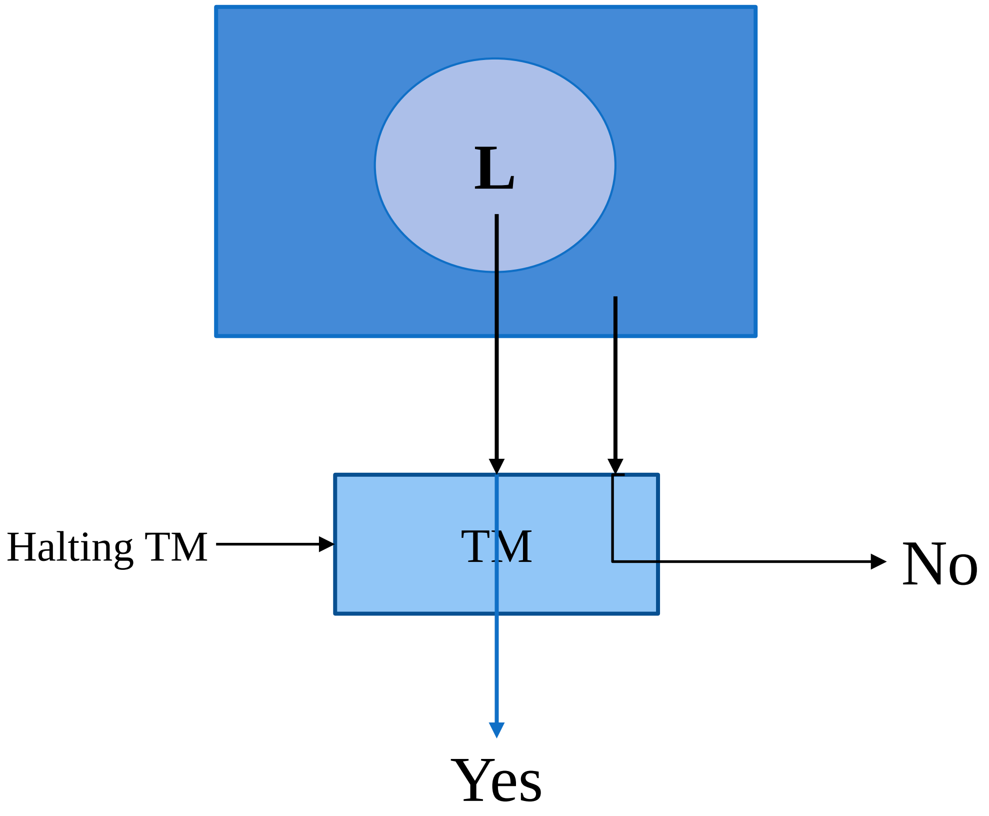
In other words:

- If for a language there is a halting turing machine then such language is called as recursive language.

- Let L be a recursive language and M be the Turing Machine that accepts it.

For string w

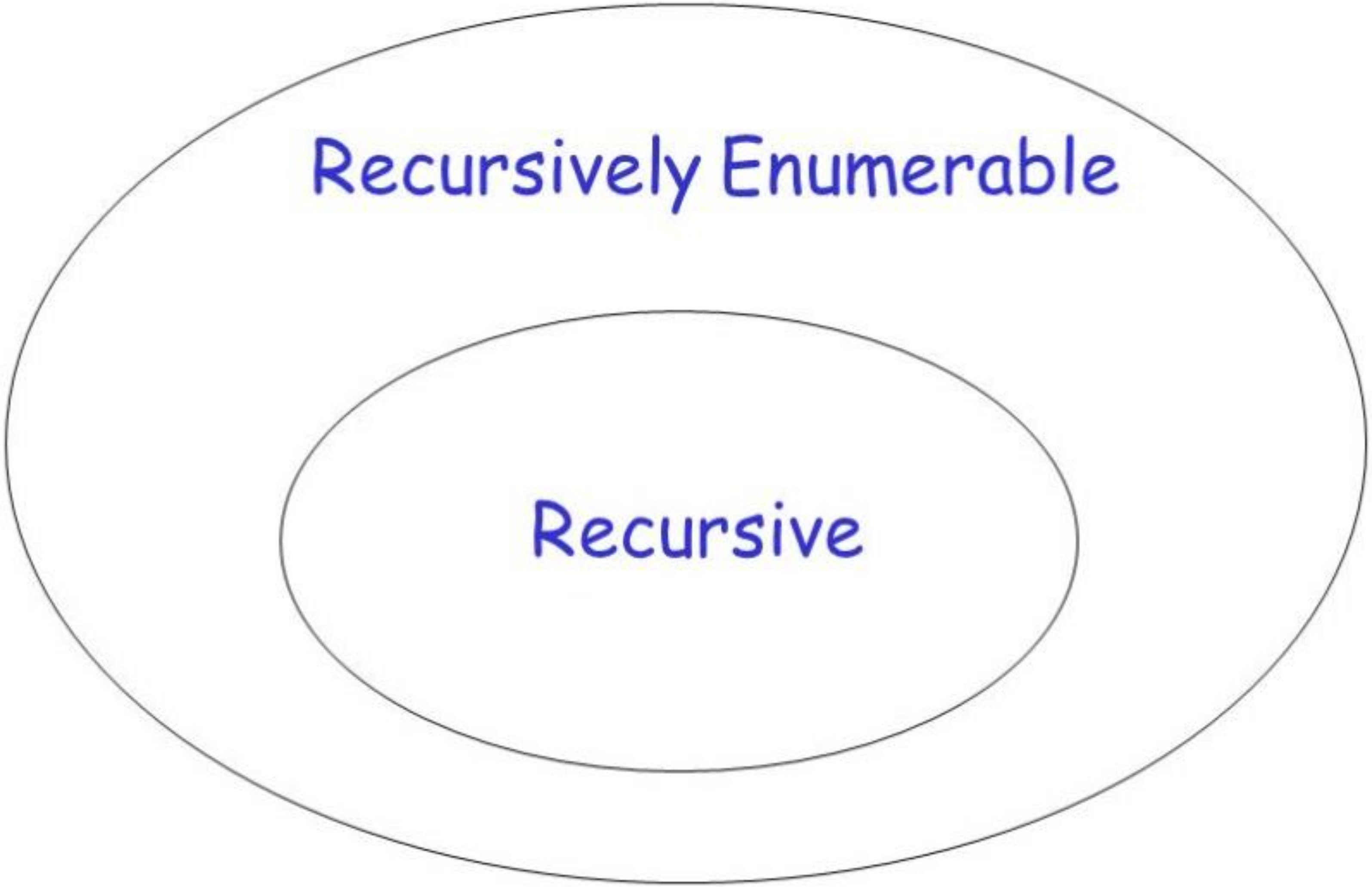
- If $w \in L$ then M halts in a final state.
- If $w \notin L$ then M halts in a non-final state.

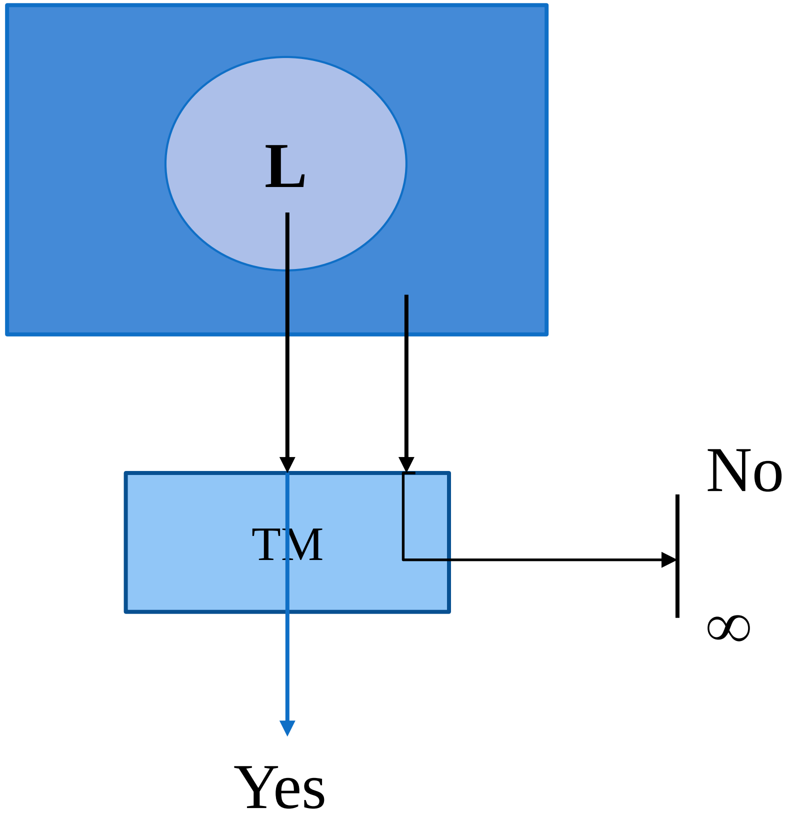


REMARK:

- A Problem/Language with two answers (yes/no) is decidable.

- A Problem/ Language is undecidable if it is not decidable.


Recursively Enumerable
Recursive


L
TM
No
∞
Yes

Closure Properties of Languages

Property	Regular	CFL	DCFL	CSL	Recursive	RE
Union	Yes	Yes	No	Yes	Yes	Yes
Intersection	Yes	No	No	Yes	Yes	Yes
Set Difference	Yes	No	No	Yes	Yes	No
Complementation	Yes	No	Yes	Yes	Yes	No
Intersection with a regular lang.	Yes	Yes	Yes	Yes	Yes	Yes
Concatenation	Yes	Yes	No	Yes	Yes	Yes
Kleen Closure	Yes	Yes	No	Yes	Yes	Yes
Kleen Plus	Yes	Yes	No	Yes	Yes	Yes
Reversal	Yes	Yes	Yes	Yes	Yes	Yes
Homomorphism	Yes	Yes	No	No	No	Yes
ϵ-free Homomorphism	Yes	Yes	No	Yes	Yes	Yes
Inverse Homomorphism	Yes	Yes	Yes	Yes	Yes	Yes
Substitution	Yes	Yes	No	No	No	Yes

1. The context free languages are closed under:
   a) Intersection
   b) Complement
   c) Kleene
   d) None of the mentioned

2. Context free languages are not closed under:
   a) Intersection
   b) Intersection with Regular Language
   c) Complement
   d) All of the mentioned

4. If L1 and L2 are context free languages, L1-L2 are context free:
   a) always
   b) sometimes
   c) never
   d) none of the mentioned

5. Recursively Enumerable languages are not closed under
   a) **Complementation**
   b) Union
   c) Intersection
   d) none of these

6. If there exists a language L, for which there exists a TM, T, that accepts every word in L and either rejects or loops for every word that is not in L, is called
   a) Recursive
   b) Recursively Enumerable
   c) NP-Hard
   d) none of the mentioned

7. Universal Turing Machine influenced the concept of

a) stored program computers.

b) interpretative implementation of programming language.

c) computability.

d) all of these.