Turing Machine

Er. Simanta Kasaju

Assistant Professor

NCIT College

Contents

-Halting Problem

-Solvable and Unsolvable problmes

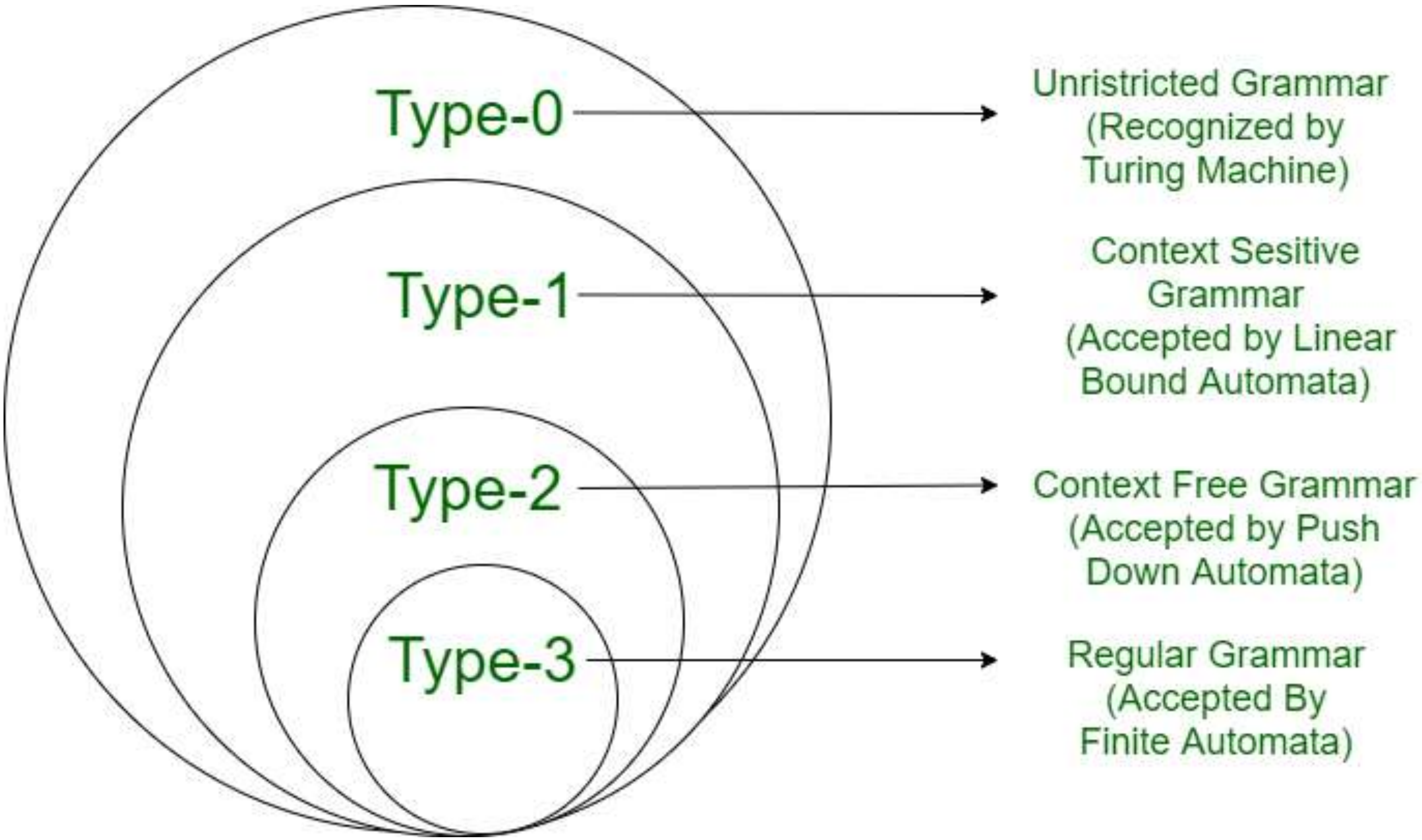
-Decidable,semi decidable and undecidable Problems

tractable and intractable problems

-P,NP,NP hard,NP complete

-Time complexity and space complexity

-Reducability


Type-0
Type-1
Type-2
Type-3
Unristricted Grammar
(Recognized by
Turing Machine)
Context Sesitive
Grammar
(Accepted by Linear
Bound Automata)
Context Free Grammar
(Accepted by Push
Down Automata)
Regular Grammar
(Accepted By
Finite Automata)

Introduction

- It is an abstract machine developed by English mathematician Alan Turing in 1936.
- It is the theoretical foundation of modern computer.

TM have

Finite set of alphabets

Finite set of states

Linear tape which is potentially infinite in both end.

- Each square or cell of tape can hold one symbol from alphabet.
- If no symbol then it contains blank.
- Reading and writing is done by tape head.

Cont..

Tape serves as

Input device

Memory available for use during computation

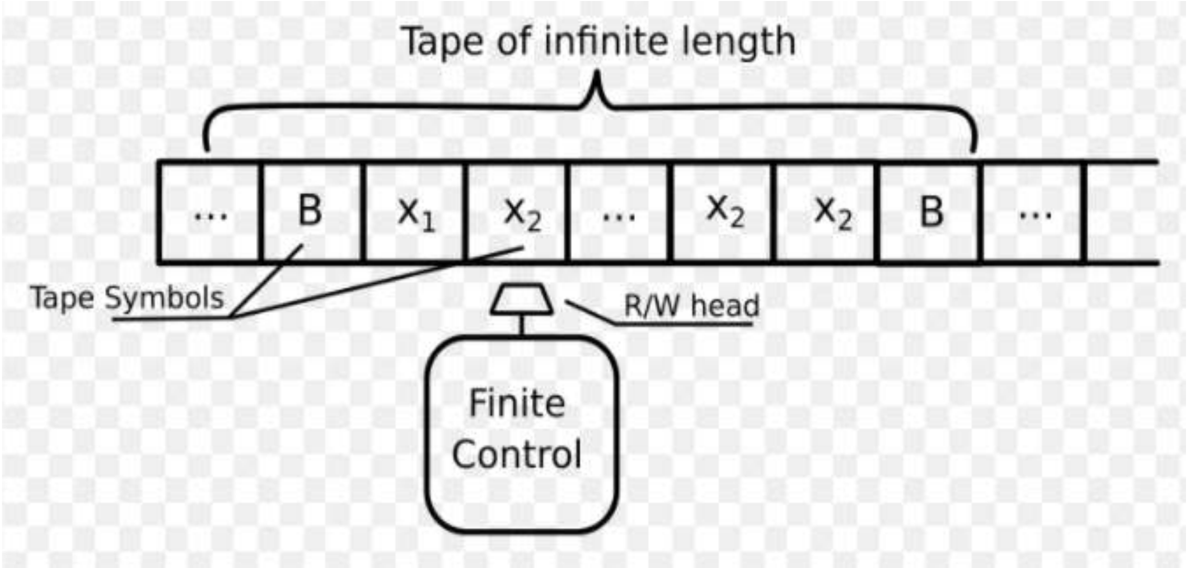
Output device

- TM is much more accurate model of general purpose computer.
- It can do everything that a real computer can do.

Features of Turing Machine

- Head or tape head can read / write
- Tape head can move in both the direction i.e. Right/left along the input
- State can also have halt state
- Special character “#” or “B” is introduced as an empty(Blank) character.
- Can accept infinite length of string in both directions.

Architecture of a physical Turing machine OR Turing machine as a physical computing Device



TM performs a sequence of computation steps. In one step it does following:

- Immediately before the computation step, TM is in state r of Q and each of K tape heads is on certain cell.

• TM switches to state s of Q (which may be equal to r) •Each tape head writes a symbol of tape alphabet in the cell it is currently scanning.

• Each tape head either moves on one cell to the left or to the right or stays at current cell.

A TM consist of:

i) A Tape Head
ii) An infinite Tape
iii) A finite control system
iv) Move of TM

Formal definition of Turing Machine:

A Turing Machine is a 7 tuple of the

- **TM is of 7 tuple :•**

TM = (∑ , Γ , Q , δ , q, q_{accept} , q_{reject}) OR TM = (∑ , Γ , Q , δ , q0 , B, F)

Where

- **∑ is a finite set of input alphabets ; ∑ is subset of Γ**

Γ is a finite set of tape alphabets ;

- **Q is a finite set of states**
- **q is a start state**
- **q_{accept} is a accepting state**
- **q_{reject} is a rejecting state**
- **δ is a transition function**

q_0 -> Start symbol
B-> Blank character
F-> Final state

Where δ : Q X Γ -> Q X Γ X {L,R,N} Here, transition function δ is the program of TM that tells what machine can do in one computation step

Instantaneous Description for Turing Machine:

- A string $x_1x_2......x_{i-1}qx_ix_{i+1}......x_n$ represents instantaneous description of TM where q is state of TM.
- The tape head is scanning the i^{th} symbol from left.
- $x_1..........x_n$ is the portion of tape between leftmost and rightmost non blank.

Moves of Turing Machine:

The move of TM , M = =(∑ , Γ , Q , δ , q0, B, F) is described by the notation ├ for single move and $\vdash^*$ for zero or more moves.

For δ (q, x_i) = (p, Y, L) i.e. next move is leftward. Then,

$x_1x_2.......x_{i-1}\ q\ x_ix_{i+1}......x_n \vdash x_1x_2.......x_{i-2}\ p\ x_{i-1}Y\ x_{i+1}......x_n$

for i=1, $q\ x_1x_2......x_n \vdash p\ B\ Y\ x_2......x_n$

for i=n and Y = B, $x_1x_2......x_{n-1}\ q\ x_n \vdash x_1x_2......x_{n-2}\ P\ x_{n-1}$

Turing Machine for computing a function

- A TM can be used to compute a function.
- For this, adopt following policy to input string to TM which is input to

computation function:

- String w is presented into form of BwB, where B is blank symbol , and placed on the tape. The head is positioned at a blank symbol which immediately follows the string w.
- The symbol to the current position of tape head can be shown as (q, BwB) or presented by Instantaneous Description of TM as BwqB.
- TM is said to halt on input w if we can reach to halting state after performing

some operations.

- i.e. for TM , M = (Q , ∑ , Γ , δ , q_0, B, {q_f}) ; TM is said to halt on input w if and only BwqB yields to Bαq_fB for some α ε Γ*

(q , BwB) ├ (q_f, BαB)

We can perform following operation using TM

- Addition
- Subtraction
- Multiplication
- Increment
- Division

TM as transducer

- 1’s Complement
- 2’s Complement

TM as Eraser
TM as Language Acceptor

Extension of Turing machine

Types of Turing machine

i. Multiple tapes Turing Machine
ii. Multiple head Turing Machine
iii. Two-way Infinite tape Turing machine
iv. K- dimensional Turing machine

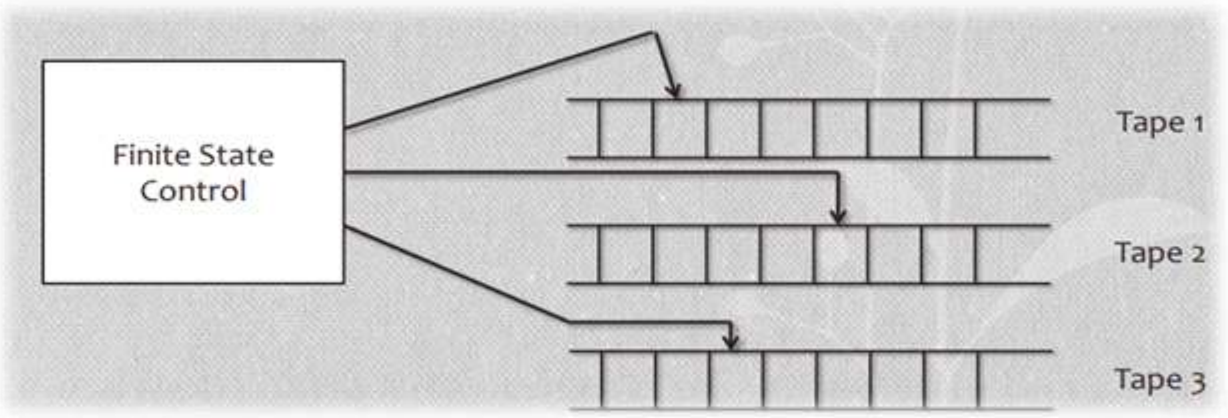
i) Multiple tapes Turing Machine:

Multi-tape Turing Machines have multiple tapes where each tape is accessed with a separate head.

Each head can move independently of the other heads. Initially the input is on tape 1 and others are blank.

At first, the first tape is occupied by the input and the other tapes are kept blank.

Next, the machine reads consecutive symbols under its heads and the TM prints a symbol on each tape and moves its heads.



Here, A Multi-tape Turing machine can be formally described as a 6-tuple

Tm= ({Q, ∑ , Γ , δ , q0, h}) where

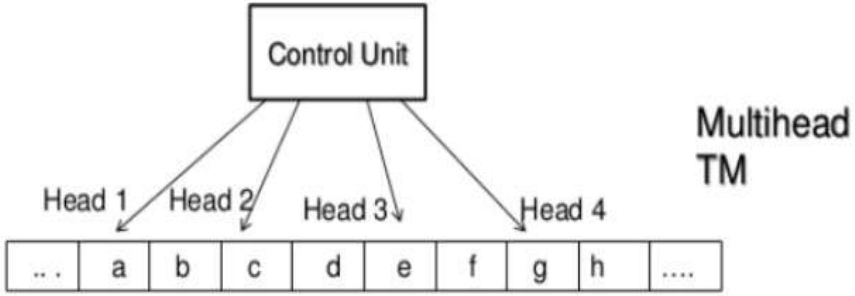
- **Q** = total no. of states
- ∑= tape symbols excluding #
- Γ= tape symbols including # i.e. ∑ U #
- **δ** = transition function which maps
- q_0= initial state
- B=Blank
- q_f = halt state

$(QX\ \Sigma^m) \rightarrow QX(\Sigma^m\ U\ \{ L, R, N\}^m)$ where,

$m \geq 1$ be an integer that indicates multiple tapes.

ii) Multiple head Turing Machine

→Single tape containing long strings of an input tape with multiple heads as shown in figure below:



→For our convenience and simplification, Let us assume there are only two heads on a tape. The tape is also assumed as one-way infinite.

Then, Let the content of the tape and the position of the two heads that is H1 and H2 be as given below

a b c # d e f # # . .

H_2 H_1

And Let the state of Tm is at q.

→Then the move function of the two headed one-way Turing may be defined as; S(state, symbol under head 1, symbol under head 2)

=(New states, (S_1,M_1), (S_2,M_2))

Prove, Every multiple turing machine has an equivalent single tape turing machine

- Let K>=1 be an integer. Any k tape TM can be converted to an equivalent one tape TM.
- Here,
- Convert a multi-tape TM M to an equivalent single tape TM S.

Key idea is to show how to simulate M with S.

- Let M has k tapes, then S simulates the effect of k tapes by storing their information on its single tape.
- It uses the new symbol B(#) as a delimiter to separate the contents of the different tapes.
- In addition to the contents of those tapes, S must keep track of the locations of the heads by writing a tape symbol with a dot(.) above it to mark the place where the head on that tape would be.
- Following figure illustrates how one tape can be used to represent three tapes.

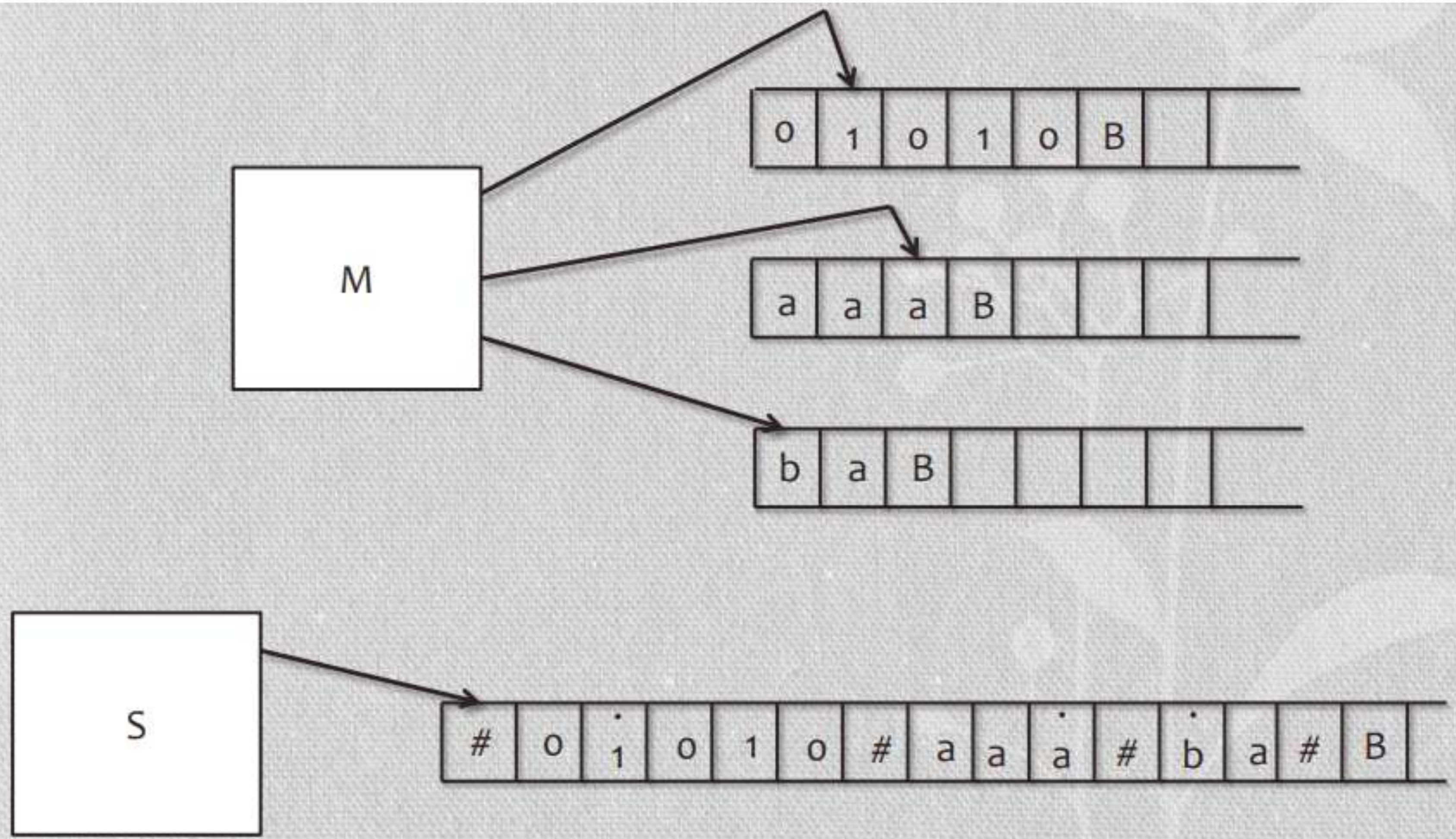


Figure : representing 3 tapes with one.

iii)Two-way Infinite tape Turing machine

→Both left side and right side are infinite

→Only one head but multiple long input string

→Its transition function **δ** is defined as;

(QX ∑)→(Q U { h })X ({ R, L} U ∑)

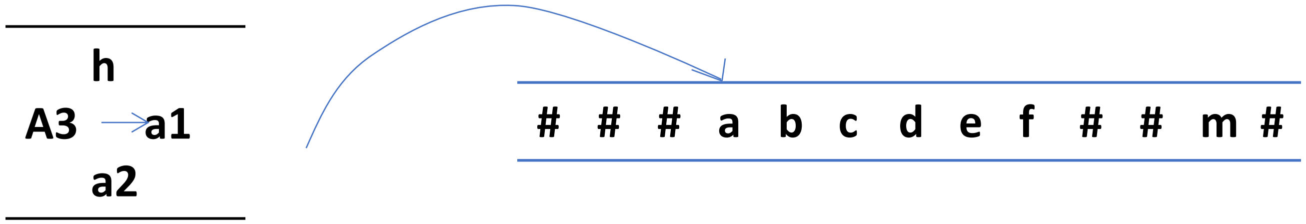


Fig: Two way infinite tape

→so No ceasing of operation without halting as there is no left end of the tape, therefore there is no possibility of jumping off the left-end of the tape.

iv) K- dimensional Turing machine

→These type of Turing machines have control over the finite states.

→It has a single head which can move right, left, up and down.

→Here δ can be defined as;

$Q X \sum \rightarrow (Q \cup \{ h \}) X (\{ R, L, U, D\} \cup \sum)$

→The configuration at a particular instant is given by;

location on a tape   move right

$(q, a, (3,3)) \rightarrow (q_1, b, R)$

Current State   symbol of a tape under head   new state   updated symbol on a tape

In general;

$(q, H_1,(i_j, i_k)) \rightarrow [\text{new state}, C_{ij},\{ R, L, U, D\}]$

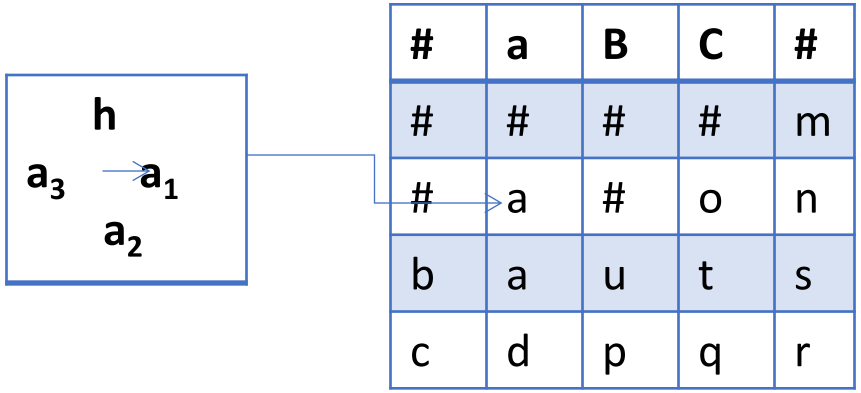


fig: 2-dimensional tape of Turing machine

Introduction(SET 1)

Topics

Introduction to Turing Machines (TM), Notations of Turing Machine, Acceptance of a string by a Turing Machines, Non-Deterministic Turing Machines,ID of TM

1. **Who proposed the concept of a Turing Machine?**

   a) Alan Turing

   b) Charles Babbage

   c) John von Neumann

   d) Claude Shannon

2. **A Turing machine consists of a finite set of states and a ______.**

   a) Stack

   b) Queue

   c) Tape

   d) Register

3. **What can the Turing machine's tape be best compared to?**

   a) A notebook

   b) A fixed memory

   c) An infinite memory

   d) A temporary buffer

4. **Which of the following is NOT an operation performed by a Turing machine?**

   a) Move left on the tape

b) Move right on the tape

c) Erase the entire tape

d) Read or write on the tape

5. The head of a Turing machine reads symbols from ______.

a) Memory

b) Stack

c) Queue

d) Tape

6. What does a Turing machine do when it encounters a halt state?

a) It continues to read the tape.

b) It resets the machine.

c) It stops all computations.

d) It clears the tape.

7. Which of the following is a common notation for representing a Turing Machine?

a) M = (Q, Σ, Γ, δ, q0, B, F)

b) M = (Q, Σ, q0, δ, B)

c) M = (Q, Σ, δ, F)

d) M = (Q, Σ, B, δ, q0, F)

8. In the notation of a Turing Machine, what does 'Σ' represent?

a) The set of all possible tapes

b) The input alphabet

c) The number of states

d) The halting condition

9. **What does the symbol 'B' or ‘#’ typically denote in the Turing machine formalism?**

a) Blank symbol

b) Backward movement

c) Binary symbol

d) Boolean operation

10. **The transition function of a Turing Machine is denoted by which symbol?**

a) α

b) δ

c) Σ

d) q

11. **Which of the following defines the action of a Turing Machine on each step?**

a) The length of the tape

b) The current symbol on the tape

c) The current state and tape symbol

d) The final state

12. **In a Turing machine, what happens when the head moves past the current end of input on the tape?**

a) The machine halts.

b) The machine writes a blank symbol.

c) The machine throws an error.

d) The machine resets.

13. **A string is accepted by a Turing Machine if the machine reaches the ______.**

a) Initial state

b) Halting state

c) Final state

d) Infinite loop

14. **In which state is a Turing Machine initialized?**

a) Final state

b) q0 (Initial state)

c) qh (Halt state)

d) Reject state

15. **Which of the following scenarios means that a Turing machine has accepted a string?**

a) The head moves off the tape.

b) The machine halts in an accepting state.

c) The machine runs infinitely.

d) The tape is full.

16. **How does a Turing machine reject a string?**

a) When it reaches a reject state

b) When it encounters a blank symbol

c) When it reaches the end of the tape

d) When the machine halts

17. **Which of the following languages is recognized by a Turing Machine?**

a) Regular languages

b) Context-free languages

c) Context-sensitive languages

d) Recursively enumerable languages

18. **What does a deterministic Turing machine imply?**

a) There are multiple possible moves from a given configuration.

b) There is exactly one move from each configuration.

c) It can halt in multiple states.

d) It reads multiple symbols at a time.

19. **Which of the following problems can a Turing machine solve?**

a) All computational problems

b) Decidable problems

c) P vs NP problems

d) Halting problem

20. **If a Turing Machine enters an infinite loop, it means:**

a) The string is accepted.

b) The string is rejected.

c) The machine never halts.

d) The machine halts after a long time.

21. **Which of the following machines is considered the most powerful computational model?**

a) Finite automaton

b) Pushdown automaton

c) Turing machine

d) Mealy machine

22. **What is the difference between a Turing machine and a finite automaton?**

a) A Turing machine has a tape, while a finite automaton does not.

b) A finite automaton can accept context-sensitive languages.

c) A Turing machine can read multiple input strings at once.

d) A finite automaton has more memory than a Turing machine.

23. **A Turing Machine M accepts a string w if starting from the initial configuration, M reaches:**

a) The start of the tape

b) An accepting configuration

c) The reject state

d) An infinite loop

24. **What does 'semi-decidable' mean in the context of Turing machines?**

a) The machine halts for all inputs.

b) The machine halts only on accepted inputs.

c) The machine accepts all inputs.

d) The machine rejects all inputs.

25. What is a "halting problem"?

a) A problem where the Turing machine halts.

b) The problem of deciding whether a Turing machine halts for a given input.

c) A problem solved by multi-tape Turing machines.

d) The process of erasing the tape in a Turing machine.

26. Which of the following problems is undecidable by a Turing machine?

a) Determining if a string belongs to a regular language

b) Checking the equality of two context-free languages

c) The halting problem

d) Recognizing regular languages

27. The blank symbol in a Turing machine represents:

a) A symbol that is part of the input alphabet

b) A symbol that is part of the tape alphabet but not the input alphabet

c) The halting condition

d) The machine reset signal

28. Consider a Turing machine with the transition function δ(q1, 1) = (q2, 0, L). What does this transition mean?

a) When in state q1 and reading 1, move to state q2, write 0, and move right.

b) When in state q1 and reading 1, move to state q2, write 0, and move left.

c) When in state q1 and reading 1, move to state q2, leave the symbol unchanged, and move left.

d) When in state q1 and reading 1, move to state q2, write 1, and move left.

29. **What is the difference between an "accepting state" and a "halting state" in a Turing machine?**

a) An accepting state halts the machine, but a halting state does not.

b) A halting state is only reached when the machine rejects a string.

c) Every accepting state is a halting state, but not every halting state is an accepting state.

d) A halting state rewinds the tape, while an accepting state moves the tape forward.

30. **Given a Turing machine with a transition δ(q, a) = (p, b, R), what can be inferred if after a certain number of steps the machine revisits state q and reads symbol a again?**

a) The machine will definitely halt.

b) The machine is in an infinite loop.

c) The string will be accepted.

d) The machine will reject the input.

31. **Which of the following is true for the set of all strings accepted by a Turing machine?**

a) It forms a context-free language.

b) It forms a recursive language.

c) It forms a recursively enumerable language.

d) It forms a regular language.

32. What can be inferred if a Turing machine enters a state and never transitions out of it, but does not reach a halting state?

a) The machine is undecidable.

b) The machine is non-deterministic.

c) The machine enters an infinite loop.

d) The machine accepts the string.

33. The language accepted by a non-deterministic Turing machine (NTM) is equivalent to the language accepted by a deterministic Turing machine (DTM). This is because:

a) Every NTM can be converted to a DTM without changing the language.

b) NTMs can only accept context-free languages.

c) NTMs are more powerful than DTMs.

d) NTMs are less powerful than DTMs.

34. Which of the following is undecidable for a Turing machine?

a) Checking if a string belongs to a regular language.

b) Checking if a Turing machine halts on all inputs.

c) Determining if a Turing machine accepts an empty language.

d) Deciding if a Turing machine halts on a specific input.

35. What is a configuration in the context of a Turing machine?

a) The current state of the tape.

b) The combination of the machine’s current state, the tape contents, and the head position.

c) The total number of symbols on the tape.

d) The number of transitions that have occurred.

36. A Turing machine M = (Q, Σ, B, δ, q0, F) accepts a language L. What does this imply about the relationship between L and Σ?

a) Every string over Σ is in L.

b) L contains only the strings that lead to a halting state in M.

c) L contains every string over Σ except the strings leading to infinite loops.

d) L is a superset of Σ.

37. Which of the following would NOT be true if a Turing machine accepted a recursively enumerable (RE) language?

a) The machine may enter an infinite loop for some inputs not in the language.

b) There exists a deterministic Turing machine that accepts the language.

c) Every string in the language leads the machine to a halting state.

d) There exists an algorithm that can decide membership for all strings in the language.

38. For a Turing machine, the "instantaneous description" (ID) refers to:

a) The state of the Turing machine at the start of execution.

b) The snapshot of the machine's state, tape contents, and head position at any moment.

c) The number of transitions remaining before the machine halts.

d) The tape contents after the machine halts.

39. How does a Turing machine recognize a non-deterministic language?

a) By exploring all possible computation paths simultaneously.

b) By simulating multiple deterministic Turing machines.

c) By using an infinite tape for non-deterministic branching.

d) By switching between deterministic and non-deterministic states.

40. Which of the following is true about the tape alphabet Γ of a Turing machine?

a) It includes all symbols in the input alphabet Σ and the blank symbol B.

b) It is the same as the input alphabet Σ.

c) It contains only the blank symbol B.

d) It only includes the symbols read by the head of the machine.

41. Which of the following is/are not an application of the Turing machine?

a) Language Recognition
b) Computers of functions on non-negative numbers
c) Generating devices
d) None of the mentioned

42. Which of the following cannot be a possibility of a TM while it processes an input?

a) Enters accepting state

b) Enters non-accepting state

c) Enters infinite loop and never halts

d) None of the mentioned

43. Complete the following statement:

Statement: A language is Turing recognizable if and only if ____________

a) an enumerator enumerates it

b) it is finite

c) all of the mentioned

d) none of the mentioned

44. A Turing machine is a

a) real machine

b) abstract machine

c) hypothetical machine

d) more than one option is correct

45. A Turing machine operates over:

a) finite memory tape

b) infinite memory tape

c) depends on the algorithm

d) none of the mentioned

46. Which of the functions are not performed by the turing machine after reading a symbol?

a) writes the symbol

b) moves the tape one cell left/right

c) proceeds with next instruction or halts

d) none of the mentioned

47. The ability for a system of instructions to simulate a Turing Machine is called _________

a) Turing Completeness

b) Simulation

c) Turing Halting

d) None of the mentioned

48. Turing machine can be represented using the following tools:

a) Transition graph

b) Transition table

c) Queue and Input tape

d) All of the mentioned

49. Which of the following is false for an abstract machine?

a) Turing machine

b) theoretical model of computer

c) assumes a discrete time paradigm

d) all of the mentioned

50. The value of n if Turing machine is defined using n-tuples:

a) 6

b) 7

c) 8

d) 5

51. Which of the following a Turing machine does not consist of?

a) input tape

b) head

c) state register

d) none of the mentioned

52. Which of the problems are unsolvable?

a) Halting problem

b) Boolean Satisfiability problem

c) Halting problem & Boolean Satisfiability problem

d) None of the mentioned

53. What is meant by the acceptance of a string by a Turing Machine?

a) The machine deletes the string
b) The machine enters a special "accept" state after processing the string
c) The string is converted into binary
d) The machine rewinds the tape and starts again

54. A What is the purpose of a Turing Machine's read/write head?

a) To move the tape
b) To read and modify symbols on the tape
c) To determine the speed of computation
d) To delete symbols from the tape

55. In a Turing Machine, what does the state transition function dictate?

a) The direction of the tape's movement
b) How the machine moves from one state to another based on the current symbol
c) How many tapes the machine will use
d) Whether the machine halts or continues

56. What is the role of the tape in a Turing Machine?

a) It holds the machine's states
b) It serves as the memory to store input and output
c) It determines the number of transitions
d) It controls the speed of computation

57. Which of the following best defines the term "decidable language"?

a) A language for which there is no Turing Machine
b) A language for which a Turing Machine can decide whether any given string belongs to it

c) A language that can be recognized but not decided
d) A language that is context free

58. What happens when a Turing Machine halts?

a) The tape resets
b) The machine stops and provides an output
c) The machine deletes the tape contents
d) The machine enters an error state

59. Which of the following is an application of Turing Machines?

a) Solving real world problems in polynomial time
b) Modeling algorithms and checking the decidability of problems
c) Increasing computational power of modern computers
d) Simplifying binary code into natural language

60. What does the term "Non deterministic" refer to in Non deterministic Turing Machines?

a) The machine has no predefined set of states
b) The machine can take multiple computational paths simultaneously
c) The machine cannot read symbols from the tape
d) The machine's tape moves in random directions

61. A Turing Machine is non deterministic when:

a) It has a fixed set of instructions for every input
b) It can choose between multiple possible transitions for a single input
c) It can solve NP hard problems in polynomial time
d) It always halts after a finite number of steps

62. In the context of Turing Machines, "reducing" one problem to another typically involves:

a) Finding the shortest computational path
b) Using one problem to help solve another via an algorithm
c) Ignoring computational complexity

d) Simulating both problems in parallel

63. Which feature of a Turing Machine allows it to compute functions recursively?

a) The ability to rewind its tape

b) The use of multiple heads

c) The infinite length of its tape

d) The transition function that loops based on the state

64. What is the main computational advantage of using Non Deterministic Turing Machines (NDTM) over Deterministic Turing Machines (DTM)?

a) NDTMs can always solve problems faster than DTMs

b) NDTMs can solve problems in constant time

c) NDTMs explore multiple computational paths simultaneously

d) NDTMs only use linear time for all computations

65. Which of the following problems is typically undecidable for a Turing Machine?

a) Computing the factorial of an integer

b) Determining whether a Turing Machine halts for a given input

c) Checking the membership of a string in a regular language

d) Sorting an array of integers in linear time

66. Which aspect of a Turing Machine is crucial for accepting context free languages (CFL)?

a) Infinite memory space

b) Non determinism

c) Stack based memory management

d) Recursion and backtracking capabilities

67. Which component of a Turing Machine is responsible for executing state transitions?

a) Input tape

b) Output tape

c) Control unit
d) State Register

68. Which type of Turing Machine is defined by having transitions based on the current state and input symbol?
a) Non Deterministic Turing Machine
b) Deterministic Turing Machine
c) Multi tape Turing Machine
d) Universal Turing Machine

69. A Turing machine has how many states?

a) Finite

b) Infinite

c) May be finite

d) None of the mentioned

70.

70. Halting state of Turing machine are:
a) Start and stop
b) Accept and reject
c) Start and reject
d) Reject and allow

71. An instantaneous description of Turing machine consists of:
a) Present state and input to be processed
b) Present state and entire input to be processed
c) Present input only
d) None of these

72. Why Turing machine is more powerful than Finite automata?
a) Turing machine head movement is continued to one direction.
b) Turing machine head moment is in both directions

c) Turing machine has capability to remember arbitrary long sequence of input string.
d) All are correct

Answers:

1. a) Alan Turing
2. c) Tape
3. c) An infinite memory
4. c) Erase the entire tape
5. d) Tape
6. c) It stops all computations.

7. a) $M = (Q, \Sigma, \Gamma, B, \delta, q0, F)$
8. b) The input alphabet
9. a) Blank symbol
10. b) δ
11. c) The current state and tape symbol
12. b) The machine writes a blank symbol.
13. c) Final state
14. b) q0 (Initial state)
15. b) The machine halts in an accepting state.
16. a) When it reaches a reject
17. d) Recursively enumerable languages
18. b) There is exactly one move from each configuration.
19. b) Decidable problems
20. c) The machine never halts.
21. c) Turing machine
22. a) A Turing machine has a tape, while a finite automaton does not.
23. b) An accepting configuration
24. b) The machine halts only on accepted inputs.
25. b) The problem of deciding whether a Turing machine halts for a given input.
26. c) The halting problem

27. b) A symbol that is part of the tape alphabet but not the input alphabet

28. b) When in state q1 and reading 1, move to state q2, write 0, and move left.

29. c) Every accepting state is a halting state, but not every halting state is an accepting state.

30. b) The machine is in an infinite loop.

31. c) It forms a recursively enumerable language.

32. c) The machine enters an infinite loop.

33. a) Every NTM can be converted to a DTM without changing the language.

34. b) Checking if a Turing machine halts on all inputs.

35. b) The combination of the machine’s current state, the tape contents, and the head position.

36. b) L contains only the strings that lead to a halting state in M.

37. d) There exists an algorithm that can decide membership for all strings in the language.

38. b) The snapshot of the machine's state, tape contents, and head position at any moment.

39. a) By exploring all possible computation paths simultaneously.

40. a) It includes all symbols in the input alphabet Σ and the blank symbol B.

41. D

42. D

43.	A
44.	D
45.	B
46.	D
47.	A
48.	D
49.	D
50.	B
51.	D
52.	C
53.	B
54.	B
55.	B
56.	B
57.	B
58.	B
59.	B
60.	B
61.	B
62.	B
63.	D
64.	C

65.	B
66.	B
67.	C
68.	B
69.	A
70.	B
71.	A
72.	C

Turing Machine (Set 2)

Topics

Turing Machine as a Language Recognizer, Turing Machine as a

Computing Function, Turing Machine as an enumerator of stings of a language, Turing Machine with Multiple Tracks, Turing Machine with Multiple Tapes,Universal Turing Machine

1. **A Turing machine is called a recognizer if it:**
   a) Rejects all inputs
   b) Accepts and halts on strings in the language
   c) Rejects valid strings
   d) Accepts strings but never halts
2. **A Turing machine that accepts an input and halts is said to:**
   a) Recognize the input
   b) Enumerate the input
   c) Reject the input
   d) Simulate the input
3. **What do we call the set of all strings for which a Turing machine halts and accepts?**
   a) Decidable language
   b) Context-free language
   c) Recursive enumerable language
   d) Regular language
4. **Which of the following is true about Turing machines?**
   a) They can only decide regular languages
   b) They cannot recognize context-sensitive languages
   c) They can recognize recursively enumerable languages
   d) They can compute only polynomial-time functions

5. **If a Turing machine halts for every input, then the language it recognizes is:**
   a) Recursively enumerable
   b) Context-free
   c) Context-sensitive
   d) Recursive
6. **A language L is recursively enumerable if:**
   a) A Turing machine halts and accepts for all strings in L
   b) A Turing machine halts on all inputs
   c) A Turing machine enumerates all strings in L
   d) The language is regular
7. **A Turing machine that enumerates a language is one that:**
   a) Accepts and halts for all strings in the language
   b) Rejects all strings not in the language
   c) Generates all strings in the language
   d) Recognizes context-free languages
8. **In what way can a Turing machine enumerate strings of a language?**
   a) In lexicographic order
   b) In random order
   c) By generating an infinite set of strings
   d) By running indefinitely without producing output
9. **Which of the following is false about enumerators?**

a) They can list all valid strings in a language
b) They produce valid strings in finite time
c) They generate all possible strings at once
d) They never generate invalid strings

10. **The output of a Turing machine enumerator for a language L is:**
   a) All possible strings in lexicographic order

b) Only valid strings of L
c) Both valid and invalid strings
d) No output

11. **The process of generating all strings of a language by a Turing machine is called:**
a) Recognition
b) Enumeration
c) Simulation
d) Deciding

12. **What is a Turing machine that always halts for any given input called?**
a) Total Turing machine
b) Universal Turing machine
c) Partial Turing machine
d) Decider Turing machine

13. **A Turing machine that does not halt for some inputs is:**
a) A decider
b) A universal Turing machine
c) A partial Turing machine
d) A total Turing machine

14. **If a Turing machine halts and rejects an input, the string is:**
a) Not in the language
b) In the language
c) Generated by the machine
d) Enumerated by the machine

15. **The class of languages for which a Turing machine always halts is:**
a) Recursive
b) Context-free
c) Recursively enumerable

d) Context-sensitive

16. What is the significance of the Halting Problem in relation to Turing machines?

a) It is undecidable
b) It is decidable by deterministic machines
c) It is solvable using recursive functions
d) It only applies to nondeterministic machines

17. A Universal Turing machine is one that:

a) Can simulate any other Turing machine
b) Can recognize only regular languages
c) Runs faster than deterministic machines
d) Has an infinite number of states

18. A language is said to be undecidable if:

a) There is no Turing machine that halts on all inputs
b) A Turing machine accepts and rejects every input
c) There is no Turing machine that recognizes the language
d) A Turing machine enumerates all strings in the language

19. What is a universal Turing machine capable of doing?

a) Simulating other Turing machines
b) Recognizing only recursively enumerable languages
c) Generating all strings of a language
d) Deciding only context-free languages

20. What happens if a Turing machine runs indefinitely?

a) It fails to recognize the input
b) It halts after a long time
c) It still accepts the input
d) It enumerates strings of the language

21. The output of a computing Turing machine is:

a) The result of a function
b) A list of possible inputs

c) An enumeration of strings
d) The halting state

22. What is the main difference between a recognizer Turing machine and an enumerator Turing machine?

a) Recognizer accepts strings, while an enumerator generates them
b) Recognizer rejects strings, while an enumerator halts on them
c) Recognizer halts on invalid strings, while an enumerator accepts all inputs
d) Both perform the same task

23. A nondeterministic Turing machine:

a) May not halt for some inputs
b) Always halts on every input
c) Enumerates only finite languages
d) Accepts only regular languages

24. A Turing machine can simulate the behavior of:

a) A finite automaton
b) A context-free grammar
c) Any machine that can perform computation
d) Only machines that recognize regular languages

25. The language accepted by a Turing machine is:

a) Infinite
b) Finite
c) Regular
d) Either finite or infinite

26. What is the output of a Turing machine when used as a language enumerator?

a) All strings of the language in some order
b) Only finite strings
c) Both valid and invalid strings

d) Randomly generated strings

27. In the context of a Turing machine as a function computing device, it computes:

a) Partial recursive functions
b) Regular functions
c) Context-free functions
d) Exponential-time functions

28. Turing machine configurations consist of:

a) The current state, tape contents, and head position
b) Only the tape contents
c) Only the current state
d) Only the head position

29. If a Turing machine never halts, it implies that:

a) The string is not part of the language
b) The machine is faulty
c) The machine recognizes infinite strings
d) The machine is generating valid outputs

30. A problem that cannot be solved by any Turing machine is:

a) Decidable
b) Recursive
c) Undecidable
d) Context-sensitive

31. The halting problem proves that there are some functions that:

a) Cannot be computed by a Turing machine
b) Are computable by a finite automaton
c) Can always be decided by a Turing machine
d) Can be computed by regular expressions

32. When used as a computing device, a Turing machine can compute:

a. Only recursive functions
b. Both recursive and partial recursive functions
c. Only regular functions
d. Only context-free functions

33. Which of the following is *not* true for a Turing machine?

a. It can simulate any algorithm
b. It can decide every language
c. It can recognize recursively enumerable languages
d. It can compute functions beyond the capabilities of a finite automaton

34. In a Turing machine, the computation proceeds based on:

a. The input alphabet alone
b. The state and the current symbol being read
c. The tape head's current position
d. The length of the input string

35. A nondeterministic Turing machine differs from a deterministic one by:

a. Allowing multiple possible next moves
b. Rejecting all inputs
c. Recognizing only regular languages
d. Halting after every input

36. Turing machines are classified as total or partial based on:

a. Whether they halt on all inputs
b. The speed of computation
c. The length of the input string
d. The number of states

37. A problem is decidable if:

a. There exists a Turing machine that always halts with a correct answer
b. It can be solved by a nondeterministic machine

c. It requires a recursive enumerable solution
d. The Turing machine halts without an answer

38. If a Turing machine halts and outputs a result, the input:

a. Is accepted by the machine
b. Is rejected
c. Has no relation to the output
d. Can be enumerated

39. An enumerator can be described as:

a. A Turing machine that lists the strings of a language
b. A Turing machine that accepts strings but does not halt
c. A finite automaton that halts on every input
d. A context-sensitive grammar

40. Turing machines can compute functions by:

a. Transitioning between states based on the input
b. Simulating finite automata
c. Generating random strings
d. Computing only simple arithmetic

41. If a Turing machine computes a function for every input, the function is:

a. Recursive
b. Partial recursive
c. Non-recursive
d. Context-free

42. What is a halting configuration in a Turing machine?

a. When the machine enters an accepting or rejecting state
b. When the machine does not read any input
c. When the machine rejects every string
d. When the machine halts but has no output

43. Let two machines be P and Q. The state in which P can simulate Q and Q can simulate P is called:

a) Turing Equivalence
b) State Equivalence
c) Universal Turing Machine
d) None of the mentioned

44. **A turing machine with several tapes in known as:**
a) Multi-tape turing machine
b) Poly-tape turing maching
c) Universal turing machine
d) All of the mentioned

45. **A multitape turing machine is ________ powerful than a single tape turing machine.**
a) more
b) less
c) equal
d) none of the mentioned

46. **In what ratio, more computation time is needed to simulate multitape turing machines using single tape turing machines?**
a) doubly
b) triple
c) quadratically
d) none of the mentioned

47. **Which of the following is a multi tape turing machine?**
a) Post turing Machine
b) Wang-B Machine
c) Oblivious turing Machine
d) All of the mentioned

48. **Are Multitape and Multitrack turing machines same?**
a) Yes
b) No

c) Somewhat yes

d) Cannot tell

49. In a n-track turing machine, _________ head/heads read and write on all tracks simultaneously.

a) one

b) two

c) n

d) infinite

50. Which of the following does not exists?

a) Turing Machine with Multiple heads

b) Turing Machine with infinite tapes

c) Turing machine with two dimensional tapes

d) None of the mentioned

51. Can a multitape turing machine have an infinte number of tapes?

a) Yes

b) No

c) may be

d) may not be

52. Every language accepted by a k-tape TM is _____ by a single-tape TM.

a) accepted

b) not accepted

c) generated

d) not generated

53. Which of the following is/are a basic TM equivalent to?

a) Multitrack TM

b) Multitape TM

c) Non-deterministic TM

d) All of the mentioned

54. **A language L is said to be Turing decidable if:**
    a) recursive
    b) TM recognizes L
    c) TM accepts L
    d) recursive & TM recognizes L
55. **Which of the following statements are false?**
    a) Every recursive language is recursively enumerable
    b) Recursively enumerable language may not be recursive
    c) Recursive languages may not be recursively enumerable
    d) None of the mentioned
56. **Choose the correct option:**
    Statement: If L1 and L2 are recursively enumerable languages over S, then the following is/are recursively enumerable.
    a) L1 U L2
    b) L2 ∩ L2
    c) Both L1 U L2 and L2 ∩ L2
    d) None of the mentioned
57. **If L is a recursive language, L' is:**
    a) Recursive
    b) Recursively Enumerable
    c) Recursive and Recursively Enumerable
    d) None of the mentioned
58. **X is a simple mathematical model of a computer. X has unrestricted and unlimited memory. X is a FA with R/W head. X can have an infinite tape divided into cells, each cell holding one symbol Name X?**
    a) Push Down Automata
    b) Non deterministic Finite Automata
    c) Turing machines
    d) None of the mentioned

59. **Which of the following is/are not an application of turing machine?**
   a) Language Recognization
   b) Computers of functions on non negative numbers
   c) Generating devices
   d) None of the mentioned
60. **Which among the following is not true for 2-way infinte TM?**
   a) tape in both directions
   b) Leftmost square not distinguished
   c) Any computation that can be performed by 2-way infinite tape can also be performed by standard TM.
   d) None of the mentioned
61. **Enumerator is a turing machine with __________**
   a) an output printer
   b) 5 input tapes
   c) a stack
   d) none of the mentioned
62. **For the following language, an enumerator will print:**
   L={$a^n b^n$|n>=0}
   a) $a^n b^n$
   b) {ab, a^2b^2, a^3b^3, …}
   c) {e, ab, a^2b^2, a^3b^3, …}
   d) None of the mentioned
63. **Complete the following statement:**
   Statement : A language is turing recognizable if an only if ___________
   a) an enumerator enumerates it
   b) it is finite
   c) all of the mentioned
   d) none of the mentioned

64. Universal Turing machine (UTM) influenced the concepts of

a) Computability
b) Interpretive implementation of programming language
c) Program and data is in same memory
d) All are correct

65. A universal Turing machine is a

a) Single-tape Turing machine
b) Two-tape Turing machine
c) Reprogrammable Truing machine
d) None of them

66. Which of the following problems is solvable?

a) Determining of a universal Turing machine and some input will halt
b) Determining of an arbitrary Turing machine is an universal Turing machine
c) Determining of a universal Turing machine can be written for fewer than k instructions for some k
d) Writing a universal Turing machine

67. The statement, “A TM can’t solve halting problem” is

a) True
b) False
c) Still an open question
d) All of these

68. What is the significance of the Universal Turing Machine in computability theory?

a. It demonstrates that any computable function can be computed by a single machine
b. It only computes regular languages
c. It is slower than nondeterministic machines
d. It accepts all languages

69. The halting problem shows that:

a. There exist some functions that cannot be computed by any Turing machine
b. All Turing machines halt on every input
c. Turing machines are not powerful enough to compute some problems
d. There are no unsolvable problems

70. What does it mean for a Turing machine to enumerate a language?

a. It accepts all strings in the language
b. It generates all strings in the language
c. It halts on all inputs
d. It rejects all strings not in the language

71. An enumerator Turing machine for a language can be described as a machine that:

a. Halts and rejects for strings not in the language
b. Produces all strings in the language in a finite amount of time
c. Outputs all the strings in the language, possibly with repetition
d. Runs indefinitely and outputs each string in the language exactly once

72. A Turing machine can compute any function if:

a. It has an infinite tape
b. The function is computable
c. It operates in polynomial time
d. The function is regular

73. Which of the following is true for a Turing machine when computing functions?

a. It halts only on valid inputs
b. It halts on all inputs
c. It may not halt for some inputs
d. It accepts only binary inputs

74. The **encoding** of a Turing machine into a Universal Turing Machine involves encoding:

a. Only the input string
b. The tape alphabet and input string
c. The states, transitions, tape alphabet, and input
d. The halting states

75. What is the primary role of the Universal Turing Machine?

a. To compute recursive functions
b. To recognize context-free languages
c. To simulate the behavior of any Turing machine
d. To enumerate recursively enumerable languages

76. In encoding a Turing machine for a UTM, the description of the machine (states, alphabet, and transitions) is usually represented as:

a. A binary string
b. A sequence of arithmetic symbols
c. A graph
d. A regular expression

77. The concept of a Universal Turing Machine is crucial because it:

a. Shows that all Turing machines have the same computational power
b. Proves the existence of an unrecognizable language
c. Demonstrates the existence of NP-complete problems

    d. Simplifies the halting problem

78. **Which of the following is *true* about the encoding process of Turing machines into a UTM?**
    a. Every Turing machine can be encoded using a unique string
    b. Only deterministic Turing machines can be encoded
    c. There are Turing machines that cannot be encoded
    d. The encoding requires non-binary representations
79. **The Universal Turing Machine proves the principle of:**
    a. Automata theory
    b. Computational completeness
    c. Simplicity in computation
    d. Incompleteness of arithmetic
80. **The Universal Turing Machine's tape is divided into two parts when simulating another Turing machine. What does each part store?**
    a. The machine's description and the input string
    b. The input string and output string
    c. The current state and the final state
    d. The transition function and the halting condition
81. **Which of the following is a key implication of the existence of a Universal Turing Machine?**
    a. It implies that some problems are undecidable
    b. It confirms that finite automata can simulate Turing machines
    c. It shows that context-free grammars are universal
    d. It proves that all languages are recursively enumerable
82. **A Universal Turing Machine is an example of which computational concept?**
    a. Parallel computing

b. A reduction technique
c. A machine that simulates any algorithm
d. A restricted computational model

83. In the process of encoding a Turing machine's description for a Universal Turing Machine, which of the following components is typically NOT part of the encoding?

a. Transition function
b. Input alphabet
c. Current state
d. Direction of the head movement

Ans

1. **b) Accepts and halts on strings in the language**
2. **a) Recognize the input**
3. **c) Recursive enumerable language**
4. **c) They can recognize recursively enumerable languages**
5. **d) Recursive**
6. **a) A Turing machine halts and accepts for all strings in L**
7. **c) Generates all strings in the language**
8. **a) In lexicographic order**
9. **c) They generate all possible strings at once**
10. **b) Only valid strings of L**
11. **b) Enumeration**
12. **a) Total Turing machine**
13. **c) A partial Turing machine**
14. **a) Not in the language**
15. **a) Recursive**
16. **a) It is undecidable**
17. **a) Can simulate any other Turing machine**
18. **a) There is no Turing machine that halts on all inputs**
19. **a) Simulating other Turing machines**
20. **a) It fails to recognize the input**
21. **a) The result of a function**

22. a) Recognizer accepts strings, while an enumerator generates them
23. a) May not halt for some inputs
24. C
25. D
26. a) All strings of the language in some order
27. a) Partial recursive functions
28. a) The current state, tape contents, and head position
29. a) The string is not part of the language
30. c) Undecidable
31. a) Cannot be computed by a Turing machine
32. b) Both recursive and partial recursive functions
33. b) It can decide every language
34. b) The state and the current symbol being read
35. a) Allowing multiple possible next moves
36. a) Whether they halt on all inputs
37. a) There exists a Turing machine that always halts with a correct answer
38. a) Is accepted by the machine
39. a) A Turing machine that lists the strings of a language
40. a) Transitioning between states based on the input
41. a) Recursive
42. a) When the machine enters an accepting or rejecting state
43. a
44. a
45. a
46. c
47. c
48. a
49. a

50. d
51. b
52. a
53. d
54. d
55. c
56. c
57. c
58. c
59. d
60. d
61. a
62. b
63. a
64. d
65. c
66. d
67. a
68. a
69. a
70. b
71. d
72. b
73. c
74. c
75. c
76. a) A binary string
77. a) Shows that all Turing machines have the same computational power

78. a) Every Turing machine can be encoded using a unique string

79. b) Computational completeness

80. a) The machine’s description and the input string

81. a) It implies that some problems are undecidable

82. c) A machine that simulates any algorithm

83. b

Turing Machine(set 3)

Topics

Curch Turing Thesis, Computational Complexity, Time and Space complexity of A Turing Machine, Intractability, Reducibility.

1. **What does the Church-Turing thesis state about computation?**
   a) All functions that are computable by any physical machine are computable by a Turing machine.
   b) Only recursive functions are computable.
   c) All physical problems can be solved in polynomial time.
   d) Turing machines can compute non-recursive functions.
2. **Which two formalizations of computation are central to the Church-Turing thesis?**
   a) Turing machines and Lambda calculus
   b) Finite automata and context-free grammars
   c) Non-deterministic automata and regular expressions
   d) Pushdown automata and context-sensitive grammars
3. **What is the implication of the Church-Turing thesis for real-world computers?**
   a) Modern computers can solve any problem in polynomial time.
   b) Any function that can be calculated by an algorithm can be computed by a Turing machine.
   c) Real-world computers are more powerful than Turing machines.
   d) Modern computers can solve non-computable problems.
4. **Which of the following is NOT an equivalent model of computation according to the Church-Turing thesis?**
   a) Turing machine
   b) Lambda calculus
   c) Finite automata
   d) Recursive functions
5. **Which of the following best describes the power of a Universal Turing Machine under the Church-Turing thesis?**
   a) It can solve any problem that any digital computer can solve.
   b) It can solve all NP-complete problems in polynomial time.

c) It is weaker than real-world computers.
d) It can simulate physical processes with infinite precision.

6. The Church-Turing thesis originally focused on:

a) The efficiency of algorithms
b) The limits of mechanical computation
c) The classification of languages
d) Cryptography

7. Which of the following statements would contradict the Church-Turing thesis?

a) There exists a machine that can compute functions faster than a Turing machine.
b) There exists a function that cannot be computed by a Turing machine.
c) Quantum computers can solve NP-hard problems in polynomial time.
d) All computable functions are Turing computable.

8. According to the Church-Turing thesis, a function that is not Turing-computable is:

a) Recursive
b) Non-recursive
c) Decidable
d) Intractable

9. The Church-Turing thesis asserts equivalence between:

a) Recursive functions and context-free languages
b) Recursive functions and Turing machines
c) NP problems and context-sensitive languages
d) Regular expressions and Turing machines

10. What does the Church-Turing thesis say about problems that are physically computable?

a) They can only be solved in exponential time.
b) They must be expressible in terms of recursive functions.
c) They are always solvable by a digital computer.
d) They are always solvable in polynomial time.

11. What does computational complexity primarily study?

a) The efficiency of algorithms
b) The power of Turing machines
c) The limitations of physical machines
d) The classification of recursive functions

12. Which of the following time complexity classes is considered to represent "feasible" computations?

a) O(n^n)
b) O(2^n)
c) O(n log n)
d) O(n!)

13. Which class of problems is believed to have solutions that cannot be computed in polynomial time?

a) NP-complete
b) Regular
c) Context-sensitive
d) Recursive

14. Which of the following is TRUE about P and NP?

a) P problems are a subset of NP problems.
b) NP problems can all be solved in polynomial time.
c) NP problems require non-deterministic Turing machines.
d) All P problems are undecidable.

15. What is the relationship between NP-complete problems and NP problems?

a) All NP-complete problems are in NP, but not all NP problems are NP-complete.
b) All NP problems are NP-complete.
c) NP-complete problems are solvable in constant time.
d) NP problems are always harder than NP-complete problems.

16. What is a polynomial-time reduction?

a) A process to transform one problem into another in polynomial time.
b) A method to solve NP problems in polynomial time.
c) A technique for compressing data efficiently.
d) A way to solve problems using finite automata.

17. Which of the following complexity classes represents the set of problems solvable in polynomial time by a deterministic Turing machine?

a) NP
b) P
c) PSPACE
d) EXPTIME

18. Which complexity class includes problems that can be solved in polynomial time by a non-deterministic Turing machine?

a) P
b) NP
c) PSPACE
d) EXP

19. Which of the following problems is NOT NP-complete?

a) Hamiltonian Path Problem
b) Traveling Salesman Problem (decision version)
c) Graph Isomorphism Problem
d) 3-SAT Problem

20. What is the relationship between P and NP in computational complexity theory?

a) P is a subset of NP.

b) NP is a subset of P.
c) P and NP are disjoint classes.
d) P contains NP-complete problems.

21. Which of the following best describes NP-complete problems?
a) Problems that can be solved in polynomial time.
b) Problems that are as hard as the hardest problems in NP.
c) Problems that can be solved by a finite automaton.
d) Problems that belong to the P complexity class.

22. The P vs NP problem asks whether:
a) All NP problems can be solved in polynomial time.
b) All P problems can be solved by non-deterministic machines.
c) NP-complete problems can be solved using recursive functions.
d) All Turing machines are equivalent in computational power.

23. The class P refers to problems that can be solved in:
a) Exponential time
b) Polynomial time
c) Constant time
d) Logarithmic time

24. What is the complexity class P?
a) Problems solvable in polynomial time
b) Problems solvable in exponential time
c) Problems verifiable in polynomial time
d) Problems that are undecidable

25. Which of the following statements is true?
a) Every problem in NP is also in P.
b) Every problem in P is also in NP.
c) Every problem in NP is NP-complete.
d) All problems in P are NP-hard.

26. What is the primary difference between P and NP?
a) P refers to problems that can be solved in polynomial time, while NP refers to problems that can be solved using non-determinism.
b) P refers to problems verifiable in polynomial time, while NP refers to problems solvable in constant time.
c) P problems are those solvable in exponential time, while NP problems are undecidable.
d) P refers to problems solvable using quantum algorithms, while NP refers to classical algorithms.

27. What is the significance of NP-complete problems?
a) Solving one NP-complete problem efficiently implies that P = NP.
b) NP-complete problems cannot be verified in polynomial time.
c) All NP-complete problems can be solved in logarithmic space.
d) NP-complete problems are not solvable by Turing machines.

28. The P versus NP problem asks:

a) Whether every problem that can be verified in polynomial time can also be solved in polynomial time.
b) Whether all exponential problems can be solved in polynomial time.
c) Whether all polynomial problems are also undecidable.
d) Whether NP-hard problems are equivalent to recursive functions.

29. Time complexity of a Turing machine measures:
a) The number of transitions made before halting.
b) The amount of memory used during computation.
c) The number of tapes used by the machine.
d) The number of characters read in the input tape.

30. Space complexity of a Turing machine refers to:
a) The amount of time the machine takes to halt.
b) The amount of memory (or tape) the machine uses during computation.
c) The number of inputs it can process simultaneously.
d) The length of the tape.

31. If a problem has a time complexity of O(2^n), it is:
a) Solvable in constant time.
b) Solvable in polynomial time.
c) Solvable in exponential time.
d) Solvable in logarithmic time.

32. Which complexity class consists of problems solvable in polynomial time by a deterministic Turing machine?
a) NP
b) PSPACE
c) P
d) EXPTIME

33. If a Turing machine uses a polynomial amount of space, it belongs to the class:
a) P
b) NP
c) PSPACE
d) EXP

34. Which of the following complexities represents a problem that is tractable?
a) O(n!)
b) O(2^n)
c) O(n^2)
d) O(3^n)

35. What is the relationship between time and space complexity for a Turing machine?
a) Time complexity is always greater than space complexity.
b) Space complexity can sometimes be smaller than time complexity.
c) Time and space complexities are always equal.
d) Space complexity is always larger than time complexity.

36. Which problem has the time complexity O(n log n)?
a) Binary search

b) QuickSort
c) Factorial computation
d) Graph traversal

37. For a Turing machine, the space complexity O(n^2) means:

a) The machine halts in quadratic time.
b) The machine uses quadratic space to solve the problem.
c) The problem cannot be solved in polynomial space.
d) The input grows exponentially.

38. Which of the following refers to the best case time complexity for an algorithm?

a) O(n)
b) Ω(n)
c) Θ(n)
d) δ(n)

39. What does the time complexity class NP represent?

a) Problems solvable by a deterministic Turing machine in polynomial time.
b) Problems verifiable by a deterministic Turing machine in polynomial time.
c) Problems solvable in exponential time.
d) Problems that cannot be solved.

40. Which of the following refers to the space complexity of a deterministic Turing machine?

a) The maximum number of configurations of the machine
b) The number of transitions made
c) The amount of tape cells used during computation
d) The size of the input

41. Which of the following best defines a tractable problem?

a) A problem solvable in polynomial time
b) A problem that requires an infinite amount of time
c) A problem solvable in constant time
d) A problem that cannot be solved by any algorithm

42. Which of the following best defines intractability?

a) Problems solvable in polynomial time
b) Problems that are NP-complete
c) Problems that require more than polynomial time to solve
d) Problems verifiable in polynomial time

43. A problem is considered intractable if:

a) It can be solved in polynomial time.
b) It cannot be solved in polynomial time.
c) It requires a constant amount of space to solve.
d) It is undecidable.

44. Which of the following statements about NP-hard problems is true?

a) NP-hard problems are easier than NP-complete problems.
b) NP-hard problems are at least as hard as NP-complete problems.
c) NP-hard problems can be solved in polynomial time.
d) NP-hard problems cannot be verified.

45. Which of the following problems is intractable?

a) The sorting problem
b) The halting problem
c) The graph isomorphism problem
d) The Boolean satisfiability problem

46. What does reducibility mean in computational complexity?

a) The process of finding the most efficient algorithm for a problem.
b) Converting one problem into another problem such that solving the second problem allows you to solve the first.
c) Reducing the number of states in a finite automaton.
d) Simplifying a recursive function.

47. What is the significance of reducibility in the context of NP-completeness?

a) If a problem is reducible to an NP-complete problem, it is in P.
b) If a problem is reducible from an NP-complete problem, it is NP-complete.
c) Reducibility shows that NP problems can be solved in logarithmic space.
d) Reducibility cannot be used to prove NP-completeness.

48. A problem A is polynomial-time reducible to problem B if:

a) A can be solved in polynomial time given a solution to B.
b) B can be solved in polynomial time given a solution to A.
c) Both A and B are NP-complete.
d) A and B are in different complexity classes.

49. If problem A is reducible to problem B in polynomial time, and B is in P, what can we say about A?

a) A is NP-hard.
b) A is also in P.
c) A is undecidable.
d) A is NP-complete.

50. What type of reduction is typically used to show that a problem is NP-complete?

a) Turing reduction
b) Polynomial-time reduction
c) Logarithmic-space reduction
d) Constant-time reduction

51. In complexity theory, what does it mean for one problem to be reducible to another?

a) The first problem is easier than the second.
b) Solving the second problem provides a solution to the first.
c) The second problem can be solved using a deterministic algorithm.
d) The first problem can be solved using a non-deterministic algorithm.

52. What does it mean if a problem is reducible to the halting problem?

a) The problem is solvable in polynomial time.
b) The problem is undecidable.
c) The problem belongs to the class P.
d) The problem is NP-complete.

53. Which of the following types of reduction preserves decidability?

a. Log-space reduction
b. Polynomial-time reduction
c. Turing reduction
d. Constant-time reduction

54. Which of the following is used to demonstrate that a problem is NP-hard?

a. Proving it is reducible from an NP-complete problem
b. Proving it is solvable in polynomial time
c. Proving it is unsolvable
d. Proving it is reducible from a P problem

55. Which problem is undecidable?

a. Boolean satisfiability
b. Halting problem
c. Graph isomorphism
d. Sorting problem

56. Which of the following reductions are typically used to show undecidability?

a. Turing reductions
b. Polynomial-time reductions
c. Constant-space reductions
d. NP reductions

57. If problem A can be reduced to problem B, which of the following is true?

a. Solving B implies solving A.
b. Solving A implies solving B.
c. B is harder than A.
d. A is undecidable if B is in P.

58. If a problem is reducible to the halting problem, it is:

a. NP-complete
b. Decidable
c. Intractable
d. Undecidable

Ans

1. **a**
2. **a**
3. **b**
4. **c**
5. **a**
6. **b**
7. **b**
8. **b**
9. **b**
10. **b**
11. **a**
12. **c**
13. **a**
14. **a**
15. **a**
16. **a**
17. **b**
18. **b**
19. **c**
20. **a**
21. **b**
22. **a**
23. **b**
24. **a**
25. **b**
26. **a**
27. **a**
28. **a**
29. **a**
30. **a**
31. **c**

32.c
33.c
34.c
35.b
36.b
37.b
38.b
39.b
40.c
41.a
42.c
43.a
44.b
45.b
46.b
47.b
48.a
49.b
50.b
51.b
52.b
53.a
54.a
55.b
56.a
57.a
58.d