Turing Machine

Turing Machine(TM)

- A Turing machine is a finite automaton that can read, write, and erase symbols on an infinitely long tape.
- The tape is divided into squares, and each square contains a symbol
- The Turing machine can only read one symbol at a time, and it uses a set of rules (the transition function) to determine its next action based on the current state and the symbol it is reading.
- The Turing machine's behavior is determined by a finite state machine,

A TM is expressed as a 7-tuple (Q, T, B, ∑, δ, q0, F) where:

Q is a finite set of states

T is the tape alphabet (symbols which can be written on Tape)

B is blank symbol (every cell is filled with B except input alphabet initially)

∑ is the input alphabet (symbols which are part of input alphabet)

δ is a transition function which maps $Q \times T \rightarrow Q \times T \times \{L,R\}$. Depending on its present state and present tape alphabet (pointed by head pointer), it will move to new state, change the tape symbol (may or may not) and move head pointer to either left or right.

q0 is the initial state

F is the set of final states. If any state of F is reached, input string is accepted.

Instantaneous description of TM

INSTANTANEOUS DESCRIPTION (ID) OF A TURING MACHINE ™

Instantaneous Description or ID :

$X_1 X_2 \ldots X_{i-1}\ q\ X_i X_{i+1} \ldots X_n$

Means:

q is the current state Tape head is pointing to X_i

$X_1 X_2 \ldots X_{i-1} X_i X_{i+1} \ldots X_n$ are the current tape symbols

- $\delta(q, X_i) = (p, Y, R)$ is same as:
  $X_1 X_2 \ldots X_{i-1}\ q\ X_i X_{i+1} \ldots X_n \vdash X_1 X_2 \ldots X_{i-1}\ Y\ p\ X_{i+1} \ldots X_n$
- $\delta(q\ X_i) = (p, Y, L)$ is same as:
  $X_1 X_2 \ldots X_{i-1}\ q\ X_i X_{i+1} \ldots X_n \vdash X_1 X_2 \ldots p X_{i-1} Y X_{i+1} \ldots X_n$

Language accepted by Turing machine

- Let TM = (Q, T, B, ∑, δ, q0, F) be a TM.
- L(TM)= {w in ∑* |q0w ⊢* apb } where a and b are tape strings and p be a final state.
- TM accepts all the language even though they are recursively enumerable
- TM also accepts the computable functions, such as addition, multiplication, subtraction, division, power function, and many more.

TM as language recognizer:

- A TM recognizes a language L iff it accepts every x ∈ L and does not accept any x ∉L.
- A TM decides a language L iff it accepts every x ∈ L and it rejects x ∉L.

Construct a TM that accepts a language L={0^n 1^n |n>0}

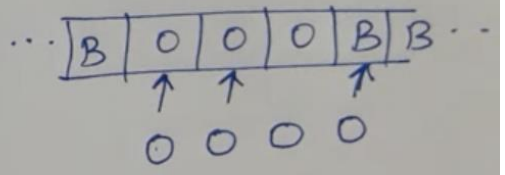
L= {01,0011,000111,…..}

Let 000111 = 000->xxx; 111->yyy

Turing Machine as a computing function

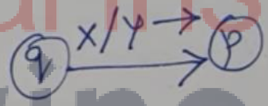
Construct a TM for the function

$f(x) = x+1$

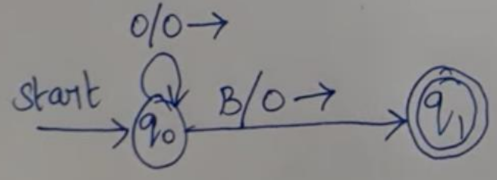


State	Tape symbols: 0	Tape symbols: B
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$
q_1	—	—

$\delta(q, X) = (p, Y, R)$



TM, $M = (\{q_0, q_1\}, \{0\}, \{0, B\}, \delta, q_0, B, \{q_1\})$



$q_0 000 \vdash 0 q_0 00B$

$\vdash 00 q_0 0B$

$\vdash 000 q_0 B$

$\vdash 0000 q_1 B$

Turing Machine as a enumerator of strings of a language

- The machine enumerates i.e. it lists out or prints the strings in a language.
- The tape is initially empty i.e. no input.
- The printer produces a series of strings.
- It may halt or loop
- The language may be infinite.
- It may print duplicates.
- A language is Turing recognizable iff some enumerator numerates it

Variation of Turing Machine

1. Multiple track Turing Machine: A k-track Turing machine(for some k>0) has k-tracks and one R/W head that reads and writes all of them one by one.

- A k-track Turing Machine can be simulated by a single track Turing machine

2. Two-way infinite Tape Turing Machine:

- Infinite tape of two-way infinite tape Turing machine is unbounded in both directions left and right.
- Two-way infinite tape Turing machine can be simulated by one-way infinite Turing machine(standard Turing machine).

3. Multi-tape Turing Machine:

- It has multiple tapes and is controlled by a single head.
- The Multi-tape Turing machine is different from k-track Turing machine but expressive power is the same.
- Multi-tape Turing machine can be simulated by single-tape Turing machine.

4. Multi-tape Multi-head Turing Machine:

- The multi-tape Turing machine has multiple tapes and multiple heads

- Each tape is controlled by a separate head
- Multi-Tape Multi-head Turing machine can be simulated by a standard Turing machine.

5. Multi-dimensional Tape Turing Machine:

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

- Multi dimensional tape Turing machine can be simulated by one-dimensional Turing machine

6. Multi-head Turing Machine:

- A multi-head Turing machine contains two or more heads to read the symbols on the same tape.
- In one step all the heads sense the scanned symbols and move or write independently.
- Multi-head Turing machine can be simulated by a single head Turing machine.

7. Non-deterministic Turing Machine:

- A non-deterministic Turing machine has a single, one-way infinite tape.
- For a given state and input symbol has at least one choice to move (finite number of choices for the next move), each choice has several choices of the path that it might follow for a given input string.
- A non-deterministic Turing machine is equivalent to the deterministic Turing machine.

Church’s Turing thesis

- that can be stated as: “The assumption that the intuitive notion of computable functions can be identified with partial recursive functions.”
- Or in simple words we can say that “Every computation that can be carried out in the real world can be effectively performed by a Turing Machine.”
- this hypothesis cannot be proved. The recursive functions can be computable after taking following assumptions:

1. Each and every function must be computable.
2. Let ‘F’ be the computable function and after performing some elementary operations to ‘F’, it will transform a new function ‘G’ then this function ‘G’ automatically becomes the computable function.
3. If any functions that follow above two assumptions must be states as computable function.

Universal Turing Machine:

- Universal Turing Machine is like a single Turing Machine that has a solution to all problem that is computable.
- Programmable Turing Machine is called Universal Turing Machine.
- The transition function is $Q \times T \rightarrow Q \times T \times \{L, R\}$, where Q is a finite set of states, T is the tape of the alphabet.
- Universal Turing Machine is a subset of all the Turing Machines.
- Universal Turing Machine contains Turing Machine description as input along with an input string, runs the Turing Machine on the input and returns the result.

Time and Space Complexity of a Turing Machine:

- For a Turing machine, the time complexity refers to the measure of the number of times the tape moves when the machine is initialized for some input symbols and the space complexity is the number of cells of the tape written.
- Time complexity all reasonable functions –
- **T(n) = O(n log n)**
- TM's space complexity –
- **S(n) = O(n)**

Intractability:

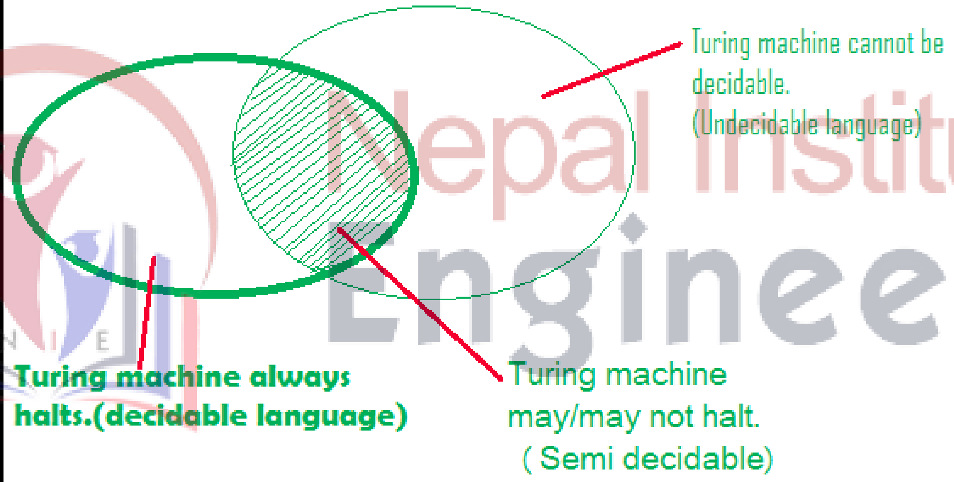
- a problem is defined as tractable if an algorithm exists to solve it that can be executed in polynomial time (or less). Problems that grow exponentially or worse are said to be intractable.
- Examples of intractable problems are:
- Traveling Salesman Problem.
- Satisfiability.

Reducibility

- Reducibility refers to the act of using the solution to one problem as a means to solve another. For example, the problem of finding the area of a rectangle reduces to the problem of multiplying the length of the rectangle by the width.
- Language A is reducible to language B (represented as A≤B) if there exists a function f which will convert strings in A to strings in B as:
- w ε A <=> f(w) ε B
- Theorem 1: If A≤B and B is decidable then A is also decidable. Theorem 2: If A≤B and A is undecidable then B is also undecidable.

- **Decidable Problems**
  A problem is decidable if we can construct a Turing machine which will halt in finite amount of time for every input and give answer as ‘yes’ or ‘no’. A decidable problem has an algorithm to determine the answer for a given input. h of the rectangle.
- Eg: **Equivalence of two regular languages**

- **Undecidable Problems**
  A problem is undecidable if there is no Turing machine which will always halt in finite amount of time to give answer as ‘yes’ or ‘no’. An undecidable problem has no algorithm to determine the answer for a given input.
- Eg: **Ambiguity of context-free languages**, **Equivalence of two context-free languages**



Relation between decidablity,semidecidablity and undecidablity

Question: Which of the following is/are undecidable?

1. G is a CFG. Is $L(G)=\phi$?
2. G is a CFG. Is $L(G)=\Sigma^*$?
3. M is a Turing machine. Is L(M) regular?
4. A is a DFA and N is an NFA. Is L(A)=L(N)?

A. 3 only
B. 3 and 4 only
C. 1, 2 and 3 only
D. 2 and 3 only

Explanation:

- Option 1 is whether a CFG is empty or not, this problem is decidable.
- Option 2 is whether a CFG will generate all possible strings (everything or completeness of CFG), this problem is undecidable.
- Option 3 is whether language generated by TM is regular is undecidable.
- Option 4 is whether language generated by DFA and NFA are same is decidable. So option D is correct.

Question: Which of the following problems are decidable?

1. Does a given program ever produce an output?
2. If L is context free language then L' is also context free?
3. If L is regular language then L' is also regular?
4. If L is recursive language then L' is also recursive?

A. 1,2,3,4
B. 1,2
C. 2,3,4
D. 3,4

Explanation:

- As regular and recursive languages are closed under complementation, option 3 and 4 are decidable problems.
- Context free languages are not closed under complementation, option 2 is undecidable.
- Option 1 is also undecidable as there is no TM to determine whether a given program will produce an output. **So, option D is correct.**

Question: Consider three decision problems P1, P2 and P3. It is known that P1 is decidable and P2 is undecidable. Which one of the following is TRUE?

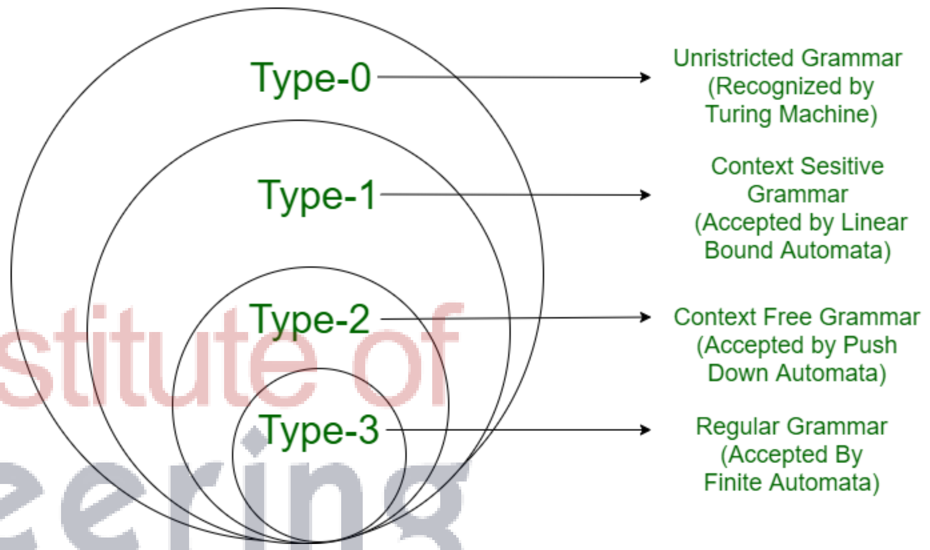
A. P3 is undecidable if P2 is reducible to P3
B. P3 is decidable if P3 is reducible to P2's complement
C. P3 is undecidable if P3 is reducible to P2
D. P3 is decidable if P1 is reducible to P3

Explanation:

- Option A says $P2 \le P3$. According to theorem 2 discussed, if P2 is undecidable then P3 is undecidable. It is given that P2 is undecidable, so P3 will also be undecidable. So option **(A) is correct**.
- Option C says $P3 \le P2$. According to theorem 2 discussed, if P3 is undecidable then P2 is undecidable. But it is not given in question about undecidability of P3. So option **(C) is not correct.**
- Option D says $P1 \le P3$. According to theorem 1 discussed, if P3 is decidable then P1 is also decidable. But it is not given in question about decidability of P3. So option **(D) is not correct**.
- Option (B) says $P3 \le P2'$. According to theorem 2 discussed, if P3 is undecidable then P2' is undecidable. But it is not given in question about undecidability of P3. So option **(B) is not correct**.

Chomsky Hierarchy:

- **Type 0: Unrestricted Grammar:** Sab --> ba ; A --> S; [VT->TT or V->V]
- **Type 1:** Context-Sensitive Grammar: S --> AB; AB --> abc; B --> b ; [V->VV;VV->TT]
- **Type 2: Context-Free Grammar:** S --> AB A --> a ; B --> b; [V->VV; V->T]
- **Type 3: Regular Grammar:** S --> a; [v->VT; V->T



Turing Machine

1. The language recognized by Turing machine is:

(A) Context free language
(B) Context sensitive language
(C) Recursively enumerable language
(D) Regular language
Ans: c
Explanation: A language is recursively enumerable (generated by Type-0 grammar) if it is accepted by a Turing machine. A TM decides a language if it accepts it and enters into a rejecting state for any input not in the language. A language is recursive if it is decided by a Turing machine.

2. Turing machine consist of:
(A) Input tape
(B) Blank symbol
(C) Tape head
(D) All of these
Ans: d
Explanation: TM consist of number of states, input alphabet, tape head, blank symbol,

3. Turing machine is more powerful than:
(A) Finite automata
(B) Push down automata
(C) Both (a) and (b)
(D) None of these
Ans: c
Explanation: Turing machines are more powerful because they can remember more than FA because FA don't have memory to remember. Turing machines are indeed more powerful than regular PDAs because it can simulate the TM's tape using two stacks: in the left stack everything is stored which is left from the head on the Turing-tape, while the symbol under the head and everything right from the head is stored in the other stack.

4. Alan Turing introduced Turing machine late in
(A) 1936
(B) 1938
(C) 1940
(D) 1935
Ans: b
Explanation: The Turing machine was invented in 1938 by Alan Turing, who called it an "a machine"

5. In definition of TM $T= (Q, \Sigma, \Gamma, q0, \delta)$ what Γ represents?
(A) Tape alphabets
(B) Input symbols
(C) Transition function
(D) Initial state

Ans: a
Explanation: Γ represents a tape alphabet which is nothing but output symbol
6. In one move the Turing machine:
(A) May change its state
(B) May write a symbol on the cell being scanned.
(C) May move the head one position left or right
(D) All of the above
Ans: d
Explanation: TM tape head can move the head position left or right, change its state and while moving input symbols on cell being scanned (Read)

7. Halting state of Turing machine are:
(A) Start and stop
(B) Accept and reject
(C) Start and reject
(D) Reject and allow
Ans: b
Explanation: In TM , machine can halt on if sting is accepted with correct output or rejected with wrong output

8. Which of the following is an extension to the basic model of Turing machine: **(A)** Multi tape Turing machine
(B) Multi head Turing machine
(C) Nondeterministic Turing machine
(D) All of the above
Ans: d
Explanation: Extension of TM are Multi tape Turing machine,Multi head Turing machine, Nondeterministic Turing machine, Deterministic Turing machine, Multi-stack TM

9. 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
Ans: c
Explanation: Turing machine has capability to remember arbitrary long sequence of input string and In FA there is no memory for storage

10. A pushdown automata behaves like a Turing machine, when it has number of auxiliary memory.
(A) 0
(B) 2
(C) 2 or more
(D) Both b and c
Ans: c
Explanation: A pushdown automaton behaves like a turning machine when the number of auxiliary memory is 2. The machines are actually represented using the 3r elements of the alphabet. In this term, the letter is the number of registers

11. 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
Ans: d
Explanation: all above 3 options are suitable for UTM

12. A universal Turing machine is a
(A) Single tape Turing machine
(B) Two-tape Turing machine
(C) Reprogrammable Truing machine
(D) None of them
Ans: c
Explanation: UTM is a reprogrammable TM, other 2 are types of TM

13. A Turing machine that is able to simulate other Turing machines:
(A) Nested Turing machines
(B) Multi tap Turing machine
(C) Universal Turing machines
(D) None of these
Ans: c
Explanation: A Turing machine that is able to simulate any other Turing machine is called a universal Turing machine (UTM, or simply a universal machine).

14. A Turing machine with several tapes in known as:
(A) Multi-tape Turing machine
(B) Universal Turing machine
(C) Poly-tape Turing machine
(D) All of the mentioned
Ans: a
Explanation:A multitape turing machine is an ordinary turing machine with multiple tapes. Each tape has its own head to control the read and write.

15. A multitape turing machine is ________ powerful than a single tape turing machine. (A)more
(B)less
(C)equal
(D) none of the mentioned
Ans:a
Explanation: The multitape turing machine model seems much powerful than the single tape model, but any multi tape machine, no matter how many tapes, can be simulated by single taped TM.

16. Which of the following is true for two stack turing machines?
(a) one read only input
(b) two storage tapes

(c) Both (a) and (b)
(d) None of the mentioned
Answ: c
Explanation: Two-stack Turing machines have a read-only input and two storage tapes. If a head moves left on either tape a blank is printed on that tape, but one symbol from a “library” can be printed.

17. A deterministic turing machine is:
(a) ambiguous turing machine
(b) unambiguous turing machine
(c) non-deterministic
(d) none of the mentioned
Ans: b
Explanation: A deterministic turing machine is unambiguous and for every input, there is exactly one operation possible. It is a subset of non-deterministic Turing machines.

18. Which of the following is true about Turing’s a-machine?
a) a stands for automatic
b) left ended, right end-infinite
c) finite number of tape symbols were allowed
d) all of the mentioned
Ansr: d
Explanation: Turings a- machine or automatic machine was left ended,right end infinite.Any of finite number of tape symbols were allowed and the 5 tuples were not in order.

19. State true or false:
Statement: Multitape turing machine have multi tapes where each tape is accessed with one head.
a) true
b) false
Ans: b
Explanation: Multitape turing machines do have multiple tapes but they they are accessed by separate heads.

20. Are Multitape and Multitrack turing machines same?
a) Yes
b) No
c) Somewhat yes
d) Cannot tell
Ans: a
Explanation: Multitrack turing machines are special types of Multitape turing machines. In a standard n-tape Turing machine, n heads move independently along n-tracks.

21. 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

Ans: d
Explanation: All of the mentioned are one or the other kind of Turing machines in existence.

22. 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
Ans: d
Explanation: Tms can be used as both: language recognizers/Computers. TMs are like universal computing machines with universal storage.

23. 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
Answer: d
Explanation: A turing machine can have many applications like : Enumerator (A turing machine with an output printer), function computer, etc.

24. 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
Ans: d
Explanation: All of the mentioned are correct statements for a two way infinite tape turing machine. Theorems say the power of such a machine is in no way superior than a standard turing machine.

25. Which of the functions can a turing machine not perform?
a) Copying a string
b) Deleting a symbol
c) Accepting a pal
d) Inserting a symbol
Answer: d
Explanation: Different turing machines exist for operations like copying a string, deleting a symbol, inserting a symbol and accepting palindromes.

26. If T1 and T2 are two turing machines. The composite can be represented using the expression:
a) T1T2
b) T1 U T2
c) T1 X T2
d) None of the mentioned

Answer: a
Explanation: If T1 and T2 are TMs, with disjoint sets of non halting states and transition function d1 and d2, respectively, we write T1T2 to denote this composite TM.

27. A turing machine that is able to simulate other turing machines:
a) Nested Turing machines
b) Universal Turing machine
c) Counter machine
d) None of the mentioned
Answer: b
Explanation: A more mathematically oriented definition with the same universal nature was introduced by church and turing together called the Church-Turing thesis(formal theory of computation).

28. Which of the problems are unsolvable?
a) Halting problem
b) Boolean Satisfiability problem
c) Both (a) and (b)
d) None of the mentioned
Answer: c
Explanation: Alan turing proved in 1936 that a general algorithm to solve the halting problem for all possible program-input pairs cannot exist.

29. Which of the following a turing machine does not consist of?
a) input tape
b) head
c) state register
d) none of the mentioned
Answer: d
Explanation: A state register is one which stores the state of the turing machine, one of the finitely many. Among these is the special start state with which the state register is initialized.

30. The value of n if turing machine is defined using n-tuples:
a) 6
b) 7
c) 8
d) 5
Answer: b
Explanation:
The 7-tuple definition of turing machine: (**Q, Σ,Γ, q0, δ,** B, F)
where Q= The finite set of states of finite control
Σ= The finite set of input symbols
Γ= The complete set of tape symbols
δ= The transition function
q0= The start state, a member of Q, in which the finite control is found initially. B= The blank symbol
F= The set of final or accepting states, a subset of Q.

- https://test.yuvayana.org/computer-science/turing-machine-mcq-test-with-answers-1/