Variants 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.

Universal Turing Machine

The Turing Machine (TM) is the machine level equivalent to a digital computer.

It was suggested by the mathematician Turing in the year 1930 and has become the most widely used model of computation in computability and complexity theory.

The model consists of an input and output. The input is given in binary format form on to the machine’s tape and the output consists of the contents of the tape when the machine halts

The problem with the Turing machine is that a different machine must be constructed for every new computation to be performed for every input output relation.

This is the reason the Universal Turing machine was introduced which along with input on the tape takes the description of a machine M.

The Universal Turing machine can go on then to simulate M on the rest of the content of the input tape.

A Universal Turing machine can thus simulate any other machine.

The idea of connecting multiple Turing machine gave an idea to Turing:

- Can a Universal machine be created that can 'simulate' other machines?
- This machine is called as Universal Turing Machine

This machine would have three bits of information for the machine it is simulating:

- A basic description of the machine.
- The contents of machine tape.
- The internal state of the machine.

The Universal machine would simulate the machine by looking at the input on the tape and the state of the machine.

It would control the machine by changing its state based on the input. This leads to the idea of a "computer running another computer".

It would control the machine by changing its state based on the input. This leads to the idea of a "computer running another computer".

The schematic diagram of the Universal Turing Machine is as follows:

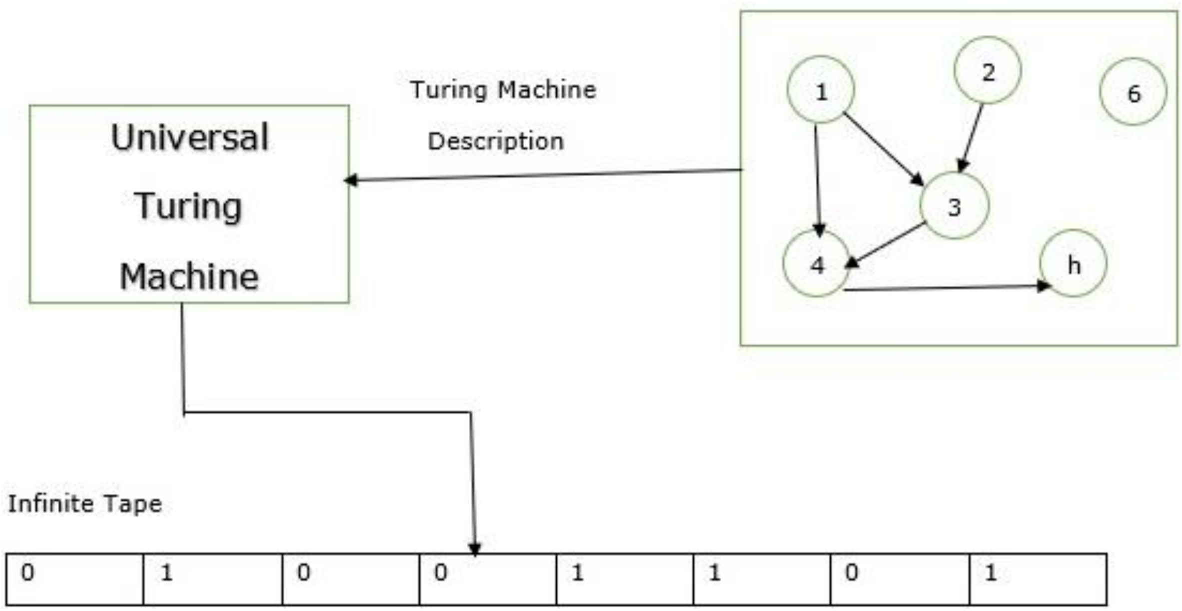

Fig. 5.:

Application of Turing Machine:

- For solving any recursively enumerable problem.
- For understanding complexity theory.
- For implementation of neural networks.

- For implementation of Robotics Applications.
- For implementation of artificial intelligence.

Turing Machine Halting Problem

Input: A Turing machine and an input string **w**.

Problem: Does the Turing machine finish computing of the string **w** in a finite number of steps? The answer must be either yes or no.

Proof: At first, we will assume that such a Turing machine exists to solve this problem and then we will show it is contradicting itself. We will call this Turing machine as a **Halting machine** that produces a 'yes' or 'no' in a finite amount of time. If the halting machine finishes in a finite amount of time, the output comes as 'yes', otherwise as 'no'. The following is the block diagram of a Halting machine −

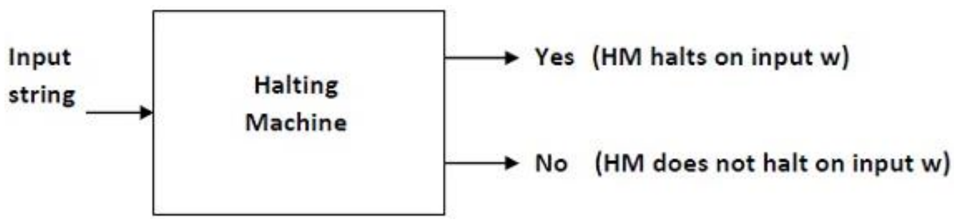

Fig. 6.6: Turing Machine Halting Problem

Now we will design an **inverted halting machine (HM)'** as:

- If **H** returns YES, then loop forever.
- If **H** returns NO, then halt.

The following is the block diagram of an 'Inverted halting machine':

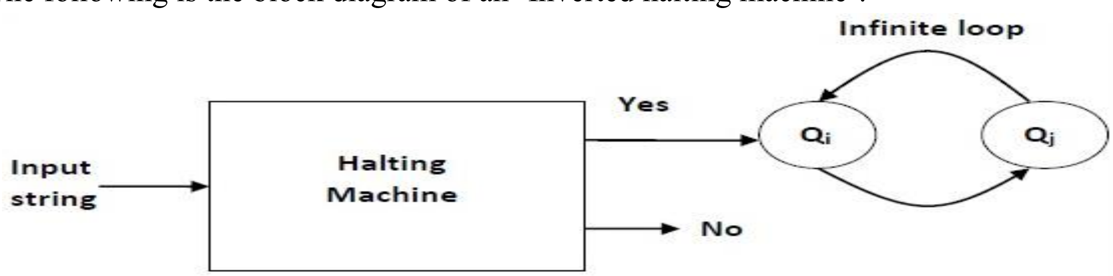

Fig. 6.7: Inverted halting machine

Further, a machine **$(HM)_2$** which input itself is constructed as follows:

- If $(HM)_2$ halts on input, loop forever.
- Else, halt.

Here, we have got a contradiction. Hence, the halting problem is **undecidable**.

Power of Turing Machine:

The turing machine has a great computational capabilities. So it can be used as a general mathematical model for modern computers.

Turing machine can model even recursively enumerable languages. Thus the advantage of turing machine is that it can model all the computable functions as well as the languages for which the algorithm is possible.

Limitations of Turing Machine:

A limitation of Turing machines is that they do not model the strengths of a particular arrangement well.

Another limitation of Turing machines is that they do not model concurrency well. For example, there is a bound on the size of integer that can be computed by an always-halting nondeterministic Turing machine starting on a blank tape.