Undecidability – Recursive and non – Recursive Languages

Dr Jasmine Selvakumari Jeya I

Senior Associate Professor

School of Computing Science and Engineering

VIT Bhopal University

jasmineselvakumarijeya@vitbhopal.ac.in

Undecidability-languages

- Recursive
- Recursively Enumerable
- Non-RE

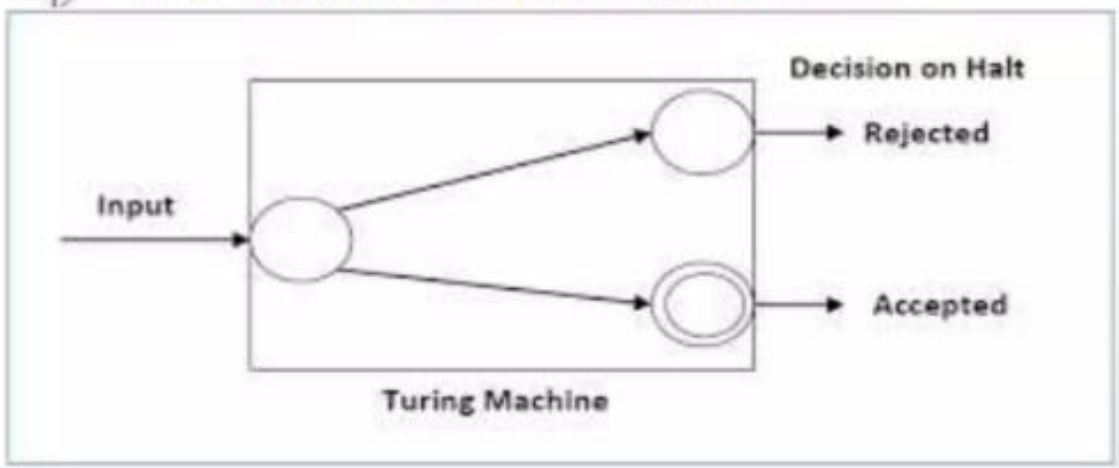
Recursive Language

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

For any string w:

$w \in L \Rightarrow M$ halts in a final state

$w \notin L \Rightarrow M$ halts in a non-final state



Recursively Enumerable Language

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

For any string w:

$w \in L \Longrightarrow M$ halts in a final state

$w \notin L \Longrightarrow M$ halts in a non-final state

or loops forever

Non- Recursively Enumerable

There are languages which does not have a Turing machine at all

Decidable Problems - A decidable problem has an algorithm to determine the answer for a given input

Example: Find whether P is prime or not

2. **Undecidable problem** – problems that has

no algorithm to determine the answer for a given input or

which have an algorithm that answers for some input

Example - no three positive integers a, b and c for any n>2 can ever satisfy the equation:

$$a^n + b^n = c^n.$$

The mortal matrix problem:

Determining, given a finite set of $n \times n$ matrices with integer entries, whether they can be multiplied in some order, possibly with repetition, to yield the zero matrix.

This is known to be undecidable for a set of six or more 3×3 matrices, or a set of two 15×15 matrices.