Name: Ayush Jain
SAP ID: 60004200182
Batch: B
Computer Engineering

FLAT Tutorial-6

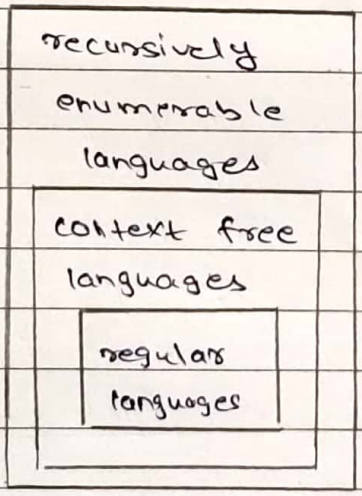
Q. 1) Write a note on Recursively enumerable language.

→ A language L is recursively enumerable language if L is the set of string accepted by some TM.

If L is a recursive enumerable language then:

If $w \in L$, then a TM halts in a final state.
If $w \notin L$ then a TM halts in a non-final state or loops forever.
Recursively enumerable languages are the formal languages that can be decideable.
According to chomsky hierarchy of formal languages, we can see the recursively enumerable languages as type 0 languages.
Some example of recursively langauges are:

1) Recursive language
2) Regular languages
3) Context-Sensitive language
4) Context free languages.



Q. 2) Write a note on Halting Problem.

→ The halting problem is the problem of deciding or concluding based on a given arbitary computer program and its input, whether that program will stop executing or run in an infinite loop for the given input. The halting problem tells us that it is not easy to write a computer program that executes in the limited time that is capable of deciding whether a program halts for input. In addition to that halting problem never say that it is not practicable to determine whether a given random program is going to halt (stop). Generally it asks the question like "Given a random program and an ip input conclude whether the given random program is going to halt when that input is given. The only way to find out is to run the algorithm and see if it halts. Turing machines can be either halting or non-halting depending on the algorithm and input associated with algorithm.

Q.3) Types of Turing machine.

→ Turing machine can also be deterministic or non-deterministic but this does not make them any more or less powerful. Many other TM variation are equivalent to the original TM. this include the following:

(1) Multi-tape turing machine

A turing machine with several tape is called multi-tape turing machine. Every tape's have their own Read/write head.

For N-tape turing machine

$M = \{(Q, \Lambda, \Sigma, \delta, q_0, B, F)\}$

(2) Multiple Track Turing Machine

A K-track Turing machine has K-tracks and one R/W head that reads and write all of them one by one. A K-track turing machine can be simulated by a single track having turing machine.

(3) Two-way infinite Tape Turing Machine

Infinite tape of two-way infinite tape turing machine is unbounded in both direction left and right.

(4) Multi-tape multi-head Turing machine

The multi-tape turing machine has multiple tapes and multiple heads. Each tape is controlled by a seperate head. Multi-tape Multi-head TM 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.

(6) Multi-head Turing Machine

A multi-head TM contain two or more head to read the symbol on same tape. In one step all the heads sense the scanned symbol and move or write independently.

(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 atleast one choice to move. Each choice has several choices of the path it might follow for given input string.

(8) Universal Turing Machine

This is a type of TM that simulates an arbitary turing machine on an arbitary input.

Q.4) Church's hypothesis:

→ The church turing thesis says that every solvable decision problem can be transformed into an equivalent TM problem. It can be explained in two ways:

(1) The church turing thesis for decision problems.

(2) The extended church-Turing thesis for decision problems.

The church turing thesis for decision problems:

There is some effective procedure to solve any decision problem if and only if there is a TM which halts for all input

strings and solves the problem.

The extended church turing thesis for decision problems:

A decision problem is said to be partially solvable if and only if there is an TM which accept precisely the element of question whose answer is yes.

The church TM explains that a decision problem has a solution if and only if there is a TM that determine the answers for every question.

Q.5) Write a note on:

(1) Application of finite automata:

(a) For designing of lexical analysis of a compiler.
(b) For recognizing the patterns using regular expression.
(c) For the designing of the combination and sequential circuits using mealy and moore machines.
(d) Used in text editors.
(e) For implementation of spell checkers.

(ii) Application of Pushdown Automata:

(a) For designing the parsing phase of a compiler.
(b) For implementation of stack application.
(c) For evaluating arithmetic expressions.
(d) For solving the tower of Manoi Problem.