Automata Theory – PYQs

1. Finite Automata (FA)

DFA (Deterministic Finite Automaton)

- Design DFA for strings with even number of 'a's over {a, b}.
- Design DFA for strings ending with "10" over {0, 1}.
- Design DFA for regular expression: (a+b)*aba(a+b)*.
- Design DFA for regular expression: (0+1)*101(0+1)*.
- Limitations of DFA.

NFA (Non-deterministic Finite Automaton)

- Design NFA for strings containing "000" or "010".
- Design NFA for binary strings starting/ending with "11" or "00".
- Convert NFA to DFA.

Regular Expressions

- Write RE for strings with odd number of 1s and any 0s.
- Write RE for strings containing "10" or "11".
- Write RE for:
  1(0+1)*0
  (aa)*(bb)*(b)
  (ab+ba)*
  (A-Z) (a-z) *(a+e+i+o+u)
  (a-z) (a-z | 0-9)* 1(0+1)*0
  (aa)*(bb)*b
  (ab+ba)*.
- Find RE for given grammar:
  S → AB / AS
  A → a / aA
  B → b

2. Moore and Mealy Machines

Definitions

- Differentiate Moore and Mealy machines.
- Describe tuples (states, inputs, outputs, transition functions).

Design Problems

- Moore/Mealy machine for strings ending with "00" or "11".
- Moore/Mealy machine to replace "abb" with "ab" .
- Moore/Mealy machine to replace "1110" with "1011".
- Moore machine to count occurrences of "bba".

3. Context-Free Grammars (CFG) and Pushdown Automata (PDA)

CFG Basics

- Define CFG, ambiguous grammar (prove ambiguity with example).
- Design CFG for L = $\{a^n b^{2n} \mid n \geq 1\}$.
- Design CFG for L = $\{0^n 1 2^n \mid n \geq 0\}$.
- Design CFG for (110+11)*(10)*.

Derivations and Parse Trees

- Leftmost/rightmost derivation for string "aaabbabbba".
  S → aB / bA
  A → aS / bAA / a
  B → bS / aBB / b

PDA Design

- PDA for L = $\{a^n b^n \mid n \geq 1\}$.
- PDA for L = $\{a^n b a^{2n} \mid n \geq 0\}$.
- Power and limitations of PDA (vs. FA and TM).

Normal Forms

- Convert CFG to Chomsky Normal Form (CNF). (110 + 11)* (10)*
- Convert CFG to Chomsky Normal Form (CNF).

  S → aAbB
  A → Ab / b
  B → Ba / a
- Convert CFG to Chomsky Normal Form (CNF).

  S → a | aA | B
  A → aBB | null
  B → Aa | b
- Convert CFG to Greibach Normal Form (GNF).

  S → XY
  X → 0X / IY / I
  Y → 1

4. Turing Machines (TM)

Basics

- Define TM, working principle, and variants (multi-tape, non-deterministic, etc.).
- Write Short note on: Halting Problem in TM.

Design Problems

- TM for L = {$a^n b^n c^n$ | $n \geq 1$}.
- TM for palindromes over {a, b}.
- TM for equal number of 0s and 1s.
- TM for addition of unary numbers.

Universal TM

- Write a note on Universal Turing Machine.

5. Compiler Design

What are the Phases of Compiler?

- Lexical analysis → Syntax analysis → Semantic analysis → Intermediate code → Optimization → Code generation.

Applications

- Applications of FA (lexical analysis), CFG (parsing), PDA (syntax checking), TM (undecidable problems).

6. Chomsky Hierarchy

Diagram and Explanation

- Type 0 (Turing Machine)
- Type 1 (Context-Sensitive Grammar)
- Type 2 (Context-Free Grammar)
- Type 3 (Regular Grammar)

7. Miscellaneous

- Minimization of DFA.
- Right-linear and Left-linear grammars – Definitions and examples.
- What do you mean by ambiguous grammar?
  Prove that the following grammar is ambiguous:
  S → as / aSbS /€