Q.1 Attempt any FOUR (20 Marks)

A. Construct DFA over {0,1} that accepts all strings containing substring "010" – 05 Marks
Key Answer: DFA with states tracking sequence "010", accept in final state.
Marks: DFA components – 2, Transitions – 2, Accepting state – 1

B. Explain the process converting CFG into GNF – 05 Marks
Key Answer: Remove useless symbols, ε-productions, unit productions, left recursion, then ensure all productions start with terminal.
Marks: Each step – 1

C. Design PDA for L = $\{0^n1^n \mid n>1\}$ – 05 Marks
Key Answer: Push 0's, pop for 1's, accept by final state.
Marks: Concept – 2, Transition – 3

D. Using Pumping Lemma, prove L = $\{a^nb^nc^n \mid n \geq 1\}$ is not regular – 05 Marks
Key Answer: Assume regular, take s = $a^pb^pc^p$, divide xyz, pumping y breaks pattern → contradiction.
Marks: Assume s – 1, Divide xyz – 1, Pumping y – 1, Contradiction – 2

E. Write short notes on Recursive and Recursively Enumerable languages – 05 Marks
Key Answer: Recursive: decidable by TM; RE: TM accepts but may not halt for non-members.
Marks: Definition – 2, Example/Difference – 3

F. Explain Equivalence Relation with example and properties – 05 Marks
Key Answer: Reflexive, Symmetric, Transitive; Example: equality on integers.
Marks: Definition – 2, Properties – 2, Example – 1

Q.2 Attempt any FOUR (40 Marks)

A. Design Moore machine to output A if ends in "010", else B – 10 Marks
Key Answer: States track last 3 input bits; output A for "010", else B.
Marks: States – 4, Transitions – 4, Output correctness – 2

B. Convert CFG into CNF: S → pSqq | pSq | pp | q | p | qq – 10 Marks
Key Answer: Remove ε/unit productions, break RHS into max 2 symbols, convert to CNF.
Marks: Steps – 6, Correct CNF – 4

C. Design PDA for L = $\{(ab)^nc^n \mid n \geq 1\}$ with example – 10 Marks
Key Answer: Push 'ab' for each n, pop for c's, accept by final state.
Marks: PDA components – 3, Transitions – 4, Example – 3

**D. Convert RE a(a+b)b → equivalent NFA then DFA* – 10 Marks
Key Answer: Step 1: Construct NFA (Thompson), Step 2: Convert NFA → DFA.
Marks: NFA – 5, DFA – 5

E. Construct leftmost derivation, rightmost derivation, parse tree for "aabb" using S → aSb | ab – 10 Marks
Key Answer: Leftmost: S → aSb → aaSbb → aabb; Rightmost: S → aSb → aabSb → aabb; Parse tree drawn.
Marks: Leftmost – 3, Rightmost – 3, Parse tree – 4

F. Design TM for L = {ww^R} with example – 10 Marks
Key Answer: TM scans input, compares ends to middle; accept if palindrome.
Marks: States – 4, Transition – 4, Example – 2

Q.3 Attempt any FOUR (20 Marks)

A. Design FSM for binary strings with even number of 0's and even number of 1's – 05 Marks

Key Answer: FSM with 4 states tracking parity of 0's and 1's (even-even, even-odd, odd-even, odd-odd); accept in even-even state.
Marks Distribution: States – 2, Transitions – 2, Acceptance – 1

B. Explain the Halting Problem – 05 Marks

Key Answer: Problem of deciding whether a Turing Machine halts on a given input; proven undecidable by contradiction.
Marks Distribution: Definition – 2, Explanation/Reasoning – 3

C. Explain PDA operation – 05 Marks

Key Answer: PDA reads input, uses stack for memory, performs push/pop according to transitions, accepts by final state or empty stack.
Marks Distribution: Operation – 3, Example – 2

D. Construct DFA for strings over {a,b} where every 'a' is immediately followed by at least one 'b' – 05 Marks

Key Answer: DFA with states tracking last symbol; reject if ‘a’ not followed by ‘b’.
Marks Distribution: States – 2, Transitions – 2, Acceptance – 1

E. Define Regular Expression and discuss equivalence with Finite Automata – 05 Marks

Key Answer: Regular Expression: symbolic representation of string sets; equivalent to FA since any RE can be converted to NFA/DFA and vice versa.
Marks Distribution: Definition – 2, Equivalence explanation – 3

F. Explain Chomsky Hierarchy – 05 Marks

Key Answer: Classification of grammars:

- Type 0 – Unrestricted (TM)
- Type 1 – Context-Sensitive
- Type 2 – Context-Free
- Type 3 – Regular
  Marks Distribution: Classification – 3, Explanation/Examples – 2