Unit 1: Introduction

1. Define an alphabet and a language. Give three examples of each.
2. Explain what a grammar is and list the types of grammars as per Chomsky hierarchy.
3. For the grammar $G=(V,\Sigma,R,S)$ with productions $S \to aSb|\epsilon$, list the strings generated by GGG of length 4.
4. Draw the Chomsky hierarchy and explain each level with examples.
5. Differentiate between language, grammar, and automaton.

Unit 2: Regular Languages and Finite Automata

1. Convert the regular expression $(a+b)*abb$ into a DFA.
2. Design a Moore machine to detect the sequence "101" in an input bit stream.
3. Given the following NFA, convert it into an equivalent DFA:
   - States: {q0, q1}
   - Alphabet: {0,1}
   - Transitions: $\delta(q0,0) = \{q0,q1\}$, $\delta(q0,1) = \{q0\}$, $\delta(q1,1) = \{q1\}$
4. Write a regular grammar corresponding to the language $L=\{a^n b^m | n,m \geq 0\}$.
5. Use the pumping lemma to prove that the language $L=\{a^n b^n | n \geq 0\}$ is not regular.
6. Minimize the following DFA (provide a transition table):
   - States: {A, B, C, D}
   - Alphabet: {0,1}
   - Transitions: A→0→B, A→1→C, B→0→A, B→1→D, C→0→D, C→1→A, D→0→C, D→1→B
7. List and prove closure properties of regular languages under union, concatenation, and Kleene star.

Unit 3: Grammars

1. Construct a CFG for the language $L=\{a^n b^n | n \geq 0\}$.
2. Convert the following CFG into Chomsky Normal Form:
   - $S \to ASA|aB$
   - $A \to B|S$
   - $B \to b|\epsilon$
3. Draw parse trees for the string "aabb" using the grammar from Q1.

4. Explain ambiguity with an example grammar and show how to remove ambiguity.
5. Prove using the pumping lemma that the language $L=\{a^n b^n c^n | n \geq 0\}$ is not context-free.
6. Design a PDA that accepts the language $L=\{a^n b^n | n \geq 0\}$ by empty stack method.
7. State closure properties of CFLs and give examples where closure fails.
8. Differentiate between deterministic and nondeterministic PDA with examples.
9. Explain context-sensitive languages and give an example grammar for the language $\{a^n b^n c^n | n \geq 1\}$

Unit 4: Turing Machines

1. Design a TM that accepts the language $L=\{a^n b^n | n \geq 1\}$.
2. Explain the difference between Turing-decidable and Turing-recognizable languages with examples.
3. Prove that the class of Turing-decidable languages is closed under union and intersection.
4. Describe the working of a nondeterministic Turing machine and explain its equivalence with deterministic TM.
5. Show that unrestricted grammars are equivalent in power to Turing machines.
6. Explain the concept of TM as an enumerator with an example.
7. Write the formal definition of a Turing machine and explain each component.

Unit 5: Undecidability

1. State and explain the Church-Turing thesis.
2. Describe the construction of a universal Turing machine.
3. Explain the diagonalization language and how it proves certain languages are undecidable.
4. List at least three undecidable problems and explain why they are undecidable.
5. Discuss the halting problem and prove its undecidability.
6. Explain the significance of universal Turing machine in computability theory.