Unit 1: Introduction

1-Mark Questions:

1. **Define Alphabet.**
2. **What is a String over an alphabet?**
3. **Define Language in the context of automata theory.**
4. **Fill in the blank: The set of all strings over an alphabet Σ is conventionally denoted by _____.**
5. **What is the empty string?**
6. **What is a Grammar in formal language theory?**
7. **What are productions or rules in a grammar?**
8. **What is a derivation in a grammar?**
9. **Fill in the blank: A Context-free grammar is classified as Type-_____ in the Chomsky Hierarchy.**
10. **What is a Type-0 grammar?**

3-Mark Questions:

1. ***Explain the difference between Kleene closure (Σ) and positive closure (Σ+) of an alphabet, providing an example for each.****
2. **Describe the four main components of a formal grammar G = {V, T, P, S}.**
3. **Briefly explain the process of generating a string from a context-free grammar.**
4. **Discuss the basic idea behind the Chomsky Classification of Grammars, mentioning the different types of languages it categorizes.**

4-Mark Questions:

1. **Given the alphabet Σ = {a, b}, illustrate the concepts of alphabet, string, and language by providing specific examples for each.**
2. **Explain the difference between terminal and non-terminal symbols in a context-free grammar, giving examples of each.**
3. **Elaborate on the Chomsky Hierarchy, specifying the characteristics of Type-0, Type-1, Type-2, and Type-3 grammars/languages in terms of production rules.**

5-Mark Questions:

1. **Provide a comprehensive explanation of Alphabets, Strings, and Languages as fundamental concepts in automata theory. Include definitions, notations (e.g., length, concatenation, Kleene star), and concrete examples to illustrate your points.**
2. **Discuss the importance of the Chomsky Hierarchy in the theory of computation. Explain how it categorizes languages based on the complexity of their grammars and the power of automata required to recognise them, listing the four types and their associated characteristics.**

Unit 2: Regular Languages and Finite Automata

1-Mark Questions:

1. **Define Regular Expression.**
2. **What are the primary operators used in Regular Expressions?**
3. **Define Deterministic Finite Automaton (DFA).**
4. **Define Nondeterministic Finite Automaton (NFA).**
5. **What is an epsilon (ε)-transition in an NFA?**
6. **True or False: The output of the Mealy machine is determined only by its current state.**
7. **Fill in the blank: A regular grammar is also known as Type-_____ grammar.**
8. **When are two states in a DFA considered equivalent?**
9. **State the Pumping Lemma for Regular Languages.**
10. ***What is the minimum length of a string formed by the regular expression (a+b) aba* **(a+b)*?****

3-Mark Questions:

1. **Explain the three regular operations (union, concatenation, and Kleene star) on languages, providing a brief example for each.**
2. **Differentiate between a Deterministic Finite Automaton (DFA) and a Nondeterministic Finite Automaton (NFA) based on their transition functions and behaviour.**
3. **Describe the process of converting an NFA with ε-transitions into an equivalent NFA without ε-transitions (ε-elimination).**
4. **State Arden's Theorem.**

5. **Explain the difference between Mealy Machine and Moore Machine in terms of how they produce output.**
6. **Using the Pumping Lemma for Regular Languages, prove that the language L = {0^n 1^n | n ≥ 0} is not regular.**

4-Mark Questions:

1. **Design a Deterministic Finite Automaton (DFA) to accept all strings of 0's and 1's that do not contain the substring "011".**
2. **Construct a Nondeterministic Finite Automaton (NFA) to accept strings containing "0101" as a substring.**
3. **Convert the given NFA to an equivalent DFA using the subset construction method.** ***You would need to provide a specific NFA diagram for this question.***
4. ***Given the regular expression (0 + 1) 010 (0 + 1)*, draw a DFA recognizing the corresponding language.****
5. **Explain the equivalence between Finite Automata and Regular Expressions. Briefly describe the steps involved in converting a DFA to a regular expression.**
6. **Given a DFA, find its minimum finite automaton.** ***You would need to provide a specific DFA for this question.***
7. **Prove that the class of regular languages is closed under the union operation.**

5-Mark Questions:

1. **Design Deterministic Finite Automata (DFA) for the following languages over alphabet {0, 1}:**
   - ***i) L1 = {x ∈ (0,1) | x contains 101 as a substring}****
   - ***ii) L2 = {x ∈ (0,1) | x contains odd number of zeros}****
   - ***iii) L3 = {x ∈ (0,1) | x ends with 11}****
   - ***iv) L4 = {x ∈ (0,1) | x starts with 001}****
2. **Explain the relationship between Finite Automata (DFA/NFA) and Regular Languages. Discuss how regular expressions can be used to describe these languages, providing examples of regular expressions for simple patterns.**
3. **Detailed Construction: Convert the following NFA into an equivalent DFA, clearly showing all steps of the subset construction and the resulting DFA**

state diagram. ***You would need to provide a specific NFA diagram for this question, similar to those found in or.***

Unit 3: Grammars (Context-Free Languages and Pushdown Automata)

1-Mark Questions:

1. **What is an ambiguous grammar?**
2. **Fill in the blank: The language accepted by a Pushdown Automata is a ________ language.**
3. **Define null (ε)-production in a CFG.**
4. **Define unit production in a CFG.**
5. **What is Chomsky Normal Form (CNF) for a CFG?**
6. **True or False: CFLs are closed under substitution.**
7. **What is the height of a parse tree in CNF for a string of length n?**
8. **Fill in the blank: A pushdown automaton is deterministic if for every pair of configuration there is at most ______ configuration that can succeed it.**
9. **What is a parse tree?**
10. **Write the condition for a left-recursive grammar.**

3-Mark Questions:

1. **Specify three practical applications of Context-Free Grammars (CFGs).**
2. **Explain the two ways a language can be accepted by a Pushdown Automaton (PDA): by final state and by empty stack.**
3. **Prove that every regular language is a Context-Free Language (CFL).**
4. **Describe the concept of ambiguity in grammars. Provide a simple example of an ambiguous grammar.**
5. **Explain the process of eliminating ε-productions from a Context-Free Grammar (CFG).**
6. **Briefly explain Chomsky Normal Form (CNF) and its importance in the study of context-free languages.**

4-Mark Questions:

1. **Construct a Context-Free Grammar (CFG) for the language L = {a^m b^n c^p d^q : m + n = p + q}.**
2. **Show that the grammar G with productions S → S+S | SS | *a* | *b is ambiguous for the string "a+a*b".**
3. **Given the grammar G with productions S → AB / CA, B → BC / AB, A → a, C → aB / b, find a reduced grammar equivalent to G (by eliminating useless productions/variables).**
4. **Convert the following Context-Free Grammar (CFG) to Chomsky Normal Form (CNF): S → ABA, A → aA / ε, B → bB / ε.**
5. **Design a Pushdown Automaton (PDA) to accept strings with more a's than b's. Trace its acceptance for the string "abbabaa".**
6. **Construct a Context-Free Grammar (CFG) for the language L = {a^n b^n | n ≥ 0}.**
7. **Convert the following CFG into Greibach Normal Form (GNF): S → AB, A → BS / b, B → SA / a.**

5-Mark Questions:

1. **Define Context-Free Grammar (CFG) and Context-Free Language (CFL). Explain how a language can be generated by a CFG, illustrating with a derivation example for the string bbaababa from the grammar S → bB / aA, A → b / bS / aAA, B → a / aS / bBB.**
2. **Discuss the importance of Normal Forms in Context-Free Grammars. Detail the Chomsky Normal Form (CNF) by defining its structure and outlining the general steps to convert any CFG into CNF.**
3. **Explain the relationship between Pushdown Automata (PDA) and Context-Free Grammars (CFG). Describe the algorithm to find a PDA corresponding to a given CFG, outlining the types of transitions used.**
4. **State and explain the Pumping Lemma for Context-Free Languages. Provide an example of a language that can be proven non-context-free using this lemma, clearly demonstrating the application of the lemma.**