Short Questions

1. Define Push Down Automaton (PDA).
2. How does a PDA differ from a finite automaton?
3. What is meant by the 'stack' in a PDA?
4. Explain the concept of non-determinism in PDAs.
5. How is the language of a PDA defined?
6. What is the role of the input alphabet in a PDA?
7. Describe the stack alphabet in the context of PDAs.
8. Explain the acceptance by final state in a PDA.
9. What does acceptance by empty stack signify in a PDA?
10. Distinguish between deterministic and non-deterministic PDAs.
11. Can every non-deterministic PDA be converted to a deterministic PDA? Why or why not?
12. Describe how a PDA can simulate a context-free grammar.
13. How can context-free grammar be converted into a PDA?
14. What is the significance of the transition function in a PDA?
15. Explain the concept of instantaneous description in PDAs.
16. Can a PDA have multiple start states? Explain.
17. Describe a PDA for the language of balanced parentheses.
18. How are transitions defined in a PDA?
19. What is the difference between a PDA and a Turing Machine?
20. Explain the concept of equivalence in the context of PDAs.
21. Can a PDA recognize every type of language? Justify your answer.
22. How does a PDA process its input and stack elements?
23. What are the limitations of PDAs compared to Turing Machines?
24. Explain how a PDA can be used to recognize palindromes.
25. Describe the process of converting a PDA to a context-free grammar.
26. What are normal forms in the context of context-free grammars?
27. Explain the significance of eliminating useless symbols in CFGs.

360 DigiTMG®
Digital Transformation | Management | Governance

28. How are ε-productions eliminated from a CFG?

29. Define Chomsky Normal Form and its importance.

30. Explain Griebach Normal Form and its application.

31. What are the key differences between Chomsky and Griebach Normal Forms?

32. Why is it necessary to convert a CFG to a normal form?

33. Describe the process of converting a CFG to Chomsky Normal Form.

34. What are the limitations of Chomsky Normal Form?

35. How does Griebach Normal Form aid in parsing algorithms?

36. Can every CFG be converted to Griebach Normal Form? Explain.

37. Discuss the role of normal forms in simplifying CFGs.

38. How do normal forms impact the computational complexity of parsing?

39. Explain the concept of left recursion in CFGs.

40. What is the purpose of removing null productions in CFGs?

41. How do normal forms affect the language generated by a CFG?

42. Are there any CFGs that cannot be converted to normal forms? Give an example.

43. Describe the impact of normal forms on the ambiguity of CFGs.

44. Explain how normal forms assist in language recognition.

45. What are the challenges in converting a CFG to normal forms?

46. State the Pumping Lemma for Context-Free Languages.

47. Explain the importance of the Pumping Lemma in the theory of computation.

48. How can the Pumping Lemma be used to prove that a language is not context-free?

49. Provide an example where the Pumping Lemma is applied to a context-free language.

50. Discuss the limitations of the Pumping Lemma for context-free languages.

51. Explain the concept of 'pumping' in the Pumping Lemma for context-free languages.

52. Can the Pumping Lemma be used to prove that a language is context-free? Explain.

360 DigiTMG® Digital Transformation | Management | Governance

53. Describe the conditions under which the Pumping Lemma holds for a context-free language.

54. How does the Pumping Lemma differ for context-free languages compared to regular languages?

55. Provide an example of a language that satisfies the Pumping Lemma but is not context-free.

56. What are the closure properties of context-free languages?

57. Is the union of two context-free languages always context-free? Explain.

58. Discuss whether the intersection of two context-free languages is context-free.

59. Are context-free languages closed under concatenation? Provide reasoning.

60. Explain if context-free languages are closed under reversal.

61. Discuss the closure property of context-free languages under the Kleene star operation.

62. Are context-free languages closed under complementation? Justify your answer.

63. Provide an example of a closure property of context-free languages.

64. How do closure properties of context-free languages differ from those of regular languages?

65. Explain the significance of closure properties in parsing and language processing.

66. Introduce the concept of a Turing Machine.

67. Describe the formal definition of a Turing Machine.

68. What is an instantaneous description in the context of Turing Machines?

69. Explain how a Turing Machine reads and writes on its tape.

70. Discuss the concept of the head movement in Turing Machines.

71. What are the possible states in a Turing Machine?

72. How does a Turing Machine recognize a language?

73. Describe the transition function of a Turing Machine.

74. Can a Turing Machine simulate a finite automaton? Explain.

75. What is the difference between a Turing Machine and a PDA?

76. Explain the concept of a universal Turing Machine.

77. Discuss the significance of the halting problem in Turing Machines.

360 DigiTMG Digital Transformation Management Governance

78. Can a Turing Machine have an infinite tape? Discuss its implications.
79. How do Turing Machines contribute to the understanding of computability?
80. Describe the limitations of Turing Machines in computational theory.
81. 81.Can a Turing Machine be non-deterministic? Explain its workings.
82. Discuss the role of the alphabet in a Turing Machine.
83. How are problems encoded in a Turing Machine?
84. Explain the significance of Turing Machines in modern computing.
85. Describe a Turing Machine that can compute a simple arithmetic operation.
86. What are the different types of Turing Machines?
87. Explain the concept of a multi-tape Turing Machine.
88. How does a non-deterministic Turing Machine differ from a deterministic one?
89. Discuss the concept of a multi-head Turing Machine.
90. What is the purpose of a multi-dimensional Turing Machine?
91. How do different types of Turing Machines compare in terms of computational power?
92. Can all types of Turing Machines solve the same class of problems? Explain.
93. Discuss the practical applications of various types of Turing Machines.
94. How do multi-tape Turing Machines simplify certain computations?
95. Explain the impact of non-determinism on the functioning of a Turing Machine.
96. Define undecidability in the context of computational theory.
97. Provide an example of a language that is not recursively enumerable.
98. Discuss an undecidable problem that is recursively enumerable.
99. List some undecidable problems about Turing Machines.
100. Define recursive languages and their characteristics.
101. Explain the Post's Correspondence Problem.
102. Describe the Modified Post Correspondence Problem.
103. Mention other notable undecidable problems.
104. Define counter machines and their role in computational theory.
105. Discuss the significance of undecidability in computer science.

360 DigiTMG® Digital Transformation | Management | Governance

106. Explain how the concept of non-determinism in Turing Machines influences the class of problems they can solve compared to deterministic Turing Machines. Discuss the theoretical implications of non-determinism on computational complexity and decision problems.

107. How do undecidable problems relate to the Turing Machine's halting problem?

108. What is the difference between decidable and undecidable problems?

109. Can undecidability be proven using reduction techniques? Provide an example.

110. Explain the concept of a recursively enumerable (RE) language.

111. Are there languages that are neither RE nor co-RE? Give an example.

112. Discuss the impact of undecidability on algorithm design.

113. How does the Church-Turing Thesis relate to undecidability?

114. Explain the significance of Rice's Theorem in the context of undecidability.

115. Can the concept of undecidability be applied to practical computing problems? Give an example.

116. Describe the implications of undecidable problems in formal language theory.

117. How do undecidable problems influence the study of computational complexity?

118. What role do counter machines play in the study of undecidability?

119. Can any real-world problems be classified as undecidable? Provide examples.

120. Discuss the relationship between undecidability and non-determinism in computational models.

121. How does the concept of undecidability impact the field of artificial intelligence?

122. Explain the relevance of undecidability in the development of programming languages.

123. Discuss the practical limitations posed by undecidable problems in software development.

124. Can undecidable problems be approximated or circumvented in algorithmic solutions? How?

125. What are the philosophical implications of undecidability in computational theory?

360 DigiTMG® Digital Transformation | Management | Governance