

Dronacharya College of Engineering

Department of Electronics and Computers Engineering

ASSIGNMENT QUESTIONS

(Session 2014-2015)

Subject with code: Compiler Design (EC-712 F)

Sem: VII / Branch: ECS

1. Convert the regular expression $(a|b)^*ab(a+b)^*$ into corresponding minimized finite state automata.
2. What is Cross Compiler? Explain Briefly Requirement of 1 Pass, 2 Pass and multipass compiler.
3. Describe the various phases of Compiler.
4. Explain in detail about compiler constructions tools.
5. Explain in detail about the role of lexical analyzer.
6. Explain briefly about the Input buffering technique with the algorithm.
7. Explain in detail about the cousins of compiler.
8. What is parser? Write the predictive parsing algorithm with example.
9. Construct predictive parsing table for the grammer
a. $E \rightarrow E+T \mid T; T \rightarrow \bar{T} * F \mid F; F \rightarrow (E) \mid id.$
10. Explain Top Down parsing with suitable example.
11. Construct the SLR parser for the following grammer with an appropriate algorithm
a. $S \rightarrow L = R, S \rightarrow R, L \rightarrow *R \mid id, R \rightarrow L$
12. Construct the SLR parser for the following grammer with an appropriate algorithm
a. $E \rightarrow E+T \mid \bar{T}, T \rightarrow T * F \mid F, F \rightarrow (E) \mid id$
13. What are syntax trees and how these can be constructed?
14. Explain syntax directed translation scheme with examples.
15. Define three-address code. Describe the various type & methods of implementing three-address statements with example.
16. What are the various data structure used for the symbol table construction and explain in detail.
17. How can Back Patching be used to generate code for Boolean expression and construct the annotated parse tree with translation scheme?
18. Discuss the various methods for translating Boolean expression.
19. Briefly explain about simple code generator.
20. Define basic block. Write an algorithm to partition a sequence of three address statements into basic block

21. Construct the DAG for the following basic block:
 $d := b * c$ $e := a + b$ $b := b * c$ $a := e - d$
22. Write in detail about the issues in the design of a code generator.
23. Write in detail about function — preserving transformations & loop optimizations.
24. Explain in detail about code- improving transformations.
25. Describe in detail about principal sources of optimization.
26. Explain in detail optimization of basic blocks with example.
27. Discuss briefly about peephole optimization.
28. State the main differences between a compiler and an interpreter.

29. What is Cross Compiler? Explain Briefly Requirement of 1 Pass, 2 Pass and multipass compiler.
30. What is Input Buffering? Explain.
31. Convert the regular expression $(a|b)^* ab(a+b)^*$ into corresponding minimized finite state automata.
32. Explain Different Phases of Compiler.
33. Explain why a left-recursive grammar cannot be parsed using the predictive top-down parsing algorithms.
34. Q.2 Consider the following CFG $G = (N = \{S, A, B, C, D\}, T = \{a, b, c, d\}, P, S)$ where the set of productions P is given below:
 $S \rightarrow A$
 $A \rightarrow BC \mid DBC$
 $B \rightarrow Bb \mid \epsilon$
 $C \rightarrow c \mid \epsilon$
 $D \rightarrow a \mid d$
 - a) Is this grammar suitable to be parsed using the recursive descent parsing method? Justify and modify the grammar if needed.
 - b) Compute the FIRST and FOLLOW set of non-terminal symbols of the grammar resulting from your answer in (a)
 - c) Construct the corresponding parsing table using the predictive parsing LL method.
 - d) Show the stack contents, the input and the rules used during parsing for the input $w = dbb$

35. What is parser? Write the predictive parsing algorithm with example.
36. Create 3- address code quadruple, triple and indirect triple for the following expression:
 - a. $a + a * (b - c) + (b - c) * d$
37. What are syntax trees and how these can be constructed?
38. Explain syntax directed translation scheme with examples.
39. What are the attributes associated with a symbol table? Discuss the operations possible with a symbol table.
40. Explain code optimization and its utility.
41. Explain Dag representation with the help of example.

42. Write a short note on loop unwinding.
43. What do you mean by common subexpression? What are the advantage of identifying it.
44. What is meant by register allocation? Why it is considered as important.