



CS6660 – COMPILER DESIGN Question Bank

UNIT I-INTRODUCTION TO COMPILERS

Part A

1. Define compiler.
2. Differentiate compiler and interpreter.
3. What is a language processing system?
4. List four software tools that analyses the source program.
5. What are the cousins of compiler?
6. Define assembler and write its types
7. What are the classifications of compiler?
8. Define preprocessor
9. What are the functions of preprocessors?
10. List the compiler construction tools.
11. What do you meant by passes?
12. What is front end and back end?
13. What do you meant by phases?
14. List the phases of compiler
15. What is meant by lexical analysis?
16. What is meant by syntax analysis?
17. What is meant by semantic analysis?
18. What is meant by intermediate code generation?
19. What are the formats of Intermediate code?
20. What are the properties of Intermediate Code
21. Define three address code.
22. What are the properties of three address code?
23. Define Symbol Table.
24. Write short notes on symbol table manager
25. Write short notes on error handler
26. List the types of compiler

Part B

1. Explain the various phases of compiler in detail. Also write down the output for the following expression after each phase $a=b*c-d$ or $a=b+c*50$.
2. Briefly explain the compiler construction tools.
3. Elaborate on grouping of phases into a pass in a compiler.
4. Discuss about the language processors in detail.
5. Discuss about language programming language basics in detail.



UNIT II - LEXICAL ANALYSIS

Part A

1. What are the issues in lexical analysis?
2. Differentiate lexeme, tokens and patterns.
3. Give the algebraic properties of regular expression
4. What are the operations on language?
5. Give the error recovery actions in lexical errors?
6. Define regular expression?
7. Give the precedence of regular expression operator?
8. Give the rules in regular expression?
9. Give the types of notational shorthand's of RE?
10. Define kleene closure or star closure and positive closure.
11. Define character class with example.
12. Define regular set?
13. What are the tasks in lexical analyzer?
14. Define finite automata and write its types.
15. What are the three parts of lexical program?
16. What are the four functions of converting regular expression to DFA directly?

Part B

1. Explain the Role of lexical analyzer in detail.
2. Write short notes on token specification.
3. Write about regular expressions and regular definitions with examples.
4. For the following regular expression, draw the NFA and also Construct minimized DFA
 - $(a/b)^* abb$
 - $(a/b)^* a (a/b)$
 - $(a/b)^* a (a/b)$
 - $(a/b)^* a(a/b) (a/b)$
 - $(a/b)^* abb (a/b)^*$
5. Construct optimized DFA for the following regular expression.
 - $(a/b)^* abb$
 - $(a/b)^* a (a/b)$
 - $(a/b)^* a (a/b)$
 - $(a/b)^* a(a/b) (a/b)$
 - $(a/b)^* abb (a/b)^*$
6. Write about the tool for generating lexical analyzer with example program.



UNIT III - SYNTAX ANALYSIS

Part A

1. What do you mean by parser and its types?
2. What are the goals of error handler in a parser?
3. What are error recovery strategies in parser?
4. Define CFG.
5. Define derivations. Give an example and its types.
6. Define parse tree.
7. Define ambiguity or ambiguous grammar.
8. Define sentential form.
9. Define yield of the string.
10. Give the several reasons for writing a grammar.
11. Define left factoring.
12. What are the difficulties with top down parsing?
13. Define top down parsing.
14. What is meant by recursive-descent parser?
15. What is a predictive parser?
16. Define left recursion. Give an example.
17. Eliminate left recursion from the grammar.
 $S \rightarrow Aa \mid b, A \rightarrow Ac \mid Sd \mid \epsilon$
18. Define LL (1) grammar.
19. What are the possibilities of non-recursive predictive parsing?
20. Define bottom up parsing.
21. What are the three techniques for constructing LR parsing table?
22. What are the actions available in shift reduce parser?
23. Define handle.
24. Define handle pruning.
25. Define viable prefixes.
26. What are the two common ways of determining precedence relations should hold between pair of terminals?
27. Define LR parser.
28. What are the drawbacks of LR parser?
29. Give the reasons for using LR parser.
30. What are the techniques for producing LR parsing Table?
31. What are the two functions of LR parsing algorithm?
32. What are two classes of items in LR parser?
33. Define augmented grammar.
34. Define LR (0) items.
35. Define SLR parser.
36. Define LALR grammar.
37. Define YACC tool.



SYED AMMAL ENGINEERING COLLEGE

(An ISO 9001:2008 Certified Institution)

Dr. E.M. Abdullah Campus, Ramanathapuram – 623502



DEPARTMENT OF INFORMATION TECHNOLOGY

Part B

1. Construct LL (1) or predictive parsing table for the given grammar.
 $S \rightarrow (L) \mid a$
 $L \rightarrow L, S \mid S$
From the above parsing table, find out whether the given string is accepted or not (a, a).
2. Write First position and Follow position rule.
3. Construct LL (1) or predictive parsing table for the given grammar.
 $S \rightarrow a \mid (T)$
 $T \rightarrow T, S \mid S$
4. What are the conflicts that occur during shift reduce parsing.
5. Check whether the following grammar is SLR or not. Explain your answer with reasons.
 $S \rightarrow L=R \mid R$
 $L \rightarrow *R \mid id$
 $R \rightarrow L$.
6. Check whether the string $id+id*id$ is accepted for the given grammar using Shift reduce parser.
 $E \rightarrow E+E$
 $E \rightarrow E * E$
 $E \rightarrow (E)$
 $E \rightarrow id$
7. Construct LR (0) parsing table for the given grammar.
 $S \rightarrow AS \mid b$
 $A \rightarrow SA \mid a$
8. Write an algorithm for construction of LR(1) or CLR items for grammar G' .
9. Show that following grammar is CLR or LR(1) but not LALR for the given grammar.
 $S \rightarrow (L) \mid a$
 $L \rightarrow L, S \mid S$
10. Show that the following grammar is LALR but not in SLR.
 $S \rightarrow Aa \mid bAc \mid dc \mid bda$
 $A \rightarrow d$
Construct LALR parsing table.
11. Construct LALR parsing table for the given grammar.
 $S \rightarrow XX$
 $X \rightarrow aX$
 $X \rightarrow b$
12. Write an algorithm to eliminate left recursion from a grammar and also explain with an example.
13. Show how the error handling and recovery actions are made by syntax analyzer.
14. Discuss about YACC tool and write a parser generator program for desk calculator.



SYED AMMAL ENGINEERING COLLEGE

(An ISO 9001:2008 Certified Institution)

Dr. E.M. Abdullah Campus, Ramanathapuram – 623502



DEPARTMENT OF INFORMATION TECHNOLOGY

UNIT IV - SYNTAX DIRECTED TRANSLATION & RUN TIME ENVIRONMENT

Part A

1. Define a syntax-directed translation.
2. Define Syntax Directed Definition.
3. Define an attribute. Give the types of an attribute .
4. Define Synthesized attributes.
5. Define annotated parse tree.
6. Define S-attributed definition.
7. Define inherited attributes.
8. What is dependency graph?
9. What is topological sort?
10. Write some methods for evaluating semantic rules.
11. Differentiate static checking.
12. Define type checker.
13. Define type systems.
14. Differentiate static and dynamic checking of types.
15. Define activation trees.
16. Define control stack.
17. Define binding of names.
18. What are the fields in an activation record?
19. Give the standard storage allocation strategies.
20. Define static allocations and stack allocations
21. What is dangling references?

Part B

1. Discuss in detail about Syntax Directed Definitions.
2. Write about Bottom-up Evaluation of S-Attribute Definitions
3. Explain about type systems and type conversions
4. Explain about Symbol table and its manipulation in detail.
5. Explain about Storage Allocation strategies.
6. Discuss about Dynamic Storage allocation



SYED AMMAL ENGINEERING COLLEGE

(An ISO 9001:2008 Certified Institution)

Dr. E.M. Abdullah Campus, Ramanathapuram – 623502



DEPARTMENT OF INFORMATION TECHNOLOGY

UNIT V - CODE OPTIMIZATION AND CODE GENERATION

Part A

1. Define basic block and flow graph.
2. Write the step to partition a sequence of 3 address statements into basic blocks.
3. Give the important classes of local transformations on basic blocks
4. Describe algebraic transformations.
5. Define DAG. Give an example.
6. Write the labels on nodes in DAG.
7. Give the applications of DAG.
8. Define Peephole optimization.
9. Write the characteristics of peephole optimization?
10. What are the structure preserving transformations on basic blocks?
11. Define Common sub-expression elimination with ex.
12. Define Dead-code elimination with ex.
13. Define Renaming of temporary variables with ex.
14. Define reduction in strength with ex.
15. Define use of machine idioms.
16. Define optimization and optimizing compiler.
17. Define code generation.
18. What are the issues in the design of code generator?
19. Give the variety of forms in target program.
20. Give the factors of instruction selections.
21. What are the sub problems in register allocation strategies?
22. Write the addressing mode and associated costs in the target machine.
23. What is meant by register descriptors and address descriptors?
24. What are the actions to perform the code generation algorithms?

Part B

1. Explain the principle sources of optimization.
2. Explain peep-hole optimization.
3. Explain about Loop optimization technique with suitable example
4. Explain about DAG and its applications with example.
5. What is global data-flow analysis? Explain in detail
6. Briefly explain the issues in design of code generator
7. Explain language facilities for dynamic storage allocation of storage data.
8. Write brief notes for storage allocation in FORTRAN program