Printed Page:- 04 Subject Code:- ACSBS0502

Roll. No:

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA

(An Autonomous Institute Affiliated to AKTU, Lucknow)

B.Tech

SEM: V - THEORY EXAMINATION (2024- 2025)

Subject: Compiler Design

Time: 3 Hours Max. Marks: 100

General Instructions:

IMP: *Verify that you have received the question paper with the correct course, code, branch etc.*

1. This Question paper comprises of three Sections -A, B, & C. It consists of Multiple Choice Questions (MCQ's) & Subjective type questions.

2. Maximum marks for each question are indicated on right -hand side of each question.

3. Illustrate your answers with neat sketches wherever necessary.

4. Assume suitable data if necessary.

5. Preferably, write the answers in sequential order.

6. No sheet should be left blank. Any written material after a blank sheet will not be evaluated/checked.

SECTION-A 20

1. Attempt all parts:-

1-a. Transition function in a Deterministic Finite Automaton (DFA) is ..... . (CO1, K1) 1

(a) $Q \times \Sigma \rightarrow Q$

(b) $Q \times \Sigma \rightarrow 2^Q$

(c) $Q \times \Sigma \rightarrow 2^n$

(d) None of these

1-b. Lexical Analyser is the ..... phase of Compiler. (CO1, K1) 1

(a) First

(b) Second

(c) Third

(d) None of the above

1-c. Grammar that produce more than one Parse tree for same sentence is ____. (CO2, K1) 1

(a) Ambiguous

(b) Unambiguous

(c) Complementation

(d) Concatenation Intersection

1-d. Parsing is categorized into ..... types. (CO2, K1) 1

(a) 2

(b) 3
(c) 1
(d) 4

1-e. The important component for semantic analysis is _____ . (CO3, K1) 1

(a) Yacc
(b) Lex
(c) Symbol Table
(d) Type Checking

1-f. A synthesized attribute is an attribute whose value at a parse tree node depends on ---------- . (CO3, K1) 1

(a) Attributes at the siblings only
(b) Attributes at parent node only
(c) Attributes at children nodes only
(d) None of the above.

1-g. The value of _____ variable is updated inside the loop by a loop-invariant value. (CO4, K1) 1

(a) loop
(b) Strength
(c) Induction
(d) Invariable

1-h. Optimization is a program transformation technique, which tries to improve the code by making it consume. (CO4, K1) 1

(a) more resources
(b) less resources
(c) previous resources
(d) None of the mentioned

1-i. The basic block and successor relationship is described by _____ . (CO5, K1) 1

(a) Control graph
(b) DAG
(c) Flow graph
(d) Hamilton graph

1-j. The _____ method merges the multiple loops into the single one. (CO5, K1) 1

(a) Constant Folding
(b) Loop rolling
(c) Loop fusion or jamming
(d) None of the above

2. Attempt all parts:-

2.a.	Explain the term Compiler. (CO1, K1)	2
2.b.	Write algorithm for FOLLOW function. (CO2, K1)	2
2.c.	Define symbol table. (CO3, K1)	2
2.d.	Write the various types of intermediate code representation. (CO4, K1)	2
2.e.	Define constant folding. (CO5, K1)	2

SECTION-B 30

3. Answer any five of the following:-

3-a.	Explain the role of linker, loader and preprocessor in the process of compilation. (CO1, K1)	6
3-b.	Explain role of lexical analyzer with example in compiler design process. (CO1, K1)	6
3-c.	Design FIRST and FOLLOW set for the following grammar. (CO2, K3) S→ ACB / CbB / Ba A→ da / BC B→ g / ε C→ h / ε	6
3-d.	Discuss bottom-up parsing. Describe bottom-up parsing techniques. (CO2, K1)	6
3.e.	Differentiate between S-attributed SDT and L-attributed SDT. (CO3, K2)	6
3.f.	Explain peephole optimization with example. (CO4, K1)	6
3.g.	Illustrate the process of register allocation for the given instructions: (CO5, K3) 1. x = y + z 2. w = x * y 3. v = w + z	6

SECTION-C 50

4. Answer any one of the following:-

4-a.	Define LEX and explain its significance in lexical analyzer generation. Demonstrate how LEX is utilized in the process of generating lexical analyzers with relevant examples. Additionally, describe the role of YACC in a compiler, outlining its structure and functionality. (CO1 , K3)	10
4-b.	Explain in detail the process of compilation, outlining each phase and its significance. Demonstrate the outputs of each phase of the compilation process for the input expression x = y + z * 10. Describe the differences between a single-pass and multi-pass compiler, and explain which approach would be more efficient for the given input. (CO1, K3)	10

5. Answer any one of the following:-

5-a.	Explain the difference between Deterministic and Non-Deterministic grammars, providing relevant examples to illustrate their differences. Then, apply the process of left factoring to simplify the following grammars: S → iEtS / iEtSeS	10

Finally, analyze the importance of left factoring in predictive parsing and discuss its role in constructing LL parsers. (CO2, K4)

5-b. Identify and explain the different types of LR parsers. Select one type of LR parser and illustrate its working with an example, explaining the steps involved in parsing. Analyse the advantages and limitations of the LR parser compared to LL parsers. (CO2, K4) 10

6. Answer any one of the following:-

6-a. Explain the basic structure and role of a symbol table in the compilation process. Identify the attributes stored in a symbol table for variables, functions, and data types. Compare and contrast different techniques used for symbol table management, such as hashing and linked lists. (CO3, K4) 10

6-b. Illustrate the difference between a Parse Tree and an Annotated Parse Tree, providing suitable examples for each. Construct the Annotated Parse Tree for the expression 5 + 4 * 6 based on operator precedence and associativity rules, and Solve the expression using a desk calculator approach. (CO3, K3) 10

7. Answer any one of the following:-

7-a. Explain code optimization in compilers and its impact on performance. Provide an example where code optimization improves efficiency. Describe any two key loop optimization techniques. Illustrate one technique with a numerical example. (CO4, K2) 10

7-b. Translate the given expression a = b * –c + b * –c into three-address code, explaining the process step by step.Represent the three-address code in triple, indirect triple, and quadruple forms. Compare the differences between these representations. (CO4, K4) 10

8. Answer any one of the following:-

8-a. Define "Dead Code" in compilers and explain its impact on program performance. Discuss the role of optimization phases and their contribution to execution efficiency. Describe the importance of dead code elimination and how it affects program size and execution time. (CO5, K3) 10

8-b. Define the process of code generation in compilers and explain its role in the compilation process. Describe the key issues encountered in designing a code generator, such as target machine dependencies, instruction selection, and optimization. (CO5, K1) 10