

Unit 4

1. Define compiler. Explain various phases of the compiler with a neat sketch
2. Differentiate between Top down and Bottom up parsing techniques
3. Explain with an example how the lex program performs lexical analysis for the arithmetic operators and identifiers in C?
4. Show that the following grammar is LALR(1)
 $S \rightarrow Aa \mid bAc \mid dc \mid bda$
 $A \rightarrow d$
5. Differentiate Token, Lexeme and Pattern with suitable examples. How to recognize the Tokens?
6. Give the basic structure of a compiler and explain various components briefly.
7. Explain with an example how the lex program performs lexical analysis for the arithmetic operators and identifiers in C.
8. Construct a CLR parsing table for the following Grammar
 $S \rightarrow L = R$
 $S \rightarrow R$
 $L \rightarrow *R$
 $L \rightarrow id$
 $R \rightarrow L$ (Write all necessary procedures).
9. Explain various Error Recovery strategies in Lexical analysis.
10. State the reasons for separating Lexical analysis and Syntax analysis.
11. Discuss how Finite Automata are used to recognize tokens and perform lexical analysis with example?
12. Describe the analysis-synthesis model of a compiler.
13. What is left-factoring? Write the algorithm to eliminate left-factoring from a grammar. Explain the same with an example
14. Eliminate left recursion in the following grammar:
 $E \rightarrow E + T \mid T,$
 $T \rightarrow T * F \mid F,$
 $F \rightarrow (E) \mid id$
15. What is left recursion? Describe the algorithm used for eliminating left recursion.
16. Construct a Finite automata and scanning algorithm for recognizing identifiers, and numerical constants in 'C' language.
17. Draw the structure of a compiler and describe various phases in the compilation process. Mention the output of the following statement: `id1=id2+id3*50` at each phase.
18. Design the LEX program that recognizes the tokens of a C language and returns the token found.
19. Compute LR(0) items for the following grammar and construct SLR parser table:
 $S \rightarrow L=R \mid R$
 $L \rightarrow *R \mid id$
 $R \rightarrow L$

Unit 5

1. Construct a Quadruple, Triple, and Indirect triple for the statement $a + a * (b - c) + (b - c) * d$?
2. Explain various storage allocation strategies with an example.
3. What are the three forms of intermediate code representations? Explain them.
4. What is an activation record? Describe various components in an activation record considering a sample c program.
5. Give Syntax Directed Translation scheme for Simple Desk Calculator.
6. Give the syntax-directed definition of a simple desk calculator and construct an annotated parse tree for the input expression $(4*7+1)*2$.
7. Differentiate Synthesized and Inherited Attributes with the example?
8. Convert the following arithmetic expression into Syntax Tree and Three Address Code $b*3(a+b)$.
9. Compare and contrast S-Attributed definitions with L-Attributed definitions.
10. How is stack storage allocation strategy different from heap allocation strategy? Describe them, mentioning their merits and demerits.

