Register No. | 9 | 5 | 0 | 7 | 1 | 9 | 1 | 2 | 0 | 3 | 7 |

FRANCIS XAVIER ENGINEERING COLLEGE,TIRUNELVELI – 627 003

(Autonomous)

QP CODE: 191261

B.E. / B.Tech Degree Examinations –Apr/May 2022

Sixth Semester

Regulation –2019

B.E - Computer Science and Engineering

19CS6601– Compiler Design

Time : 3 Hours

Maximum Marks : 100

Part A **(10 x 2 = 20 Marks)**

Answer ALL Questions

1. What are the functions of Preprocessor?
2. State some compiler construction tools?
3. Define Token, Pattern and Lexeme?
4. Differentiate DFA and NFA
5. What are the problems with top down parsing?
6. Why left factoring is needed for parsing?
7. Write the three address code for the statement a=b*-c + b*-c ?
8. What are the various representations of intermediate code
9. What is code motion?
10. Define basic block and flow graph.

Part B **(5 x 13 = 65 Marks)**

Answer ALL Questions

11. a) What are the various phases of compiler? Explain each phase in detail. (13)

(OR)

b) (i)Describe the Language Processing System in detail (13)

12. a) Analyze the roles of lexical analyzer with suitable example? (13)

Draw the transition diagram that recognizes the lexemes matching the token relop(relational operator)

(OR)

b) Explain the procedure for converting Regular Expression to DFA. Also find the minimum number of states for the regular expression (a + b) * b(a +b) over the alphabet {a, b}. in a deterministic finite-state automaton (DFA) accepting L (13)

13. a) Construct a predictive parsing table for the grammar (13)

E -> E + T / F

T -> T * F / F

F -> (E) / id

(OR)

b) Give the LALR parsing table for the grammar. (13)

S -> L = R / R

L -> * R / id

R -> L

14. a) What is back patching. Explain the syntax directed Translation for Boolean expression using back patching. (13)

(OR)

b) What is a three address code? Mention its types. How would you implement these address statements? Explain with suitable examples. (13)

15. a) Explain in detail principal sources of optimization. (13)

(OR)

b) Write in detail about the issues in the design of code generator. (13)

Part C **(1 x 15 = 15 Marks)**

Answer ALL Questions

16. a) Consider the grammar. (15)

E -> E + T

E -> T

T -> T * F

T -> F

F -> (E) / id

Construct an LR parsing table for the above grammar. Give the moves of the LR parser on id * id + id

(OR)

b) Explain about Optimization of basic blocks. State the code generation algorithm with suitable example. (15)