Code No: R2031421 **R20** SET - 1

III B. Tech I Semester Regular Examinations, Dec/Jan-2022-23
COMPILER DESIGN
[**Common to** CSE (AIML), CSE(AI), CSE(DS), CSE(AIDS), AIDS, AIML,CSD]
Time: 3 hours Max. Marks: 70

Answer any **FIVE** Questions **ONE** Question from **Each unit**
All Questions Carry Equal Marks
*****

UNIT-I

1. a) Explain about input buffering schemes in lexical analysis. [7M]
   b) Write a regular expression for identifiers and reserved words. Design the transition diagrams for them. [7M]

(OR)

2. a) Explain various phases of compiler with the example: i:=i*70+j+2. [10M]
   b) Construct DFA equivalent to regular expression r= b (a + b)* a (ab)*. [4M]

UNIT-II

3. a) Construct LL(1)parsing table for the following grammar: [10M]
   S → L = R, S → R, L → *R, L → id, R → L
   b) Define Ambiguous grammar? Explain it with an Example. [4M]

(OR)

4. a) Construct the predictive parser for the following grammar [7M]
   S → (L) | a L →L,S | S.
   b) State and explain the rules used to compute first and follow functions with the help of E→E+T|T T→T*F|F F→F*|a|b. [7M]

UNIT-III

5. a) What is an intermediate code? Explain different types of intermediate codes forms and represent the following statement in different forms: W = (A+B) – (C+D) + (A + B + C) [7M]
   b) Differentiate between Synthesized and Inherited attributes with suitable examples. [7M]

(OR)

6. a) Explain the structure of LR parses and various functions of it. [4M]
   b) S→L=R|R L→* R|id R→L. For the given grammar construct SLR parse table. Explain Rules. [10M]

UNIT-IV

7. a) What is code optimization? Explain about various levels and types of optimizations. [7M]
   b) What is meant by activation of procedure? How it can be represented with activation tree and record? Explain with quick sort example. [7M]

(OR)

8. a) What is a Flow Graph? Explain how a given program can be converted in to a Flow graph? [7M]
   b) With an example explain the following loop optimization techniques: (i) Code motion (ii) Induction variable elimination and (iii) strength reduction. [7M]

UNIT-V

9. a) Write the algorithm for a simple code generator. [7M]
   b) Discuss about register allocation and assignment in target code generation. [7M]

(OR)

10. a) Explain various issues that affect the efficiency of generated code. [7M]
    b) What are the different object code forms in code generation and explain. [7M]

Code No: R2031421 **R20** SET - 2

III B. Tech I Semester Regular Examinations, Dec/Jan-2022-23
COMPILER DESIGN
[**Common to** CSE(AIML), CSE(AI), CSE(DS), CSE(AIDS), AIDS, AIML,CSD]
Time: 3 hours Max. Marks: 70

Answer any **FIVE** Questions **ONE** Question from **Each unit**
All Questions Carry Equal Marks
*****

UNIT-I

1. a) Explain how lex program will perform the lexical analysis for the following patterns in C: Identifiers, comments, numerical constants and arithmetic operators. [7M]
   b) Construct NFA equivalent to regular expression r= (a + b)* ab (aa+bb) and convert it into DFA. [7M]

(OR)

2. a) Consider the following Conditional statement: if (x > 3) then y = 5 else y = 10; How does lexical analyzer help the above statement in process of compilation? [7M]
   b) Define Regular Expression. Explain about the Properties of Regular Expressions. Give examples. [7M]

UNIT-II

3. a) Check whether the grammar is LL (1) or not, and construct a predictive parsing table for following grammar: S -> iEtSS$_1$/ a, S$_1$→eS / ε, E → b. [7M]
   b) Define Ambiguous Grammar? Check whether the grammar S→aAB, A→bC/cd, C→cd, B→c/d, is Ambiguous or not? [7M]

(OR)

4. a) Parse the input string intid,id; using predictive parser for the grammar:
   S → TL;
   T →int | float
   L →L,id | id [7M]
   b) What is Dangling ELSE ambiguity? How it can be solved with LR parsers? Explain with an example. [7M]

UNIT-III

5. a) Write the quadruple, triple, indirect triple for the expression -(a*b) + (c+d)-(a+b+c+d). [7M]
   b) For the grammar below: E → E + T | T, T → num .num | num Give an SDD to determine the type of each term T and expression E. [7M]

(OR)

6. a) Differentiate LR(0) and LR(1) items with examples [4M]
   b) S→ L=R|R L→ *R|id R→L construct CLR(1) parser and explain the procedure. [10 M]

UNIT-IV

7. a) What is a leader of basic block? Write and explain the algorithm used to find leaders. Draw flow graph for matrix multiplication. [7M]
   b) Explain in detail about global common sub expression elimination technique. [7M]

(OR)

8. a) Discuss in detail the role of dead code elimination and strength reduction during code optimization of a compiler [7M]
   b) Draw and explain the Runtime memory organization static storage allocation strategy with pros and cons. [7M]

UNIT-V

9. a) Give an example to show how DAG is used for register allocation. [7M]
   b) What are the object code forms? Explain the issues in code generation. [7M]

(OR)

10. a) Discuss about register allocation and assignment in target code generation. [7M]
    b) Discuss about how to write a simple code generation algorithm. [7M]

Code No: R2031421 **R20** SET - 3

III B. Tech I Semester Regular Examinations, Dec/Jan-2022-23
COMPILER DESIGN
[**Common to** CSE(AIML), CSE(AI), CSE(DS), CSE(AIDS), AIDS, AIML,CSD]
Time: 3 hours Max. Marks: 70

Answer any **FIVE** Questions **ONE** Question from **Each unit**
All Questions Carry Equal Marks
*****

UNIT-I

1. a) Discuss about the role of Lexical analyzer, tokens and patterns. [7M]
   b) Explain bootstrapping a compiler with suitable diagrams. [7M]

(OR)

2. a) How to generate object code for X=Y+Z*15 through different phases of compiler? [10M]
   b) Construct a Finite Automaton for the Regular Expression (00+11)*010*10. [4M]

UNIT-II

3. a) Verify whether the following grammar is LL(1) or not? [7M]
   E → E + T | T
   T → T* F / F
   F → (F) |a|b.
   b) Construct recursive descent parser for the given grammer. [7M]
   bexpr →bexpr OR bterm | bterm
   bterm →bterm AND bfactor | bfactor
   bfactor → NOT bfactor | (bexpr) | TRUE | FALSE

(OR)

4. a) Explain the rules to perform preprocessing steps of top down parser. Explain with given grammar G. S →Aa| bAc| Bc| bBa A → d B→ d. [7M]
   b) Discuss about error recovery strategies in predictive parsing. [7M]

UNIT-III

5. a) Differentiate inherited and synthesized attributes with an example. [7M]
   b) Write Syntax directed definition for constructing syntax tree of an expression derived from the grammar E → E + T | E – T | T T →(E) | id | num. [7M]

(OR)

6. a) S→L=R|R L→*R|id R→L construct LALR(1) parser. [10M]
   b) With LALR(1) parse table constructed above check the acceptance of any input string. [4M]

UNIT-IV

7. a) Explain storage allocation strategies used for recursive procedure calls. [7M]
   b) Explain in brief about peephole optimization techniques. [7M]

(OR)

8. a) What are the principle sources of optimization? Give the classification of code optimization. [7M]
   b) Describe the various fields in an activation record. [7M]

UNIT-V

9. a) Write the algorithm for a simple code generator. [7M]
   b) Discuss about register allocation and assignment in target code generation. [7M]

(OR)

10. a) Explain various issues that affect the efficiency of generated code. [7M]
    b) What are the different object code forms in code generation and explain. [7M]

Code No: R2031421 **R20** SET - 4

III B. Tech I Semester Regular Examinations, Dec/Jan-2022-23
COMPILER DESIGN
[**Common to** CSE(AIML), CSE(AI), CSE(DS), CSE(AIDS), AIDS, AIML,CSD]
Time: 3 hours Max. Marks: 70

Answer any **FIVE** Questions **ONE** Question from **Each unit**
All Questions Carry Equal Marks
*****

UNIT-I

1. a) Write regular expressions for the following languages: Explain operations on Regular expressions. [7M]
   i) All strings over the English alphabet that contain the five vowels in order.
   ii) All strings of a's and b's that do not contain the subsequence abb.
   b) Define lexeme, token and pattern. Identify the lexemes that make up the tokens in the following program segment. Indicate corresponding token and pattern. void swap(inti, int j) { int t; t=i; i=j; j=t; } [7M]

(OR)

2. a) Explain the various phases of a compiler in detail. Also write down the output for the following expression after each phase a: =b*d+20.5. [10M]
   b) Describe the languages denoted by the following regular expressions: [4M]
   (i) (a|b)*a(a|b)(a|b. (ii) a*ba*ba*ba*

UNIT-II

3. a) Construct the LL(1) parser for G: S->(L)|a L-> L,S|S and check the acceptance of input string (a,(a,a)) [7M]
   b) Compute FIRST and FOLLOW for the grammar: S -> S S + \ S S * \ a. [7M]

(OR)

4. a) Eliminate left recursion in the following grammar A → ABd | Aa | a B → Be | b. [7M]
   b) Describe the structure of non recursive predictive parser and error recovery strategies. [7M]

UNIT-III

5. a) Construct the collection of LR(0) item sets and draw the go to graph for the grammar S → S S | a | ϵ. Indicate the conflicts (if any) in the various states of the SLR parser. [7M]
   b) Explain in detail how an L-attributed grammar can be converted into a translation scheme. [7M]

(OR)

6. a) Discuss the evolution order of SDTs. Also write its applications [7M]
   b) Write the SDD for a simple type declaration and draw the annotated parse tree for the declaration float a, b, c. [7M]

UNIT-IV

7. a) Explain in brief about Heap Storage allocation strategy. [7M]
   b) Explain with suitable example various sources of loop optimization. [7M]

(OR)

8. a) Generate basic blocks and flow graph for quick sort algorithm. [7M]
   b) Discuss the role of structure preserving transformations in code optimization. [7M]

UNIT-V

9. a) Give an example to show how DAG is used for register allocation. [7M]
   b) What are the object code forms? Explain the issues in code generation. [7M]

(OR)

10. a) Discuss about register allocation and assignment in target code generation. [7M]
    b) Discuss about how to write a simple code generation algorithm. [7M]