Code No: R2032052 **R20** SET -1

III B. Tech II Semester Regular Examinations, July -2023

COMPILER DESIGN

(Computer Science and Engineering)

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 the phases of a compiler indicating the inputs and outputs of each phase in translating the statement "a=p+r*36.0". [7M]
   b) Discuss about the role of lexical analyzer. Explain with program. [7M]

(OR)

2. a) Explain various data structures used in lexical analysis. [7M]
   b) Write a Regular Expression for identifier, reserved words & relation operators. Design a transition diagram for each of them. [7M]

UNIT-II

3. a) Explain the role of parser. Explain types of grammars used for parsing. [7M]
   b) Write an algorithm for constructing a predictive parsing table. Give Example [7M]

(OR)

4. a) What is an ambiguous grammar? Write a procedure to eliminate the same with an example. [7M]
   b) Consider the following grammar [7M]
   S → (L) |a L → L, S |S
   Construct leftmost and Right most derivations and parse trees for the following sentences:
   i. (a,(a,a)) ii. (a,((a,a),(a,a))).

UNIT-III

5. a) Explain the structure of the LR Parsers and Difference between LR and LL Parsers. [7M]
   b) What is an LR(0) item? Construct an SLR parsing table for the grammar G: S→ L=R |R, L → *R | id, R → L. Is it SLR(1) grammar? [7M]

(OR)

6. a) What are different intermediate code forms? Discuss different Three Address code types and implementations of Three Address statements. [7M]
   b) Write a note on simple type checker and list the different types of type checking. [7M]

UNIT-IV

7. a) Explain various storage allocation strategies with its merits and demerits. [7M]
   b) Define activation records. Explain how it is related with runtime storage allocation. [7M]

(OR)

8. a) What is runtime stack? Explain the storage allocation strategies used for recursive procedure calls. [7M]
   b) What is a flow graph? Explain how flow graph can be constructed for a given program. [7M]
   Main()

```
{    int sum, n, i;
     sum=0;
     for i:=1 to n do
       sum:=sum+i;
     write(sum);
}
```

UNIT-V

9.	a)	What is an induction variable, invariant variable, deadcode? Explain with an example.	[7M]
	b)	Discuss Global Register Allocation in code generation.	[7M]
		(OR)	
10.	a)	Give an example to show how DAG is used for register allocation.	[7M]
	b)	Generate code for the following C statements: i) x=f(a)+f(a) ii) y=x/5;	[7M]

Code No: R2032052 **R20** SET -2

III B. Tech II Semester Regular Examinations, July -2023

COMPILER DESIGN

(Computer Science and Engineering)

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 the boot strapping process with suitable examples and diagrams. [7M]
   b) Construct an FA equivalent to the regular expression (0+1)*(00+11)(0+1)* [7M]

(OR)

2. a) Write about tokens generated by lexical analyzers. Describe the lexical errors and various error recovery strategies with suitable examples. [7M]
   b) Define Regular Expression. Explain the properties of Regular Expressions. Discuss with suitable examples. [7M]

UNIT-II

3. a) Compute FIRST and FOLLOW for the grammar: [7M]
   S → S S + \ S S * \ a.
   b) Write about various types of top down parsing. Discuss about the error recover in predictive parsing. [7M]

(OR)

4. a) Give an algorithm to eliminate productions containing useless symbols and ambiguous productions from a grammar. [7M]
   b) Construct predictive parse table for the following grammar. [7M]
   E → E + T/T
   T → T *F/F
   F → F /a/b

UNIT-III

5. a) List and explain different types of LR Parsers. Differentiate LR(0) and LR(1) items and their associated parsers. [7M]
   b) Construct Canonical LR parsing table for the following grammar. S→L=R | R [7M]
   L→*R | id
   R→L

(OR)

6. a) Compare and contrast SLR with LALR. Show the following grammar is LALR(1) [7M]
   S→ Aa | bAc | dc | bda
   A→ d
   b) What do you mean by attributed grammars? Discuss the translation scheme for Converting an infix expression to its equivalent postfix form. [7M]

UNIT-IV

7. a) What are the principles associated with designing calling sequences and the layout of activation records? [7M]
   b) What is the role of code Optimizer in compiler? Is it a mandatory phase? Explain the various sources of optimization. [7M]

(OR)

8. a) Explain how data flow equations are set up and solved for improving code. [7M]

   b) Discuss basic blocks and flow graphs with an example. [7M]

UNIT-V

9. a) Generate code for the following C program using any code generation algorithm. [7M]

```
main()
{
 int I;
 int a[10];
 while(i<=10)
 a[i]=0;
}
```

   b) Explain the main issues in code generation. How to handle them? Discuss. [7M]

(OR)

10. a) Discuss about register allocation and assignment in target code generation. [7M]

    b) Discuss how induction variables can be detected and eliminated from the given intermediate code [7M]

```
B2: i:= i+1
    t1:=4*j
    t2:=a[t1]
    if t2<10 goto B2
```

Code No: R2032052 **R20** SET -3

III B. Tech II Semester Regular Examinations, July -2023

COMPILER DESIGN

(Computer Science and Engineering)

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 various building blocks used to design a language translator. [7M]
   b) Differentiate between [7M]
   i) Phase and a pass ii) single-pass and multi-pass compiler.

(OR)

2. a) What is LEX? Discuss the usage of LEX in Lexical Analyzer generation. [7M]
   b) Construct a Finite Automata and Scanning algorithm for recognizing identifiers, numerical constants in C language. [7M]

UNIT-II

3. a) Define Context Free Grammar. Explain how it is suitable for parsing? Explain the recursive descent parser with example. [7M]
   b) Design a non-recursive predictive parser for the following grammar: [7M]
   S → AaAb | BbBb
   A → e
   B → e where a, b, e are terminals.

(OR)

4. a) Given the following grammar: E -> E + E | E - E | E * E | E / E | - E | int Show two different left-most derivations with the help of parse trees for the string int + int * int / int. What does this tell you? [7M]
   b) Explain left recursion and left factoring with examples. [7M]

UNIT-III

5. a) Define LR(k) parser. Explain the model of LR parser and various functions used in it for parser construction. [7M]
   b) How to handle ambiguity through LR parsers? Discuss about the Dangling – Else ambiguity. [7M]

(OR)

6. a) Give syntax directed translation scheme for simple desk circulator. [7M]
   b) Show that the following grammar: [7M]
   S → Aa|bAc|Bc|bBa
   A → d
   B → d
   Is LR(1) but not LALR(1).

UNIT-IV

7. a) Give the general structure of an activation record? Explain the purpose of each component involved in it. [7M]
   b) Explain various machine independent code optimization techniques. [7M]

(OR)

Code No: R2032052 **R20** SET -3

8. a) Write a short note on peephole optimization and various operations used in it. [7M]
   b) Describe Loop unrolling? Describe its advantage with your own examples. [7M]

UNIT-V

9. Explain the code generation algorithm in detail with an example. [14M]

(OR)

10. a) Discuss basic blocks and flow graphs with an example [7M]
    b) Generate code for the following: [7M]
    i) x=f(a)+f(a)+f(a) ii) x=f(f(a)) iii) x=++f(a) iv) x=f(a)/g(b,c)

Code No: R2032052 **R20** SET -4

III B. Tech II Semester Regular Examinations, July -2023

COMPILER DESIGN

(Computer Science and Engineering)

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 about Phases of a compiler. Explain each with an example. [8M]
   b) Explain about Input Buffering in lexical Analyzer with an example. [6M]

(OR)

2. a) Describe the need and functionality of linkers, assemblers and loaders. [7M]
   b) State the steps to convert a regular expression to NFA. Explain with an example. [7M]

UNIT-II

3. a) What are the preprocessing steps required for constructing Predictive parsing table. Explain with example. [7M]
   b) Define a Parser. What is the role of grammars in Parser construction? Construct the Predictive parsing table for the grammar G: E → E+T |T, E →T*F |F, F → (E) |id. [7M]

(OR)

4. a) What is an LL(1) grammar? Can you convert every context free grammar into LL(1). How to check the grammar is LL(1) or not? Explain the rules, [7M]
   b) Consider the following grammar [7M]
   E → T + E|T
   T→V*T|V
   V → id
   Write down the procedures for the non-terminals of the grammar to make a recursive descent parser.

UNIT-III

5. a) Write the rules used to construct SLR Parser. Give example. [7M]
   b) Generate the three address code for the following code fragment. [7M]
   a = b + 1 x = y + 3 y = a / b a = b + c

(OR)

6. a) What is an LALR(1) grammar?. Construct LALR parsing table for the following grammar: S→ CC, C → cC , C → c|d . [7M]
   b) Write and explain the LR Parsing algorithm.. [7M]

UNIT-IV

7. a) Explain static and stack storage allocations? [7M]
   b) Translate the arithmetic expression a[i]=b*c-b*d into a syntax tree, quadruples and triples. [7M]

(OR)

8. a) Write pseudocode for finding sum of 'n' numbers. And identify basic blocks then construct the flow graph for it. Explain the rules used for this. [7M]
   b) Explain the following peephole optimization techniques; [7M]
      i) Elimination of Redundant Code
      ii) Elimination of Unreachable Code

UNIT-V

9. a) Explain the main issues in code generation. [7M]
   b) Explain the following terms: [7M]
      i) Register Descriptor ii) Address Descriptor iii) Instruction Costs

(OR)

10. a) Give an example to show how DAG is used for register allocation. [7M]
    b) Generate code for the following C program using any code generation algorithm. [7M]

```
main()
{
 int I;
 int a[10];
 while(i<=10)
 a[i]=0;
}
```