Reg. No. : □□□□□□□□□□□□

Question Paper Code : 30455

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2024.

Sixth Semester

Computer Science and Engineering

CS 8602 – COMPILER DESIGN

(Common to : Computer Science and Business Systems)

(Regulations 2017)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. State the difference between compiler and interpreter.
2. Define the role of input buffering in Lexical Analysis.
3. What are the components of a context free grammar?
4. List the possible conflicts that occur in CLR and LALR.
5. Show that three address code sequence for the assignment statement. z = (a–b)+ (c–d)* (e–f)
6. Define syntax tree.
7. How the memory is managed during run-time?
8. Define leap memory.
9. What are the characteristics of peephole optimization?
10. What is a Directed Acyclic Graph?

PART B — (5 × 13 = 65 marks)

11. (a) Construct minimum-State DFA for the regular expression: $(a \mid b) * ab\,(a \mid c)$. Give $\Sigma = \{a,b,c\}$

Or

(b) Consider the language "*All strings with exactly two 'a's*" $\Sigma = \{a, b\}$.

(i) Construct Regular Expression. (3)

(ii) Construct NFA using Thompson's construction method. (5)

(iii) Construct DFA from the NFA. (5)

12. (a) Create SLR parsing table for the following grammar.

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

Identify whether the Grammar is parsed by SLR. If yes, parse the string (a,(a, (a, a))) or Justify why not parsed by SLR.

Or

(b) Construct LALR parsing table.

$S \rightarrow Aa \mid bAc \mid dC \mid bda$

$A \rightarrow d$

Identify whether the Grammar is parsed by LALR. If yes, parse the string bdc or Justify why not parsed by LALR.

13. (a) (i) Generate intermediate code for the given program segments

```
int main ( )
{
int n, k = 0;
scanf (“%d”, &n);
for (i=2; i < n; i++)
{
          if ((n%i) = = 0)
          break;
}
k = 1;
if (i = = n)
          printf (“number is prime”);
else
          printf (“number is not printed”);
}
```

(8)

(ii) Construct Quadruples and Triples for storing above three address statement. (5)

Or

(b) (i) Explain t he process of back patching. (3)

(ii) Illustrate back patching in the given Boolean expression with justification

```
if (a! = 10) && (x < 0 | | x >= 9)
    x +=1;
else
    x = 0;
```

(10)

14. (a) Develop a Fibonacci series algorithm to print five integers by using the concepts of activation tree. (13)

Or

(b) (i) Discuss the issues in code generation. (5)

(ii) Explain the design of a simple code generator with a suitable example. (8)

15. (a) Show the code sequence generated by the code generation algorithm (13)

u : = a – c

v : = t + u

d : = v + u / d

Apply peephole optimization technique in a generated code.

Or

(b) Consider the following part of code. (13)

```
int main ( )
{
        int n, k = 0;
        scanf ("%d", &n);
        for (i=2; i < n; i++)
        {
                if ((n%i) = = 0)
                break;
        }
        k = 1;
        if (i = = n)
                printf("number is prime");
        else
                printf("number is not printed");
}
```

Identify the basic blocks and draw dominator tree in the given program and Find the natural loop in the basic block.

PART C — (1 × 15 = 15 marks)

16. (a) The Fahrenheit to Celsius Convertor Application is designed with following formula C = (F – 32)* 5/9.

Where F is a floating point number. Illustrate how this statement is processed in each compliation phases with an output generated from each phase. The usage of Symbol Table and Error handling phase applied in each compliation phases for this statement need to be elaborated. (15)

Or

(b) Construct an operator precedence passer for the given grammar. Using the parser, check whether the string $(a,(a,a))$ is valid. (15)

$L \rightarrow L, S \mid S$

$S \rightarrow a \mid (L)$