

B.Tech III Year II Semester (R13) Regular & Supplementary Examinations May/June 2017

COMPILER DESIGN
(Information Technology)

Time: 3 hours

Max. Marks: 70

PART – A
(Compulsory Question)

- 1 Answer the following: (10 X 02 = 20 Marks)
- Compare compiler and interpreter. Also write advantages and disadvantages of both.
 - How applications of compiler technology improve levels of abstraction in the generation of programming languages?
 - Differentiate recursive descent parsing and predictive parsing.
 - Construct all possible parse trees corresponding to the string $i+j*k$.
 - Explain quadruple and triple with example.
 - What are different forms of target programs?
 - What is dead code? Give suitable example.
 - Give difference between heap storage and hash table.
 - What is peephole optimization?
 - Generate code for $x = *p$ for target machine.

PART – B
(Answer all five units, 5 X 10 = 50 Marks)

UNIT – I

- 2 What is significant use of regulator expression in lexical analysis? Give rules to define regular expression over alphabets and algebraic laws for regular expression.

OR

- 3 How is input buffering solve look ahead problem with sentinels? Support answer with look ahead pseudo code with sentinels.

UNIT – II

- 4 Consider following grammar:

$E \rightarrow EBE$
 $E \rightarrow \text{num}$
 $E \rightarrow (E)$
 $B \rightarrow +$
 $B \rightarrow -$
 $B \rightarrow *$
 $B \rightarrow \backslash$

- Explain why this grammar is suitable to form the basis for recursive descent parsing.
- Use left factoring and left recursion removal to obtain an equivalent grammar that can be used as the basis for a recursive descent parsing.

OR

- 5 Show that following grammar is unambiguous:

$S \rightarrow aSb / bSa / b$ for string 'abbbaabbbbaaab'. Also draw a parse tree.

Contd. in page 2

UNIT – III

6 Describe various representations of three address codes. Translate the expression:
 $-(a + b) * (c + d) + (a + b + c)$ into quadruples and triples.

OR

7 What is S-attribute and L-attribute syntax directed definition (SDD)? Obtain the postfix syntax directed translation for: $S \rightarrow EN$
 $S \rightarrow E + T / E - T / T$
 $T \rightarrow T * F / T / F / F$
 $F \rightarrow (E) / \text{digit}$
 $N \rightarrow ;$

UNIT – IV

8 Explain in detail the strategy for reducing fragmentation in heap memory.

OR

9 Why run time storage management is required? How is simple stack implemented?

UNIT – V

10 Explain code generation algorithm and generate code for $w = (A - B) + (A - C) + (B - C)$.

OR

11 Explain main issues in code generation.
