

[Total No. of Questions - 9] [Total No. of Printed Pages - 3]
(2064)

14717

B. Tech 6th Semester Examination

Compiler Design

CS-6003

Time : 3 Hours

Max. Marks : 100

The candidates shall limit their answers precisely within the answer-book (40 pages) issued to them and no supplementary/continuation sheet will be issued.

Note : Attempt one question from each section A, B, C and D.
Section E is compulsory.

SECTION - A

1. A *cross compiler* is one that runs on a machine to generate target code for another machine. Identify a few cases where such a cross compiler will be useful. (20)
2. What is the structure of a compiler? Specify the role of lexical analyzer in compiler design. (20)

SECTION - B

3. Construct the operator precedence parser for the following grammar:

$S \rightarrow (L)a$

$L \rightarrow L, S | S$

Show the parsing of the string “(a, ((a,a),(a,a)))” using the parser constructed. (20)

14717/900

[P.T.O.]

4. Consider the following grammar

$$E \rightarrow E+T|T$$
$$T \rightarrow T*F|F$$
$$F \rightarrow (E)|id$$

Construct an equivalent grammar with no left recursion. (20)

SECTION - C

5. What is the role of intermediate code generation in overall compiler design? Show the annotated parse tree and code generation process for the following, arithmetic expression:
 $a+(b-c)*d$ (20)
6. Show how the expression $x-2xy$ might be translated into an abstract syntax tree, one address code, two-address code, and three address code. (20)

SECTION - D

7. Specify the necessary and sufficient conditions for performing
- (a) constant propagation
 - (b) dead code elimination
 - (c) loop optimization (20)
8. Describe the structure of a Lex program, and clearly mention the steps involved in translating, compiling and executing a Lex program. (20)

SECTION - E

9. (a) What is the importance of look-ahead operator in lexical analysis phase.
- (b) Describe the steps involved in "Bootstrapping".

- (c) Draw the parse tree for an arithmetic expression $a^*(b+c)$.
- (d) What do you mean by LR(1) parsing?
- (e) Translate the arithmetic expression $a^*(b+c)$ into postfix notation.
- (f) What is type system? Discuss static and dynamic checking of types.
- (g) Name various machine-independent code optimization techniques.
- (h) How addressing modes can be used for reducing the memory access time?
- (i) Explain organizing techniques for searching in a symbol table.
- (j) Differentiate between macros and functions? (10×2=20)