Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 80303

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2016.

Sixth Semester

Computer Science and Engineering

CS 6660 — COMPILER DESIGN

(Common to Sixth Semester Information Technology)

(Regulations 2013)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is a symbol table?

2. List the various compiler construction tools.

3. List the rules that form the BASIS.

4. Differentiate tokens, patterns, lexeme.

5. Construct a parse tree for –(id + id)

6. What is meant by handle pruning?

7. Write down syntax directed definition of a simple desk calculator.

8. List Dynamic Storage allocation techniques.

9. Identify the constructs for optimization in basic block.

10. What are the characteristics of peephole optimization?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Explain the phases of compiler with a neat diagram. (10)

(ii) Write notes on compiler Construction tools. (6)

Or

(b) (i) Explain the need for grouping of phases. (8)

(ii) Explain the various errors encountered in different phases of compiler. (8)

12. (a) (i) Discuss the role of lexical analyzer in detail with necessary examples. (8)

(ii) Discuss how finite automata is used to represent tokens and perform lexical analysis with examples. (8)

Or

(b) (i) Conversion of regular expression (a/b)*abb to NFA. (8)

(ii) Write an algorithm for minimizing the number of states of a DFA. (8)

13. (a) (i) Construct parse tree for the input string w = cad using top down parser. (6)

$S \rightarrow cAd$

$A \rightarrow ab \mid a$

(ii) Construct parsing table for the grammar and find moves made by predictive parser on input id+id*id and find FIRST and FOLLOW. (10)

$E \rightarrow E + T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)/id$

Or

(b) (i) Explain ambiguous grammar $G : E \rightarrow E + E \mid E * E \mid (E) \mid -E \mid id$ for the sentence id+id*id. (6)

(ii) Construct SLR parsing table for the following grammar :

$G : E \rightarrow E + T \mid TT \rightarrow T * F \mid FF \rightarrow (E) \mid id$. (10)

14. (a) (i) A Syntax-Directed Translation scheme that takes strings of a's, b's and c's as input and produces as output the number of substrings in the input string that correspond to the pattern a(a | b)*c+(a | b)*b. For example the translation of the input string "abbcabcababc" is "3".

(1) Write a context-free grammar that generate all strings of a's. b's and c's.

(2) Give the semantic attributes for the grammar symbols.

(3) For each production of the grammar present a set of rules for evaluation of the semantic attributes. (8)

(ii) Illustrate type checking with necessary diagram. (8)

Or

(b) Explain the following with respect to code generation phase.

(i) Input to code generator

(ii) Target program

(iii) Memory management

(iv) Instruction selection

(v) Register allocation

(vi) Evaluation order. (16)

15. (a) (i) Write an algorithm for constructing natural loop of a back edge. (8)

(ii) Explain any four issues that crop up when designing a code generator. (8)

Or

(b) Explain global data flow analysis with necessary equations. (16)

———————