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

Question Paper Code : 60391

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

Sixth Semester

Computer Science and Engineering

CS 2352/CS 62/10144 CS 602 — PRINCIPLES OF COMPILER DESIGN

(Regulations 2008/2010)

(Common to PTCS 2352 – Principles of Compiler Design for B.E. (Part-Time) Fifth Semester – Computer Science and Engineering – Regulations 2009)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Define linear and hierarchical analysis.
2. What is the function of deterministic finite automata?
3. Find the parse tree for the sentence (a, (a, a)).
4. What is "Dangling-else" ambiguity?
5. State the two representation of syntax tree.
6. Define the function of quadruples.
7. What is the function of getreg?
8. Define peephole optimization.
9. What are the criteria for code-improving transformations?
10. What is dead-code elimination?

PART B — (5 × 16 = 80 marks)

11. (a) (i) State and explain the phases of a compiler. (8)

(ii) What are the cousins of the compiler? Explain with suitable example. (8)

Or

(b) (i) What are the issues in lexical analysis? (6)

(ii) Prove that any deterministic finite automaton for the regular expression $(a \mid b)^* a(a \mid b)\,(a \mid b) \ldots .(a \mid b)$, where there are $n - 1$ $(a \mid b)$'s at the end, must have at least 2^n states. (10)

12. (a) (i) What are the different strategies that a parser can employ to recover from a syntactic error? Explain them in detail. (8)

(ii) State and explain the algorithm for eliminating left recursion with example. (8)

Or

(b) (i) Show that no left-recursive grammar can be LL(1). (8)

(ii) Explain type checking of expression and functions with example. (8)

13. (a) (i) What is three-address code. Explain the types of three address statement with example. (8)

(ii) How to keep track of scope information during declaration? Explain it with example. (8)

Or

(b) (i) State and explain the methods of translating boolean expressions. (8)

(ii) Explain how batch patching can be used to generate code for Boolean expression and flow control statements in one pass. (8)

14. (a) (i) Explain the primary structure-preserving transformation on basic blocks with example. (8)

(ii) What are the issues in the design of code generation. Explain in detail with example. (8)

Or

(b) (i) State and explain the DAG representation of Basic blocks with example. (8)

(ii) Explain the steps carried out for generating code from dags with suitable example. (8)

15. (a) (i) Explain the function-preserving transformation with example. (8)

(ii) State and explain the algorithm for constructing the natural loop of a back edge. (8)

Or

(b) (i) Explain the procedure for global common sub expression elimination. (8)

(ii) Explain the data-flow analysis of structure programs with example. (8)