Reg. No. :

Question Paper Code : 11160

B.E./B.Tech. DEGREE EXAMINATION, MAY/JUNE 2014.

Seventh Semester

Computer Science and Engineering

080230045 — PRINCIPLES OF COMPILER DESIGN

(Regulation 2008)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. List the principal facilities provided by compiler-compilers.

2. Define the term cross-compiler.

3. What are the conditions that distinguish a DFA from an NFA?

4. Write a regular expression to represent strings from $\Sigma = \{0,1\}$ that has no two consecutive 1's.

5. Given a left recursive pair of productions $A \to A\,\alpha / \beta$, eliminate the left recursion.

6. Given a grammar $S \to (S) / SS / \varepsilon$, find the leftmost derivation of $(\ )((( ))\,(\ ))$.

7. What is the postfix notation of intermediate code?

8. Define the term back patching.

9. What is a flow graph?

10. State algebraic simplification in code optimization.

PART B — (5 × 16 = 80 marks)

11. (a) What are the phases of a compiler? Explain the function of each phase in the compilation process.

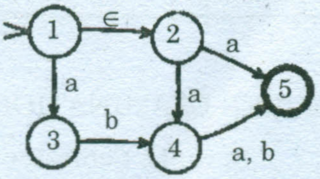
Or

(b) (i) Discuss about the various compiler construction tools. (10)

(ii) Explain the process of Bootstrapping (6)

12. (a) (i) Explain the subset construction algorithm for constructing a DFA from an NFA. (6)

(ii) Using the subset construction algorithm construct an equivalent DFA for the given NFA. (10)



Or

(b) Explain the process of constructing a NFA from a regular expression. Design an NFA for the regular expression aa^* / bb^*.

13. (a) (i) Explain in detail about operator precedence parsing. (8)

(ii) Discuss how to compute the precedence relations for the given operator precedence grammar $E \rightarrow E + T / T$, $T \rightarrow T^* F / F$, $F \rightarrow (E) / id$. (8)

Or

(b) (i) Explain in detail about predictive parsing. (8)

(ii) Given the grammar $E \rightarrow TE'$ $E' \rightarrow + TE' / \varepsilon$ $T \rightarrow FT'$ $T' \rightarrow *FT' / \varepsilon$ $F \rightarrow (E) / id$, construct the predictive parsing table. (8)

14. (a) Explain in detail about the different kinds of intermediate codes with suitable examples.

Or

(b) Explain in detail about the syntax directed translation scheme for simple assignment statements.

15. (a) Discuss in detail about loop optimization.

Or

(b) (i) Explain in detail the DAG representation of Basic Blocks. (8)

(ii) Discuss about Peephole optimization. (8)

———————