G 6181

M.E. DEGREE EXAMINATION, MAY/JUNE 2007.

Second Semester

Computer Science and Engineering

CS 1653 — COMPILER DESIGN

(Regulation 2005)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Comment on the efficiency of the compiler if the number of passes in compilation is increased.

2. Why the regular expressions are used though the lexical constructs of any programming language can be described using context free grammars?

3. Write regular expression for recognizing the set of all strings w such that $|w| \bmod 3 = 2$, w contains only 0s.

4. What is meant by state minimization of a DFA?

5. Define ambiguous sentence and ambiguous grammar.

6. What are the advantages of LR(1) items over LR(0) items?

7. What is meant by synthesized and inherited translations?

8. Suggest any two solutions to minimize/avoid temporaries generated during three-address code generation.

9. Give any two examples for strength reduction of operations.

10. How the leaders for basic blocks are identified?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Write about the various cousins of any compiler. (8)

(ii) Discuss any four translators and their advantages in detail. (8)

Or

(b) Explain all the phases of a compiler with the help of an example. (16)

12. (a) (i) Construct the minimal state deterministic finite automata for the regular expression $(a/b)^* a\,(a/b)(a/b)$. (10)

(ii) Draw a transition diagram to recognize the following C language operators ++, --, +=, -=, = =, !=. (6)

Or

(b) (i) List the various types of lexical errors. How does a scanner recover from lexical errors? (6)

(ii) Show that the regular expressions $((\in | a)\, b^*)^*$ and $(a^* | b^*)^*$ are denoting the same language by constructing their minimal state DFA. (10)

13. (a) (i) Consider the grammar with production rules $S \rightarrow (L) | a$ and $L \rightarrow L, S \mid S$. Remove left recursion in this grammar and show this grammar is LL(1) by constructing predictive parsing table. (12)

(ii) Explain the predictive parsing algorithm. (4)

Or

(b) Show that the following grammar is LR(1) but not LALR(1). (16)

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$

$A \rightarrow d$

$B \rightarrow d$

14. (a) (i) Explain the translation scheme for processing the declarations of variables including arrays and records. (12)

(ii) Write a method for reusing temporary names. (4)

Or

(b) Translate the following assignment statement into three address codes. (16)

$$A[i, j] := B[i, j] + C[A[k, 1]] + D[i + j]$$

15. (a) (i) Write the code generation algorithm to produce assembly language from three-address code. (8)

(ii) Explain the issues in the design of a code generator. (8)

Or

(b) Discuss the principal sources of optimization in detail. (16)