NOVEMBER/DECEMBER 2018

MCS21 — COMPILER DESIGN

Time : Three hours Maximum : 75 marks

SECTION A — (5 × 6 = 30 marks)

Answer ALL questions.

1. (a) Define NFA. Explain with examples.

Or

(b) Write short notes on lexical analyzer.

2. (a) Discuss about Recursive-Descent Parsing.

Or

(b) Consider the grammar

$E \rightarrow E - E, E \rightarrow E * E, E \rightarrow E / E.E \rightarrow id$.
Construct leftmost / derivation and rightmost derivation for the sentence id-id/id.

3. (a) What is a syntax tree? Construct a syntax tree for if-then-else.

Or

(b) Elaborate on overloading of operators.

4. (a) Write short notes on Activation records.

Or

(b) Differentiate call-by-value and call-by-reference.

5. (a) Write short notes on Register Allocation.

Or

(b) How a DAG is used to represent basic block? Explain.

SECTION B — (3 × 15 = 45 marks)

Answer any THREE questions.

6. Write the algorithm to construct NFA from Regular Expression.

7. Discuss about Bottom-up parsing.

8. Give a detailed account on assigning space at compiler-construction time.

9. What are the various storage allocation strategies? Explain.

10. Write the steps followed in code-generation algorithm and the function *getreg*.
