2053

B.E. (Computer Science and Engineering)
Sixth Semester
CS-604: Compiler Design

Time allowed: 3 Hours   Max. Marks: 50

NOTE: *Attempt* *five* *questions in all, including Question No. I which is compulsory*
and selecting two questions from each Section.

x-x-x

I.   Write short answers of the following:
   a. What is left recursion?
   b. What do you mean by structural and name equivalence of types?
   c. What is shift-reduce conflict?
   d. What is loop jamming?
   e. What is an activation record? List various informative fields that it contains.   **(5×2=10)**

Section-A

II.
   a. Describe the structure of compiler. Why the process of compilation is divided into various phases?
   b. What is left factoring? What are the problems that arise due to left factoring in the design of top-down parsers? Write an algorithm to remove left factoring from a given grammar.   **(5,5)**

III.
   a. Describe in brief the role of lexical analyzer.
   b. Construct a predictive parsing table for the following grammar, where S is the start symbol:
      $S \rightarrow iEtST \mid a$
      $T \rightarrow eS \mid \in$
      $E \rightarrow b$ **(5,5)**

IV.
   a. What is a context-free grammar? Illustrate with an example the different components of a context-free grammar. What are the advantages of using context-free grammar to specify a language?
   b. Define handle. What are the issues involved in handle pruning? How is handle pruning implemented in an operator precedence parser?   **(5,5)**

Section-B

V.
   a. What do you mean by intermediate code? What are the advantages of generating it? Describe various methods for representing three address statements.
   b. Describe in detail principal sources of code optimization.   **(5,5)**

VI. Define a syntax directed translation that records the maximum number of nested Kleene star operators of a regular expression R in its attribute R. depth. The grammar is given:
   $R \rightarrow a \mid b \mid \varepsilon \mid R_1 \circ R_2 \mid R_1 + R_2 \mid R_1 \mid (R_1)^*$. For example, the regular expression $(a)^* \mid ((b)^* + a)^*$ has depth 2.   **(10)**

VII.
   a. What are basic blocks? Describe the steps of partitioning a sequence of three-address statements into list of basic blocks.
   b. Describe in detail different storage allocation strategies along with merits and demerits of each.   **(5,5)**

x-x-x