Reg. No.										

Manipal Institute of Technology, Manipal

(A Constituent Institute of Manipal University)

VI SEMESTER B.TECH (COMPUTER SCIENCE AND ENGINEERING)

END SEMESTER EXAMINATIONS, MAY/JUNE 2016

SUBJECT: LANGUAGE PROCESSORS (CSE 302)
REVISED CREDIT SYSTEM
4-5-2016

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates

- Answer **ANY FIVE FULL** questions.
- Missing data, if any, may be suitably assumed.

1A. Consider the statement DECLARE (ARG1, ARG2) in a programming language in which DECLARE either refers to a function name or an array-name. In order to recognize DECLARE as a lexeme, how many character lookaheads are essential? Show all such lookahead characters. Outline the two algorithms for moving the *forward* pointer using two-buffer scheme, one that uses sentinels and the other one without sentinels. How many tests are needed in each algorithm to advance the *forward* pointer? Explain these tests. 4M

1B. Show an architecture diagram to model a lexical analyzer generated by LEX program and mention the task of each component. 2M

1C. Discuss the procedure involved in constructing regular expression r_i from regular definition d_i such that definitions are not recursive. Show the application of procedure to the following sequence of regular definitions and write the regular expression denoted by each of $X_0, X_1, \ldots X_n$ separately. Show the conversion of regular expression denoted by X_n into an NFA using Thomson algorithm.

$X_0 \rightarrow a \mid b$

$X_1 \rightarrow X_0\ X_0$

$X_2 \rightarrow X_1\ X_1$ and so on … finally X_n is shown below

$X_n \rightarrow X_{n-1}\ X_{n-1}$ 4M

2A. Considering a source program coded in a high level language as input, what language processors are available to translate the source program into a target program? Give the block diagram for each one of these. 2M

2B. For the following grammar, clearly describing all the steps required in predictive parsing, show the parsing action for the input string 5 + a + b

$E \rightarrow 5 + T \mid 3\text{-}T$

$T \rightarrow V \mid V{*}V \mid V + V$

$V \rightarrow a \mid b$ 5M

2C. State conditions for identifying a LL(1) grammar using FIRST and FOLLOW rules. Apply the rules to the grammar in Question (2B), showing all the steps. 3M

3A. For the grammar S → SS | a | ε, clearly describing all the steps required in SLR parsing, construct LR(0) DFA and the SLR parsing table. 3M

Note the marking for all DFAs: Mark the states starting from 0 and provide a proper layout for the DFAs so that DFA evolves from left to right side. From each state, draw the transitions for grammar symbols in the order in which the grammar symbols appear within that state and label the edges accordingly.

3B. For the following grammar, construct LR(1) DFA and LR(1) parsing table. Is this grammar LR(1)? Is this grammar LALR(1)? 5M

S → aAd | ace | bAe
A → c

3C. What is meant by a viable prefix? Generate longest viable prefix from the DFA generated in Question (3B). 2M

4A. For the following grammar, write the syntax directed definition and show the annotated parse tree for computing the value of expression (3+4) * (5+6) 3M

S → E
E → E + T | T
T →T * F | F
F → (E) | digit

4B. Write the syntax directed definition (SDD) for counting the number of balanced parentheses. Apply the SDD to the input string ((())) and construct the annotated parse tree. 3M

4C. For the grammar, write SDD for creating the syntax tree. Show the steps in constructing Abstract Syntax Tree (AST) for the expression a := b* - (c – d) + b* - (c – d) and display AST. Also display Directed Acyclic Graph (DAG). 4M

S → id := E
E → E + E
E → E – E
E → E * E
E → – E
E → (E)
E → id

5A. Translate the expression (a+b) * (c+d) + (a+b+c) into i) quadruples ii) triples iii) indirect triples. Sequentially store the instructions starting from 1. 3M

5B. Suppose nesting of procedure calls are invoked such that a procedure *m* calls procedure *n* then what different cases exist for the activation of n? For each case, discuss activation of *m*. State relationship between the sequence of procedure calls/returns with activation tree. 4M

5C. Generate code for the following statements assuming all variables are stored in memory location. Determine the total cost of the generated code by marking the costs of each instruction separately. if x < y goto L 3M

6A. Translate the following program into three address statements by sequentially numbering each three address statement starting from 1. Identify leaders and basic blocks and show the construction of the flow graph. In the flow graph, mark all basic blocks and identify the loops. 4M

```
while (p< q)
          if(m<n)
            r = s +t ;
          else  r = s-t;
```

6B. What are the two ways to represent the contents of a basic block? Justify the need for its efficient representation. 2M

6C. Discuss the functions of both phases of the assembler during translation of a source program. 4M

****