RAMAIAH UNIVERSITY OF APPLIED SCIENCES

Faculty of Engineering & Technology
Semester End Examination Question Paper – B. Tech.

Department: Computer Science and Engineering

Programme: B. Tech in Computer Science and Engineering

Semester/Batch: 6th / 2016

Date of Examination: 20-05-2019

Course Code: CSC310A

Course Title: Compilers

Semester End Examination-Theory

INSTRUCTIONS TO STUDENTS:

1. Answer any **FIVE** full questions
2. Use only SI units
3. Use of non-programmable scientific calculator is permitted
4. Use of data handbook permitted wherever applicable
5. Missing data may be appropriately assumed
6. Indicate the question number (including its part as applicable) for your answers

Maximum Duration: 3 Hours **Maximum Marks: 100**

IMPORTANT:
You may take this question paper away at the end of the examination. Please keep it in a safe place for future reference

Question 1 **(8 + 3 + 3 + 6 = 20 Marks)**

a. Describe the phases of a compiler.
b. Explain features of a lexical analyser.
c. Justify the separation of lexical analysis and parsing phases.
d. Construct LL(1) parsing table for the given grammar:

$$S -> iEtST|a$$
$$T -> eS|\varepsilon$$
$$E -> b|a$$

Question 2 **(4 + 4 + 12 = 20 Marks)**

a. Describe loop unrolling and redundancy elimination in expression evaluation, with an example each.
b. Prove that the given grammar is ambiguous and convert it to an unambiguous grammar:

$$BExp -> BExp\ OR\ BExp/$$
$$BExp\ AND\ BExp/$$
$$NOT\ BExp/$$
$$True/$$
$$False$$

c. Explain the algorithm of CLR(1) parser for the following grammar by parsing the given string: $((id + id) - (id - id))$

$$E -> E + T/T$$
$$T -> T - F/F$$
$$F -> (E)/id$$

Question 3 **(3 + 4 + 5 + 8 = 20 Marks)**

a. Identify the precedence and associativity for the expression generated by grammar:

$$E -> E * F$$
$$E \rightarrow F + E/F$$
$$F -> F - F/id$$

b. Explain the differences between SLR(1) and LALR(1) parsers.
c. Perform Syntax Directed Translation (SDT) for type checking by parsing the given string:$(2.5 + 3.7) == 8$ in a top down approach.
d. Perform L - Attributed SDT to determine the amount of memory required to allocate a two-dimensional array by parsing the given string: $int\ a[][] = \{\{1,2,3,4\},\{5,6,7,8\}\};$. Consider the size of *int* to be 12 bytes.

Question 4 **(4 + 8 + 8 = 20 Marks)**

a. Explain the procedure to detect loops in any three-address code.
b. Develop an algorithm of LL(1) parser for the given grammar by parsing the string: $abab$

$$S -> A$$
$$A -> BA/\ \varepsilon$$
$$B -> aB/b$$

c. Identify the conflict(s) in the given grammar and recommend modifications to the grammar. Prove that the resulting grammar is appropriate to build an LL(1) parser:

$$S -> (L)|a$$
$$L -> L + S|R$$
$$R \rightarrow cseAB \mid csaaB$$
$$A \rightarrow c \mid Ac$$
$$B \rightarrow b$$

Question 5 **(14 + 6 = 20 Marks)**

a. Construct the Canonical Set of LR (1) items for the given grammar and verify whether the grammar is appropriate for: i) CLR (1) ii) LALR (1) parser

$$E \rightarrow TE'$$
$$E' \rightarrow +TE'|\varepsilon$$
$$T \rightarrow FT'$$
$$T' \rightarrow * FT'|\varepsilon$$
$$F \rightarrow id$$

b. Consider the given code:

```
for (i=0;i<100;i++){
 q[i]=i;
}
value=200;
x=4;
do
{
item = 10;
value = value + item;
value--;
} while(value<100);
if(x<2)
x=x+2;
```

Identify the suitable optimisation technique(s) and apply it to improve the efficiency of the code.

Question 6 **(4 + 6 + 10 = 20 Marks)**

a. Explain quadruples and indirect triples with an example.

b. Perform L-attributed SDT to generate three address code. Demonstrate that the grammar is L-attributed by parsing the given input: $x = a + b * c$.

c. Consider the given code:

```
n=5;
switch (a+b)
{
case (1): do{
                a = a*i;
                i++;
        } while(i < n);
        break;
case (2):
        while(i<n){
        a=a+i*i;
        i++;
        }
}
```

i. Convert the block of code to three-address code representation.

ii. Identify the loops in the three-address code generated in Q.6b.i.

oooOooo