USN | | | | | | | | | | |

06CS63

Sixth Semester B.E. Degree Examination, June/July 2013

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. Explain the different phases of compiler, with a block diagram. **(10 Marks)**

b. How is input buffering of lexical analyzer implemented? **(04 Marks)**

c. Construct transition diagrams for the following:
   i) Relop
   ii) Identifier
   iii) Unsigned number **(06 Marks)**

2 a. Explain any two error-recovery strategies in parser. **(04 Marks)**

b. Consider the following grammar:
   $E \rightarrow I \mid E + E \mid E*E \mid (E)$
   $I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$
   i) Check whether the grammar is ambiguous or not for the given input string w = a * b + a.
   ii) If ambiguous construct an equivalent unambiguous grammar. **(06 Marks)**

c. Construct LL(1) parsing table for the grammar given below:
   $E \rightarrow E * T \mid T$
   $T \rightarrow id + T \mid id$ **(10 Marks)**

3 a. Define bottom-up parsing. Check the acceptance of the input string "bbb*b++" for the grammar: $E \rightarrow EE + \mid EE* \mid b$ **(04 Marks)**

b. Explain different actions in shift-reduce parser with an example. Also describe the conflicts during shift-reduce parsing. **(06 Marks)**

c. Construct SLR(1) parsing table for the following grammar:
   $S \rightarrow SA|A$
   $A \rightarrow a$ **(10 Marks**

4 a. Write an algorithm for constructing LR(1) items. **(05 Marks)**

b. Construct LALR(1) parsing table for the following grammar:
   $S \rightarrow CC$
   $C \rightarrow cC|d$ **(10 Marks)**

c. Briefly explain Yacc generator. **(05 Marks)**

PART – B

5 a. Describe S-attributed and L-attributed definitions. **(04 Marks)**

b. Define syntax-directed definition. Write the syntax-directed definition for simple desk calculator and give annotated parse tree for the expression (7 – 2) * (8 – 1)n. **(10 Marks)**

c. Explain any two syntax-directed translation schemes. **(06 Marks)**

6 a. Briefly explain different types of intermediate codes, with the expression,
$a := b * - c + b * - c$. **(12 Marks)**

b. Construct a DAG for the expression given below:
$((x + y) - ((x + y)) * (x - y))) + ((x + y) * (x - y))$ **(04 Marks)**

c. Give SDTS (Syntax-Directed Translation scheme) for switch-statement. **(04 Marks)**

7 a. Explain the concept of heap management. Give the memory hierarchy of a computer. **(08 Marks)**

b. What is Garbage collector? Design goals for garbage collector. **(06 Marks)**

c. Briefly explain activation records. **(06 Marks)**

8 a. What are basic blocks? Write an algorithm for partitioning into basic blocks. **(06 Marks)**

b. Explain code generation algorithm. Generate the code for the following expression:
$w = (a - b) + (a - c) + (a - c)$ **(10 Marks)**

c. Briefly explain GetReg function. **(04 Marks)**

* * * * *