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

10IS662

Sixth Semester B.E. Degree Examination, Dec.2016/Jan.2017
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. Differentiate between computer and Interpreter. List and explain the various phases of a compiler and show the output of each phase for the expression a : = b + c * 25. **(10 Marks)**

b. Explain the technique of input buffering used by the lexical analyser. **(06 Marks)**

c. Construct transition diagram for the following :
   i) Unsigned numbers
   ii) Relational operators. **(04 Marks)**

2 a. Write the algorithm used of eliminating left recursion. Use this algorithm and eliminate left recursion on the given grammar
$S \rightarrow C \mid a$
$C \rightarrow Dd \mid c$
$D \rightarrow Cc \mid d$ **(07 Marks)**

b. Write the "dangling else" grammar and show that it is ambiguous. Rewrite the grammar incorporating the rule "Match each else with the closest unmathed then". **(08 Marks)**

c. List and explain the various error recovery strategies. **(05 Marks)**

3 a. Construct the predictive parsing table for the given grammar
$E \rightarrow TE'$
$E' \rightarrow +E \mid \varepsilon$
$T \rightarrow FT'$
$T' \rightarrow T \mid \varepsilon$
$F \rightarrow PF'$
$F' \rightarrow *F' \mid \varepsilon$
$P \rightarrow (E) \mid a \mid b \mid ep$ **(10 Marks)**

b. Discuss the conflicts that can arise during shift reduce parsing giving one example for each type. Do shift reduce parse for the string (a, (a, a)) and indicate the presence of conflicts if any. Use the given grammar
$S \rightarrow (L) \mid a$
$L \rightarrow L, S \mid S$ **(10 Marks)**

4 a. Show that the following grammar is LR(1).
$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$
$A \rightarrow d$
$B \rightarrow d$ **(10 Marks)**

b. Explain the procedure for the construction of an LA LR parser. **(10 Marks)**

PART – B

5 a. Define :
   i) Inherited attribute
   ii) Synthesized attribute. **(02 Marks)**

b. Develop the grammar and SDD for a simple desk calculator and show the annotated parse tree for the expression 1 * 2 * 3* (4+5) n. **(10 Marks)**

c. Rewrite the actions of desk calculator SDD so that they manipulate the parser stack explicitly. Illustrate the parser stack implementation. **(08 Marks)**

6 a. Construct a DAG for the arithmetic expression 2* x+y * (2*x – y). Show the steps for constructing the DAG. **(08 Marks)**

b. Generate intermediate code for the statement "if (x < 100 || x > 200 && xi = y) x = 0", along with the required syntax directed translation scheme. Avoid redundant Gotos. **(12 Marks)**

7 a. Show the structure of activation record. Explain the purpose of each item on the activation record. **(08 Marks)**

b. Explain the strategy for reducing fragmentation in heap memory. **(08 Marks)**

c. List and explain the performance metrics to be considered when designing a garbage collector. **(04 Marks)**

8 a. Generate the code for the expression
   X = (a – b) + (a + c) **(04 Marks)**

b. What are basic blocks? Explain the DAG representation of basic blocks. **(08 Marks)**

c. Describe the next use information. Write an algorithm to determine the liveness and next use information for each statement in a basic block. **(08 Marks)**

* * * * *