Total No. of Questions : 8] *[Total No. of Printed Pages :2*

Roll No .....

CS-701 (GS)

B.E. VII Semester

Examination, December 2017

Grading System (GS)

Compiler Design

Time : Three Hours

Maximum Marks : 70

Note: i) Attempt any five questions.
ii) All questions carry equal marks.

1. a) What is meant by Input buffering? Explain the use of sentinels in recognizing tokens. 7
   b) Explain the various phases of compiler with the help of diagram. Take any one example to elaborate complete working of compiler. 7

2. a) Write a lex specification and the pattern matching routine to display the identifier and the line number of its occurrences. 7
   b) Explain left recursion and show how it is eliminated. Describe the algorithm used for eliminating left recursion. 7

3. a) Show that the following grammar is LR(1) but not LALR(1). 7
   $S \rightarrow Aa/bAc/Bc/bBa$
   $A \rightarrow d$
   $B \rightarrow d$
   b) What are the causes of backtracking in top down parser? Explain with an example. 7

4. a) Assuming suitable syntax directed definition, construct a syntax tree for the expression $a-4+e$. 7

522

   b) Consider the grammar:
   $S \rightarrow ACB / CbB / Ba$
   $A \rightarrow da/BC$
   $B \rightarrow g/\in$
   $C \rightarrow h/\in$
   Calculate FIRST and FOLLOW. 7

5. a) What are the difficulties faced by memory allocation for variable length requirements? Under what circumstances does external fragmentation happen. 7
   b) Explain the concept of type checking and type conversion with an example. 7

6. a) Translate the following code into a three address code:
   a = 10;
   b = 6;
   if (a>b+5)
   b = 5; 7
   b) What is DAG? Construct DAG for the basic block:
   D: = B*C
   E: = A+B
   B: = B*C
   A: = E–D 7

7. a) Analyse the possible causes of a dead code. Explain with an example how the compiler can catch the presence of a dead code. 7
   b) Explain the common subexpression elimination copy propagation and transformation for moving loop invariant computations in detail. 7

8. Write short notes on: 14
   a) Peephole optimization
   b) Symbol table
   c) Global Data flow analysis
   d) Back patching.

******

523