Sixth Semester B.E. Degree Examination, June/July 2015
Compiler Design

Time: 3 hrs.   Max. Marks:100

Note: Answer any FIVE full questions, selecting atleast TWO questions from each part.

PART – A

1 a. Explain with a diagram, the phases of complier.   **(08 Marks)**
   b. Write regular definitions for the following using extended regular expression notation :
      i) identifier
      ii) unsigned number.   **(06 Marks)**
   c. Write a program for look ahead code with sentinels.   **(06 Marks)**

2 a. Define left – recursive grammer. Eliminate left recursion from the following grammer :
$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id.$$ **(05 Marks)**
   b. Given the grammer :
      $S \rightarrow AaAb \mid BbBa$
      $A \rightarrow \epsilon$
      $B \rightarrow \epsilon$
      i) compute FIRST( ) and FOLLOW( ) functions
      ii) construct predictive parsing table
      iii) parse the input string w = ab.   **(09 Marks)**
   c. Show that the following grammer is ambiguous $E \rightarrow E + E \mid E * E \mid (E) \mid id$, write an
      equivalent un-ambiguous grammer for the same.   **(06 Marks)**

3 a. What is meant by handle pruning? construct Bottom – up parse tree for the input string
      w = aaa + a ++. Using the grammer :
      $S \rightarrow SS + \mid SS * \mid a$.   **(06 Marks)**
   b. Explain the working of shift reduce parser. Parse the input string id * id. Using the grammer
      of question no, 2(a).   **(08 Marks)**
   c. With a diagram, explain the model of an LR parser.   **(06 Marks)**

4 a. Write an algorithm to construct LALR parsing table.   **(06 Marks)**
   b. Construct the parsing table for LALR(1) parser using the grammer :
      $S \rightarrow CC$
      $C \rightarrow aC$
      $C \rightarrow d$.   **(10 Marks)**
   c. Compare LALR and canonical LR parsers.   **(04 Marks)**

PART – B

5   a.   Explain the concept of syntax directed definition.   (04 Marks)
    b.   Consider the context free grammer given below :
         S → EN
         E → E + T | E – T | T
         T → T * F | T/F | F
         F → (E) | digit
         N → ;
         i)   Obtain SDD for the above grammer
         ii) Annotated parse tree for the input string 5 * 6 + 7.   (10 Marks)
    c.   Define :
         i)   Synthesized attribute
         ii) Inherited attribute.   (06 Marks)

6   a.   Construct DAG and three address code for the following expression :
         a + a * (b – c) + (b – c) * d.   (08 Marks)
    b.   Explain the following with an example : i) quadruples ii) triples.   (08 Marks)
    c.   Generate three address code for the following statement :
         switch (ch)
         { case 1 : c = a + b ; break ;
           case 2 : c = a – b ; break ;
         }   (04 Marks)

7   a.   With a neat diagram, describe the general structure of an activation record.   (06 Marks)
    b.   Explain in the strategy for reducing fragmentation in heap memory.   (08 Marks)
    c.   Explain briefly the performance metrics to be considered while designing a garbage
         collector.   (06 Marks)

8   a.   Discuss the various issues in the design of a code generator.   (10 Marks)
    b.   What are basic blocks and flow graphs? Write an algorithm to partition the three address
         instructions into basic blocks.   (06 Marks)
    c.   List the characteristics of a peephole optimization.   (04 Marks)

* * * * *