Question Paper Code : X 60388

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2020
Sixth Semester
Computer Science and Engineering
CS 2352/CS 62/10144 CS 602 – PRINCIPLES OF COMPILER DESIGN
(Regulations 2008/2010)
(Common to PTCS 2352 – Principles of Compiler Design for BE (Part – Time)
Fifth Semester – CSE – Regulations 2009)

Time : Three Hours   Maximum : 100 Marks

Answer ALL questions

PART – A   (10×2=20 Marks)

1. Define tokens, patterns and lexemes.

2. Mention the issues in a lexical analyzer.

3. Eliminate the left recursion for the grammar

   S → A a | b

   A → Ac | S d | ∈

4. What is meant by coercion ?

5. Draw syntax tree for the expression a = b* – c + b* – c.

6. Define backpatching.

7. Define flow graph.

8. How to perform register assignment for outer loops ?

9. What are the global common sub expressions ?

10. What are the control-flow constraints ?

PART – B   **(5×16=80 Marks)**

11. a) Explain in detail about the phases of compiler and translate the statement
        pos : = init + rate * 60.   **(16)**

                        (OR)

    b) Compare NFA and DFA. construct a DFA directly from an augmented regular
        expression ((ε/a)b*)*.   **(16)**

12. a) i) Construct a predictive parser for the following grammar.   **(10)**
           S → (L) | a
           L → L, S | S.
        ii) List all LR (0) items for the following grammar   **(6)**
           S → AS | b
           A → SA | a.

                        (OR)

    b) i) What are the different storage allocation strategies ? Explain.   **(8)**
        ii) Specify a type checker which can handle expressions, statements and
            functions.   **(8)**

13. a) i) Distinguish between quadruples and triples with example.   **(8)**
        ii) What are the rules for type checking ? Give an example.   **(8)**

                        (OR)

    b) i) State and explain the algorithm for unification.   **(8)**
        ii) Explain the one pass code generation using back patching with example. **(8)**

14. a) i) Explain in detail about the various issues in design of code generator. **(10)**

        ii) Write an algorithm to partition a sequence of three address statements
            into basic blocks.   **(6)**

                        (OR)

    b) i) Explain the code-generation algorithm in detail.   **(8)**
        ii) Construct the dag for the following basic block.   **(8)**
           d : = b* c
           e : = a + b
           b : = b * c
           a : = e – d

15. a) i) Write in detail about loop optimization.   **(8)**
        ii) Discuss the characteristics of peephole optimization.   **(8)**

                        (OR)

    b) Discuss in detail about global data flow analysis.   **(16)**

_____