2021

(July)

COMPUTER SCIENCE

(Honours)

(Compiler Design)

(CS–602 AT)

Marks : 75

Time : 3 hours

The figures in the margin indicate full marks for the questions

Answer **one** question from each Unit

UNIT—I

1. *(a)* Explain the functions of *lexical analyzer*, *syntax analyzer* and *intermediate code generator* of a compiler with the help of examples. 3+3+3=9

(b) What type of language does a finite automaton accept? Design a finite automaton to recognize an identifier. 2+4=6

2. *(a)* Differentiate between a compiler and an interpreter. What are single-pass and two-pass compilers? 3+2=5

(b) Construct the NFA for the regular expression a b(a|b) abb. Convert the NFA to its corresponding DFA and minimize the DFA. 10

UNIT—II

3. *(a)* Define a context-free grammar (CFG). Can a CFG be ambiguous? If so, explain with an example. 2+1+3=6

(b) Briefly explain the difference between top-down parsing and bottom-up parsing. What are the rules that govern the calculation of the functions FIRST and FOLLOW? Calculate the FIRST and FOLLOW of the grammar given below : 1+4+4=9

E TE`
E` TE` |
T FT`
T` *FT` |
F (E)|**id**

4. *(a)* Consider the grammar

expr → expr + term
expr → term
expr → term fact
term → fact
fact → (expr)
fact → id

Construct the set of LR(0) items and build an SLR parsing table. 9

(b) Construct the sets of LR(1) items for the augmented grammar given below : 6

S` → S
S → CC
C → cC|d

UNIT—III

5. *(a)* Discuss the importance of type checking. Differentiate between static and dynamic checkings. 3+4=7

(b) How do the various phases of a compiler manipulate the symbol table? 8

6. *(a)* What do you understand by a type expression? What are the pros and cons of weakly typed language over strongly typed language? 3+4=7

(b) Define the scope of a symbol in a programming language. Explain the different types of scopes possible in block-structured languages. 2+6=8

UNIT—IV

7. *(a)* What is an activation record? Elaborate on the different constituents of the activation record. 2+8=10

(b) What is intermediate code? How is the generation of intermediate code more advantageous over direct code generation? 1+4=5

8. *(a)* What do you understand by runtime storage allocation? Differentiate between static and dynamic allocations. Why is recursion not supported in static allocation? 2+4+2=8

(b) Describe two features each in using *high-level* intermediate representation and *low-level* intermediate representation in intermediate languages. 2

(c) Describe with an example how a *quadruple* can be used to implement three-address code. 5

UNIT—V

9. *(a)* Discuss the factors affecting target code generation. 5

(b) Consider the following sequence of statements :

x y z
w p y
y y z
p w x

Construct the corresponding directed acyclic graph (DAG). Perform code generation assuming two registers are available. 4+6=10

10. *(a)* Discuss the factors influencing optimization. 6

(b) Explain with examples any *three* optimization transformations : 3+3+3=9

(i) Folding and constant propagation

(ii) Common subexpression elimination

(iii) Variable propagation

(iv) Strength reduction

★ ★ ★