2023

(May/June)

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)* Distinguish between compiler and interpreter bringing out clearly the situations in which each of them is more suitable than the other. 5

(b) Describe briefly each of the phases of a compiler with the help of a diagram. 8

(c) What is a token? 2

2. *(a)* What is the use of regular expression in lexical analysis? 3

(b) Distinguish between NFA and DFA. Design a finite automaton to recognize the following regular expression : 4+6=10

$$a \cdot (a \cdot b)^* \cdot a \cdot b^*$$

(c) Write down the regular expressions for binary strings with even numbers of '0's. 2

UNIT—II

3. *(a)* What is an ambiguous grammar? Identify if the following grammar is ambiguous or not : 2+6=8

$$S \rightarrow SbS \mid a$$

(b) Construct LR(0) parsing table for the following grammar : 7

$$S \rightarrow AA,\ A \rightarrow aA \mid b$$

[S is the starting variable]

4. *(a)* Explain the role of syntax analyzer in compiler design. 3

(b) Construct LR(1) item and LALR(1) parsing table for the following grammar : 12

$$S \rightarrow CC,\ C \rightarrow cC \mid d$$

[S is the starting variable]

UNIT—III

5. *(a)* What do you understand by the type checking? 3

(b) Discuss any two kinds of type expressions that are used to represent the types of language construct. 4

(c) How are symbol tables classified based on the scope rules? Give an example of each. 2+6=8

6. *(a)* Discuss how the following phases of a compiler use the information stored in the symbol table (any *two*) : 5

(i) Semantic analysis

(ii) Error detection

(iii) Optimization

(b) What are the sets of operations that are carried out on a symbol table? 4

(c) Draw the type trees for the following pairs of type declarations and argue whether they are type equivalent or not : 6

```
struct list1{              struct list2{
int val;                   int val;
struct list1 *next;        struct list2 *next;
}                          }
```

UNIT—IV

7. *(a)* Below are the three types of entities that have to be managed at runtime. Describe each one of the them in brief : 6

   (i) Generated code

   (ii) Data objects

   (iii) Stack

   (b) Explain how the quadruple can be used to implement three-address instruction. Show the annotated parse tree and code generation process for the following arithmetic expression : 2+3+4=9

   $x := (y + z)^* (-w + v)$

8. *(a)* Describe an activation record of a program. Show the snapshots of the stack of activation records at the time when *main* has called *g*, *g* has called *f*, and *f* has in turn called *g* in the following program : 2+5=7

```
int x=2;
void f(int n){
        static int x=1;
        g(n);
        x--;
}
```



```
void g(int m){
        int y=m-1;
        if (y>0) {
                f(y);
                x--;
        }
}
main() {g(x); return 0;}
```

   (b) What are the features of three-address code? Show how it can represent the assignment statement with binary and unary operators respectively. 4+4=8

UNIT—V

9. *(a)* Assuming that two registers are available, perform code generation using dynamic programming for the following statement : 9

   Target code format is "*OP*, T_i, R_j", where *OP* stands for operator, T_i stands for reference and R_j stands for register.

   $(a^* b) + (c - (d + e))$

   (b) Discuss any two loop optimizations with appropriate examples. 3+3=6

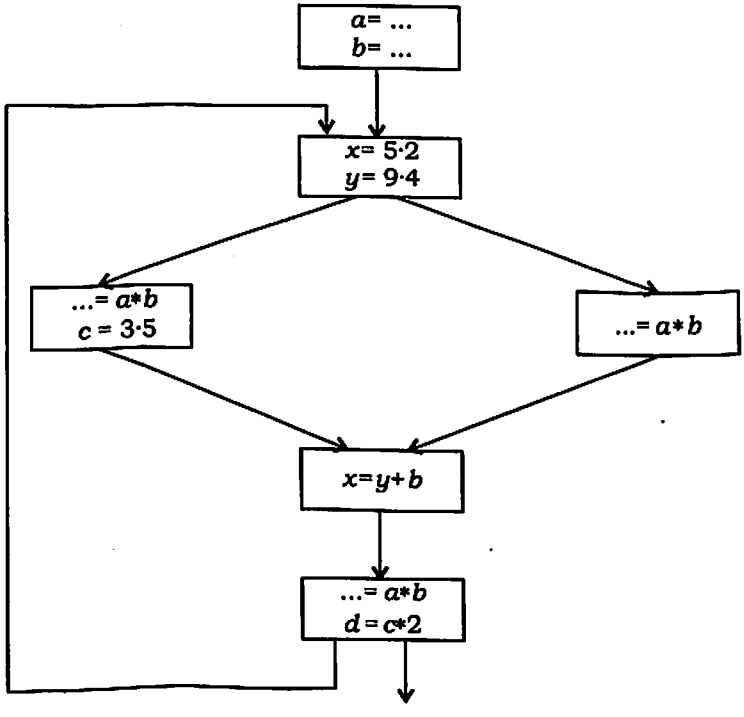
10. *(a)* Mention any two factors that can affect code generation. What is a basic block? How can it be constructed? 2+1+3=6

(b) Study the following flow graph and indicate with justification whether or not any of the following optimization transformations can be applied to it : 3+3+3=9

(i) Common subexpression elimination

(ii) Elimination of dead code

(iii) Constant propagation



★★★