Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 90928

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2025

Fifth Semester

Computer Science and Engineering

CS 3501 — COMPILER DESIGN

(Regulations 2021)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — ($10 \times 2 = 20$ marks)

1. What do you mean by static scope in the context of block structure programming languages?

2. What is the role of lexical analyzer in compiler?

3. Write grammar for the language "The set of all strings of 0's and 1's such that every 0 is immediately followed by at least one 1"

4. Compare between top-down parsing and bottom-up parsing.

5. What do you mean by dependency graph? Give an example.

6. Suppose that a is an array of integers, and that f is a function from integers to integers. Then, how the following assignment will be translated into three-address code?

$$n = f(a[i]).$$

7. Draw the subdivision of run-time memory into code and data areas.

8. List the storage allocation strategies with its usage.

9. How the compiler will eliminate dead code? Give an example.

10. What is the purpose of constant folding?

PART B — (5 × 13 = 65 marks)

11. (a) (i) Most languages are case sensitive, so keywords can be written only one way, and the regular expressions describing their lexemes are very simple. However, some languages, like SQL, are case insensitive, so a keywords can be written either in lowercase or in uppercase or in any mixture of cases. Thus, the SQL keyword SELECT can also be written select, Select, or sEIEcT, for instance show how to write a regular expression for a keyword in a case insensitive language. Illustrate the idea by writing the expression for "select" in SQL. (7)

(ii) Compare and contrast between compilation and interpretation. (6)

Or

(b) (i) Draw the structure and phases of a compiler and illustrate in detail. (6)

(ii) Tagged languages like HTML or XML are different from conventional programming languages in that the punctuation (tags) are either very numerous (as in HTML) or a user-definable set (as in XML). Further, tags can often have parameters Suggest how to divide the following HTML document into appropriate lexemes.

Here is a photo of <B>my house</B>:
<P><IMG SRC = "house gif">
See <A HREF = 'morePix.html">More Pictures</A> if you liked that one.<P>

Which lexemes should get associated lexical values and what should those values be? (7)

12. (a) (i) Write a Yacc program that tells whether its input is a palindrome (sequence of characters that read the same forward and backward). (6)

(ii) Show that the following grammar:

$S \rightarrow A\,a\,A\,b \mid B\,b\,B\,a$

$A \rightarrow \varepsilon$

$B \rightarrow \varepsilon$

is LL(1) but not SLR(1). (7)

Or

(b) The following is a grammar for regular expressions over symbols a and b only, using + in place of | for union, to avoid conflict with the use of vertical bar as a metasymbol in grammars:

rexpr	*rexpr* + *rterm* | *reterm*
rterm	*rterm rfactor*\ *rfactor*
rfactor	*rfactor* * | *rprimary*
rprimary	**a** | **b**

(i) Left factor this grammar.

(ii) Does left factoring make the grammar suitable for top-down parsing?

(iii) In addition to left factoring, eliminate left recursion from the original grammar.

(iv) Is the resulting grammar suitable for top-down parsing? (3+3+3+4)

13. (a) Below is a grammar for expressions involving operator + and integer or floating- point operands. Floating-point numbers are distinguished by having a decimal point.

$$E \rightarrow E + T | T$$

$$T \rightarrow \textbf{num} \bullet \textbf{num} \mid \textbf{num}$$

(i) Give an SDD (Syntax Directed Definition) to determine the type of each term T and expression E.

(ii) Extend your SDD of (i) to translate expressions into postfix notation. Use the unary operator intToFloat to turn an integer into an equivalent float. (6+7)

Or

(b) (i) Determine the types and relative addresses for the identifiers in the following sequence of declarations:

```
float x;
record { float x; float y;} p;
record { int tag; float x; float y; } q;
```

(7)

(ii) Explain in detail about one-pass code generation using backpatching. (6)

14. (a) Consider the below code snippet to computer Fibonacci number recursively. Suppose that the activation record for f includes the following elements in order: (return value, argument n, local s, local t); there will normally be other elements in the activation record as well. The questions below assume that the initial call is *f*(5)

```
int f (int n) {
    int t, s;
    if (n < 2) return 1;
    s = f(n-1);
    t = f(n-2);
    return s+t;
}
```

(i) Show the complete activation tree.

(ii) What does the stack and its activation records look like the first time f (1) is about to return?

(iii) What does the stack and its activation records look like the fifth time f (1) is about to return? (5+4+4)

Or

(b) Suppose the heap consists of seven chunks, starting at address 0. The sizes of the chunks, in order, are 80, 30, 60, 50, 70, 20, 40 bytes. When we place an object in a chunk, we put it at the high end if there is enough space remaining to form a small chunk (so that the smaller chunk can easily remain on the linked list of free space). However, we cannot tolerate chunks of fewer that 8 bytes, so if an object is almost as large as the selected chunk, we give it the entire chunk and place the object at the low end of the chunk. if we request space for objects of the following sizes: 32, 64, 48, 16, in that order, what does the free space list look like after satisfying the requests, if the method of selecting chunks in Best fit. (13)

15. (a) (i) Construct the DAG for the basic block

$d = b * c$
$e = a + b$
$b = b * c$
$a = e - d$ (6)

(ii) Illustrate in detail about Data Flow Analysis with its properties. (7)

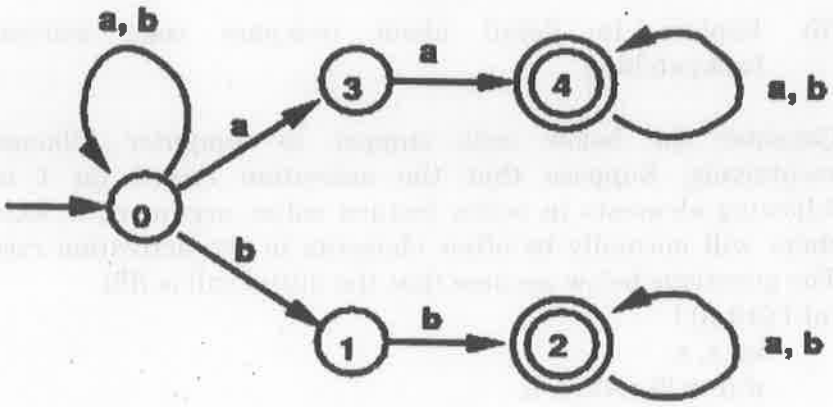
Or

(b) (i) Construct an algorithm that will perform redundant-instruction elimination in a sliding peephole on target machine code. (8)

(ii) Write in detail about recent trends in compiler design. (5)

PART C — ($1 \times 15 = 15$ marks)

16. (a) Convert the given NFA machine (M1) into a DFA machine (M2) and then identify the language accepted by both M1 and M2.



Or

(b) Exemplify in detail 4 different types of grammars and regular expressions. Construct Context-free grammar for the language $L = a^n b^{2n}$ where $n >= 1$.