

# JK LAKSHMIPAT UNIVERSITY, JAIPUR

## INSTITUTE OF ENGINEERING AND TECHNOLOGY

End Term Examination (Regular), April-May 2018  
B. Tech in Computer Science Engineering, Semester VI

Subject Code: CSE603

Roll No.....

Subject Name: Compilers

Time: 3 hours

Max. Marks: 100

### Instructions to students:

1. Do not write anything other than your roll number on question paper.
2. Assume suitable data wherever required and mention it clearly.
3. Writing appropriate units, nomenclature, and drawing neat sketches/schematics/ flow chart wherever, required are an integral part of the answer.

|       |                                                                                                                                                                                                                                                                                                          |    |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Q.1A  | Objective type multiple choice questions (10 questions @ 1 mark)                                                                                                                                                                                                                                         | 10 |
| i.    | A Lex compiler generates<br>a. Lex object code    b. Transition tables    c. C tokens    d. None of these                                                                                                                                                                                                |    |
| ii.   | Yacc semantic action is a sequence of<br>a. Tokens    b. Expression    c. C statement    d. Rules                                                                                                                                                                                                        |    |
| iii.  | In which addressing mode the operand is given explicitly in the instruction?<br>a. Absolute mode    b. Immediate mode    c. Indirect mode    d. Index mode                                                                                                                                               |    |
| iv.   | DAG representation of a basic block allows<br>a. Automatic detection of local common sub expressions<br>b. Automatic detection of induction variables<br>c. Automatic detection of loop invariant<br>d. None of the above                                                                                |    |
| v.    | We have the following basic block:<br>$T1 = a + b$<br>$T2 = c + d$<br>$T3 = e - T2$<br>$T4 = T1 - T3$<br>Assuming that all operands are initially in memory and the final value of the computation in memory, the minimum number of MOV instructions in the code generated for this block is .....       |    |
| vi.   | Regular expression to recognize a valid variable which starts with a letter followed by any number of letters or digits is .....                                                                                                                                                                         |    |
| vii.  | Regular expression to recognize a character constant is .....                                                                                                                                                                                                                                            |    |
| viii. | Given the following expression grammar:<br>$E \rightarrow E * F \mid F + E \mid F$<br>$F \rightarrow F - F \mid id$<br>Which of the following is true?<br>a) * has higher precedence than +<br>b) - has higher precedence than *<br>c) + and - have same precedence<br>d) + has higher precedence than * |    |

ix. Consider the grammar

$$E \rightarrow E + n \mid E \times n \mid n$$

For a sentence  $n + n \times n$ , the handles in the right-sentential form of the reduction are

- A. a)  $n$ ,  $E + n$  and  $E + n \times n$
- B. b)  $n$ ,  $E + n$  and  $E + n \times n$
- C. c)  $n$ ,  $n + n$  and  $n + n \times n$
- D. d)  $n$ ,  $E + n$  and  $E \times n$

x. Match the following:

| List-I                   | List-II                 |
|--------------------------|-------------------------|
| A. Lexical analysis      | 1. Graph coloring       |
| B. Parsing               | 2. DFA minimization     |
| C. Register allocation   | 3. Post-order traversal |
| D. Expression evaluation | 4. Production tree      |

- Q.2 Short answer type questions (4 questions @ 2.5 marks) 10
- i. Explain differences in between compiler, interpreter and assembler.
  - ii. Explain input buffering?
  - iii. Write a Lex program to check the token as valid date?
  - iv. Explain role of YACC in compilation of a program.
- Q.3A Illustrate the translation of the following statement on all phases of compiler 05  
 $a := b * b * a + d / e$
- Q.3B Explain about Evaluation order for SDDs. 05
- Q.4A Remove the left recursion from the productions:  $E \rightarrow E + T \mid T$  05  
 $T \rightarrow T * F \mid F$
- Q.4B How the syntax of a statement is validated by a syntax analyzer? 05
- Q.5A Draw an annotated parse tree for syntax directed definition and translation for the 05  
 $4 * 3 + 2$
- Q.5B Explain the division of run time memory. Draw the block diagram of an 05  
activation record.
- Q.6A Consider the following CFG grammar, 05  
 $S \rightarrow aABe$   
 $A \rightarrow Abc \mid b$   
 $B \rightarrow d$   
where 'a', 'b', 'c', 'e' and 'd' are terminals, and 'S' (start symbol), 'A' and 'B' are non-terminals.
- a) Parse the sentence "abbcd" using right-most derivations.
  - b) Parse the sentence "abbcd" using left-most derivations.
  - c) Draw the parse tree.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |  |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Q.6B | <p>Consider the following grammar describing a certain sort of rested lists: 05</p> <p><math>S \rightarrow TS   TU</math><br/> <math>T \rightarrow U.T   U</math><br/> <math>U \rightarrow x   y   [S]</math></p> <p>i) Left factor this grammar.<br/> ii) Give the First and FOLLOW sets for each non terminal in the grammar obtain in part (i).<br/> iii) Using this information construct an LL parsing table for the grammar obtained in part (i).</p> |  |
| OR   |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |  |
| Q.6B | <p>Consider the following grammar: 05</p> <p><math>G = (\{D, T, L\}, \{int, real, ,(comma), id\}, \{ D \rightarrow TL, T \rightarrow int   real, L \rightarrow L, id   id\}, D)</math></p> <p>Write down the syntax directed definitions (SDDs) for the above grammar and use these SDDs to construct annotated parse tree for the input string 'real id<sub>1</sub>, id<sub>2</sub>, id<sub>3</sub>'.</p>                                                  |  |
| Q.7A | <p>Explain different data structures used for building symbol table. How the data structures handle the following program fragment. 05</p> <pre>#include&lt;stdio.h&gt; int *assignval (int *x, int val) {     *x = val;     return x; }  void main () {     int *x = malloc(sizeof(int));     if (x) {         x = (int *)malloc(sizeof(int));     }     printf("%d\n", *x);     free(x); }</pre>                                                          |  |
| Q.7B | <p>Define a Quadruple. How it is different from triples? Convert the following expression into three address code and quadruple. 05</p> <p><math>(a+b)/(c-d)*(e+f)</math></p>                                                                                                                                                                                                                                                                               |  |
| OR   |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |  |
| Q.7B | <p>Define back patching and semantic rules for Boolean expression. Derive the three address code for the expression, <math>P &lt; Q</math> or <math>R &lt; S</math> and <math>T &lt; U</math> 05</p>                                                                                                                                                                                                                                                        |  |
| Q.8A | <p>Construct the FIRST and FOLLOW for the grammar and further build a parsing table for top down parser: 10</p> <p><math>A \rightarrow BC   EFGH   H</math><br/> <math>B \rightarrow b</math><br/> <math>C \rightarrow c   \epsilon</math><br/> <math>E \rightarrow e   \epsilon</math><br/> <math>F \rightarrow CE</math></p>                                                                                                                              |  |

$G \rightarrow g$   
 $H \rightarrow h \mid \epsilon$

- Q.9A i. Construct DAG of basic blocks after converting the following code in three-address representation: 05  
     $\{ (a/b*c) + (a/d) - (c+(a*b)) \}$  05  
ii. Write a short note on garbage collection.
- Q.9B i. What are issues in design of code generator, loop optimization? 05  
ii. Explain how directed acyclic graph is created for the expression  $a:=b*b*a+d/e$  05

OR

- Q.9B Write short note on the following: (any two) (@ 5 marks each) 10  
i. Access to non-local data on the stack  
ii. Process for identifying a block in a C program  
iii. loop invariant computation