

JK LAKSHMIPAT UNIVERSITY, JAIPUR

INSTITUTE OF ENGINEERING AND TECHNOLOGY

End Term Examination (Regular), April-May 2016
B. Tech in Computer Science Engineering, Semester VI



Subject Code: CSE603

Roll No.....

Subject Name: Compiler Design

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. The lexical analysis for a modern computer language such as Java needs the power of which one of the following machine models in a necessary and sufficient sense?

- (a) Finite state automata (b) Deterministic PDA
(c) Non-deterministic PDA (d) Turing machine

- ii. A ----- is a compact notation that is used to represent the patterns corresponding to a token.

- (a) Transition diagram (b) Regular expression (c) Alphabet (d) Input buffer

- iii. Consider the following grammar:

$G = (\{S, C, S_1\}, \{a, b, i, t, e\}, \{S \rightarrow iCtSS_1 \mid a, S_1 \rightarrow eS \mid \epsilon, C \rightarrow b\}, S)$

The grammar is not LL(1) because:

- (a) It is left recursive (b) It is right recursive
(c) It is ambiguous (d) It is not left factored

- iv. Consider the following two sets of LR(1) items of an CLR(1) grammar.

$X \rightarrow c.X, c \mid d$ $X \rightarrow c.X, \$$
 $X \rightarrow .cX, c \mid d$ $X \rightarrow .cX, \$$
 $X \rightarrow .d, c \mid d$ $X \rightarrow .d, \$$

Which of the following statements related to merging of the two sets in the corresponding LALR parsing is/are FALSE?

- (1) Cannot be merged since look-ahead are different.
(2) Can be merged but will result in Shift-Reduce conflict.
(3) Can be merged but will result in Reduce-Reduce conflict.
(4) Cannot be merged since goto on c will lead to two different sets.
(a) (1) only (b) (2) only (c) (1) and (4) only (d) (1), (2), (3) and (4)

- v. The syntax directed definition given below is

Production	Semantic Rules
$A \rightarrow LM$	$L.i := f_1(A.i); M.i := g_1(L.s); A.s := h_1(M.s)$
$A \rightarrow QR$	$R.i := f_2(A.i); Q.i := g_2(R.s); A.s := h_2(Q.s)$

- (a) a L-attributed definition (b) non-L-attributed definition
(c) S-attributed definition (d) Data insufficient

vi. Consider the following syntax directed translation scheme:

Production	Translation scheme
$S \rightarrow aS$	$S \rightarrow aS\{m := m + 3; \text{print}(m);\}$
$S \rightarrow bS$	$S \rightarrow bS\{m := m * 2; \text{print}(m);\}$
$S \rightarrow \epsilon$	$S \rightarrow \epsilon\{m := 0;\}$

What will be printed if the string derived from the grammar is 'a a b a b b'?

(a) 0 0 3 6 9 12 (b) 0 0 0 3 6 9 12 (c) 0 0 0 3 6 9 12 15 (d) 0 0 3 9 6 12

vii. Let $A = 2$, $B = 3$, $C = 4$ and $D = 5$, what is the final value of the prefix expression:

$+ * AB - CD$

(a) 5 (b) 10 (c) -10 (d) -5

viii. In static allocation, names are bound to storage at ----- time.

(a) Compile (b) Runtime (c) Debugging (d) Both (a) and (b)

ix. Which of the following is NOT performed during compilation?

(a) Dynamic memory allocation (b) Type checking
(c) Symbol table management (d) Inline expansion

x. Which of the following is FALSE regarding a basic block?

(a) The first statement is leader.
(b) Any statement that is a target of conditional/unconditional jump is a leader.
(c) The last statement is a leader.
(d) Immediately next statement of an unconditional or conditional jump is a leader.

Q.2 Short answer type questions (4 questions @ 2.5 marks)

10

- Briefly explain the concept of bootstrapping.
- Eliminate the left recursion from the following grammar:

$G = (\{E, T, F\}, \{+, -, *, /, (,), \text{id}\}, P, E)$
where P consists $E \rightarrow E + T \mid E - T \mid T$
 $T \rightarrow T / F \mid T * F \mid F$
 $F \rightarrow (E) \mid \text{id}$

- What are the L-attributed definitions?
- Define basic block. How can you find leaders in basic blocks?

Q.3A Demonstrate the output generated by various phases of compiler for the following instruction of source program during compilation.

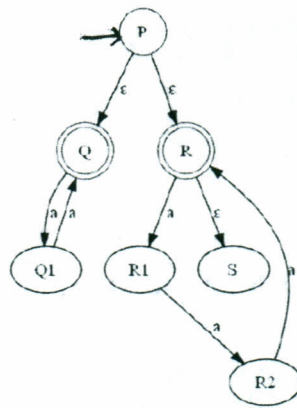
05

$\text{Position} = \text{Initial} + \text{Rate} * 60$

Assuming that the variables Position, Initial and Rate have been declared as float.

Q.3B Convert the following NFA with ϵ -moves into DFA.

05



- Q.4A Enlist and explain the various functions of lexical analyzer. Illustrate why do we use pair of buffers for input buffering? 05
- Q.4B Illustrate the input (LEX specifications) and output (Lexical analyzer) of the LEX to recognize the following strings: 05
ws (white space), if, then, else, end, id (identifier)
- Q.5A Construct the recursive-descent parser for the grammar G (given below) and illustrate the parsing procedure for the input string 'adb'. 05
 $G = (\{S, A, B\}, \{a, b, c, d\}, \{S \rightarrow aABb, A \rightarrow c \mid \epsilon, B \rightarrow d \mid \epsilon\}, S)$
- Q.5B i. Construct the LL(1) predictive parsing table for the grammar given in Q.5A. 05
ii. Demonstrate the parsing procedure for the input string 'acb' using the parsing table.
- Q.6A Consider the following grammar: 05
 $G = (\{E, T, F\}, \{+, *, (,), id\}, P, E)$
where P consists $E \rightarrow E + T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid id$
Determine the following:
i. Leading and Trailing lists for each nonterminal.
ii. Operator precedence parsing table.
- Q.6B Show that the following grammar is not SLR(1): 05
 $G = (\{S, A\}, \{a, b, c, d\}, \{S \rightarrow Aa \mid bAc \mid dc \mid bda, A \rightarrow d\}, S)$
OR
- Q.6B Construct the LALR(1) parsing table for the following grammar: 05
 $G = (\{S, C\}, \{c, d\}, \{S \rightarrow CC, C \rightarrow cC \mid d\}, P, S)$
- Q.7A Write the syntax directed definitions (SDDs) to construct a syntax tree for the following grammar: 05
 $G = (\{E\}, \{+, -, id, num\}, \{E \rightarrow E + E \mid E - E \mid (E) \mid id \mid num\}, E)$
Use these SDDs to construct a syntax tree for the string 'a - 4 + c'.
- Q.7B Consider the following grammar: 05
 $G = (\{E, T\}, \{+, -, digit\}, \{E \rightarrow E + T \mid E - T \mid T, T \rightarrow digit\}, E)$
Write down the translation scheme for the above grammar to perform top-down translation. Apply this scheme to construct annotated parse tree for the

expression '9 - 5 + 2'.

OR

Q.7B Translate the statement $a := -b * (c + d)/e$ into quadruple, triple and indirect triple representations. 05

Q.8A i. Translate the following statement to intermediate code: 05

$A[i, j] := B[i, j] + C[A[k, l]] + D[i, j]$

Suppose that A, B and D are arrays of size 30×40 , and C is array of size 20. There are 4 bytes/word.

ii. Construct syntax tree and postfix notation for the following expression: 05

$(a + b * (b * c)) \uparrow d - e / (f + g)$

Q.9A i. Explain the dynamic allocation strategies with example. 05

ii. What will be the output of the following program, if compiler uses following parameter passing methods? Justify your answers. 05

(1) Call by value (2) Call by reference (3) Call by name

The program is given as:

```
void main (void)
{
    int a, b;
    void fun(int, int, int);
    a = 2, b = 3;
    fun(a, a + b, a);
    printf("%d", a);
}

void fun(int x, int y, int z)
{
    x = x + 1;
    z = z + x;
}
```

Q.9B i. Briefly explain the various data structures used for implementing the symbol table. 05

ii. Determine the basic blocks for the three-address code segment given below and draw a flow graph to demonstrate the flow of control between the basic blocks. 05

1. $PRODUCT := 0$ 2. $j := 1$ 3. $t_1 := 4 * j$ 4. $t_2 := X[t_1]$
5. $t_3 := 4 * j$ 6. $t_4 := Y[t_2]$ 7. $t_5 := t_2 * t_4$ 8. $t_6 := PRODUCT + t_5$
9. $PRODUCT := t_6$ 10. $t_7 := j + 1$ 11. $j := t_7$ 12. if $j \leq 20$ goto (3)

OR

Q.9B i. Construct a directed acyclic graph (DAG) for the three-address code segment given above. 05

ii. Explain code generation phase with simple code generation algorithm. 05