WEST BENGAL UNIVERSITY OF TECHNOLOGY

IT-605C

COMPILER DESIGN

Time Allotted: 3 Hours | Full Marks: 70

The questions are of equal value.
The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable.
All symbols are of usual significance.

GROUP A
(Multiple Choice Type Questions)

1. Answer *all* questions. $10 \times 1 = 10$

(i) Which of the task is performed by Lexical Analyser?

(A) Correcting Errors
(B) Stripping out comments and white spaces
(C) Performs the expansion of macro
(D) All of these

(ii) Consider a grammar $A \rightarrow aS_1 \mid aS_2$. The left factored grammar produced from the grammar is

(A) $A' \rightarrow aA$
$A \rightarrow S_1 \mid S_2$

(B) $A \rightarrow aA'$
$A' \rightarrow aS_1 \mid aS_2$

(C) $A \rightarrow aA'$
$A' \rightarrow S_1 \mid S_2$

(D) None of these

(iii) Which of the following is a example of bottom up parsing?

(A) LL Parsing
(B) Predictive parsing
(C) Recursive descent parsing
(D) Shift-Reduce Parsing

(iv) Handle pruning forms the basis of _____

(A) bottom up parsing
(B) top down parsing
(C) predictive parsing
(D) recursive descent parsing

(v) Which of the following is not true for Syntax Directed Translation?

(A) It is an extension of CFG
(B) Parsing process is used to do the translation
(C) It does not permit the subroutines to be attached to the production of a CFG.
(D) It generates the intermediate code.

(vi) The reverse polish or suffix notation is known as

(A) Infix notation
(B) Prefix notation
(C) Postfix notation
(D) None of the above

(vii) Left factoring guarantees

(A) Not occurring of backtracking
(B) Cycle free parse tree
(C) Error free target code
(D) Correct LL(1) parsing table

(viii) Which of the following is not a loop optimization?

(A) Induction variable elimination
(B) Loop jamming
(C) Loop unrolling
(D) Loop heading

(ix) The lastpos of a .(dot) node with leaves c_1 and c_2 is

(A) if (nullable(c_1))
lastpos(c_1) $\cup$ lastpos(c_2)
else lastpos(c_2)

(B) if (nullable(c_2))
lastpos(c_1) $\cup$ lastpos(c_2)
else lastpos(c_1)

(C) if (nullable(c_1))
lastpos(c_1) $\cup$ lastpos(c_2)
else lastpos(c_1)

(D) if (nullable(c_2))
lastpos(c_1) $\cup$ lastpos(c_2)
else lastpos(c_2)

(x) By which of the following, bodies of two loops merges to form a single loop?

(A) Loop unrolling (B) Strength reduction
(C) Loop concatenation (D) Loop fusion

GROUP B
(Short Answer Type Questions)

Answer any *three* questions. 3×5 = 15

2. Construct DFA directly from the regular expression – (a|b)* abb. 5

3. Consider the following conditional statement, then find out how many tokens are possible and what are those? 5
if ($x >= 5$) then $y = 10$ else $y = 11$;

4. What is 'handle'? Consider the grammer – $E \rightarrow E + E \mid E * E \mid$ id. From this, find the handles of the right sentential forms of reduction of the string id + id * id. 2+3

5. Translate the following expression – $(a + b) * (c + d) + (a + b + c)$ into quadruples and triples separately. 5

6. (a) Draw the DAG for the following basic block - 2

$d := b*c$
$e := a + b$
$b := b*c$
$a := e - d$

(b) Draw the syntax tree for the following arithmetic expression – 3
$a * -(b + c / d)$.

GROUP C
(Long Answer Type Questions)

Answer any *three* questions. 3×15 = 45

7. (a) Construct the predictive parsing table for the following grammar - 12
$E \rightarrow E + T \mid T$
$T \rightarrow T * F \mid F$
$F \rightarrow (E) \mid$ id
(b) Show how the input string id + id * id is parsed by using the above generated parsing table. 3

8. Construct the SLR(1) parsing table for the following grammar – 15
$E \rightarrow E + T \mid T$
$T \rightarrow T * F \mid F$
$F \rightarrow (E) \mid$ id

9. (a) What do you mean by LL(1) grammar? 5
(b) Describe with diagram the role of a parser. 5
(c) Define left-factoring with an example. 5

10.(a) Explain L-attributed definitions in brief with an example. 6
(b) Differentiate between quadruples, triples and indirect triples. 6
(c) What is activation record? 3

11. Write short notes on any *three* of the following: 3×5
(a) Terminal table and Literal table
(b) Input buffering
(c) Predictive Parser
(d) Code optimization
(e) Peephole optimization