



Name :
Roll No. :
Invigilator's Signature :

CS/MCA/SEM-5/MCAE-504A/2012-13
2012
COMPILER DESIGN

Time Allotted : 3 Hours

Full Marks : 70

The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words
as far as practicable.

GROUP - A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for any *five* of the following :
5 × 2 = 10
- i) If G is $S \rightarrow aS / bS / a / b$, then L(G) is :
 - a) $\{a, b\}^*$
 - b) $\{a, b\}^+$
 - c) $\{a, b, S\}$
 - d) None of these.
 - ii) Context free grammar is accepted by :
 - a) Turing Machine
 - b) Finite Automata
 - c) Push Down Automata
 - d) None of these.
 - iii) A symbol table is a :
 - a) Compilation phase
 - b) Error handler
 - c) Data structure
 - d) None of these.
 - iv) Bottom up parsing is a right choice to handle a larger class of grammar.
 - a) True
 - b) False
 - c) Not always
 - d) Irrelevant.



- v) The difference between DAG and Syntax tree lies in the fact that :
- a) A node in a Syntax tree for a common sub expression has more than one parent
 - b) A node in a DAG for a common sub expression has more than one parent
 - c) A node in a Syntax tree for a common sub expression may have more than one parent.
- vi) What is not the phase of a compiler ?
- a) Syntax analyzer
 - b) Code generator
 - c) Code optimizer
 - d) Code linker.
- vii) What is the first phase of a compiler ?
- a) Code generator
 - b) Code optimizer
 - c) Lexical analyzer
 - d) Syntax analyzer.

GROUP - B

(Short Answer Type Questions)

Answer any *three* of the following. $3 \times 5 = 15$

2. Generate 3 address code for the following program segment
- ```
sum = 0;
for (j = 1; j<=10;j++)
sum=sum+a[j]+b[j];
```
3. a) What do you mean Left recursion ?  
b) Eliminate the left recursion from the following grammar:
- $$S \rightarrow (L) / a$$
- $$L \rightarrow L, S / S$$
- 2 + 3



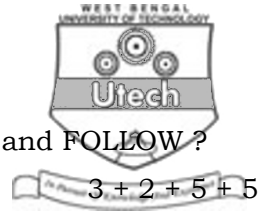
4. Compare different implementation of 3 – address code. 5
5. a) What is DAG ?  
 b) Draw the DAG for the following expression  
 $a + (b*d) + c* (b*d) + e + a/(b*d)$
6. Find out FIRST and FOLLOW for the following grammar :  
 $E \rightarrow E + T / T$   
 $T \rightarrow TF / F$   
 $F \rightarrow F * / a$

**GROUP – C**

**( Long Answer Type Questions )**

Answer any *three* of the following.  $3 \times 15 = 45$

7. a) A grammar is given below :  
 $S \rightarrow aS \mid aSbS \mid \epsilon$   
 Show that the grammar is ambiguous by constructing two parse trees and two leftmost derivations for **aab**.
- b) Consider the following grammar :  
 $S \rightarrow CC$   
 $C \rightarrow cC \mid d$   
 Construct the canonical collection of LR(1) items for this grammar.  $8 + 7$
8. a) Draw the DAG for the expression  
 $a + a * (b - c) + ( b - c ) * d$
- b) What is syntax tree ?
- c) Write the three address code for the following :  
 for (i = 1; i < 10; i++)  
 if (a < 10)  
 a = a + b;  
 else  
 a = a-b;



- d) What are the rules to compute FIRST and FOLLOW ?
9. Briefly explain each of the following with example  $5 \times 3 = 15$
- i) Constant Folding
  - ii) Common sub expression elimination
  - iii) Dead code elimination
  - iv) Loop unrolling
  - v) Code motion.
10. Write short notes on the following (any *three*) :  $3 \times 5 = 15$
- a) Three address code
  - b) Peephole optimization
  - c) Basic Block
  - d) Symbol table.
11. a) Discuss the procedure to convert a regular expression to corresponding NFA with figure, and hence convert the following regular expression to NFA :
- $(a | b)^*(ab)^*aabb$
- b) Eliminate the left recursion of the following productions:
- $bexpr \rightarrow bexpr \text{ or } bterm | bterm$
- $bterm \rightarrow bterm \text{ and } bfactor | bfactor$
- $bfactor \rightarrow \text{not } bfactor | (bexpr) | \text{true} | \text{false}$
- and hence find out the FIRST and FOLLOW of the above productions.  $6 + 9$
-