

Time Allotted : 3 Hours

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

(Objective Type Questions )

Answer any five of the following :

1. a) Draw finite automata for $(a+b) * a b b$
b) Draw NDFA for ( $\left.a b^{*}+b a^{*}\right)$ using Thompson's Construction Rule.
2. What will be the type checking rules for the following?
(i) $\quad E \rightarrow E 1+E 2$
(ii) $E \rightarrow E 1 \bmod E 2$
(iii) $S \rightarrow$ if C then $S 1$
$2+2+3$
3. a) Define Lexemes and Tokens.
b) What are the different kinds of tokens ? Give examples.
c) What kind of expression or grammar is used in specifying tokens?
$2+3+2$

30064(M.Tech)
[ Turn over
4. What is the input an output of a compiler ? Which phases of the compiler
a) Process linear input and produces linear output?
b) Process linear input and produces hierarchical output?
c) Process hierarchical and produces linear output?
d) Mention the inputs and outputs of these phases individually.
$1+2+2+2$
5. a) When does intermediate code generation take place ?
b) Translate the expression $x=m / n+p^{*} q$ into
(i) Quadruples
(ii) Triples
(iii) Indirect Triples.
( Given : " / " and " * " have higher precedence than "+")

$$
1+2+2+2
$$

6. a) What is Lex tool ?
b) Specify the input format.
c) If identifiers and keywords follow the same structure then how does the Lex differentiae them ?
d) If you use Lex tool to write Lexical Analyser, do you still need to design any automat?
$3+2+1+1$

7. a) Draw a DFA State Transition Diagram for identification of unsigned positive numbers with optional fraction part. (Examples : 1, 12, 123•45 )
b) Identify the lexemes with tokens from the following lines of $C$ code.
```
void stat();
main() {
    int counter;
for (counter = 0; counter < 5; counter++) {
    stat ();
                }
                } 3 + 4
```

8. a) Type checking in compiler : is tit static or dynamic ?
b) It is done in which phase of the compiler? What is the input to it?
c) What kind of grammar is required to implement type checking?
d) What kind of grammar is required to implement intermediate code generation?
e) What is the use of back patching? $2+2+1+1+1$

## GROUP - B

## ( Long Answer Type Questions )

Attempt any five

$$
5 \times 7=35
$$

9. Discuss on the role of parsing in compilation of a program.

Discuss on Chomsky hierarchy of grammars with suitable examples.

CS/M.Tech(CSE)/SEM-2/MCSE-204/2013

10. What is ambiguous grammar ? Check whether the CFG $S \rightarrow a S b|S S| \square$ ambiguous. Design anon-ambiguous grammar for arithmetic expression involving symbols : +, *, -, $\div$, (, ), and numerical digits.
11. What is left-recursion issue in parsing? Write an algorithm for removal of indirect left-recursion. Remove left-recursion issue from the grammar $: S \rightarrow A a|b ; A \rightarrow A c| S d \mid f$.
12. How does left-factoring make sense for predictive parsing ? Write left-factoring algorithm. Apply left-factoring for the grammar $: S \rightarrow i E i S|i E t S e S| a ; E \rightarrow b$.
13. Design a top-down recursive descent parser for the following grammar with necessary transformation.
$E \rightarrow E+E\left|E{ }^{*} E\right|(E) \mid$ id
14. Construct predictive parsing table for the following grammar :
$E \rightarrow T E^{\prime} ; E^{\prime} \rightarrow+T E^{\prime}\left|\square ; T \rightarrow F T^{\prime} ; T^{\prime} \rightarrow{ }^{*} F T^{\prime}\right| \square ; F \rightarrow(E) \mid$ id.
15. Construct SLR parsing table for the following grammar :
$S \rightarrow E ; E \rightarrow E+T\left|T ; T \rightarrow T^{*} F\right| F ; F \rightarrow i d$.

