

**GUJARAT TECHNOLOGICAL UNIVERSITY****BE - SEMESTER-VII(NEW) EXAMINATION – SUMMER 2019****Subject Code: 2170701****Date: 10/05/2019****Subject Name: Compiler Design****Time: 02:30 PM TO 05:00 PM****Total Marks: 70****Instructions:**

1. Attempt all questions.
2. Make suitable assumptions wherever necessary.
3. Figures to the right indicate full marks.

		MARKS
<b>Q.1</b>	(a) Define lexemes, patterns and tokens.	<b>03</b>
	(b) Differentiate compilers and interpreters.	<b>04</b>
	(c) Explain analysis of source program for compilers.	<b>07</b>
<b>Q.2</b>	(a) Give regular definition for signed and unsigned numbers.	<b>03</b>
	(b) Check whether the following grammar is ambiguous or not. $S \rightarrow (S) S$ $S \rightarrow \epsilon$	<b>04</b>
	(c) Draw DFA from regular expression without constructing NFA. $(a   b   c)^* a (b   c)^* \#$	<b>07</b>
	<b>OR</b>	
	(c) Draw NFA from regular expression using Thomson's construction and convert it into DFA. $(a   b)^* a b^* a$	<b>07</b>
<b>Q.3</b>	(a) Define handle and handle pruning.	<b>03</b>
	(b) Construct operator precedence relations table for following grammar. $E \rightarrow E + E$ $E \rightarrow E - E$ $E \rightarrow E * E$ $E \rightarrow (E)$ $E \rightarrow id$ Assume suitable operator associativity and precedence.	<b>04</b>
	(c) Construct recursive descent parser for following grammar. $E \rightarrow T A$ $A \rightarrow + T A$ $A \rightarrow \epsilon$ $T \rightarrow F B$ $B \rightarrow * F B$ $B \rightarrow \epsilon$ $F \rightarrow ( E )$ $F \rightarrow id$	<b>07</b>
	<b>OR</b>	
<b>Q.3</b>	(a) Differentiate top down parsing and bottom up parsing.	<b>03</b>
	(b) Construct syntax directed translation scheme for infix to postfix conversion.	<b>04</b>
	(c) Construct LL(1) parsing table for following grammar. Check whether the grammar is LL(1) or not. $A \rightarrow A a B$ $A \rightarrow x$ $B \rightarrow B C b$ $B \rightarrow C y$ $C \rightarrow C c$ $C \rightarrow \epsilon$	<b>07</b>

- Q.4** (a) Define Intermediate code and its importance. **03**  
 (b) Construct LR(0) item sets for following grammar. **04**  
 $S \rightarrow AaAb$   
 $S \rightarrow BbBa$   
 $A \rightarrow \epsilon$   
 $B \rightarrow \epsilon$   
 (c) Explain various error recovery schemes in detail. **07**
- OR**
- Q.4** (a) Differentiate LR(1) and LALR(1) parsers. **03**  
 (b) Construct syntax tree and DAG for following expression. **04**  
 $a = (b+c+d) * (b+c-d) + a$   
 (c) Explain quadruples, triples and indirect triples with examples. **07**
- Q.5** (a) Define basic block with a simple example. **03**  
 (b) Explain activation record. **04**  
 (c) Explain various methods of peephole optimization. **07**
- OR**
- Q.5** (a) Explain static storage allocation. **03**  
 (b) Explain any two parameter passing methods. **04**  
 (c) Explain various issues in design of code generator. **07**

\*\*\*\*\*