



Reg No.: \_\_\_\_\_

Name: \_\_\_\_\_

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**  
**SIXTH SEMESTER B.TECH DEGREE EXAMINATION(R&S), MAY 2019**

**Course Code: CS304**  
**Course Name: COMPILER DESIGN**

Max. Marks: 100

Duration: 3 Hours

**PART A***Answer all questions, each carries 3 marks.*

Marks

- |   |  |     |
|---|--|-----|
| 1 | Describe input buffering scheme in lexical analyzer.   | (3) |
| 2 | Construct a regular expression to denote a language L over $\Sigma = \{0,1\}$ accepting all strings of 0's and 1's that do not contain substring 011 | (3) |
| 3 | Consider the context free grammar $S \rightarrow aSbS \mid bSaS \mid \epsilon$<br>Check whether the grammar is ambiguous or not                      | (3) |
| 4 | What is Recursive Descent parsing? List the problems faced in designing such a parser  | (3) |

**PART B***Answer any two full questions, each carries 9 marks.*

- |   |   |     |
|---|---|-----|
| 5 | a) Explain the different phases in the design of a compiler.  | (5) |
|   | b) Find the FIRST and FOLLOW of the non-terminals in the grammar<br>$S \rightarrow aABe$<br>$A \rightarrow Abc \mid b$<br>$B \rightarrow d$         | (4) |
| 6 | a) Design a recursive descent parser for the grammar<br>$E \rightarrow E + T \mid T$<br>$T \rightarrow T * F \mid F$<br>$F \rightarrow (E) \mid id$ | (5) |
|   | b) Develop a lexical analyzer for the token identifier.   | (4) |
| 7 | a) What is left recursive grammar? Give an example. What are the steps in removing left recursion?  | (5) |
|   | b) Explain any four compiler writing tools  | (4) |

**PART C***Answer all questions, each carries 3 marks.*

- 8 Explain the main actions in a shift reduce parser (3)
- 9 What are different parsing conflicts in SLR parsing table? (3)
- 10 What are annotated parse trees? Give examples. (3)
- 11 What are L-attributed definitions and S-attributed definitions in a syntax directed translation scheme? (3)

**PART D***Answer any two full questions, each carries 9 marks.*

- 12 a) Find the LR(0) items for the grammar (4)  
 $S \rightarrow SS \mid a \mid \epsilon$ .
- b) Explain bottom-up evaluation of s-attributed definitions. (5)
- 13 a) Derive LALR (1) parsing algorithm for following grammar (6)  
 $S \rightarrow AS/b$   
 $A \rightarrow SA/a$
- b) Design a type checker for simple arithmetic operations. (3)
- 14 a) Explain the syntax directed definition of a simple desk calculator. (5)  
Explain operator grammar and operator precedence parsing (4)

---

**PART E***Answer any four full questions, each carries 10 marks.*

- 15 a) Explain storage organization and storage allocation strategies (10)
- 16 a) Explain intermediate code generation of an assignment statement (10)
- 17 a) Explain quadruples, triples and dags with an example each. (10)
- 18 a) Explain the principal sources of optimization (10)
- 19 a) Explain optimization of basic blocks (5)  
b) With suitable examples explain loop optimization. (5)
- 20 a) Explain issues in design of a code generator (5)  
b) Explain simple code generation algorithm (5)

\*\*\*\*\*