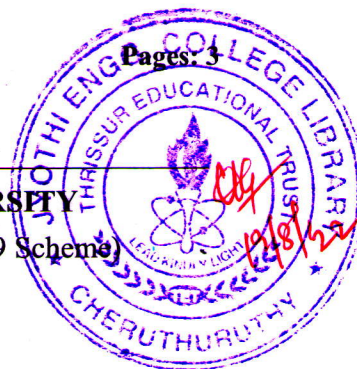


Reg No.: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Sixth Semester B.Tech Degree Examination June 2022 (2019 Scheme)

**Course Code: CST302****Course Name: COMPILER DESIGN**

Max. Marks: 100

Duration: 3 Hours

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

Marks

- | | | |
|----|---|-----|
| 1 | Find the lexemes in the following programming statement.
sum = a * (b -10) ;
Define tokens and patterns for the above statement. | (3) |
| 2 | Explain the importance of sentinels in input buffering used in lexical analysis | (3) |
| 3 | With an example write the steps to remove left recursion? | (3) |
| 4 | Find FIRST set and FOLLOW set of each nonterminal in the following grammar
E → E A E (E) - E id
A → + * | (3) |
| 5 | What are viable prefixes? | (3) |
| 6 | What are the different parsing conflicts in the SLR parsing table? | (3) |
| 7 | Differentiate between synthesized attributes and inherited attributes with an example. | (3) |
| 8 | What is the role of activation record in compiler design? | (3) |
| 9 | Explain code motion with an example. | (3) |
| 10 | Write the algorithm for partitioning a sequence of three-address instructions into basic blocks | (3) |

PART B*Answer one full question from each module, each carries 14 marks.***Module I**

- | | | |
|----|--|-----|
| 11 | a) Explain the working of different phases of a compiler. Illustrate with a source language statement. | (8) |
| | b) Explain different compiler construction tools. | (6) |

OR

- | | | |
|----|--|-----|
| 12 | a) Explain the role of transition diagrams in recognition of tokens. | (7) |
| | b) Explain bootstrapping with an example. | (7) |

Module II

- 13 a) i. Show that the grammar $S \rightarrow iCtSeS \mid iCtS \mid b$, $C \rightarrow a$ is ambiguous. (6)
 ii. Eliminate ambiguity from the above grammar.
 b) Construct a Recursive descent Parser for handling Arithmetic Expressions. (8)

OR

- 14 a) Write Non-recursive predictive parsing algorithm. (6)
 b) Prove that the following grammar is not LL(1) (8)

$$S \rightarrow iEtSS' \mid a$$

$$S \rightarrow eS \mid \epsilon$$

$$E \rightarrow b$$

Module III

- 15 a) Construct canonical LR(0) collection of items for the grammar below. (9)
 $S \rightarrow L = R \mid R$
 $L \rightarrow * R \mid id$
 $R \rightarrow L$
 Prove that this grammar is not SLR(1).
 b) What is handle pruning? Indicate the handles in the reduction of the sentence aaabbb to the start symbol using the grammar (5)
 $S \rightarrow aABb$, $A \rightarrow aA \mid a$, $B \rightarrow bB \mid b$

OR

- 16 a) Derive LR (1) parsing table for following grammar (9)
 $S \rightarrow Aa \mid bAc \mid Bc \mid bBa$
 $A \rightarrow d$
 $B \rightarrow d$
 b) Write all moves by the LR parser for parsing the input 'bdc'. [use the parsing table created in question number 16.a] (5)

Module IV

- 17 a) Write the SDD for a simple type declaration and draw the annotated parse tree for the declaration float a, b, c. (7)
 b) With an SDD for a desk calculator, write the steps involved in the bottom up evaluation for the expression $(3*5)-2$. (7)

OR

- 18 a) Explain static allocation and heap allocation strategies. (7)
b) Construct the DAG and three address code for the expression $a+a*(b-c)+b*(b-c)+b$ (7)

Module V

- 19 a) With suitable examples explain loop optimization techniques (7)
b) With suitable example of a basic block, explain the code-improving transformations of a basic block. (7)

OR

- 20 a) Explain issues in design of a code generator (6)
b) Write the code generation algorithm. Using this algorithm generate code sequence for the expression $x = (a - b) + (a + c) + (a + c)$ (8)
