

1200CST302052302

Reg No.: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

B.Tech Degree S6 (R, S) / S6 (PT) (R) Examination June 2023 (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 | What is the relevance of input buffering in lexical analysis? | (3) |
| 2 | Draw a transition diagram to represent relational operators. | (3) |
| 3 | With an example write the steps to remove left recursion. | (3) |
| 4 | What is left factoring? Left factor the following grammar, C | (3) |
| | $E \rightarrow E + T \mid T$ | |
| | $T \rightarrow \text{float} \mid \text{float} * T \mid (E)$ | |
| 5 | Differentiate CLR and LALR parsers. | (3) |
| 6 | What are the possible actions of a shift-reduce parser? | (3) |
| 7 | Convert the expression $a = b * -c + b * -c$ into quadruple? | (3) |
| 8 | Define SDD with an example. | (3) |
| 9 | Explain common sub expression elimination with an example. | (3) |
| 10 | How the peephole optimization technique makes its role in the compilation process? | (3) |

PART B

Answer one full question from each module, each carries 14 marks.

Module I

- | | | |
|----|---|-----|
| 11 | a) What are the various phases of a compiler? Explain each phase in detail by using the input statement. <code>position := initial + rate * 60</code> | (8) |
| | b) Differentiate tokens, patterns and lexemes with the help of an example. | (6) |

OR

- | | | |
|----|--|-----|
| 12 | a) Write short notes on compiler construction tools. | (6) |
| | b) Explain in detail about buffer pairs and sentinels. | (8) |

Module II

- 13 a) Find FIRST set and FOLLOW set of each nonterminal in the following grammar. (6)

$$S \rightarrow aBDh \mid bBc$$

$$B \rightarrow eC$$

$$C \rightarrow bC \mid \epsilon$$

$$D \rightarrow EF$$

$$E \rightarrow g \mid \epsilon$$

$$F \rightarrow f \mid \epsilon$$

- b) Explain the error recovery strategies in parsing. (8)

OR

- 14 a) i) Show that the given grammar is ambiguous or not. (7)

$$E \rightarrow E + E \mid E - E$$

$$E \rightarrow E * E \mid E / E$$

$$E \rightarrow E \wedge E$$

$$E \rightarrow (E) \mid id$$

Also eliminate ambiguity from the above grammar.

(Precedence order: id, (, ^, *, /, + and -)

- b) Construct a non-recursive predictive parsing table for the following grammar: (7)

$$S \rightarrow (L) \mid a$$

$$L \rightarrow L, S \mid S$$

Also prove that the grammar is LL(1) or not.

Module III

- 15 a) What is a shift-reduce parser? Explain in detail the conflicts that may occur during shift-reduce parsing. (8)

- b) Construct canonical LR(1) collection of items for the grammar below: (6)

$$S \rightarrow CC$$

$$C \rightarrow aC$$

$$C \rightarrow d$$

OR

- 16 Consider the grammar (14)

$$S \rightarrow Aa \mid bAc \mid dc \mid bda$$

$$A \rightarrow d$$

Construct a LALR parsing table for the grammar given above. Verify whether the input string "bdc" is accepted by the grammar or not.

Module IV

- 17 a) Define the following terms and give suitable example for each. (6)
 i) Three-address code ii) Triples iii) Quadruples.
 b) Explain static allocation and heap allocation strategies. (8)

OR

- 18 a) Construct a syntax directed definition for an assignment statement. (7)

$S \rightarrow E$

$E \rightarrow E1 + E2$

$E \rightarrow E1 * E2$

$E \rightarrow - E1$

$E \rightarrow (E1)$

$E \rightarrow id$

Also construct an annotated parse tree for the input string: $6 * 8 + 5$.

- b) Generate an intermediate code for the following code segment along with the required syntax-directed translation scheme. (7)

if ($a > b$)

$x = a + b$

else

$x = a - b$

where a and x are of real and b of int type data.

Module V

- 19 a) Explain different code optimization techniques. (8)
 b) Generate a code sequence for the assignment $d = (a - b) + (a - c) + (a - c)$. (6)

OR

- 20 a) Explain the design issues of a code generator. (7)
 b) Illustrate the optimization of basic blocks with examples. (7)
