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 (6)

S -> iCtSeS | iCtS | b , C -> a is ambiguous.

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 → iEtSS' | a

S → eS | ε

E → b

Module III

15 a) Construct canonical LR(0) collection of items for the grammar below. (9)

S → L = R | R

L → * R | id

R → 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 → aABb , A → aA | a , B → bB | b

OR

16 a) Derive LR (1) parsing table for following grammar (9)

S → Aa | bAc | Bc | bBa

A → d

B → 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)

****

Reg No.:_____ Name:_____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Sixth Semester B.Tech Degree Supplementary Examination May 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 Construct a regular expression for the language that consists of all strings ending with 00 over $\Sigma = \{0,1\}$. (3)

2 Define tokens, lexemes and patterns with suitable examples for each. (3)

3 Given a grammar : (3)

$S \rightarrow (L) | a$

$L \rightarrow L,S | S$

(i) Is the grammar ambiguous? Justify.

(ii)Build a parse tree for the string (a,((a,a), (a,a))).

4 Compute the FIRST and FOLLOW for the following Grammar. (3)

$S \rightarrow SS | AB$

$A \rightarrow Aa | a$

$B \rightarrow Bb | b$

5 Define an operator grammar. Give an example. (3)

6 What are viable prefixes? (3)

7 Explain quadruples, triples and indirect triples with suitable examples. (3)

8 What are L-attributed definitions and S-attributed definitions in a syntax directed translation scheme? (3)

9 Construct the syntax tree and then draw the DAG for the statement: (3)

e := (a*b) + (c-d) *(a*b)

10 Explain any three issues in the design of a code generator. (3)

PART B

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

Module I

11 a) Explain in detail the various phases of the compiler with a neat diagram. Illustrate the output of each phase for the input, (9)

sum := a + b * 30

where a and b are float variables.

b) Apply bootstrapping to develop a compiler for a new high level language N on machine P. (5)

OR

12 a) Explain the role of transition diagrams in recognition of tokens. Draw the transition diagram for the regular definition: (8)

relop → < | <= | = | <> | >= | >

b) List and explain any three tools that help a programmer in building a compiler efficiently. (6)

Module II

13 a) Consider the following grammar: (9)

E→ E + T | T

T→ T * F |F

F→ ~ F | (E) | id

(i) Remove left recursion from the grammar.

(ii) Construct a predictive parsing table.

(iii) Justify the statement " The grammar is LL (1)".

b) Design a recursive descent parser for the grammar: S→cAd, A→ab/ b (5)

OR

14 a) Left factor the following grammar and then obtain LL(1) parsing table. (7)

S → TL;

T → int | float

L → L , id | id

Is the grammar LL(1)? Justify your answer.

b) Write all the moves by the LL(1) parser for parsing the input "int id,id;". [Use the parsing table created in question number 14.a] (7)

Module III

15 a) Construct canonical LR(0) collection of items for the grammar below. (10)

S → L = R

S → R

L → * R

L → id

R→ L

Also identify a shift reduce conflict in the LR(0) collection constructed above.

b) Write an algorithm for computing the closure of an LR(0) items. (4)

OR

16 a) Construct LALR parse table for the grammar: $A \rightarrow BB,\ B \rightarrow bB \mid d$ (10)

b) What are the different parsing conflicts in the SLR parsing table? (4)

Module IV

17 a) Write the SDD for a desk calculator and draw the annotated parse tree for the expression: $4 * 5 + 6 - (3 * 2)$ (8)

b) Explain bottom- up evaluation of s-attributed definitions. (6)

OR

18 a) Write syntax directed definition to construct syntax tree and three address code for assignment statements. (7)

b) Explain static allocation and heap allocation strategies. (7)

Module V

19 a) With suitable examples explain the following loop optimization techniques: (7)
(i) Code motion (ii) Induction variable elimination and (iii) strength reduction

b) Explain the optimization of basic blocks. (7)

OR

20 a) For the following C statement, write the three-address code and quadruples. (8)

S:= A – B + C * D – E + F

Also convert the three-address code into machine code.

b) Write the Code Generation Algorithm and explain the *getreg* function. (6)

****

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. $\text{position} := \text{initial} + \text{rate} * 60$ (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)

$$S \rightarrow aBDh \mid bBc$$

$$B \rightarrow eC$$

$$C \rightarrow bC \mid \in$$

$$D \rightarrow EF$$

$$E \rightarrow g \mid \in$$

$$F \rightarrow f \mid \in$$

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 /, + 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 -> E

E -> E1 + E2

E -> E1 * E2

E -> - E1

E -> (E1)

E -> 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)

****