Reg No.:_____ Name:_____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

B.Tech Degree S6 (R, S) / S6 (PT) (R, S) Examination May 2024 (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	Define lexeme, tokens and patterns using the following source language statement as example while(a>b){result= a+ b;}	(3)
2	Describe bootstrapping in compiler design using necessary diagrams.	(3)
3	Write the algorithm to remove left recursion and remove left recursion from the following grammar E->E0E0S | 00 S->S11 | 1	(3)
4	Write the algorithm of left factoring. Left factor the following grammar S->abAA |ab A->abA|ab	(3)
5	Define operator grammar. Construct operator precedence table for the following grammar E->E+E|E*E|id.	(3)
6	What are the different operations in a shift reduce parser	(3)
7	Explain the structure of activation record.	(3)
8	Compare L-attributed and S-attributed Syntax directed definitions.	(3)
9	Construct the syntax tree and then draw the DAG for the statement d+a*(b-c) + (b-c)*d	(3)
10	With suitable example explain induction variable elimination technique for loop optimization.	(3)

PART B

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

Module I

11	a)	Explain the different phases of a compiler for a source language statement	(8)

c = sum - row * 10. Show the input and output at each of the compiler phases. Assume c, sum and row are floating point variables

b) Draw transition diagrams to represent the following (6)
(i)relational operators (ii)identifiers.

OR

12 a) Write a note on input buffering with necessary diagrams. Specify (8)
the advantages of using two buffer system and sentinels in input buffering.

b) Explain any four compiler construction tools. (6)

Module II

13 a) Consider the following grammar (8)
S-> (L) | a
L-> L,S | S
(i)Remove left recursion from the grammar.
(ii)Construct a predictive parsing table.
(iii)Justify the statement " The grammar is LL (1)"

b) Write the recursive descent parsing procedure for the following grammar. (6)
S -> iCtSS' | a
S' - > eS | ε
C -> b

OR

14 a) Write algorithms for finding FIRST and FOLLOW sets. What is the role of FIRST and FOLLOW in a predictive parser? (8)

b) Compute the FIRST and FOLLOW sets for the non terminals in (6)
the following Grammar

S -> ADB | DbB | Ba
A ->da | BD
B -> g | Ɛ
D -> h | Ɛ

Module III

15 a) Construct SLR parsing table for the following grammar. Check if the grammar is SLR or not. Justify your answer. (8)

E->T +E |T
F-> id

b) What are handle and handle pruning? Indicate the handles in the reduction of a (6)

sentence bbaaab to the start symbol using the grammar

S ->aB | bA
A -> a | aS | bAA
B -> b | bS | aBB

OR

16 a) Construct CLR parsing table for the given grammar. Is the grammar is CLR grammar? Justify your answer. (8)

S->L=R | R
L->*R | id
R->;L

b) Write a note on different conflicts in shift reduce parser. (6)

Module IV

17 a) With the help of syntax directed definition of a simple desk calculator, evaluate the expression (3+5/2)*(2+4/3) and draw the annotated parse tree (8)

b) Write an SDD for a simple type declaration. With the SDD of simple type declaration, write the steps involved in the evaluation of inherited attributes for the statement given below. (6)

int a,b,c

OR

18 a) Write a note on stack allocation, heap allocation and static allocation strategies. (8)

b) What are the different representations of three address code? Write the different three address code representations of the following expression. (6)

(a+b)*(b+c)*(a+b+c)

Module V

19 a) Explain any four principal sources of optimization (8)

b) What is a basic block? Explain about the structure preserving transformations on a basic block? (6)

OR

20 a) Explain peephole optimization techniques with example. (8)

b) Explain any three issues in the design of a code generator (6)

****