A 1200CST302042502 Pages: 3

JYOTHI ENGG. COLLEGE LIBRARY
THRISSUR EDUCATIONAL TRUST
LEAD KINDLY LIGHT
CHERUTHURUTHY
29/4/25

Reg No.:_______________ Name:__________________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

B.Tech Degree S6 (R, S) / S6 (PT) / (WP) Exam April 2025 (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 token and lexeme with examples. (3)

2 What are the different error recovery strategies in lexical phase? (3)

3 Consider the grammar (3)

S → AB

A → aaA | Є

B →Bb | Є

Derive the string **aab** using both leftmost derivation and right most derivation.

4 What is ambiguous grammar? Give example. (3)

5 Write the algorithm to compute FIRST(X) and FOLLOW(X). (3)

6 What is an operator grammar? Give example. (3)

7 Write the SDT to evaluate arithmetic expressions for the following grammar (3)

E -> E + T | T

T -> F

F -> num .

8 Translate a[i] = b*c - b*d into quadruple. (3)

9 What are the three characteristics of peephole optimization? (3)

10 List out the source language issues that occur in the code generation phase of a compiler. (3)

PART B

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

Module I

11 a) Explain the different phases of compiler with a neat diagram. Illustrate the output of each phase for the input: **p := i + r * 10** where **p, i** , and **r** are integer variables. (10)

153

b) List out any four compiler writing tools. (4)

OR

12 a) Explain the front-end back end model of compiler?. (5)

b) Explain about recognizing tokens with help of transition diagrams of relational operators, identifiers and digits. (9)

Module II

13 a) Define Ambiguity. Show S-> aSbS | bSaS | ϵ is an ambiguous grammar. (4)

b) Explain the algorithm for LL (1) parsing table construction. Construct a LL (1) parsing table for the grammar. (10)

S → (S) S | ϵ

OR

14 a) Consider the following grammar

S → (L) | a

L → L,S | S

(i)Remove left recursion from the grammar. (2)

(ii)Construct a predictive parsing table. (4)

(iii)Justify the statement " The grammar is LL (1)" (2)

b) What is recursive descent parsing and write its drawbacks? Write a procedure for a typical nonterminal in a top down parser. (6)

Module III

15 a) Construct the LR (0) items for the following grammar (7)

S → L=R

S → R

L → * R

L → id

R → L

b) Explain the different conflicts in LR (0) parsing with an example. (7)

OR

16 a) Check whether the following grammar is CLR or not (7)

S → XX

X → aX | b

b) Construct SLR parsing table for the grammar (7)

E → E + n | n

Module IV

17 a) Differentiate synthesized attributes and inherited attributes with example. (5)

b) With an SDD for a desk calculator, draw the annotated parse tree and write the steps involved in the bottom up evaluation for the expression (4*5) –2. (9)

OR

18 a) Construct the three address code and DAG of the following statement (6)

s= (a-b) * (b+c) + (a-b)

b) Explain static allocation and heap allocation strategies. (8)

Module V

19 a) What are the needs for optimization phase in compiler? (4)

b) Explain about the principal sources of optimization. (10)

OR

20 a) Explain about the issues in design of a code generator. (8)

b) Explain about register and address descriptors. (6)

****