USN : _____ Course Code : 16CS63

Sixth Semester B.E Makeup Examination, SEPT._OCT._2020

COMPILER DESIGN

Time: 3 hrs Max.Marks :100

Instructions :1. Answer any Five full Questions selecting at least One Full Question from Each Unit. 2. Each Question carry Equal Marks. 3. Missing Data may be suitably assumed. 4. Draw Figures wherever necessary.

MODULE 1 **L CO PO M**

1a. Explain with a diagram the phases of a Compiler. Show the transition made by each of these phases for the statement a=b + c * 5, where a , b and c are reals. [2] [1] [1] [8]

1b. Explain typical Language Processing System with a neat diagram [2] [1] [1] [6]

1c. Design a transition diagram to recognize the following tokens.
i) Integer constant ii) Identifier [3] [1] [3] [6]

OR

2a. Explain with a neat diagram the interaction between Lexical Analyser and the parser. [2] [1] [1] [6]

2b. Design the transition diagram and hence write program to recognize the token below
i) Relational operator ii)unsigned number [3] [1] [3] [8]

2c. What are the applications of a Compiler? Explain. [2] [1] [1] [6]

MODULE 2

3a. Give the algorithm for the Left Recursion .Apply the technique and eliminate Left recursion from the following grammar.

A→BC | a
B→CA |Ab
C→AB|CC|a

3b. Explain Panic mode and phrase level error recovery strategies. [3] [1] [3] [8]

3c. Find the First and Follow for the given grammar [2] [1] [1] [6]

E→ E+T | T
T→T*F | F
F→(E) | id

OR [3] [1] [3] [6]

4a. Develop predictive parsing table for the following grammar ,show the moves made by the parser for the given input string (a,(a,a))

S→(L) | a
L→L , S | S

4b. Give the algorithm for constructing First and Follow sets with an example. [3] [1] [3] [8]

4c. Write an Algorithm to Left Factor a grammar .Give the Left Factored grammar for the following. [2] [1] [1] [6]

S→iEtS | iRtSeS | a
E→b

[3] [1] [3] [6]

MODULE 3

5a. Construct LALR parsing table for the given grammar. S' →S S→ CC C→cC | d [3] [1] [3] [8]

5b. Explain the working of Shift reduce parser. [2] [1] [1] [6]

5c. Analyse the grammar and hence find LR(0) items

S→(S) S

S→€ [4] [1] [3] [6]

OR

6a. Construct SLR Parsing Table for the following grammar. Show the moves made by the parser for the input aa*a+ S→SS+ | SS*| a [3] [1] [1, 3] [10]

6b. Write an algorithm for constructing SLR parsing table and hence Explain the conflicts of Shift reduce parsing with suitable examples [2] [1] [1] [10]

MODULE 4

7a. Explain the parser stack implementation of postfix SDT with an example [2] [2] [1] [8]

7b. Construct Directed Acyclic Graph for the expression

a + a * (b – c) + (b – c) * d [3] [2] [3] [6]

7c. Explain the following with an example

i) Quadruples ii) Triples iii) Indirect Triples [2] [2] [1] [6]

OR

8a. Write Syntax Directed definition for flow of control statements [2] [2] [1] [8]

8b. Construct a Dependency Graph for the declaration float id1 , id2 , id3 [3] [2] [3] [6]

8c. Write annotated parse tree for 6 * 5 + 7m using top down approach . write semantic rules for each. [3] [2] [3] [6]

MODULE 5

9a. Discuss the issues in the design of a code generator [2] [3] [1] [10]

9b. Illustrate with an example , common sub expression and dead code elimination methods [2] [3] [1] [10]

OR

10a. Apply the code generation algorithm to translate the basic block shown below

t = a- b

u = a – c

v = t + u

a = d

d = v + u

assume t , u , v are temporaries local to the block while a , b , c, d are variables that are line on exit from the block [3] [3] [3] [10]

10b. Explain the following with suitable examples

i) Basic blocks

ii) Flow Graphs [2] [3] [1] [10]

USN | | | | | | | | | | |

15CS63/16CS63

Sixth Semester B.E. Fast Track Semester End Examination, July/August 2019
COMPILER DESIGN

Time: 3 Hours Max. Marks: 100

Instructions: 1. *UNIT I & V are Compulsory.*
2. *Answer any one full question from remaining each UNITS.*
3. *Assume any missing information*

UNIT – I (Compulsory)

			L	CO	PO	M
1	a.	With suitable example explain the role of LEXICAL ANALYSER in the compilation process and interaction between parser.	(2)	(1)	(1,12)	(10)
	b.	Construct Transition diagram for token **Relation Operators** (<, <=, >, >=, <>, =) of PASCAL language and design a lexical analyzer in C++ using techniques suitable for hand implementation to recognize the same. Assume suitable C++ functions to read, failure and retract operations.	(6)	(1)	(3, 5)	(10)

UNIT – II

			L	CO	PO	M
2	a.	Consider the Grammar **S→ S + S | SS | (S) | S* | a** 1. Write LMD and RMD for the input string **w = (a + a) * a** 2. Construct Parse Tree for the same string 3. What is ambiguous grammar ? Is the above grammar ambiguous? Justify.	(3)	(1)	(12)	(10)
	b.	What is Left Recursion? Give the Algorithm to eliminate left recursion. Apply the Algorithm and eliminate the left recursion for the following grammar. **A→ABd | Aa | a** **B→Be | b**	(2)	(1)	(3,12)	(10)

OR

			L	CO	PO	M
3	a.	What is the need of left factoring? Given the grammar **A→ aAB | aBc | aAc** **B→D** 1. Left factor the above grammar 2. Define FIRST and FOLLOW symbols and Construct FIRST and FOLLOW sets for above grammar	(4)	(1)	(3, 12)	(10)
	b.	Explain the model of Predictive Parser with Parsing Algorithm.	(2)	(1)	(3,12)	(10)

UNIT - III

			L	CO	PO	M
4	a.	What is Handle? For the following grammar **S→ SS+ | SS* | a** 1. Indicate the handle for right sentential forms. **aaa*a++** 2. Write the configuration of shift Reduce parser for the input : **aaa*a++**	(3)	(1)	(1,3)	(10)
	b.	Explain the working Model of LR parser with Parsing Algorithm	(2)	(1)	(1,12)	(10)

Note: L (Level), CO (Course Outcome), PO (Programme Outcome), M (Marks)

OR

5 a What is LR(0) item in SLR parser? Give the Algorithm to build the collections of sets of valid LR(0) items along with two procedures **CLOSURE and GOTO**

(2) (1) (1,12) (10)

b Construct canonical sets of LR(0) items for the following grammar.

S→ 0 S 1 | 01

(3) (1) (1,3) (10)

UNIT – IV

L CO PO M

6 a. Define Synthesized Attribute and Give SDD for simple calculator and Draw the Annotated parse tree for expression 3+5 * 4n

(3) (2) (3,5) (10)

b. Define Inherited Attribute and Give SDD for simple **TYPE declaration of C-language for Float and Integer data-types.** Draw the Annotated parse tree for type declaration **float a, b, c**

(3) (2) (3,5) (10)

OR

7 a. Write S-Attributed Definition to Construct Syntax Trees for Simple arithmetic expression and Indicate the steps of construction of syntax tree for the expression **a-4*c**

(3) (2) (3,5) (10)

b. What is Three Address Code ? Construct the Three Address code for the following expression

1. **a + b + (a +b)**
2. **a + a * (b – c) + (b – c) * d**

(4) (2) (3,5) (10)

UNIT –V (Compulsory)

L CO PO M

8 a. Generate the Target code for the following three address statements assuming all variables are in memory locations.

1. **x=y-z**
2. **b=a[i]**
3. **if x < y goto L**

(4) (3) (3, 12) (10)

b. Discuss the following code optimization techniques

1. Finding Local Common Sub-expression and elimination
2. Loop Optimization

(2) (3) (5,12) (10)

USN | | | | | | | | | | |

15CS63

Sixth Semester B.E. Makeup Examination, June 2018

COMPILER DESIGN

Time: 3 Hours **Max. Marks: 100**

Instructions: *1. Units IV and V are compulsory. Answer any one question from each of the remaining units.*
2. Provide examples, wherever needed.

UNIT – I

1 **a.** Explain with a neat diagram, the phases of a compiler. Show the translation of the input statement: position = initial + rate * 60 (Assume position, initial and rate to be floats) **14 M**

(Level [2], CO [1], PO [1])

b. Design a lexical analyzer in C++ for relational operators of 'C' programming language. **06 M**

(Level [6], CO [1], PO [3])

OR

2 **a.** Show with a neat diagram, the interaction between a lexer and a parser. **12 M**
Define the terms: token, pattern and lexeme. Identify the tokens generated for the 'C' input: while (I < 10)

(Level [1,2,3], CO [1], PO [1,3])

b. Compare the two schemes for input buffering used by a lexer. **08 M**

(Level [2], CO [1], PO [1])

UNIT – II

3 **a.** What is Left Recursive grammar and how would you eliminate it? Apply the technique and eliminate left recursion from the following Grammar. **06 M**

S→AS | b
A→SA | a

(Level [1,3], CO [2], PO [3, 5])

b. Explain the model of predictive parser with parsing Algorithm. **06 M**

(Level [2], CO [2], PO [1,3])

c. Design the parsing table for predictive parser for the following grammar and verify the grammar to be LL(1). **08 M**

S→iEtS | iEtSeS | a
E→b

(Level [3, 4], CO [2], PO [3])

OR

4 **a.** Consider the grammar **06 M**

S→ a | ^ | (T)
T→ T, S | S

Show the Leftmost and Rightmost derivation for the following sentence . (a,(a, a))

(Level [2], CO [2], PO [3])

b. Explain Recursive-Descent parsing algorithm and discuss the difficulties to implement Recursive Descent Parser. **06 M**

(Level [2], CO [2], PO [1])

c. What is Left Factoring? Explain the algorithm for Left factoring the grammar G. Apply the technique and left factor the following grammar. **08 M**

$S \rightarrow aS \mid Aa \mid Bb$
$A \rightarrow abB \mid aB \mid cdg \mid cdeB \mid cdfB$
$B \rightarrow b$

(Level [2, 3], CO [2], PO [3])

UNIT – III

5 a. Construct SLR Parsing Table for the following grammar: **12 M**

$S \rightarrow SS+ \mid SS* \mid a$

Show the moves made by the parser on input: aa*a+

(Level [6,2], CO [2], PO [3,5])

b. Explain the working of a Shift Reduce Parser. Show with suitable examples the conflicts that may occur during shift reduce parsing. **08 M**

(Level [2], CO [2], PO [1])

OR

6 a. What is an LR(1) item? Construct canonical LR(1) collection of items and the Automaton for the following grammar: **12 M**

$S \rightarrow CC$
$C \rightarrow aC \mid d$

(Level [1,6], CO [2], PO [1,3,5])

b. Compare LL and LR Parsing methods. Also Compare the different kinds of LR parsers **08 M**

(Level [2], CO [2], PO [1])

UNIT – IV

7 a. How would you define Syntax Directed Definition? Construct semantic rules for the following grammar and show the annotated parse for the string 3+5*4n **06 M**

$L \rightarrow En$
$E \rightarrow E+T$
$E \rightarrow T$
$T \rightarrow T*F$
$T \rightarrow F$
$F \rightarrow (E)$
$F \rightarrow digit$

(Level [1, 3], CO [3], PO [1, 3])

b. What is DAG? Construct a DAG, Three address code, Quadruple and Triple representation for the following expression. **10 M**

a + a * (b-c) + (b-c) *d.

(Level [1, 3], CO [4], PO [1, 3])

c Construct the semantic rules for translation of while statement of C language. **04 M**

(Level [3], CO [3], PO [1, 3])

UNIT –V

8 a. Explain in brief, various issues in code generation phase. **10 M**

(Level [2], CO [4], PO [1])

b. Explain the following with suitable examples: **10 M**

Basic Blocks
Flow graphs

(Level [2], CO [4], PO [1])