Nirma University

Institute of Technology

Semester End Examination (RPR), June-2022

B.Tech. in Computer Science & Engineering, Semester -VII
2CS701 – Compiler Construction

Roll / Exam No.		Supervisor's initial with date	

Time: 3 Hours

Max. Marks : 100

Instructions:
1. Attempt all questions.
2. Figures to right indicate full marks.
3. Use section-wise separate answer book.
4. Draw neat sketches wherever necessary.

SECTION – I

Q.1 **Do as directed** [18]

(A) Demonstrate step by step compiler processing for following (6)
CO1BL3 statement through all phases of compiler:
Interest = (Principal * Rate * Term) / 100

(B) Is following grammar SLR? Justify. (6)
CO2BL5
$S \rightarrow L - R$
$S \rightarrow R$
$L \rightarrow *R$
$L \rightarrow id;$
$R \rightarrow L$

(C) Check whether the given grammar is LL(1) or not. Prove your (6)
CO2BL5 answer with proper parsing table and explanation. (ε is NULL)
$S \rightarrow iEtSS' | a$
$S' \rightarrow eS | \varepsilon$
$E \rightarrow b$

Q.2 **Do as directed.** [16]

(A) Define terms Activation Tree and Activation Record in static (6)
CO1BL1 allocation of space with example.

(B) Design the Recursive Descent Parser for the following grammar. (6)
CO2BL6
$E \rightarrow TE'$
$E' \rightarrow +TE'$
$T \rightarrow FT'$
$T' \rightarrow *FT' | \varepsilon$
$F \rightarrow (E) | id$

OR

(B) Convert the following C code into 3 address code: (6)
CO2BL6 *while (x<y) {*
x:=x+1;
if (x%2==1) then y:=y+1;
else y:=y-2;
}

(C) Explain the following issues for the design of a code generator. (4)
CO1BL1 1. Register allocation
2. Instruction Selection

Q.3 Do as directed. [16]

(A) "For any Top down parsing, left recursion removal and left factoring (6)
CO1BL2 are important". Write your opinion about the statement and justify
with suitable example.

(B) What is annotated tree? Explain it by designing an annotated tree (6)
CO2BL2 for the string "5 + 3 * 4" with the following grammar rules.

Production	Semantic Rules
L → E **return**	Print ("answer is" + E.val)
E → E$_1$ + T	E.val = E$_1$.val + T.val
E → T	E.val = T.val
T → T$_1$ * F	T.val = T$_1$.val + F.val
T → F	T.val = F.val
F → (E)	F.val = E.val
F → Num	F.val = (int) Num.lexval

OR

(B) Design a LL(1) Parse table for the following grammar. Explain the (6)
CO2BL3 error recovery strategy for the input string "aab".
S → AbS | e | ε   A → a | cAd

(C) Construct the minimized DFA for the transition table of DFA given (4)
CO2BL6 below.

States	a	B
1	2	6
2	1	3
3	2	4
4	4	2
5	4	5
6	5	4

Accepting state = {3,6}

Starting State = {1}

Section II

Q.4 Do as directed. [18]

(A) Compare various representations of three address statement with [6]
CO1BL4 suitable example.

(B) Construct the DAG for the given below expression [6]
CO2BL3 ((x + y) - ((x + y) / (x − y))) + ((x + y) * (x − y))

(C) What is dead code in code optimization? Eliminate the dead code [6]
CO4BL3 from the following code fragment, if exists:
int x;
void insert ()
{ int i;
i=1;
x=1;
x=2;
return;
x=3;}

Q.5   Do as directed.   [16]

(A)   Construct operator precedence function table from below given   [6]
CO2BL3   operator precedence relation table.

	id	+	*	$
id		·>	·>	·>
+	<·	·>	<·	·>
*	<·	·>	·>	·>
$	<·	<·	<·	·>

(B)   Trace LR Parsing for input string "baaab" using below given LR   [6]
CO2BL3   parse table. Does input string match the Grammar? Why?
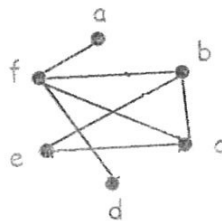
LR Parse Table

State	Action a	b	$	Goto S	T	Production Rules
0	s3	s4		1	2	1. S → TT
1			acc			2. T → aT
2	s3	s4	a		5	3. T → b
3	s3	s4			6	
4	r3	r3	r3			
5			r1			
6	r2	r2	r2			

OR

(B)   Perform optimal registers allocation using graph coloring method   [6]
CO4BL3   for below given register interference graph

(C)   Differentiate Synthesized and Inherited attributes with suitable   [4]
CO1BL4   example.

Q.6   Do as directed.   [16]

(A)   What do you mean leader in basic blocks? Write all properties of   [6]
CO1BL1   leader.

(B)   Write translation scheme to calculate decimal value from BCD code.   [6]
CO2BL3   For example, value of 1001 0111 0101 is 975.

OR

(B)   Generate Syntax Directed Definition to translate infix expression to   [6]
CO2BL3   postfix expression

(C)   Apply left recursion elimination to following grammar: [4]
CO2BL4   S → Aa | b
           A → Ac | sd | ∈