The LNM Institute of Information Technology

Department: Computer Science and Engineering

Compiler Design (CSE 0437)

Examination Type: End Term

Time: 180 minutes **Date: May 4, 2019** **Max. Marks:** 180

Instruction: All the questions are compulsory. Marks are typically proportional to the time to answer. Make and State Assumptions if and when required.

Case Study # 1: Consider the following subset of a 'C like' language: it has only declarative (only of int and float types), read, write, assignment (over arithmetic expressions involving binary operators), if-then (with boolean expressions over "and / or"), goto and stop statements; assume that the below mentioned program (to calculate the surface area and volume of a sphere) is a syntactically valid program of the above subset ('tpo' stands for 'to-the-power-of'): [Total Marks = 90]

Valid_Program	Intermediate Code
main {	
float r ; float h ; float a ;	
10: read (r) ;	
if (r le 0)	
then goto 20 ;	
a := 4 * 22 / 7 * r tpo 2 ;	
v := 4 / 3 * 22 / 7 * r tpo 3 ;	
write (a) ;	
write (v) ;	
20: stop ;	
}	

[Q1]. Design an unambiguous CFG that accepts programs of the above subset. [Marks 20]

[Q2]. Design a SDD of the above CFG to generate max 3-address based Intermediate Code (IC) for all the valid programs of the CFG. [Marks 20]

[Q3]. For the above mentioned valid program, generate the Intermediate Code; do not make any changes to the given source program. [Marks 20]

[Q4]. Identify five Semantic Analysis cases and do the corrections to the above IC. [Marks 15]

[Q5]. Identify five Intermediate Code Optimization cases and do the corrections to the IC after Q4. [Marks 15]

Case Study # 2: Consider the following unambiguous CFG:

CFG = {NT, t, P, B} wherein B is the Start Symbol in NT; NT = {B, R, V, RO, C}; set of terminal symbols, t == {a, o, id, l, e, n} and the set of productions, P, is as given in the table below: [Total Marks = 90].

P1	B → B a R	P2	B → B o R	P3	B → R	P4	R → V RO C
P5	V → id	P6	RO → l	P7	RO → e	P8	C → n

(Note: P1, P2, . . . P8 are the numbers of the productions; l is lower-case alphabet of L)

[Q6]. Generate the canonical set of SLR (1) states for the above CFG of Case Study # 2. [Marks 20].

[Q7]. Compute and show the FOLLOW set for each production (P1 to P8). [Marks 16].

[Q8]. Generate the corresponding SLR (1) Parsing-Table. [Marks 20].

[Q9]. Consider the following valid sentence of the above CFG and show the SLR (1) Parsing-Stack. [Marks 20]: id l n o id e n

[Q10]. Show the resulting Parse-Tree of the above valid sentence as a result of SLR(1) parsing. [Marks 14].