BCSE 3rd year 2nd semester (supple) Examination – 2023

Subject: **Compiler Design**

Time : Three hours **Full Marks: 100**

Different parts of the same question should be answered together.

Group-1 (20 marks)	Answer the following questions: (10+10) 1. With examples explain the differences between regular languages, context free languages and context sensitive languages. With reference to the different phases of compilers, discuss which types of languages are used in compiler design. 2. Divide the following C++ program into appropriate lexemes, associate suitable tokens with the lexemes. How do you store them in the symbol table? "float limitedSquare(x) float x; { */*returns x-squared, but never more than 100*/* return (x <= - 10.0 || x >= 10.0) ? 100 : x * x ; }"
Group-2 (40 marks)	Answer Question no. 3 and any two from this group. (10+11+!1) 3. (a) While constructing a parsing table, why do you need to construct the FIRST and FOLLOW sets? Explain your answer with examples. OR With an example explain how do you remove indirect left recursion in a grammar. (c) Discuss how the lookahead is used in LR(1) parsing (Definition is not required. Explain with an example). OR What is a viable prefix? Explain with an example. 5+5=10 4. Consider the grammar: $s \rightarrow sy\ st \mid T\ T\ st \mid list\ st$ $list \rightarrow list\ R \mid \varepsilon$ $sy \rightarrow T$ $st \rightarrow A$ *(s is the start symbol; sy, st, and list are non-terminals; T, R and A are terminals)* (a) Is the above grammar LL(1)? Why or why not? If it is not LL(1), then convert it to LL(1) (b) Construct the LL(1) parsing table for the above grammar. 15 5. (a) Construct the LR(1) item set for the following grammar and draw the corresponding DFA. $p \rightarrow b$ $b \rightarrow B\ sl\ E$ $sl \rightarrow sl\ ;\ s \mid s$ $s \rightarrow b \mid v = e$ $v \rightarrow ID \mid ID\ e$ $e \rightarrow e + t \mid t$ $t \rightarrow v \mid v\ (\ e\ )$ (p is the start symbol; b, sl, s, v, e, t are non-terminals; B, E, ID, `;', `=', `+', `(' and `)' are terminals.) (meanings are given here: p is program, b is block, sl is statement list, s is statement, v is variable, e is expression, t is term, B means `begin', E means `end', ID is identifier, `;', `=', `+', `(' and `)' are operators and punctuations.) 15

	6. (a) Consider the following grammar: *S --> AB* *S --> BA* *A --> aaB* *A --> a* *B --> bbA* *B --> a* (Terminals = *{a, b}*, Non-terminals = *{S, A, B}*, Start Symbol = *S*) Construct LR(0) item set for the above grammar. (c) Construct the SLR parsing table for the above grammar. (d) Show the actions of the parser for the input string: (aabba). 15
Group-3 (24 marks)	Answer any two questions from this group. 7. Explain the terms and give an example of each. *inherited attribute, L-attributed definition, static scoping.* *12* 8. The following grammar generates binary numbers: $S \rightarrow L$ $L \rightarrow L\ B \mid B$ $B \rightarrow 0 \mid 1$ Design an S-attributed definition SDD to compute S.val, the decimal-number value of an input string. Write a translation scheme for the above SDD. Discuss how you evaluate SDD using bottom-up parsers? 12 9. (a) What is the use of symbol table? In which phases of compiler a symbol table is used? (b) What are the operations performed on a symbol table when it is implemented as a single hash table? (c) What is a three-address code? Write a three address code for the following expression: a[i] = b * c + b * d 12

Group-4 (16 marks)	Answer the following questions: (8+8)
	
	10. Optimize the following code and discuss each optimization technique that you have applied stating their advantages:
	
	`for (i = 0, i<n; i++)`
	`{`
	`  j = c * 3.14;`
	`  if (i%2)`
	`  {`
	`  x += (4*j + 5*i);`
	`  y += (7 + 4*j);`
	`  }`
	`}`
	
	11. (a) With an example explain what are 'liveness' and 'next use' of variables. Why do you need this information?
	(c) Discuss the use of 'Register Descriptor' and 'Address Descriptor'.