B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

July 2024 Semester End Main Examinations

Programme: B.E.   **Semester: V**
Branch: Computer Science and Engineering   **Duration: 3 hrs.**
Course Code: 22CS5PCCPD   **Max Marks: 100**
Course: Compiler Design

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
                2. Missing data, if any, may be suitably assumed.

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

				CO	PO	Marks
			UNIT - I			
1	a)		Analyze how the input *sum= a+b*30* is processed by the compiler through the various phases.	*CO2*	*PO2*	**10**
	b)		Design a transition diagram for relational operators such as < , <=, > , >=,== ,!= . Emulate the same with a simple pseudocode.	*CO1*	*PO1*	**8**
	c)		Identify and list the lexemes in the following statement *printf("Age = %d \n",age);*	*CO1*	*PO1*	**2**
			UNIT - II			
2	a)		Give the algorithm for finding FIRST and FOLLOW Set	*CO1*	*PO1*	**6**
	b)		Construct LR(0) Item set , SLR Parsing table for the following grammar. Validate with appropriate reasoning if the given grammar is SLR or not $S \rightarrow L=R \mid R \quad L \rightarrow *R \mid id \quad R \rightarrow L$	*CO2*	*PO2*	**8**
	c)		Verify if the following grammar is ambiguous? $E \rightarrow E+E \mid E-E \mid E*E \mid (E) \mid a \mid b$ If ambiguous convert the same to unambiguous grammar with following associativity and precedence ( ) having highest priority + having second highest priority and right associative - having third highest priority and right associative * having least priority and left associative	*CO2*	*PO2*	**6**
			OR			
3	a)		Find FIRST and FOLLOW set of all variables given below $S \rightarrow ACB \mid CbB \mid Ba \quad A \rightarrow da\mid BC$ $B \rightarrow g \mid \varepsilon \quad C \rightarrow h \mid \varepsilon$	*CO1*	*PO1*	**4**
	b)		Analyze the following grammar, $S \rightarrow CC \quad C \rightarrow cC \quad C \rightarrow d$ i) Obtain LR(1) item set ii) Design CLR parsing table iii) Justify if the grammar is suitable for CLR or not iv) Show the sequence of moves made by the parser for parsing the input string "ccdd"	*CO2*	*PO2*	**10**

		c)	Analyze if the following grammar is suitable for Predictive parsing. With appropriate justification E → 5+T | 3-T T→ V | V*V | V+V V→ a | b	*CO2* *PO2*	**6**
			UNIT - III		
4	a)		Write SDD for a desk calculator having operations +, * for a top down parser. Show the dependency graph for the input 1+2*3. Give any 2 topological sorting order.	*CO1* *PO1*	**12**
	b)		Write the SDT for int [2][3]. Show the annotated parse tree.	*CO1* *PO1*	**8**
			UNIT - IV		

5	a)		Analyze the following code snippet and give its equivalent three address code
			
			  while(a<c and b>d)
			{
			 if a=1
			   then c=c+1
			 else
			    while (a <= d)
			     a=a+b
			}
				CO3
				
				PO2
					5

	b)		Give DAG for the expression (a+b)*(a+b+c)	*CO3* *PO2*	**3**
	c)		Outline with a simple diagram and with appropriate SDD for flow control statement such as if,if-else ,while	*CO1* *PO1*	**12**
			OR		
6	a)		What is meant by three address code? Construct DAG and equivalent three address code for ((x+y)-((x+y)*(x-y))) + ((x+y)* (x-y)) .	*CO3* *PO1*	**6**
	b)		Give SDT for various Boolean expressions	*CO1* *PO1*	**8**
	c)		Show the three address code in various representation such as quadruples, triples for the expression *-(a+b)*(c+d)+(a+b+c)*	*CO3* *PO1*	**6**
			UNIT - V		
7	a)		List issues in design of code generator	*CO1* *PO1*	**5**
	b)		Convert the following 3 address code to assembly target instruction i)*p=y ii)a[j]=c	*CO3* *PO1*	**5**

	c)		Give intermediate code, basic block and flow graph for the code snippet given below (assume i and j are int of size 4 bytes)
			
			      for i from 1 to 10 do
			      for j from 1 to 10 do
			       a[i,j]=0.0;
			      for i from 1 to 10 do
			       a[i,i] = 1.0;
				CO3
				
				PO2
					10

******