14CS73 USN 1 N T 1 4 C C 0 9 1

Seventh Semester B.E. Degree Examination, December 2017
Computer Science and Engineering
Compiler Design (14CS73)

Max. Marks:100

Instructions: 1. Answer one full question from each unit.
2. Any missing Data can be suitably assumed.

				CO; BL
		UNIT-I		
1.	**a.**	Illustrate the compiler operations over the following statement step by step. K=(a+b)*(a+b)-f	08 Marks	1;3
	b.	Appraise about the recognition of C tokens using regular definitions. Do it for any 4 different categories of C tokens.	08 Marks	1;2
	c.	How will the compiler identify and overcome the errors in the Source code?	04 Marks	1;2
2.	**a.**	Consider the conditional operator is used in a statement to find out biggest of 2 numbers. Let the statement is stored across the input buffers. Examine the operations of lexical analyzer using the input buffers.	05 Marks	1;4
	b.	How will the compiler convert source code to machine code? Illustrate it.	05 Marks	1;2
	c.	Provide transition diagrams for the following: i) Arithmetic operators (any 2) ii) Relational operators (any 2) iii) Logical operators (any 2) iv) Keywords (any 2) v) Variables	10 Marks	1;2
		UNIT-II		
3.	**a.**	*Construct parsing table for the LL(1) grammar given below, finding the first and follow sets. (note: € denotes epsilon)* E→ TE' E'→ +TE'|€ T T→ FT' T'→*FT'|€ F→ (E) |id	10 Marks	2;6
	b.	Solve the problem of shift reduce parsing for the input string id*id+id with the grammar, E→ E+T|T T→ T*F|F F E→ (E) |id	04 Marks	2;3
	c.	Construct LALR parsing table for the grammar, S→CC C→ cC |d	06 Marks	2;6
4.	**a.**	Compute sets of LR(0) items and construct the automation and SLR parsing table for the grammar, **E→ E+T |T** **T→ T*F |F** **F→ (E) |id**	10 Marks	2;6
	b.	Describe YACC tool and its applications.	04 Marks	2;2
	c.	In the grammar, S→ iSeS | iS | a , Examine the states in which dangling else ambiguity exists, by identifying the sets of LR(0) items/parsing table.	06 Marks	2;4

Possession of any kind of written material, mobile/ electronics gadgets & scribbling on QP, amounts to Malpractice

USN □□□□□□□□□□

14CS73

UNIT-III

5. a. Obtain the DAG for the expression. 08 Marks 3;3

a+b +(a+b)+(a+b)*a

Translate the expression into three address code, quadruple and triple representations.

b. Design a SDD for computing basic arithmetic operations and array types. Also construct annotated parse tree for 08 Marks 3;3

int a [2] [2].

c. Explain the procedure of generating intermediate code for procedures. 04 Marks 3;2

6. a. Generate the semantic rules for the following productions using back patching. 11 Marks 3;3

P→ S
S→ if (B) MS
B→ B_1 | | M B_2 | E_1 rel E_2
E→ id | digit

Show the steps in back patch process for generating intermediate code for

If (x<100 | | X!=y) x=1;

b. Translate the following expressions into three address code. Construct DAG for each expression. 09 Marks 3;3

i) a*(b+-c) + (b+-c)

ii) c + b [i] + d[j]

iii) a+a+(a+a+a+(a+a+a+a))

UNIT-IV

7. a. Construct assembly language code for the following three address statements, assuming all variables are stored in memory locations. 07 Marks 4;6

a) X=1

b) X=a

c) X=a+1

d) X=a+b

e) X=b*c

b. *Generate code for the following three address statements assuming stack allocation where register SP points to the top of the stack.* 07 Marks 3;6

Call p
Call q
return
Call r
return
return

c. How can you determine the liveness and next use information for each statement in a basic block? 06 Marks 4;4

8. a. Organize the flow graph of the following code segment:

i) For loop to find out the sum & minimum element of n array elements of integers. Assume Integer occupies 2 bytes. 10Marks 4;3

b. How will be an optimal target program generated by code generator? 06 Marks 4;2

c. Differentiate between Address Descriptor and Register Descriptor 04 Marks 4;4

UNIT-V

9. a. Describe green compilation and related tools. 08 Marks 5;2

b. Explain live variable analysis and its use. 05 Marks 5;2

c. Describe region based analysis and justify its usefulness for the data flow problems. 07 Marks 5;5

10. a. With specific examples, examine the way optimization can be achieved with constant propagation and partial redundancy elimination. 06 Marks 5;4

b. Discuss various sources of Optimization. 08 Marks 5;2

c. Explain power optimization with Compilers. 06 Marks 5;2