QPC: RD20BTECH359 AR 19 Reg. No □□□□□□□□□□

GIET UNIVERSITY, GUNUPUR – 765022

B. Tech (Fifth Semester) Examinations, December – 2022

BPCCS5030 / BPCCT5030– Compiler Design

(CSE and CST)

Time: 3 hrs Maximum: 70 Marks

Answer ALL Questions

The figures in the right hand margin indicate marks.

PART – A: (Multiple Choice Questions) **(1 x 10 = 10 Marks)**

Q.1. Answer ***ALL*** questions CO # PO #

a. Outputs of lexical analysis phase is 1 1

i. Keywords ii. Symbols

iii. Tokens iv. All of the other options

b. The regular expression (0|1)*(0|1)(0|1) represents a language with 2 2

i. Nonempty binary strings ii. Empty and nonempty binary strings

iii. Odd nonempty strings iv. Even nonempty strings

c. For a context-free grammar, left-hand side of production rules should contain 1 1

i. Single nonterminal ii. At most three grammar symbols

iii. at most two grammar symbols iv. None of the other options

d. In shift-reduce parsing, handle is at 2 2

i. Bottom of the stack ii. Anywhere in the stack

iii. Top of the stack iv. Nowhere in the stack

e. For the grammar rule B → abbS | bS, FIRST(B) is equal to 2 3

i. {a} ii. {a, b}

iii. {a, b, S} iv. {S}

f. For the grammar rule B → abbSd | bS, FOLLOW(S) is equal to 2 3

i. {a} ii. {a, b}

iii. {a, b, S} iv. {d, $} 2 1

g. A predictive parser

i. Needs backtracking ii. Does not need backtracking

iii. May not terminate iv. None of the other options

h. Which of the following is NOT possible to realize if activation record is static 4 2

i. Passing parameters ii. Creating local variables

iii. Supporting recursion iv. None of the other options

i. Control link points to the 4 1

i. Current activation record ii. Parent activation record

iii. Child activation record iv. None of the other options

j. Three address code may be represented by 4 2

i. Quadruple ii. Triples

iii. Indirect Triples iv. All of these

PART – B: (Short Answer Questions) **(2 x 10 = 20 Marks)**

Q2. Answer ALL questions CO # PO #

a. What is the front end of a compiler? 2 1

b.	Define Boot Strapping with example.	2	2
c.	Briefly Explain the Concept of Derivation.	1	1
d.	What is a common prefix for grammar? How you can remove the common prefix.	2	1
e.	Define LR(0) items.	3	2
f.	Draw an NFA for (a+b)* abb.	3	1
g.	What is reduce-reduce conflict in LR parser?	4	2
h.	What is dead code? What is the necessity of dead code elimination?	3	3
i.	Define Operator grammar, explain with an example	4	2
j.	Describe Handle pruning, and explain with an example.	3	1

PART – C: (Long Answer Questions) (10 x 4 = 40 Marks)

Answer ***ALL*** questions

		Marks	CO #	PO #
3.a.	Explain the phases of the compilation for the instruction Value = X +Y * 10	5	1	1
b.	Construct the DFA for the following regular expression R=ab(a|b)*	5	2	3
	(OR)			
c.	Test whether the grammar is LL(1) or not and construct a predictive parsing table for it. S → AaAb| BbBa , A → ε , B → ε	6	2	2
d.	Explain Recursive descent parsing.	4	2	3
4.a	Construct the LR(0) parsing table for the following grammar: E → E + T, E → T, T →T*F, T →F, F → (E), F → id And check the string w=id+id*id is generated by the above grammar or not.	10	2	3
	(OR)			
b.	Construct the CLR parsing table for the following grammar: S→ AA, A→aA, A→b And check the string w=aabb is generated by the above grammar or not.	10	2	3
5.a.	Define TAC. Write the TAC for x=A[i][j]+5	5	3	3
b.	Define DAG. Represent the DAG for instruction a + a * (b-c) + (b-c) * d	5	3	2
	(OR)			
c.	Define TAC. write the Quadruple & triple for the following expression x= -(a+b) * (c+d) - (a+b+c)	5	3	3
d.	Define SDD. Write the SDD for S→ if(B) S1 else S2	5	4	2
6.a.	Describe the principle of a source of optimization.	5	4	3
b.	Represent the control flow for If(x<10|| x>20 &&x !=y){ x=0}	5	4	3
	(OR)			
c.	Stepwise describe the code generation for the instruction x=(a+b)-(c(c+d)-e)	5	4	1
d.	Briefly describe the storage organization in the runtime environment.	5	4	3

--- End of Paper ---