SRM Institute of Science and Technology
College of Engineering and Technology
SCHOOL OF COMPUTING

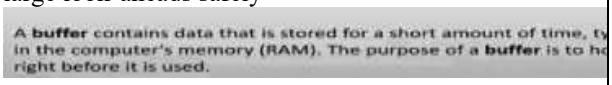
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

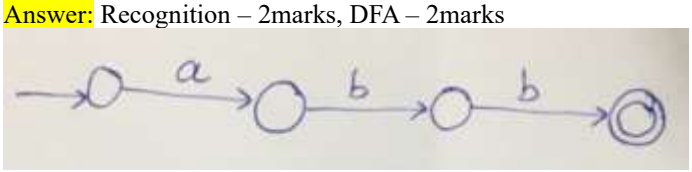
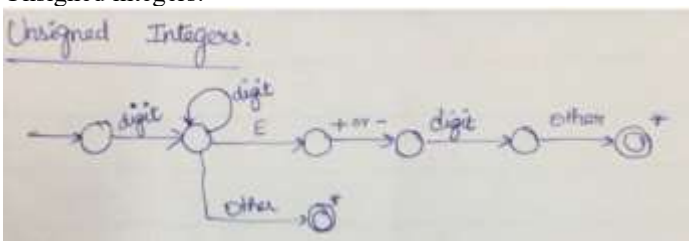
Mode of Exam
OFFLINE

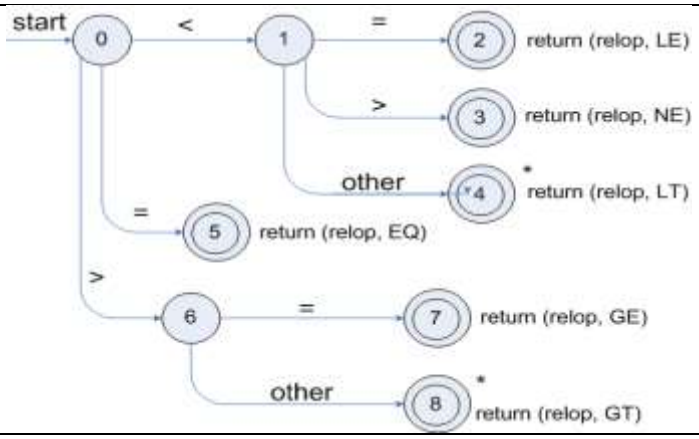
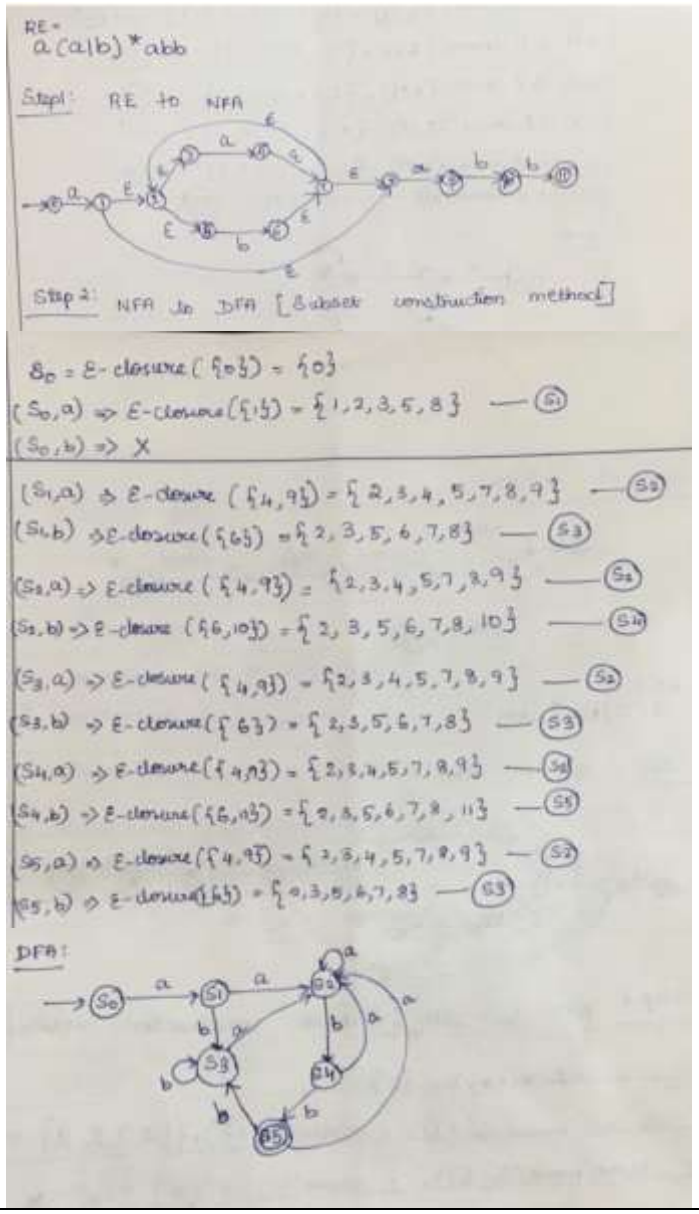
Academic Year: 2022-23 (EVEN)

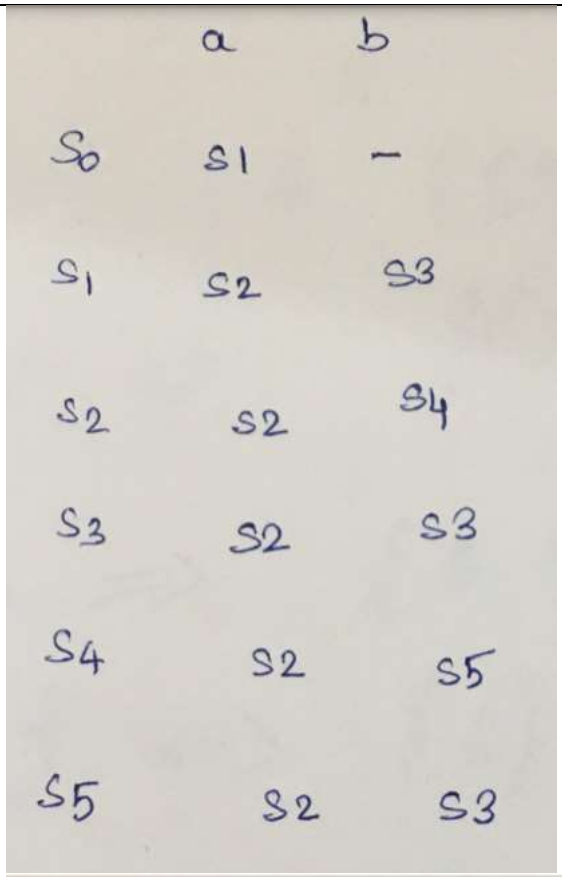
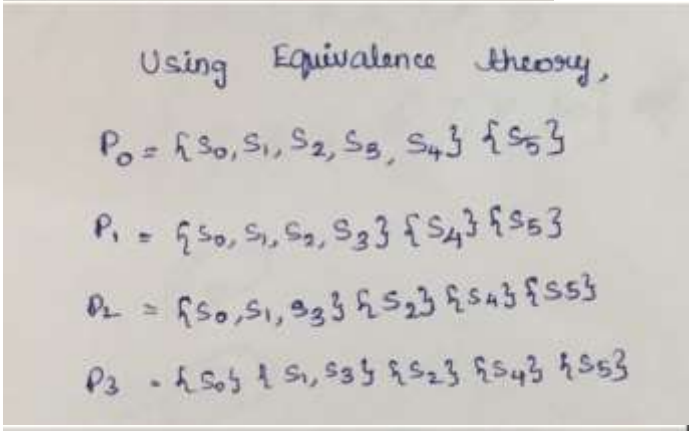
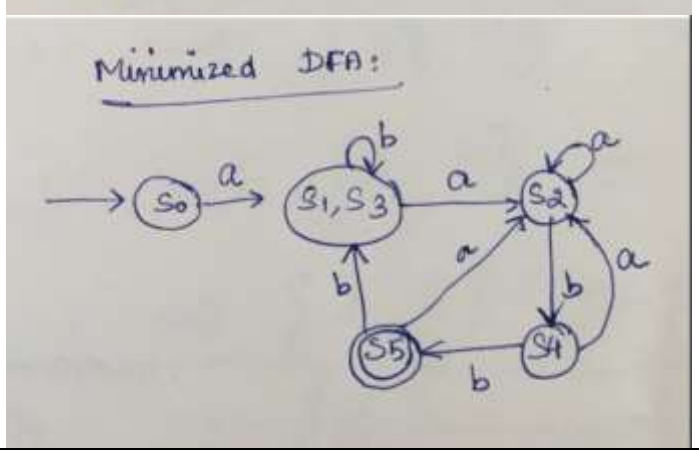
Test: CLAT-1 **Date: 17.2.2022**
Course Code & Title: 18CSC304J COMPILER DESIGN **Duration: 1 HOUR**
Year & Sem: III & V **Max. Marks: 25**

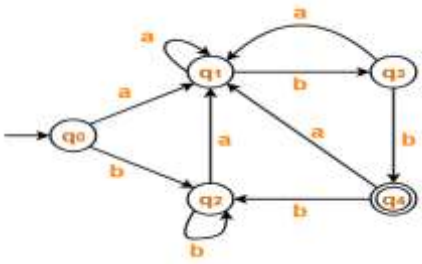
Q. No	Question	Marks	BL	CO	PO	PI Code
	Part – A (5 x 1 = 5 Marks) Instructions: Answer ALL					
1	NFA with ϵ transitions _____ **a) Increases computations** **b) Decreases computations** **c) Decreases number of states** **d) Increases uncertainty** **Ans: a**	1	1	1	1	1.3.1
2	What are the maximum number of tokens generated in the lexical analysis phase for the statement? printf("a = %f, &a = %d, b=%d", a, &a,b); **a) 10** **b) 12** **c) 17** **d) 18** **Ans: b**	1	2	1	1	1.1.2
3	If L,D, S denote the sets of letters, digits and underscore respectively. Then , which can possibly define an identifier? a) **$S(LUD)^+$** b) **(LUS)(LUDUS)*** c) **(LUS)(LUD)*** d) **L(L.D.S)*** **Ans: b**	1	2	1	1	1.1.2
4	The error of missing parenthesis detection occurs in _____ phase. **a) Semantic** **b) Lexical** **c) Syntax** **d) Syntax and lexical** **Ans: c**	1	1	1	1	1.3.1
5	I: DFA's can be constructed for all the languages II: The strings accepted by DFA will be accepted by NFA What can be said about these two statements? **a) Only II is false** **b) Only I is false** **c) I is false and II is true** **d) II is true and I is false** **Ans: c or d**	1	2	1	2	2.1.1
	Part – B (2 x 4 = 8 Marks) Instructions: Answer TWO					
6	Explain the process of input buffering for the given source code.	4	3	1	2	2.1.1

	int i,j;
	i=i+1;
	j=j+1;
	Explain the process with one buffer(size:5) and two buffer (size 5) concepts
	
	Answer: Definition and One buffer scheme with example (2 marks), two buffer scheme – 2 marks
	• Sometimes lexical analyzer needs to look ahead some symbols to decide about the token to return
	  • In C language: we need to look after -, = or < to decide what token to return
	  • In Fortran: DO 5 I = 1.25
	• We need to introduce a two buffer scheme to handle large look-aheads safely
	 
	
	Two pointers – Begin pointer (bp), Forward pointer (fp)					

7	Raju is authoring a book on compiler. He makes sure that the first page is an index page followed by two acknowledgement pages. Design a DFA for the language L=all strings over {a,b}. Note: index page and acknowledgment pages are referred to strings 'a', 'b' respectively. Answer: Recognition – 2marks, DFA – 2marks  	**4**	**2**	**1**	**2**	**1.1.2**
8	Draw the transition diagrams for unsigned integers and relational operators. Answer: unsigned integers – 2marks relational operators – 2marks Unsigned integers:  Relational operators:	**4**	**1**	**1**	**1**	**1.3.1**

						
Part – C (1 x 12 = 12 Marks) Instructions: Answer any ONE						
9	Convert the following RE=a(a|b)*abb to DFA using subset construction method and minimize it. Answer: RE to NFA – 4marks, NFA to DFA – 4marks, minimization of DFA – 4marks 	**12**	**3**	**1**	**2**	**2.1.2**

	  					
		OR				
10	a. Perform minimization technique on the following DFA	**5+7**	**3**	**1**	**2**	**2.1.2**



Answer:

	a	b
→q0	q1	q2
q1	q1	q3
q2	q1	q2
q3	q1	*q4
*q4	q1	q2

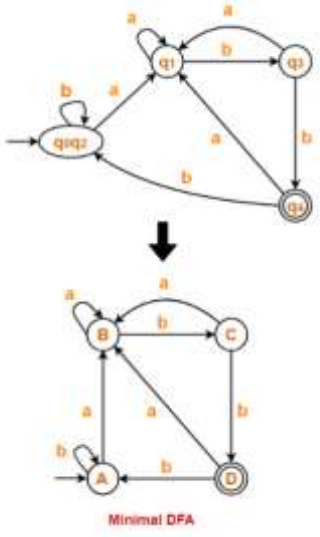
Now using Equivalence Theorem, we have-

$P_0 = \{ q_0 , q_1 , q_2 , q_3 \} \{ q_4 \}$

$P_1 = \{ q_0 , q_1 , q_2 \} \{ q_3 \} \{ q_4 \}$

$P_2 = \{ q_0 , q_2 \} \{ q_1 \} \{ q_3 \} \{ q_4 \}$

$P_3 = \{ q_0 , q_2 \} \{ q_1 \} \{ q_3 \} \{ q_4 \}$



b. Define token, pattern and lexeme with example

Definitions – each 1 mark

A token is a pair a token name and an optional token value

A pattern is a description of the form that the lexemes of a token may take

A lexeme is a sequence of characters in the source program that matches the pattern for a token

Example

Token	Informal description	Sample lexemes
if	Characters i, f	if
else	Characters e, l, s, e	else
comparison	< or > or <= or >= or == or !=	<=, !=
id	Letter followed by letter and digits	pi, score, D2
number	Any numeric constant	3.14159, 0, 6.02e23
literal	Anything but " sorrounded by "	"core dumped"

```
printf("total = %d\n", score);
```