Date:

Name	:	Ibadullah Shaikh
Roll No	:	19k-0257
Section	:	BCS-4G
Course	:	Theory of Automata
Date	:	14-03-2021
Day	:	Sunday

ASSIGNMENT NO # 01

x — x — x — x — x

Question No: 01

$L_1 = (aa)^*(bb)^* + (aa)^* a (bb)^* b$

$L_2 = (aaaa)(a)^* (\lambda + b + bb + bbb)$

$L_3 = (\lambda + a + aa + aaa)(\lambda + b + bb + bbb + bbbb)$

$L_4 = (ab^3 a^* b^*) + (a^2 b^2 a^* b^*) + (a^3 b a^* b^*)$

$L_5 = abbb(b)^* (a+b)(a+b)^*$

$L_6 = aa(a+b)^* aa + ab(a+b)^* ab + ba(a+b)^* ba + bb(a+b)^* bb$

$L_7 = 1^* (01)^* 00 (10)^* 1^*$

$L_8 = b^* a b^*$

$L_9 = b^*(a+\lambda) b^* (a+\lambda) b^* (a+\lambda) b^*$

L_{10} =
↳ $(a+b)^*a\ (a+b)^*b\ (a+b)^* + (a+b)^*b\ (a+b)^*a\ (a+b)^*$

L_{11} = $(0+1)^*\ 01$

L_{12} = $\lambda + 0 + 1 + (0+1)^*\ (00 + 11 + 01)$

L_{13} = $1^* + (1^*01^*0)^*\ 1^*$

L_{14} = $(0+1)^*\ 00\ (0+1)^*\ 00 (0+1)^*$

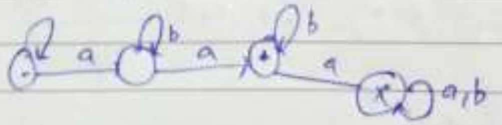
L_{15} = $0^*\ 1^*\ 00^*\ 1^* 0^*$

L_{16} = $[(a+b)(a+b)(a+b)]^*$

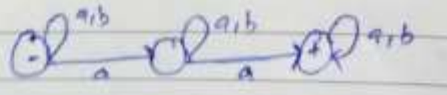
L_{17} = $(aaa+b)^*$

QUESTION No:- 02.

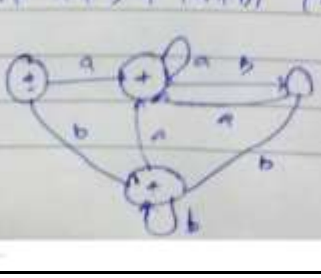
(a) $b^*\ a\ b^*\ a\ b^*$



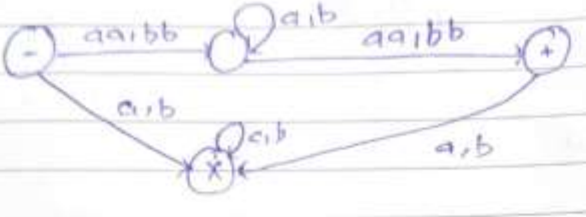
(b) $(a+b)^*a\ (a+b)^*a\ (a+b)^*$



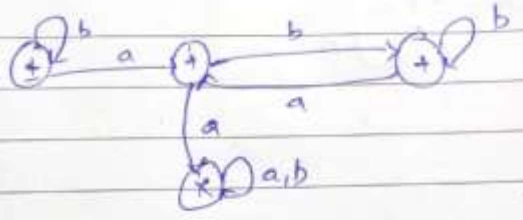
(c) $\lambda + a + b + (a+b)^*\ (aa + ba, bb)$



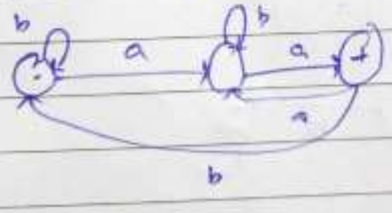
(d) $(aa+bb)^*$ ~~and~~ $(a+b)^*(aa+bb)^*$



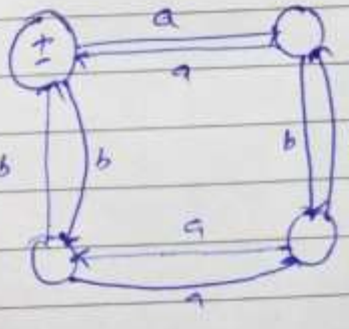
(e) $\lambda + a + b^* + (b^*ab^*)^*$



(f) $b^*(ab^*ab^*)^*$

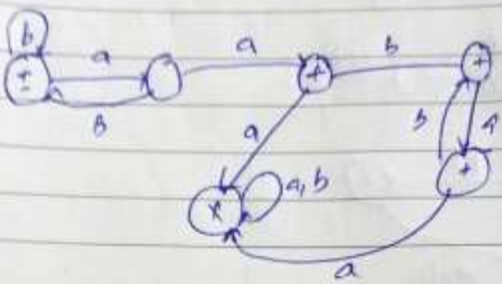


(g) $[aa+bb+(aa+bb)^*(ab+ba)(ab+ba)^*]^*$

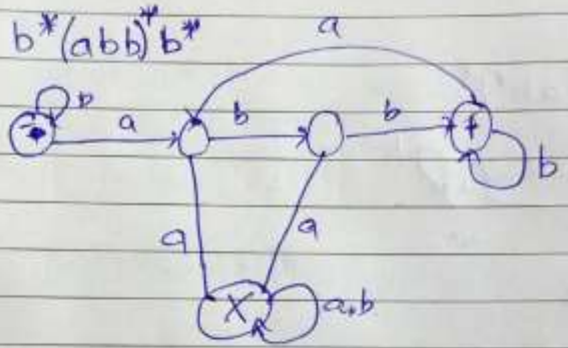


Date:

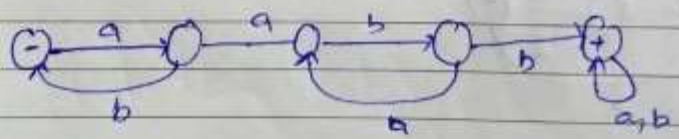
(h) $(ab + b^* aa b^* + ba + bb)^*$



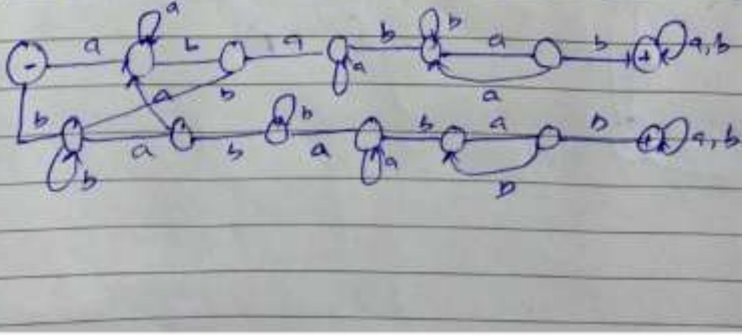
(i) $b^*(abb)^* b^*$



(j) $(a+b)^* \ (aa)^* (a+b)^* (bb)^* (a+b)^*$



(k) $(a+b)^* aba (a+b)^* bab (a+b)^* + (a+b)^* bab \ (a+b)^* aba \ (a+b)^*$



Date:

QUESTION No :- 03

(a) Language: The language of all strings containing ab as substring.

Regular Exp: $(a+b)^* ab (a+b)^*$

(b) Language: The language of all strings containing even a's & even b's in it.

Regular Exp: $[aa + bb + (aa+bb)^* (ab+ba)(ab+ba)^*]^*$

(c) Language: The language of all strings containing (abbaab) as substring.

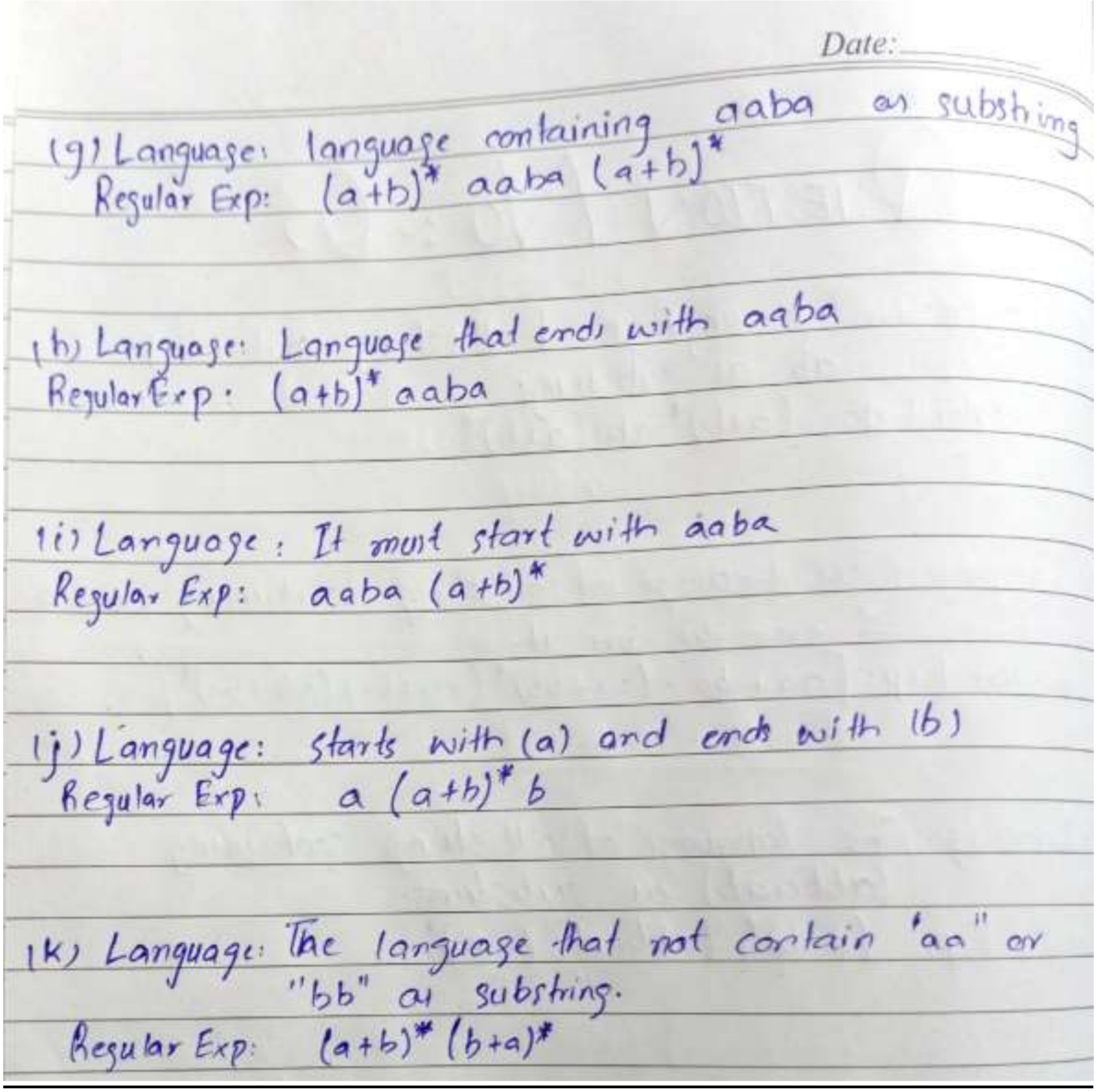
Regular Exp: $(a+b)^* (abbaab) (a+b)^*$

(d) Language: No (aa) comes and ends with b.

Regular Exp: $b^* (ab)^* b^*$

e) Language: The language containing even number of a's ~~and~~ as substring.

) Language: The language containing always b, and if starts with a, then it must ends with 'a' as well.



Date:

(g) Language: language containing aaba as substring
Regular Exp: $(a+b)^*\ aaba\ (a+b)^*$

(h) Language: Language that ends with aaba
Regular Exp: $(a+b)^*\ aaba$

(i) Language: It must start with aaba
Regular Exp: $aaba\ (a+b)^*$

(j) Language: starts with (a) and ends with (b)
Regular Exp: $a\ (a+b)^*\ b$

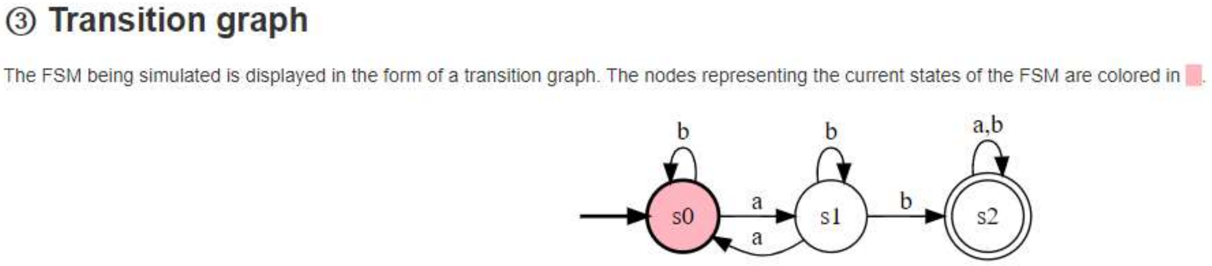
(k) Language: The language that not contain "aa" or "bb" as substring.
Regular Exp: $(a+b)^*\ (b+a)^*$

Q3 (b)

i. Code:

```
#states
s0
s1
s2
#initial
s0
#accepting
s2
#alphabet
a
b
#transitions
s0:b>s0
s0:a>s1
s1:b>s2
s1:a>s0
s1:b>s1
s2:a,b>s2
```

Screenshot:



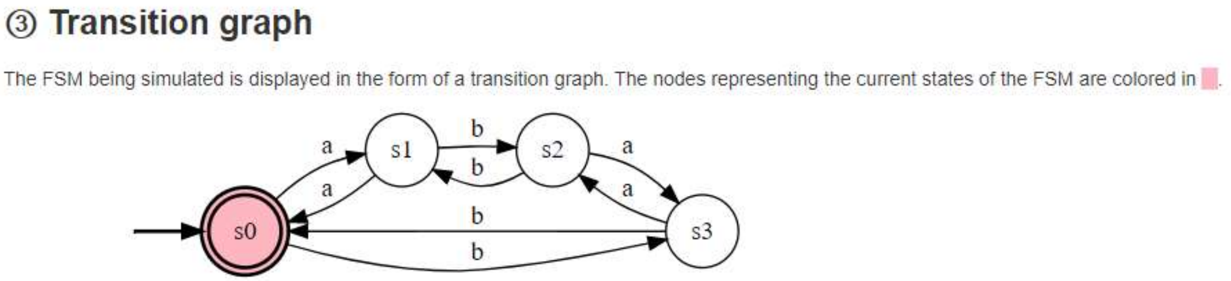
③ Transition graph

The FSM being simulated is displayed in the form of a transition graph. The nodes representing the current states of the FSM are colored in ■.

ii. Code:

```
#states
s0
s1
s2
s3
#initial
s0
#accepting
s0
#alphabet
a
b
#transitions
s0:a>s1
s1:a>s0
s1:b>s2
s2:b>s1
s2:a>s3
s3:a>s2
s3:b>s0
s0:b>s3
```

Screenshot:



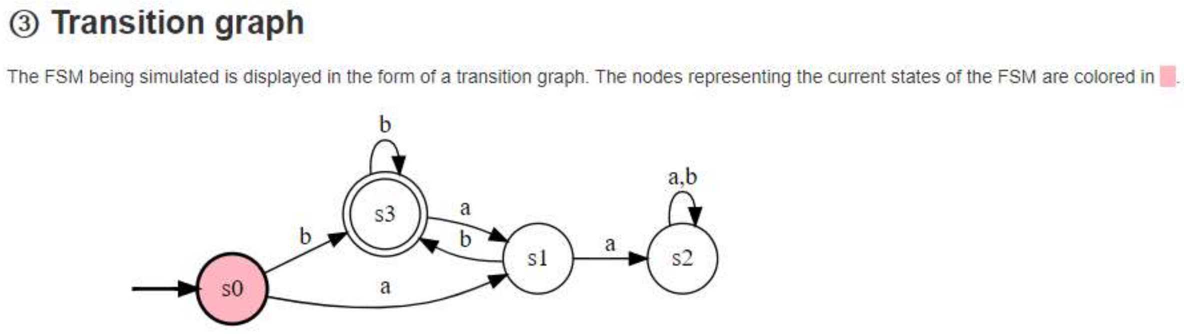
③ Transition graph

The FSM being simulated is displayed in the form of a transition graph. The nodes representing the current states of the FSM are colored in ■.

iii. Code:

```
#states
s0
s1
s2
s3
#initial
s0
#accepting
s3
#alphabet
a
b
#transitions
s0:a>s1
s1:a>s2
s2:a,b>s2
s0:b>s3
s1:b>s3
s3:b>s3
s3:a>s1
```

Screenshot:



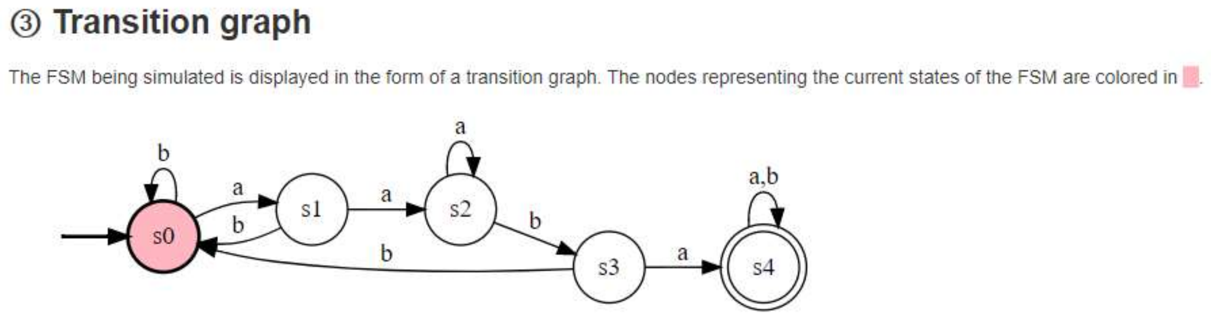
③ Transition graph

The FSM being simulated is displayed in the form of a transition graph. The nodes representing the current states of the FSM are colored in 🟥.

iv. Code:

```
#states
s0
s1
s2
s3
s4
#initial
s0
#accepting
s4
#alphabet
a
b
#transitions
s0:a>s1
s0:b>s0
s1:b>s0
s1:a>s2
s2:a>s2
s2:b>s3
s3:b>s0
s3:a>s4
s4:a,b>s4
```

Screenshot:



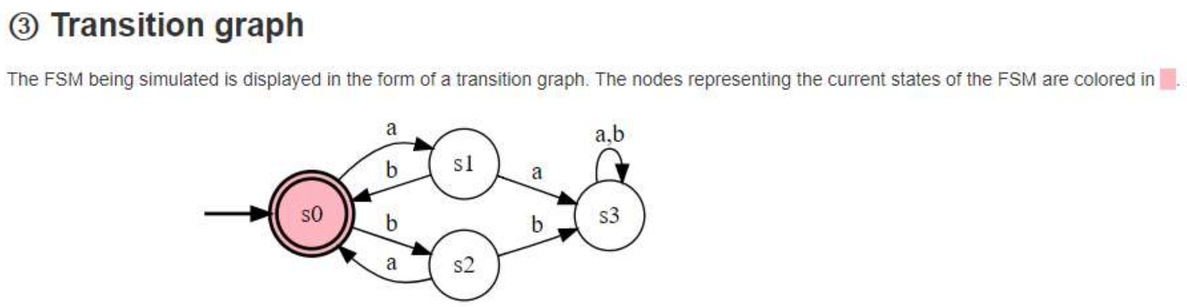
③ Transition graph

The FSM being simulated is displayed in the form of a transition graph. The nodes representing the current states of the FSM are colored in 🟥.

v. Code:

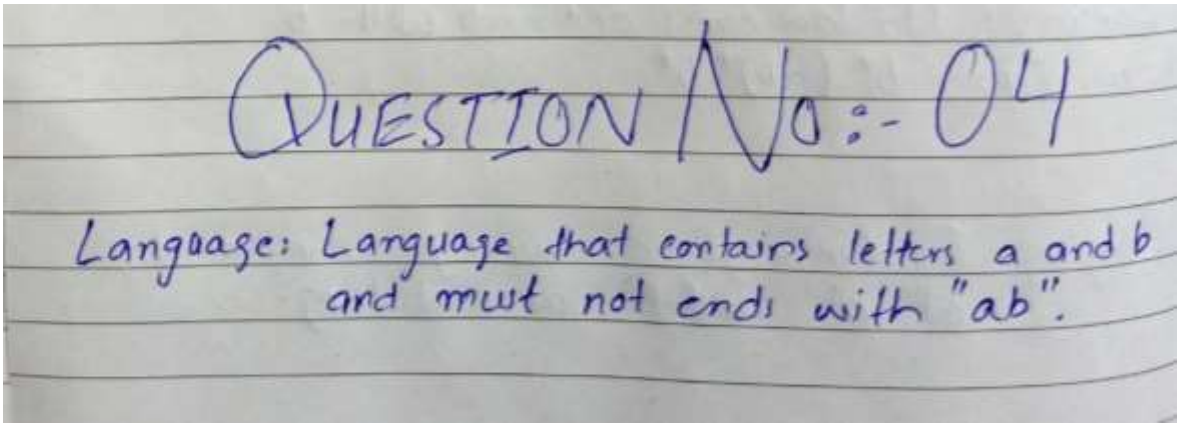
```
#states
s0
s1
s2
s3
#initial
s0
#accepting
s0
#alphabet
a
b
#transitions
s0:a>s1
s0:b>s2
s1:b>s0
s1:a>s3
s2:a>s0
s2:b>s3
s3:a,b>s3
```

Screenshot:



③ Transition graph

The FSM being simulated is displayed in the form of a transition graph. The nodes representing the current states of the FSM are colored in 🟥.



QUESTION No:- 04

Language: Language that contains letters a and b and must not ends with "ab".

Q4 (b):

Ans: Here is the link of Q4, and all the 5 automatas are provided in this link, please find the attached link.

Link: https://colab.research.google.com/drive/1lLqgDRYLpuaOg8BZibzMT5B_YNdoNBXu?usp=sharing

---

Q5:

Ans: As far as my experience is concerned, to be honest, I was not hoping that I will do this question, but I understood the code and then got some errors as well, but I myself solved all those errors and completed the question. By solving this question, I realises that there is much more that we need to learn e.g: Python, syntanxes etc. In the end I personally learned something new and will try to implement it further as well.

Date:

QUESTION No # 06

=> Regular Expressions that should comply with the variable declaration in C Language.

Sol:

Rules:

(i) Cannot start with numbers.

(ii) Cannot start with (-) hyphen

(iii) Cannot start with (.) dot (,) comma.

(iv) Cannot contain (.) dot (,) comma, &, (-), special characters

(v) Can be started with capital Alphabet and small English Alphabet.
eg: (a, b, c, ..., z) or (A, B, C, ..., X, Y, Z)

Regular Expression:

$$\left[(A \ldots z)^{*} + (a \ldots z)^{*} + \underset{\substack{\Downarrow \\ \text{underscore}}}{(\_)^{*}} \right]^{+} (0 \ldots 9)^{*}$$

$$\left[(A \ldots z)^{*} + (a \ldots z)^{*} + (\_)^{*} \right] .$$