Name: _____Demetrius Johnson______2-11-2021_________

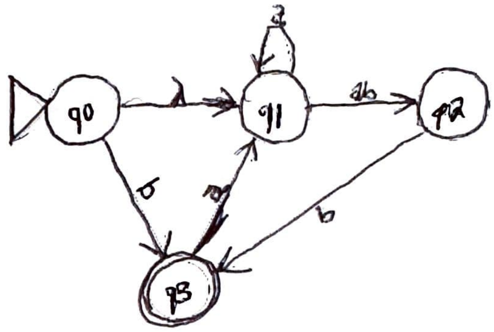
No calculators are allowed. Show your work. The points for each problem are indicated. Problems with incomplete work may receive partial, or no credit.

COs: [No direct COs]

Points: _______________ / 25

1. [10 pts] Convert the following transition graph into a finite automaton by filling in the (final) table and **identifying** the start and accept states. (You do not have to draw the transition diagram for the FA – filling in the final table is enough – **note**: this isn't the intermediary table)

FA State	a	b
▷0	(1, 1.21)	3
(1, 1.21)	(1, 1.21)	2
2	BH	3
3	(1, 1.21)	BH
BH	BH	BH



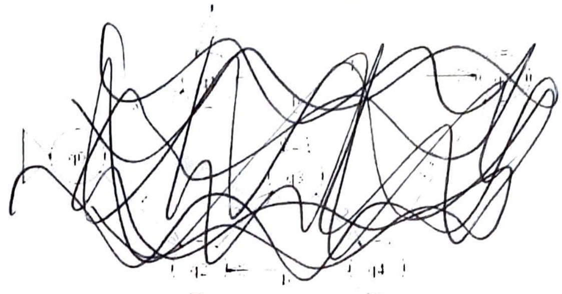
Note: there may be more rows than you need in the table

= accept state

TG state	a	b
▷0	(1, 1.21)	3
~~(1, 1.21)~~	~~(1, ...) crash~~	~~crash 2)~~
1	(1, 1.21)	crash
2	crash	3
3	1	crash

~~1. [10 pts] Convert the following transition graph into the equivalent regular expression~~

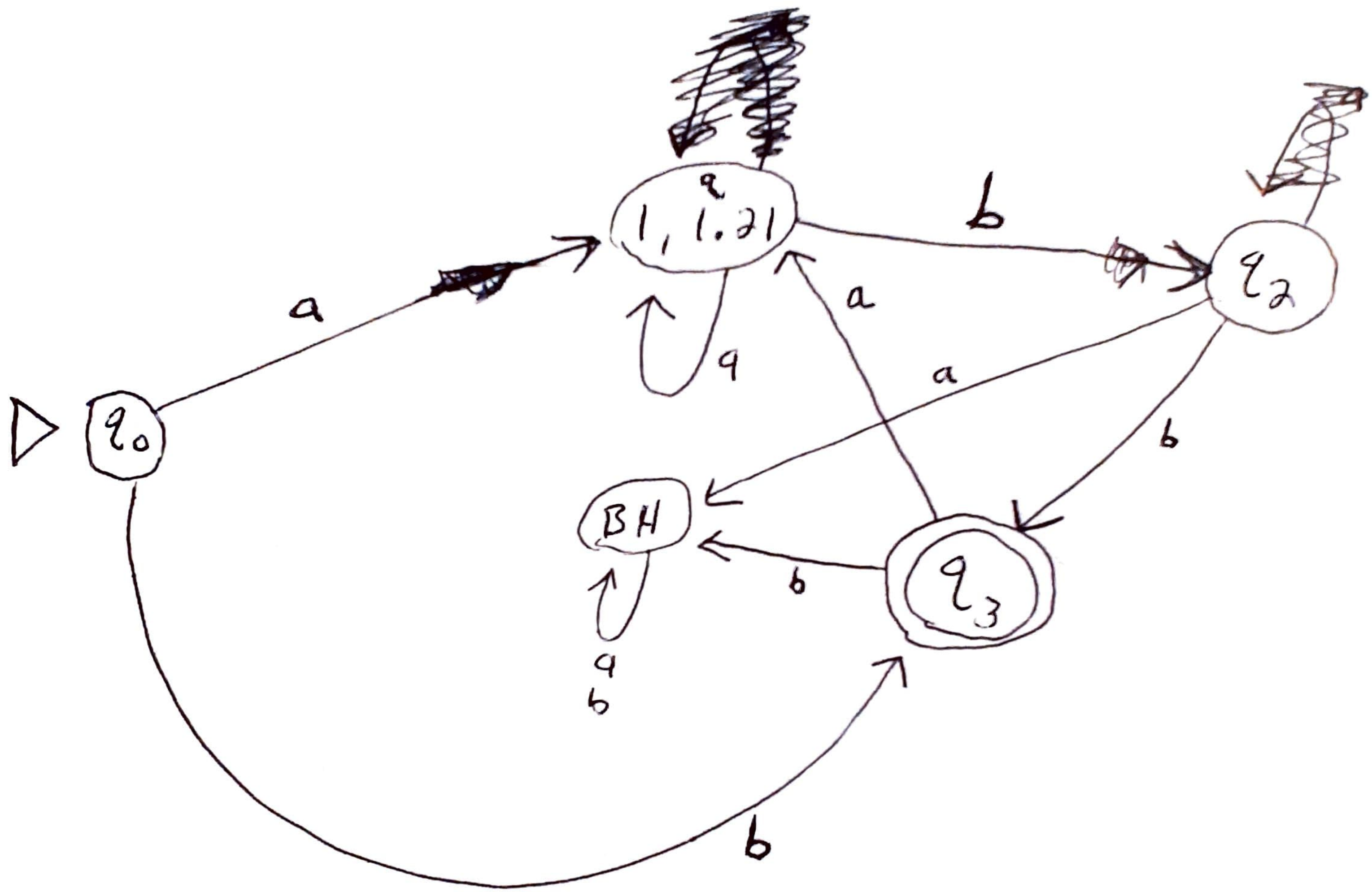
Question 1 work



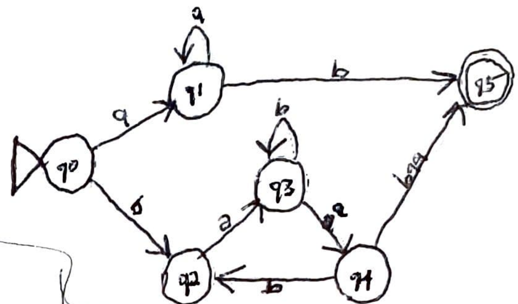
(#) = Accept State

FA state	a	b
DO	(1, 1.21)	3
(1, 1.21)	1, BH	2
2	BH	3
3	1	BH
1	(1.2, 1)	BH

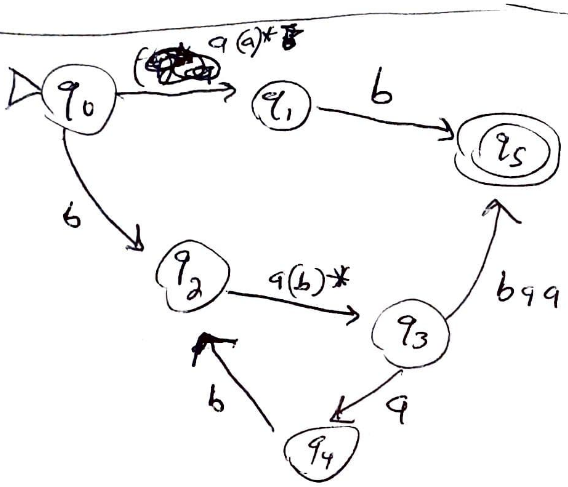
Question 1 Transition Diagram



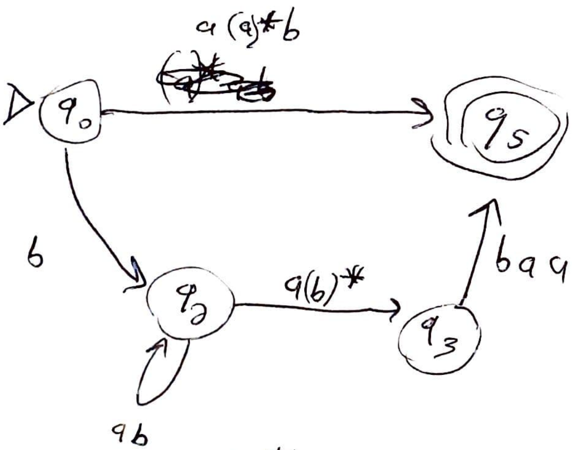
2. [10 pts] Convert the following transition graph into the equivalent regular expression



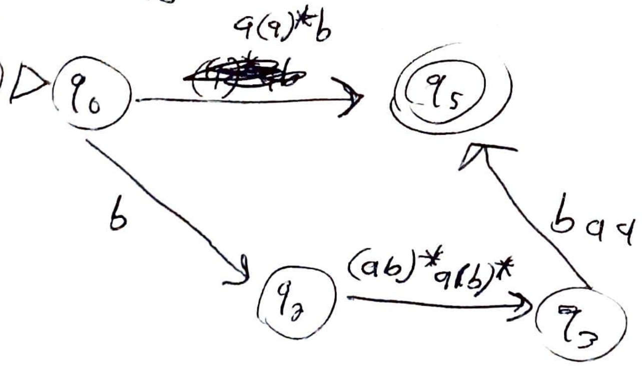
Step 1



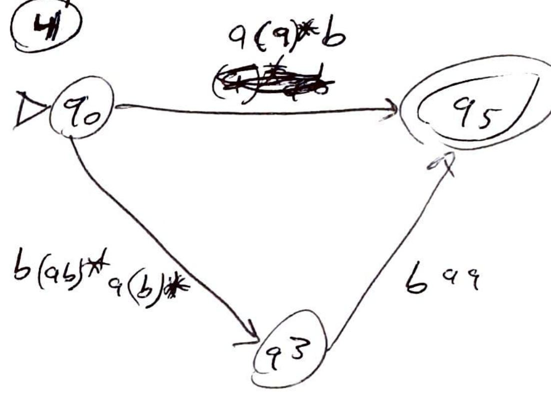
Step 2



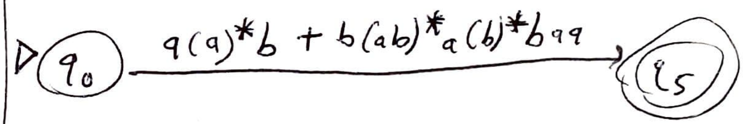
Step 3



Step 4



Final solution:

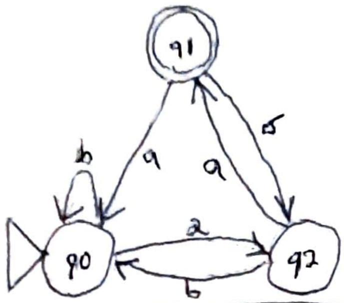


$$\boxed{a(a)^*b + b(ab)^*a(b)^*baa}$$

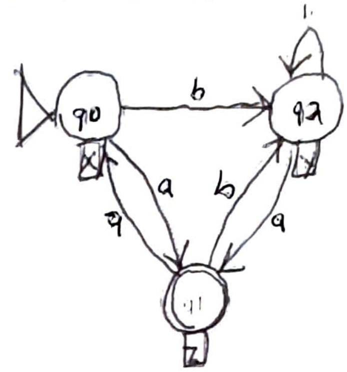
Regular Expression

3. [5 pts] Name the accept states in the machine $L_2' \cup L_1$ (read as: **L_2 prime** union **L_1**)

L_1:



L_2:



Complement

$(L_1 \cap L_2') \cup (L_1' \cap L_2)$

Composite Machine of L_1 and L_2

L_1 state L_2	a	b
(0, x)	(2, Z)	(0, Y)
(2, Z)	(1, X)	(0, Y)
(0, Y)	(2, Z)	(0, Y)
(1, x)	(0, z)	(2, Y)
(0, Z)	(2, x)	(0, Y)
(2, x)	(1, z)	(0, Y)
(1, Z)	(0, x)	(2, Y)
(2, Y)	(0, z)	(1, Y)
(1, Y)	(0, z)	(2, Y)

- $L_2' \cup L_1$

"words that L_2 do not accept union with words that L_1 does accept."

of Comp. Machine

- If both end in Accept, then $L_1 \cap L_2$
- If only 1 ends in Accept, then $L_1 \cup L_2 = \{(2,z), (1,x), (0,z), (1,y)\}$

~~$L_2' \cup L_1 = \{(1,x), (1,y)\}$~~

~~Solution~~

~~L_2 Does Not Accept~~

~~$L_2' \cup L_1 = \{(2,z), (0,z),$~~

$\underline{L_2' \cup L_1} = \{(0,x), (0,y), (1,x), (2,x), (2,y), (1,y), (1,z)\}$

Solution

$\underline{L_2' \cup L_1}$ means "Anywhere in the Composite machine where L_2 does not accept or where L_1 does accept."

(union)