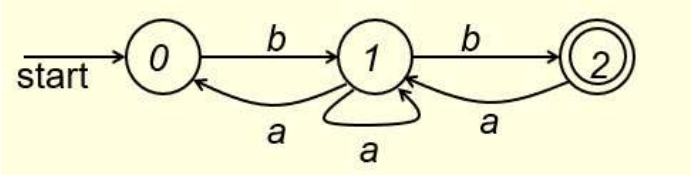
CS375 Homework Assignment 3 Solution Set (40 points)

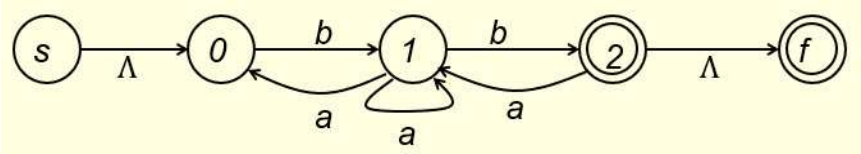
Due Date: February 4, 2025

1. (6 points)

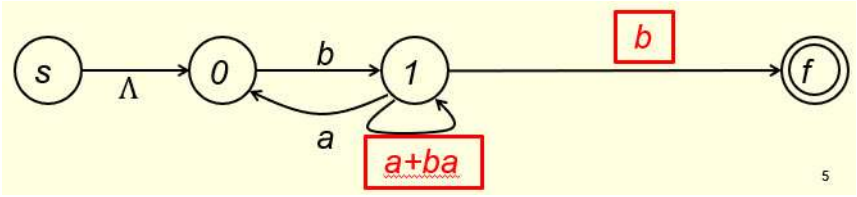
To transform the following NFA to a regular expression,



we first connect a new start state s to the start state of the given NFA and connect the final state of the given NFA to a new final state f as follows.

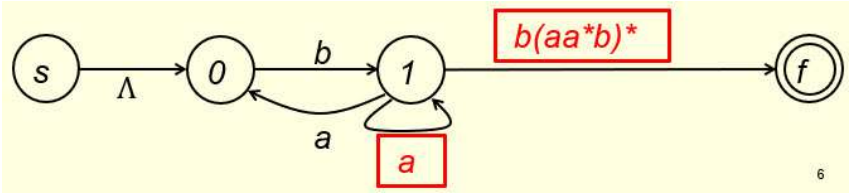


If we eliminate state 2 first, the modified NFA becomes of the following form. Fill out the blanks in the following figure. (2 points)



……. (*)

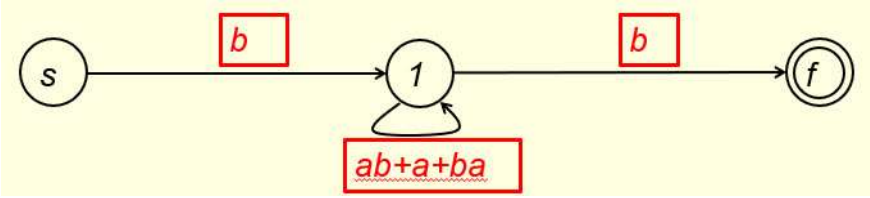
or



…….

(**)

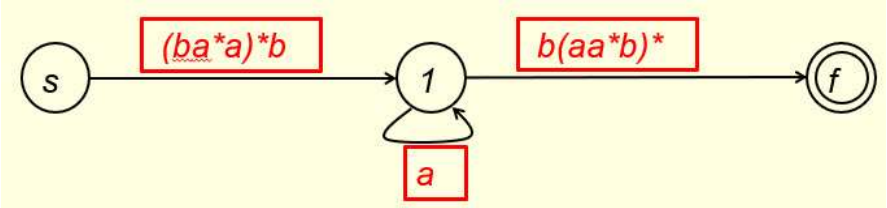
If we eliminate state 0 in (*) then, we get the following NFA. Fill out the blanks below (only the new blanks will be graded). (3 points)



…….

($)

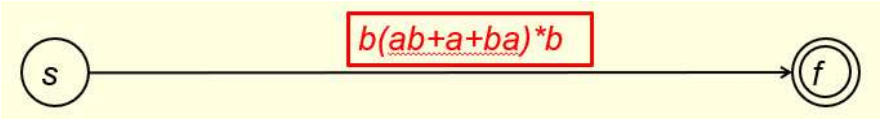
If we eliminate state 0 in (**), we get the following NFA. Fill out the blanks below (only the new blanks will be graded). (3 points)



…..

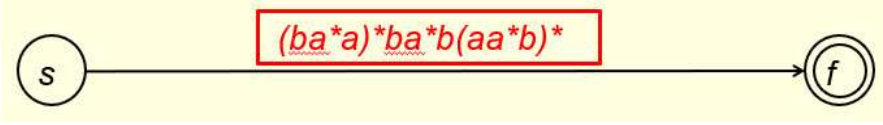
($$)

By eliminating state 1 in ($) we get the following NFA. Fill out the blank below. (1 point).



….. (#)

If we eliminate state 1 in ($$), we get the following NFA. Fill out the blank below. (1 point)

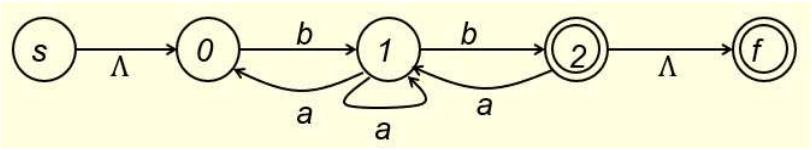


….. (##)

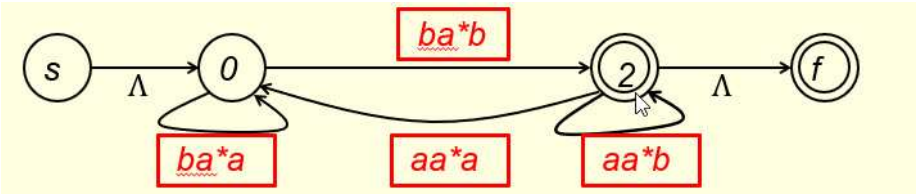
The expressions in (#) and (##) are both the regular expression of the given NFA. You only have to do one case here, either (*), ($) and (#), or (**), ($$) and (##).

2. (14 points)

   In Question #1, instead of eliminating states of the new NFA in the order of 2, 0 and 1,



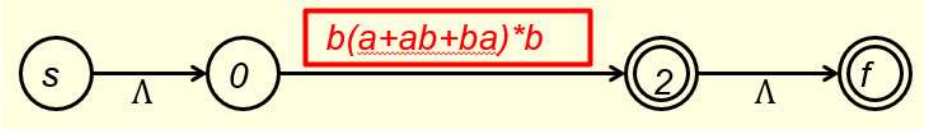
if we eliminate state 1 first, we get the following NFA. Fill out the blanks in the following figure. (8 points)



…..

(*)

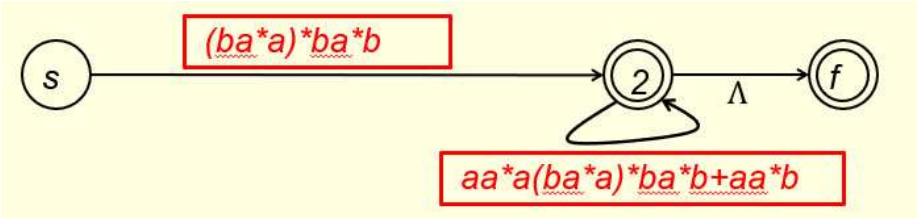
or 12 points)



…..

(**)

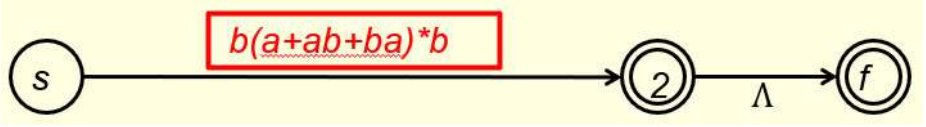
If we eliminate state 0 in (*) then, we get the following NFA. Fill out the blanks in the figure. (4 points)



….

($)

If we eliminate state 0 in (**), we get the following NFA. Fill out the blanks in the following figure. (1 point)



…..

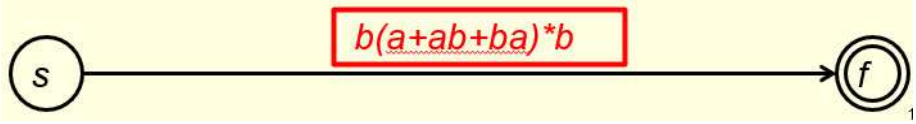
($$)

By eliminating state 2 in ($), we get the following NFA. Fill out the blank in the figure. (2 points)

…..

(#)

If we eliminate state 2 in ($$), we get the following NFA. Fill out the blank in the following figure. (1 point)



…..

(##)

The expression in (#) is also the expression of the given NFA. Note that the expression in (##) is the same as the expression in (#) of Question 1. Again, you only have to do one case here, either (*), ($) and (#), or (**), ($$) and (##).

In the following, I will show that

(ba*a)*ba*b(aa*a(ba*a)*ba*b+aa*b)*=b(a+ab+ba)*b ….. (!!)

I need the following three expression formulas (see slide 48 of the notes “Regular Languages and Finite Automata-I” for these formulas) in the proof:

(RS)*R = R(SR)* (F1)

(R+S)* = R*(SR*)* (F2)

(RS*)* = Λ+ R(R+S)* (F3)

First, I will show that (ba*a)*ba*b=b(a+ab)*b

(ba*a)*ba*b
=ba*(aba*)*b (F1: R=ba*, S=a)
=b(a+ab)*b (F2: R=a, S=ab)

So,

Left side of (!!)
= b(a+ab)*b(aa*a(ba*a)*ba*b+aa*b)*
= b(a+ab)*b((aa*a(ba*a)*ba*+aa*)b)* (Factoring b out)
= b(a+ab)*[b(aa*a(ba*a)*ba*+aa*)]*b (F1: R=b, S= aa*a(ba*a)*ba*+aa*)
= b(a+ab)*[baa*a(ba*a)*ba* + baa*]*b
= b(a+ab)*[baa*aba*(aba*)* + baa*]*b (F1: R=ba*, S=a)
= b(a+ab)*[ba(a*aba*(aba*)* + a*)]*b (Factoring ba out)

= b(a+ab)*[ba(a*ab(a+ab)* + a*)]*b (F2: R=a, S=ab)

= b(a+ab)*[baa*(ab(a+ab)* + Λ)]* (Factoring a* out)

= b(a+ab)*[baa*(aba*)*]*b (F3: R=ab, S=a)

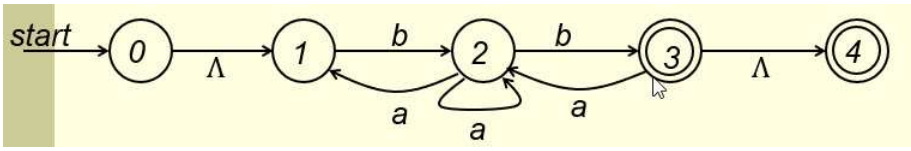
= b(a+ab)*[ba(a+ab)*]*b (F2: R=a, S=ab)

= b(a+ab+ba)*b (F2: B=a+ab, S=ba)

= Right side of (!!)

3. (10 points)

Given the following NFA over the alphabet {a, b},



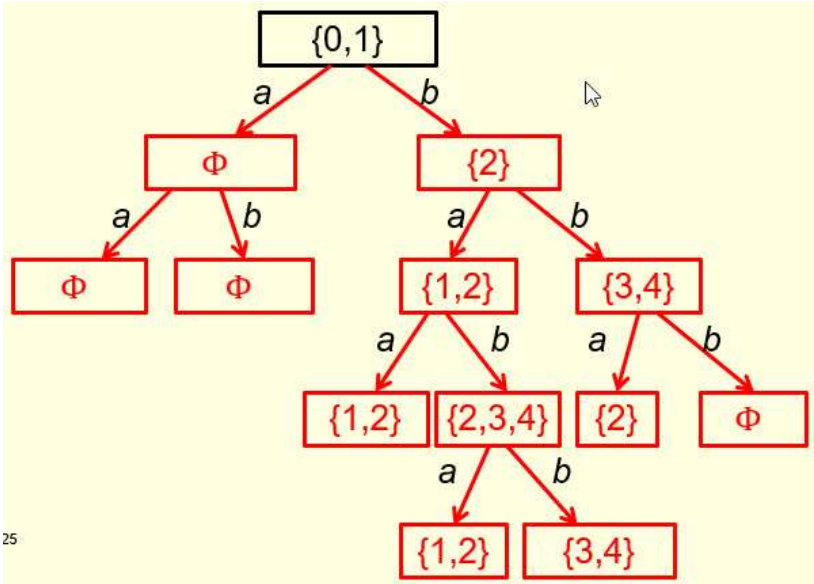
to transform it to a DFA, first construct Λ-closures of the states of the NFA. Fill out the blanks for state 0, state 1, state 3 and Φ below. (2 points)

Λ(0) = {0, 1} Λ(1) = {1}

Λ(2) = {2} Λ(3) = {3, 4}

Λ(4) = {4} Λ(Φ) = Φ

Next，to get the states of the DFA, we construct the following tree. Fill out the blanks in the tree. (6 points)



Distinct nodes in the above tree are listed below (for each level, nodes are taken from the tree from left to right). Fill out the blanks. (2 points)

{0,1}	Φ	{ 2 }	{1, 2}	{3, 4}	{2,3,4}

4. (10 points)

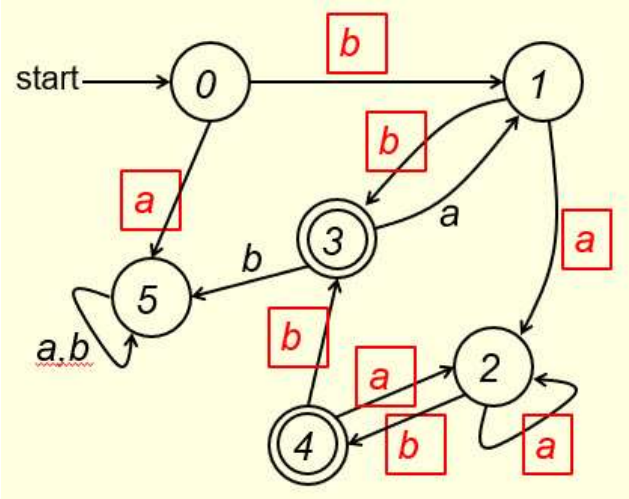
The above states are the states of the DFA to be constructed for the NFA given in Question #3. We have the following transition table for this DFA. Fill out the blanks in the table. (4 points)

	T_D	a	b
S	{0,1}	Φ	{2}
	{2}	{1,2}	{3,4}
	{1,2}	{1,2}	{2,3,4}
F	{3,4}	{2}	Φ
F	{2,3,4}	{1,2}	{3,4}
	Φ	Φ	Φ

Now replace the six states from top down with 0, 1, 2, 3, 4 and 5, respectively, the transition table of the DFA is of the following form. Fill out the blanks in the table. (2 points)

	T_D	a	b
S	0	5	1
	1	2	3
	2	2	4
F	3	1	5
F	4	2	3
	5	5	5

Hence, a DFA can be constructed as follows. Fill out the blanks in the DFA. (4 points)



Regular expression of this DFA equals regular expression of the NFA given in Question 1. Another way to see this is: a string that can be accepted by this DFA if and only if that string can be accepted by the NFA given in Question 1.