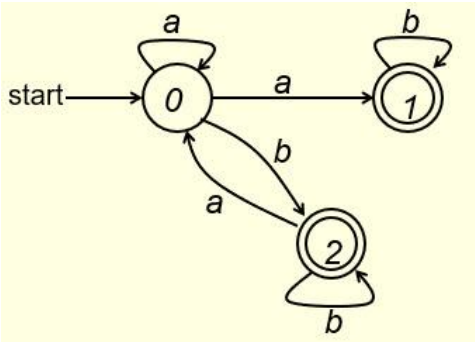
CS375 Homework Assignment 3 Solution Set (40 points)

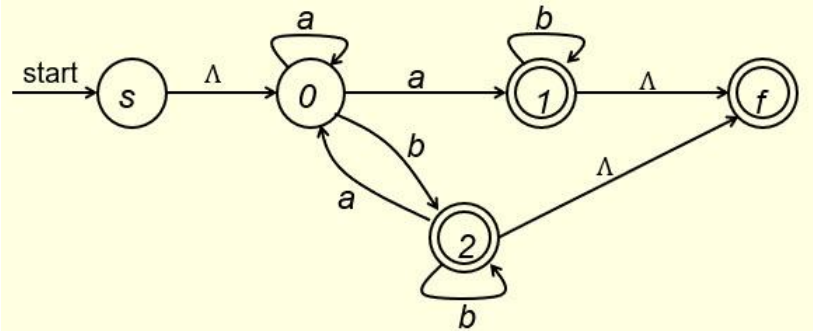
Due Date: Sept. 25, 2026

1. (8 points)

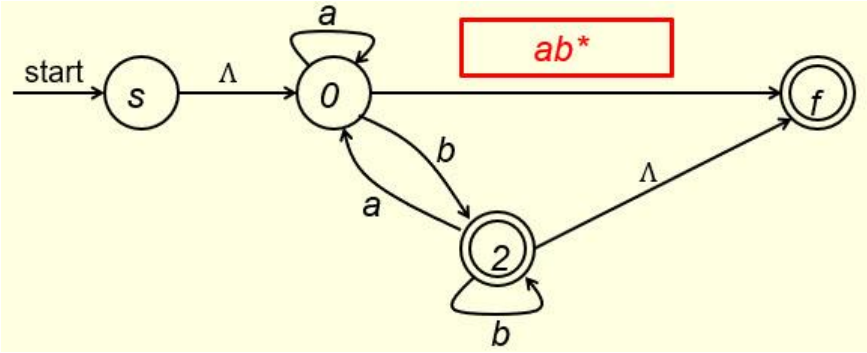
   To transform the following NFA to a regular expression,



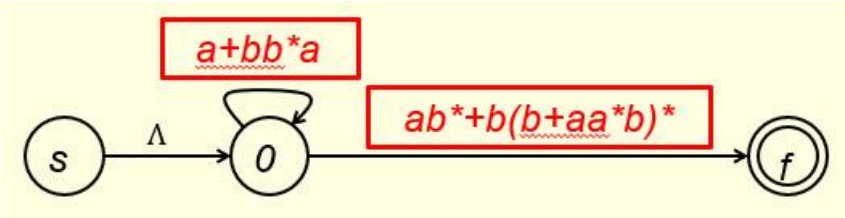
   we first connect a new start state s to the start state of the given NFA and connect each final state of the given NFA to a new final state f as follows.



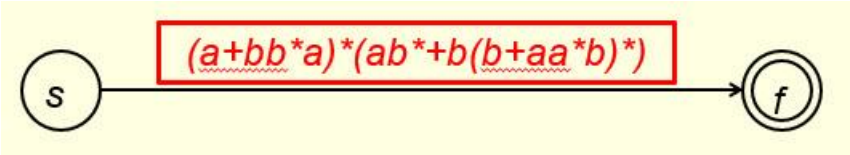
   If we eliminate state 1 first, the modified NFA becomes of the following form. Fill out the blank in the following figure. (1 point)



If we eliminate state 2 then, we get the following NFA. Fill out the blanks below. (5 points)

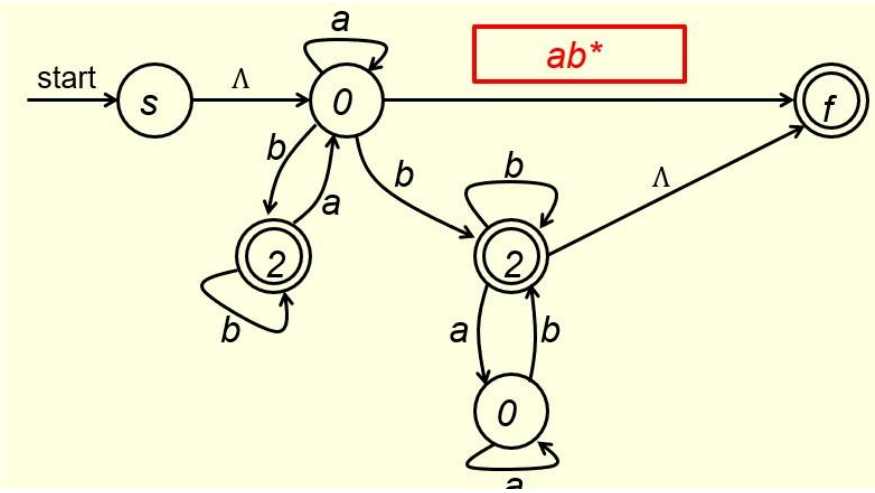


By eliminating state 0 we get the following NFA. Fill out the blank below. (2 points).

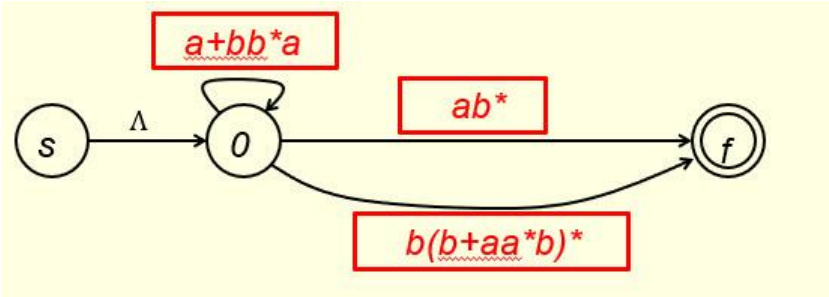


The expression in the above box is the regular expression of the given NFA. Actually, since a+bb*a = b*a and b+aa*b = a*b, the above expression can be simplified as (b*a)*(ab* + b(a*b)*). However, you are not required to do the simplification here.

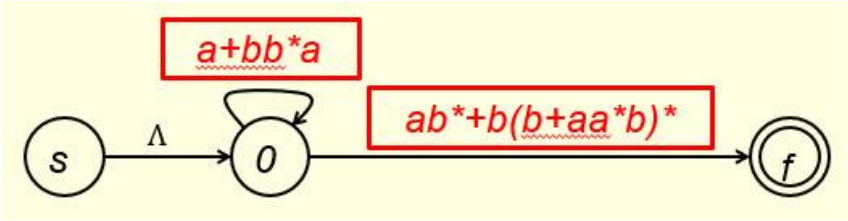
Note that after state 1 is eliminated, the NFA is functionally equivalent to the following case:



Now if we eliminate state 2, the inner loop and the outer loop of state 0 merge into a single loop with label *a+bb*a* and the path from state 0 through state 2 to state f become a single edge path with label *b(b+aa*b)** as follows:



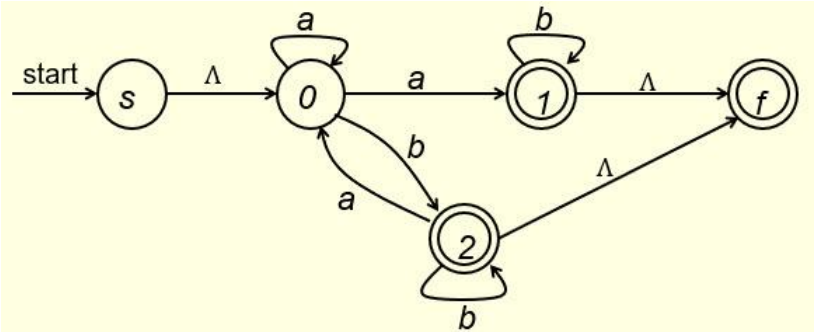
Therefore, after merge the two edges between state 0 and state f, we get the following result:



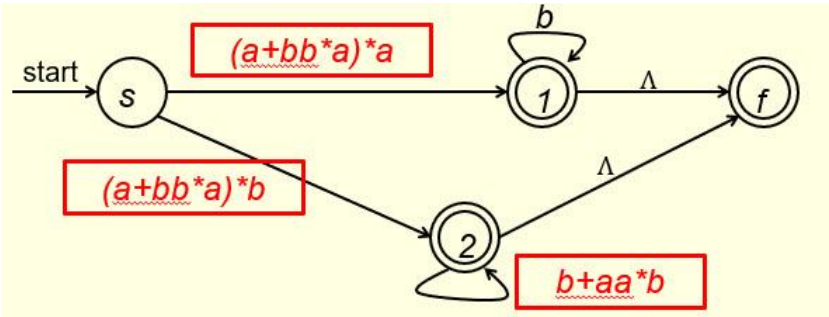
.

2. (11 points)

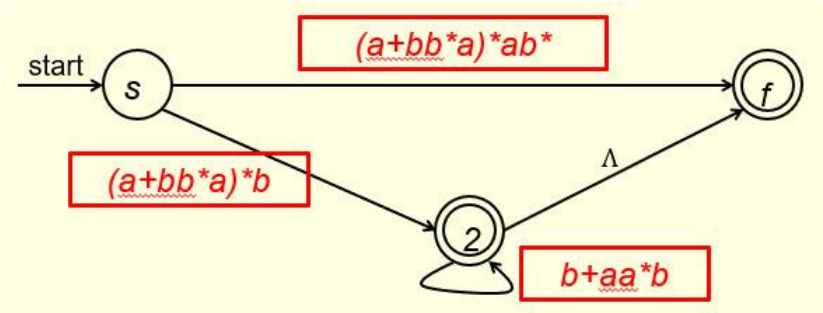
In Question #1, instead of eliminating states of the new NFA in the order of 1, 2 and 0,



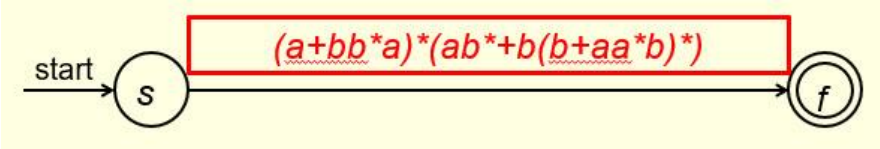
if we eliminate state 0 first, we get the following NFA. Fill out the blanks in the following figure. (6 points)



If we eliminate state 1 then, we get the following NFA. Fill out the blanks in the figure (only the new one will be graded. (1 point)

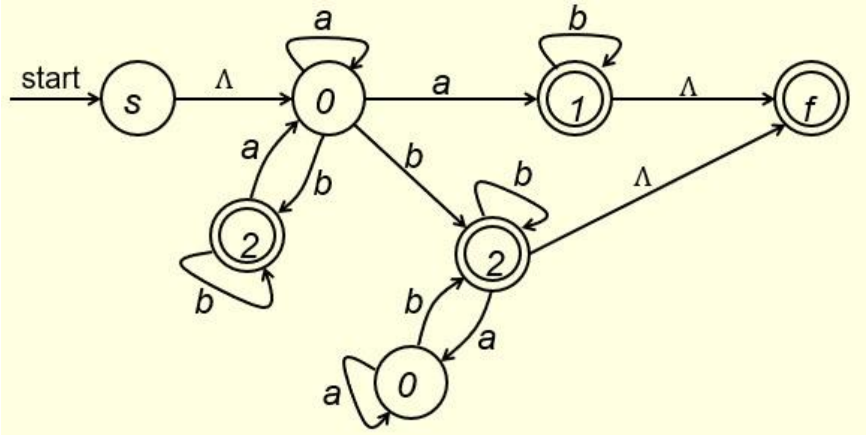


By eliminating state 2, we get the following NFA. Fill out the blank in the figure (4 points)

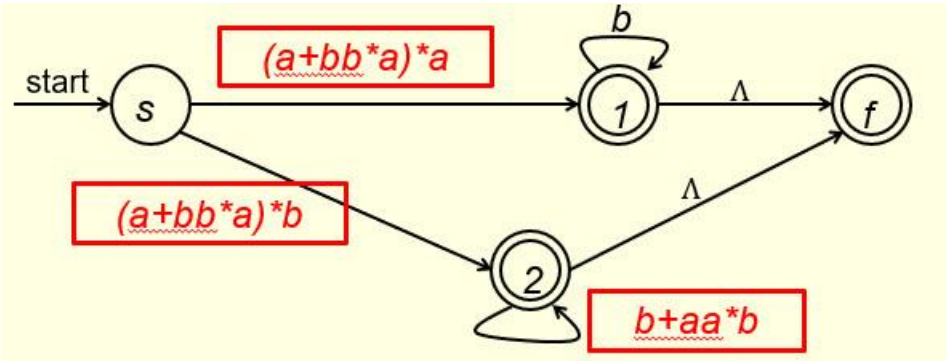


This expression should be the same as the one you get for Question #1. If your result here is different from your result for Question #1, then your result for Question #1, or your result for Question #2, or even both are wrong.

Again, the approach used in step 2 of Question #1 can be used here for the first step. The new NFA is functionally equivalent to the following case:

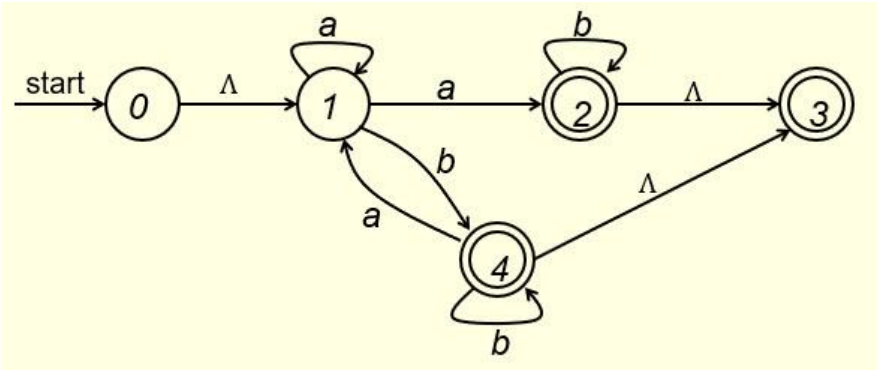


Now, if state 0 is eliminated, the path from state s through state 0 to state 1 becomes a single edge path with label *(a+bb*a)*a*, the path from state s through state 0 to state 2 becomes a single edge path with label *(a+bb*a)*b,* and the inner loop and outer loop of state 2 merge into a single loop with label *b+aa*b*. That is why we have the following result:



3. (9 points)

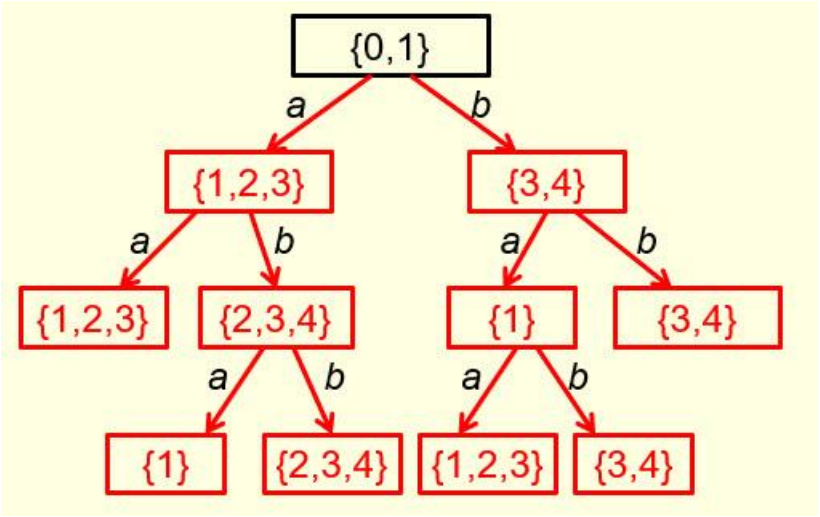
Given the following NFA over the alphabet {a, b},



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 4 and Φ below. (2 points)

Λ(0) = {0, 1}	Λ(1) = {1}
Λ(2) = {2, 3}	Λ(3) = {3}
Λ(4) = {3, 4}	Λ(Φ) = Φ

Next，to get the states of the DFA, we construct a tree as follows. Fill out the blanks in the following tree. (5 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}	{1,2,3}	{3,4}	{2,3,4}	{1}

4. (12 points)

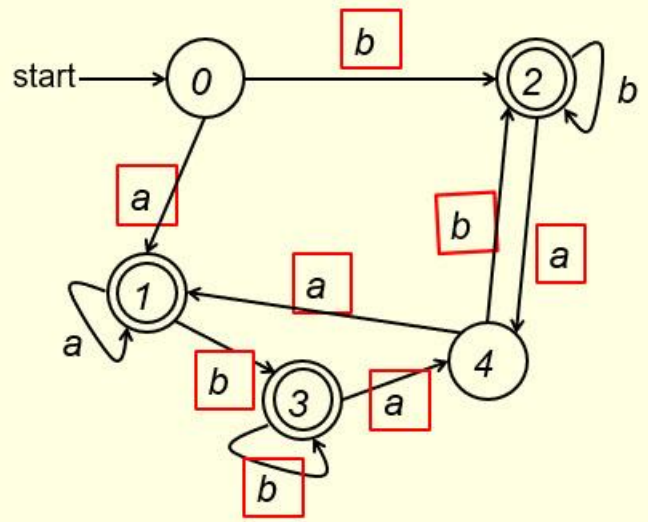
The above states are the states of the DFA to be constructed for the NFA given in Question #2. We have the following transition table for this DFA. Fill out the blanks in the table. (4 points)

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

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

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

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



Using the approach used in Question #1 or Question #2, One can show that the regular expression of the above DFA is

((bb*a + aa*bb*a)(bb*a + aa*bb*a)* + Λ)(aa* + bb* + aa*bb*) …. (E1)

(E1) can be simplified as

(a*bb*a)*(a*ab* + bb*) …. (E2)

It is easy to see that (E2) is also a regular expression of the NFA given in Question #1. Actually we can prove that (E2)=(b*a)*(ab*+b(a*b)*). Hence the above DFA is equivalent to the NFA given in Question #1 (note that the portion of the NFA between the start state 0 and the final state 3 in Question #3 is the NFA given in Question #1).