CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: \_\_\_**Derek Trom**\_\_\_\_\_\_\_\_

**Home Assignment 2: 102/100 points + 10 points (optional)**

Q1. [10/10] Find all strings in L((*ab* + *b*)\* b (*a* + *ab*)\*) of length ***less than*** four.

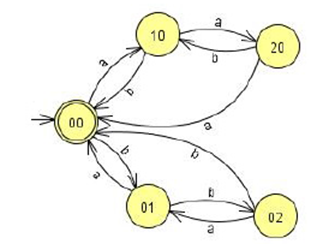
1. b, ba, bb, abb, baa, bab, bba, bbb

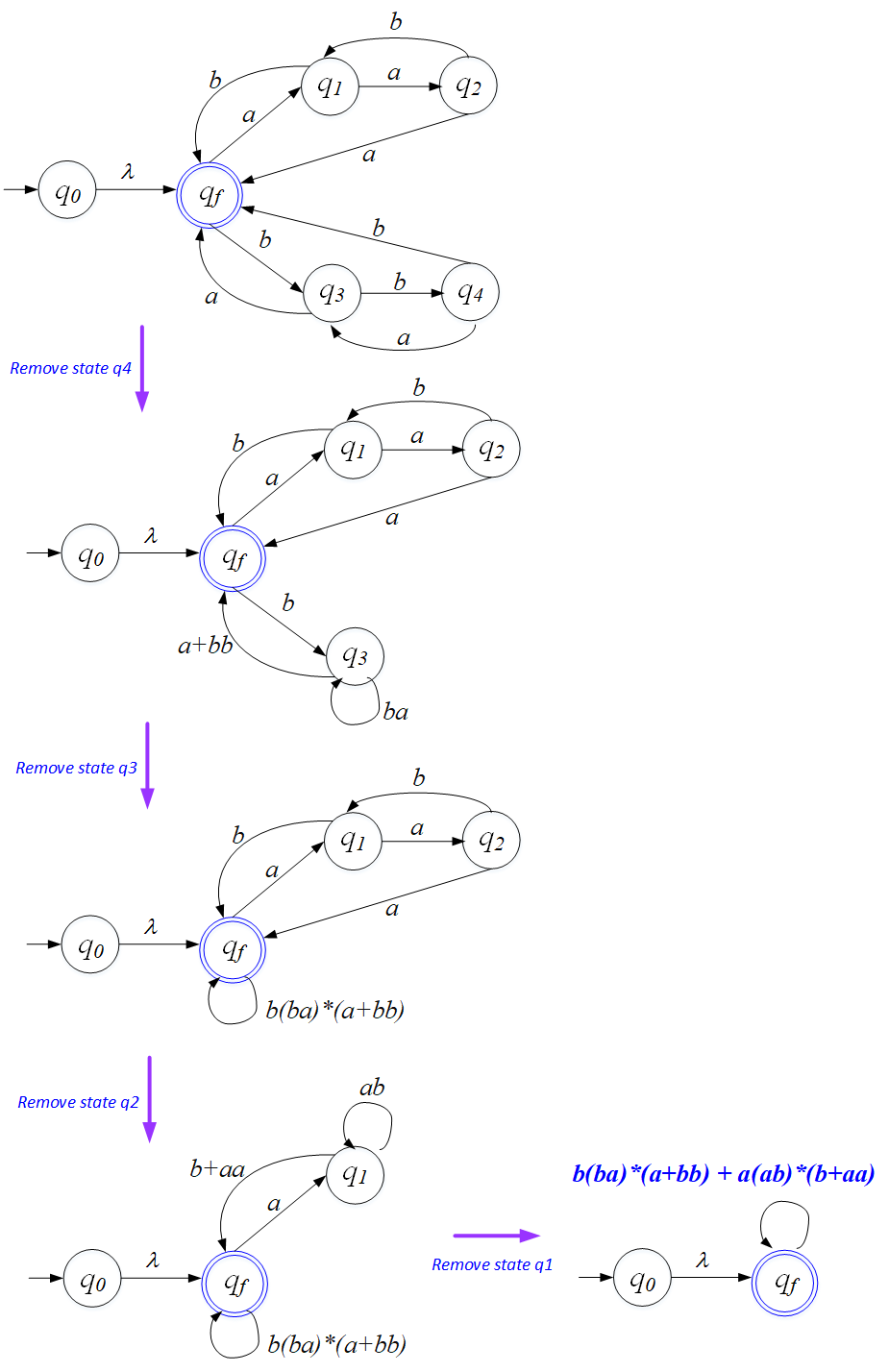
Q2. [17/10] Give a ***regular expression*** for the language

1. [10/10] L = {*anbm* | (*n*+*m*) is odd}.
   1. (aa)\*b(bb)\*
   2. a(aa)\*(bb)\*
   3. ((aa)\*b(bb)\*)+(a(aa)\*(bb)\*)
2. [7/10, optional] L = {*w* ∈ {*a, b*}\* | ( *na*(*w*) - *nb*(*w*) ) mod 3 = 0}. Hint: Apply Thm 3.2.
   1. r = (ba)\*((a+bb)(ab)\*(b+aa))\*

* See the attached solution

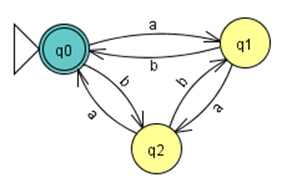
Case 1: NFA M, L(M) = L

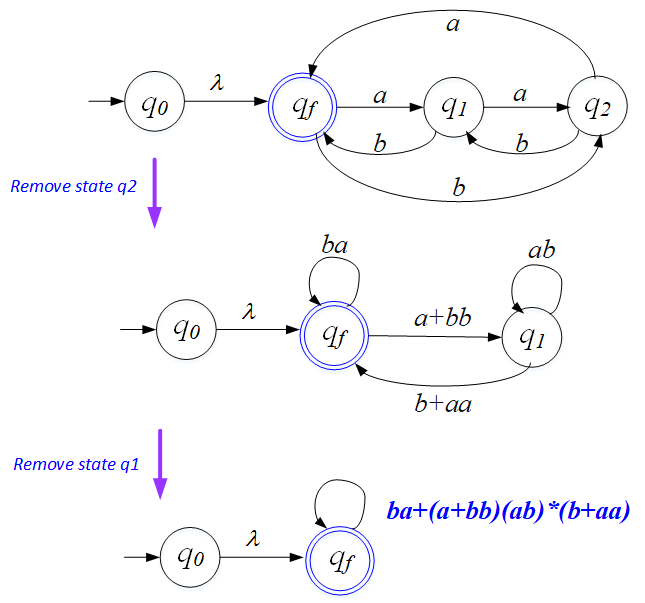




So, the REX is: *b(ba)\*(a+bb)+a(ab)\*(b+aa).*

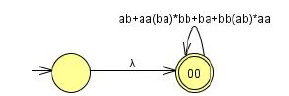
Case 2: NFA M





So, the REX is: *ba* + (*a+bb)*(*ab*)\*(*b*+*aa*).

Case 3:

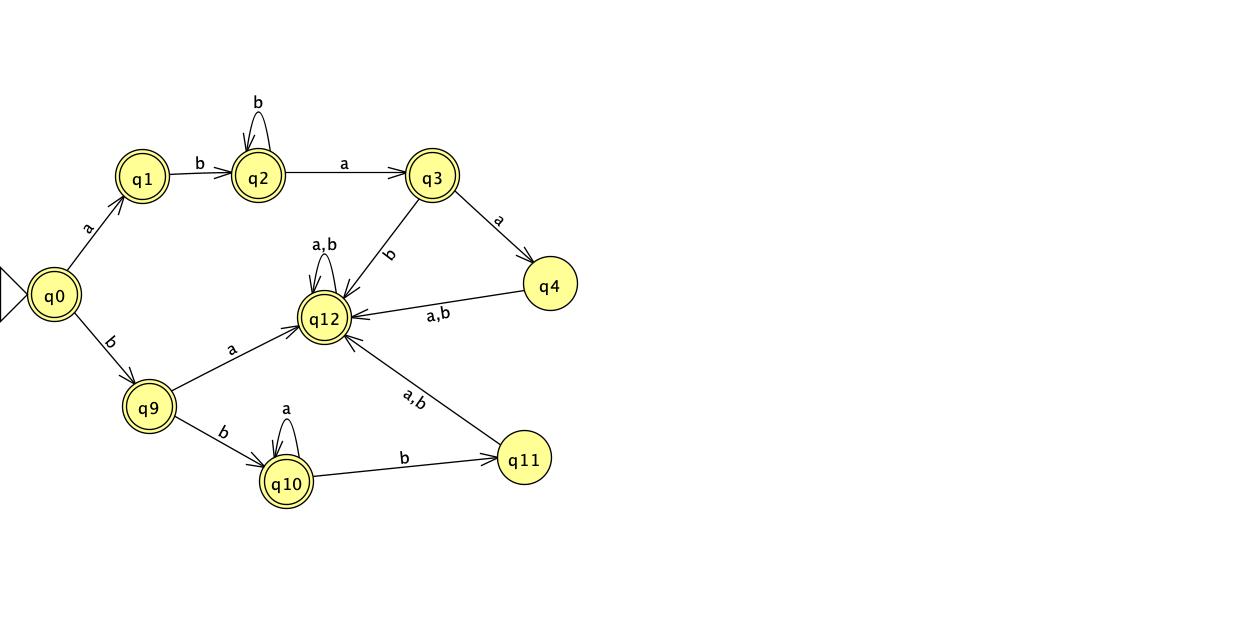


The REX is: (*ab* + *aa*(*ba*)\**bb* + *ba* + *bb*(*ab*)\**aa*)\*.

Q3. [8/10] Using the construction in Theorem 3.1, construct an NFA that accepts the complement of the

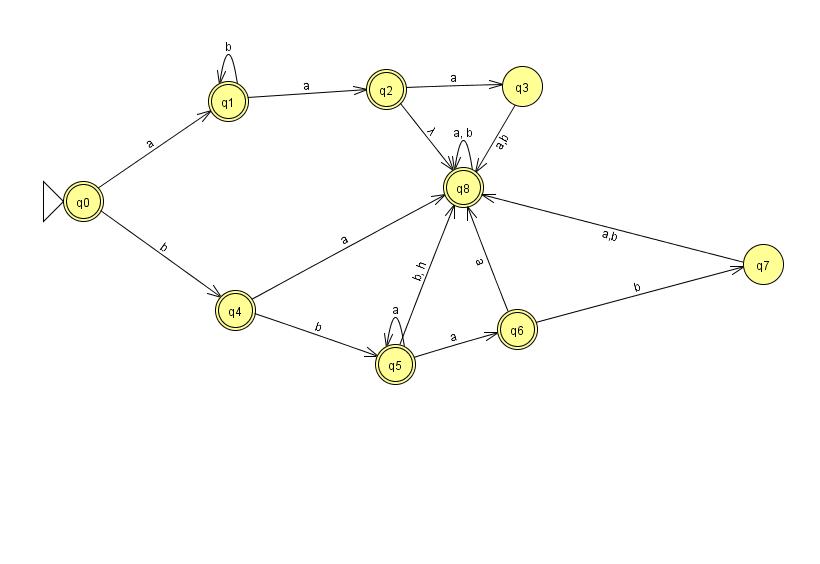
Language L(*ab*\**aa* + *bba*\**ab*).

NFA was constructed from L then converted to DFA and Final and nonfinal states reversed.

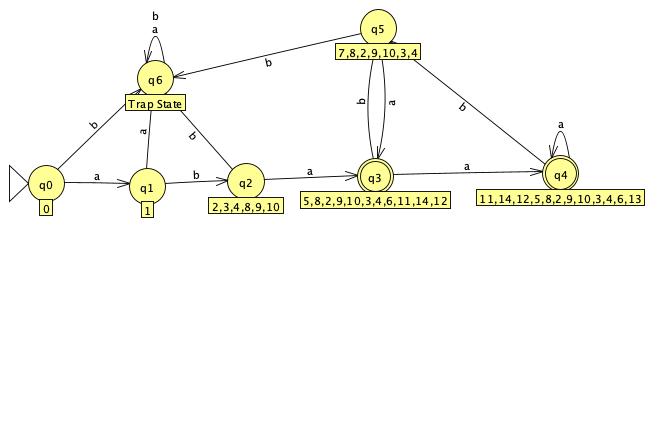


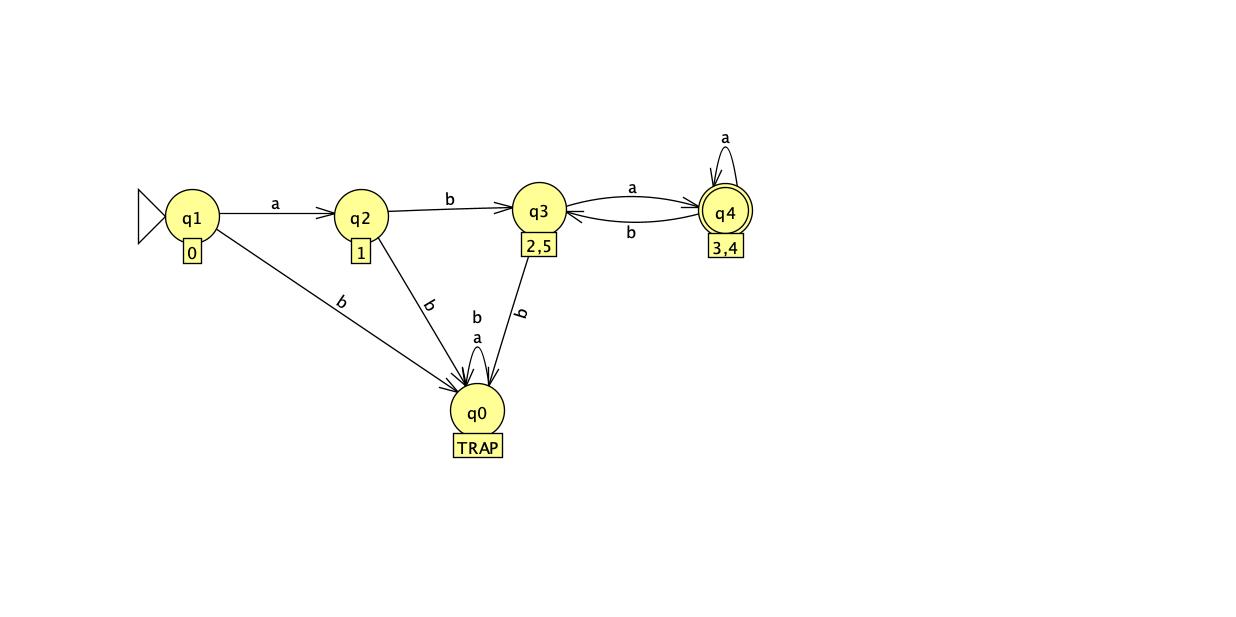
* See the attached solution

Convert the above NFA to a DFA. Then, complement it.



Q4. [20/20] Construct a ***minimal DFA*** that accepts the following language

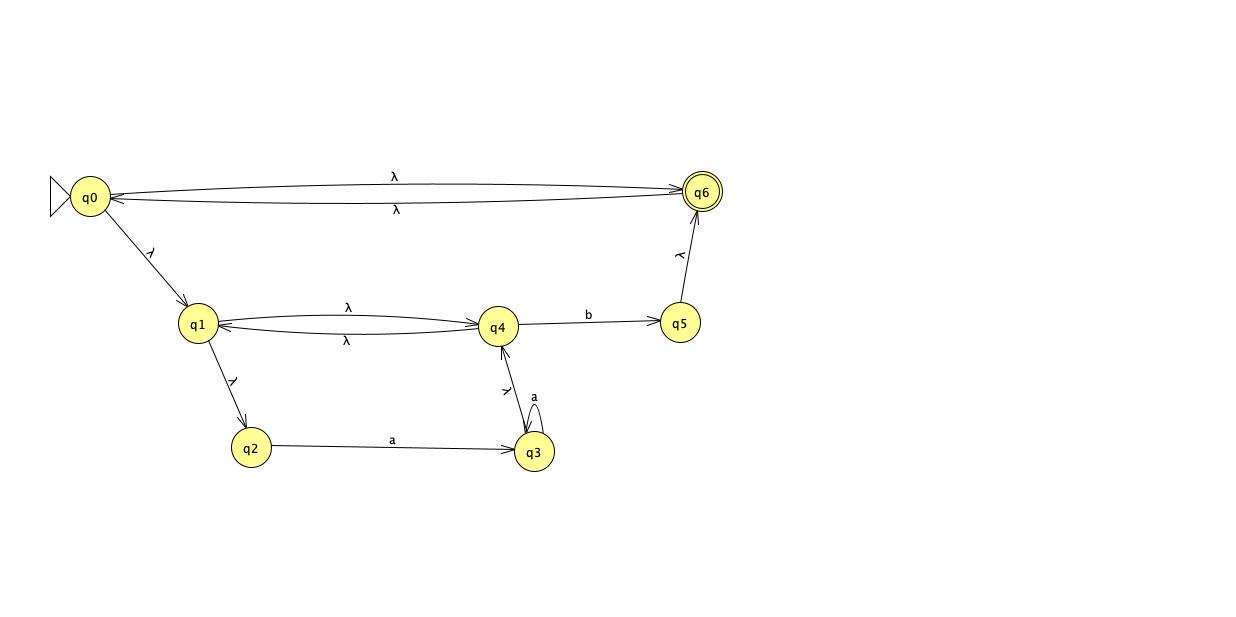
1. [10/10] L(*ab*(*a*+*ab*)\*(*a*+*aa*))
   1. dfa
   2. minimal dfa



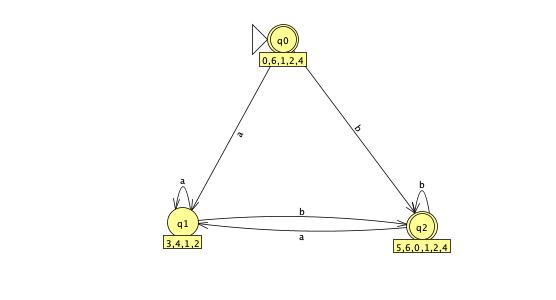
1. [10/10] L((*aa*\*)\**b*)\*)

Hint: Start with constructing an NFA (by Theorem 3.1), convert it to DFA, then get the minimal DFA by mark & reduce procedures.

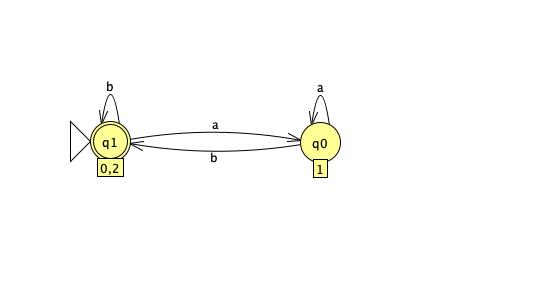
* 1. NFA



* 1. DFA

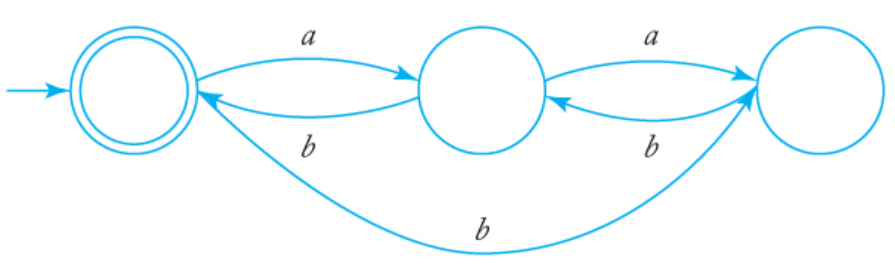


* 1. Min DFA



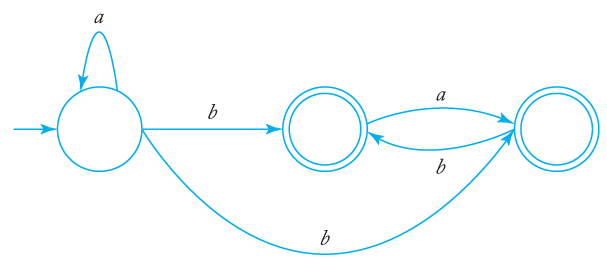
Q5. [17/20] Find ***regular expressions*** for the languages accepted by the following automaton.

1. [10/10]



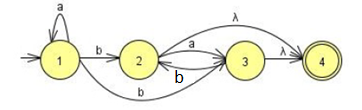
* 1. r = ((a+bb)(ab)\*b)\*

1. [7/10]

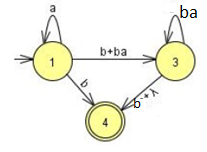
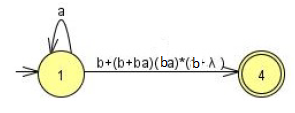


* 1. r = a\*(b+bb)(a+b)\*
* See the attached solution

**is equivalent to**

**with a single final state.**

Remove q2: Remove q3:

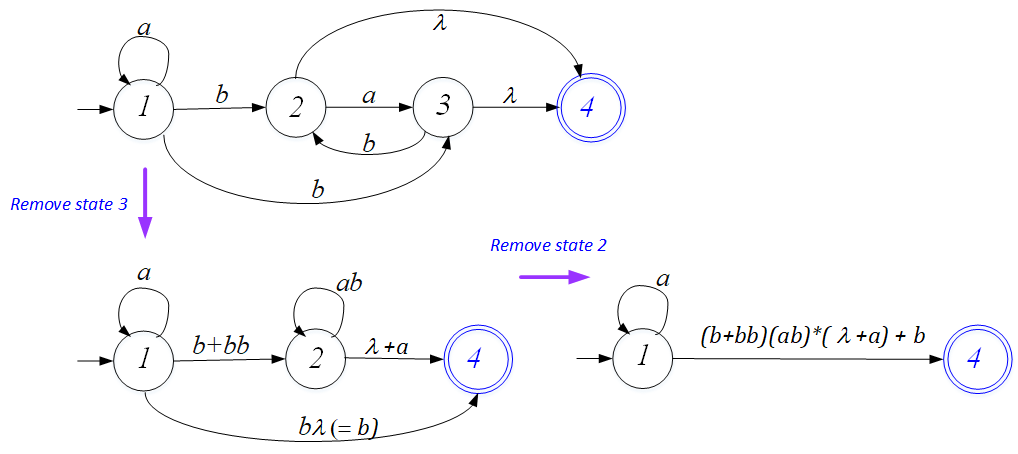
 

The regular expression then is: **r = *a*\*(b + (b + b*a*)(b*a*)\*(b + λ))**

**= *a*\*(b + b( λ + *a*)(*ba*)\*(b + λ )**

**= a\*b(λ + ( λ + *a*)(*ba*)\*(b + λ )**

OR



The regular expression then is: **r = *a*\*(b + (*b* + *bb*)(*ab*)\*( λ+*a*))**

**= *a*\*(b + (*λ* + *b*)b(*ab*)\*( λ+*a*))**

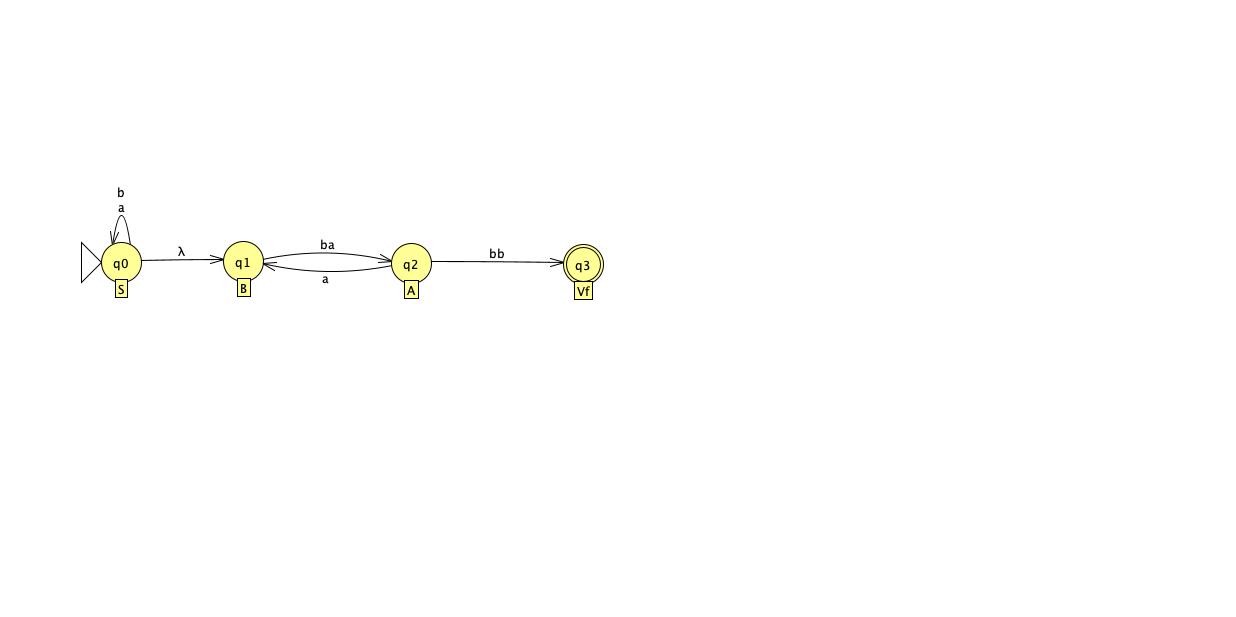
**= *a*\*(b + (*λ* + *b*)(b*a)\*b*( λ+*a*))**

**Any of them are correct.**

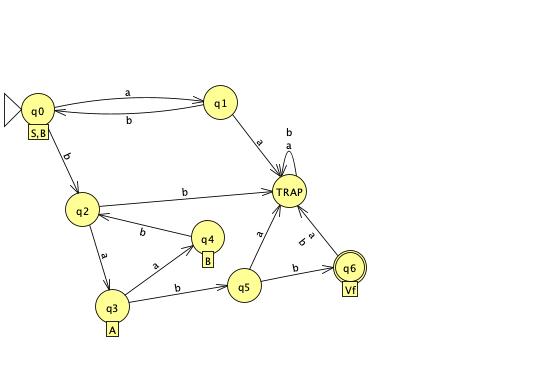
Q6. [10/10] Construct a ***DFA*** that accepts the language generated by the *grammar*

S → *ab*S | B, A → *a*B | *bb,* B → *ba*A.

NFA



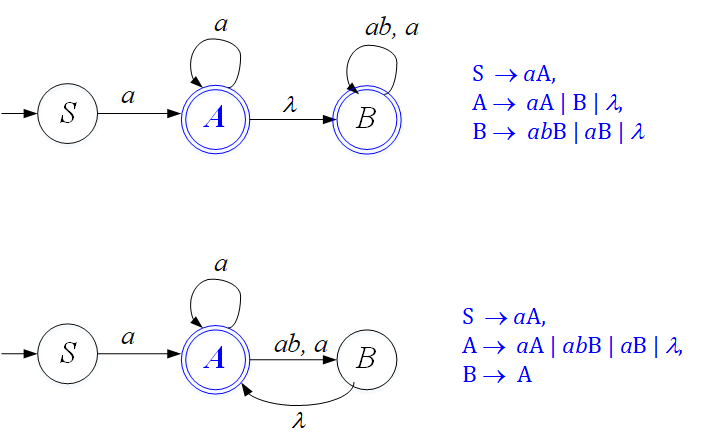
DFA



Q7. [20/20] Find a ***regular grammar*** that generates the language on Σ={a, b}

1. [10/10] *L*(*aa*\*(*ab*+*a*)\*)
   1. S → aA
   2. A → aA | aB | λ
   3. B → bA

* See the attached sample solution

1. 
2. { S → *a*A, A → *a*A | B | λ, B → *ab*B | *a*B | λ }
3. or { S → *a*A, A → *aA* | *ab*B | *a*B | λ, B → A }, etc.
4. [10/10] the language consisting of all strings with no more than two *a*’s.
   1. S → aA | bS | λ
   2. A → bA | aB | λ
   3. B → bB | λ