CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: \_Grant Haataja \_

**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.

*Ans*. *b, bb, ba, abb, bbb, baa, bab, bba.*

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

1. [10/10] L = {*anbm* | (*n*+*m*) is odd}.

*Ans.* (*aa*)\**b*(*bb*)\* + *a*(*aa*)\*(*bb*)\*.

1. [7/10, optional] L = {*w* ∈ {*a, b*}\* | ( *na*(*w*) - *nb*(*w*) ) mod 3 = 0}. Hint: Apply Thm 3.2. .

*Ans.* First we show the DFA for the above language.

A picture containing drawing

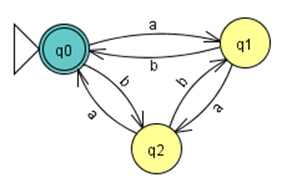
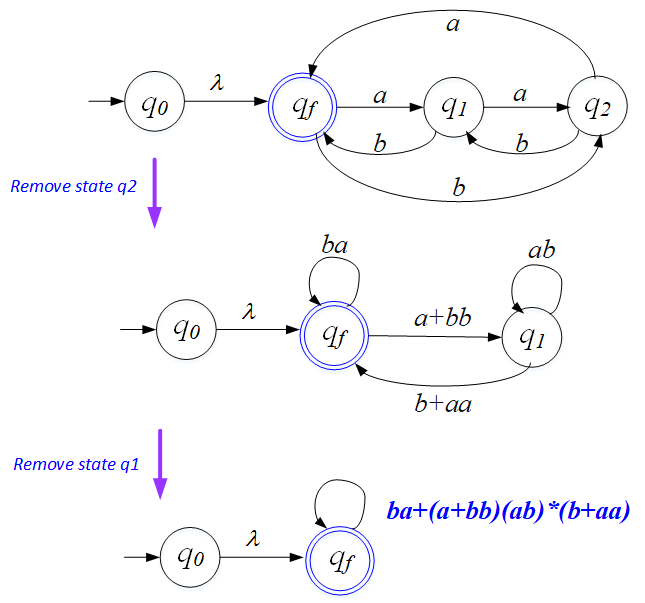
Description automatically generated

Then we reduce the automaton to a generalized transition graph.

A drawing of a person

Description automatically generated

From the GTG we obtain the regular expression: (*ab*+(*b*+*aa*)(*ba*)\*(*a*+*bb*))\*.

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

*Ans.* We first want to find the regular expression for the complement of the given language. To do this, we start by constructing an NFA for the given language.

A picture containing map, drawing

Description automatically generated

We then convert this NFA into an equivalent DFA.

A picture containing drawing

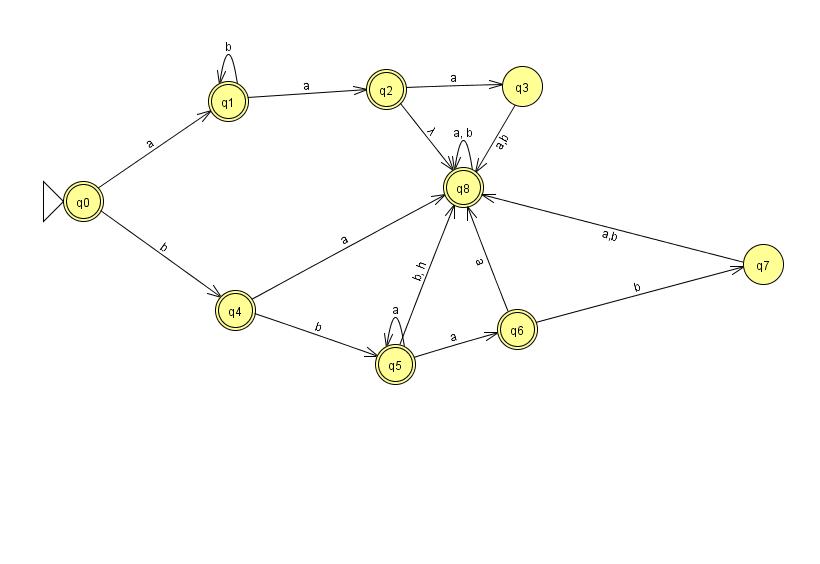
Description automatically generated

Next, we create the complement of this DFA by switching the accepting states to non-accepting states and vice versa.

A picture containing drawing

Description automatically generated

* See the attached sample answer



We then find the corresponding regular expression for this DFA, which is equivalent to the complement of our given regular expression.

Our new regular expression is L(*λ*+*a*+*aa*+*abb*\*+*abb*\**a*+*b*+*bb*+*bbaa*\*).

Finally, we follow Theorem 3.1 to systematically construct an NFA that accepts the above regular expression.

A close up of a logo

Description automatically generated

Q4. [19/20] Construct a ***minimal DFA*** that accepts the following language

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

*Ans.* First we construct an NFA to represent the given regular expression.

A picture containing drawing

Description automatically generated

Then, we convert the NFA into an equivalent DFA.

A picture containing drawing

Description automatically generated

Finally, we prove our DFA is minimal by showing that no state is unreachable, and no pair of states are indistinguishable.

δ(*q0, a*) = *q1* and *a* is in L so *q1* is not unreachable.

δ(*q0, ab*) = *q2* and a*b* is in L so *q2* is not unreachable.

δ(*q0, aba*) = *q3* and *aba* is in L so *q3* is not unreachable.

δ(*q0, b*) = *q4* and *b* is in L so *q4* is not unreachable.

States *q0* and *q1* are distinguishable since δ\*(*q0, aba*) ∈ F but δ\*(*q1, aba*) ∉ F.

States *q0* and *q2* are distinguishable since δ\*(*q2, ab*) ∈ F but δ\*(*q0, ab*) ∉ F.

States *q0* and *q3* are distinguishable since δ\*(*q3, a*) ∈ F but δ\*(*q0, a*) ∉ F.

States *q0* and *q4* are distinguishable since δ\*(*q0, aba*) ∈ F but δ\*(*q4, aba*) ∉ F.

States *q1* and *q2* are distinguishable since δ\*(*q1, ba*) ∈ F but δ\*(*q2, ba*) ∉ F.

States *q1* and *q3* are distinguishable since δ\*(*q3, a*) ∈ F but δ\*(*q1, a*) ∉ F.

States *q1* and *q4* are distinguishable since δ\*(*q1, ba*) ∈ F but δ\*(*q4, ba*) ∉ F.

States *q2* and *q3* are distinguishable since δ\*(*q3, ba*) ∈ F but δ\*(*q2, ba*) ∉ F.

States *q2* and *q4* are distinguishable since δ\*(*q2, a*) ∈ F but δ\*(*q4, a*) ∉ F.

States *q3* and *q4* are distinguishable since δ\*(*q3, a*) ∈ F but δ\*(*q4, a*) ∉ F.

Therefore, the DFA is minimal.

1. [9/10] 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.

*Ans.* We start again by constructing an NFA.

A picture containing drawing

Description automatically generated

Then, we convert the NFA into an equivalent DFA.

A picture containing drawing, clock

Description automatically generated

Finally, we prove our DFA is minimal by showing that no state is unreachable, and no pair of states are indistinguishable.

δ(*q0, a*) = *q0* and *a* is in L so *q0* is not unreachable.

δ(*q0, b*) = *q1* and *b* is in L so *q1* is not unreachable.

δ(*q0, ba*) = *q2* and *b* is in L so *q2* is not unreachable.

States *q0* and *q1* are distinguishable since δ\*(*q0, a*) ∈ F but δ\*(*q1, a*) ∉ F.

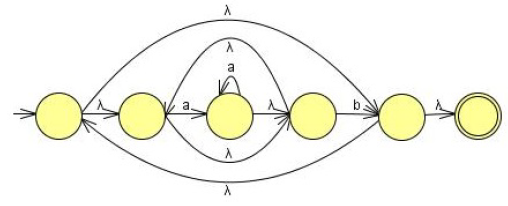
States *q0* and *q2* are distinguishable since δ\*(*q0, ab*) ∈ F but δ\*(*q2, ab*) ∉ F.

States *q1* and *q2* are distinguishable since δ\*(*q1, b*) ∈ F but δ\*(*q2, b*) ∉ F.

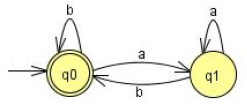
Therefore, the DFA is minimal.

* See the attached solution

Start with

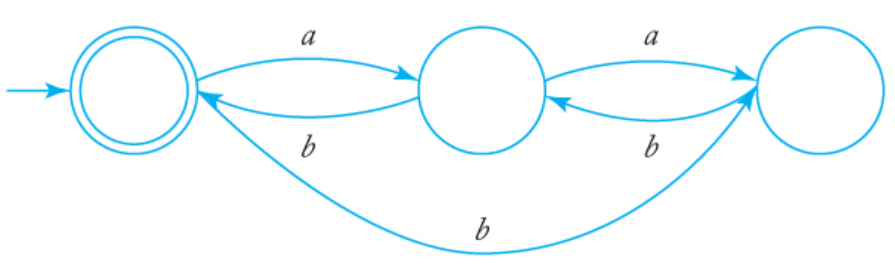


Convert NFA-to-DFA then reduce the # of states by mark-reduce in Chapter 2 to get the minimal DFA:



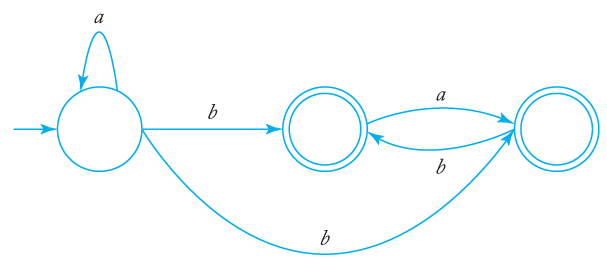
Q5. [18/20] Find ***regular expressions*** for the languages accepted by the following automaton.

1. [10/10]



*Ans.* *L*((*ab*+(*b*+*aa*)(ba)\*bb)\*)

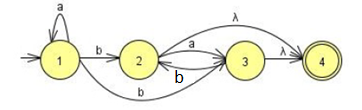
1. [8/10]



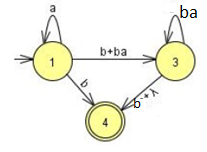
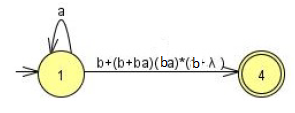
*Ans. L*(*a*\*(*b*+*ba*+(*bb*+*bab*)(*ab*)\*(*λ*+*a*)))

* See the attached solution

**is equivalent to**

**with a single final state.**

Remove q2: Remove q3:

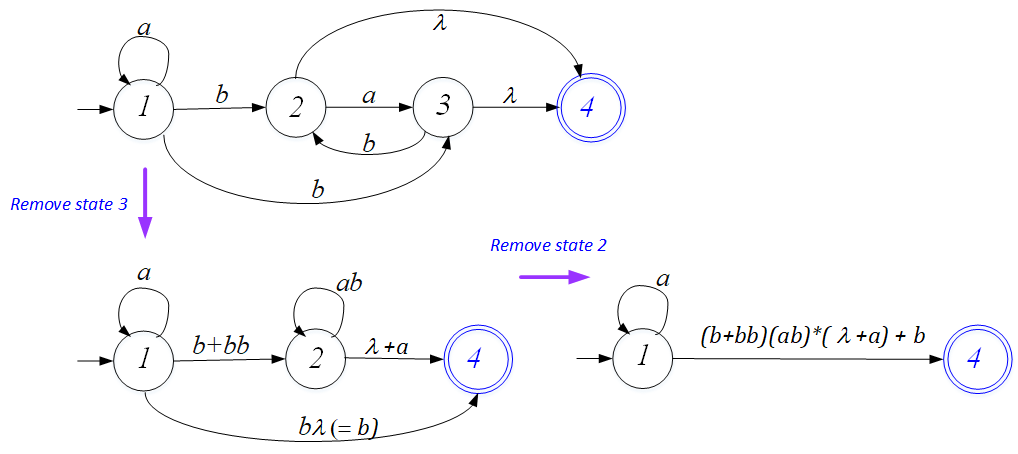
 

The regular expression then is: **r = *a*\*(b + (b + b*a*)(b*a*)\*(b + λ))**

**= *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.

*Ans.*

A close up of a logo

Description automatically generated

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

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

*Ans.* S → *a*A , A → *a*A | *a* | *ab*B*,* B → A | *λ*.

1. [10/10] the language consisting of all strings with no more than two *a*’s.

*Ans.* S → *b*S | *a*A | *b* | *λ*, A → *b*A | *a*B| *b* | *λ,* B → *b*B | *b* | *λ*.