CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: Elena Corpus

**Home Assignment 3: 66/100 points + 25 points (optional)**

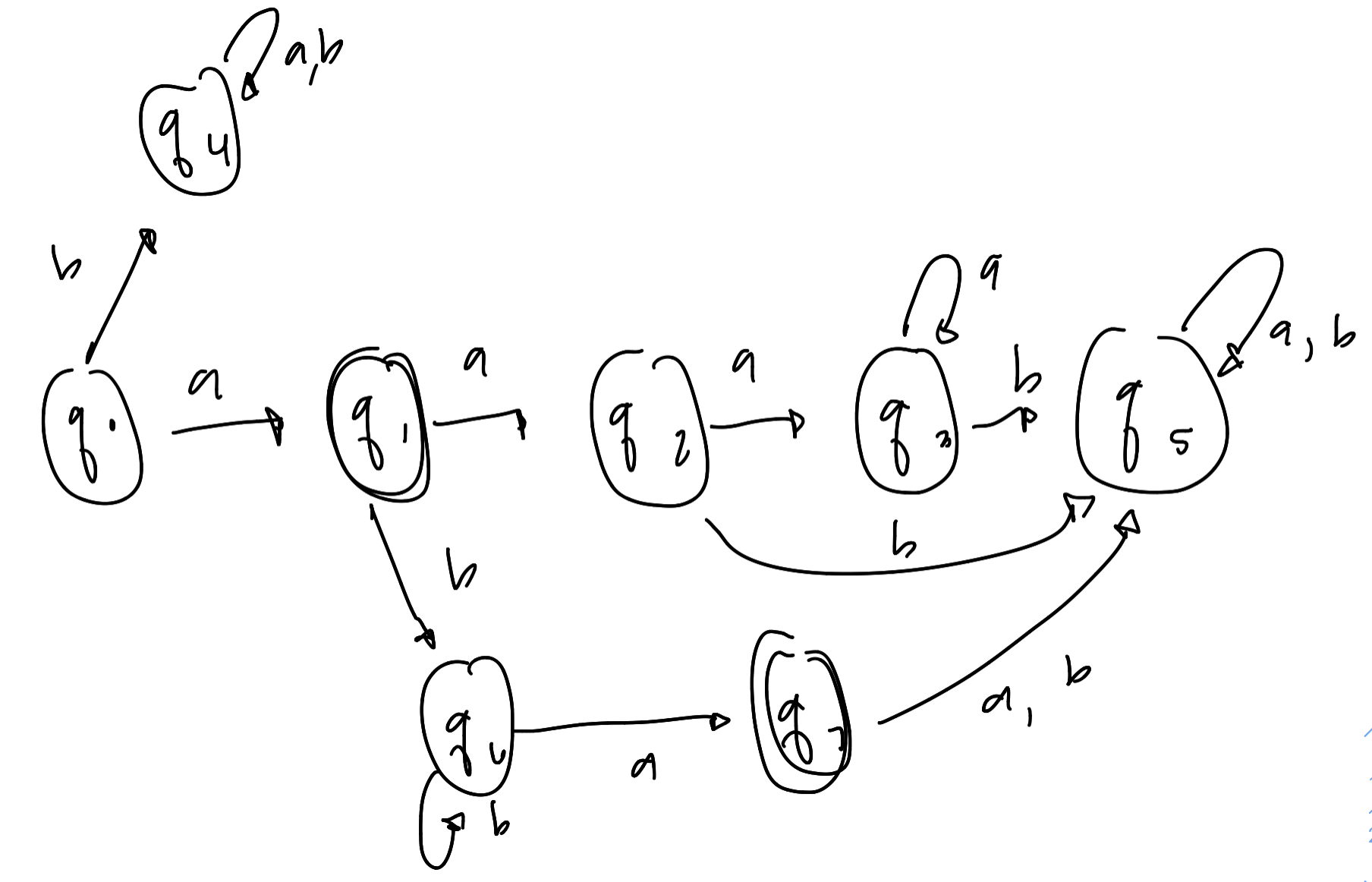
Q1. [9/10]

1. Use the construction in Theorem 4.1 to find an DFA that accept L(*ab\*a*\*) Ç L(*a\*b\*a*).

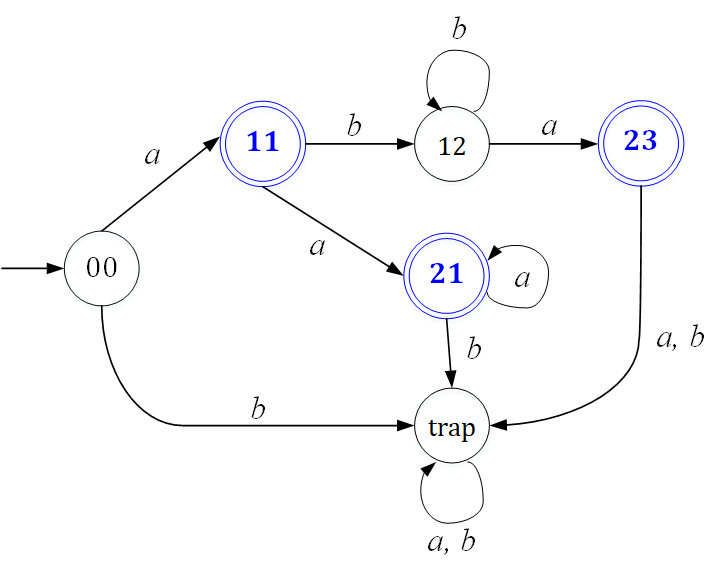
L1 = {a, ab, aa, aba, abb, aaa, abba, abbb, aaaa...}

L2 = {a, ba, aa, aba, aaa, bba, abba, bbba, aaaa...}

L1 A L2 = {a, aa, aba, abba, aaaa....}



* See the attached answer

****

1. [5/5, Optional] Give the regular expression for the above language in 1) that is accepted by your DFA.

{aa\* + ab\*a}

Q2. [0/10] The ***complementary or (cor)*** of two sets L1 and L2 is defined as

cor(L1, L2) = {*w* | *w* ÏL1 or *w* Ï L2, }.

Show that the family of regular languages is ***closed*** under ***cor.***

Since *w* ∉L1 or *w* ∉L1 , w ∈ or w ∈ .

Thus, cor(L1, L2) = { w | w ∈ or w ∈ } = ∪ .

The result then follows from closure under union and complement of regular language.

Q3. [0/10] The family of regular languages are closed under arbitrary ***homomorphism***.

Prove or disprove h(L1 ∩ L2) =h(L1) ∩ h(L2) is a regular language where L1 and L2 are regular.

L1 and L2 are regular, but R1 and R2 represent each

R1 + R2 = L1 U L2

R1R2 = L1 L2

R1\* = L1 \*

Let M be a DFA that accepts L1

(closed under complement and union) L1 U L2 = L1∩ L2 is a regular language

Let w = S1 S2 … Sn be a word over S. Then wR denotes the word Sn … S2 S1

lR = l

Let L be a language, then LR denotes LR = {wR : w Î L} = regular language

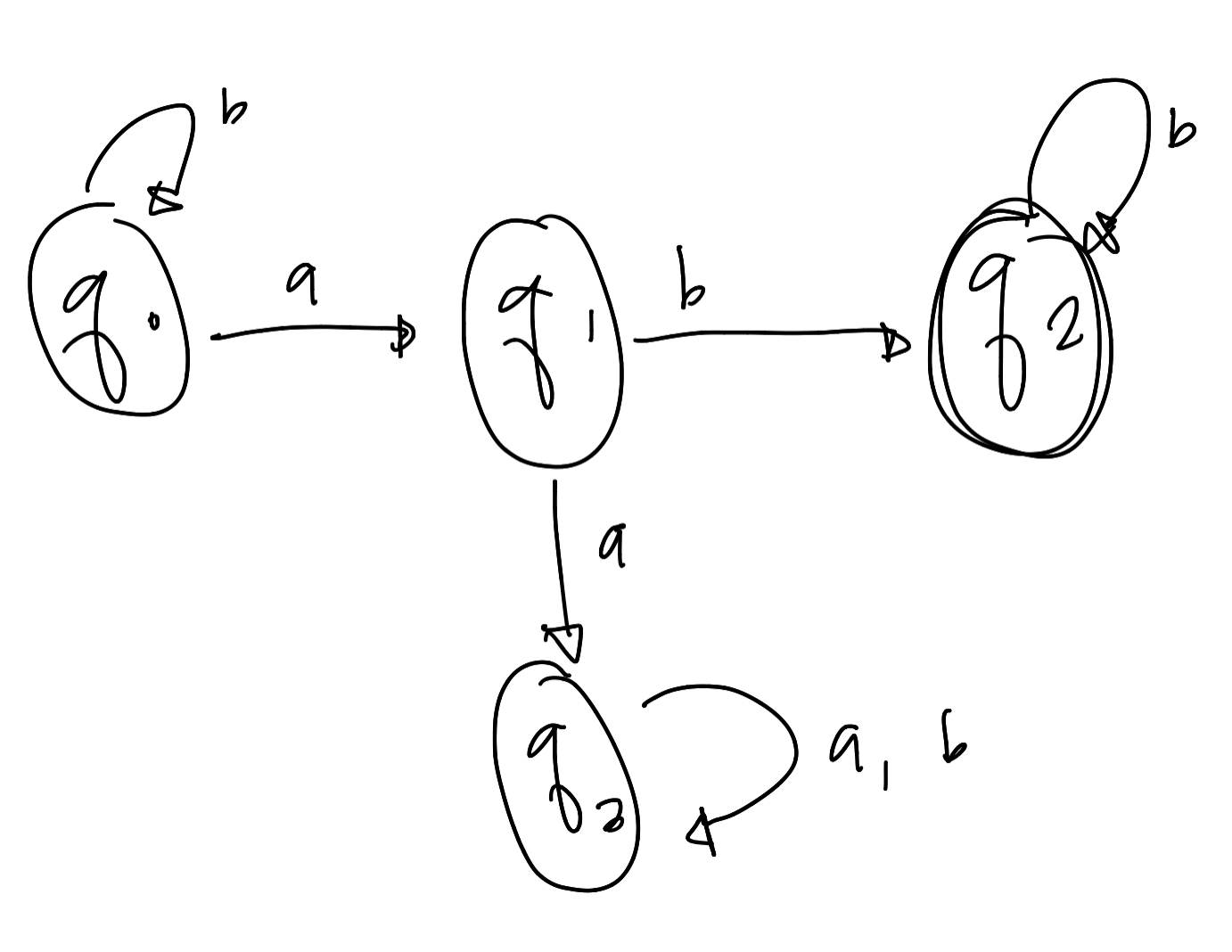
False. For example,

L1 = L(*a*\*), L2 = L(b\*), h(*a*) = *a* and h(b) = *a*, then, h(L1 ∩ L2) = ∅.

But, h(L1) ∩ h(L2) = L(*a*\*).

Q4. [10/10] Let L1 = {L(*b*\**abb*\*) and L2 = L(*bab*\*). Find the ***right quotient*** of L1 with L2, L1/L2.

1. [5/5] Let M be a DFA s.t. L(M) = L(L1). By applying Thm. 4.4, construct a DFA M’ s.t. L(M’) = L1/L2.



1. [5/5] Then, give a regular expression for L(M’) = L1/L2.

L(M’) = b\*

Q5. [10/10] If L is a regular language, prove that the language L2 = { *uv* | *u*Î LR , *v* ÎL } is also regular.

Prove that LR is also regular language

So given a regular language L,

So by definition there must be a finite state automata, with finite number of states, which starts from the initial state and finishes to a final state, with some transitions, and give us a string which is generated by language L.

there is exactly one final state.

reverse the direction of the transitions

change the initial state to the final state, and the final state to the initial state.

Hence, we have generated a FSM for LR. Hence LR is also a regular language.

u is generated by a regular language

v is generated by a regular language

L2 = regular, prove Concatenation of two regular languages is also a regular language.

Let there are two Regular languages L1 and L2, which are recognized by corresponding FSMs M1 and M2.

M1 has a final state and M2 has initial state, if we connect the final state of M1 to initial state of M2, then we have created a FSM which accepts L1 L2(also make final state of M1 a non-final state).

Now we can say than L2, is a regular language.

Since L is regular, so is LR.

Therefore, by closure under concatenation, L2 is regular.

Q6. [9/10] The ***left quotient*** of a regular language L1 with respect to L2 is defined as:

L2/L1 = { *y* | *x*Î L2 , *xy* ÎL1 }

Show that the family of regular languages is ***closed*** under the ***left quotient*** with a regular language.

Hint: Do NOT construct a DFA that accepts L2/L1 but use the definition of L2/L1 and the closure

properties of regular language.

Reverse language of L2/L1 is

L2/ L1)R = {yR : yRxR ∈ L1r, xR ∈ L2R} = L12/ L2R

We show the regularity by finding the relation between left quotient and right quotient.

y ∈ L2/L1 if and only if *xy* ∈ L1 for some x ∈ L2

if and only if *yRxR* ∈ L1R for some xR ∈ L2R

if and only if *yR* ∈ L1R/ L2R

if and only if  *y* ∈ (L1R/L2R)R

This shows that L2/L1 = (L1R/L2R)R

Therefore, by closure under reverse and right quotient, we have closure under left quotient

Q7. [8/10] Disprove that L1 = L1L2/L2 for all languages L1 and L2 . Give a counter example.

L1 / L2 = {e}

(L1/ L2) L2 = {b, bb}

L1 \* L2 = {ab, abb, bb, bbb}

(L1 L2)/ L2 = {a, ab, epsilon, b, bb}

Disproves all but;

(L1/ L2)\*L2 <= (L1 \* L2)/ L2 and L1 ⊆ (L1 \*L2)/L2

To Disprove =

Let L1 = {a, b} and L2 = {b, ab} then (L1/ L2)\* L2 = {b, ab} /⊆ {e, a, b, aa, ba} = L1 L2/L2

To disprove L1 = {a}, L2 = Ø, (L1 \* L2) / L2 = Ø / Ø - Ø ⊄ {a}

L1 = { *anbn* | *n* ≥ 0 }, L2 = { *bm* | *m* ≥ 0 }.

Then, L1L2 ={ *anbn+m* | *n, m* ≥ 0 } = { *anbm* | *m* ≥ *n* ≥ 0 }.

Then, L1L2/L2 = { *anbm* | *n≠ 0, m* ≥ 0 }

Q8. [10/10] A language is said to be a ***palindrome*** language if L = LR. (4.2-3)

Show that there exists an ***algorithm*** for determining if a given regular language is a palindrome language.

1. Represent the regular language L as a DFA, set to D
2. Construct a NFA as N, that accepts LR
3. Convert N into an equivalent dfa, D so that L(D) = LR
4. Theorem 4.7 to determine if L= Lr, which will give if it is a palindrome

Q9. [0/20] Pumping Lemma

1. [0/10] Prove that the language L = {*anbkcn* | *n* ³ 0, *k* ³ *n* } is ***not regular***.

{aabbbcc, abbcc, abc, aaabbbccc....}

Consider aabbcc : x = a, y = abbb, z = cc

|y| 4 >= 0

Suppose L is regular and we’re given m, then let’s choose w = *ambmcm* ∈ L with |w| ≥ *m* for *k* = *n* = *m*.

Let’s partition w to xyz, s.t. |y| ≥ 1 and |xy| ≤ *m* > 0, for any 0 < m .≤ |w|

Because |y| ≥ 1 and |xy| ≤ *m*, it’s clear that y = *aq* is the only possible choice for some q ≥ 1, i.e.

x = *ap*, y =*aq* , z = *am-p-qbmcm*, p, q > 0 so |xy| = p+q ≤ *m*.≤ |w|

If L is regular, any *wi = xyiz* should be in L for any *i* ≥ 0.

However, w0 = xy0z = *apa0**am-p-qbmcm =**am-qbmcm* ∉ L*.* since *m-q ≠ m* for *q > 0.*

Thus, L is not regular.

1. [0/10, Optional] Prove that the language L = {*w* | *na*(*w*) ¹ *nb*(*w*)} is ***not regular***.

|xy| = 5

|c| = length = 7

5 <= 5 … True

By the closure property of of the regular language, if  is not regular, then L is not regular.

So, show that = { *w* | *na*(*w*) = *nb*(*w*)} is not regular. Then, its complement L is not regular.

Assume that is regular. Then, apply the Pumping Lemma to w = *ambm* ∈L.

Let’s partition w to xyz, s.t. |y| ≥ 1 and |xy| ≤ *m* > 0, for any 0 < m .≤ |w|

Because |y| ≥ 1 and |xy| ≤ *m*, it’s clear that y = *aq* is the only possible choice for some q ≥ 1, i.e.

x = *ap*, y =*aq* , z = *am-p-qbm*, p, q > 0 so |xy| = p+q ≤ *m*.≤ |w|

If is regular, any *wi = xyiz* should be in for any *i* ≥ 0.

However, w0 = xy0z = *apa0**am-p-qbm =**am-qbm* ∉ *.* since *m-q ≠ m* for *q > 0.*

Thus, is not regular.

Therefore, L is not regular.

1. [0/10] Prove or disprove that L1 È L2 is not regular language if L1 and L2 are not regular languages.

For all K >= 0 , xy^k z

For k = 2

(a)(abbb)^2(cc) = aabbbabbbcc = not a regular language

Disapprove because it can be regular

L1 UL2 = a\*b\* which is regular

False.

Counter Example: Let L1 = { *anbm* | *n* ≤ *m* } and L1 = { *anbm* | *n* > *m* }.

Both L1 and L2 are not regular.

But, L1 ∪ L2 = L(*a*\**b*\*) is regular.

Q10 [5/10, optional] The ***min*** of a language L is defined as

***min***(L) = { *w* ÎL | there is no *u* ÎL, *v*ÎS+, such that *w* = *uv* }.

Show that the family of regular languages is closed under the ***min*** operation.

Describe the strings which are ineligible for min(L) and exclude them using set difference. The ineligible strings are LΣ +, since w ∈ LΣ + means that w = xy where x ∈ L. That is, w has a proper prefix x which is in L.

thus, min(L) = L − LΣ+ , which is regular since we know that regular languages are closed under set difference.

A string w ∈ L is in min(L) if and only if it has no prefix u ∈ L.

This suggests that a DFA for min(L) can be constructed from DFA for L by removing the outgoing edges from all .