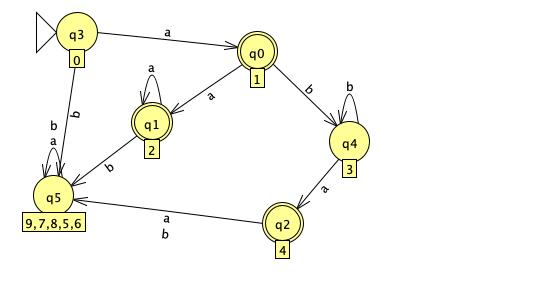
CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: \_\_\_\_\_\_\_\_Derek Trom\_\_\_\_\_\_

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

Q1. [10]

1. Use the construction in Theorem 4.1 to find an DFA that accept L(*ab\*a*\*) ∩ L(*a\*b\*a*).
   1. 
2. [5, Optional] Give the regular expression for the above language in 1) that is accepted by your DFA.
   1. (aa\*)+(ab\*a)

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

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

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

1. Assume that L1 and L2 are regular. Let the r1 and r2 denote L1 and L2 respectively. Then r1+r2 denotes L1 ∪ L2, r1r2 denotes L1 ∩ L2, r1\* denotes L1\*. Let M = (Q, Σ, δ, q0, F) be a DFA that accepts L1. Then = (Q, Σ, δ, q0, Q-F) accepts since the regular languanges are closed under the compliment and union. = is a regular language. Let w = s1,s2…sn be a word over Σ. Then wR represents sn… s2,s1 the reverse of w. Let L be a language the LR denotes LR = {wR : w ∈ L} which is the reverse of L. Thus cor(L1, L2) = {*w* | *w* ∉L1 or *w* ∉ L2, } are regular where L1 and L2 are regular.

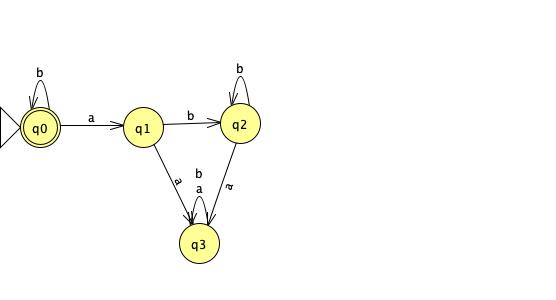
Q3. [10] The family of regular languages are closed under arbitrary ***homomorphism***.

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

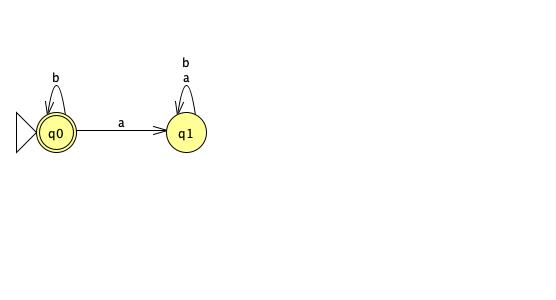
1. Let’s suppose that L1 and L2 are regular as stated.We know that L1 ∩ L2 is regular because they are regular under closure where L1 and L2  are regular. Now because L1 and L2 are regular and L1 ∩ L2 is also regular it is proven that h(L1 ∩ L2) =h(L1) ∩ h(L2) is also regular.

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

1. [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. The DFA M’



* 1. The minimized dfa



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

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

1. If L is a regular language then *u*∈ LR is a regular language under closure of reversal. *v* ∈L is also regular because it is assumed that L is regular. By theorem 4.1 the concatenation of u and v is also regular given LR and L are regular. Thus proving that L2 is also regular.

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

L2/L1 = { *y* | *x*∈ L2 , *xy* ∈L1 }

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.

1. We get the reverse of the language which is (L2/L1)R= { *y* R| *yRxR*∈ L1R , *xR*∈L2R }= L1R/L2R
2. If L1 and L2 are regular we have that L1R and L2R are regular. Now, because the right-quotient of L1R with L2R, i.e., L1R / L2R , is regular (from Theorem 4.4 in the textbook), we have that (L2/L1) R is regular. We have that ((L2/L1) R) R = L2/L1 is regular.

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

Let L1 = am , m≥0, the strings in L1 are {∈,a,aa,aaa,aaaa,aaaaa,aaaaaa,..}. Let L2 = bn n≥1,the strings in L2 are {b,bb,bbb,bbbb,bbbbb,bbbbbb,..} and so on. L1L2 a will equal ambn m≥0 n≥1. L1L2 will be the set of strings with 0 or more a’s followed by at least 1 b. So if L1 contains a, L2 contains bb, the L1L2 will be abb. L1L2/L2 contains ab. Thus L1 is not necessarily equal to L1L2/L2.

Q8. [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. Find an NFA for L that contains only one accepting state. Reverse all the arcs on the machine that make the accepting the initial state and the accepting the initial. Convert it then to a dfa. This DFA accepts Lr. Then show that the two accept the same language.

Q9. [20] Pumping Lemma

1. [10] Prove that the language L = {*anbkcn* | *n* ≥ 0, *k* ≥ *n* } is ***not regular***.
   1. Let m be the constant in the pumping lemma. We choose w = ambmcm ∈ L, |w| ≥ m. For all possible x, y, z with w = xyz, |xy| ≤ m, |y| ≥ 1.
   2. In the case of x = a m−r , y = a r , z =, r ≥ 1. We let i = 0. xy0 z = am−r am ∉ L, because m − r ≠ m.
   3. Thus L is not regular.
2. [10, Optional] Prove that the language L = {*w* | *na*(*w*) ≠ *nb*(*w*)} is ***not regular***.
   1. To show this we can use it’s compliment to prove it. Assume L is regular. Therefore pumping lemma holds for L. Let m be the magic number for this language. If we consider the example ambm it is clearly longer then m and belongs to the language because a and b are equal. By pumping lemma the string can be represented as |xy| <= m and y is non empty. So xyiz is in L for all values of L. But since xy <= m and there are m a’s at the beginning of the string. XY must start with only a’s. So y = ak with k >0. If we see xy2z that would mean am+kb. Thus this means that a’s do not equal b’s and it would not be part of the language. Pumping lemma does not hold for this and it is not regular and because L = {*w* | *na*(*w*) ≠ *nb*(*w*)} is the compliment of L we can say this is also not regular.
3. [10] Prove or disprove that L1 ∪ L2 is not regular language if L1 and L2 are not regular languages.
   1. Suppose L is not regular
   2. Consider w = ambn = xyz
   3. With |xy| <= m = l and |y|>= 1
   4. Then 1<= |y| = k<l
   5. So, y = ak, l >k>=1.
   6. However when i = 0, w0 = am+kbm ∈ L.

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

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

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

1. In order to show that it is closed under the min operation. We 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.