CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: Pedro Schmidt \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Due 10/4 by 5:00pm

**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*).

A close up of a clock

Description automatically generatedA clock on the wall

Description automatically generated

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

Regular Expression is {aba}

{aa\*+ab\*a}

Q2. [10] The ***complementary or (cor)*** of two sets S1 and S2 is defined as

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

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

***Proof:***

Cor (L1, L2) = {*w* | *w* ∉L1 or *w* ∉ L2,} is regular

Cor (L1, L2) = the compliment of L1 Uthe compliment of L2

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

If L1 & L2 are regular languages,

then we can assume that T1 = {Q1, E, d1 , q0, Q1 – f1} and T2 = { Q2, E, d2 ,q0, Q1 – f2} are the DFA’s that accept L1 and L2. This means that the compliment of T1 and the compliment of T2 (the DFA’s thereof) accept the compliment of L1 and the compliment of L2. This proves that L1 and L2 are regular, and compliments of L1 and L2 are also regular by the definition of complementation of DFA.

From last equation we know that the regular languages are closed under finite union. Meaning Cor (L1, L2) is 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.

First of all,

F( a O b) = F(a) O1F(b)

O and O1 are binary operation.

F(a),F(b) -> g1

We know L1 ∩ L2 = L1 U L2 , L1 = complement of L2

We already prove that L1 and L2 are regular then L1 ∩ L2 is also regular, by definition we know homomorphism so , h (L1 ∩ L2) =h(L1) ∩ h(L2) is also regular because L1,L2 and L1 ∩ L2 are regular. This claim can be proven.

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.

A close up of a clock

Description automatically generated

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

DFA of L , L1/ L2= L(b\*)

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

Given L is a regular expression and L2 = { *uv* | *u*∈ LR , *v* ∈L }. By building an FA that accepts L2 then L2 is considered regular. Let P be the Fa that accepts L. Now the FA that accepts L2 must be PR. If we merge P and PR with a beta transition then this new FA accepts, but L and L2, therefore if L is regular, then there exists a language L2 such that L2 = { *uv* | *u*∈ LR , *v* ∈L }.

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.

(L2/L1 )R: = { *y* | *x*∈ L2 , *xy* ∈L1 } R

=

L1 R \L2 R = { *y* R | *x* R ∈ L2 R , *x* R *y* R ∈L1 R }

Since L1 andL2 are regular, L1 R and L2 R are also regular by definition. Now we have L1 R \L2 R is regular, so taking the reverse again we see,

(L1 R \L2 R ) R = L2 RR / L1 RR = L2/L1

The left quotient of any 2 regular languages are closed.

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

If L1 = L1L2/L2

Counterexample: Let L1={a, aa}, L2={b, c} . Then, L1 ∩ L2 ={a}, so L1  ={a,c}. L1 L2 ={ab, ac, aab, aac}, so L1 ∩L2= {ac, aab, aac}.

Therefore, in this case L1 =L1∩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. We can represent L as a DFA D.
2. 4.2-3 theorem in the book we can construct a NFA N that accepts LR.
3. We can convert N into an equivalent DFA, D. The algorithm for a palindrome language exist and is x= xR

Q9. [20] Pumping Lemma

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

Assume that L is regular so uphold all conditions of pumping lemma. Consider string w= aabbbcc, thus W&L. Let w =u v x with u =aa , v =bbb , x = cc.

Now examine condition3 of pumping lemma:

Let 1=0 so string u v i x = aa(bbb)cc = aa cc , which does not belong to L since it breaks k≥ n, since n =2 k=0.

This is a contradiction L is not a regular language.

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

Assume that Lis regular so uphold all conditions of pumping lemma.

Consider string w= *am* *bm* where m ≥ 0. Clearly w & L since na = nb = m

Examine condition 1 of pumping lemma:

For each I >=0, u v x e r when w = u v x

Since w= *am* *bm*, split a into *am-j* *a2j*, *bm* where j>0.

W now fits our desired form.

Let L= 2 , so *UV 2* X *am-j* (*aj*)*2*  *bm* = *am-j* *a2j*, *bm*.

If we let m =2 and j = 1, we get w = aaabb.

We see ha ≠ nb. This is a contradiction

L is not regular.

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

Claim L1 & L2 are not regular does imply that L1 ∪ L2 is non regular

Let L1 = {*anbm, n* ≥ m} and L2 = {*anbm, m* ≥ n}. We can see that L1, and L2 are both non regular.

However, L1 ∪ L2 = a\*b\*, and this is regular.

The claim is false, thus L1 ∪ L2 is not a regular language if L1 and L2  are not regular is disproven

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.

The given language can be rewritten as follows: min(L) = {w | w is in L, but no proper prefix of w is

1. L = {ending with ab}

U = a , v =6

So the language will be as follow

L = {ab, aab, aaab, bab, baab }

If we take u = a ∈ u ∉ L

If w = ab, we cannot take value u = ab as v become ∈

1. L = {take substring ending with aab}

L {aab,aaab,abaab,bbaab}

Take w=abb

u =a, v =ab (or) u =L , v = aab (or) u = aa, v = bab

From above 3 w = uv { u ∉ L}

{ u ∉ E}

From that we can say min(L) is regular