CSci 435: Formal Languages and Automata

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

**Home Assignment 3: 108/100 points + 25 points (optional)**

Q1. [8/10]

1. 5/10 Use the construction in Theorem 4.1 to find an DFA that accept L(*ab\*a*\*) ∩ L(*a\*b\*a*).

*Ans.*

First, we show the DFA’s for each of the sides of the intersection.

Diagram

Description automatically generated

A close up of a clock

Description automatically generated

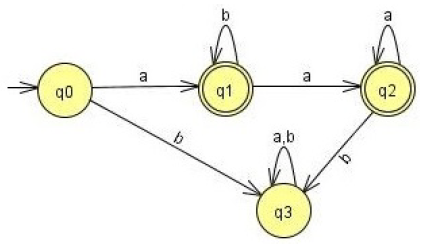
Then, we construct a DFA for the intersection of those two DFA’s from a combination of states and transition functions from each of the sides of the intersection.

A close up of a clock

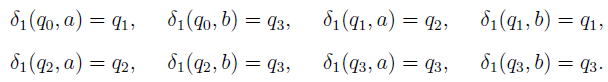
Description automatically generated

Let’s construct a DFA that accepts L(***ab\*a\**** ) and a DFA that accepts L(*a\*b\*a*) separately.

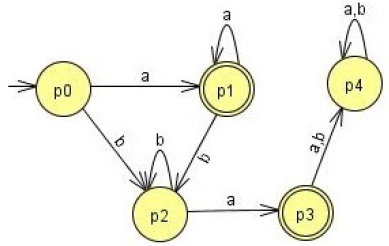
A DFA for L(***ab\*a*\*** ) is:



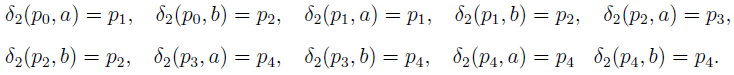
The transitions with final states *q1, q2* are



Similarly, a DFA for L(*a\*b\*a*) is:

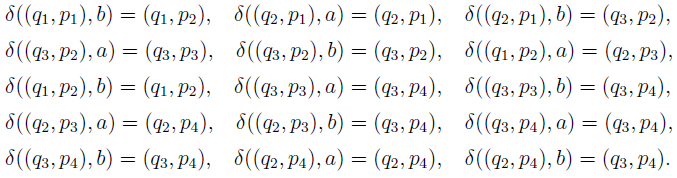


The transition with final state *p1*, *p3* are given by

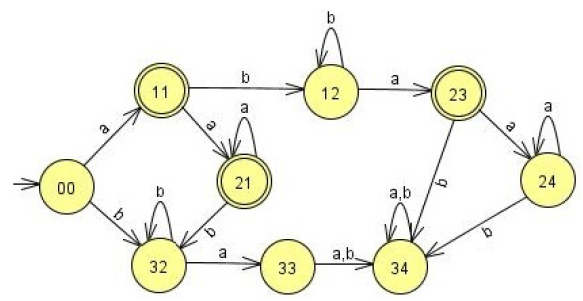


From this, we can get all the transitions for the accessible states as follows:



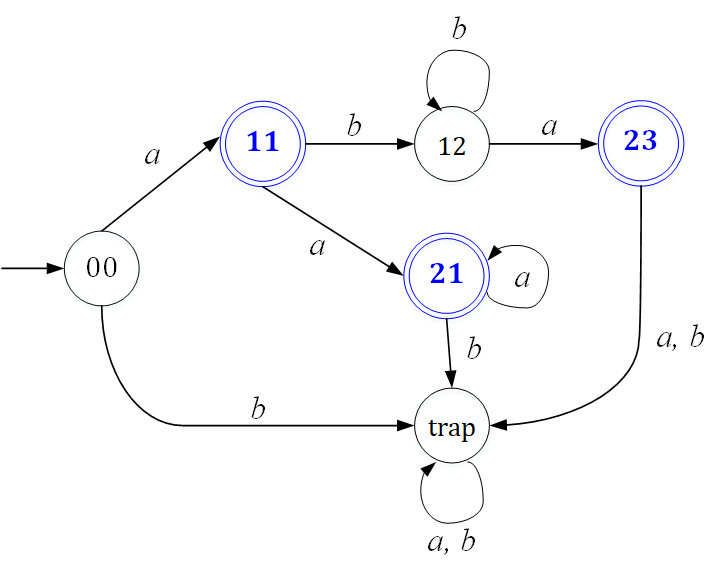


When this construction is complete, the final DFA is



In the above, transition from the states (00), (21) and (23) with b, b, {a, b} to a single trap state.

Then, the minimal DFA is:

****

Any DFA that accepts L(*a + aaa\* + abb\*a*) before minimization would be considered.

1. [3/5, Optional] Give the regular expression for the above language in 1) that is accepted by your DFA.

*Ans.* L(*ab\*a*).

L(*a + aaa\* + abb\*a*) = L(*aa*\* + *ab*\**a*)

Q2. [10/10] 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.***

*Ans.* cor(L1, L2) = {*w* | *w* ∉L1 or *w* ∉ L2, } = {*w* | *w* ∉L1} ∪ {*w* | *w* ∉L2}. Suppose L1 and L2 are regular. Let M1 = (Q1, Σ, δ1, q0, F1) and M2 = (Q2, Σ, δ2, p0, F2) be DFAs that accept L1 and L2, respectively. Then the complements of those DFAs, 1 = (Q1, Σ, δ1, q0, Q1 - F1) and 2 = (Q2, Σ, δ2, p0, Q2 – F2) accept 1 and 2. So, {*w* | *w*∈1} and {*w* | *w*∈2} are regular. Thus, the family of regular languages is closed under cor.

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.

*Ans*. Let h(L1 ∩ L2) =h(L1) ∩ h(L2) be an arbitrary homomorphism on L1 ∩ L2, where L1 and L2 are regular languages. Then, by Theorem 4.3, we know h(L1) and h(L2) are regular. Hence, we have an intersection of two regular languages. We can apply DeMorgan’s Law to show that the intersection is equal to the union and complementation. h(L1) ∩ h(L2) = . Since we know from slides 3-4 of Chapter 4 that regular languages are closed under union and complementation, this proves that h(L1) ∩ h(L2) is also a 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.

*Ans*. First we construct a DFA that accepts L1.

Diagram

Description automatically generated

Next, we check each of the states *q0* through *q3* to see if there is a path that follows *bab*\* to the final state. Since only *q0* works, then only that state can be a final state in the DFA M’ that represents the right quotient L1/L2.

A picture containing diagram

Description automatically generated

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

*Ans.* L1/L2 = L(*b*\*).

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

*Ans.* From the definition of L2, we have L2 = L­RL, where LR is the reversal of L. Since L is a regular language, then by Theorem 4.2, LR is also regular. Thus, L­RL is regular since the family of regular languages is closed under concatenation. Therefore, L2 is regular.

Q6. [10/10] 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.

*Ans.* By taking the reversal of the given left quotient, we have (L2/L1)R = {*y*R| *x*R∈, *y*R*x*R∈} =/. If L1 and L2 are regular, then by Theorem 4.2 and are also regular. We know that the right quotient with two regular languages is also a regular language by Theorem 4.4, so / = (L2/L1)R is a regular language. And by using Theorem 4.2 again we have that ((L2/L1)R)R = L2/L1 is a regular language. Therefore, the family of regular languages is closed under the left quotient with a regular language.

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

*Ans*. Let L1 = L(b\*) and L2 = L(bab\*). So we have two regular expressions r1 = b\* and r2 = bab\*. L1L2/L2 = L(b\*bab\*)/L(bab\*). To find the regular expression which denotes this right quotient, we begin by constructing a DFA that accepts L(b\*bab\*).

Diagram

Description automatically generated

Then we see which states can get to the final state by the path bab\*, which are both q0 and q1. So, the regular expression for the above right quotient is L(b\*bab\*), which does not equal L1.

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.

*Ans.* If we are given a regular language L, we know by Theorem 4.2 that LR is also regular. Then, we can use the equality algorithm in Theorem 4.7 which tests equality of any two regular languages to determine if L and LR are equal.

Q9. [30/20] Pumping Lemma

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

*Ans.* Assume that L is regular, so the pumping lemma must hold. Given an arbitrary positive integer *m*, we pick a string *w* such that > *m*. Let *w* = *ambm+1cm*∈L. We split the string *w* into substrings such that *w* = *xyz*, with the constraints ≤ *m*, and ≥ 1. So, we have *x* = *ar, y* = *as*, *z* = *am-r-sbm+1cm*, *r*+*s* ≤ *m*, *s* ≥ 1. Then, by pumping with *i* = 0, we have *w0* = *ar*(*as*)0*am-r-sbm+1cm* = *am-sbm+1cm* which is clearly not in L since there are a different amount of *a*’s and *c*’s. This contradicts the pumping lemma, which proves that L is not regular.

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

*Ans.* Assume that L is regular. Since regular languages are closed under complementation, we know that = {*w* | *na*(*w*) = *nb*(*w*)} must also be regular. Then, the pumping lemma must hold for . Given an arbitrary positive integer *m*, we pick a string *w* such that > *m*. Let *w* = *ambm* ∈L. We split the string *w* into substrings such that *w* = *xyz*, with the constraints ≤ *m*, and ≥ 1. So, we have *x* = *am-r, y* = *ar*, *z* = *bm*, *r* ≥ 1. Then, by pumping with *i* = 0, we have *w0* = *am-r*(*ar*)0*bm* = *am-rbm* which is clearly not in L since *m* – *r* ≠ *m*. This contradicts the pumping lemma, which proves that is not regular. Therefore, since regular languages are closed under complementation, L is also not regular.

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

*Ans.* This is false. Counterexample: Let L1 = {anbm, n≥ m} and L2 = {anbm, n≤ m}. Then L1 ∪ L2 = a\*b\*, which is a regular language.

Q10 [10/10, 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.

*Ans*. To show the strings that are not part of ***min***(L), we have LΣ+, since the strings in LΣ+ are of the form *w* = *xy* where *x* is in L. That is to say, those strings have a prefix string that is in L. To show ***min***(L) we subtract those strings from L. Thus, we have ***min***(L) = L - LΣ+. Since regular languages are closed under set difference, this proves that regular languages are also closed under the ***min*** operation.