COMP 3803 Assignment 1

October 8, 2021

Question 1.

Name: Braeden Hall
Student Number: 101143403

Question 2.

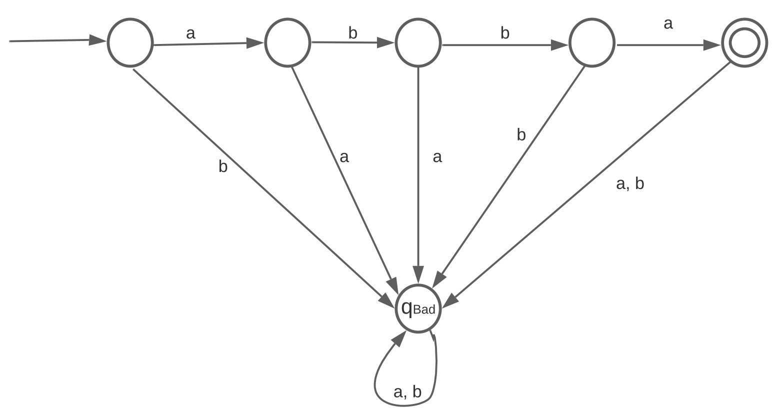
Define:

$$A = \{w : \text{w ends with } b\}$$

In the given DFA, any time you read a b you must be in the accept state. Thus all string that end in b will be accepted by the DFA. Furthermore, any time you read an a you are in the start state. Since the start state is not an accept state, any string that ends in an a will not be accepted. Therefore, A is the language of the given DFA.

Question 3.

1. DFA:

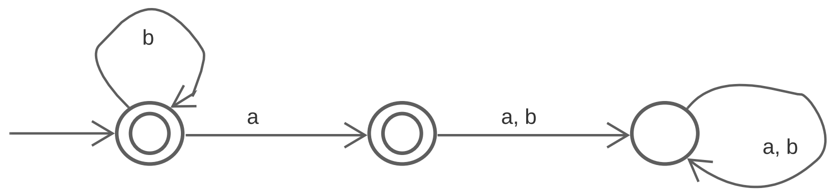


Correctness:

- The DFA can only process strings over the alphabet $\sum = \{a, b\}$
- If you read any string other than "abba" you will end up in q_{Bad}.
- If you read the string "abba" followed by any string of length ≥ 1 you will end up in q_{Bad}.
- If you end up in q_{Bad} there is no edge to leave that state.
- q_{Bad} is not an accept state.

Therefore, the given DFA accepts the language $\{abba\}$.

2. DFA:



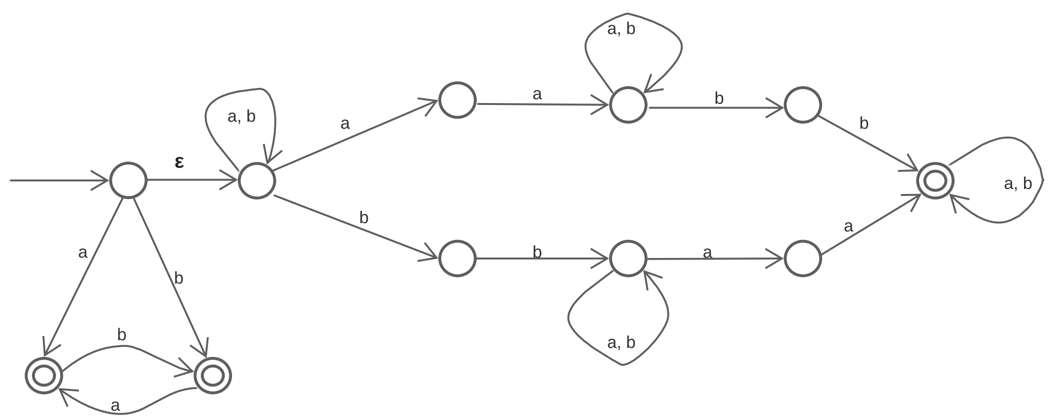
Correctness:

- The DFA can only process strings over the alphabet $\sum = \{a, b\}$
- Start state is an accept state
- While reading b's you stay in the start state
- If you read an a you transition to a different accept state
- If you are in the second accept state and you read any character then you end up in q_{Bad}
- If you end up in q_{Bad} there is no edge to leave that state.
- q_{Bad} is not an accept state.

Therefore, the given DFA accepts $\{w \in \{a, b\}^* : w$ does not contain aa and w does not contain $ab\}$.

Question 4.

NFA:



Question 5.

Define:

$$B = \{w : \text{length of } w \text{ is a multiple of } 3\}$$

If we assume, without loss of generality, that the alphabet of A and B is $\sum = \{0, 1\}$, then we see that the following regular expression (RegEx) accepts the language B:

$$\Big((0 \cup 1)(0 \cup 1)(0 \cup 1)\Big)^*$$

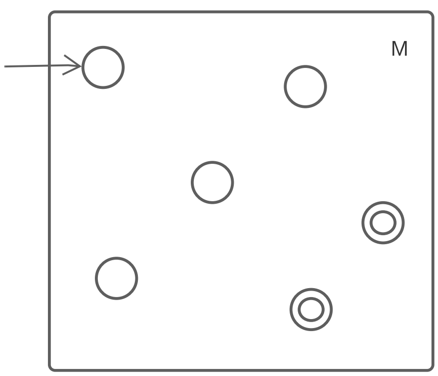
We have seen in class that it is possible to convert any RegEx into an NFA. We have seen that a language is regular if and only if there exists an NFA that accepts that language. Thus, B is a regular language.

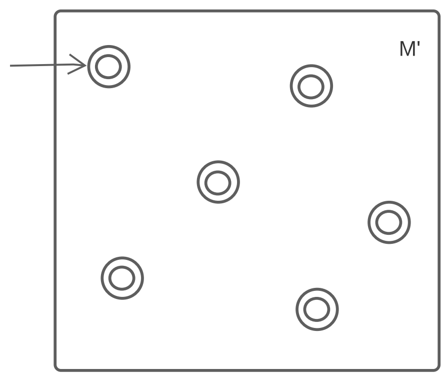
If we look at the definition of A_3 we see that elements of A_3 must be elements of A and their length must be a multiple of 3. In other words, $w \in A_3 \iff w \in A \cap B$. Therefore, $A_3 = A \cap B$

We have also seen that the intersection of 2 regular languages is a regular language. Therefore, $A \cap B$ is a regular language. Since $A_3 = A \cap B$ this means that A_3 is a regular language.

Question 6.

Let M be an example DFA that accepts the language A
Let M' be an example DFA that accepts the language A'





As the example shows, in order to construct a DFA that accepts A' we simply take the DFA that accepts A and turn all states into accept states. That is, you set the set of accept states (F') of M' equal to the set of states (Q) of A; all else remains the same ($Q' = Q$, $\delta' = \delta$ etc.). As an example, let us take some string $s \in A$. As it is in A it must be accepted by M. If we were to process the first half of s ($s_{\frac{n}{2}}$) we would end up in some state q_s of M that may or may not be an accept state. However, since $s_{\frac{n}{2}}$ is a prefix of s and $s \in A$ we want $s_{\frac{n}{2}} \in A'$. For that reason q_s must be an accept state of M'.

Since it possible for the prefix of some string in A to end at any state of M we must make all states accept states in M'. Therefore, because there exists a DFA that accepts the language A', A' is a regular language.

Question 7.

(x, y) is awesome.
Thus, $\exists z$ for which $xz \in A$ and $yz \notin A$ or $xz \notin A$ and $yz \in A$

Proof by contradiction:
Assume:

- $q_x = q_y$
- without loss of generality, that $xz \in A$ and $yz \notin A$

This means that, from the state q_x, we can read the string z and end up in an accept state of M. It means we can also start in state q_y and read z and end up in a state that is not an accept state. This means that from state $q_x = q_y$ we must be able to read z and simultaneously end up in an accept state and a state that is not an accept state. Obviously this is not possible. Thus, we have a contraction and $q_x \neq q_y$.

Question 8.

Assume that A is a regular language.
This means $\exists$finite automata M such that $L(M) = A$. Since M is a finite automata it must contain a *finite* set of states Q. Let us keep this in mind as we progress.

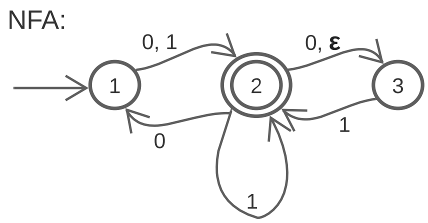
Let us use the same definition of an awesome pair from the previous question. We can observe that, for $n, m \in \mathbb{N}_0$ (non-negative integers), $n \neq m$ the pair (a^n, a^m) is awesome. The proof of this is trivial. We let the string $z = b^n$. Then we note that the string $a^n b^n \in A$, while $a^m b^n \notin A$. Therefore, the pair (a^n, a^m) is awesome.

Now, using the proof from the previous question, we can conclude that $q_{a^n} \neq q_{a^m}$. This means that q_{a^n} and q_{a^m} are 2 different states in the set Q of M.

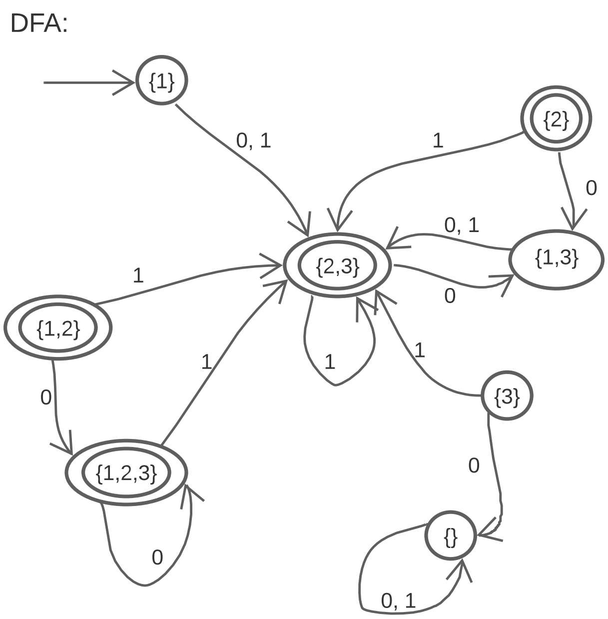
The only constraint on m is that it is a non-negative integer that is not equal to n. There are infinite numbers that fit that description, and each of them requires a state q_{a^m} in order for M to properly process and to reject strings not in A. This means that Q is an infinite set of states. This is a contradiction because by the definition of a finite automata, Q must be a finite set of states. Therefore, there is no finite automata M such that $L(M) = A$, which means that A is not a regular language.

Question 9.

NFA:



DFA with all states possible states listed:



DFA with only reachable states listed: