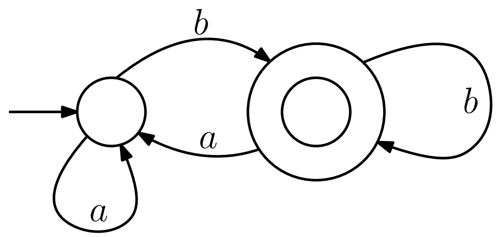
COMP 3803 — Solutions Assignment 1

Question 1: Write your name and student number.

Solution: Lionel Messi, 30

Question 2: What is the language of the following DFA? The alphabet is $\{a, b\}$. Justify your answer.



Solution: Denote the start state by q_a and the accept state by q_b. The following three claims frollow from the state diagram.

- No matter whether we are in the state q_a or q_b, if we read an a, we move to the state q_a.
- No matter whether we are in the state q_a or q_b, if we read a b, we move to the state q_b.
- To reach q_b, we must read at least one symbol (because q_b is not the start state).

This implies:

- We are in the state q_a if and only if we have read the empty string ϵ or the last symbol read is an a.
- We are in the state q_b if and only if the last symbol read is a b.

It follows that the language of the DFA is the set of all non-empty strings in $\{a, b\}^*$ that end with the symbol b.

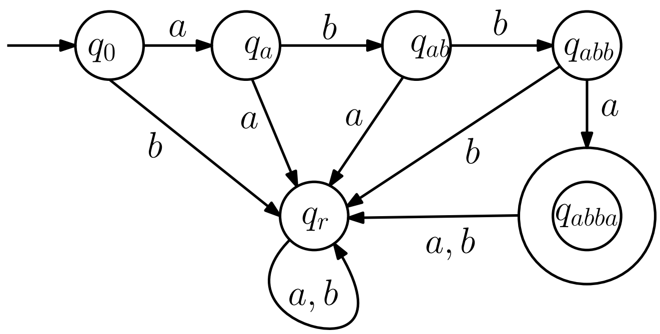
Question 3: For each of the following two languages, construct a DFA that accepts the language. In both cases, the alphabet is $\{a, b\}$. For each DFA, justify correctness.
(3.1) $\{abba\}$.
(3.2) $\{w \in \{a, b\}^* : \ w \text{ does not contain } aa \text{ and } w \text{ does not contain } ab\}$.

Solution: We start with the language $\{abba\}$. We need a DFA that accepts the string $abba$ and rejects all other strings. We use the following states

- q_0: We have not read any symbol.

- q_a: We have read exactly one symbol, namely a.
- q_{ab}: We have read exactly two symbols, namely ab.
- q_{abb}: We have read exactly three symbols, namely abb.
- q_{abba}: We have read exactly fours symbols, namely $abba$.
- q_r: The input string is non-empty and not equal to $abba$.

The state q_0 is the start state, whereas q_{abba} is the only accept state. The state diagram is given below.



Next we do

$$\{w \in \{a, b\}^* : \; w \text{ does not contain } aa \text{ and } w \text{ does not contain } ab\}.$$

After a bit of thought, we see that this language consists of the following strings:

- ϵ.
- a.
- b^n, for $n \geq 1$.
- $b^n a$, for $n \geq 1$.

We can shorten this to:

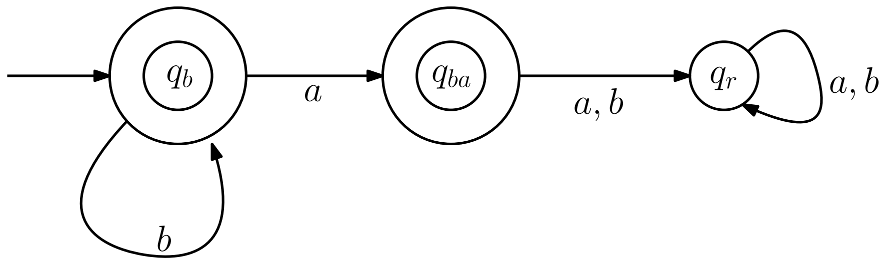
- b^n, for $n \geq 0$.
- $b^n a$, for $n \geq 0$.

We will use the following states:

- q_b: We have read b^n for some $n \geq 0$.
- q_{ba}: We have read $b^n a$ for some $n \geq 0$.

- q_r: The input string is not in the language.

The start state is q_b, whereas both q_b and q_{ba} are accept states. The state diagram is given below.



Question 4: Construct an NFA whose language is the set of all strings $w \in \{a, b\}^*$ such that

- w contains both aa and bb, or
- w does not contain aa and w does not contain bb.

Solution: We define the languages

$$A = \{w \in \{a, b\}^* : \ w \text{ contains both } aa \text{ and } bb\}$$

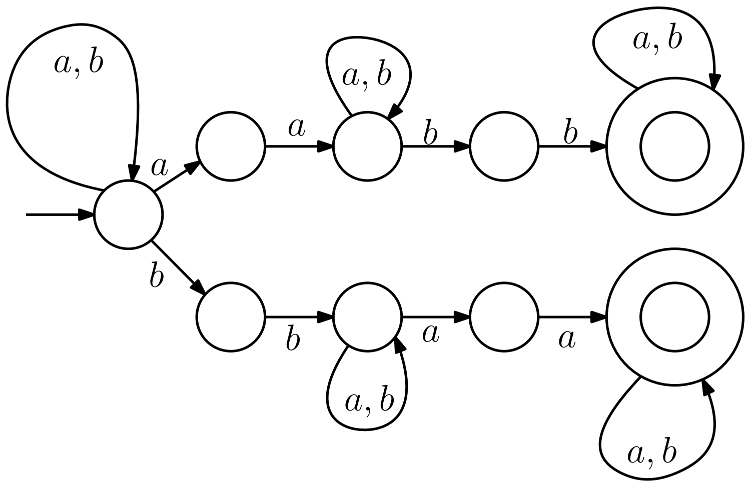
and

$$B = \{w \in \{a, b\}^* : \ w \text{ does not contain } aa \text{ and } w \text{ does not contain } bb\}$$

The question asks for an NFA that accepts the union of A and B. Below, we will construct an NFA M_A for A and a DFA M_B for B. By applying the union construction to M_A and M_B, we will obtain the NFA for $A \cup B$.

We start with the NFA M_A. Note that a string w belongs to A if and only if (i) the occurence of aa is to the left of the occurence of bb, or (ii) the occurence of aa is to the right of the occurence of bb.

Below you see the state diagram of an NFA that accepts M_A. The top branch is used to check for condition (i), whereas the bottom branch is used to check for condition (ii). Observe that for every string of type (i), there exists a path from the start state to the upper accept state. Similarly, for every string of type (ii), there exists a path from the start state to the lower accept state. Finally, the only way to reach an accept state is when the string is of type (i) or (ii). Thus, this NFA accepts the language M_A.



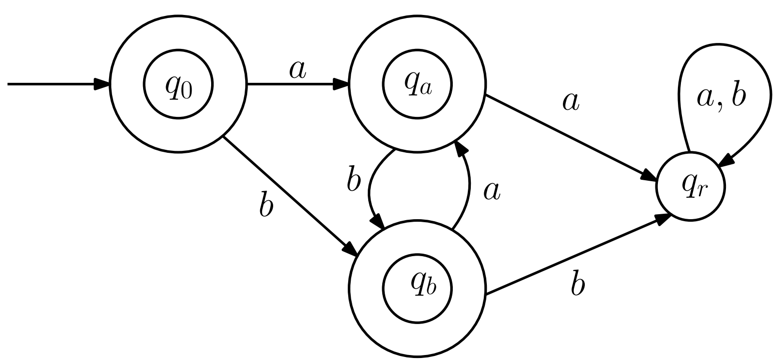
Next, we construct the DFA M_B. Observe that B consists of the following strings:

- ϵ.
- the length is at least one, each non-final[1] a is followed by b, and each non-final b is followed by a. We call such strings *good*.

We will use the following states:

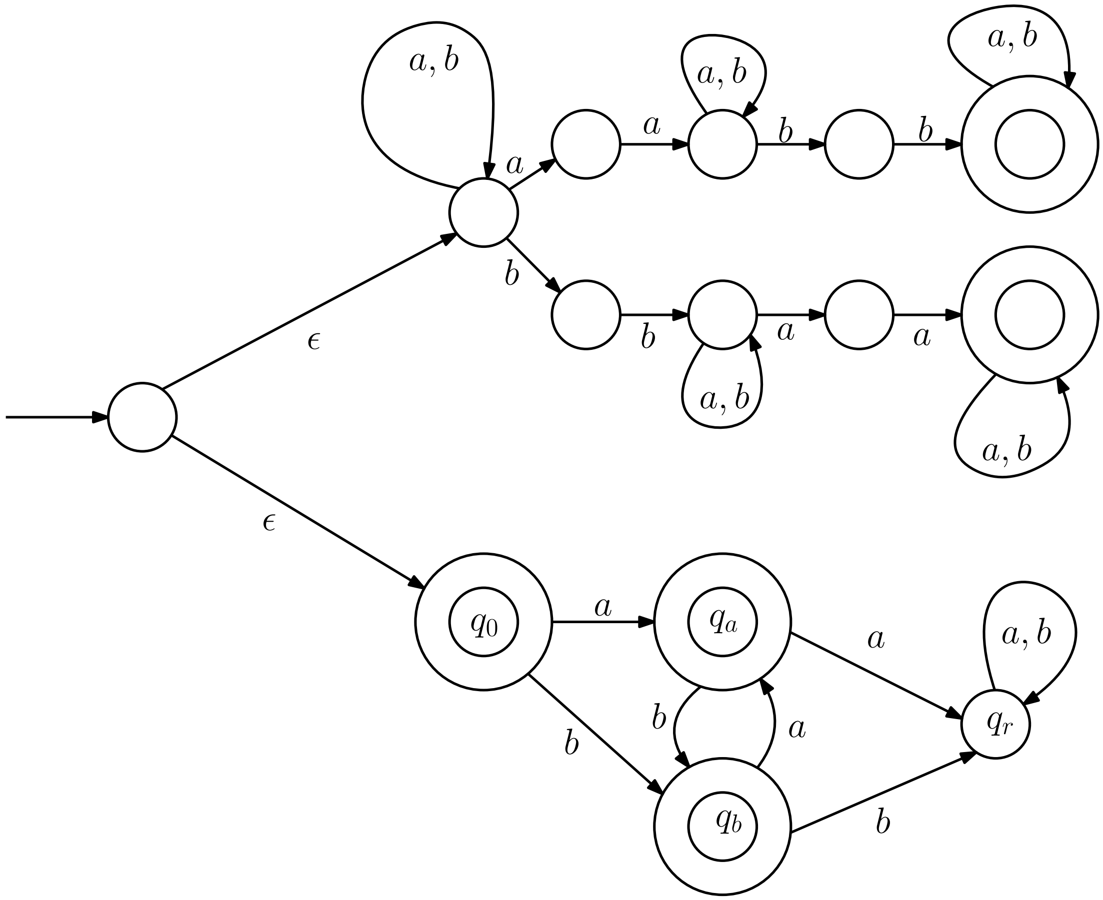
- q_0: No symbol has been read.
- q_a: The string read so far is good and the last symbol read is a.
- q_b: The string read so far is good and the last symbol read is b.
- q_r: The input string contains aa or bb.

The start state is q_0, the accept states are q_0, q_a, and q_b. The state diagram of M_B is given below.



[1] i.e., not the rightmost symbol

To obtain the final NFA, we apply the union construction to M_A and M_B:



Question 5: Let A be an arbitrary regular language over the alphabet Σ. We define the language

$$A_3 = \{w \in A : \text{ the length of } w \text{ is a multiple of three}\}.$$

(Note that zero is a multiple of three.)

Prove that A_3 is a regular language. You may use any result that was proven in class.

Solution: Let B_3 be the language

$$B_3 = \{w \in \Sigma^* : \text{ the length of } w \text{ is a multiple of three}\}.$$

Then,

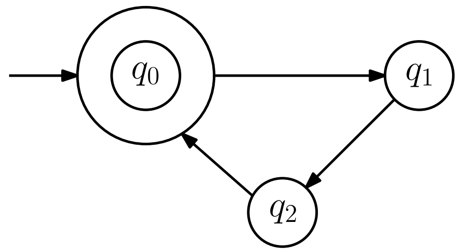
$$A_3 = A \cap B_3.$$

We are given that A is regular. If we can show that B_3 is regular, then it follows that A_3 is regular as well, because we have seen in class that the intersection of two regular languages is regular.

To prove that B_3 is regular, we construct a DFA with three states:

- q_0: the number of symbols read so far, modulo 3, is equal to 0,
- q_1: the number of symbols read so far, modulo 3, is equal to 1,
- q_2: the number of symbols read so far, modulo 3, is equal to 2.

The state q_0 is the start state and it is the only accept state. The state diagram is given below. Every transition (i.e., directed edge) is labeled with all symbols in the alphabet Σ. For example, if $\Sigma = \{a, b, c\}$, then every transition is labeled with a, b, c.



Question 6: Let A be an arbitrary regular language over the alphabet Σ. We define the language

$$A' = \{u \in \Sigma^* : \text{ there exists a string } v \in \Sigma^* \text{ such that } uv \in A\}.$$

(In words, A' consists of all prefixes of strings in A.)

Prove that A' is a regular language.

Hint: Let M be a DFA that accepts A. How would you change the state diagram of M such that the resulting state diagram is a DFA that accepts A'?

Solution: Since we are given that A is a regular language, there is a DFA M that accepts A. Let w be an arbitrary string in A. Then, M accepts w: When M starts in the start state and reads the string w, it follows a uniquely defined path in the state diagram (because M is deterministic). Every state on this path corresponds to a prefix of w, i.e., a prefix u such that $w = uv$ for some string $v \in \Sigma^*$. Each such prefix is in the language A' and, thus, must be accepted by the DFA that we are trying to define. We obtain this DFA by turning all states on the path into an accept state.

In other words, if $M = (Q, \Sigma, \delta, q_0, F)$ is the DFA that accepts A, then the following DFA $M' = (Q, \Sigma, \delta, q_0, F')$ accepts A':

- A state q of Q is an accept state in F' if and only if the state diagram of M contains a path from q to some accept state of F.
- Note that M' has the same set of states, the same transition function, and the same start state as M.

Question 7: Let A be an arbitrary regular language over the alphabet Σ. For two strings x and y in Σ^*, we say that the pair (x, y) is *awesome*, if there exists a string z in Σ^* such that (i) $xz \in A$ and $yz \notin A$ or (ii) $xz \notin A$ and $yz \in A$.

Let M be a DFA that accepts the language A, let (x, y) be an awesome pair, let q_x be the state that M is in after having read x, and let q_y be the state that M is in after having read y.

Prove that $q_x \neq q_y$.

Solution: The proof is by contradiction. Thus, we assume that $q_x = q_y$.

Let z be an arbitrary string in Σ^*. Let q_{xz} be the state that M is in after having read xz, and let q_{yz} be the state that M is in after having read yz.

We first show that $q_{xz} = q_{yz}$.

- Consider what happens when the DFA M reads the string xz: It starts in the start state. After having read x, it is in the state q_x. When it reads z (starting in q_x), M follows a uniquely defined path, say P, in the state diagram (because M is deterministic). The DFA ends in the state q_{xz}.
- Consider what happens when the DFA M reads the string yz: It starts in the start state. After having read y, it is in the state q_y, which is equal to q_x (by our assumption). When it reads z (starting in $q_y = q_x$), M follows the same path P as above, (again, because M is deterministic). Therefore, the DFA ends in the state q_{xz}. As a result, $q_{xz} = q_{yz}$.

We have shown the following: For every string z in Σ^*, whether M reads xz or yz, it ends in the same state. This means that either both xz and yz are in A, or both xz and yz are not in A. This contradicts the fact that the pair (x, y) is awesome.

Question 8: Use the previous question to prove that the language

$$A = \{a^n b^n : n \geq 0\}$$

is not regular.

Hint: Use a proof by contradiction. If n and m are distinct nonnegative integers, is the pair (a^n, a^m) awesome? You are not allowed to use the Pumping Lemma.

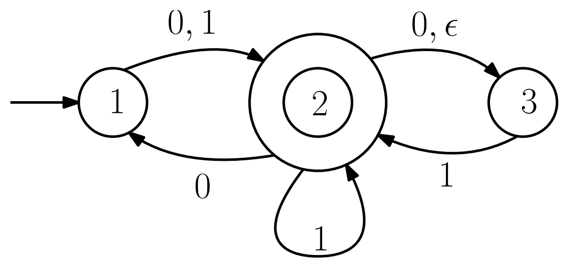
Solution: Let n and m be two distinct nonnegative integers. We first prove that the pair (a^n, a^m) is awesome. To prove this, we need a string z in $\{a, b\}^*$ such that (i) $a^n z \in A$ and $a^m z \notin A$ or (ii) $a^n z \notin A$ and $a^m z \in A$. It is clear that we can take $z = b^n$; we can also take $z = b^m$.

To show that A is not regular, we use a proof by contradiction. Thus, we assume that A is a regular language. Let M a DFA that accepts A.

For each integer $n \geq 0$, let q_n be the state that M is in after having read a^n. From the previous question, for all $n \geq 0$, $m \geq 0$, $n \neq m$, the states q_n and q_m are different.

It follows that all states $q_0, q_1, q_2, \ldots$ are pairwise distinct. Thus, the DFA M has an infinite number of states. This is a contradiction.

Question 9: Use the construction given in class to convert the following NFA to an equivalent DFA.



Solution: In the original assignment, I did not give names to the states. As you can see, above, I have added the names 1, 2, and 3.

Each state of the DFA corresponds to a subset of $\{1, 2, 3\}$. Thus, the DFA has the following eight states:

$$\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}\{1, 2, 3\}.$$

The accept states of the DFA are all states that contain the accept state 2 of the NFA. Thus, the DFA has the following four accept states:

$$\{2\}, \{1, 2\}, \{2, 3\}\{1, 2, 3\}.$$

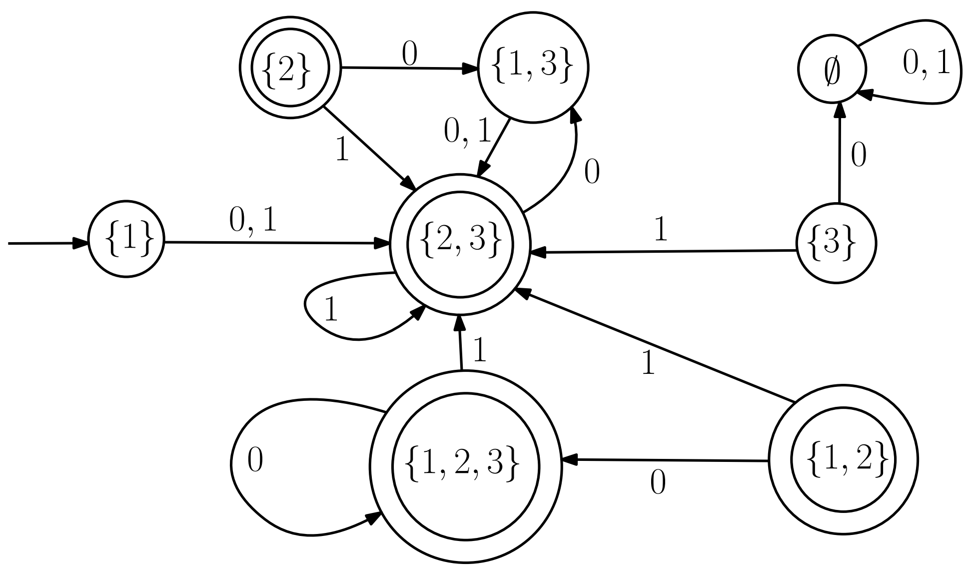
What is the start state of the DFA: We take the start state 1 of the NFA and add to it all states that can be reached by making zero or more ϵ-transitions. Since there are no ϵ-transitions leaving state 1, the start state of the DFA is $\{1\}$.

Now we need the transitions of the DFA:

- State $\emptyset$, read 0 or 1: stay in state $\emptyset$.
- State $\{1\}$, read 0: In the NFA, start in state 1, then read 0 followed by zero or more ϵ-transitions. In this way, we can reach the states 2 and 3.
- State $\{1\}$, read 1: In the NFA, start in state 1, then read 1 followed by zero or more ϵ-transitions. In this way, we can reach the states 2 and 3.
- State $\{2\}$, read 0: In the NFA, start in state 2, then read 0 followed by zero or more ϵ-transitions. In this way, we can reach the states 2 and 3.
- State $\{2\}$, read 1: In the NFA, start in state 2, then read 1 followed by zero or more ϵ-transitions. In this way, we can reach the states 2 and 3.
- State $\{3\}$, read 0: In the NFA, start in state 3, then read 0 followed by zero or more ϵ-transitions. In this way, we cannot reach any state.
- State $\{3\}$, read 1: In the NFA, start in state 3, then read 1 followed by zero or more ϵ-transitions. In this way, we can reach the states 2 and 3.
- State $\{1, 2\}$, read 0: Take the union of the DFA transitions for $\{1\}$ and $\{2\}$, which is $\{1, 2, 3\}$.

- State $\{1, 2\}$, read 1: Take the union of the DFA transitions for $\{1\}$ and $\{2\}$, which is $\{2, 3\}$.
- State $\{1, 3\}$, read 0: Take the union of the DFA transitions for $\{1\}$ and $\{3\}$, which is $\{2, 3\}$.
- State $\{1, 3\}$, read 1: Take the union of the DFA transitions for $\{1\}$ and $\{3\}$, which is $\{2, 3\}$.
- State $\{2, 3\}$, read 0: Take the union of the DFA transitions for $\{2\}$ and $\{3\}$, which is $\{1, 3\}$.
- State $\{2, 3\}$, read 1: Take the union of the DFA transitions for $\{2\}$ and $\{3\}$, which is $\{2, 3\}$.
- State $\{1, 2, 3\}$, read 0: Take the union of the DFA transitions for $\{1\}$, $\{2\}$, and $\{3\}$, which is $\{1, 2, 3\}$.
- State $\{1, 2, 3\}$, read 1: Take the union of the DFA transitions for $\{1\}$, $\{2\}$, and $\{3\}$, which is $\{2, 3\}$.

Here is the state diagram of the DFA:



We can simplify this:

- The state $\emptyset$ cannot be reached from the start state $\{1\}$.
- The state $\{2\}$ cannot be reached from the start state $\{1\}$.
- The state $\{3\}$ cannot be reached from the start state $\{1\}$.

- The state $\{1, 2\}$ cannot be reached from the start state $\{1\}$.
- The state $\{1, 2, 3\}$ cannot be reached from the start state $\{1\}$.

This leads to the final DFA: