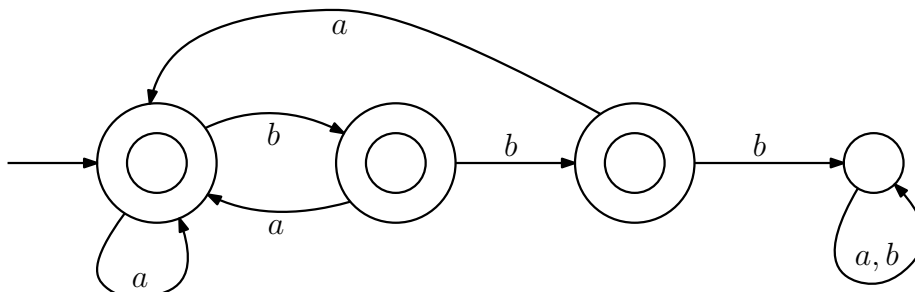


COMP 3803 — Fall 2025 — Solutions Problem Set 1

Question 1: What is the language of the following DFA? The alphabet is $\{a, b\}$. As always, justify your answer.



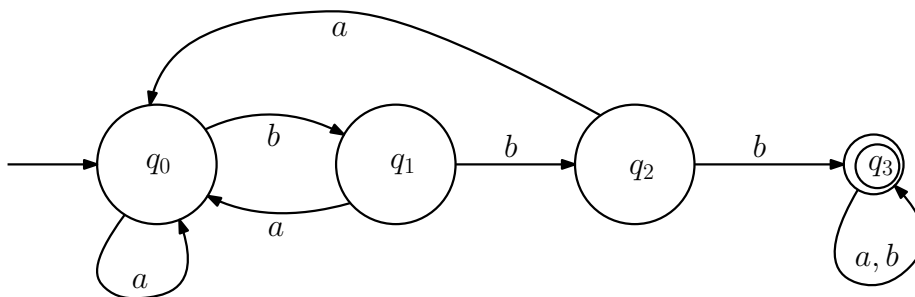
Solution: Staring at the state diagram for a few hours will convince you that the answer is

$$\{w \in \{a, b\}^* : w \text{ does not contain } bbb\}.$$

There are of course different ways to prove this. The DFA below is obtained by “flipping” the states. If we can show that this new DFA accepts the language

$$\{w \in \{a, b\}^* : w \text{ contains } bbb\},$$

then we are done.



Here are the meanings of the states:

- q_0 : We have not seen bbb . The last symbol read was a or we have not read any symbol.
- q_1 : We have not seen bbb . The last two symbols read were ab or we have read only one symbol which was b .
- q_2 : We have not seen bbb . The last two symbols read were bb .
- q_3 : We have seen bbb . We stay in this state forever.

Question 2: Construct a DFA that accepts the language

$$A = \{a u a a v a : u \text{ and } v \text{ are in } \{a, b\}^*\}.$$

The alphabet is $\{a, b\}$. As always, justify your answer.

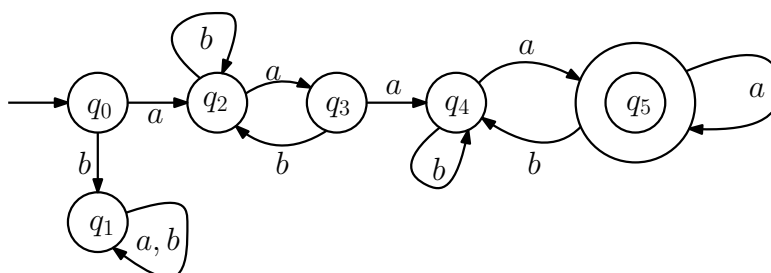
Solution: The DFA will check the following:

- The string starts with a .
- After the leftmost symbol (which is a), the string contains aa . It will do this by searching for the leftmost occurrence of aa (after the leftmost symbol).
- The substring after the leftmost occurrence of aa ends with a .

We will use six states:

- q_0 : Nothing has been read.
- q_1 : The leftmost symbol is b .
- q_2 : The leftmost symbol is a . After the leftmost symbol, we have not read aa . The last symbol read was b or the leftmost a .
- q_3 : The leftmost symbol is a . After the leftmost symbol, we have not read aa . The last symbol read was a (which is not the leftmost a).
- q_4 : The leftmost symbol is a . After the leftmost symbol, we have read the first occurrence of aa . The last two symbols read were this first occurrence of aa or the last symbol read was b .
- q_5 : The leftmost symbol is a . After the leftmost symbol, we have read the first occurrence of aa . The last symbol read is a (which is not part of the first occurrence of aa).

This leads to the following state diagram:



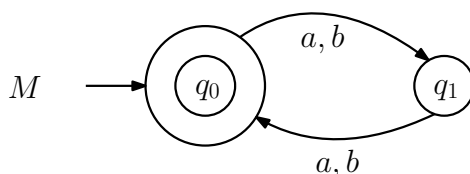
Question 3: Let A be a regular language over the alphabet $\Sigma = \{a, b\}$, and assume that the empty string ε is contained in A . Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts A .

Show how to modify M to obtain a DFA that accepts the language $A \setminus \{\varepsilon\}$. As always, justify your answer.

Solution: Since M is a DFA and $\varepsilon \in A$, the start state q_0 of M must be an accept state.

I will refer to M' as the DFA we are aiming for. Thus, M' will accept all strings in A , except for the empty string ε .

As a first idea, we could let M' be a copy of M with the start state q_0 being a non-accept state. This does not work as the example below shows. The DFA M accepts all strings having an even length. If we make the start state a non-accept state, then the resulting DFA does not accept any string.

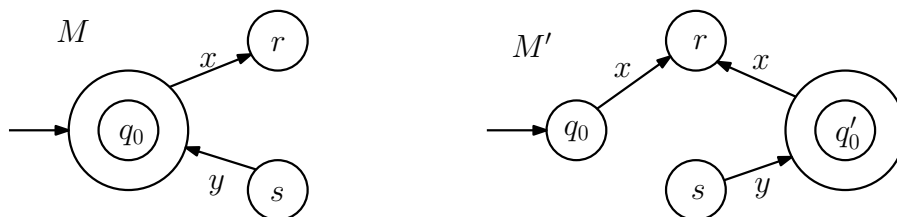


The trick is to make the start state q_0 a non-accept state, while introducing a new state, say q'_0 , which will be an accept state. In this way, the empty string is not accepted. We have to guarantee the following, for every walk in M that starts in its start state q_0 :

- Every visit of q_0 in M (except the first one) will be “rerouted” through the new accept state q'_0 in M' .

Here is how we obtain M' (refer to the beautiful figure):

- Make the start state q_0 a non-accept state, while keeping it being the start state in M' .
- Introduce a new state q'_0 and make it an accept state.
- For every $x \in \Sigma$, keep the transition $q_0 \xrightarrow{x} r$ and add the transition $q'_0 \xrightarrow{x} r$.
- For every $y \in \Sigma$, remove the transition $s \xrightarrow{y} q_0$ and add the transition $s \xrightarrow{y} q'_0$.
- All other transitions are the same as those in M .



Convince yourself of the following:

- In M' , after leaving the start state q_0 , we can never return to it.
- Imagine we run M and M' in parallel. Each time M returns to the start state q_0 , the new DFA M' returns to the new accept state q'_0 .

Question 4: In class, we have seen the following three operations:

- If A and B are languages over the same alphabet, then their *union* is the language

$$A \cup B = \{w : w \in A \text{ or } w \in B\}.$$

- If A and B are languages over the same alphabet, then their *concatenation* is the language

$$AB = \{uv : u \in A \text{ and } v \in B\}.$$

- If A is a language, then the *star* of A is the language

$$A^* = \{u_1 u_2 \dots u_k : k \geq 0 \text{ and } u_i \in A \text{ for all } i = 1, 2, \dots, k\}.$$

Professor Justin Bieber claims that the following two facts imply that the star of a regular language is regular:

Fact 1: The union of two regular languages is regular.

Fact 2: The concatenation of two regular languages is regular.

Professor Bieber's Proof: Let A be an arbitrary regular language. We define the languages A^k , for $k \geq 0$, recursively as follows:

- $A^0 = \{\varepsilon\}$.
- For $k \geq 1$, $A^k = AA^{k-1}$, i.e., A^k is the concatenation of A and A^{k-1} .

It is obvious that A^0 is regular, because there exists a DFA that only accepts the empty string ε . From this, a straightforward induction proof, that uses **Fact 2** in the induction step, implies that A^k is regular for every $k \geq 0$.

Again, using a straightforward induction proof, that uses **Fact 1** in the induction step, we see that for every integer $k \geq 0$,

$$A^0 \cup A^1 \cup A^2 \cup \dots \cup A^k \tag{1}$$

is regular. Since (1) holds for all integers $k \geq 0$, we conclude that A^* is regular. QED

Is Professor Bieber's Proof correct? As always, justify your answer.

Solution: Justin's proof of (1) is correct. He correctly proves that for every $k \geq 0$, there exists a DFA that accepts the language

$$A^0 \cup A^1 \cup A^2 \cup \dots \cup A^k.$$

Note that this DFA depends on the integer k . In particular, for each k , we get a different DFA.

To prove that A^* is regular, we have to prove that there exists a DFA M such that for every integer $k \geq 0$, M accepts

$$A^0 \cup A^1 \cup A^2 \cup \dots \cup A^k.$$

In other words, we need one single DFA M that works for all k ; M must be independent of k .

To summarize, Justin has proved

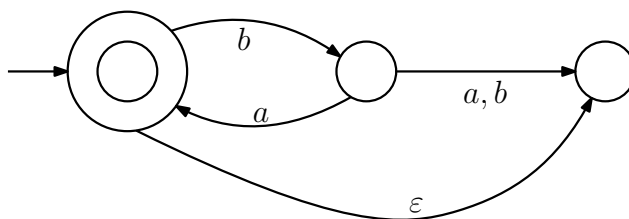
$$\forall k \exists \text{ DFA } M : M \text{ accepts } A^0 \cup A^1 \cup A^2 \cup \dots \cup A^k.$$

He should have proved

$$\exists \text{ DFA } M \forall k : M \text{ accepts } A^0 \cup A^1 \cup A^2 \cup \dots \cup A^k.$$

I am sure that you remember from COMP 1805, that $\forall \exists$ is not logically equivalent to $\exists \forall$. Sorry Justin, no marks for this question.

Question 5: What is the language of the following NFA? The alphabet is $\{a, b\}$. As always, justify your answer.



Solution: If you stare at the state diagram long enough, then you will see that the language is $\{ba\}^*$, which is the same as

$$\{(ba)^n : n \geq 0\}.$$

Here is the justification:

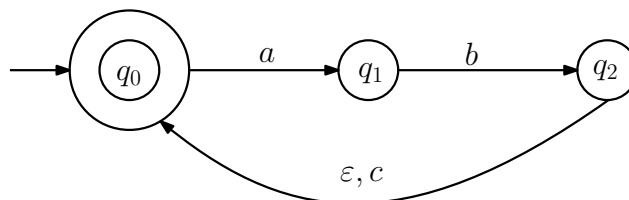
- For any $n \geq 0$, the string $(ba)^n$ is accepted: Loop n times from the leftmost state to the middle state and back to the leftmost state.
- The only way to be in the accept state (after the entire string has been read) is by doing this looping.

Question 6: Construct an NFA with three states that accepts the language

$$A = \{ab, abc\}^*.$$

The alphabet is $\{a, b, c\}$. As always, justify your answer.

Solution: Here we go:



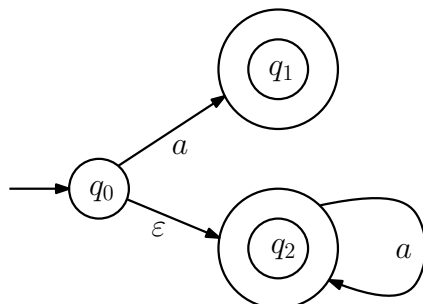
- The empty string ε is accepted.
- The string ab is accepted.
- The string abc is accepted.
- Any string that consists of a finite concatenation of strings ab and abc is accepted. The only way to be in the accept state (after the entire string has been read) is when the input string is such a finite concatenation.

Question 7: Construct an NFA M with alphabet $\{a\}$ that has the following two properties:

- M accepts the language $\{a\}^*$.
- There is a transition in the state diagram of M , such that the language of the NFA obtained by removing this transition is equal to $\{a\}$.

As always, justify your answer.

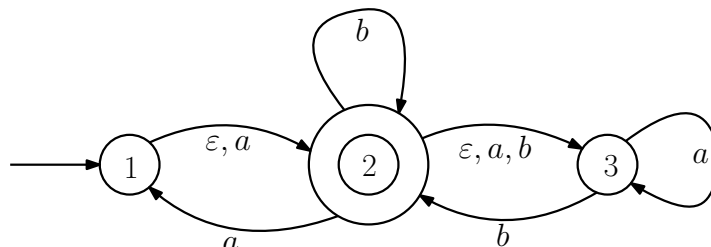
Solution: Here we go:



For every $n \geq 0$, this NFA accepts the string a^n : From the start state, follow the ε -transition to state q_2 . Then read a^n .

If we remove the ε -transition, then the resulting NFA only accepts the string a .

Question 8: Use the construction given in class to convert the following NFA (with alphabet $\{a, b\}$) to an equivalent DFA.



Show the full state diagram of the DFA; it has $2^3 = 8$ states. Afterwards, simplify the diagram by removing states that cannot be reached from the start state (in case this is possible).

Solution: Each state of the DFA corresponds to a subset of $\{1, 2, 3\}$. Thus, the DFA has $2^3 = 8$ states. The accept states of the DFA are all states that contain the accept state 2 of the NFA. Thus, the DFA has $2^2 = 4$ accept states.

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 following zero or more ε -transitions. Note that state 2 can be reached by following one ε -transition from state 1. Also, state 3 can be reached by following two ε -transitions from state 1. Thus, the start state of the DFA is $\{1, 2, 3\}$.

Now we need the transitions of the DFA. Remember, to simulate one step of the NFA, the DFA reads one symbol (a or b) and then follows zero or more ε -transitions.

- State \emptyset , read a or b : stay in state \emptyset .
- State $\{1\}$, read a : In the NFA, read a and go to state 2. Then, from state 2, follow zero or more ε -transitions. This takes us to state 2 or state 3. Thus, the DFA goes to state $\{2, 3\}$.
- State $\{1\}$, read b : The NFA is stuck in this case. Thus, the DFA goes to state \emptyset .
- State $\{2\}$, read a : In the NFA, read a and go to state 1 or 3. From state 1, follow zero or more ε -transitions, which takes us to state 1, 2, or 3. There are no ε -transitions leaving state 3. Thus, the DFA goes to state $\{1, 2, 3\}$.
- State $\{2\}$, read b : In the NFA, read b and stay in state 2 or go to state 3. From state 2, an ε -transition takes us to state 3. There are no ε -transitions leaving state 3. Thus, the DFA goes to state $\{2, 3\}$.

- State $\{3\}$, read a : In the NFA, read a and stay in state 3. There are no ε -transitions leaving state 3. Thus, the DFA stays in state $\{3\}$.
- State $\{3\}$, read b : In the NFA, read b and go to state 2. Then, from state 2, follow zero or more ε -transitions. This takes us to state 2 or state 3. Thus, the DFA goes to state $\{2, 3\}$.
- State $\{1, 2\}$, read a : Take the union of the DFA transitions for $\{1\}$ and $\{2\}$, which is $\{1, 2, 3\}$.
- State $\{1, 2\}$, read b : Take the union of the DFA transitions for $\{1\}$ and $\{2\}$, which is $\{2, 3\}$.
- State $\{1, 3\}$, read a : Take the union of the DFA transitions for $\{1\}$ and $\{3\}$, which is $\{2, 3\}$.
- State $\{1, 3\}$, read b : Take the union of the DFA transitions for $\{1\}$ and $\{3\}$, which is $\{2, 3\}$.
- State $\{2, 3\}$, read a : Take the union of the DFA transitions for $\{2\}$ and $\{3\}$, which is $\{1, 2, 3\}$.
- State $\{2, 3\}$, read b : Take the union of the DFA transitions for $\{2\}$ and $\{3\}$, which is $\{2, 3\}$.
- State $\{1, 2, 3\}$, read a : Take the union of the DFA transitions for $\{1\}$, $\{2\}$, and $\{3\}$, which is $\{1, 2, 3\}$.
- State $\{1, 2, 3\}$, read b : Take the union of the DFA transitions for $\{1\}$, $\{2\}$, and $\{3\}$, which is $\{2, 3\}$.

Here is the state diagram of the DFA:

