

COMP 3803 — Solutions Assignment 1

Question 1: For each of the following languages, construct a DFA that accepts the language. In all cases, the alphabet is $\{0, 1\}$. For each DFA, justify correctness.

(1.1) $\{w \in \{0, 1\}^* : w \text{ starts with 1 and ends with 0}\}$.

(1.2) $\{w \in \{0, 1\}^* : \text{every odd position in } w \text{ is 1}\}$. The positions are numbered $1, 2, 3, \dots$

(1.3) $\{w \in \{0, 1\}^* : w \text{ has length at least 3 and its third symbol is 0}\}$.

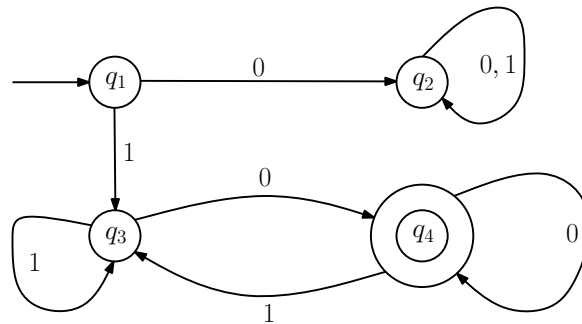
(1.4) $\{\epsilon, 0\}$.

Solution: We start with

$$\{w : w \text{ starts with 1 and ends with 0}\}.$$

We use the following states:

1. q_1 : start state; we are in this state if no symbol has been read.
2. q_2 : we are in this state if the first symbol is a 0 (in which case we will stay in this state forever, because the string must be rejected).
3. q_3 : we are in this state if the first symbol is a 1 and the last symbol that was read is also a 1.
4. q_4 : we are in this state if the first symbol is a 1 and the last symbol that was read is a 0. This is the only accept state.



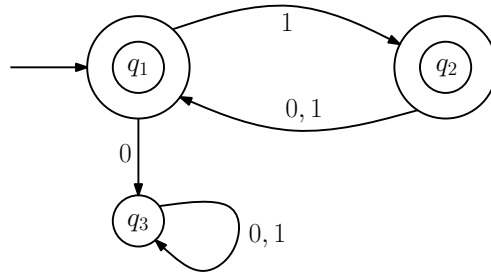
Next we do

$$\{w : \text{every odd position in } w \text{ is 1}\}.$$

Note that the empty string ϵ is in this language. We use the following states:

1. q_1 : we have read an even number of symbols and every odd position read so far was a 1. This is an accept state. It is also the start state, because the empty string satisfies this condition.

2. q_2 : we have read an odd number of symbols and every odd position read so far was a 1. This is an accept state.
3. q_3 : we are in this state if we have read a 0 at some odd position (since the input string must be rejected, we will stay in this state forever).

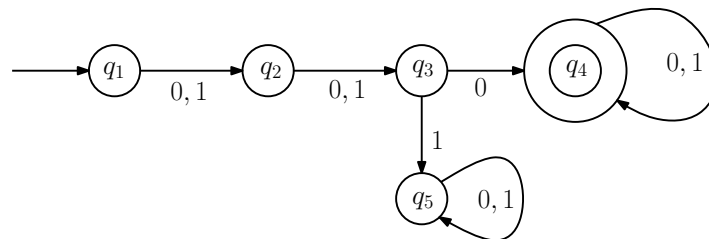


Next we do

$$\{w : w \text{ has length at least 3 and its third symbol is } 0\}.$$

We use the following states:

1. q_1 : start state; we are in this state if no symbol has been read.
2. q_2 : we are in this state if exactly one symbol has been read.
3. q_3 : we are in this state if exactly two symbols have been read.
4. q_4 : we are in this state if at least three symbols have been read and the third symbol was a 0. This is the accept state.
5. q_5 : we are in this state if at least three symbols have been read and the third symbol was a 1.



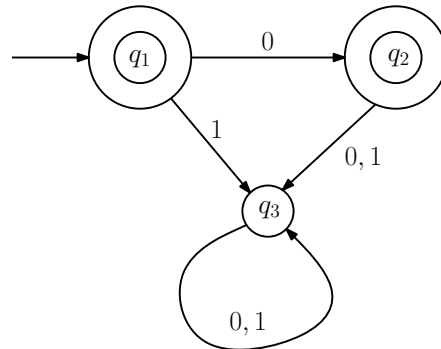
Finally we do

$$\{\epsilon, 0\}.$$

We use the following states:

1. q_1 : start state; we are in this state if no symbol has been read. This is an accept state, because the empty string ϵ must be accepted.

2. q_2 : we are in this state if the first symbol is a 0 and we have read exactly one symbol. This is an accept state.
3. q_3 : we are in this state if
 - (a) the first symbol is a 1 or
 - (b) the first symbol is a 0 and we have read at least two symbols.



Question 2: For each of the following languages, construct an NFA that accepts the language. In all cases, the alphabet is $\{0, 1\}$. For each NFA, justify correctness.

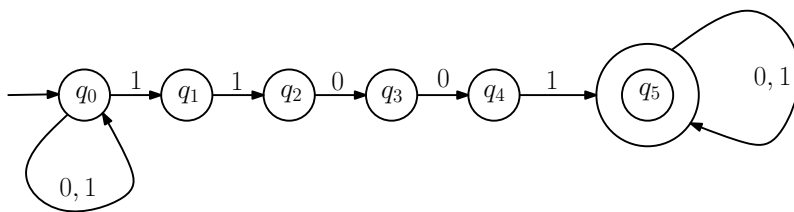
- (2.1) $\{w : w \text{ contains the substring } 11001\}$.
- (2.2) $\{w : w \text{ has length at least } 2 \text{ and does not end with } 10\}$.
- (2.3) $\{w : w \text{ begins with } 1 \text{ or ends with } 0\}$.

Solution: First we do

$$\{w : w \text{ contains the substring } 11001\}.$$

We will use the following states:

1. q_0 : start state; we are in this state if we did not yet reach the first symbol of the substring 11001.
2. q_1 : We have read the first bit of the substring 11001.
3. q_2 : We have read the second bit of the substring 11001.
4. q_3 : We have read the third bit of the substring 11001.
5. q_4 : We have read the fourth bit of the substring 11001.
6. q_5 : We have seen the substring 11001. This is the only accept state.



Next we do

$$\{w : w \text{ has length at least 2 and does not end with } 10\}.$$

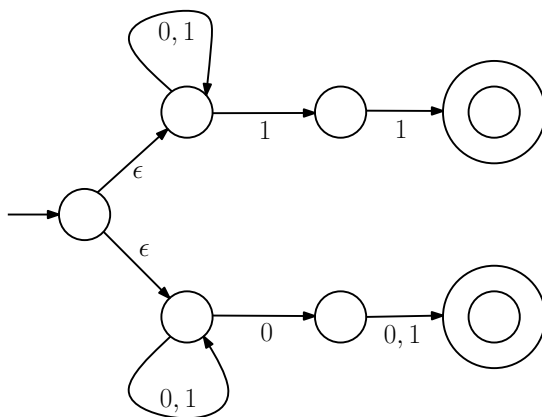
We can write this language as the union of

$$A = \{w : w \text{ has length at least 2 and ends with } 11\}$$

and

$$B = \{w : w \text{ has length at least 2 and ends with } 00 \text{ or } 01\}.$$

Thus, we can create an NFA for A and an NFA for B , and then apply the union-construction.



Finally, we do

$$\{w : w \text{ begins with } 1 \text{ or ends with } 0\}.$$

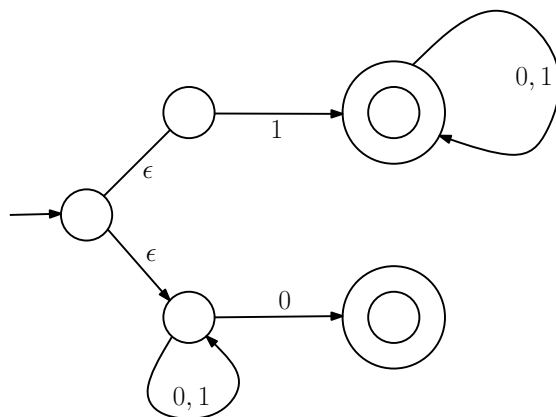
This language is the union of

$$A = \{w : w \text{ begins with } 1\}$$

and

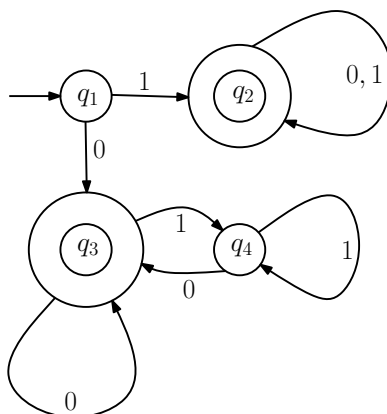
$$B = \{w : w \text{ ends with } 0\}.$$

Thus, we can create an NFA for A and an NFA for B , and then apply the union-construction.



By the way, here is a DFA for the same language. It uses the following states:

1. q_1 : start state; we did not read anything yet.
2. q_2 : the first symbol is a 1.
3. q_3 : the first symbol is a 0 and the last symbol read was 0.
4. q_4 : the first symbol is a 0 and the last symbol read was 1.



Question 3: Let A be a language over the alphabet $\Sigma = \{0, 1\}$, and let \bar{A} be the *complement* of A . Thus, \bar{A} is the language consisting of all binary strings that are not in A .

Assume that A is a regular language. Let $M = (Q, \Sigma, \delta, q, F)$ be a nondeterministic finite automaton (NFA) that accepts A .

Consider the NFA $N = (Q, \Sigma, \delta, q, \bar{F})$, where $\bar{F} = Q \setminus F$ is the complement of F . Thus, N is obtained from M by turning all accept states into non-accept states, and turning all non-accept states into accept states.

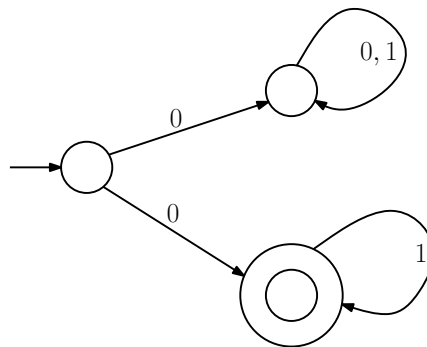
(3.1) Is it true that the language accepted by N is equal to \bar{A} ? Justify your answer.

(3.2) Assume now that M is a deterministic finite automaton (DFA) that accepts A . Define N as above; thus, turn all accept states into non-accept states, and turn all non-accept states

into accept states. Is it true that the language accepted by N is equal to \bar{A} ? Justify your answer.

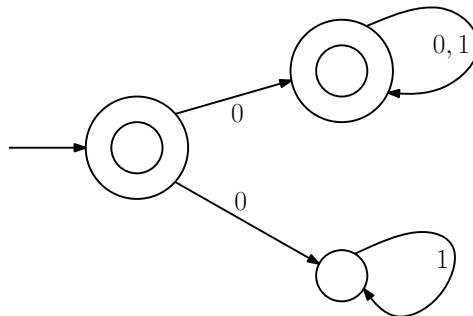
Solution: For the first part, the answer to the question is “no”. Let us see why: The NFA M accepts the language A . This means that a string s is in A if and only if there exists at least one path in the state diagram of M that takes us from the start state to an accept state. It may happen, however, that for some string s in A , there is also a path in the state diagram that takes us from the start state to a non-accept state. Such a string s will be accepted by the NFA N . In other words, it may happen that the NFA N accepts strings that are in A .

Here is an example. Assume that $A = \{01^n : n \geq 0\}$. The following NFA M accepts A :



Consider the string $s = 0111$. This string belongs to A and is accepted by M : When reading the first bit, which is 0, follow the edge from the start state to the accept state; then read the next three bits, which are all 1, and stay in the accept state.

The NFA N looks as follows:



This NFA also accepts the string $s = 0111$: When reading the first bit, which is 0, follow the edge from the start state to the top-right accept state; then read the next three bits, which are all 1, and stay in the top-right accept state.

Thus, both NFA's accept the string $s = 0111$. Since $s \in A$, it follows that the NFA N accepts strings that belong to A and, therefore, do not belong to \bar{A} . This means that $L(N) \neq \bar{A}$.

Here is an even easier example: Assume that $A = \{\epsilon\}$. Then \bar{A} is the set of all non-empty binary strings.

The following NFA M accepts the language A : There is one state, which is both the start state and an accept state. There are no transitions in the state diagram (i.e., there are no arrows at all).

The corresponding NFA N has one state, which is the start state and a non-accept state. Since there is no accept state, the language accepted by N is \emptyset , which is not equal to \bar{A} .

For the second part, the answer is “yes”:

- For every string s in A , the *unique* path in the state diagram of M takes us from the start state to an accept state.
- For every string s not in A , the *unique* path in the state diagram of M takes us from the start state to a non-accept state.

Since N is obtained from M by “flipping” the states, we have:

- For every string s in A , the *unique* path in the state diagram of N takes us from the start state to a non-accept state.
- For every string s not in A , the *unique* path in the state diagram of N takes us from the start state to an accept state.

This means that the DFA N accepts \bar{A} .

Question 4: Let A and B be two regular languages over the same alphabet Σ . Prove that the difference of A and B , i.e., the language

$$A \setminus B = \{w : w \in A \text{ and } w \notin B\}$$

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

First Solution: We modify the proof of Theorem 2.3.1 (we have seen this in class):

Since A and B are regular languages, there are DFA's $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ that accept A and B , respectively. In order to prove that $A \setminus B$ is regular, we have to construct a finite automaton M that accepts $A \setminus B$. In other words, M must have the property that for every string $w \in \Sigma^*$,

$$M \text{ accepts } w \Leftrightarrow M_1 \text{ accepts } w \text{ and } M_2 \text{ rejects } w.$$

As we did for $A \cup B$ in class, the idea is to run M_1 and M_2 simultaneously. We define the set Q of states of M to be the Cartesian product $Q_1 \times Q_2$. If M is in state (r_1, r_2) , this means that

- if M_1 would have read the input string up to this point, then it would be in state r_1 , and

- if M_2 would have read the input string up to this point, then it would be in state r_2 .

This leads to the finite automaton $M = (Q, \Sigma, \delta, q, F)$, where

- $Q = Q_1 \times Q_2 = \{(r_1, r_2) : r_1 \in Q_1 \text{ and } r_2 \in Q_2\}$. Observe that $|Q| = |Q_1| \times |Q_2|$, which is finite.
- Σ is the alphabet of A and B (recall that we assume that A and B are languages over the same alphabet).
- The start state q of M is equal to $q = (q_1, q_2)$.
- The set F of accept states of M is given by

$$F = \{(r_1, r_2) : r_1 \in F_1 \text{ and } r_2 \notin F_2\} = F_1 \times (Q_2 \setminus F_2).$$

- The transition function $\delta : Q \times \Sigma \rightarrow Q$ is given by

$$\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a)),$$

for all $r_1 \in Q_1$, $r_2 \in Q_2$, and $a \in \Sigma$.

Second Solution:

- We first observe that $A \setminus B = A \cap \overline{B}$.
- De Morgan tells us that

$$\overline{A \cap \overline{B}} = \overline{A} \cup B.$$

- By taking complements, we get

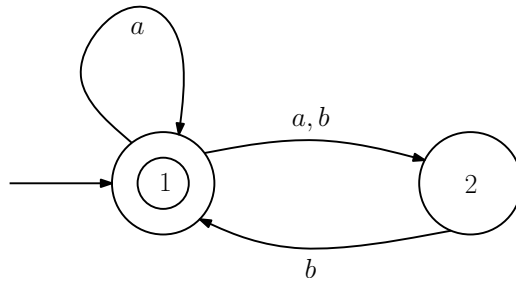
$$A \cap \overline{B} = \overline{\overline{A} \cup B}.$$

- Thus, we have

$$A \setminus B = \overline{\overline{A} \cup B}.$$

- Since A is regular, it follows from Question 3 that \overline{A} is regular.
- Since \overline{A} and B are regular, their union $\overline{A} \cup B$ is regular (we have seen this in class).
- Since $\overline{A} \cup B$ is regular, it follows from Question 3 that its complement is regular. But this complement is equal to $A \setminus B$.

Question 5: Use the construction given in class to convert the following NFA to an equivalent DFA.



Solution: Since there are no ϵ -transitions, this is the easy case of the construction. Following the construction given in class, the DFA has the following four states:

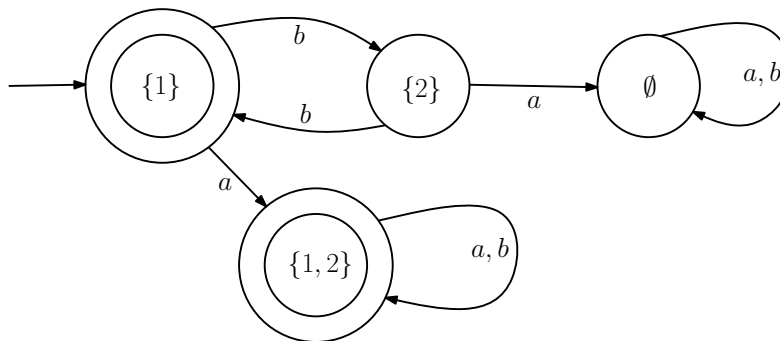
$$\emptyset, \{1\}, \{2\}, \{1, 2\}.$$

The start state of the DFA is the state $\{1\}$.

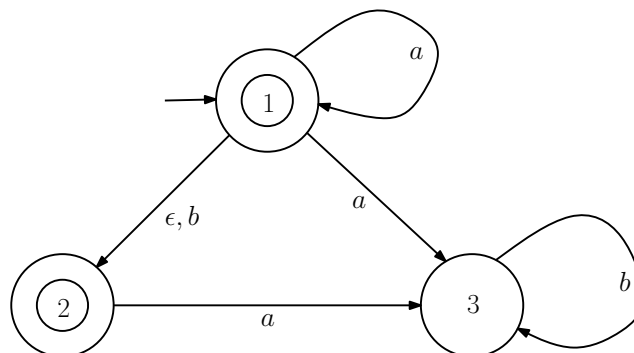
The set of accept states of the DFA consists of all states of the DFA that contain at least one accept state of the NFA. That is, the set of accept states of the DFA consists of all states of the DFA that contain 1. This gives the following two accept states:

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

The transition function of the DFA is specified in the following state diagram:



Question 6: Use the construction given in class to convert the following NFA to an equivalent DFA.



Solution: Following the construction given in class, the DFA has the following eight states:

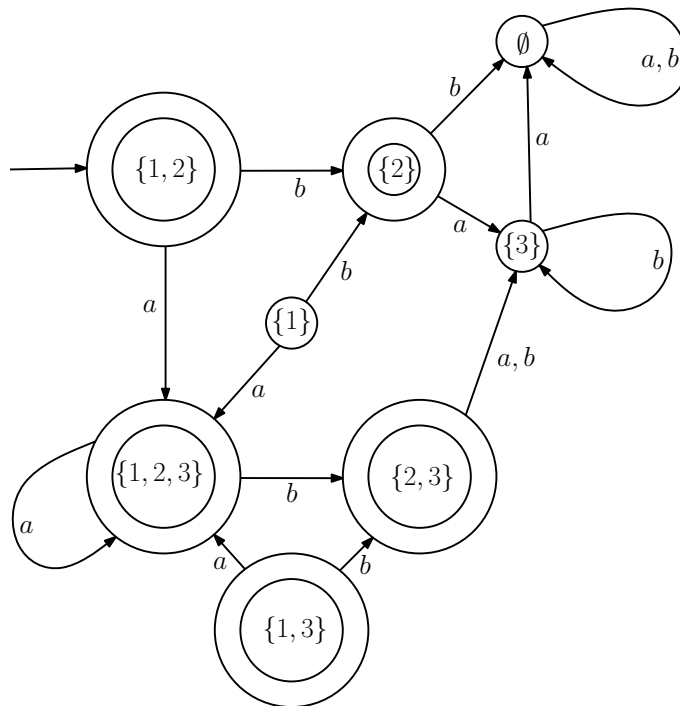
$$\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}.$$

The start state of the DFA is the set of all states in the NFA that can be reached from its start state by state by making zero or more ϵ -transitions. Thus, the start state is $\{1, 2\}$.

The set of accept states of the DFA consists of all states of the DFA that contain at least one accept state of the NFA. That is, the set of accept states of the DFA consists of all states of the DFA that contain 1 or 2. This gives the following six accept states:

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

The transition function of the DFA is specified in the following state diagram:



Even though this is a correct solution, we can simplify this DFA, because the states $\{1\}$ and $\{1, 3\}$ cannot be reached from the start state. Thus, we can just remove these two states. This gives the following DFA:

