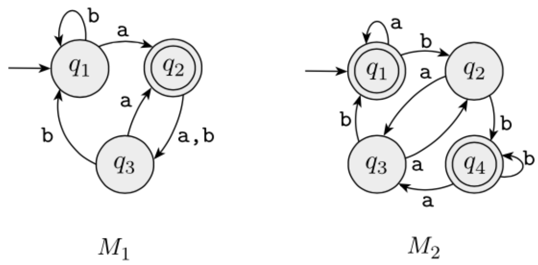
Exercise I, Theory of Computation 2025

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 (*Exercise 1.1 in Sipser's book*) The following are the state diagrams of two DFAs, M_1 and M_2.



Answer the following questions about each of these machines.

1a What is the start state?

1b What is the set of accepting states?

1c What sequence of states does the machine go through on input aabb?

1d Does the machine accept the string aabb?

1e Does the machine accept the string ε?

Solution:

	M_1	M_2
1a	q_1	q_1
1b	$\{q_2\}$	$\{q_1, q_4\}$
1c	q_1, q_2, q_3, q_1, q_1	q_1, q_1, q_1, q_2, q_4
1d	No	Yes
1e	No	Yes

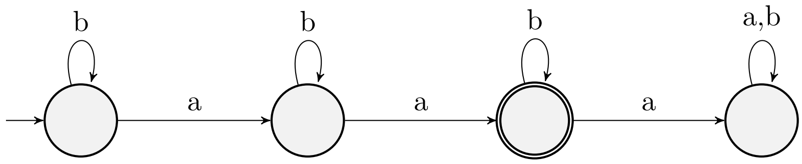
2 (*Based on Exercise 1.4 in Sipser's book*) Each of the following languages is the intersection of two simpler languages. In each part, construct DFAs for the simpler languages, then combine them using the construction discussed in class to give the state diagram of a DFA for the language given. In all parts, $\Sigma = \{a, b\}$.

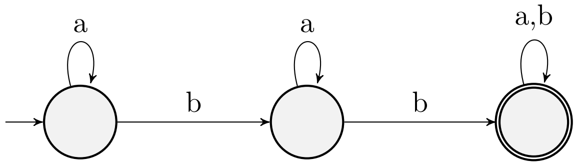
2a $\{w \mid w$ has exactly two a's and at least two b's$\}$

2b $\{w \mid w$ has an even number of a's and each a is immediately followed by at least one b$\}$

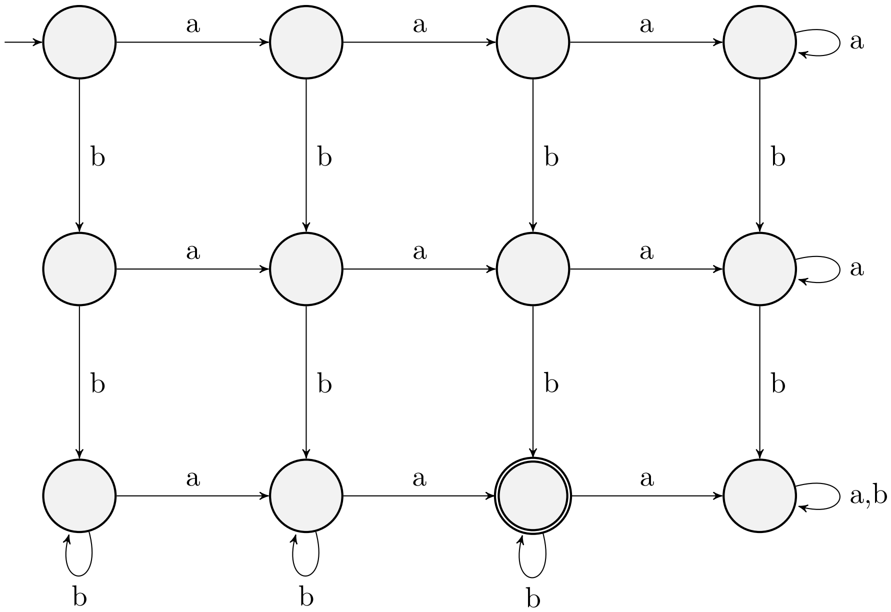
Solution:

2a The following are DFAs for the two languages $\{w \mid w$ has exactly two a's$\}$ and $\{w \mid w$ has at least two b's$\}$, respectively.



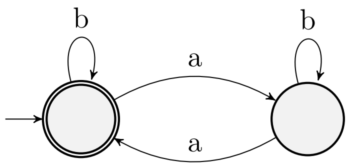


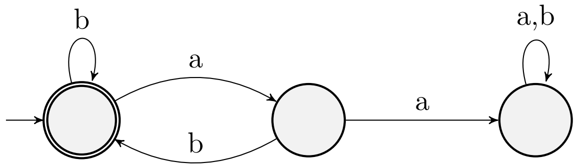
Combining them using the intersection construction gives the following DFA.



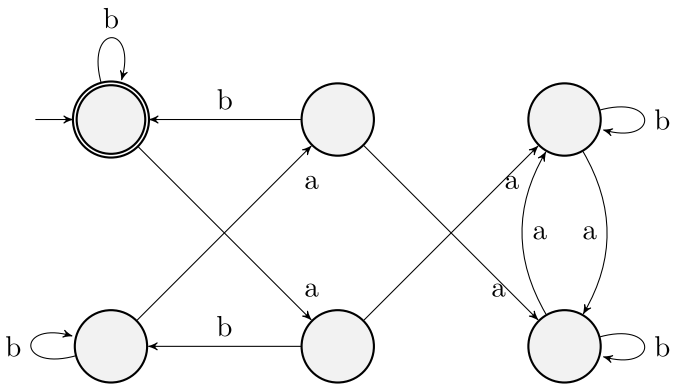
Though the problem doesn't ask you to simplify the DFA, the three states on the very right can be combined into a single one, as they all act as "dead states" from which one can never reach an accepting state.

2b These are DFAs for the two languages $\{w \mid w \text{ has an even number of a's}\}$ and $\{w \mid \text{each a in } w \text{ is immediately followed by at least one b}\}$.





Combining them using the intersection construction gives the following DFA.



Though the problem doesn't ask you to simplify the DFA, the two states on the right can be combined into a single one, as they both act as "dead states" from which one can never reach an accepting state.

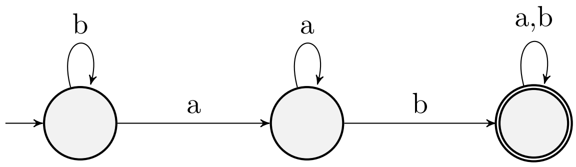
3 (*Based on Exercise 1.5 in Sipser's book*) Each of the following languages is the complement of a simpler language. In each part, construct a DFA for the simpler language, then use it to give the state diagram of a DFA for the language given. In all parts, $\Sigma = \{a, b\}$.

3a $\{w \mid w$ does not contain the substring ab$\}$

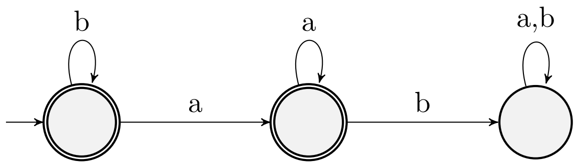
3b $\{w \mid w$ does not contain the substring baba$\}$

Solution:

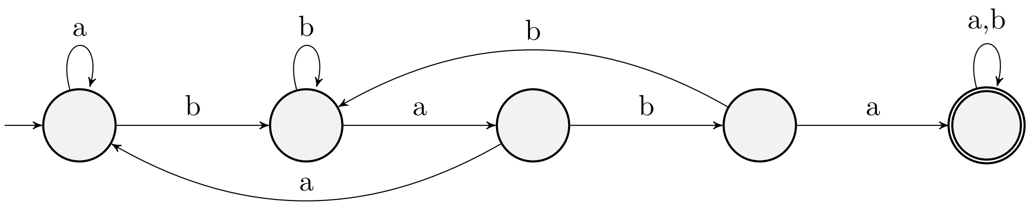
3a The DFA below recognizes the language $\{w \mid w$ contains ab$\}$.



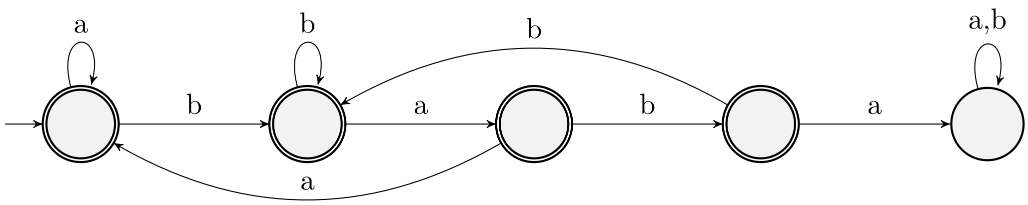
The DFA below recognizes its complement i.e. $\{\ w \mid w$ does not contain ab $\}$.



3b The DFA below recognizes the language $\{w \mid w$ contains baba$\}$.



The DFA below recognizes its complement i.e. $\{w \mid w$ does not contain baba$\}$.



4 Suppose A_1, A_2, and A_3 are regular languages over the alphabet Σ. Prove that $(A_1 \cup A_2) \cap A_3$ is regular by giving a formal description of a finite automaton recognizing it.

Solution: We know that A_1, A_2, and A_3 are regular languages. Hence, there are finite automata that recognize them. Let $M_1 = (Q_1,\ \Sigma\ ,\delta_1,\ q_1,\ F_1)$ such that $L(M_1) = A_1$. In other words, let M_1 be the automaton that recognizes A_1. Define M_2 and M_3 similarly for A_2 and A_3.

We first construct $M_{12} = (Q_{12},\ \Sigma\ ,\delta_{12},\ q_{12},\ F_{12})$ where

- $Q_{12} = Q_1 \times Q_2$,
- $\delta_{12}((s_1, s_2), a) = (\delta_1(s_1, a), \delta_2(s_2, a))$,
- $q_{12} = (q_1, q_2)$,
- $F_{12} = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.

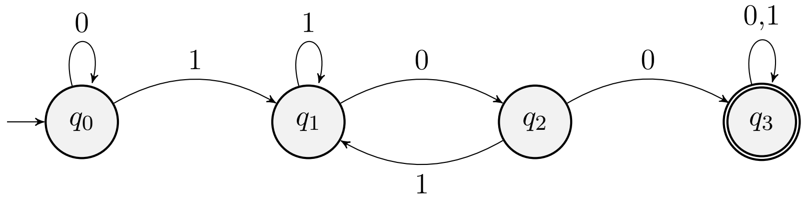
Observe that M_{12} is the machine such that $L(M_{12}) = A_1 \cup A_2$.

We then construct $M_{123} = (Q_{123},\ \Sigma\ ,\delta_{123},\ q_{123},\ F_{123})$ where

- $Q_{123} = Q_{12} \times Q_3$,
- $\delta_{123}((s_1, s_2), a) = (\delta_{12}(s_1, a), \delta_3(s_2, a))$,
- $q_{123} = (q_{12}, q_3)$,
- $F_{123} = (F_{12} \times F_3)$.

Observe $L(M_{123}) = L(M_{12}) \cap A_3 = (A_1 \cup A_2) \cap A_3$ as required.

5 For the automaton given below, describe the language it recognizes. Prove that your description is correct.



Solution: Denote the given automaton by M. The language recognized by M is

$$L = \{w \mid w \text{ has "100" as a substring}\}\ .$$

We provide two different proofs of this below.

First solution (ad-hoc): We show that the automaton stops in the only accepting state q_3, if and only if the input string has "100" as a substring. We prove the two directions separately.

First, consider any input $s \in L$. Clearly, we can write $s = x100y$ for some possible empty words x, y. According to the illustration of the automaton, we know that starting from any state q, reading "100" will lead to state q_3. Indeed, the four different paths when reading "100" are: $(q_0, q_1, q_2, q_3), (q_1, q_1, q_2, q_3), (q_2, q_1, q_2, q_3)$ and (q_3, q_3, q_3, q_3). Therefore, no matter what state q the automaton is in after reading x, after reading "100" we are in state q_3. Since no paths lead out of state q_3, the automaton will stay in q_3 until the entire string is read, and thus accept, just as desired.

For the converse, suppose that the automaton accepts an input s. Then, there exists some $i \in \mathbb{N}$ such that after reading s_i, the i-th symbol of s, the automaton is in state q_3 for the first time. Moreover, since the shortest path from the initial state to state q_3 is of length 3, we must have $i \geq 3$. Since this is the first time we reach q_3, we must be in state q_2 after reading s_{i-1}, according to the graph. Therefore, we must have that $s_i = 0$. But the only way to get t state q_2 is from q_1, thus we deduce that $s_{i-1} = 0$ also. Finally, since all incoming arrows at state q_1 are labelled 1, we must have read the symbol 1 to get there. We conclude that therefore $s_{i-2} = 1$ and consequently s contains"100" as a substring, as desired.

Second solution (induction): We proceed by induction, following the framework discussed in class.

Claim. Let x be an arbitrary input to the automaton. We consider four cases:

1. If x does not contain any 1's, the automaton M terminates in state q_0.
2. If the last symbol of x is a "1", but x does not contain "100" as a substring, then M terminates in state q_1.
3. If the last two symbols of x are "10", but x does not contain "100" as a substring, then M terminates in state q_2.
4. If x contains "100" as a substring, then M terminates in state q_3.

Note that these four cases are mutually exclusive and together cover all possible inputs. Also, note that it suffices to prove this claim since it implies that the automaton terminates in the only accepting state q_3 if and only if the input contains "100" as a substring.

We now prove the above claim by induction on the length of the input x:

Base case. If x has length 0 (i.e. x is the empty string), then x does not contain any 1's and M terminates in the start state q_0, as desired.

Inductive step. Let x be some non-empty input string. By the *inductive hypothesis*, we can assume that the claim is true for all input strings that are shorter than x.

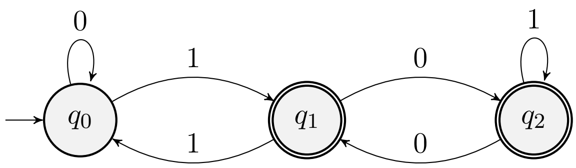
Let x' be the string obtained from x by removing the last symbol. This string is clearly shorter than x, so we can apply the inductive hypothesis to reason about which state M is in just before reading the last symbol of the input x.

1. If x' does not contain any 1's we know that after reading x', M is in state q_0. If the last symbol of x is a 0, then up feeding it to M, it stays in state q_0, as desired. If the last symbol of x is a 1 instead, then x ends with a 1 but does not contain "100" as a substring. But in this case, M will go to state q_1, also as desired.
2. If the last symbol of x' is a "1" but x' does not contain "100", then after reading x', M is in state q_1. We again consider the last symbol of x. If it's a 0, then the last two symbols of x are "10" but x does not contain "100" as a substring, while M ends up in state q_2, as desired. If it's a 1 instead, then the last symbol of x is a 1 and M ends in state q_1, as desired.
3. If the last two symbols of x' are "10" but x' does not contain "100", then after reading x', M is in state q_2. We consider the last symbol of x once more. If it's a 0, then x contains "100", and M ends in state q_3, as desired. If it's a 1, then x ends with a 1 but does not contain "100" as a substring, while M goes to state q_1, as desired.

4. Finally, if x' contains "100" as substring, then after reading x', M is in state q_3. But x' is a substring of x, so x also contains "100" as a substring. Moreover, no matter what the last symbol of x is, M will stay in state q_3, just as desired.

Thus, we conclude that the claim holds true for all input strings x.

6 The following automaton is over the alphabet $\Sigma = \{0, 1\}$ and the set of states is $Q = \{q_0, q_1, q_2\}$. The start state is q_0 and the accepting states are $F = \{q_1, q_2\}$.



6a Write down the transition function δ for this automaton.

6b* Describe the language recognized by the automaton and prove the correctness of your claim.

Hint: *Try all inputs of a fixed length in some natural order and look for a pattern*

Solution:

6a The transition function $\delta : Q \times \Sigma \to Q$ is given by

$$\delta(q_0, 0) = q_0, \quad \delta(q_0, 1) = q_1;$$
$$\delta(q_1, 0) = q_2, \quad \delta(q_1, 1) = q_0;$$
$$\delta(q_2, 0) = q_1, \quad \delta(q_2, 1) = q_2.$$

6b* The automaton recognizes the binary strings w if and only if w, when interpreted as an integer in base two, is not divisible by 3. Note that the empty string represents 0 by convention. Going forth, we will treat binary strings interchangeably with the integers they represent in base two. Let M denote the automaton in question.

Claim. Let x be an arbitrary input. We consider three cases:

1. If $x = 3k$ for some $k \in \mathbb{N}$, then M terminates in state q_0.
2. If $x = 3k + 1$ for some $k \in \mathbb{N}$, then M terminates in state q_1.
3. If $x = 3k + 2$ for some $k \in \mathbb{N}$, then M terminates in state q_2.

Note that these cases are exhaustive. Moreover, observe that q_1 and q_2 are the only accepting states, so it suffices to prove the claim to solve the problem. We prove the claim by induction on the length of the input string x.

Base case. If x has length 0 (i.e. x is the empty string), it represents the integer 0, which is of the form $3k$. Meanwhile, M terminates in the start state q_0, as desired.

Inductive step. Let x be a non-empty input string. By the *inductive hypothesis* we can assume the claim is true for all strings shorter than x.

Let x' be the string obtains from x after removing the last symbol. Since x' is clearly shorter than x, we can apply the inductive hypothesis to x'. We use this to distinguish cases depending on which state M is in after reading all but the last symbol of x:

1. If M is in state q_0, then we have $x' = 3k$ for some $k \in \mathbb{N}$. We consider the last symbol of x. If it's a 0, we have $x = 2x' = 3 \cdot (2k)$, and M will stay in state q_0, as desired. If instead it's a 1, then we have $x = 2x' + 1 = 3 \cdot (2k) + 1$, and M goes to state q_1, as desired.

2. If M is in state q_1, then we have $x' = 3k + 1$ for some $k \in \mathbb{N}$. Again, we consider the last symbol of x. If it's a 0, we have $x = 2x' = 2 \times (3k + 1) = 3 \cdot (2k) + 2$, and M goes to q_2, as desired. If instead it's a 1, we have $x = 2x' + 1 = 2 \cdot (3k + 1) + 1 = 3 \cdot (2k + 1)$, and M goes to q_0, as desired.

3. Finally, if M is in state q_2, then we have $x' = 3k + 2$. If the last symbol of x is a 0, then $x = 2x' = 3 \cdot (2k + 1) + 1$, and M goes to q_1, as desired. If instead it's a 1, then we have $x = 2x' + 1 = 3 \times (2k + 1) + 2$, and M goes to q_2, as desired.

Thus, we conclude that the claim holds for all input strings x.