1-2. Use a general algorithm to design a non-deterministic finite automaton recognizing the language $0(1 \cup 0^*)$. After that, use the general algorithm to design a deterministic finite automaton recognizing this same language.
3-4. Use a general algorithm to transform the following finite automaton into the corresponding regular expression.

- This automaton has two states: a state $q_1$ which is a start state and a final state, and another state $q_2$.
- In the state $q_1$, 0 leads to $q_2$, and 1 leads to $q_1$.
- In the state $q_2$, 0 leads to $q_2$, and 1 leads to $q_1$.

\[
\begin{align*}
R_{s_1} &= R_{s_1} J \cup (R_{s_1} K R_{s_1}^{*} K K R_{s_1} J) \\
R_{s_2} &= R_{s_2} g, U(R_{s_2}, R_{s_2}^{*} g, R_{s_2} g) \\
&= \Lambda \cup (\emptyset 0^* 1) \\
&= \Lambda \\
R_{s_1, q_1} &= R_{s_1, q_1} U(R_{s_1, q_1} R_{s_2} R_{s_2}^{*} q_2, R_{s_2} q_2) \\
&= 1 \cup (0 0^* 1) \\
R_{s_2, f} &= R_{s_2, f} U(R_{s_2}, R_{s_2}^{*} q_2, R_{s_2} f) \\
&= \Lambda \cup (0 0^* \emptyset) \\
&= \Lambda \\
R_{s_1, f} &= R_{s_1, f} U(R_{s_1, f}, R_{s_1} q_1, R_{s_1} f) \\
&= \emptyset \cup (0 \emptyset 0^* \emptyset) \\
&= \emptyset
\end{align*}
\]
5. Let $A$ be a language recognized by the automaton from Problem 3, and let $B$ be the language corresponding to the following automaton:

- This automaton has 3 states, the starting state $q_1$, which is also final, a state $q_2$, and a final state $q_3$.
- When you see 0, from $q_1$ you move to $q_3$, from $q_2$ you move to $q_1$, and from $q_3$ you move to $q_2$.
- When you see 1, from $q_1$ you move to $q_2$, from $q_2$ you move to $q_3$, and from $q_3$ you move to $q_1$.

Use the algorithm that we learned in class, with pairs of states from two automata as states of the new deterministic automaton, to describe the automata recognizing the union and the intersection of these two languages.
6. Prove that the language \( \{a^n b^{n+1} \} = \{\lambda, abb, aabbb, aaabbb, \ldots \} \) is not regular.

Proof by contradiction.

Let's assume that \( L \) is regular. Then by pumping lemma there exists \( p \) s.t. \( \\
\forall w \in L, \exists \in \mathbb{N} (p \geq \frac{|w|}{p} \Rightarrow \exists x y z (w = x y z \land |x y z| \\leq p \land |x y| < p) \land \forall i \in \mathbb{N} (xy^i z \in L)) \).

Let us take \( w = a^p b^{p+1} \) and let's get a contradiction.

\[
w = a^p b^{p+1} = a \cdot a \cdot b \cdot b \cdot b
\]

\[
|w| = 2p + 1 \geq p, \text{ so}
\]

\[
x y z \quad \text{and } |x y| \leq p
\]

So, for this word, there exist \( x, y, z \) as in pumping lemma.

Since \( |x y| \leq p \), \( x \) and \( y \) are among the first \( p \) symbols of the word \( w \), and these symbols are \( a \)'s.

So \( y \) consists only of \( a \)'s.

By pumping lemma \( x y y z \) is in \( L \).

But when go from \( x y z \) to \( x y y z \), we add \( y \), i.e. we add \( a \)'s and we do not add any \( b \)'s, so \( x y y z \) is not \( L \).