1. Design a DFA over the alphabet \{a, b\} that accepts precisely the:

   a) strings with an even number of ‘a’s.

   **Solution:**

   ![Diagram](image)

   b) strings with an odd number of ‘a’s.

   **Solution:**

   ![Diagram](image)

   c) strings with number of ‘a’s divisible by 3.

   **Solution:**

   ![Diagram](image)

   d) strings with number of ‘a’s divisible by \(n\), for some \(n \in \mathbb{N}\).
Solution: You can’t divide by 0 so the problem is not well-defined for 0. For natural numbers $n > 0$ let $M$ be an automaton with statespace $Q = \{0, 1, \ldots, n - 1\}$ so that:

- each state has a self-loop labelled $b$
- for each $k < n$, there is an $a$-labelled transition from state $k$ to state $(k + 1) \mod n$
- 0 is the initial state and the only final state.

I claim that this works. Can you think of a proof?

2. a) Given a DFA $M$, how can we obtain a DFA $M'$ such that $M'$ accepts exactly those strings that are rejected by $M$? (Hint: think about the solutions to a) and b) in the previous question)

Solution: We swap the final and the non-final states. Formally, if $M = (Q, \Sigma, \delta, s, F)$ we let $M' \overset{\text{def}}{=} (Q, \Sigma, \delta, s, F')$ where $F' \overset{\text{def}}{=} Q - F = \{q \in Q | q \notin F\}$. Recall that, by definition, $x \in L(M)$ if and only if $s \xrightarrow{x} q$ and $q \in F$. So

$$x \in L(M') \iff s \xrightarrow{x} q \text{ and } q \in F' \iff s \xrightarrow{x} q \text{ and } q \notin F \iff x \notin L(M)$$

b) Prove that the class of regular languages is closed under complement. That is, if $L$ is a regular language then so is $\sim L = \{x \in \Sigma^* | x \notin L\}$.

Solution: Suppose that $L$ is regular. Then there is a DFA $M$ that accepts it. Let $M'$ be the DFA obtained by swapping final and non-final states, as done in the previous solution. Then $L(M') = \{x \in \Sigma^* | x \notin L(M)\} = \{x \in \Sigma^* | x \notin L\} = \sim L$, thus $\sim L$ is a regular language.

3. Prove the following:

a) $\emptyset$ and $\Sigma^*$ are regular languages.
Solution: Any DFA with an empty set of accepting states accepts the empty language. Thus, $\emptyset$ is regular. An automaton with only one state, say $s$, which is both initial and final, and with a transition of the form $(s, a, s)$ for every $a \in \Sigma$ will accept $\Sigma^*$.

b) if $L_1$ and $L_2$ are regular then $L_1 \cup L_2$ is regular.

Solution: If $L_1$ and $L_2$ are regular, then by definition there must be two DFAs $M_1$ and $M_2$ such that $L_1 = L(M_1)$ and $L_2 = L(M_2)$. W.l.o.g., assume that the set of states of $M_1$ and $M_2$ are disjoint. Then, we can build a new automaton, say $M$, that has $M_1$ and $M_2$ as subcomponents and a fresh initial state $s$ that has an $\epsilon$-move to the initial state of $M_1$ and $M_2$, respectively. It is easy to see that every path in $M$, except for the first transition, is also a path in one of the two automata. Thus, $L(M) = L_1 \cup L_2$. We can conclude that $L_1 \cup L_2$ is regular.

An alternative proof consists in building the cross product of $M_1$ and $M_2$ by picking $(Q_1 \times F_2) \cup (F_1 \times Q_2)$ as the set of final states, where $Q_i$ and $F_i$ are respectively, the set of states and the set of final states of $M_i$, for $i \in \{1, 2\}$. (Provide a proof of this statement) Note that, the same construction does not work if either $M_1$ or $M_2$ is an NFA. (Why?)

c) If $L_1$ and $L_2$ are regular then $L_1 \cap L_2$ is regular.

Solution: If $L_1$ and $L_2$ are regular, then by definition there must be two DFAs $M_1$ and $M_2$ such that $L_1 = L(M_1)$ and $L_2 = L(M_2)$. Assume that $F_i$ is the set of final states of $M_i$, for $i \in \{1, 2\}$. It should be straightforward to see that the cross product of $M_1$ and $M_2$ with set of final states $F_1 \times F_2$ is an automaton that recognises $L_1 \cap L_2$. (Why?) Note that, the same construction does not work if either $M_1$ or $M_2$ is an NFA. (Why?)

d) if $L_1$ and $L_2$ are regular then $L_1 L_2 = \{xy | x \in L_1, y \in L_2\}$ is regular.

Solution: If $L_1$ and $L_2$ are regular, then by definition there must exist two DFAs $M_1$ and $M_2$ such that $L_1 = L(M_1)$ and $L_2 = L(M_2)$. 


Assume that \( s_i \) is the initial state of \( M_i \), and \( F_i \) is the set of final states of \( M_i \), for \( i \in \{1, 2\} \).

We can built an \( \epsilon \)NFA \( M \) that accepts \( L_1L_2 \) as follows. We assume w.l.o.g. that the sets of states of \( M_1 \) and \( M_2 \) are disjoint. We can build \( M \) by composing \( M_1 \) and \( M_2 \) as follows: (1) the initial state of \( M \) is \( s_1 \), (2) we add an \( \epsilon \)-move from each final state in \( F_1 \) to \( s_2 \), and (3) the set of final states of \( M \) is \( F_2 \). Prove that \( M \) accepts \( L_1L_2 \).

Note that, the construction still works if \( M_i \) is either a DFA or an NFA or an \( \epsilon \)NFA.

e) If \( L \) is regular then \( L^* = \{ x_1 \ldots x_k \mid k \in \mathbb{N}, x_i \in L \} \) is regular

**Solution:** If \( L \) is regular, then by definition there must exist a DFAs \( M \) such that \( L = L(M) \). Assume that \( s \) is the initial state of \( M \), and \( F \) is the set of final states of \( M \).

We can built an \( \epsilon \)NFA \( M' \) that accepts \( L^* \) as follows. \( M' \) is the same as \( M \) with the difference that there is an \( \epsilon \)-move from each final state in \( F \) to \( s \), and (2) the set of \( M' \) final states is \( F \cup \{s\} \). Prove that \( M' \) accepts \( L^* \).

Note that, the construction still works if \( M \) is either a DFA or an NFA or an \( \epsilon \)NFA.

4. Suppose that \( M = (Q, \Sigma, \delta_M, s, F) \) is a DFA. Define \( \delta_M: \Sigma^* \rightarrow Q \) recursively as follows:

\[
\hat{\delta}_M(\epsilon) \stackrel{\text{def}}{=} s, \quad \hat{\delta}_M(x\sigma) \stackrel{\text{def}}{=} \delta_M(\hat{\delta}_M(x), \sigma)
\]

(a) Prove that \( x \in L(M) \) if and only if \( \hat{\delta}_M(x) \in F \);

**Solution:** We first prove that \( s \xrightarrow{\epsilon} \hat{\delta}_M(x) \). We can do this by induction on \( x \). For the base case \( x = \epsilon \) and by definition \( \hat{\delta}_M(\epsilon) = s \) and, again by definition, \( s \xrightarrow{\epsilon} s \), so the claim holds for the base case. Now the inductive step is: assume that for some \( x \in \Sigma^* \) we have that \( s \xrightarrow{x} \hat{\delta}_M(x) \), now \( \hat{\delta}_M(x\sigma) = \delta_M(\hat{\delta}_M(x), \sigma) \). By the inductive hypothesis \( s \xrightarrow{\epsilon} \hat{\delta}_M(x) \) and \( \hat{\delta}_M(x) \xrightarrow{\sigma} \hat{\delta}_M(x\sigma) \) so \( s \xrightarrow{\epsilon} \hat{\delta}_M(x\sigma) \) as required.
Using what we have just proved, if \( \hat{\delta}_M(x) \in F \) then \( s \xrightarrow{x} \hat{\delta}_M(x) \) and \( \hat{\delta}_M(x) \) is a final state, so \( x \in L(M) \).

To prove the converse (if \( x \in L(M) \) then \( \hat{\delta}_M(x) \in F \)), we first need to show that if \( s \xrightarrow{x} q \) and \( s \xrightarrow{x} q' \) then \( q = q' \). Again, we can do this by induction on \( x \), and this is left as an exercise for you. Once we have established this, if \( x \in L(M) \) then there exists \( q \in F \) such that \( s \xrightarrow{x} q \). Now also \( s \xrightarrow{x} \hat{\delta}_M(x) \) and so, by the exercise, \( \hat{\delta}_M(x) = q \). This completes the proof.

(b) Suppose that \( M_1 \) and \( M_2 \) are DFAs and \( N = M_1 \times M_2 \). Prove that for any \( x \in \Sigma^* \) we have

\[
\hat{\delta}_N(x) = (\hat{\delta}_{M_1}(x), \hat{\delta}_{M_2}(x)).
\]  

**Solution:** Induction on \( x \). For \( x = \epsilon \), \( \hat{\delta}_N(x) = s_N = (s_{M_1}, s_{M_2}) \) by the definition of the product construction, and \( (\hat{\delta}_{M_1}(x), \hat{\delta}_{M_2}(x)) = (s_{M_1}, s_{M_2}) \). For the inductive step,

\[
\begin{align*}
\hat{\delta}_N(x \sigma) &= \hat{\delta}_N(\hat{\delta}_N(x), \sigma) \quad \text{(Definition of } \hat{\delta}_N) \\
&= \hat{\delta}_N((\hat{\delta}_{M_1}(x), \hat{\delta}_{M_2}(x)), \sigma) \quad \text{(Inductive hypothesis)} \\
&= (\delta_{M_1}(\hat{\delta}_{M_1}(x), \sigma), \delta_{M_2}(\hat{\delta}_{M_2}(x), \sigma)) \quad \text{(Definition of } \delta \text{ in product construction)} \\
&= (\hat{\delta}_{M_1}(x \sigma), \hat{\delta}_{M_2}(x \sigma)) \quad \text{(Definitions of } \hat{\delta}_{M_1}, \hat{\delta}_{M_2})
\end{align*}
\]

(c) Use (1) to prove that \( L(N) = L(M_1) \cap L(M_2) \).

**Solution:** All the hard work has now been done.

\[
x \in L(N) \iff \hat{\delta}_N(x) \in F_N \quad \text{(By (a))}
\]
\[
\iff \hat{\delta}_N(x) \in F_{M_1} \times F_{M_2} \quad \text{(Defn. of final states in prod. constr.))}
\]
\[
\iff (\hat{\delta}_{M_1}(x), \hat{\delta}_{M_2}(x)) \in F_{M_1} \times F_{M_2} \quad \text{(By (b))}
\]
\[
\iff \hat{\delta}_{M_1}(x) \in F_{M_1} \text{ and } \hat{\delta}_{M_2}(x) \in F_{M_2} \quad \text{(Defn. of cartesian product)}
\]
\[
\iff x \in L(M_1) \text{ and } x \in L(M_2) \quad \text{(By (a))}
\]
\[
\iff x \in L(M_1) \cap L(M_2)
\]