1. Convert the following NFA into a DFA using the subset construction. Show clearly which subset of the states of the NFA corresponds to each state of the DFA that you have constructed. Do not include states that are not reachable.

Solution.

2. Convert the following NFAs into DFAs using the subset construction.

Solution.

3. Use the procedure from Lecture 3 in order to come up with a regular expression for the following NFA. You don’t need to follow the recursion to the end. Try to simplify
Solution. Label the initial state 0, the final state 2 and the other state 1. Then the language defined by this automaton will be described by the regular expression $\alpha_{0,2}^{\{0,1,2\}}$. Now

$$\alpha_{0,2}^{\{0,1,2\}} = \alpha_{0,2}^{\{0,2\}} + \alpha_{0,1}^{\{0,2\}} (\alpha_{1,1}^{\{1\}}) \alpha_{1,2}^{\{1\}}$$

It is clear that $\alpha_{0,2}^{\{0,2\}} = b^*a$. Similarly, $\alpha_{0,1}^{\{0,2\}} = b^*a$, $\alpha_{1,1}^{\{1\}} = b + \epsilon$ and $\alpha_{1,2}^{\{1\}} = b$. So we get

$$\alpha_{0,2}^{\{0,1,2\}} = b^*a + b^*a(b + \epsilon)^*b = b^*a + b^*ab^* = b^*ab^*$$

4. Give regular expressions for the following sets of strings over $\{a, b\}$. Aim for simplicity.

a) strings with an even number of $a$'s;

**Solution.** If you think about this for a little bit you can probably just come up with a correct solution, e.g. $b^* + (b^*ab^*)^*$. Otherwise, you can use the procedure that goes from an automaton to a regular expression.

Let us number the states 0, 1 where 0 is the initial state. Then the language accepted by this automaton is $\alpha_{00}^{\{0,1\}}$. We can take away either state 0 or state 1. Here let’s do both and see what regular expressions we get. First take away 0, we get the following using the the formula from Lecture 3:

$$\alpha_{00}^{\{0,1\}} = \alpha_{00}^{\{0\}} + \alpha_{00}^{\{1\}} (\alpha_{00}^{\{1\}})^* \alpha_{00}^{\{1\}}$$

$$= \epsilon + \alpha_{00}^{\{1\}} + \alpha_{00}^{\{1\}} (\alpha_{00}^{\{1\}})^* \alpha_{00}^{\{1\}}$$

(since $\alpha_{00}^{\{1\}}$ matches $\epsilon$)

$$= (\alpha_{00}^{\{1\}})^*$$

So it is enough to work out $\alpha_{00}^{\{1\}}$ – the paths from 0 to 0 that pass only through 1 as an intermediate state. For this it should be enough to examine the automaton: $\alpha_{00}^{\{1\}} = \epsilon + b + ab^*a$. So the final answer is $(\epsilon + b + ab^*a)^* = (b + ab^*a)^*$, which is nice and simple.
The other choice would be to take away 1, then:

\[
\alpha_{00}^{\{0,1\}} = \alpha_{00}^{\{0\}} + \alpha_{01}^{\{0\}} (\alpha_{11}^{\{0\}})^* \alpha_{01}^{\{0\}}
\]

Now \(\alpha_{00}^{\{0\}} = \epsilon + b^*\), \(\alpha_{01}^{\{0\}} = b^*a\), \(\alpha_{11}^{\{0\}} = \epsilon + b + ab^*a\) and \(\alpha_{10}^{\{0\}} = ab^*\). So the final answer is \(\epsilon + b^* + (b^*a(b + ab^*a)^*b^*) = b^* + (b^*a(b + ab^*a)^*ab^*)\).

b) strings with an odd number of \(b\)'s;

**Solution.** Again, here you can probably come up with something without too much effort, for example: \(a^*ba^*(ba^*ba^*)^*\). But let’s look at what the construction gives us.

![Diagram](image1)

Taking away the initial state:

\[
\alpha_{01}^{\{0,1\}} = \alpha_{01}^{\{1\}} + \alpha_{00}^{\{1\}} (\alpha_{00}^{\{0\}})^* \alpha_{01}^{\{1\}}
\]

\[
= \alpha_{01}^{\{1\}} + (\alpha_{00}^{\{1\}})^* \alpha_{01}^{\{1\}}
\]

(Since \(\alpha_{00}^{\{1\}} \) matches \(\epsilon\))

\[
= (\alpha_{00}^{\{1\}})^* \alpha_{01}^{\{1\}}
\]

Now \(\alpha_{00}^{\{1\}} = \epsilon + a + ba^*b\) and \(\alpha_{01}^{\{1\}} = ba^*,\) so the final answer is \((\epsilon + a + ba^*b)^*ba^* = (a + ba^*b)^*ba^*\). You can check yourself what happens if you start by taking away state 1.

c) strings with an even number of \(a\)'s or an odd number of \(b\)'s.

**Solution.** Clearly this will be \(\alpha + \beta\) where \(\alpha\) is any regular expression for strings with an even number of \(a\)’s and \(\beta\) is any regular expression for strings with an odd number of \(b\)’s: so for example \((b + ab^*a)^* + (a + ba^*b)^*ba^*\)

**d) strings with an even number of \(a\)'s and an odd number of \(b\)'s;**

**Solution.** Here I needed to use the fact that we have an automaton to guide us. Remember that this automaton can be obtained via the product construction.

![Diagram](image2)

Using the formula to take away state 3 gives:

\[
\alpha_{03}^{\{0,1,2,3\}} = \alpha_{03}^{\{0,1,2\}} + \alpha_{03}^{\{0,1,2\}} (\alpha_{33}^{\{0,1,2\}})^* \alpha_{33}^{\{0,1,2\}}
\]
Since whatever $\alpha_{33}^{(0,1,2)}$ is, it will match the empty string, the right hand side can be simplified to:

$$= \alpha_{33}^{(0,1,2)}(\alpha_{33}^{(0,1,2)})^*$$

(1)

Formulas for $\alpha_{03}^{(0,1,2)}$ and $\alpha_{33}^{(0,1,2)}$ can be obtained by taking away state 1:

$$\alpha_{03}^{(0,1,2)} = \alpha_{03}^{(0,2)} + \alpha_{01}^{(0,2)}(\alpha_{11}^{(0,2)})^*\alpha_{13}^{(0,2)}$$

$$\alpha_{33}^{(0,1,2)} = \alpha_{33}^{(0,2)} + \alpha_{31}^{(0,2)}(\alpha_{11}^{(0,2)})^*\alpha_{13}^{(0,2)}$$

It is easy to see that $\alpha_{03}^{(0,2)} = b$, $\alpha_{0,1}^{(0,2)} = a$, $\alpha_{11}^{(0,2)} = \alpha_{33}^{(0,2)} = \varepsilon + aa + bb$ and $\alpha_{13}^{(0,2)} = \alpha_{31}^{(0,2)} = ab + ba$.

So $\alpha_{03}^{(0,1,2)} = b + a(aa + bb)(ab + ba)$ and $\alpha_{33}^{(0,1,2)} = \varepsilon + aa + bb + (ab + ba)(aa + bb)(ab + ba)$. Substituting into (1) and slightly simplifying results in:

$$\alpha_{03}^{(0,1,2,3)} = (b + a(aa + bb)(ab + ba))(aa + bb + (ab + ba)(aa + bb)(ab + ba))^*$$

which is the required regular expression.

5. Given an NFA $M = (Q, \Sigma, \Delta, s, F)$, define a function $\tilde{\Delta} : \Sigma^* \to 2^Q$ recursively as follows:

$$\tilde{\Delta}(\varepsilon) = \{s\}$$

$$\tilde{\Delta}(x\sigma) = \bigcup_{q \in \tilde{\Delta}(x)} \Delta(q, \sigma).$$

Prove that $x \in L(M)$ if and only if $\exists f \in F. f \in \tilde{\Delta}(x)$.

Solution. First we prove that $s \xrightarrow{\sigma} q$ if and only if $q \in \tilde{\Delta}(x)$. We can do this by induction on $x$. For $x = \varepsilon$, $s \xrightarrow{\sigma} q$ exactly when $q = s$ and $\tilde{\Delta}(\varepsilon) = \{s\}$, so the claim holds. For $x = y\sigma$, the inductive hypothesis tells us that $s \xrightarrow{\sigma} q$ if and only if $q \in \tilde{\Delta}(y)$. Now

$$s \xrightarrow{\sigma} q \iff \exists q'. s \xrightarrow{\sigma} q' \land q' \xrightarrow{\sigma} q$$

(Definition of $\xrightarrow{\sigma}$)

$$\iff \exists q'. q' \in \tilde{\Delta}(y) \land q' \xrightarrow{\sigma} q$$

(Ind. hyp.)

$$\iff \exists q'. q' \in \tilde{\Delta}(y) \land q' \in \Delta(q, \sigma)$$

(Definition of $\xrightarrow{\sigma}$)

$$\iff q \in \tilde{\Delta}(y\sigma)$$

(Definition of $\tilde{\Delta}(y\sigma)$)

6. Prove that the subset construction preserves the language accepted. That is, starting with any NFA $M$ and constructing the corresponding DFA $M'$ using the subset construction, show that $L(M) = L(M')$.

Solution. We start with an NFA $M = (Q, \Sigma, \Delta : Q \times \Sigma \to 2^Q, s, F)$ and obtain the DFA $M' = (2^Q, \Sigma, \delta : 2^Q \times \Sigma \to 2^Q, s, F')$ where $\delta(U, \sigma) = \bigcup_{u \in U} \Delta(u, \sigma)$ and $F' = \{U | \exists f \in F. f \in U\}$. 
Now suppose that \( x \in L(M) \), that is, using the conclusion of the previous question, \( \exists f \in F. f \in \hat{\Delta}(x) \).

Recall that, for a DFA, in Tutorial 1 we defined \( \hat{\delta} : \Sigma^* \to Q \) by \( \hat{\delta}(\epsilon) = s \) and \( \hat{\delta}(x\sigma) = \delta(\hat{\delta}(x), \sigma) \).

We will now prove that if \( M' \) is obtained from \( M \) using the subset construction as recalled above, then for all \( x \in \Sigma^* \) we have \( \hat{\Delta}(x) = \hat{\delta}(x) \). The proof proceeds by induction on \( x \): if \( x = \epsilon \) then \( \hat{\Delta}(\epsilon) = \{ s \} \) and \( \hat{\delta}(\epsilon) = s_{M'} = \{ s \} \). Now if \( x = y\sigma \) then

\[
\hat{\Delta}(y\sigma) = \bigcup_{q \in \hat{\Delta}(y)} \Delta(q, \sigma) = \bigcup_{q \in \delta(\hat{\delta}(y), \sigma)} \Delta(q, \sigma) = \delta(\hat{\delta}(y), \sigma) = \hat{\delta}(y\sigma)
\]

where the first equality follows by the definition of \( \hat{\Delta} \), the second by the inductive hypothesis, the third by the definition of the \( \delta \) we got via the subset construction and the final by the definition of \( \hat{\delta} \). So now all the hard work is done:

\[
x \in L(M) \iff \exists f \in F. f \in \hat{\Delta}(x) \iff \exists f \in F. f \in \hat{\delta}(x) \iff \hat{\delta}(x) \in F' \iff x \in L(M')
\]

where the first equivalence follows from Exercise 4, the second by what we have just proved, the third by the definition of \( F' \) and the fourth from the exercise 5.a) in Tutorial 1.

7. Prove that the construction in Lecture 2 taking an \( \epsilon \)NFA \( M \) to an ordinary NFA \( M' \) preserves the language accepted.

**Solution.** Recall that the construction takes an \( \epsilon \)NFA \( M = (Q, \Sigma, \Delta, s, F) \) to an NFA \( M' = (Q, \Sigma, \Delta', s, F') \) where

\[
\Delta'(q, \sigma) = \{ q' \mid q \xrightarrow{\sigma} q' \}
\]

\[
F' = \{ q \mid \exists f \in F. q \xrightarrow{f} f \}
\]

We prove that for all \( x \in \Sigma^* \) and \( q \in Q \) we have that \( s \xrightarrow{x} q \) in \( M \) iff \( \exists q' \in Q. \ x \xrightarrow{\sigma} q' \) in \( M' \) and \( q' \xrightarrow{\sigma} q \). The base case \( x = \epsilon \) is trivial. Suppose \( x = y\sigma \), then

\[
s \xrightarrow{y\sigma} q \iff \exists q_1. \ s \xrightarrow{y} q_1 \text{ and } q_1 \xrightarrow{\sigma} q
\]

\[
\iff \exists q_1, q_2. \ s \xrightarrow{y} q_2 \text{ and } q_2 \xrightarrow{\sigma} q_1 \text{ and } q_2 \xrightarrow{\sigma} q \quad \text{(Ind. hyp.)}
\]

\[
\iff \exists q_3. \ x \xrightarrow{\sigma} q' \text{ and } q' \xrightarrow{\sigma} q
\]

Now, using what we have just proved:

\[
x \in L(M) \iff \exists f \in F. s \xrightarrow{f} f \text{ in } M
\]

\[
\iff \exists f \in F, q \in Q. s \xrightarrow{\sigma} q \text{ in } M' \text{ and } q \xrightarrow{f} f
\]

\[
\iff \exists q \in F'. s \xrightarrow{\sigma} q \text{ in } M'
\]

\[
\iff x \in L(M')
\]