## Problem Set for Tutorial 3 - TDDD14/TDDD85

## 1 DFA Minimization



Figure 1: $M_{1}$


Figure 2: $M_{2}$.

## Exercise 1.

1. Minimize the DFA in Figure 1.
2. Minimize the DFA in Figure 2.

## Solutions



Figure 3: The DFA from Figure 1 minimized.


Figure 4: A DFA from Figure 2 minimized.

## Solution to Exercise 1.

1. A minimal DFA is given in Figure 3.
2. A minimal DFA is given in Figure 4.

## 2 Regular Expressions

Exercise 2. Let $r=\left(1+00^{*} 11\right)\left(0+1(0+10)^{*} 11\right)^{*}$. Which of the following strings belong to $L(r)$ ?

1. 010001 ,
2. 00111011,
3. 1100110,
4. 101100,
5. 10011001. 

Exercise 3. Give regular expressions for the following languages over the alphabet $\{0,1\}$.

1. The set of all strings ending in 00 .
2. The set of all strings in which the substring 00 occurs at most once.

Exercise 4. Construct an $\mathrm{NFA}_{\varepsilon}$ which accepts the language defined by the regular expression $10+(0+11) 0^{*} 1$.

Exercise 5. Show that the equalities below hold for regular expressions. ( $r, s$ and $t$ denote arbitrary regular expressions over some alphabet.)

1. $r+t=t+r$,
2. $r(s+t)=r s+r t$,
3. $(r+\epsilon)^{*}=r^{*}$,
4. $r \emptyset=\emptyset r=\emptyset$,
5. $\emptyset^{*}=\epsilon$.

Exercise 6. Give regular expressions that define

1. the language accepted by the DFA in Figure 5.
2. the language accepted by the DFA in Figure 6.


Figure 5: $M_{3}$


Figure 6: $M_{4}$

## Solutions

## Solution to Exercise 2.

1. 010001: no.
2. 00111011: yes.
3. 1100110: yes.
4. 101100: no.
5. 10011001: no.

Hint: no particular method is required here. Work systematically and use the fact that any string matched by the regular expression needs to consist of two substrings, one matching $\left(1+00^{*} 11\right)$, and one matching $\left(0+1(0+10)^{*} 11\right)$ zero or more times.

## Solution to Exercise 3.

1. $(0+1)^{*} 00$,
2. $(1+01)^{*}(\epsilon+0+00)(1+10)^{*}$.

Solution to Exercise 4. By decomposing the regular expression syntactically according to the recursive definition of regular expressions, an $\mathrm{NFA}_{\varepsilon}$ can be constructed systematically in a bottom-up fashion by successively joining $\mathrm{NFA}_{\varepsilon} \mathrm{S}$ corresponding to subexpressions according to the regular operator ( ${ }^{*}$, concatenation, + ) in question. For details, consult the corresponding lecture manuscript. The resulting $\mathrm{NFA}_{\varepsilon}$ is shown in Figure 7.

Solution to Exercise 5. For each pair of regular expressions $x$ and $y$ we need to verify that $L(x)=L(y)$. We provide detailed solutions to the first two exercises since the last three uses the same method.

1. $r+t=t+r: L(r+t)=L(r) \cup L(t)=L(t) \cup L(s)$ (the last equality follows from the fact that union is commutative, i.e., the order of the sets do not matter).
2. $r(s+t)=r s+r t: L(r(s+t))=L(r) L(s+t)=L(r)(L(s) \cup L(t))=L(r) L(s) \cup L(r)=$


Figure 7: $M_{5}$
$L(r) L(t)=L(r s) L(r t)$ (here we use the fact that set concatenation distributes over union, i.e., for any sets $A, B, C$ we have that $A(B \cup C)=A B \cup A C)$.

Solution to Exercise 6. When presenting these solutions and simplifying a regular expression $s$ to an equivalent regular expression $t$ (i.e., $L(s)=L(t)$ ) we for simplicity write $s=t$ rather than $L(s)=L(t)$.

1. (a) Redraw and add a new start and final state to the GNFA.


The state elimination steps are performed as follows.
(b) Eliminate state 1. There are four paths through state 1:

- $S_{1} \rightarrow 1 \rightarrow 2: \varepsilon \emptyset^{*} 0+\emptyset=0$
- $S_{1} \rightarrow 1 \rightarrow 3: \varepsilon \emptyset^{*} 1+\emptyset=1$
- $2 \rightarrow 1 \rightarrow 2: 0 \emptyset^{*} 0+\emptyset=00$
- $2 \rightarrow 1 \rightarrow 3: 0 \emptyset^{*} 1+1=01$

The GNFA now looks as follows.

(c) Eliminate state 3. There are four paths through state 3:

- $S_{1} \rightarrow 3 \rightarrow 2: 1 \emptyset^{*}(0+1)+0=1(0+1)+0$
- $S_{1} \rightarrow 3 \rightarrow S_{f}: 1 \emptyset^{*} \varepsilon+\emptyset=1$
- $2 \rightarrow 3 \rightarrow 2:(01+1) \emptyset^{*}(0+1)+00=(01+1)(0+1)+00$
- $2 \rightarrow 3 \rightarrow S_{f}:(01+1) \emptyset^{*} \varepsilon+\varepsilon=01+1+\varepsilon^{a}$.

The GNFA now looks as follows.

(d) Eliminate state 2. There is one path from state $S$ to state $S f$ :

- $S_{1} \rightarrow 2 \rightarrow S f:(1(0+1)+0)((01+1)(0+1)+00)^{*}(01+1+\varepsilon)+1$.

This regular expression describes the language of the original DFA.

2. Redraw and add a new start and final state to the GNFA.


The state elimination steps are performed as follows.
3. Eliminate state $q_{3}$. There are four paths through state $q_{3}$ :

- $q_{1} \rightarrow q_{3} \rightarrow q_{1}: 0 \emptyset^{*} 0+\emptyset=00$
- $q_{1} \rightarrow q_{3} \rightarrow q_{2}: 0 \emptyset^{*} 1+1=01+1$
- $q_{2} \rightarrow q_{3} \rightarrow q_{1}: 0 \emptyset^{*} 0+0+1=00+1$
- $q_{2} \rightarrow q_{3} \rightarrow q_{2}: 0 \emptyset^{*} 1+\emptyset=01$

The GNFA now looks as follows.

4. Eliminate state $q_{2}$. There are two paths through state $q_{2}$ :

- $q_{1} \rightarrow q_{2} \rightarrow q_{1}:(01+1)(01)^{*}(00+1)+00$
- $q_{1} \rightarrow q_{2} \rightarrow S_{f}:(01+1)(01)^{*} \varepsilon+\emptyset=(01+1)(01)^{*}$

The GNFA now looks as follows.

5. Eliminate state $q_{1}$. There is only one path.

- $S_{1} \rightarrow q_{1} \rightarrow S f: \varepsilon\left((01+1)(01)^{*}(00+1)+00\right)^{*}(01+1)(01)^{*}+\emptyset=\left((01+1)(01)^{*}(00+\right.$ $1)+00)^{*}(01+1)(01)^{*}$


This regular expression describes the language of the original DFA.

[^0]
## 3 Advanced and Exam Like Exercises

Exercise 7. Let the languages $L_{1}$ and $L_{2}$ be defined as follows:

- $L_{1}$ is defined by the regular expression $(a+b)^{*} b b a(a+b)^{*}$.
- $L_{2}$ is the language of strings over $\{a, b\}^{*}$ containing the string $a b$.

Give a regular expression $R$ such that $L(R)=L_{1}-L_{2}$, i.e, $L(R)=\left\{w \mid w \in L_{1} \wedge w \notin L_{2}\right\}$. Explain your reasoning and why your solution is correct.

Exercise 8. Using a standard method, construct a regular expression defining the same language as the DFA whose transition function $\delta$ is given by

|  | $a$ | $b$ |
| :---: | :---: | :---: |
| $A$ | $A$ | $C$ |
| $B \mathrm{~F}$ | $A$ | $B$ |
| $C \mathrm{~F}$ | $B$ | $A$ |

Exercise 9. For each pair of regular expressions $R_{1}$ and $R_{2}$ below, answer whether they generate the same language $\left(L\left(R_{1}\right)=L\left(R_{2}\right)\right)$. If no, give a string which belongs to one of the languages and does not belong to the other. If yes, show that they are equivalent, e.g., by (1) computing $L\left(R_{1}\right)$ and $L\left(R_{2}\right)$ as far as you can and (2) verifying that the two resulting sets are equal. For the last step, an informal explanation is sufficient.

1. $\varepsilon+\varepsilon$ and $\varepsilon$.
2. $\emptyset+\emptyset$ and $\emptyset$.
3. $a(b+c+\varepsilon)$ and $a b+b c$.
4. $(a b+a)^{*} a$ and $a(b a+a)^{*}$.

## Solutions

Solution to Exercise 7. Hint: while it is possible to solve the problem by a systematic approach by constructing a DFA for $L_{1} \cap \overline{L_{2}}$ and converting this DFA to a regular expression, it is much easier to construct the regular expression directly. Hence, how can you adapt the regular expression $(a+b)^{*} b b a(a+b)^{*}$ so that it does not match $a b$ ?

## Solution to Exercise 8.

1. Redraw and add a new start and final state to the GNFA.


The state elimination steps are performed as follows.
2. Eliminate state $B$. There are 2 paths through state $B$ :

- $C \rightarrow B \rightarrow A: a b^{*} a+b$
- $C \rightarrow B \rightarrow S_{f}: a b^{*} \varepsilon+\varepsilon=a b^{*}+\varepsilon$

The GNFA now looks as follows.

3. Eliminate state $A$. There are two paths through state $A$ :

- $S_{1} \rightarrow A \rightarrow C: \varepsilon a^{*} b+\emptyset=a^{*} b$
- $C \rightarrow A \rightarrow C:\left(a b^{*} a+b\right) a^{*} b+\emptyset=\left(a b^{*} a+b\right) a^{*} b$

The GNFA now looks as follows.

4. Eliminate state $C$.

$$
\text { - } S_{1} \rightarrow C \rightarrow S f: a^{*} b\left(\left(a b^{*} a+b\right) a^{*} b\right)^{*}\left(a b^{*}+\varepsilon\right)+\emptyset=a^{*} b\left(\left(a b^{*} a+b\right) a^{*} b\right)^{*}\left(a b^{*}+\varepsilon\right) \text {. }
$$

We obtain the following GNFA.


This regular expression describes the language of the original DFA.

## Solution to Exercise 9.

1. $L(\varepsilon+\varepsilon)=L(\varepsilon) \cup L(\varepsilon)=\{\varepsilon\} \cup\{\varepsilon\}=\{\varepsilon\}$. Hence, they are the same.
2. $\emptyset+\emptyset$ and $\emptyset$ : a similar argument to the above shows that they are the same.
3. $L(a(b+c+\varepsilon))=L(a) L(b+c+\varepsilon)=\{a\}\{a, b, \varepsilon\}=\{a a, a b, a\}$ which is not the same as $L(a b+b c)=\{a b, b c\}$.
4. $L\left((a b+a)^{*} a\right)=\{a b, a\}^{*}\{a\}$, i.e., the set of strings starting with an arbitrary combination of $a b$ and $a$ and ending with $a$. For the other expression we see that $L\left(a(b a+a)^{*}\right)=\{a\}\{b a, a\}^{*}$, i.e., the set of strings starting with an $a$ and ending with an arbitrary combination of $b a$ or $a$. Since $a$ concatenated with $b a$ gives us the same as $a b$ concatenated with $a$ it is not hard to see that the two expressions describe the same language.

[^0]:    ${ }^{a}$ Note in particular that we in this case cannot simplify this to $01+1$.

