Technische Universität München **Winter term 2018/19**
I7
Prof. J. Esparza / S. Sickert **19.10.2018**

Automata and Formal Languages — Homework 1

Due 23.10.2017

These exercises are either adapted from the lecture notes or are newly created and the solutions will be published as part of the lecture notes.

Exercise 1.1
Give a regular expression and a NFA for the language of all words over $\Sigma = \{a, b\}$ ...

1. ...beginning and ending with the same letter.
2. ...having two occurrences of a at distance 3.
3. ...with no occurrences of the subword aa.
4. ...containing exactly two occurrences of aa.
5. ...that can be obtained from $abaab$ by deleting letters.

Exercise 1.2
Prove or disprove: Every regular language is recognized by a NFA ...

1. ...having one single initial state.
2. ...having one single final state.
3. ...whose states are all initial.
4. ...whose states are all final.
5. ...whose initial states have no incoming transitions.
6. ...whose final states have no outgoing transitions.
7. ...such that all input transitions of a state (if any) carry the same label.
8. ...such that all output transitions of a state (if any) carry the same label.

Which of the above hold for DFAs? Which ones for NFA-ϵ?

NFA-ϵ will be introduced on Monday, but you probably know them from your Bachelor's course.

Exercise 1.3
Prove or disprove: the languages of the regular expressions $(1 + 10)^*$ and $1^*(101^*)^*$ are equal.

Exercise 1.4

The *reverse* of a word $w \in \Sigma^*$ is defined as

$$w^R = \begin{cases} \varepsilon & \text{if } w = \varepsilon, \\ a_n a_{n-1} \cdots a_1 & \text{if } w = a_1 a_2 \cdots a_n \text{ where each } a_i \in \Sigma. \end{cases}$$

The *reverse* of a language $L \subseteq \Sigma^*$ is defined as $L^R = \{w^R \mid w \in L\}$.

(a) Give a regular expression for the reverse of $\big((a + ba)^* ba(a + b)\big)^* ba$.

(b) Give an algorithm that takes as input a regular expression r and returns a regular expression r^R such that $\mathcal{L}(r^R) = \big(\mathcal{L}(r)\big)^R$.

(c) Let A be an NFA. Describe an NFA B such that $L(B) = L(A)^R$.

(d) Does your construction in (c) works for DFAs as well? More precisely, does it preserve determinism?

Exercise 1.5

Recall that a nondeterministic automaton A accepts a word w if at least one of the runs of A on w is accepting. This is sometimes called the *existential* accepting condition. Consider the variant in which A accepts w if *all* runs of A on w are accepting (in particular, if A has no run on w then it accepts w). This is called the *universal* accepting condition. Notice that a DFA accepts the same language with both the existential and the universal accepting conditions.

Intuitively, we can visualize an automaton with universal accepting condition as executing all runs in parallel. After reading a word w, the automaton is simultaneously in all states reached by all runs labelled by w, and accepts if all those states are accepting.

Consider the family L_n of languages over the alphabet $\{0, 1\}$ given by $L_n = \{ww \in \Sigma^{2n} \mid w \in \Sigma^n\}$.

1. Give an automaton of size $\mathcal{O}(n)$ with universal accepting condition that recognizes L_n.
2. Give an NFA recognizing L_n. [Hint:]

Solution 1.1
We write Σ^* for $(a+b)^*$.

1. $a\Sigma^*a + b\Sigma^*b$
2. $\Sigma^*a\Sigma^*\Sigma^*a\Sigma^*$
3. $(b+ab)^*(\varepsilon+a)$
4. $(b+ab)^* \left(aaa + aab(b+ab)^*aa\right) \left(\varepsilon + b(b+ab)^*\right)$
5. $(a+\varepsilon)(b+\varepsilon)(a+\varepsilon)(b+\varepsilon)(b+\varepsilon)$

Solution 1.3
The languages are equal. We prove that each of them is included in the other. We identify r and $L(r)$

Let $w \in 1^*(101^*)^*$. Then there are $x, y_1, \ldots, y_m$ such that $w \in 1^x(101^{y_1})(101^{y_1})\cdots(101^{y_m})$, and so $w \in 1^x(10)1^{y_1}(10)1^{y_2}\cdots(10)1^{y_m}$, which proves $w \in (1+10)^z$ for $z = x+1+y_1+1+\cdots+1+y_m$. So $w \in (1+10)^*$.

For the other inclusion, let $w \in (1+10)^*$. Then $w \in (1+10)^n$ for some $n > 0$. We prove $w \in 1^*(101^*)^*$ by induction on n. If $n = 0$, then $w = \varepsilon$, which belongs to $1^0(101^*)^0$. If $n > 0$, then either $w = w'1$ or $w = w'10$ for some word $w' \in (1+10)^{n-1}$. By induction hypothesis there are $x, y_1, \ldots, y_m \geq 0$ such that $w' \in 1^x(101^{y_1})\cdots(101^{y_m})$. If $w = w'1$, then $w \in 1^x(101^{y_1})\cdots(101^{y_m+1})$, and so $w \in 1^x(101^*)^m$. If $w = w'10$ then $w' \in 1^x(101^{y_1})\cdots(101^{y_m})(101^0)$, and so $w \in 1^x(101^*)^{m+1}$. In both cases $w \in 1^*(101^*)^*$.

Solution 1.4

(a) We reverse the transitions of A and swap its initial and final states. More formally, let $A = (Q, \Sigma, \delta, Q_0, F)$. We define B as $B = (Q, \Sigma, \delta', F, Q_0)$ where $\delta'(p, a) = \{q \in Q \mid p \in \delta(q, a)\}$.

(b) No, if A is deterministic, then B is not necessarily deterministic. For example, the construction applied to the DFA of #1.2(a) for M_2 does not yield a DFA.