ETH zürich
Departement Informatik

Theoretische Informatik

Prof. Dr. Juraj Hromkovič
Dr. Hans-Joachim Böckenhauer
https://courses.ite.inf.ethz.ch/theo_inf_21

Exemplary Solutions – Sheet 5

Zürich, October 29, 2021

Solution to Exercise 13

(a) Let $L_1 = \{u\#v\#w \mid u, v, w \in \{0,1\}^+ \text{ and } \text{Number}(u) \cdot \text{Number}(v) = \text{Number}(w)\}$. We prove that L_1 is not regular using Lemma 3.12 as well as the pumping lemma. Because L_1 is defined over an alphabet of size 3, the Kolmogorov complexity argument is not directly applicable.

Proof using Lemma 3.12. Suppose that L_1 is regular. Then there exists an automaton $A_1 = (Q, \{0,1,\#\}, \delta, q_0, F)$ with $L(A_1) = L_1$. Let $m = |Q|$. We consider the words

$$1\#1\#, 1^2\#1\#, 1^3\#1\#, \ldots, 1^{m+1}\#1\#\,.$$

Since these are $m+1$ words, i.e., more words than the number of A_1's states, there exist $i, j \in \{1, \ldots, m+1\}$ with $i < j$ such that

$$\hat{\delta}(q_0, 1^i\#1\#) = \hat{\delta}(q_0, 1^j\#1\#).$$

By Lemma 3.12, for all $z \in \{0,1,\#\}^*$, we have

$$1^i\#1\#z \in L_1 \iff 1^j\#1\#z \in L_1.$$

However, choosing $z = 1^i$ leads to a contradiction because $1^i\#1\#z = 1^i\#1\#1^i \in L_1$ and $1^j\#1\#z = 1^j\#1\#1^i \notin L_1$. Hence, the assumption is wrong and the language L_1 is not regular.

Proof using the pumping lemma. Suppose that L_1 is regular. Then the pumping lemma (Lemma 3.14) yields a constant $n_0 \in \mathbb{N}$ such that every word $w \in \{0,1,\#\}^*$ with $|w| \geq n_0$ can be split into three parts y, x, and z so that

(i) $|yx| \leq n_0$,
(ii) $|x| \geq 1$, and
(iii) either $\{yx^kz \mid k \in \mathbb{N}\} \subseteq L_1$ or $\{yx^kz \mid k \in \mathbb{N}\} \cap L_1 = \emptyset$.

We choose the word $w = 1^{n_0}\#1\#1^{n_0}$. It clearly holds that $|w| \geq n_0$. Hence, there exists a decomposition $w = yxz$ of w satisfying the conditions (i), (ii),

and (iii). Because of (i), $|yx| \leq n_0$ holds, thus $y = 1^l$ and $x = 1^m$ for $l, m \in \mathbb{N}$ with $m \leq n_0$. Because of (ii), $m > 0$ holds. Because $w \in L_1$, (iii) implies that

$$\{yx^kz \mid k \in \mathbb{N}\} = \{1^{n_0+(k-1)\cdot m}\#1\#1^{n_0} \mid k \in \mathbb{N}\} \subseteq L_1\,.$$

However, this is a contradiction because $yx^2z = 1^{n_0+m}\#1\#1^{n_0} \notin L_1$. Hence, the assumption is wrong and the language L_1 is not regular.

(b) Let $L_2 = \{0^p \mid p \in \mathbb{N} \text{ is a prime number}\}$. We first show that L_2 is not regular using the pumping lemma.

Proof using the pumping lemma. Suppose that L_2 is regular. Then the pumping lemma (Lemma 3.14) yields a constant $n_0 \in \mathbb{N}$ such that every word $w \in \{0\}^*$ with $|w| \geq n_0$ can be split into three parts y, x, and z so that

(i) $|yx| \leq n_0$,
(ii) $|x| \geq 1$, and
(iii) either $\{yx^kz \mid k \in \mathbb{N}\} \subseteq L_2$ or $\{yx^kz \mid k \in \mathbb{N}\} \cap L_2 = \emptyset$.

We choose the word $w = 0^p$ for a prime number $p \geq n_0$. It clearly holds that $|w| \geq n_0$. Hence, there exists a decomposition $w = yxz$ of w satisfying the conditions (i), (ii), and (iii). Because of (ii), $m > 0$ holds. Because $w \in L_2$, (iii) implies that

$$\{yx^kz \mid k \in \mathbb{N}\} = \{0^{p+(k-1)\cdot m} \mid k \in \mathbb{N}\} \subseteq L_2\,.$$

Now we choose $k = p + 1$. Then $yx^kz = 0^{p+(k-1)m} = 0^{p+pm} = 0^{p\cdot(m+1)}$. Hence, we derive a contradiction because $p \cdot (m + 1)$ is no prime number (recall that $m > 0$). Hence, the assumption is wrong and the language L_2 is not regular.

A consequence of the prime number theorem. We can also show that L_2 is not regular using the Kolmogorov complexity argument or Lemma 3.12. To this end, we need the prime number theorem (Theorem 2.67 in the textbook) that we have not proved in the lecture. The prime number theorem implies that the difference between two consecutive prime numbers can be arbitrarily large: if the difference was upper bounded by some $k \in \mathbb{N}$, then there were at least n/k prime numbers among the first n natural numbers. This would contradict the prime number theorem saying that there are approximately $n/\ln n$ such prime numbers.

Now we use this observation in the Kolmogorov complexity argument showing that L_2 is not regular.

Proof using the Kolmogorov complexity argument. Suppose that $L = L_2$ is regular. Let p_m be the m-th prime number. Then $0^{p_{m+1}-p_m-1}$ is the first word in the language

$$L_{0^{(p_m)+1}} = \{y \mid 0^{(p_m)+1}y \in L\}.$$

Theorem 3.19 yields a constant c, independent of m, so that

$$K(0^{p_{m+1}-p_m-1}) \leq \lceil \log_2(1+1) \rceil + c = 1 + c.$$

Since there are only finitely many programs of constant length at most $1 + c$, but the above consequence of the prime number theorems implies that there are

infinitely many words of the form $0^{p_{m+1}-p_m-1}$, we derive a contradiction. Hence, the assumption is wrong and the language L_2 is not regular.

We can also use the above consequence of the prime number theorem to show that L_2 is not regular, using Lemma 3.12.

Proof using Lemma 3.12. Suppose that L_2 is regular. Then there exists an automaton $A_2 = (Q, \{0\}, \delta, q_0, F)$ with $L(A_2) = L_2$. Let $m = |Q|$. We choose $m+1$ distinct prime numbers $p_{l_0}, \ldots, p_{l_m}$ such that the differences $p_{l_j+1} - p_{l_j}$ between consecutive prime numbers are pairwise distinct and – without loss of generality – monotonically increasing. Such prime numbers exist due to the above consequence of the prime number theorem. Then we consider the words

$$0^{p_{l_0}}, 0^{p_{l_1}}, \ldots, 0^{p_{l_m}} .$$

Since these are $m+1$ words, i.e., more words than the number of A_2's states, there exist $i, j \in \{0, \ldots, m\}$ with $i < j$ such that

$$\hat{\delta}_{A_2}(q_0, 0^{p_{l_i}}) = \hat{\delta}_{A_2}(q_0, 0^{p_{l_j}}).$$

By Lemma 3.12, for all $z \in \{0\}^*$, we have

$$0^{p_{l_i}} z \in L_2 \iff 0^{p_{l_j}} z \in L_2.$$

However, choosing $z = 0^{p_{l_i+1}-p_{l_i}}$ leads to a contradiction because

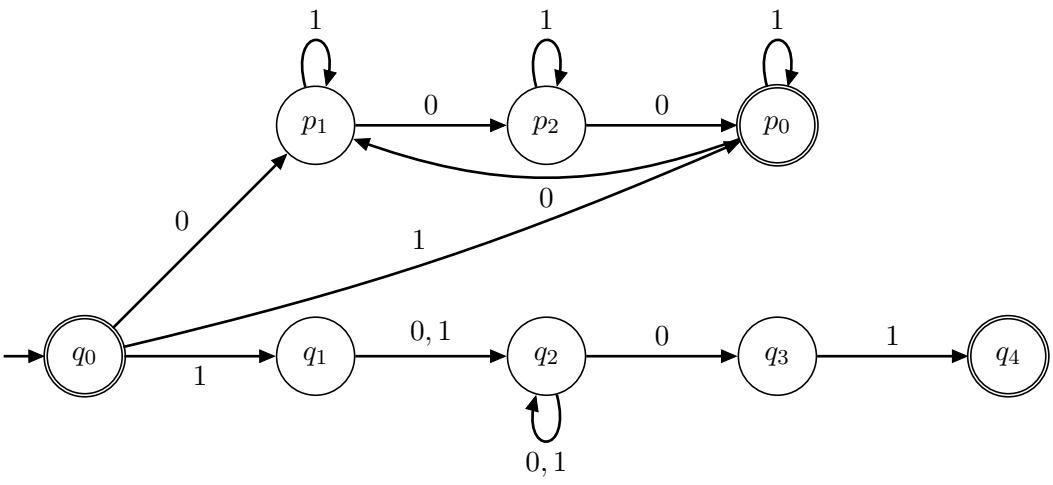
$$0^{p_{l_i}} z = 0^{p_{l_i+1}} \in L_2$$

and $0^{p_{l_j}} z = 0^{p_{l_j}+(p_{l_i+1}-p_{l_i})} \notin L_2$ by the assumption on the chosen prime numbers. Hence, the assumption is wrong and the language L_2 is not regular.

Solution to Exercise 14

(a) The following nondeterministic finite automaton M accepts the language

$$L = \{x \in \{0,1\}^* \mid |x|_0 \bmod 3 = 0 \text{ or } x = 1y01 \text{ for } y \in \{0,1\}^+\} .$$



This automaton consists of two subautomata for the two languages

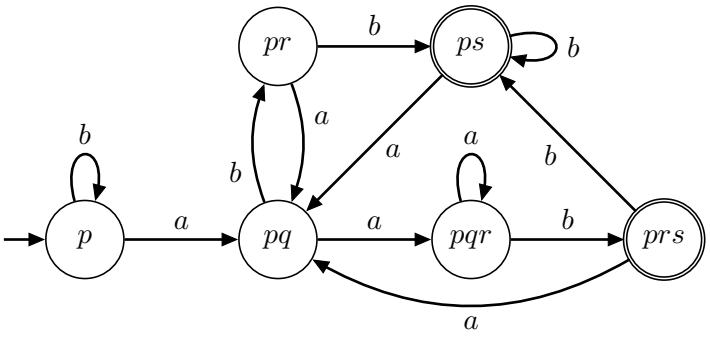
$$L_1 = \{x \in \{0,1\}^* \mid |x|_0 \bmod 3 = 0\}$$

and

$$L_2 = \{x \in \{0,1\}^* \mid x = 1y01 \text{ for } y \in \{0,1\}^+\}$$

with $L_1 \cup L_2 = L$. The automaton M branches from the initial state q_0 nondeterministically into one of the two subautomata. In the states p_0, p_1, and p_2, M counts the number of zeros in the input modulo three. If the count is 0, the first of two conditions of L is satisfied and M accepts the input in the state p_0. In the states q_1 through q_4, M looks for the pattern $y01$ from the second condition of L. The automaton M decides nondeterministically in the state q_2 when the suffix 01 starts. The prefix 1 of the overall pattern $1y01$ is read upon the transition from q_0 to q_1. The state q_0 must be accepting because the empty word λ is contained in $L_1 \subseteq L$.

(b) Applying the power set construction to the provided nondeterministic finite automaton yields the following deterministic finite automaton A. All nonreachable states have been left out from A. For the sake of the diagram's simplicity, the labels of the states have been shortened, e.g., pqr stands for $\langle\{p,q,r\}\rangle$.



Solution to Exercise 15

(a) Because L_1 and L_2 are regular languages, there exist finite automata

$$A_1 = (Q_1, \{a,b\}, \delta_1, q_{0,1}, F_1) \text{ and } A_2 = (Q_2, \{a,b\}, \delta_2, q_{0,2}, F_2)$$

with $L(A_1) = L_1$ and $L(A_2) = L_2$. We provide a finite automaton A with $L(A) = L$ that implies the regularity of $L = L_1\{c\}L_2$. Without loss of generality, we assume that the sets Q_1, Q_2, and $\{q_s\}$ are pairwise disjoint, where q_s is an additional state. Let $A = (Q_1 \cup Q_2 \cup \{q_s\}, \{a,b,c\}, \delta, q_{0,1}, F_2)$ be the automaton with the transition function δ defined as follows:

$$\begin{aligned}
&\delta(q,s) = q' \text{ for all } s \in \{a,b\} \text{ and } q \in Q_1 \text{ with } \delta_1(q,s) = q',\\
&\delta(q,s) = q' \text{ for all } s \in \{a,b\} \text{ and } q \in Q_2 \text{ with } \delta_2(q,s) = q',\\
&\delta(q,c) = q_{0,2} \text{ for all } q \in F_1,\\
&\delta(q,c) = q_s \text{ for all } q \in (Q_1 - F_1) \cup Q_2, \text{ and}\\
&\delta(q_s,s) = q_s \text{ for all } s \in \{a,b,c\}.
\end{aligned}$$

The automaton A first proceeds exactly like the automaton A_1, as long as symbols from $\{a, b\}$ are read, except that the states $F_1 \subseteq Q_1$ are not accepting in A. If c is read in one of the states from F_1, the automaton A makes a transition to the initial state $q_{0,2}$ of the automaton A_2 and then proceeds exactly like A_2, as long as symbols from $\{a, b\}$ are read. If c is read in some state outside F_1, the automaton A makes a transition to the sink state q_s that is not accepting and stays there independently of the remaining input. Hence, exactly the words from $L_1\{c\}L_2 = \{w_1 c w_2 \mid w_1 \in L_1, w_2 \in L_2\}$ are accepted.

(b) Since L is regular, there exists a finite automaton $A = (Q, \Sigma, \delta, q_0, F)$ with $L(A) = L$. We describe a finite automaton A^R with $L(A^\mathsf{R}) = L^\mathsf{R}$, which proves the claim.

We first assume that A has exactly one accepting state, i.e., $F = \{f\}$ for some $f \in Q$. In this case, it is sufficient to reverse all transitions. Formally, this yields in general a nondeterministic finite automaton $A^\mathsf{R} = (Q, \Sigma, \delta^\mathsf{R}, f, \{q_0\})$ with

$$\delta^\mathsf{R}(q, s) = \{q' \in Q \mid \delta(q', s) = q\} \text{ for all } s \in \Sigma \text{ and } q \in Q.$$

This automaton obviously accepts L^R and, using the powerset construction from Theorem 3.26 in the textbook, we can transform it into an equivalent deterministic finite automaton.

If the automaton A does not have exactly one accepting state, we define $A_f = (\Sigma, Q, \delta, q_0, \{f\})$, for each state $f \in F$, and observe that $L(A) = \bigcup_{f \in F} L(A_f)$. Hence, we can apply the above construction $|F|$ times to construct, for each $f \in F$, a deterministic finite automaton A_f^R with $L(A_f^\mathsf{R}) = L(A_f)^\mathsf{R}$, and combine the resulting automata into a large product automaton A^R accepting the language $L(A^\mathsf{R}) = \bigcup_{f \in F} L(A_f^\mathsf{R}) = \Big(\bigcup_{f \in F} L(A_f)\Big)^\mathsf{R}$.