ETH zürich
Departement Informatik

Theoretische Informatik

Prof. Dr. Juraj Hromkovič
Dr. Hans-Joachim Böckenhauer
`https://courses.ite.inf.ethz.ch/theoInf20`

Exemplary Solutions – Sheet 3

Zürich, October 9, 2020

Solution to Exercise 7

We present an indirect proof of the statement. Suppose that there exist infinitely many random prime numbers. By definition, this means that there exist infinitely many $k \in \mathbb{N} - \{0\}$ such that

$$K(p_k) \geq \lceil \log_2(p_k + 1) \rceil - 1 \geq \log_2(p_k) - 1, \tag{1}$$

where p_k is the k-th prime number.

Furthermore, the m-th prime number p_m for every $m \in \mathbb{N} - \{0\}$ can be computed by a Pascal program C_m which contains the binary representation of m, carries out a primality test for all natural numbers in increasing order, and returns the m-th number passing the primality test. All parts of the program C_m except for the binary representation of m have a constant length. Hence, the length of the machine code of C_m is $\lceil \log_2(m+1) \rceil + c \leq \log_2 m + c'$ for constants c and $c' = c + 1$.

For each $n \in \mathbb{N}$, let $\text{Prim}(n)$ denote the number of primes less than or equal to n. By the prime number theorem (Theorem 2.67, Theorem 2.3 in the German version of the book),

$$\text{Prim}(n) < \frac{n}{\ln n - \frac{3}{2}}$$

holds for all $n > 67$. By definition, we have $m = \text{Prim}(p_m)$ and thus

$$\text{Prim}(p_m) = m < \frac{p_m}{\ln(p_m) - \frac{3}{2}}$$

which finally implies that

$$K(p_m) \leq \log_2\left(\frac{p_m}{\ln(p_m) - \frac{3}{2}}\right) + c' = \log_2(p_m) - \log_2\left(\ln(p_m) - \frac{3}{2}\right) + c' \,. \tag{2}$$

Combining the two bounds (1) and (2) on the Kolmogorov-complexity yields

$$\log_2(p_l) - 1 \leq \log_2(p_l) - \log_2\left(\ln(p_l) - \frac{3}{2}\right) + c'$$

for infinitely many $l \in \mathbb{N}$. It follows that, for those l,

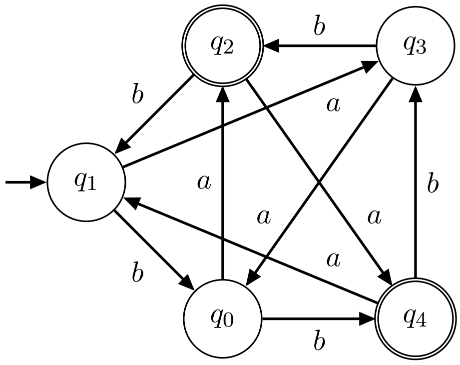
$$\log_2\left(\ln(p_l) - \frac{3}{2}\right) \leq c' + 1 \,,$$

which is impossible since c' is a constant and $\log_2(\ln(p_l) - 3/2)$ is unbounded for growing p_l. Hence, our assumption is false and the statement to be proved holds.

Solution to Exercise 8

(a) The following finite automaton accepts the language

$$L_1 = \{\, w \in \{a,b\}^* \mid (2|w|_a - |w|_b + 1) \bmod 5 \in \{2,4\} \,\}.$$



This automaton has a state q_i for every possible value i of $(2|w|_a - |w|_b + 1) \bmod 5$ and reaches the state q_j for $j = (2|x|_a - |x|_b + 1) \bmod 5$ after reading a prefix x. Hence, q_2 and q_4 are the accepting states of the automaton and q_1 is the initial state since $(2|\lambda|_a - |\lambda|_b + 1) \bmod 5 = 1$ for the empty word λ.

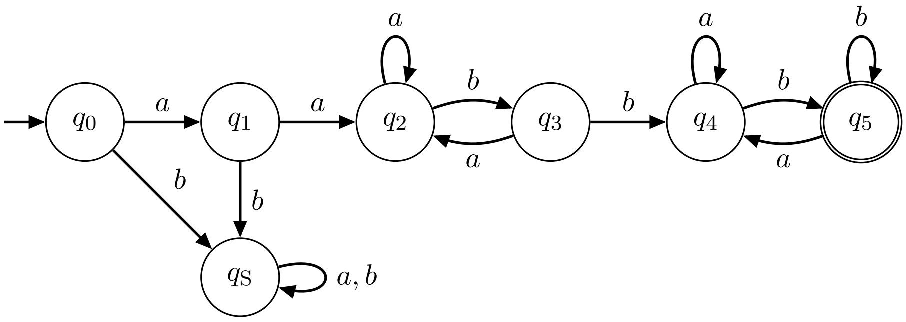
This yields the classes

$$\mathrm{Kl}[q_i] = \{\, w \in \{a,b\}^* \mid (2|w|_a - |w|_b + 1) \bmod 5 = i \,\},$$

for $0 \le i \le 4$.

(b) The following finite automaton accepts the language

$$L_2 = \{\, aaxb \mid x \in \{a,b\}^* \text{ and } x \text{ contains } bb \text{ as a subword} \,\}.$$



Using the transitions from q_0 to q_1 and from q_1 to q_2, this automaton checks if the input word starts by aa. If this is not the case, the automaton reaches the trap state q_S and stays in it for the rest of the computation, thus rejecting each such word. Otherwise, it checks if the subword bb occurs using the states q_2 and q_3. Once this is the case, the automaton checks if the word ends by b using the states q_4 and q_5, and, if so, it accepts in q_5.

This yields the following classes of states:

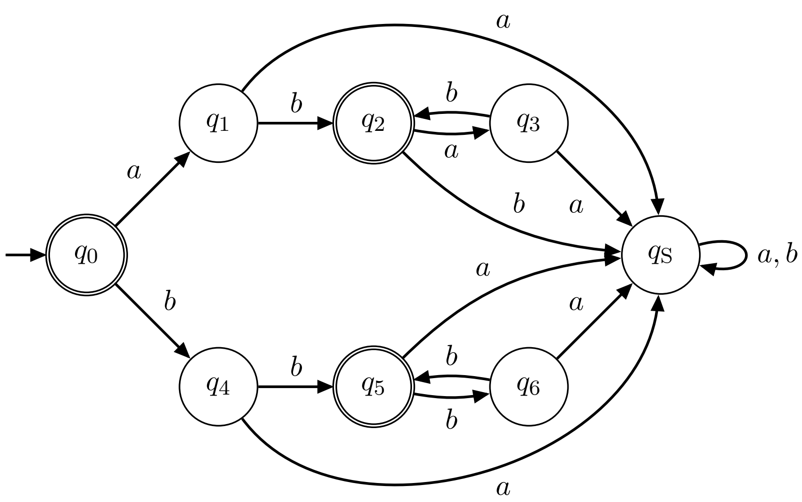
$$
\begin{aligned}
&\mathrm{Kl}[q_0] = \{\lambda\}\\
&\mathrm{Kl}[q_1] = \{a\}\\
&\mathrm{Kl}[q_2] = \{\, aax \mid x \in \{ba, a\}^* \,\}\\
&\mathrm{Kl}[q_3] = \{\, aaxb \mid x \in \{ba, a\}^* \,\}\\
&\mathrm{Kl}[q_5] = L_2\\
&\mathrm{Kl}[q_S] = \{\, bx \mid x \in \{a, b\}^* \,\} \cup \{\, abx \mid x \in \{a, b\}^* \,\}\\
&\mathrm{Kl}[q_4] = \{a, b\}^* - \left(\mathrm{Kl}[q_S] \cup \bigcup_{i=0}^{3} \mathrm{Kl}[q_i] \cup \mathrm{Kl}[q_5] \right)
\end{aligned}
$$

Note that it is often helpful to choose a suitable order in which the classes are described.

Solution to Exercise 9

(a) The following finite automaton accepts the language

$$L_1 = \{\, x^k \mid x \in \{ab, bb\},\ k \in \mathbb{N} \,\}.$$

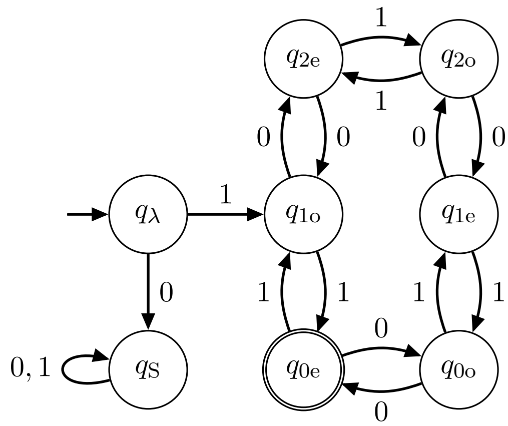


This construction is based on the following idea:

The automaton ends in the initial state q_0 only when given the empty word λ as input. Since $\lambda = x^0 \in L$ for arbitrary x, the state q_0 is an accepting state. Using the transitions from q_0 to q_1 and further to q_2, or from q_0 to q_4 and further to q_5, respectively, the automaton checks if the input starts by a prefix $x \in \{ab, bb\}$. If this is not the case, the automaton reaches the trap state q_S and stays in it for the rest of the computation, thus rejecting each such word. Using the states q_2 and q_3, or q_5 and q_6, respectively, the automaton then checks if the rest of the input is of the form x^k for some $k \in \mathbb{N}$: After reading $x = ab$, the automaton reaches the accepting state q_2, in particular, already after reading the prefix ab (corresponding to $k = 1$). Analogously, the automaton reaches the accepting state q_5 after reading $x = bb$. If the input deviates from this pattern, the automaton immediately reaches the trap state.

(b) The following finite automaton accepts the language

$$L_2 = \{ w \in \{0,1\}^* \mid |w| \text{ is even and } w = \mathrm{Bin}(n) \text{ for some } n \in \mathbb{N} \text{ divisible by } 3 \}.$$



This construction is based on the following idea:

For each prefix x read so far, the automaton stores in its states if the prefix is of even or odd length as well as the value of $\mathrm{Number}(x) \bmod 3$. Let $i \in \{0,1,2\}$. If the automaton reaches the state $q_{i\mathrm{e}}$ (or $q_{i\mathrm{o}}$, respectively) on a prefix, then the prefix is of even (or odd, respectively) length and represents a number congruent to i modulo 3.

By the definition of the binary representation, a word $a_1 a_2 \dots a_n \in \{0,1\}^*$ satisfies

$$\mathrm{Number}(a_1 \dots a_n) = 2 \cdot \mathrm{Number}(a_1 \dots a_{n-1}) + \mathrm{Number}(a_n)$$

and thus

$$\begin{aligned}\mathrm{Number}(a_1 \dots a_n) \bmod 3 &= (2 \cdot \mathrm{Number}(a_1 \dots a_{n-1}) + \mathrm{Number}(a_n)) \bmod 3\\ &= (2 \cdot \mathrm{Number}(a_1 \dots a_{n-1}) \bmod 3 + \mathrm{Number}(a_n)) \bmod 3\end{aligned}$$

This computation is implemented by the transitions of the automaton.

The empty word and all words starting by a 0 (except for the word 0) are not binary representations of any number and the word 0 is of odd length. Hence, the automaton reaches the trap state q_S on all words starting by a 0.

Solution to Bonus Exercise 1

We follow the approach in the proof of Theorem 2.4 in the German version of the textbook.[1] We seek to improve the lower bound on the number of prime numbers less than or equal to k from this theorem. The idea is to improve the self-delimiting encoding $\overline{\mathrm{Bin}}$ from the textbook. To this end, we define a new encoding $\widehat{\mathrm{Bin}_4}$ as follows: Let $m \in \mathbb{N} - \{0\}$, let $l = \lceil \log_2(m+1) \rceil$ be the length of the binary representation of m and let $\mathrm{Bin}(m) = a_1 a_2 \dots a_l$. Let $\alpha = -l \bmod 4$. Then let $\mathrm{Bin}_4(m) = 0^\alpha \mathrm{Bin}(m) = b_1 b_2 \dots b_{l'}$ be a modified binary representation of m with a number of leading zeros such that the total length $l' = l + \alpha$ is divisible by 4. Now we define

$$\widehat{\mathrm{Bin}_4}(m) = b_1 b_2 b_3 b_4\, 0\, b_5 b_6 b_7 b_8\, 0 \dots b_{l'-3} b_{l'-2} b_{l'-1} b_{l'}\, 1\,.$$

[1] Note that Theorem 2.72 in the English version claims a better result, but the proof given there does not take into account some necessary applications of the ceiling function in the calculations.

This means that we insert, after every 4 symbols in the binary representation $\mathrm{Bin}_4(m)$, a control bit such that a control bit 1 delimits the end of the encoding.

Analogously to the proof of Theorem 2.4, we can now represent a number n by the pair $(m, n/p_m)$ in which p_m is the largest prime factor of n. Then we represent the pair $(m, n/p_m)$ by

$$\mathrm{Word}(m, n/p_m) = \widehat{\mathrm{Bin}_4}(\lceil \log_2(\lceil \log_2(m+1)\rceil+1)\rceil)\mathrm{Bin}(\lceil \log_2(m+1)\rceil)\mathrm{Bin}(m)\mathrm{Bin}(n/p_m)\,.$$

Analogously to the proof of Theorem 2.4, we use a self-delimiting encoding to represent the length of the length of m, but we use our improved encoding.

This allows us to bound the length of our representation by

$$\begin{aligned}|\mathrm{Word}(m, n/p_m)| \leq \frac{5}{4} &\cdot (\lceil \log_2(\lceil \log_2(\lceil \log_2(m+1)\rceil + 1)\rceil + 1)\rceil + 3)\\ &+ \lceil \log_2(\lceil \log_2(m+1)\rceil + 1)\rceil\\ &+ \lceil \log_2(m+1)\rceil + \lceil \log_2((n/p_m)+1)\rceil\,.\end{aligned}$$

Up to 3 leading zeros in our self-delimiting encoding yield the term $+3$.

Following the argument in the proof of Theorem 2.4, we derive that there exist infinitely many natural numbers n such that

$$|\mathrm{Word}(m, n/p_m)| \geq \lceil \log_2(n+1)\rceil - 2 \tag{3}$$

and

$$K(n) \geq \lceil \log_2(n+1)\rceil - 2\,. \tag{4}$$

We plug in the inequality (3) into the previous bound on the length of $\mathrm{Word}(m, n/p_m)$ and, using bounds on the ceiling function analogously to the proof of Theorem 2.4, we obtain

$$\begin{aligned}\lceil \log_2(n+1)\rceil - 2 \leq\ & \frac{5}{4} \cdot (\lceil \log_2(\lceil \log_2(\lceil \log_2(m+1)\rceil + 1)\rceil + 1)\rceil + 3)\\ &+ \lceil \log_2(\lceil \log_2(m+1)\rceil + 1)\rceil\\ &+ \lceil \log_2(m+1)\rceil + \lceil \log_2((n/p_m)+1)\rceil\\ \leq\ & \frac{15}{4} + \frac{5}{4}\log_2\log_2\log_2 m + \log_2\log_2 m + \log_2 m + \log_2(n/p_m) + 15\,,\end{aligned}$$

and thus

$$\log_2 n \leq \frac{5}{4}\log_2\log_2\log_2 m + \log_2\log_2 m + \log_2 m + \log_2(n/p_m) + 21\,.$$

Analogously to the textbook, it follows that

$$p_m \leq 2^{21} \cdot m \cdot \log_2 m \cdot (\log_2\log_2 m)^{5/4}\,.$$

Using Lemma 2.69 from the textbook (Lemma 2.6 in the German version), we conclude from (4), analogously to the proof of Theorem 2.4, that

$$\mathrm{Prim}(k) \geq \frac{k}{2^{21} \cdot \log_2 k \cdot (\log_2\log_2 k)^{5/4}}$$

holds for infinitely many natural numbers $k \in \mathbb{N}$. The task statement is thus proved for $d = 2^{21}$.

Note: One can also prove the task statement by using the self-delimiting encoding from the proof of Theorem 2.4 to represent the length of the length of the length of m. This yields the bound

$$\mathrm{Prim}(k) \geq \frac{k}{d' \cdot \log_2 k \cdot \log_2 \log_2 k \cdot (\log_2 \log_2 \log_2 k)^2}$$

for a constant d', which implies the bound from the task statement for infinitely many natural numbers $k \in \mathbb{N}$.