

Unambiguous finite automata over a unary alphabet[☆]

Alexander Okhotin¹

Department of Mathematics, University of Turku, Turku FI-20014, Finland

Abstract

Nondeterministic finite automata (NFA) with at most one accepting computation on every input string are known as unambiguous finite automata (UFA). This paper considers UFAs over a one-letter alphabet, and determines the exact number of states in DFAs needed to represent unary languages recognized by n -state UFAs in terms of a new number-theoretic function \tilde{g} . The growth rate of $\tilde{g}(n)$, and therefore of the UFA–DFA tradeoff, is estimated as $e^{\Theta(\sqrt[3]{n \ln^2 n})}$. The conversion of an n -state unary NFA to a UFA requires UFAs with $g(n) + O(n^2) = e^{(1+o(1))\sqrt{n \ln n}}$ states, where $g(n)$ is the greatest order of a permutation of n elements, known as Landau’s function. In addition, it is shown that representing the complement of n -state unary UFAs requires UFAs with at least $n^{2-o(1)}$ states in the worst case, while the Kleene star requires up to exactly $(n - 1)^2 + 1$ states.

Key words: Finite automata, unary languages, ambiguity, descriptiveness, state complexity, Landau’s function

1. Introduction

This paper is concerned with a noteworthy family of automata located between deterministic finite automata (DFA) and nondeterministic finite automata (NFA): the *unambiguous finite automata* (UFA), that is, NFAs that have at most one accepting computation for every string. Apparently, this family was first studied by Schmidt [32], whose unpublished thesis contains an interesting method of proving lower bounds for UFAs based upon the rank of certain matrices, and a $2^{\Omega(\sqrt{n})}$ lower bound on the tradeoff between UFAs and DFAs. These methods were further elaborated by Leung [17, 18] and by Hromkovič et al. [11], who studied degrees of nondeterminism in finite automata. In particular, Leung [18] established a precise $2^n - 1$ UFA–DFA tradeoff. Computational complexity of testing properties of UFAs was studied by Stearns and Hunt [34] and recently by Björklund and Martens [4].

In the special case of a *unary alphabet* $\Sigma = \{a\}$, finite automata are known to have succinctness properties much different from the general case. Lyubich [19] and Chrobak [5] have shown that the DFA–NFA tradeoff in the unary case is $g(n) + O(n^2)$, where

$$g(n) = \max\{\text{lcm}(p_1, \dots, p_k) \mid k \geq 1, p_1 + \dots + p_k \leq n\} = e^{(1+o(1))\sqrt{n \ln n}}$$

is the maximum order of an element in the group of permutations of n objects, known as *Landau’s function* [16]. As a matter of fact, the periodic behaviour of an n -state NFA is exactly described by the definition of $g(n)$, while the initial non-periodic segment of the language contains strings of length at most n^2 . Succinctness of two-way automata over a unary alphabet has received particular attention in the works of

[☆]The results in this paper were first announced in Technical Report 951 of Turku Centre for Computer Science in September 2009 [25], and were presented at the MFCS 2010 conference held in Brno, Czech Republic, 23–27 August 2010 [26].

Email addresses: alexander.okhotin@utu.fi (Alexander Okhotin)

¹Supported by the Academy of Finland under grant 134860.

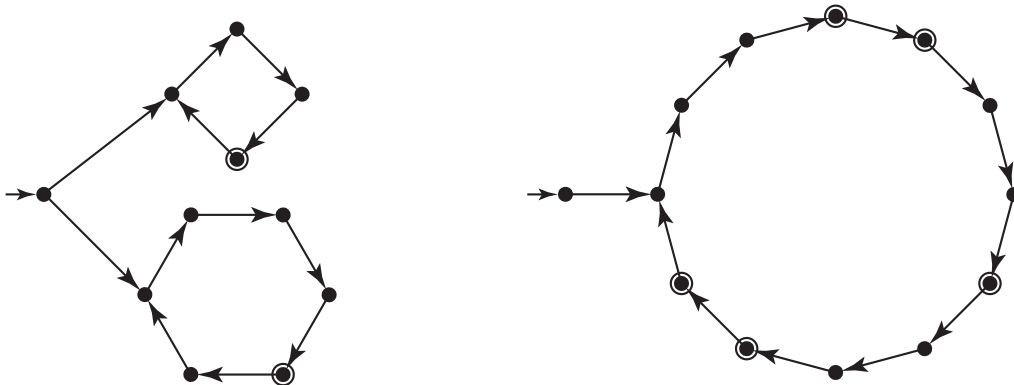


Figure 1: An 11-state unary UFA and the 13-state minimal equivalent DFA.

Chrobak [5], Mereghetti and Pighizzini [22], Geffert et al. [6] and Kunc and Okhotin [14, 15]. For a detailed survey of descriptive complexity of finite automata, the reader is referred to a paper by Holzer and Kutrib [10].

Turning to unary UFAs, their non-triviality was first noted by Ravikumar and Ibarra [29], who obtained a superpolynomial lower bound $e^{\frac{1}{8} \ln^{3/2} n}$ on the UFA–DFA tradeoff. An upper bound $g(n) + O(n^2)$ follows from the NFA–DFA tradeoff. However, these bounds are far apart, and neither of them is exact. This paper undertakes to establish the precise tradeoff between UFAs and DFAs, and to explain the combinatorial essence of unary UFAs in the same way as NFAs were explained by Lyubich [19] and Chrobak [5].

To begin *ab ovo*, consider the smallest example of a non-trivial UFA, presented in Figure 1, left. This automaton is unambiguous, because only strings of even length are accepted in the first cycle, and only strings of odd length are accepted in the second cycle. The UFA has $1 + 4 + 6 = 11$ states, while the smallest equivalent DFA shown on the right requires $1 + \text{lcm}(4, 6) = 13$ states.

This example is unlike the known lower bound examples for unary NFAs and for two-way automata [5], in which the cycles are chosen to be relatively prime, in order to maximize their least common multiple. For instance, an 11-state NFA can use three cycles of length 2, 3 and 5, as well as a dedicated initial state, so that any equivalent DFA would require $1 + \text{lcm}(2, 3, 5) = 31$ states. This maximizes the least common multiple, as in the definition of Landau’s function: consider that $g(10) = g(11) = \text{lcm}(2, 3, 5)$. However, such an NFA is bound to be ambiguous. In contrast, in the example given in Figure 1, the common divisor 2 of the lengths of the cycles reduces the value of their least common multiple, but this is necessary to ensure the unambiguity of the automaton.

The above reasoning can be extended to unary UFAs in general. First, an arbitrary unary NFAs is transformed to the *Chrobak normal form* [5], in which there is one tail of states, ending with transitions into one or more disjoint cycles. As proved by Jiang, McDowell and Ravikumar [13, Thm. 2.2], any UFA can be transformed to this normal form without increasing the number of states, and hence, for all state complexity purposes, one can consider only automata of this form. Furthermore, as established in Section 2, the accepting states in a normal form UFA have to obey the same kind of restriction as in Figure 1: for every two accepting states from two different cycles, their offsets must be distinct modulo the greatest common divisor of the lengths of these cycles. This requirement can be embedded in the definition of Landau’s function, leading to the following new variant of this function:

$$\tilde{g}(n) = \max \left\{ \text{lcm}(p_1, \dots, p_k) \mid k \geq 1, p_1 + \dots + p_k \leq n, \right. \\ \left. \text{there exist such offsets } f_1, \dots, f_k \text{ with } f_i \in \{0, \dots, p_i - 1\}, \text{ that} \right. \\ \left. \text{for all } i, j \text{ (with } i \neq j), f_i \not\equiv f_j \pmod{\text{gcd}(p_i, p_j)} \right\},$$

where f_1, \dots, f_k stand for the positions of some accepting states of a UFA in their respective cycles. In the next Section 3, the worst case of the UFA-to-DFA transformation is reformulated in terms of $\tilde{g}(n)$ as follows:

transforming an n -state UFA with a unique initial state to a DFA requires $\max_{1 \leq \ell < n} \tilde{g}(n - \ell) + \ell$ states in the worst case, and if a UFA may have multiple initial states, the tradeoff function equals $\max_{0 \leq \ell < n} \tilde{g}(n - \ell) + \ell$. Both functions are asymptotically equivalent to $\tilde{g}(n)$,

The growth rate of $\tilde{g}(n)$ is studied in Section 4. Reaching the maximum value of $\text{lcm}(p_1, \dots, p_k)$ under the conditions in the definition of $\tilde{g}(n)$ is an optimization problem of balancing the following two requirements. On the one hand, the cycle lengths p_1, \dots, p_k should have as few common divisors as possible, so that their least common multiple is greater. On the other hand, having common multiples and letting them be sufficiently large is necessary for the offsets f_1, \dots, f_k to be distinct modulo those common multiples. The key part of the given estimation is showing that the condition of the existence of such f_1, \dots, f_k implies the inequality $p_1 + \dots + p_k > \frac{4}{9}k^3 \ln k - \frac{8}{27}k^3 \sqrt{\ln k}$. The latter inequality is then used to establish an upper bound $\tilde{g}(n) \leq e^{(1+o(1))\sqrt[3]{2}\sqrt[3]{n \ln^2 n}}$. Inferring a fairly close lower bound $\tilde{g}(n) \geq e^{(1+o(1))\sqrt[3]{\frac{2}{9}}\sqrt[3]{n \ln^2 n}}$ is a relatively simple task, carried out by choosing each p_i to be k times the i -th prime, and using the well-known estimations of the sum and the product of the first k primes. This leads to the approximation of $\tilde{g}(n)$ as $2^{\Theta(\sqrt[3]{n \ln^2 n})}$, and this approximation also applies to the UFA–DFA tradeoff for unary languages.

A close lower bound on the tradeoff between NFAs and UFAs is established in the next Section 5, using the matrix methods of Schmidt [32]. The tradeoff is found to be of the order of the original Landau’s function, that is, $e^{(1+o(1))\sqrt{n \ln n}}$.

The question of how the basic operations on languages affect the number of states in unary UFAs is approached in Sections 6–7. The state complexity of operations on unary DFAs was first studied by Yu, Zhuang and K. Salomaa [36], and later elaborated by Pighizzini and Shallit [27]; similar questions for unary NFAs were answered by Holzer and Kutrib [9]. In this paper, the complexity of complementing UFAs is addressed in Section 6, and a family of such n -state unary UFAs is presented, that any UFAs for their complements require at least $n^{2-o(1)}$ states. This for the first time shows that the complement of a UFA sometimes requires additional states (which is an unsettled problem mentioned by Hromkovič et al. [11]). In the last Section 7, the methods of Yu et al. [36] are applied to show that the Kleene star of an n -state UFA can be represented by a UFA with $(n - 1)^2 + 1$ states, and that this number of states is necessary in the worst case.

2. Chrobak normal form for unambiguous automata

A *nondeterministic finite automaton* (NFA) is a quintuple $A = (\Sigma, Q, Q_0, \delta, F)$, where Σ is an input alphabet, Q is a finite non-empty set of states; $Q_0 \subseteq Q$ is the set of initial states; $\delta : Q \times \Sigma \rightarrow 2^Q$ is the transition function; $F \subseteq Q$ is the set of accepting states. The automaton A is said to accept a string $w = a_1 \dots a_n$ if there exists a sequence of states $r_0, \dots, r_n \in Q$, in which $r_0 \in Q_0$, $r_i \in \delta(r_{i-1}, a_i)$ for all i , and $r_n \in F$. The language recognized by an NFA, denoted by $L(A)$, is the set of all strings it accepts. The transition function is extended to $\delta : 2^Q \times \Sigma^* \rightarrow 2^Q$ by $\delta(q, \varepsilon) = \{q\}$, $\delta(q, aw) = \bigcup_{q' \in \delta(q, a)} \delta(q', w)$ and $\delta(S, w) = \bigcup_{q \in S} \delta(q, w)$.

In some literature, NFAs are defined with a unique initial state, that is, with $Q_0 = \{q_0\}$. Every NFA can be converted to an NFA with a unique initial state by adding a new initial state.

A *deterministic finite automaton* (DFA) is an NFA with a unique outgoing transition from each state by each symbol ($|\delta(q, a)| = 1$ for all q, a) and with a unique initial state ($|Q_0| = 1$). An NFA A is a *partial DFA*, if $|Q_0| = 1$ and $|\delta(q, a)| \leq 1$ for all q and a . All these variants of finite automata define the same family of languages, known as the *regular languages*.

An intermediate family of *unambiguous finite automata* (UFA) is defined as follows. An NFA is said to be *unambiguous*, if for every string $w \in L(A)$, the corresponding sequence of states $r_0, \dots, r_{|w|}$ in the definition of acceptance is unique.

The first lower bound argument for UFAs was given by Schmidt [32, Thm. 3.9] in his proof of a $2^{\Omega(\sqrt{n})}$ lower bound on the NFA–UFA tradeoff. The following general statement of Schmidt’s lower bound method is due to Leung [18]:

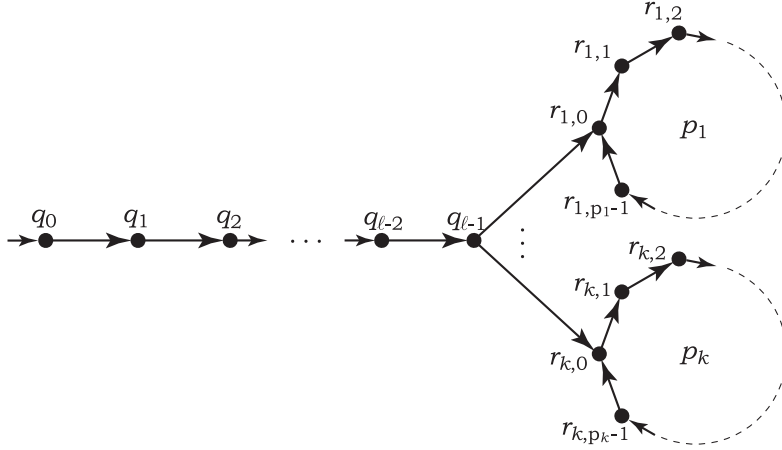


Figure 2: Chrobak normal form of a unary NFA.

Schmidt's Theorem [32, 18]. *Let $L \subseteq \Sigma^*$ be a regular language over any finite alphabet Σ , and let $\{(u_1, v_1), \dots, (u_n, v_n)\}$ with $n \geq 1$ and $u_i, v_i \in \Sigma^*$ be a finite set of pairs of strings. Consider the integer matrix $M \in \mathbb{Z}^{n \times n}$, defined by*

$$M_{i,j} = \begin{cases} 1, & \text{if } u_i v_j \in L; \\ 0, & \text{otherwise.} \end{cases}$$

Then every UFA recognizing L has at least rank M states.

This theorem can be effectively applied to some particular languages and well-chosen sets of pairs, for which the matrix is of such a fortunate form, that its rank is evident: this was done by Schmidt himself [32], by Leung [18] and by Hromkovič et al. [11], and this method shall be employed again in Section 5 of the present paper. However, Schmidt's Theorem cannot be used as a general method of determining the state complexity of an arbitrary given language. Even in the case of a unary alphabet, the matrix $M_{i,j}$ belongs to the class of *circulant matrices*, and the problem of determining the rank of a circulant matrix of 0s and 1s, studied by Ingleton [12], is surprisingly hard in the general case.

This paper considers finite automata over a *unary alphabet* $\Sigma = \{a\}$. Regular languages over a unary alphabet are equivalent to ultimately periodic sets of natural numbers: that is, for every regular $L \subseteq a^*$, there exist such integers $\ell \geq 0$ and $p \geq 1$, that $a^n \in L$ if and only if $a^{n+p} \in L$ for all $n \geq \ell$; the least such values of ℓ and p are known as the *tail* and the *period* of the language L .

The study of UFAs over a unary alphabet undertaken in this paper begins with the following normal form of NFAs.

Definition 1 (Chrobak [5]). *An NFA over the alphabet $\Sigma = \{a\}$ is said to be in Chrobak normal form, if its set of states is $\{q_0, \dots, q_{\ell-1}\} \cup \bigcup_{i=1}^k R_i$, with $\ell \geq 0$, $k \geq 0$, $R_i = \{r_{i,0}, \dots, r_{i,p_i-1}\}$ and $1 \leq p_1 < p_2 < \dots < p_k$, the unique initial state is q_0 if $\ell \geq 1$, or there is a set of initial states $\{r_{1,0}, \dots, r_{k,0}\}$ if $\ell = 0$, and the transitions are:*

$$\begin{aligned} \delta(q_i, a) &= \{q_{i+1}\} & (0 \leq i \leq \ell - 2), \\ \delta(q_{\ell-1}, a) &= \{r_{1,0}, r_{2,0}, \dots, r_{k,0}\} & (\text{if } \ell \geq 1), \\ \delta(r_{i,j}, a) &= \{r_{i,j+1 \bmod p_i}\} & (1 \leq i \leq k, 0 \leq j \leq p_i - 1). \end{aligned}$$

The set of accepting states may be arbitrary.

The states $\{q_0, \dots, q_{\ell-1}\}$ are called the tail of the NFA, and each R_i is called a cycle.

It is known from Chrobak [5] that every NFA with n states can be transformed to an equivalent NFA in this normal form, with the tail of length $\ell = O(n^2)$ and with $\sum_{i=1}^k p_i \leq n$ total states in the cycles. The growth in the number of states is thus at most quadratic.

In the special case of *finite* unary languages, NFAs can be transformed to a much simpler form without increasing the number of their states:

Proposition 1 (Mandl [20]). *For every NFA recognizing a finite language over a one-letter alphabet there exists a partial DFA with the same number of states recognizing the same language.*

Turning to unary UFAs, in this case the transformation to the Chrobak normal form can always be done without increasing the number of states:

Proposition 2 (Jiang, McDowell, Ravikumar [13, Thm. 2.2]). *For every UFA over a unary alphabet there exists (and can be effectively constructed) a UFA in Chrobak normal form with the same number of states recognizing the same language. Furthermore, if the original UFA has a unique initial state, then so does the resulting UFA.*

Once a UFA is converted to the Chrobak normal form, the following key restriction of unambiguous automata is exposed:

Criterion of Unambiguity. *An NFA $(\{a\}, Q, q_0, \delta, F)$ in Chrobak normal form recognizing an infinite language over a unary alphabet is unambiguous if and only if for every two accepting states $r_{i,f}, r_{j,f'} \in F$ with $i \neq j$, the offsets f and f' are different modulo $\gcd(p_i, p_j)$.*

The same property was independently established by Anselmo and Madonia [1, Prop. 7], who presented it for arbitrary automata, not necessarily in the Chrobak normal form.

The proof uses the Chinese Remainder Theorem in the following formulation:

Chinese Remainder Theorem. *Let $p, p' \geq 1$ and $i, i' \geq 0$ be any integers with $i \equiv i' \pmod{\gcd(p, p')}$. Then there exists an integer $n \geq 0$ with $n \equiv i \pmod{p}$ and $n \equiv i' \pmod{p'}$.*

Proof of the Criterion of Unambiguity. \ominus Let the automaton be unambiguous and suppose there exist two states $r_{i,f}, r_{j,f'} \in F$ with $i \neq j$ and $f \equiv f' \pmod{\gcd(p_i, p_j)}$. The latter condition makes a generalized version of the Chinese Remainder Theorem applicable to f, f', p_i and p_j , and it asserts that there exists a number $n \geq 0$ with $n \equiv f \pmod{p_i}$ and $n \equiv f' \pmod{p_j}$. Then the string $a^{\ell+n}$ has two accepting computations, one in the component R_i and the other in R_j , which contradicts the assumption that the automaton is unambiguous.

\ominus Assume that the conditions on accepting states hold, and suppose that the automaton is ambiguous. Then there is a string $a^{\ell+n}$ with $n \geq 0$, accepted in two different cycles, R_i and R_j ; more precisely, in some states $r_{i,f}$ and $r_{j,f'}$. Accordingly, $n \equiv f \pmod{p_i}$ and $n \equiv f' \pmod{p_j}$, and therefore $f \equiv n \equiv f' \pmod{\gcd(p_i, p_j)}$, which contradicts the condition. \square

The Criterion of Unambiguity, in particular, implies that the lengths of the cycles cannot be primes (unless there is a unique cycle), and that $\gcd(p_i, p_j) \geq 2$ for any two distinct cycles. For example, the UFA in Figure 1 in the introduction has $\gcd(4, 6) = 2$, and accepting states are separated by the parity of their offsets.

3. UFA–DFA tradeoff

An upper bound on the number of states in a DFA needed to represent a unary language recognized by an n -state NFA has been established by Lyubich [19]. It is asymptotically equivalent to the maximum order of a permutation of n elements:

$$g(n) = \max\{\text{lcm}(p_1, \dots, p_k) \mid k \geq 1, p_1 + \dots + p_k \leq n\}.$$

This function is known as *Landau's function*, as its $e^{(1+o(1))\sqrt{n \ln n}}$ growth rate was determined by Landau [16], see also Miller [23] for a more accessible argument and Szalay [35] for an even more precise estimation.

Twenty years after Lyubich, an asymptotically matching lower bound on the unary NFA to DFA tradeoff was obtained by Chrobak [5], who also gave a new, combinatorial proof of Lyubich's upper bound. These results can be stated as follows:

Proposition 3 (Lyubich [19], Chrobak [5]). *For every n -state unary NFA there exists a DFA recognizing the same language, with the tail of length at most $n^2 + n$ and the cycle of length at most $g(n)$. Conversely, for every n there is a language recognized by an n -state NFA, such that every equivalent DFA must have a cycle of length $g(n)$.*

The essence of this result is a natural correspondence between unary NFAs and Landau's function. The numbers p_1, \dots, p_k in the definition of $g(n)$ correspond to lengths of cycles of an NFA in Chrobak normal form, the sum $p_1 + \dots + p_k$ represents the number of states in an NFA, and an equivalent DFA has to have $\text{lcm}(p_1, \dots, p_k)$ states.

This analysis of NFAs can be extended to UFAs, if the additional constraints on their Chrobak normal form given in the Criterion of Unambiguity are embedded into the definition of Landau's function. This leads to the following variant of this function:

$$\tilde{g}(n) = \max \left\{ \text{lcm}(p_1, \dots, p_k) \mid \begin{array}{l} k \geq 1, p_1 + \dots + p_k \leq n, \\ \text{there exist such offsets } f_1, \dots, f_k \text{ with } f_i \in \{0, \dots, p_i - 1\}, \text{ that} \\ \text{for all } i, j \text{ (with } i \neq j), f_i \not\equiv f_j \pmod{\text{gcd}(p_i, p_j)} \end{array} \right\},$$

For n up to 9, the value of $\tilde{g}(n)$ is n . The next value is $\tilde{g}(10) = 12$, given by $k = 2, p_1 = 4, p_2 = 6, f_1 = 0$ and $f_2 = 1$ with $0 \not\equiv 1 \pmod{\text{gcd}(4, 6)}$. The growth rate of this function can be estimated as $e^{\Theta(\sqrt[3]{n \ln^2 n})}$, and this estimation will be the subject of the next section. Now the task is to express the tradeoff between UFAs and DFAs using this function, which is done in the following theorem.

Theorem 1. *For every $n \geq 1$, the following number of states is sufficient and in the worst case necessary for a DFA to recognize a unary language recognized by an n -state UFA with multiple initial states:*

$$f_{\text{UFA-DFA}}(n) = \begin{cases} n + 1, & \text{if } n \leq 9 \\ \max_{0 \leq \ell < n} \tilde{g}(n - \ell) + \ell, & \text{if } n \geq 10 \end{cases}$$

For UFAs with a unique initial state, the tradeoff function takes the following form:

$$f_{\text{UFA}_1\text{-DFA}}(n) = \begin{cases} n + 1, & \text{if } n \leq 10 \\ \max_{1 \leq \ell < n} \tilde{g}(n - \ell) + \ell, & \text{if } n \geq 11 \end{cases}$$

For $n \leq 9$, Theorem 1 states that UFAs are not yet any more powerful than partial DFAs, and thus can be simulated by DFAs with $n + 1$ states, with the lower bound witnessed by a finite language. Once there are sufficiently many states to reach the first non-trivial values of \tilde{g} , one can encode the periods and the offsets from the definition of \tilde{g} within a witness language; this is done in Lemma 1 below, which establishes the lower bounds on $f_{\text{UFA-DFA}}(n)$ and $f_{\text{UFA}_1\text{-DFA}}(n)$. The matching upper bounds are given in the next Lemma 2. These results are put together in the proof of Theorem 1, presented after Lemmata 1–2.

The first lemma gives a lower bound on the UFA–DFA tradeoff by constructing a witness UFA for all appropriate n and ℓ , so that every DFA for the same language would require $\tilde{g}(n - \ell) + \ell$ states.

Lemma 1. *Let $k \geq 2, \ell \geq 0, p_1, \dots, p_k \geq 2$ and $f_1, \dots, f_k \geq 0$ with $0 \leq f_i < p_i$ be any integers satisfying the following three conditions:*

- (a). $f_i \not\equiv f_j \pmod{\text{gcd}(p_i, p_j)}$ for all i, j with $1 \leq i < j \leq k$,
- (b). $\text{lcm}(p_1, \dots, p_{i-1}, p_{i+1}, \dots, p_k)$ is not divisible by p_i for all i with $1 \leq i \leq k$, and
- (c). $f_i = p_i - 1$ for some i .

Then the language

$$L = a^\ell \cdot \bigcup_{i=1}^k a^{f_i} (a^{p_i})^*$$

has a UFA with $\ell + p_1 + \dots + p_k$ states, while the smallest DFA for this language has $\ell + \text{lcm}(p_1, \dots, p_m)$ states.

Proof. The construction of a UFA in Chrobak normal form recognizing L is entirely obvious: it has a tail of length ℓ and cycles of length p_1, \dots, p_k , each with a unique accepting state at the offset f_i . As $f_i \not\equiv f_j \pmod{\text{gcd}(p_i, p_j)}$ by assumption, the condition of the Criterion of Unambiguity is satisfied.

Let $p = \text{lcm}(p_1, \dots, p_k)$ and consider a DFA with the tail of length ℓ and the cycle of length p , which recognizes the language L . To see that there is no smaller DFA for L , it is sufficient to prove that for every two distinct states $q = \delta(q_0, a^m)$ and $q' = \delta(q_0, a^{m'})$, with $0 \leq m < m' < \ell + p$, there exists a string accepted from one of these states and not accepted from the other. If $m' - m \equiv 0 \pmod{p}$, then $m < \ell$, and the string $a^{\ell-1-m}$ is not accepted from q , for the reason that $a^{\ell-1} \notin L$. At the same time, $a^{\ell-1-m}$ is accepted from q' , because $a^{\ell+p-1} \in L$ by the condition (c).

It remains to consider the case of $m' - m \not\equiv 0 \pmod{p}$. Then the length of one of the cycles in the UFA does not divide $m' - m$; assume, without loss of generality, that $m' - m$ is not divisible by p_1 . It is claimed that there exists a number $n \in \{0, \dots, p-1\}$ equivalent to $f_1 + m' - m$ modulo p_1 , such that the string $a^{\ell+p+n-(m'-m)}$ is in L , but $a^{\ell+p+n} \notin L$. This would prove the statement, because the string $a^{\ell+p+n-m'}$ is then accepted from q and rejected from q' .

Suppose, for the sake of contradiction, that there is no such number. Then, for every number n equivalent to $n_1 = f_1 + m' - m$ modulo p_1 , the string $a^{\ell+n}$ is in L . Let

$$L_i = a^\ell \cdot a^{f_i} \cdot (a^{p_i})^*,$$

so that $L = L_1 \cup \dots \cup L_k$. Since $m' - m$ is not divisible by p_1 , $m' - m \not\equiv 0 \pmod{p_1}$, hence $n_1 \not\equiv f_1 \pmod{p_1}$, and accordingly $a^{\ell+n} \in L_2 \cup \dots \cup L_k$. A contradiction is derived by applying the following statement $k-1$ times:

Claim 1. *Let $2 \leq i \leq k$ and let n_{i-1} be a number with $0 \leq n_{i-1} < \text{lcm}(p_1, \dots, p_{i-1})$. Assume that $a^{\ell+n} \in L_i \cup L_{i+1} \cup \dots \cup L_k$ for all $n \geq 0$ equivalent to n_{i-1} modulo $\text{lcm}(p_1, \dots, p_{i-1})$. Then there exists a number n_i with $0 \leq n_i < \text{lcm}(p_1, \dots, p_{i-1}, p_i)$, such that $a^{\ell+n} \in L_{i+1} \cup \dots \cup L_k$ for every number $n \geq 0$ equivalent to n_i modulo $\text{lcm}(p_1, \dots, p_{i-1}, p_i)$.*

Indeed, the first application of the claim, for $i = 2$, gives a number n_2 , such that $a^{\ell+n} \in L_3 \cup \dots \cup L_k$ for every n with $n \equiv n_2 \pmod{\text{lcm}(p_1, p_2)}$, the second application yields n_3 with $a^{\ell+n} \in L_4 \cup \dots \cup L_k$ for $n \equiv n_3 \pmod{\text{lcm}(p_1, p_2, p_3)}$, and so on. Finally, for $i = k$ the claim leads to the conclusion that there is a number n_k , such that $a^{\ell+n_k} \in \emptyset$, which is a contradiction.

It remains to prove the claim. Consider two numbers, n_{i-1} and $n_{i-1} + \text{lcm}(p_1, \dots, p_{i-1})$. It is known that $\text{lcm}(p_1, \dots, p_{i-1})$ is non-zero modulo p_i (otherwise p_i would divide $\text{lcm}(p_1, \dots, p_{i-1})$, contradicting assumption (b)). Then $n_{i-1} \not\equiv n_{i-1} + \text{lcm}(p_1, \dots, p_{i-1}) \pmod{p_i}$, and therefore at least one of these numbers must be different from f_i modulo p_i ; denote this number by n_i .

Since $n_i \equiv n_{i-1} \pmod{\text{lcm}(p_1, \dots, p_{i-1})}$, all numbers equivalent to n_i modulo $\text{lcm}(p_1, \dots, p_i)$ are equivalent to n_{i-1} modulo $\text{lcm}(p_1, \dots, p_{i-1})$, and thus, for every such number n , the string $a^{\ell+n}$ must be in $L_i \cup L_{i+1} \cup \dots \cup L_k$ by assumption. But since none of these numbers are equivalent to f_i modulo p_i , none of the corresponding strings belong to L_i . Therefore, all these strings are in $L_{i+1} \cup \dots \cup L_k$, which proves the claim and completes the proof of the lemma. \square

The matching upper bound is implied by the following lemma. This lemma performs a straightforward deconstruction of a UFA, from which one can extract suitable values of periods and offsets matching the definition of \tilde{g} .

Lemma 2. *For every n -state UFA in Chrobak normal form with a tail of length $\ell \geq 0$, there exists a DFA with at most $\ell + \tilde{g}(n - \ell)$ states recognizing the same language.*

Proof. Let p_1, \dots, p_k be the lengths of the cycles in this UFA. Then it is well-known that there is an equivalent DFA with $\text{lcm}(p_1, \dots, p_k) + \ell$ states [5, Thm. 4.4].

Consider one accepting state from each cycle: $r_{1,f_1}, r_{2,f_2}, \dots, r_{k,f_k} \in F$. By the Criterion of Unambiguity, $f_i \not\equiv f_j \pmod{\text{gcd}(p_i, p_j)}$ for all $i \neq j$. Then these numbers satisfy the definition of \tilde{g} , and accordingly $\text{lcm}(p_1, \dots, p_k) \leq \tilde{g}(n - \ell)$, which shows that the above DFA has at most $\tilde{g}(n - \ell) + \ell$ states. \square

The theorem is now established as a consequence of the above lemmata.

Proof of Theorem 1. Note that $\tilde{g}(10) = \text{lcm}(4, 6) = 12$, and therefore, for every $n \geq 11$, $\tilde{g}(n - \ell) + \ell > n + 1$ for $\ell = n - 10$. Furthermore, the numbers 4 and 6 are the smallest two numbers with a common divisor and with their least common multiple larger than either of them, and accordingly, $\tilde{g}(n) = n$ for $n < 10$. Then the function stated in the theorem can be equivalently expressed as follows:

$$f_{\text{UFA-DFA}}(n) = \max(n + 1, \max_{0 \leq \ell < n} \tilde{g}(n - \ell) + \ell).$$

The first claim is that every n -state unary UFA can be transformed to an equivalent DFA with $f_{\text{UFA-DFA}}(n)$ states. If the UFA recognizes a finite language, then, by Proposition 1, this language is recognized by an n -state partial DFA, and hence by an $(n + 1)$ -state complete DFA. If the language recognized by the UFA is infinite, then, according to Proposition 2, one can assume that the UFA is in Chrobak normal form; let ℓ be the length of the tail. Then a DFA with $\tilde{g}(n - \ell) + \ell$ states recognizing the same language exists due to Lemma 2. In both cases, the number of states is at most $f_{\text{UFA-DFA}}(n)$.

To prove the **lower bound**, fix $n \geq 1$. The language $\{a^{n-1}\}$ has a partial DFA (and hence a UFA) with n states, but every complete DFA for this language requires $n + 1$ states, and therefore $f_{\text{UFA-DFA}}(n) \geq n + 1$. It remains to prove that $f_{\text{UFA-DFA}}(n) \geq \tilde{g}(n - \ell) + \ell$ for every $\ell \in \{1, \dots, n - 1\}$.

Choose ℓ , so that the number $\tilde{g}(n - \ell) + \ell$ is the greatest possible, and consider the number $\tilde{g}(n - \ell)$, which is given by $\text{lcm}(p_1, \dots, p_k)$ for some $k \geq 1$, $p_1, \dots, p_k \geq 2$ and $f_1, \dots, f_k \geq 0$ with $p_1 + \dots + p_k \leq n - \ell$ and $f_i \not\equiv f_j \pmod{\text{gcd}(p_i, p_j)}$ for all $i \neq j$. Furthermore, the number $\text{lcm}(p_1, \dots, p_k)$ is by definition *the greatest* among all numbers k , p_i and f_i meeting the above constraints.

It is claimed that every cycle length p_i contributes something to the least common multiple, that is, $\text{lcm}(p_1, \dots, p_{i-1}, p_{i+1}, \dots, p_k)$ is not divisible by p_i . Indeed, if $\text{lcm}(p_1, \dots, p_{i-1}, p_{i+1}, \dots, p_k)$ is a multiple of p_i , then $\text{lcm}(p_1, \dots, p_{i-1}, p_{i+1}, \dots, p_k) = \text{lcm}(p_1, \dots, p_k)$, and accordingly $\tilde{g}(p_1 + \dots + p_k) = \tilde{g}(p_1 + \dots + p_k - p_i)$, which implies that $\tilde{g}(n - \ell - p_i) + \ell + p_i > \tilde{g}(n - \ell) + \ell$. Then $\ell' = \ell + p_i$ leads to a greater value $\tilde{g}(n - \ell') + \ell'$, which contradicts the choice of ℓ .

The next claim is that the offsets f_1, \dots, f_k can be adjusted, so that $f_1 = p_1 - 1$. It is sufficient to add the number $p_1 - f_1 - 1$ to all offsets, that is, to redefine the offsets as $f'_i = f_i + p_1 - f_1 - 1$. The condition $f'_i \not\equiv f'_j \pmod{\text{gcd}(p_i, p_j)}$ is preserved, because $f'_i - f'_j \equiv f_i - f_j \pmod{\text{gcd}(p_i, p_j)}$.

It has thus been demonstrated that all conditions of Lemma 1 are satisfied, and hence there exists a language representable by an n -state UFA, for which every DFA must have $\text{lcm}(p_1, \dots, p_k) + \ell = \tilde{g}(n - \ell) + \ell$ states. \square

The values of $\tilde{g}(n)$ for small values of n , calculated by an exhaustive search, are given in Table 1, along with the computed lengths of cycles p_1, \dots, p_k . The next columns of the table give the precise number of states in a DFA needed to simulate an n -state UFA over a unary alphabet, as well as witness languages, on which this bound is reached. The last two columns contain similar results for UFAs with a unique initial state. These languages and their state complexity are determined on the basis of the values of $\tilde{g}(n)$ according to Lemma 1.

4. Estimations of \tilde{g}

The function \tilde{g} characterizes the expressive power of unary UFAs, and estimating the growth rate of this function, especially in comparison with g , is essential to understand the power of ambiguity in finite automata over a one-letter alphabet. The values of these two functions for small values of their argument

n	$\tilde{g}(n)$	UFA to DFA		UFA ₁ to DFA	
		$f(n)$	witness language	$f_1(n)$	witness language
1	1	2	$\{\varepsilon\}$	2	$\{\varepsilon\}$
2	2	3	$\{a\}$	3	$\{a\}$
3	3	4	$\{a^2\}$	4	$\{a^2\}$
4	4	5	$\{a^3\}$	5	$\{a^3\}$
5	5	6	$\{a^4\}$	6	$\{a^4\}$
6	6	7	$\{a^5\}$	7	$\{a^5\}$
7	7	8	$\{a^6\}$	8	$\{a^6\}$
8	8	9	$\{a^7\}$	9	$\{a^7\}$
9	9	10	$\{a^8\}$	10	$\{a^8\}$
10	12 = lcm(4,6)	12	$a^3(a^4) \cup a^4(a^6)^*$	11	$\{a^9\}$
11	12 = lcm(4,6)	13	$a^4(a^4) \cup a^5(a^6)^*$	13	$a^4(a^4)^* \cup a^5(a^6)^*$
12	12 = lcm(4,6)	14	$a^5(a^4) \cup a^6(a^6)^*$	14	$a^5(a^4)^* \cup a^6(a^6)^*$
13	13	15	$a^6(a^4) \cup a^7(a^6)^*$	15	$a^6(a^4)^* \cup a^7(a^6)^*$
14	24 = lcm(6,8)	24	$a^5(a^6) \cup a^6(a^8)^*$	16	$a^7(a^4)^* \cup a^8(a^6)^*$
15	24 = lcm(6,8)	25	$a^6(a^6) \cup a^7(a^8)^*$	25	$a^6(a^6)^* \cup a^7(a^8)^*$
16	30 = lcm(6,10)	30	$a^5(a^6) \cup a^6(a^{10})^*$	26	$a^7(a^6)^* \cup a^8(a^8)^*$
17	30 = lcm(6,10)	31	$a^6(a^6) \cup a^7(a^{10})^*$	31	$a^6(a^6)^* \cup a^7(a^{10})^*$
18	40 = lcm(8,10)	40	$a^7(a^8) \cup a^8(a^{10})^*$	32	$a^7(a^6)^* \cup a^8(a^{10})^*$
19	40 = lcm(8,10)	41	$a^8(a^8) \cup a^9(a^{10})^*$	41	$a^8(a^8)^* \cup a^9(a^{10})^*$
20	42 = lcm(6,14)	42	$a^5(a^6) \cup a^6(a^{14})^*$	42	$a^9(a^8)^* \cup a^{10}(a^{10})^*$
21	42 = lcm(6,14)	43	$a^6(a^6) \cup a^7(a^{14})^*$	43	$a^6(a^6)^* \cup a^7(a^{14})^*$
22	60 = lcm(10,12)	60	$a^9(a^{10}) \cup a^{10}(a^{12})^*$	44	$a^7(a^6)^* \cup a^8(a^{14})^*$
23	60 = lcm(10,12)	61	$a^{10}(a^{10}) \cup a^{11}(a^{12})^*$	61	$a^{10}(a^{10})^* \cup a^{11}(a^{12})^*$
24	70 = lcm(10,14)	70	$a^9(a^{10}) \cup a^{10}(a^{14})^*$	62	$a^{11}(a^{10})^* \cup a^{12}(a^{12})^*$
25	70 = lcm(10,14)	71	$a^{10}(a^{10}) \cup a^{11}(a^{14})^*$	71	$a^{10}(a^{10})^* \cup a^{11}(a^{14})^*$
26	84 = lcm(12,14)	84	$a^{11}(a^{12}) \cup a^{12}(a^{14})^*$	72	$a^{11}(a^{10})^* \cup a^{12}(a^{14})^*$
27	84 = lcm(12,14)	85	$a^{12}(a^{12}) \cup a^{13}(a^{14})^*$	85	$a^{12}(a^{12})^* \cup a^{13}(a^{14})^*$
28	90 = lcm(10,18)	90	$a^9(a^{10}) \cup a^{10}(a^{18})^*$	86	$a^{13}(a^{12})^* \cup a^{14}(a^{14})^*$
29	90 = lcm(10,18)	91	$a^{10}(a^{10}) \cup a^{11}(a^{18})^*$	91	$a^{10}(a^{10})^* \cup a^{11}(a^{18})^*$
30	120 = lcm(8,10,12)	120	$a^7(a^8) \cup a^8(a^{10}) \cup a^9(a^{12})^*$	92	$a^{11}(a^{10})^* \cup a^{12}(a^{18})^*$
31	120 = lcm(8,10,12)	121	$a^8(a^8) \cup a^9(a^{10}) \cup a^{10}(a^{12})^*$	121	$a^8(a^8)^* \cup a^9(a^{10})^* \cup a^{10}(a^{12})^*$
32	126 = lcm(14,18)	126	$a^{13}(a^{14}) \cup a^{14}(a^{18})^*$	122	$a^9(a^8)^* \cup a^{10}(a^{10})^* \cup a^{11}(a^{12})^*$
33	126 = lcm(14,18)	127	$a^{14}(a^{14}) \cup a^{15}(a^{18})^*$	127	$a^{14}(a^{14})^* \cup a^{15}(a^{18})^*$
34	168 = lcm(8,12,14)	168	$a^7(a^8) \cup a^9(a^{12}) \cup a^8(a^{14})^*$	128	$a^{15}(a^{14})^* \cup a^{16}(a^{18})^*$
35	168 = lcm(8,12,14)	169	$a^8(a^8) \cup a^{10}(a^{12}) \cup a^9(a^{14})^*$	169	$a^8(a^8)^* \cup a^{10}(a^{12})^* \cup a^9(a^{14})^*$
36	180 = lcm(9,12,15)	180	$a^8(a^9) \cup a^9(a^{12}) \cup a^{10}(a^{15})^*$	170	$a^9(a^8)^* \cup a^{11}(a^{12})^* \cup a^{10}(a^{14})^*$
37	180 = lcm(9,12,15)	181	$a^9(a^9) \cup a^{10}(a^{12}) \cup a^{11}(a^{15})^*$	181	$a^9(a^9)^* \cup a^{10}(a^{12})^* \cup a^{11}(a^{15})^*$
38	240 = lcm(10,12,16)	240	$a^9(a^{10}) \cup a^8(a^{12}) \cup a^{10}(a^{16})^*$	182	$a^{10}(a^9)^* \cup a^{11}(a^{12})^* \cup a^{12}(a^{15})^*$
39	240 = lcm(10,12,16)	241	$a^{10}(a^{10}) \cup a^9(a^{12}) \cup a^{11}(a^{16})^*$	241	$a^{10}(a^{10})^* \cup a^9(a^{12})^* \cup a^{11}(a^{16})^*$
40	240 = lcm(10,12,16)	242	$a^{11}(a^{10}) \cup a^{10}(a^{12}) \cup a^{12}(a^{16})^*$	242	$a^{11}(a^{10})^* \cup a^{10}(a^{12})^* \cup a^{12}(a^{16})^*$
41	240 = lcm(10,12,16)	243	$a^{12}(a^{10}) \cup a^{11}(a^{12}) \cup a^{13}(a^{16})^*$	243	$a^{12}(a^{10})^* \cup a^{11}(a^{12})^* \cup a^{13}(a^{16})^*$
42	336 = lcm(12,14,16)	336	$a^{11}(a^{12}) \cup a^{12}(a^{14}) \cup a^{13}(a^{16})^*$	244	$a^{13}(a^{10})^* \cup a^{12}(a^{12})^* \cup a^{14}(a^{16})^*$
43	336 = lcm(12,14,16)	337	$a^{12}(a^{12}) \cup a^{13}(a^{14}) \cup a^{14}(a^{16})^*$	337	$a^{12}(a^{12})^* \cup a^{13}(a^{14})^* \cup a^{14}(a^{16})^*$
44	336 = lcm(12,14,16)	338	$a^{13}(a^{12}) \cup a^{14}(a^{14}) \cup a^{15}(a^{16})^*$	338	$a^{13}(a^{12})^* \cup a^{14}(a^{14})^* \cup a^{15}(a^{16})^*$
45	336 = lcm(12,14,16)	339	$a^{14}(a^{12}) \cup a^{15}(a^{14}) \cup a^{16}(a^{16})^*$	339	$a^{14}(a^{12})^* \cup a^{15}(a^{14})^* \cup a^{16}(a^{16})^*$
46	420 = lcm(12,14,20)	420	$a^{11}(a^{12}) \cup a^{12}(a^{14}) \cup a^{13}(a^{20})^*$	340	$a^{15}(a^{12})^* \cup a^{16}(a^{14})^* \cup a^{17}(a^{16})^*$
47	420 = lcm(12,14,20)	421	$a^{12}(a^{12}) \cup a^{13}(a^{14}) \cup a^{14}(a^{20})^*$	421	$a^{12}(a^{12})^* \cup a^{13}(a^{14})^* \cup a^{14}(a^{20})^*$
48	420 = lcm(12,14,20)	422	$a^{13}(a^{12}) \cup a^{14}(a^{14}) \cup a^{15}(a^{20})^*$	422	$a^{13}(a^{12})^* \cup a^{14}(a^{14})^* \cup a^{15}(a^{20})^*$
49	420 = lcm(12,14,20)	423	$a^{14}(a^{12}) \cup a^{15}(a^{14}) \cup a^{16}(a^{20})^*$	423	$a^{14}(a^{12})^* \cup a^{15}(a^{14})^* \cup a^{16}(a^{20})^*$
50	560 = lcm(14,16,20)	560	$a^{13}(a^{14}) \cup a^{12}(a^{16}) \cup a^{14}(a^{20})^*$	424	$a^{15}(a^{12})^* \cup a^{16}(a^{14})^* \cup a^{17}(a^{20})^*$

Table 1: UFA–DFA tradeoff with witness languages.

n	$g(n)$	$\tilde{g}(n)$
4	4	4
5	6 = lcm(2,3)	5
6	6 = lcm(2,3)	6
7	12 = lcm(3,4)	7
8	15 = lcm(3,5)	8
9	20 = lcm(4,5)	9
10	30 = lcm(2,3,5)	12 = lcm(4,6)
11	30 = lcm(2,3,5)	12 = lcm(4,6)
12	60 = lcm(3,4,5)	12 = lcm(4,6)
13	60 = lcm(3,4,5)	13
14	84 = lcm(3,4,7)	24 = lcm(6,8)
15	105 = lcm(3,5,7)	24 = lcm(6,8)
16	140 = lcm(4,5,7)	30 = lcm(6,10)
17	210 = lcm(2,3,5,7)	30 = lcm(6,10)
18	210 = lcm(2,3,5,7)	40 = lcm(8,10)
19	420 = lcm(3,4,5,7)	40 = lcm(8,10)
20	420 = lcm(3,4,5,7)	42 = lcm(6,14)
21	420 = lcm(3,4,5,7)	42 = lcm(6,14)
22	420 = lcm(3,4,5,7)	60 = lcm(10,12)
23	840 = lcm(3,5,7,8)	60 = lcm(10,12)
24	840 = lcm(3,5,7,8)	70 = lcm(10,14)
25	1260 = lcm(4,5,7,9)	70 = lcm(10,14)
26	1260 = lcm(4,5,7,9)	84 = lcm(12,14)
27	1540 = lcm(4,5,7,11)	84 = lcm(12,14)
28	2310 = lcm(2,3,5,7,11)	90 = lcm(10,18)
29	2520 = lcm(5,7,8,9)	90 = lcm(10,18)
30	4620 = lcm(3,4,5,7,11)	120 = lcm(8,10,12)
31	4620 = lcm(3,4,5,7,11)	120 = lcm(8,10,12)
32	5460 = lcm(3,4,5,7,13)	126 = lcm(14,18)
33	5460 = lcm(3,4,5,7,13)	126 = lcm(14,18)
34	9240 = lcm(3,5,7,8,11)	168 = lcm(8,12,14)
35	9240 = lcm(3,5,7,8,11)	168 = lcm(8,12,14)
36	13860 = lcm(4,5,7,9,11)	180 = lcm(9,12,15)
37	13860 = lcm(4,5,7,9,11)	180 = lcm(9,12,15)
38	16380 = lcm(4,5,7,9,13)	240 = lcm(10,12,16)
39	16380 = lcm(4,5,7,9,13)	240 = lcm(10,12,16)
40	27720 = lcm(5,7,8,9,11)	240 = lcm(10,12,16)
⋮	⋮	⋮
77	9699690 = lcm(2,3,5,7,11,13,17,19)	1848 = lcm(22,24,28)
78	12252240 = lcm(5,7,9,11,13,16,17)	2520 = lcm(15,18,21,24)
79	19399380 = lcm(3,4,5,7,11,13,17,19)	2520 = lcm(15,18,21,24)
80	19399380 = lcm(3,4,5,7,11,13,17,19)	2520 = lcm(15,18,21,24)
⋮	⋮	⋮
117	2677114440 = lcm(5,7,8,9,11,13,17,19,23)	11880 = lcm(24,27,30,33)
118	3375492120 = lcm(5,7,8,9,11,13,17,19,29)	11880 = lcm(24,27,30,33)
119	3375492120 = lcm(5,7,8,9,11,13,17,19,29)	11880 = lcm(24,27,30,33)
120	5354228880 = lcm(5,7,9,11,13,16,17,19,23)	14040 = lcm(24,27,30,39)
⋮	⋮	⋮
157	209280511440 = lcm(5,7,9,11,13,16,17,19,29,31)	55440 = lcm(16,18,30,42,44)
158	209280511440 = lcm(5,7,9,11,13,16,17,19,29,31)	65520 = lcm(16,18,30,42,52)
159	232908956280 = lcm(5,7,8,11,13,17,19,23,27,29)	65520 = lcm(16,18,30,42,52)
160	232908956280 = lcm(5,7,8,11,13,17,19,23,27,29)	65520 = lcm(16,18,30,42,52)

Table 2: Landau's function $g(n)$ vs. its variant $\tilde{g}(n)$

are compared in Table 2, which also includes the expansions of $g(n)$ and $\tilde{g}(n)$ as least common multiples of some cycle lengths p_1, \dots, p_k .

One can see that the extra condition in the definition of $\tilde{g}(n)$, that of the existence of offsets f_1, \dots, f_k , of which every two f_i, f_j are distinct modulo $\gcd(p_i, p_j)$, leads to a significant reduction of the least common multiple. Furthermore, note that the cycle lengths p_1, \dots, p_k have to become larger in order to accommodate their common divisors, and therefore fewer of them can be fit with the same upper bound on their sum: for example, for $n = 160$, the value $g(n)$ is reached using ten different cycles, while the definition $\tilde{g}(n)$ allows only five. The latter observation turns out to be crucial in the analysis of the function \tilde{g} , and the first step towards determining its growth rate is estimating the maximum number of cycles k for a given sum of cycle lengths.

Lemma 3. *Let $k \geq 1$ and let $\pi_1, \dots, \pi_k \geq 2$ be any integers, for which*

- (a). *there exist $f_1, \dots, f_k \in \mathbb{N}$ with $f_i \not\equiv f_j \pmod{\gcd(\pi_i, \pi_j)}$ for all $i \neq j$, and*
- (b). *$\text{lcm}(\pi_1, \dots, \pi_{i-1}, \pi_{i+1}, \dots, \pi_k)$ is not divisible by π_i , for each $1 \leq i \leq k$.*

Then $\pi_1 + \dots + \pi_k > \frac{4}{9}k^3 \ln k - \frac{8}{27}k^3 \sqrt{\ln k}$.

As in Lemma 1, the condition (b) of each cycle contributing something to the least common multiple is essential: if it is lifted, then taking k cycles each of length k gives $\sum \pi_i = k^2$, and the statement does not hold.

For each i , let $r_i = \frac{\text{lcm}(\pi_1, \dots, \pi_k)}{\text{lcm}(\pi_1, \dots, \pi_{i-1}, \pi_{i+1}, \dots, \pi_k)}$ be the contribution of the i -th cycle to the least common multiple, and let $s_i = \frac{\pi_i}{r_i}$. Then the numbers r_1, \dots, r_k are pairwise relatively prime, each of them is at least 2 by the condition (b), and hence $\gcd(\pi_i, \pi_j) = \gcd(s_i, s_j)$ for $i \neq j$. In this notation, the statement of the lemma can be equivalently reformulated as follows:

$$\min_{\substack{r_1, \dots, r_k \geq 2 \\ \text{relatively prime}}} \min_{\substack{s_1, \dots, s_k \in \mathbb{N}: \\ \exists f_1, \dots, f_k \in \mathbb{N} \\ f_i \not\equiv f_j \pmod{\gcd(s_i, s_j)}}} \sum_{i=1}^k r_i s_i > \frac{4}{9}k^3 \ln k - \frac{8}{27}k^3 \sqrt{\ln k}.$$

The proof proceeds by simplifying the expression in the left-hand side, decreasing its value, but in the end still obtaining a value greater than $\frac{4}{9}k^3 \ln k - \frac{8}{27}k^3 \sqrt{\ln k}$. The first simplification step is replacing the combinatorial condition on s_1, \dots, s_k involving the numbers f_1, \dots, f_k with the following numerical consequence of this condition:

Claim A. *Let $k \geq 1$ and $s_1, \dots, s_k \geq 2$ be any such numbers, that there exist offsets f_1, \dots, f_k with $f_i \in \{0, \dots, s_i - 1\}$, satisfying $f_i \not\equiv f_j \pmod{\gcd(s_i, s_j)}$ for all $i \neq j$. Then $\frac{1}{s_1} + \dots + \frac{1}{s_k} \leq 1$.*

Proof. Let $s = \text{lcm}(s_1, \dots, s_k)$. An i -th cycle is said to *cover* a number $n \in \{0, \dots, s-1\}$, if $f_i \equiv n \pmod{s_i}$. Then each i -th cycle covers exactly $\frac{s}{s_i}$ different numbers, and, in total, $\sum_{i=1}^k \frac{s}{s_i}$ numbers are covered.

Suppose $\sum_{i=1}^k \frac{1}{s_i} > 1$. Then $\sum_{i=1}^k \frac{s}{s_i} > s$, that is, more than s numbers in $\{0, \dots, s-1\}$ are covered. Accordingly, some number n must be covered by two different cycles, that is, $f_i \equiv n \pmod{s_i}$ and $f_j \equiv n \pmod{s_j}$. Therefore, $f_i \equiv n \equiv f_j \pmod{\gcd(s_i, s_j)}$, which contradicts the assumption. \square

In order to obtain the smallest values of the sum $\sum r_i s_i$, the numbers s_i should be as small as possible, but too small values are not allowed by Claim A. For example, for $k = 3$ and $r_1 = 2, r_2 = 3, r_3 = 7$, the smallest possible values of s_i are $s_1 = s_2 = s_3 = 3$ or $s_1 = s_2 = 4, s_3 = 2$. The former choice leads to the sum $2 \cdot 3 + 3 \cdot 3 + 7 \cdot 3 = 36$, while the latter gives $2 \cdot 4 + 3 \cdot 4 + 7 \cdot 2 = 34$. Note that taking any smaller values of s_i would violate the condition of Claim A, while any greater values would increase the sum; therefore, the least value of $\sum r_i s_i$ for the given k and r_i is 34.

Aiming to estimate this minimum, it is convenient to allow the values of s_i to be any positive real numbers. This will slightly reduce the value of the minimum, but will make it analytically calculable as follows:

Claim B. Let $a_1, \dots, a_m > 0$ be any positive real numbers. Then

$$\min_{\substack{x_1, \dots, x_k \in \mathbb{R}_+ \\ \frac{1}{x_1} + \dots + \frac{1}{x_k} = 1}} \sum_{i=1}^k a_i x_i = (\sqrt{a_1} + \dots + \sqrt{a_k})^2$$

and the minimum is reached at the point $x_i = \frac{\sqrt{a_1 + \dots + \sqrt{a_k}}}{\sqrt{a_i}}$.

Proof. This is an exercise in analysis. Eliminating one of the variables as

$$x_k = \frac{1}{1 - \frac{1}{x_1} - \dots - \frac{1}{x_{k-1}}},$$

the task is to find the minimum of the following function:

$$f(x_1, \dots, x_{k-1}) = a_1 x_1 + \dots + a_{k-1} x_{k-1} + \frac{a_k}{1 - \frac{1}{x_1} - \dots - \frac{1}{x_{k-1}}}.$$

Its partial derivative by x_i is

$$\frac{\partial f}{\partial x_i} = a_i - \frac{a_k}{x_i^2 \left(1 - \frac{1}{x_1} - \dots - \frac{1}{x_{k-1}}\right)^2}.$$

Taking the necessary condition of an extremum, $\frac{\partial f}{\partial x_i} = 0$ for all i , and assuming new variables $y_i = \frac{1}{x_i}$ leads to the following system of equations:

$$\frac{y_i^2}{(1 - y_1 - \dots - y_{k-1})^2} = \frac{a_i}{a_k} \quad (\text{for } 1 \leq i \leq k-1).$$

Since both y_i and $1 - y_1 - \dots - y_{k-1}$ are positive, this system can be reformulated as

$$\frac{y_i}{1 - y_1 - \dots - y_{k-1}} = \sqrt{\frac{a_i}{a_k}} \quad (\text{for } 1 \leq i \leq k-1).$$

Now each variable y_i with $2 \leq i \leq k-1$ can be expressed through y_1 by dividing the i -th equation by the first one:

$$\frac{y_i}{y_1} = \sqrt{\frac{a_i}{a_1}} \quad (\text{for } 2 \leq i \leq k-1).$$

Substituting $y_i = y_1 \sqrt{\frac{a_i}{a_1}}$ in the first equation results in

$$\frac{y_1}{1 - \sum_{j=1}^{k-1} y_1 \sqrt{\frac{a_j}{a_1}}} = \sqrt{\frac{a_1}{a_k}},$$

and therefore

$$y_1 = \frac{1}{\sum_{j=1}^k \sqrt{\frac{a_j}{a_1}}}.$$

Returning to the original variables, f attains its minimum at $x_i = \sum_{j=1}^k \sqrt{\frac{a_j}{a_i}}$, and its value at this point is $\sum_{i=1}^k \sum_{j=1}^k \sqrt{a_i a_j} = (\sqrt{a_1} + \dots + \sqrt{a_k})^2$, which proves the claim. \square

Therefore, there is a lower bound $(\sqrt{r_1} + \dots + \sqrt{r_k})^2$ on the sum $\sum_{i=1}^k r_i s_i$, and the next task is to estimate the least value of this sum for all applicable r_i , that is, for every choice of pairwise relatively prime $r_1, \dots, r_k \geq 2$. In fact, the minimum is achieved by taking the first k primes.

Claim C. Let $2 \leq r_1 < \dots < r_k$ be any pairwise relatively prime natural numbers. Then $p_i \leq r_i$, where p_i is the i -th prime.

Proof. Suppose that $r_i < p_i$ for some i . Each r_j with $j < i$ is less than r_i , and hence r_j must have a prime factor $r'_j \leq p_{i-1}$. Since the primes r'_1, \dots, r'_{i-1} must be pairwise distinct, it follows that $\{r'_1, \dots, r'_{i-1}\} = \{p_1, \dots, p_{i-1}\}$, and thus every prime factor of r_i must belong to this set, which contradicts the assumption that r_1, \dots, r_k are relatively prime. \square

Therefore, the sum is decreased (or unaltered) by replacing each r_i with the i -th prime:

$$(\sqrt{r_1} + \dots + \sqrt{r_k})^2 \geq (\sqrt{p_1} + \dots + \sqrt{p_k})^2.$$

In order to estimate the sum $\sum_{i=1}^k \sqrt{p_i}$, consider the following known fact:

Proposition 4 (Rosser [30]). $p_n > n \ln n$ for all $n \geq 1$.

It remains to calculate the resulting sum:

Claim D. $\sum_{n=1}^k \sqrt{n \ln n} > \frac{2}{3}k\sqrt{k \ln k} - \frac{2}{9}k\sqrt{k}$ for all $k \geq 1$.

Proof. For k up to 3, the inequality can be verified by direct calculations, so assume $k \geq 4$. The idea is to approximate the sum $\sum_{n=4}^k \sqrt{n \ln n}$ with the integral $\int_3^k \sqrt{x \ln x} dx$. Integrating by parts,

$$\begin{aligned} \int \sqrt{x \ln x} dx &= x\sqrt{x \ln x} - \int x d\sqrt{x \ln x} = x\sqrt{x \ln x} - \int x \frac{\ln x + 1}{2\sqrt{x \ln x}} dx = \\ &= x\sqrt{x \ln x} - \frac{1}{2} \int \sqrt{x \ln x} dx - \frac{1}{2} \int \sqrt{\frac{x}{\ln x}} dx, \end{aligned}$$

and solving the resulting equation gives

$$\int \sqrt{x \ln x} dx = \frac{2}{3}x\sqrt{x \ln x} - \frac{1}{3} \int \sqrt{\frac{x}{\ln x}} dx.$$

Then, using the facts that $f(x) = \sqrt{x \ln x}$ is increasing on $[e, +\infty)$, and that $\sqrt{\frac{x}{\ln x}} \leq \sqrt{x}$ for all $x \geq e$,

$$\begin{aligned} \sum_{n=1}^k \sqrt{n \ln n} &= \sqrt{2 \ln 2} + \sqrt{3 \ln 3} + \sum_{n=4}^k \sqrt{n \ln n} > \sqrt{2 \ln 2} + \sqrt{3 \ln 3} + \int_3^k \sqrt{x \ln x} dx = \\ &= \sqrt{2 \ln 2} + \sqrt{3 \ln 3} + \frac{2}{3}k\sqrt{k \ln k} - \frac{2}{3}\sqrt{3 \ln 3} - \frac{1}{3} \int_3^k \sqrt{\frac{x}{\ln x}} dx > \sqrt{2 \ln 2} - \sqrt{3 \ln 3} + \frac{2}{3}k\sqrt{k \ln k} - \frac{1}{3} \int_3^k \sqrt{x} dx = \\ &= \frac{2}{3}k\sqrt{k \ln k} - \frac{2}{9}k\sqrt{k} + \frac{2}{9}3\sqrt{3} + \sqrt{2 \ln 2} - \sqrt{3 \ln 3}. \end{aligned}$$

To show that the latter value is greater than $\frac{2}{3}k\sqrt{k \ln k} - \frac{2}{9}k\sqrt{k}$, as desired, it is left to demonstrate that the constant terms sum up to a positive value. First, note that $\frac{2}{3}\sqrt{3} > 1$ is equivalent to $2 > \sqrt{3}$, which is true. The second term is estimated as $\sqrt{2 \ln 2} > 1$, which is equivalent to another true statement $2^2 > e$. For the third term, $\sqrt{3 \ln 3} < 2$ holds if and only if $3^3 < e^4 \approx 55$, which is true as well. Altogether,

$$\frac{2}{3}\sqrt{3} + \sqrt{2 \ln 2} - \sqrt{3 \ln 3} > 1 + 1 - 2 = 0,$$

which completes the proof. \square

With all these auxiliary results established, Lemma 3 is proved by the following chain of inequalities.

Proof of Lemma 3.

$$\begin{aligned}
& \min_{\substack{r_1, \dots, r_k \geq 2 \\ \text{relatively prime}}} \min_{\substack{s_1, \dots, s_k \in \mathbb{N} \\ \exists f_1, \dots, f_k \in \mathbb{N} \\ f_i \not\equiv f_j \pmod{\gcd(s_i, s_j)}}} \sum_{i=1}^k r_i s_i \stackrel{\text{Cl.A}}{\geq} \min_{\substack{r_1, \dots, r_k \geq 2 \\ \text{relatively prime}}} \min_{\substack{s_1, \dots, s_k \in \mathbb{N} \\ \frac{1}{s_1} + \dots + \frac{1}{s_k} \leq 1}} \sum_{i=1}^k r_i s_i \geq \\
& \geq \min_{\substack{r_1, \dots, r_k \geq 2 \\ \text{relatively prime}}} \min_{\substack{x_1, \dots, x_k \in \mathbb{R}_+ \\ \frac{1}{x_1} + \dots + \frac{1}{x_k} \leq 1}} \sum_{i=1}^k r_i x_i \stackrel{\text{Cl.B}}{=} \min_{\substack{r_1, \dots, r_k \geq 2 \\ \text{relatively prime}}} (\sqrt{r_1} + \dots + \sqrt{r_k})^2 \stackrel{\text{Cl.C}}{=} \\
& = (\sqrt{p_1} + \dots + \sqrt{p_k})^2 \stackrel{\text{P4}}{>} \left(\sum_{i=1}^k \sqrt{i \ln i} \right)^2 \stackrel{\text{Cl.D}}{>} \left(\frac{2}{3} k \sqrt{k \ln k} - \frac{2}{9} k \sqrt{k} \right)^2 > \\
& > \frac{4}{9} k^3 \ln k - \frac{8}{27} k^3 \sqrt{\ln k}.
\end{aligned}$$

□

The next lemma reformulates this estimation by giving a lower bound on k as a function of n .

Lemma 4. *Under the assumptions of Lemma 3, $k < \frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}$, where $n = \pi_1 + \dots + \pi_k \geq 55$.*

The condition that $n \geq 55 > e^4$ is needed to ensure that the denominator of the fraction under the cubic root is positive.

Proof. Suppose $k \geq \frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}$. Then $k^3 \geq \frac{27}{4} \frac{n}{\ln n - 2\sqrt{\ln n}}$ and $\ln k \geq \frac{1}{3} (\ln n - \ln(\ln n - 2\sqrt{\ln n}) + \ln \frac{27}{4})$, and since the function $f(k) = \frac{4}{9} k^3 \ln k - \frac{8}{27} k^3 \sqrt{\ln k} = k^3 \sqrt{\ln k} (\frac{4}{9} \sqrt{\ln k} - \frac{8}{27})$ is increasing,

$$\begin{aligned}
& \frac{4}{9} k^3 \ln k - \frac{8}{27} k^3 \sqrt{\ln k} \geq \\
& \geq n \frac{\frac{27}{4} \cdot \frac{4}{9} \cdot \frac{1}{3} (\ln n - \ln(\ln n - 2\sqrt{\ln n}) + \ln \frac{27}{4}) - \frac{27}{4} \cdot \frac{8}{27} \cdot \frac{1}{\sqrt[3]{3}} \sqrt{\ln n - \ln(\ln n - 2\sqrt{\ln n}) + \ln \frac{27}{4}}}{\ln n - 2\sqrt{\ln n}} = \\
& = n \frac{\ln n - \ln(\ln n - 2\sqrt{\ln n}) + \ln \frac{27}{4} - \frac{2}{\sqrt[3]{3}} \sqrt{\ln n - \ln(\ln n - 2\sqrt{\ln n}) + \ln \frac{27}{4}}}{\ln n - 2\sqrt{\ln n}} > \\
& > n \frac{\ln n - \ln \ln n + 1 - \frac{2}{\sqrt[3]{3}} \sqrt{\ln n + 2}}{\ln n - 2\sqrt{\ln n}} > n,
\end{aligned}$$

where the last inequality is established by showing that $2\sqrt{\ln n} > \ln \ln n - 1 + \frac{2}{\sqrt[3]{3}} \sqrt{\ln n + 2}$ for all applicable values of n . Substituting $x = \sqrt{\ln n}$, consider the function $h(x) = 2x - 2 \ln x + 1 - \frac{2}{\sqrt[3]{3}} \sqrt{x^2 + 2}$. It is easy to calculate that $h(2) > 0$ and to verify that $h'(x) = 2 - \frac{2}{x} - \frac{2}{\sqrt[3]{3}} \frac{x}{\sqrt{x^2 + 2}} > 0$ for all $x \geq 2$. Hence, the function is positive for all $x \geq 2$, and accordingly the inequality holds for all $n \geq e^4$.

It has thus been shown that $\frac{4}{9} k^3 \ln k - \frac{8}{27} k^3 \sqrt{\ln k} > n$, contrary to Lemma 3. The contradiction obtained proves the lemma. □

The following upper bound of $\tilde{g}(n)$ can be inferred from this bound on k .

Theorem 2 (Upper bound). $\tilde{g}(n) < e^{\sqrt[3]{2n \ln^2 n (1+o(1))}}$.

The proof of the theorem, which is presented below, relies *only* on the upper bound on k , and otherwise ignores the additional constraints in the definition of \tilde{g} as compared to g . Using further properties of \tilde{g} in this proof might have led to a better bound.

The first step is to simplify the model by replacing the least common multiple of the cycle lengths, as in the definition of \tilde{g} , with the product of these cycle lengths, and then allowing them to be real numbers. Then, as it is well-known, the maximum of the product is reached for all factors being identical:

Proposition 5. $\max_{x_1 + \dots + x_k \leq x} x_1 \dots x_k = \left(\frac{x}{k}\right)^k$ for every $k \in \mathbb{N}$ and $x \in \mathbb{R}_+$.

Another fact about elementary functions is that $\left(\frac{n}{k}\right)^k$ reaches its maximum at $k = \frac{n}{e}$, and since the values of k allowed by Lemma 4 are much smaller, one should choose k as large as possible to obtain the greatest value of $\left(\frac{n}{k}\right)^k$.

Proposition 6. The function $f(y) = \left(\frac{n}{y}\right)^y$ increases on $0 < y \leq \frac{n}{e}$, has a maximum at $y = \frac{n}{e}$ and decreases on $\frac{n}{e} \leq y$.

Proof of Theorem 2. The upper bound is proved by the following chain of inequalities, which uses Lemma 4, Proposition 5 and Proposition 6. In the first three lines, the condition of the existence of appropriate offsets f_1, \dots, f_k from the definition of \tilde{g} is abbreviated to an ellipsis.

$$\begin{aligned}
\tilde{g}(n) &= \max_{k \geq 1} \{ \text{lcm}(\pi_1, \dots, \pi_k) \mid \pi_1 + \dots + \pi_k \leq n \text{ and } \langle \dots \rangle \} = \\
&= \max_{k \geq 1} \{ \text{lcm}(\pi_1, \dots, \pi_k) \mid \pi_1 + \dots + \pi_k \leq n, \frac{\text{lcm}(\pi_1, \dots, \pi_k)}{\text{lcm}(\pi_1, \dots, \pi_{i-1}, \pi_{i+1}, \dots, \pi_k)} \geq 2, \text{ and } \langle \dots \rangle \} = \\
&= \max_{1 \leq k < \frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}} \{ \text{lcm}(\pi_1, \dots, \pi_k) \mid \pi_1 + \dots + \pi_k \leq n \text{ and } \langle \dots \rangle \} \leq \\
&\leq \max_{1 \leq k < \frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}} \{ \pi_1 \dots \pi_k \mid \pi_1 + \dots + \pi_k \leq n \} \leq \\
&\leq \max_{1 \leq k < \frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}} \max_{\substack{x_1, \dots, x_k \in \mathbb{R}_+ \\ x_1 + \dots + x_k \leq n}} \prod_{i=1}^k x_i = \max_{1 \leq k < \frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}} \left(\frac{n}{k}\right)^k \leq \\
&\leq \left(\frac{n}{\frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}}\right)^{\frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}} = e^{\frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}} \ln \left(\frac{\sqrt[3]{4}}{3} n^{\frac{2}{3}} \sqrt[3]{\ln n - 2\sqrt{\ln n}}\right)} < \\
&< e^{\frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}} \ln \left(n^{\frac{2}{3}} \sqrt[3]{\ln n}\right)} = e^{\frac{3}{\sqrt[3]{4}} \frac{\sqrt[3]{n}}{\sqrt[3]{\ln n}} \sqrt[3]{\frac{\ln n}{\ln n - 2\sqrt{\ln n}}} \left(\frac{2}{3} \ln n + \frac{1}{3} \ln \ln n\right)} = \\
&= e^{\frac{3}{\sqrt[3]{4}} \frac{\sqrt[3]{n}}{\sqrt[3]{\ln n}} \sqrt[3]{1 + \frac{2\sqrt{\ln n}}{\ln n - 2\sqrt{\ln n}}} \frac{2}{3} \ln n \left(1 + \frac{\ln \ln n}{2\sqrt{\ln n}}\right)} = e^{\sqrt[3]{2} \sqrt[3]{n} (\ln n)^{\frac{2}{3}} (1+o(1))}.
\end{aligned}$$

□

The second task is to establish a lower bound on \tilde{g} . The argument is based upon the following known facts about primes. Let p_i denote the i -th prime.

Proposition 7 (folklore; see Bach and Shallit [2], OEIS [24, A007504]). $\sum_{i=1}^k p_i = (1 + o(1)) \frac{1}{2} k^2 \ln k$.

Proposition 8 (Rosser [31]; OEIS [24, A002110]). $\prod_{i=1}^k p_i = e^{k \ln k + k \ln \ln k - k + o(k)} = e^{(1+o(1))k \ln k}$.

Using these facts, the following lower bound on $\tilde{g}(n)$ shall be established:

Theorem 3 (Lower bound). $\tilde{g}(n) > e^{\sqrt[3]{\frac{2}{9}} \sqrt[3]{n \ln^2 n} (1+o(1))}$.

Proof. For any k , consider the numbers kp_i with $i \in \{1, \dots, k\}$. These numbers satisfy the definition of \tilde{g} with $f_i = i - 1$ for each i . Let $s_k = k \sum_{i=1}^k p_i$ be the sum of these numbers. Then the value of \tilde{g} on s_k must be at least $\text{lcm}(kp_1, \dots, kp_k) = k \prod_{i=1}^k p_i$.

By Proposition 7, the argument of \tilde{g} is estimated as

$$s_k = k \sum_{i=1}^k p_i = (1 + o(1)) \frac{1}{2} k^3 \ln k.$$

Note that

$$\begin{aligned} s_{k+1} &= (1 + o(1)) \frac{1}{2} (k+1)^3 \ln(k+1) = (1 + o(1)) \frac{1}{2} k^3 (\ln k) \frac{(k+1)^3 \ln(k+1)}{k^3 \ln k} = \\ &= (1 + o(1)) \frac{1}{2} k^3 (\ln k) \left(1 + \frac{O(k^2)}{k^3}\right) \left(1 + \frac{\ln \frac{k+1}{k}}{\ln k}\right) = (1 + o(1)) \frac{1}{2} k^3 \ln k. \end{aligned}$$

Let $f : \mathbb{N} \rightarrow \mathbb{R}$ be the infinitesimal function, for which $s_{k+1} = (1 + f(k)) \frac{1}{2} k^3 \ln k$. Fix any n and consider the greatest number k with $s_k \leq n$. Then

$$n < s_{k+1} = (1 + f(k)) \frac{1}{2} k^3 \ln k. \quad (1)$$

Using Proposition 8 to estimate the product of the first k primes, the above lower bound on $\tilde{g}(n)$ is

$$k \prod_{i=1}^k p_i = e^{(1+o(1))k \ln k}.$$

From the inequality (1), one can infer a lower bound on $k \ln k$ of the form $(1 + o(1)) \sqrt[3]{\frac{2}{9}} \sqrt[3]{n} \ln^{\frac{2}{3}} n$. This is proved as follows:

$$\begin{aligned} \sqrt[3]{\frac{2}{9}} \sqrt[3]{n} \ln^{\frac{2}{3}} n &< \sqrt[3]{\frac{2}{9}} \sqrt[3]{1 + f(k)} \frac{1}{\sqrt[3]{2}} k \sqrt[3]{\ln k} \ln^{\frac{2}{3}} ((1 + f(k)) \frac{1}{2} k^3 \ln k) = \\ &= \sqrt[3]{\frac{1}{9}} \sqrt[3]{1 + f(k)} k \sqrt[3]{\ln k} \left(3 \ln k + \ln(1 + f(k)) + \ln \frac{\ln k}{2}\right)^{\frac{2}{3}} = \\ &= \sqrt[3]{\frac{1}{9}} \sqrt[3]{1 + f(k)} k \sqrt[3]{\ln k} \sqrt[3]{9} (\ln^{\frac{2}{3}} k) \left(1 + \frac{\ln(1 + f(k))}{3 \ln k} + \frac{\ln \frac{\ln k}{2}}{3 \ln k}\right)^{\frac{2}{3}} = k \ln k (1 + o(1)). \end{aligned}$$

Altogether, the above calculations lead to the following lower bound on $\tilde{g}(n)$:

$$\tilde{g}(n) \geq \tilde{g}(s_k) \geq \prod_{i=1}^k p_i = e^{(1+o(1))k \ln k} > e^{(1+o(1)) \sqrt[3]{\frac{2}{9}} \sqrt[3]{n} \ln^{\frac{2}{3}} n}.$$

□

According to Theorems 2–3, the values of the function \tilde{g} are confined within the following bounds:

$$e^{\sqrt[3]{\frac{2}{9}} \sqrt[3]{n \ln^2 n} (1+o(1))} < \tilde{g}(n) < e^{\sqrt[3]{2} \sqrt[3]{n \ln^2 n} (1+o(1))}.$$

Corollary 1. $\tilde{g}(n) = e^{\Theta(\sqrt[3]{n \ln^2 n})}$.

Improving this estimation is an interesting theoretical question. Perhaps it could be proved that \tilde{g} is of the order $e^{C \sqrt[3]{n \ln^2 n} (1+o(1))}$, for some constant C with $0.605 < \sqrt[3]{\frac{2}{9}} \leq C \leq \sqrt[3]{2} < 1.260$. In anticipation of such a result, it is worthwhile to elaborate on the constants obtained in the above proof.

The first function estimated in the proof is the least number $n = n(k)$, for which k cycles may be used in the definition of $\tilde{g}(n)$. Lemma 3 gives a lower bound of $\frac{1}{9}(1 + o(1))k^3 \ln k$. At the same time, the proof of Theorem 3 contains an example with the sum $\frac{1}{2}(1 + o(1))k^3 \ln k$. Possibly, the actual function here could

be represented as $C'(1 + o(1))k^3 \ln k$ for $\frac{4}{9} \leq C' \leq \frac{1}{2}$. The gap between $C' = \frac{4}{9}$ and $C' = \frac{1}{2}$ reflects several essential simplifications made in the course of the proof, and narrowing this gap might require an entirely different argument.

Suppose the least number $n = n(k)$ allowing k cycles were estimated as $C'(1 + o(1))k^3 \ln k$. Then an accordingly revised Lemma 4 would give $k < (1 + o(1))\sqrt[3]{\frac{3}{C'}}\sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}$, which would in turn modify the upper bound on $\tilde{g}(n)$ given in Theorem 2 to $e^{\sqrt[3]{\frac{8}{9C'}}\sqrt[3]{n \ln^2 n(1+o(1))}}$. Provided that examples with $n = C'(1 + o(1))k^3 \ln k$ are also constructed in Theorem 3, the lower bound on \tilde{g} would become $e^{\sqrt[3]{\frac{1}{9C'}}\sqrt[3]{n \ln^2 n(1+o(1))}}$. The exponents in these bounds differ by a factor of 2, which is another measure of inefficiency of the arguments in this section.

Returning to the UFA–DFA tradeoff, note that the tradeoff function satisfies $\tilde{g}(n) \leq f_{\text{UFA-DFA}} \leq \tilde{g}(n) + n$, while in the case of UFAs with a unique initial state, $\tilde{g}(n-1) \leq f_{\text{UFA}_1\text{-DFA}} \leq \tilde{g}(n-1) + n$. Therefore, both functions asymptotically behave as \tilde{g} :

Corollary 2. *The number of states in a DFA sufficient and, in the worst case, necessary to represent languages recognized by n -state UFAs, with a multiple initial states or with a unique initial state, is*

$$\begin{aligned} f_{\text{UFA-DFA}}(n) &= \tilde{g}(n) + O(n) = e^{\Theta(\sqrt[3]{n \ln^2 n})}, \\ f_{\text{UFA}_1\text{-DFA}}(n) &= \tilde{g}(n) + O(n) = e^{\Theta(\sqrt[3]{n \ln^2 n})}. \end{aligned}$$

5. NFA–UFA tradeoff

By the results of Lyubich [19] and Chrobak [5] mentioned above as Proposition 3, an n -state NFA can be transformed to an equivalent DFA with $g(n) + n^2$ states. The transformation begins by converting an n -state NFA to the Chrobak normal form with a tail of length $O(n^2)$ and with at most n states in the cycles, and then proceeds by determinizing the cycles, making at most $g(n)$ states. A close lower bound is given by a family of n -state NFAs, for which the equivalent DFA requires $g(n)$ states. Though the exact values of the NFA–DFA tradeoff function are not known, these two bounds are asymptotically tight.

Consider the transformation of an n -state NFA to an equivalent UFA. It can obviously be achieved simply by transforming the given NFA to a DFA. It turns out that for some NFAs no better transformation is possible:

Lemma 5. *For all $k \geq 1$ and $p_1, \dots, p_k \geq 2$, the language*

$$L = \{\varepsilon\} \cup a \bigcup_{i=1}^k \{\varepsilon, a, a^2, \dots, a^{p_i-2}\} (a^{p_i})^* = \{a^n \mid n \not\equiv 0 \pmod{\text{lcm}(p_1, \dots, p_k)}\} \cup \{\varepsilon\}$$

has an NFA with $1 + \sum_{i=1}^k p_i$ states, while the smallest UFA for L needs at least $1 + \text{lcm}(p_1, \dots, p_k)$ states.

Proof. The NFA for L is in Chrobak normal form, with the tail of length 1 and with k cycles of length p_1, \dots, p_k .

The smallest DFA for L contains an accepting initial state and a cycle of length $\text{lcm}(p_1, \dots, p_k)$, which has a non-accepting last state, with the rest of the states being accepting. It remains to show that there does not exist any smaller UFA recognizing this language. This can be done using the method of Schmidt [32].

Let $n = \text{lcm}(p_1, \dots, p_k)$ and consider the strings $u_i = v_i = a^{i-1}$ for $1 \leq i \leq n+1$. The corresponding $(n+1) \times (n+1)$ matrix M is defined by

$$M_{i,j} = \begin{cases} 0, & \text{if } i+j = n+2 \text{ or if } i=j = n+1, \\ 1, & \text{otherwise.} \end{cases}$$

Then its determinant can be calculated by first subtracting the first row from the rest of the rows, and then by adding each row to the first row:

$$\det M = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 0 & 1 \\ 1 & 1 & 1 & \dots & 0 & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & \dots & 1 & 1 & 1 \\ 1 & 0 & 1 & \dots & 1 & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \\ 0 & 0 & 0 & \dots & -1 & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & -1 & \dots & 0 & 0 & 1 \\ 0 & -1 & 0 & \dots & 0 & 0 & 1 \\ -1 & 0 & 0 & \dots & 0 & 0 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 & n-1 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \\ 0 & 0 & 0 & \dots & -1 & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & -1 & \dots & 0 & 0 & 1 \\ 0 & -1 & 0 & \dots & 0 & 0 & 1 \\ -1 & 0 & 0 & \dots & 0 & 0 & 0 \end{vmatrix} = (-1)^{\lfloor \frac{n}{2} \rfloor} \cdot (n-1).$$

Since the determinant is non-zero, the matrix has full rank $n + 1$, and accordingly, by Schmidt's Theorem, every UFA for this language must have at least $n + 1$ states. \square

The above lemma gives a $g(n)$ lower bound on the NFA–UFA transformation, while the $g(n) + n^2$ upper bound is obtained by transforming an NFA to a DFA. This leads to the following theorem.

Theorem 4. *For every $n \geq 1$, the number of states in a UFA sufficient and, in the worst case, necessary to represent languages recognized by n -state NFAs is $g(n) + O(n^2) = e^{(1+o(1))\sqrt{n \ln n}}$.*

6. Complementing unary UFAs

For any $n \geq 1$, let $f(n)$ be the least such integer, that for every n -state UFA over a unary alphabet, the complement of the language it recognizes is representable by an $f(n)$ -state UFA. The function $f: \mathbb{N} \rightarrow \mathbb{N}$ is called the *state complexity* of complementation for unary UFAs, and the language L_n representable by an n -state UFA, the complement of which requires $f(n)$ states, is called the *witness language*.

The complexity of complementing DFAs and NFAs is well investigated. Complementing a DFA is trivial, because it is sufficient to complement its set of accepting states: hence, for DFAs, the state complexity of complementation is n . Representing the complement of an n -state NFA over a two-letter alphabet, as shown by Birget [3], may require an NFA with up to 2^n states; for unary NFAs, Holzer and Kutrib [9] proved that the state complexity of complementation is $g(n) + O(n^2)$. In both cases, complementing some NFAs basically requires determinizing them, and the witness languages are very similar to those for the NFA–DFA transformation.

The situation with UFAs is rather complicated. Consider the following facts. For alphabets with at least two letters, the UFA–DFA tradeoff is 2^n [18], that is, the same as the NFA–DFA tradeoff; for a unary alphabet, the UFA–DFA tradeoff is $\tilde{g}(n) + O(n) = e^{\Theta(\sqrt[3]{n \ln^2 n})}$, which is more or less comparable to the NFA–DFA tradeoff, $g(n) + O(n^2) = e^{(1+o(1))\sqrt{n \ln n}}$. However, for all known languages that witness these UFA–DFA tradeoffs—those that have small UFAs, but require large DFAs—one can change the sets of accepting states in their small UFAs and obtain UFAs for their complements. Up to date, not a single example of a UFA is known, which would require a larger UFA to represent its complement.

The results on unary UFAs obtained in this section represent the first attempt at analyzing the complexity of complementing UFAs. On the one hand, for a substantial class of UFAs, a UFA for their complement can be constructed by changing the set of accepting states, like in the case of DFAs. On the other hand, it shall be proved that complementing some specially constructed UFAs requires additional states.

To begin with, the following subclass of UFAs allows efficient complementation.

Lemma 6. *Let $A = (\Sigma, Q, q_0, \delta, F)$ be a unary UFA in Chrobak normal form recognizing an infinite language, and assume that there exists a number p that divides the length of every cycle, and the offsets f, f' of every two accepting states $r_{i,f}, r_{j,f'} \in F$ with $i \neq j$ are different modulo p . Then there exists and can be effectively constructed a set F' , such that $A' = (\Sigma, Q, q_0, \delta, F')$ is a UFA recognizing $\overline{L(A)}$.*

Proof. Under these assumptions, the set $\{0, \dots, p-1\}$ is partitioned into disjoint sets S_1, \dots, S_k , such that a state $r_{i,f}$ may be accepting only if the number f modulo p is in S_i . The set S_i is thus the “domain of expertise” of the i -th cycle. In other words, a string $a^{\ell+n}$ may be accepted only in the (uniquely determined) i -th cycle with $(n \bmod p) \in S_i$.

Then the new set of accepting states is defined as follows:

$$F' = \{q_i \mid q_i \notin F\} \cup \{r_{i,f} \mid (f \bmod p) \in S_i, r_{i,f} \notin F\}.$$

The conditions of the Criterion of Unambiguity are still met for the new automaton (that is, it remains a UFA), because its cycles have the same “domains of expertise” as in the original UFA.

To see that the new UFA recognizes the complement of the language of the original UFA, consider a string $a^{\ell+n}$, let i be the number n taken modulo p and let f be n taken modulo p_i . Then $a^{\ell+n}$ is accepted by the original automaton if and only if $r_{i,f} \in F$. At the same time, by the construction, this string is accepted by the new automaton if and only if $r_{i,f} \notin F$. \square

In particular, this lemma is applicable to all UFAs with $k = 2$ cycles, such as the one in Figure 1. But for $k \geq 3$, the lengths of the cycles need not be all divided by a single common divisor, which gives examples of UFAs not covered by the above lemma. Sometimes the lengths of the cycles may have a single common divisor, yet this common divisor is not enough to separate the accepting states as per the Criterion of Unambiguity, and the separation is based on larger gcds of individual pairs of cycles. The following example illustrates the latter case.

Example 1. Let $k = 3$ and consider cycle lengths $p_1 = 8$, $p_2 = 10$ and $p_3 = 12$, where $\gcd(8, 10) = 2$, $\gcd(8, 12) = 4$ and $\gcd(10, 12) = 2$. Then the numbers $f_1 = 7$, $f_2 = 8$ and $f_3 = 9$ satisfy the condition in the Criterion of Unambiguity, as $7 \not\equiv 8 \pmod{2}$, $7 \not\equiv 9 \pmod{4}$ and $8 \not\equiv 9 \pmod{2}$. This leads to a UFA with $1 + 8 + 10 + 12 = 31$ states recognizing the language $a^8(a^8)^* \cup a^9(a^{10})^* \cup a^{10}(a^{12})^*$, which is a witness language for $f_{UFA-DFA}(31) = \text{lcm}(8, 10, 12) + 1 = 121$.

However, $\gcd(8, 10, 12) = 2$ and $7 \equiv 9 \pmod{2}$, and thus Lemma 6 is not applicable to this UFA, and would not be applicable for any choice of offsets f_1, f_2, f_3 .

The next lemma considers the case of three cycles that have no common divisor. It turns out that representing the complement of such a language requires a UFA with a greater number of states.

Lemma 7. Let p_1, p_2, p_3 be any three pairwise distinct primes. Then the language $L = L_1 \cup L_2 \cup L_3$, where

$$\begin{aligned} L_1 &= a\{a^{p_1}, a^{2p_1}, \dots, a^{(p_2-1)p_1}\}(a^{p_1 p_2})^*, \\ L_2 &= a\{a^{p_2}, a^{2p_2}, \dots, a^{(p_3-1)p_2}\}(a^{p_2 p_3})^* \quad \text{and} \\ L_3 &= a\{a^{p_3}, a^{2p_3}, \dots, a^{(p_1-1)p_3}\}(a^{p_1 p_3})^*, \end{aligned}$$

has a UFA with $p_1 p_2 + p_2 p_3 + p_1 p_3 + 1$ states, while every NFA for \bar{L} contains at least $p_1 p_2 p_3$ states.

Proof. The construction of the UFA for L is straightforward. It has a tail of length 1 and three cycles of length $p_1 p_2$, $p_2 p_3$ and $p_1 p_3$, with accepting states $r_{1, \ell_1 p_1}$ for $\ell_1 \in \{1, \dots, p_2-1\}$, $r_{2, \ell_2 p_2}$ for $\ell_2 \in \{1, \dots, p_3-1\}$ and $r_{3, \ell_3 p_3}$ for $\ell_3 \in \{1, \dots, p_1-1\}$. Such a UFA for $p_1 = 3$, $p_2 = 4$ and $p_3 = 5$ is presented in Figure 3. To see that the condition of the Criterion of Unambiguity is satisfied, consider the offsets of accepting states in the cycles, taken modulo p_1 , p_2 and p_3 :

	(mod p_1)	(mod p_2)	(mod p_3)
$\{a^{p_1}, a^{2p_1}, \dots, a^{(p_2-1)p_1}\} \pmod{p_1 p_2}$	0	$\{1, \dots, p_2-1\}$	anything
$\{a^{p_2}, a^{2p_2}, \dots, a^{(p_3-1)p_2}\} \pmod{p_2 p_3}$	anything	0	$\{1, \dots, p_3-1\}$
$\{a^{p_3}, a^{2p_3}, \dots, a^{(p_1-1)p_3}\} \pmod{p_1 p_3}$	$\{1, \dots, p_1-1\}$	anything	0

Now the offsets of accepting states in the first and the second cycles are different modulo $p_2 = \gcd(p_1 p_2, p_2 p_3)$, the second and the third cycles are separated modulo $p_3 = \gcd(p_2 p_3, p_1 p_3)$, and the first and the third cycles have the offsets inequivalent modulo $p_1 = \gcd(p_1 p_2, p_1 p_3)$.

In order to show that no NFA for \bar{L} can have fewer than $p_1 p_2 p_3$ states, it is sufficient to establish the following statement:

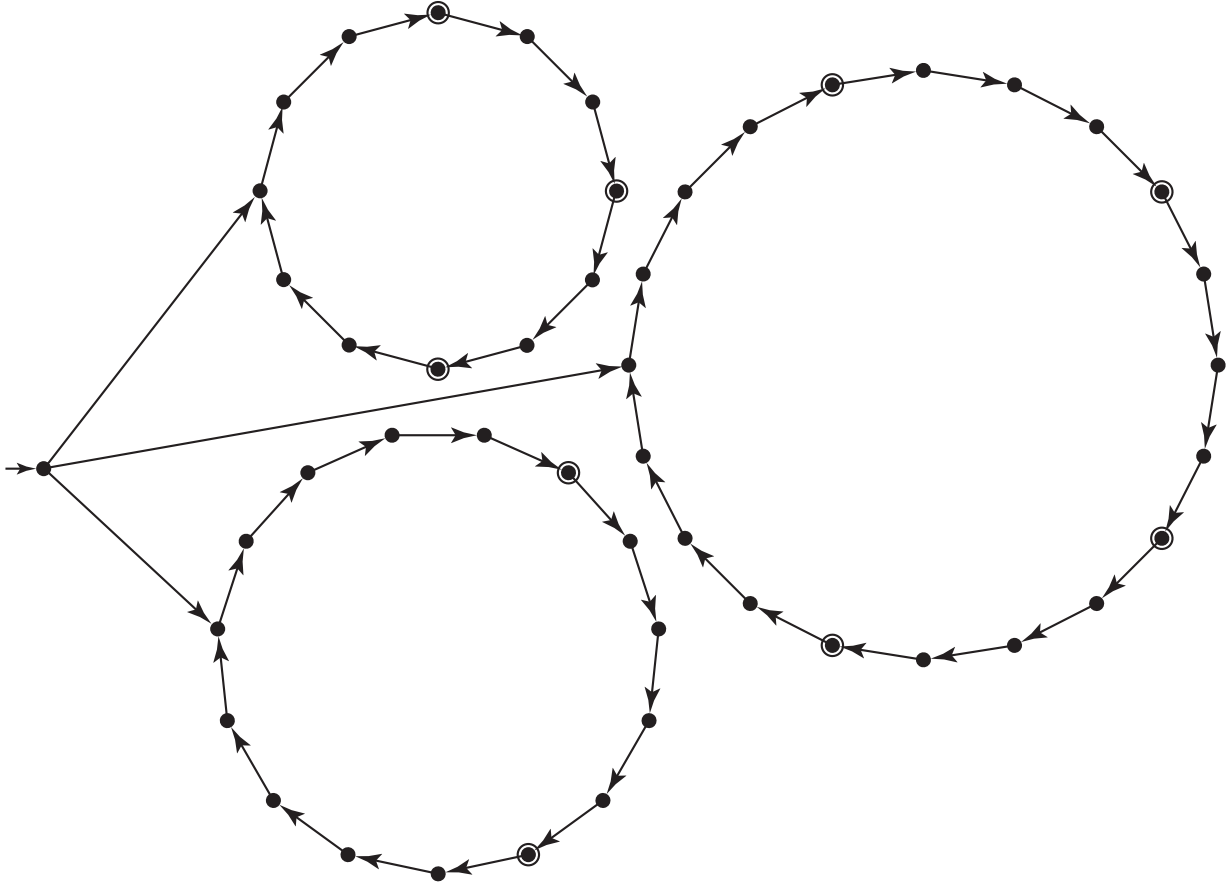


Figure 3: A 48-state unary UFA that requires a 60-state UFA for its complement.

Claim 2. Consider any infinite regular subset of \bar{L} that contains at least one string a^{1+n} with $n \equiv 0 \pmod{p_1 p_2 p_3}$ in its periodic part. Then the period of the subset is divisible by $p_1 p_2 p_3$.

Let p be the period of this subset. By the symmetry, it is sufficient to prove that p is a multiple of p_1 . In order to obtain a contradiction, suppose that $p \not\equiv 0 \pmod{p_1}$. Let a^{1+n} be any string in the periodic part of this subset that satisfies $n \equiv 0 \pmod{p_1 p_2 p_3}$. Then the string a^{1+n+pp_3} belongs to this subset as well. It is claimed that $n + pp_3 \not\equiv 0 \pmod{p_1}$: indeed, $n \equiv 0 \pmod{p_1}$ by the assumption, and $pp_3 \not\equiv 0 \pmod{p_1}$, because p_3 is relatively prime with p_1 , and p is not divisible by p_1 . On the other hand, $n + pp_3 \equiv 0 \pmod{p_3}$. Therefore, $a^{1+n+pp_3} \in L_3$, which contradicts the assumption that $a^{1+n+pp_3} \in \bar{L}$. The contradiction obtained completes the proof of Claim 2.

Consider any NFA recognizing the language \bar{L} , and the equivalent NFA in Chrobak normal form. It is sufficient to prove that one of the cycles in the normalized NFA must be of length at least $p_1 p_2 p_3$. Since, by Proposition 3, the combined length of the cycles in the new NFA cannot exceed the total number of states in the original NFA, the latter must have at least $p_1 p_2 p_3$ states.

Since the language \bar{L} contains infinitely many strings a^{1+n} with $n \equiv 0 \pmod{p_1 p_2 p_3}$, any NFA in Chrobak normal form recognizing this language must have a cycle containing at least one accepting state, in which such a string a^{1+n} is accepted. The set of strings accepted in this cycle forms a subset of \bar{L} meeting the conditions of Claim 2. Then the claim asserts that the length of this cycle must be a multiple of $p_1 p_2 p_3$, which completes the proof of the lemma. \square

In particular, applying this lemma to $p_1 = 3$, $p_2 = 4$ and $p_3 = 5$ gives a language recognized by a UFA

with $48 = 12 + 20 + 15 + 1$ states given in Figure 3, while its complement requires a UFA with at least $60 = 3 \cdot 4 \cdot 5$ states.

Witness languages of the form constructed in Lemma 7 lead to the following fairly modest lower bound.

Theorem 5. *The state complexity of complementation for UFAs over a unary alphabet is greater than $\frac{1}{42}n\sqrt{n}$ (for all $n \geq 867$) and at most $f_{\text{UFA-DFA}}(n)$.*

Proof. The upper bound is immediate, since every UFA can be determinized and then complemented.

The proof of the lower bound relies on a result of Ramanujan [28] that for every $m \geq 17$ there are at least three primes between $\frac{m}{2}$ and m . Let n be any number greater than $3 \cdot 17^2 = 867$. Then there exist three primes p_1, p_2, p_3 with

$$\sqrt{\frac{n}{12}} < p_1 < p_2 < p_3 \leq \sqrt{\frac{n}{3}}.$$

By Lemma 7, there is a language L recognized by a UFA with

$$p_1 p_2 + p_2 p_3 + p_1 p_3 + 1 \leq 3\sqrt{\frac{n}{3}} \left(\sqrt{\frac{n}{3}} - 2 \right) + 1 = n - 6\sqrt{\frac{n}{3}} + 1 \leq n$$

states, while every UFA for \bar{L} needs to have at least

$$p_1 p_2 p_3 > \left(\sqrt{\frac{n}{12}} \right)^3 = \left(\frac{1}{12} \right)^{\frac{3}{2}} n\sqrt{n} > \frac{1}{42} n\sqrt{n}$$

states, as claimed. \square

Better lower bounds can be obtained from the following generalization of Lemma 7 to any odd number of cycles:

Lemma 8. *Let $k \geq 1$ and let p_1, \dots, p_{2k+1} be any pairwise relatively prime numbers. Then the language $L = \bigcup_{i=1}^{2k+1} L_i$, where*

$$L_i = \{ a^{1+n} \mid n \not\equiv 0 \pmod{p_i}, n \equiv 0 \pmod{p_{i+1} \cdots p_{i+k}} \}$$

(with all arithmetic in subscripts done modulo $2k+1$), has a UFA with $1 + \sum_{i=1}^{2k+1} p_i p_{i+1} \cdots p_{i+k}$ states, while every NFA for \bar{L} contains at least $p_1 \cdots p_{2k+1}$ states.

Proof. A UFA for L has a tail of length 1 and $2k+1$ cycles, with each i -th cycle of length $\pi_i = p_i p_{i+1} \cdots p_{i+k}$ containing accepting states $r_{i, \ell p_{i+1} \cdots p_{i+k}}$ for $\ell \in \{1, \dots, p_i - 1\}$.

To see that the condition of the Criterion of Unambiguity is satisfied, consider any i -th and any j -th cycles with $i \neq j$. Since the difference of i and j modulo $2k+1$ is at most k , either the number p_i is in $\{p_{j+1}, \dots, p_{j+k}\}$, or p_j belongs to $\{p_{i+1}, \dots, p_{i+k}\}$. Assume, without loss of generality, that the former is the case. Then p_i is a common divisor of π_i and π_j , and for every two accepting states $r_{i, \ell p_{i+1} \cdots p_{i+k}}$ and $r_{j, \ell' p_{j+1} \cdots p_{j+k}}$, the number $\ell p_{i+1} \cdots p_{i+k}$ is non-zero modulo p_i , while $\ell' p_{j+1} \cdots p_{j+k}$ is divisible by p_i . Therefore, $\ell p_{i+1} \cdots p_{i+k} \not\equiv \ell' p_{j+1} \cdots p_{j+k} \pmod{\gcd(\pi_i, \pi_j)}$.

A lower bound on the size of any NFA recognizing \bar{L} is based upon the following property:

Claim 3. *Every infinite regular subset of \bar{L} with its periodic part containing any string a^{1+n} with $n \equiv 0 \pmod{p_1 \cdots p_{2k+1}}$ has period divisible by $p_1 \cdots p_{2k+1}$.*

The proof of the claim generalizes that of Claim 2 in the previous argument. If p is the period of such a subset and it is not divisible by some p_i , for any $i \in \{1, \dots, 2k+1\}$, then the string $w = a^{1+n+pp_1 \cdots p_{i-1} p_{i+1} \cdots p_{2k+1}}$ belongs to this subset as well. The claim is that $n+pp_1 \cdots p_{i-1} p_{i+1} \cdots p_{2k+1} \not\equiv 0 \pmod{p_i}$. First, $n \equiv 0 \pmod{p_i}$ by the assumption, and secondly, since each p_j with $j \neq i$ is relatively prime with p_i and p is not divisible by p_i , the number $pp_1 \cdots p_{i-1} p_{i+1} \cdots p_{2k+1}$ is non-zero modulo p_i . At the same time, $n+pp_1 \cdots p_{i-1} p_{i+1} \cdots p_{2k+1} \equiv n \equiv 0 \pmod{p_j}$ for every $j \neq i$. Therefore, the string w must belong to L_i , which contradicts the assumption that $w \in \bar{L}$.

Now consider that an NFA for the language \bar{L} should accept all strings in $a(p_1 \cdots p_{2k+1})^*$, and hence, as in the proof of Lemma 7 above, the equivalent NFA in Chrobak normal form must accept infinitely many of these strings in a certain cycle. Applying the above claim to the subset of \bar{L} accepted in that cycle shows that the cycle's length must be a multiple of $p_1 \cdots p_{2k+1}$. Therefore, the original NFA has at least $p_1 \cdots p_{2k+1}$ states. \square

Lemma 9. *Let $k \geq 1$. Then, for all $n \geq (2k+1)(4(2k+1) \ln 4(2k+1))^{k+1}$, the number of states in an NFA necessary to represent complements of n -state UFAs over a unary alphabet is at least $\frac{1}{2^{2k+1}(2k+1)^2} \cdot n^{2-\frac{1}{k+1}}$.*

Proof. Let r_i denote i -th Ramanujan prime, that is, the smallest integer, such that for every $m \geq r_i$ there are at least i primes between $\frac{m}{2}$ and m . The existence of such a number for every i was proved by Ramanujan [28], and the first values are $r_1 = 2, r_2 = 11, r_3 = 17, r_4 = 29, r_5 = 41, r_6 = 47, r_7 = 59$.

Let n be any number greater than $(2k+1) \cdot (r_{2k+1})^{k+1}$ (for $k = 1, 2, 3, \dots$, this means that $n \geq 867, 344605, 84821527, \dots$). Then there exist $2k+1$ primes p_1, \dots, p_{2k+1} with

$$\frac{1}{2} \sqrt[k+1]{\frac{n}{2k+1}} < p_1 < \dots < p_{2k+1} \leq \sqrt[k+1]{\frac{n}{2k+1}}.$$

According to Lemma 8, there exists a language L recognized by a UFA with

$$1 + \sum_{i=1}^{2k+1} p_i p_{i+1} \cdots p_{i+k} \leq (2k+1) \sqrt[k+1]{\frac{n}{2k+1}} \left(\left(\frac{n}{2k+1} \right)^{\frac{k}{k+1}} - 2 \right) + 1 = n - 2(2k+1) \sqrt[k+1]{\frac{n}{2k+1}} + 1 \leq n$$

states, but every UFA for \bar{L} must have at least

$$p_1 \cdots p_{2k+1} \geq \left(\frac{1}{2} \sqrt[k+1]{\frac{n}{2k+1}} \right)^{2k+1} = \frac{1}{2^{2k+1}(2k+1)^{\frac{2k+1}{k+1}}} \cdot n^{\frac{2k+1}{k+1}} \geq \frac{1}{2^{2k+1}(2k+1)^2} \cdot n^{2-\frac{1}{k+1}}$$

states, which proves the lower bound.

It remains to estimate the least n , to which the above argument applies. The following bounds on Ramanujan primes were recently obtained by Sondow [33]: $2i \ln 2i < r_i < 4i \ln 4i$. Then $(2k+1) \cdot (r_{2k+1})^{k+1} < (2k+1)(4(2k+1) \ln 4(2k+1))^{k+1}$. \square

Theorem 6. *The state complexity of complementation for UFAs over a unary alphabet is at least $n^{2-o(1)}$ and at most $f_{UFA-DFA}(n)$.*

Proof. According to Lemma 9, the function $f(n)$ defined by

$$f(n) = \max_{k: n \geq n_0(k)} \frac{1}{2^{2k+1}(2k+1)^2} \cdot n^{2-\frac{1}{k+1}} = \max_{k: n \geq n_0(k)} n^{2-\frac{1}{k+1} - \log_n(2^{2k+1}(2k+1)^2)},$$

where $n_0(k) = \lceil (2k+1)(4(2k+1) \ln 4(2k+1))^{k+1} \rceil$, is a lower bound on the state complexity of complementation. Define a new function $h(n)$, so that $f(n) = n^{2-h(n)}$. The goal is to prove that $\lim_{n \rightarrow \infty} h(n) = 0$.

Fix an arbitrary real number $\varepsilon > 0$ and set $k = \lfloor \frac{1}{\varepsilon} \rfloor$, so that $\frac{1}{k+1} < \varepsilon$. Let $\hat{n} = \max(n_0(k), n_1(k))$, where $n_1(k) = (2^{2k+1}(2k+1)^2)^{\frac{1}{\varepsilon - \frac{1}{k+1}}}$. Then, for every $n \geq \hat{n}$, since $n \geq n_0(k)$,

$$f(n) \geq n^{2-\frac{1}{k+1} - \log_n(2^{2k+1}(2k+1)^2)}.$$

At the same time, $n \geq n_1(k)$ implies that $n^{\varepsilon - \frac{1}{k+1}} \geq 2^{2k+1}(2k+1)^2$, and hence $\varepsilon - \frac{1}{k+1} \geq \log_n(2^{2k+1}(2k+1)^2)$. Accordingly,

$$f(n) \geq n^{2-\varepsilon + (\varepsilon - \frac{1}{k+1}) - \log_n(2^{2k+1}(2k+1)^2)} \geq n^{2-\varepsilon},$$

and therefore $h(n) \leq \varepsilon$. \square

It is interesting to note that the witness languages constructed in this section, such as the UFA in Figure 3, use a significantly larger overlap between the prime factorizations of their cycle lengths than required by the definition of a UFA. This is done in order to avoid the case described in Lemma 6, for which the complementation is efficient, as well as any variants of that case. On the other hand, the hardest languages for the UFA–DFA tradeoff presented in Table 1, which, by their nature, have the least possible overlap between the cycle lengths, can all be complemented by changing the set of accepting states (even though Lemma 6 is not general enough to cover all cases of direct complementation). This suggests that the complexity of complementing unary UFAs is probably much lower than the UFA–DFA tradeoff. Narrowing the gap between the lower bound and the upper bound given in Theorem 6 is suggested for future research.

7. State complexity of intersection and Kleene star

Consider the state complexity of intersection of two UFAs, and of the Kleene star of a single UFA. These two operations can be handled by the known methods developed for other kinds of finite automata.

Intersection has state complexity mn both for DFAs [21, 36] and for NFAs [9], and both over unary and larger alphabets. It maintains the same complexity for UFAs:

Lemma 10. *For every alphabet Σ and for all $m, n \geq 1$, the intersection of any two UFAs over Σ with m and n states is recognized by a UFAs with mn states.*

The proof is by the standard *direct product construction*, which always produces a UFA for UFA arguments.

A matching lower bound for select values of m, n is already known:

Proposition 9 (Holzer, Kutrib [9]). *For all relatively prime $m, n \geq 2$, the language $(a^{mn})^* = (a^m)^* \cap (a^n)^*$ requires an NFA with at least mn states.*

Theorem 7. *The state complexity of intersection for UFAs over a unary alphabet is at most mn . This bound is reachable for all relatively prime m, n .*

The last operation to be considered is the *Kleene star*, $L^* = \bigcup_{k \geq 0} L^k$: its state complexity for unary DFAs is $(n - 1)^2 + 1$, obtained by Yu, Zhuang and Salomaa [36, Thm. 5.3]. Their result extends to the UFAs, in spite of the differences between the two models.

Lemma 11 (Yu, Zhuang and Salomaa [36]). *For every language $L \subseteq a^*$ recognized by an n -state NFA with $n \geq 2$, there exists a DFA for L^* with $(n - 1)^2 + 1$ states.*

Strictly speaking, Yu, Zhuang and Salomaa [36] established this result for L represented by a DFA. However, with a minor adjustment, their argument proves Lemma 11 as stated. The proof is included for completeness.

Proof. Let $A = (\{a\}, Q, Q_0, \delta, F)$ be any NFA, and assume that it accepts any non-empty string: if it doesn't, then $L^* = \{\varepsilon\}$ has a 2-state DFA. Let a^m be the shortest non-empty string in L . Then $m \leq n$. If $m = n$, then A is a simple cycle with $Q_0 = F = \{q_0\}$, and therefore $L = L^*$ and A is an n -state DFA for L^* . Hence, assume $m \leq n - 1$ and construct an NFA $B = (\{a\}, Q, Q_0, \delta', F)$ for the language L^+ by adding ε -transitions from each state in F to each state in Q_0 . It remains to demonstrate that applying the subset construction to B yields a DFA with at most $(n - 1)^2 + 1$ states, and that this DFA can be modified to accept ε (and thus recognize L^*) without exceeding this number of states.

Since $a^m \in L(A)$, the set $\delta(Q_0, a^m) \subseteq Q$ contains an accepting state $q \in F$, and accordingly $\delta'(Q_0, a^m)$ contains Q_0 as a subset, reached by ε -transitions from q . Based on this fact, it is claimed that the sets of states $\delta'(Q_0, a^{im})$ with $i \geq 0$ form an ascending chain:

$$Q_0 = \delta'(Q_0, \varepsilon) \subseteq \delta'(Q_0, a^m) \subseteq \delta'(Q_0, a^{2m}) \subseteq \dots \subseteq \delta'(Q_0, a^{im}) \subseteq \dots \quad (2)$$

The first inclusion has already been proved, and every next inclusion is inferred from the previous one as follows. Assume that $\delta'(Q_0, a^{(i-1)m}) \subseteq \delta'(Q_0, a^{im})$, that is, $\delta'(Q_0, a^{im}) = \delta'(Q_0, a^{(i-1)m}) \cup \tilde{Q}$ for some set $\tilde{Q} \subseteq Q$. Then

$$\begin{aligned} \delta'(Q_0, a^{(i+1)m}) &= \delta'(\delta'(Q_0, a^{im}), a^m) = \delta'(\delta'(Q_0, a^{(i-1)m}) \cup \tilde{Q}, a^m) = \\ &= \delta'(\delta'(Q_0, a^{(i-1)m}), a^m) \cup \delta'(\tilde{Q}, a^m) \supseteq \delta'(Q_0, a^{im}), \end{aligned}$$

which proves the induction step and establishes the chain of inclusions (2).

Let k be the least number with $\delta'(Q_0, a^{km}) = \delta'(Q_0, a^{(k+1)m})$, so that the sequence of sets (2) is strictly increasing up to $\delta'(Q_0, a^{km})$, and the rest of its elements are the same as $\delta'(Q_0, a^{km})$. Since all these sets are subsets of Q with $|Q| = n$, every i -th element has cardinality $|\delta'(Q_0, a^{im})| \geq i + 1$, and hence the number k is at most $n - 1$.

Consider first the case when the sequence (2) converges to the set of all states, that is, $\delta'(Q_0, a^{km}) = Q$. Then the transition from this set by a leads to the same set of (all) states, because $\delta'(Q, a) \supseteq \delta'(\delta'(Q_0, a^{km-1}), a) = Q$. Therefore, the DFA for the language L^+ , obtained by the subset construction, has a tail of length km and a cycle of length 1 (and hence L^+ is co-finite). As long as $k > 0$, the initial state of this DFA is guaranteed to be outside of the cycle, and can be set as accepting to obtain a DFA for L^* with the same number of states. The total number of states in this DFA is $km + 1 \leq (n - 1)(n - 1) + 1$, as claimed. If $k = 0$, then $L = a^*$, and the language $L^* = L$ has a 1-state DFA.

If $\delta'(Q_0, a^{km}) \neq Q$, then $|\delta'(Q_0, a^{km})| \leq n - 1$ and $k \leq n - 2$. Accordingly, the DFA for L^+ obtained by the subset construction has a tail of length km and a cycle of length m , so that the total number of states is $k(m + 1) \leq (n - 2)n = (n - 1)^2 - 1$. Now, if $k \geq 1$, then the initial state of the DFA can be set as accepting, so that it recognizes L^* ; and if $k = 0$, then the DFA for L^+ contains $m \leq n - 1$ states, and an extra state can be added to obtain an n -state DFA for L^* . \square

As in the case of DFAs, lower bounds on the star of UFAs use witness languages with a co-finite star. It turns out that for co-finite unary languages, UFAs are no more succinct than DFAs.

Lemma 12. *Let $L \subseteq a^*$ be a co-finite language, let a^m be the longest string not in L . Then the smallest NFA in Chrobak normal form for L contains $m + 2$ states and coincides with the smallest DFA for L .*

Proof. The construction of an $(m + 2)$ -state DFA is obvious.

Let $A = (\{a\}, Q, q_0, \delta, F)$ be any NFA in Chrobak normal form recognizing L . Let it have a tail of length ℓ and $k \geq 1$ cycles of length p_1, \dots, p_k . It is claimed that every string of length ℓ or more is accepted by A .

Let $n \geq \ell$ and consider the string $a^{\ell + m \cdot \text{lcm}(p_1, \dots, p_k)}$, which is longer than a^m and hence is in L . Let this string be accepted in an i -th cycle, that is, in the state $r_{i, n - \ell + m \cdot \text{lcm}(p_1, \dots, p_k)}$, where the arithmetic is modulo p_i . Since this is the same state as $r_{i, n - \ell}$, the string a^n is accepted in that state as well.

As the string a^m should not be accepted by A , the number m is at most $\ell - 1$. Therefore, the tail of A contains at least $m + 1$ states, while the cycles consist of at least one state, which proves the lower bound of $m + 2$ states. \square

Theorem 8. *For every $n \geq 1$, the Kleene star of every n -state UFA is representable by a UFA with $(n - 1)^2 + 1$ states, and this number of states is necessary in the worst case.*

Proof. The upper bound is given in Lemma 11.

For the lower bound, consider the language $L = a^{n-1}(a^n)^*$. As noted by Yu, Zhuang and Salomaa [36], its star L^* is co-finite, and the longest string not belonging to it is $a^{(n-2)n}$. Then, by Lemma 12, every UFA for L^* requires at least $(n - 2)n + 2 = (n - 1)^2 + 1$ states. \square

8. Future work

The investigation of unary UFAs with the largest equivalent DFAs has led to a new variant of Landau's function, and this function deserves a further study. In particular, it remains to understand the form of

cycle lengths, on which the maximum least common multiple is achieved. It would also be interesting to obtain a more precise approximation than the given $\tilde{g}(n) = e^{\Theta(\sqrt[3]{n \ln^2 n})}$, perhaps an estimation of the form $\tilde{g}(n) = e^{C \sqrt[3]{n \ln^2 n} (1+o(1))}$. Another question of interest is to determine an efficient method of computing the values of \tilde{g} : the brute-force calculations carried out by the author to fill Table 2 were sufficient to calculate the values of $\tilde{g}(n)$ only for n up to 165.

The complexity of operations on UFAs—in particular, the complexity of complementing them—is left as the main open problem. It would be a surprise, if the complement of an n -state UFA could always be represented using as few as n^2 states; but it is also unlikely to require as many as $f_{\text{UFA-DFA}}(n)$ states. Obtaining any tighter bounds requires a deeper analysis than provided in this paper, and a better understanding of unary UFAs. This is a goal for future research.

Acknowledgements

I am indebted to Oksana Yakimova for kindly explaining me what to do with the integral $\int_1^k \sqrt{x \ln x} dx$. I am grateful to Galina Jirásková, Michael Domaratzki and Hermann Gruber for their helpful comments on the manuscript. Thanks are due to the anonymous referees from MFCS 2010 for alerting me of the previous work on unary UFAs by Ravikumar and Ibarra [29], and to the referee from I&C for noticing a mistake in the author’s earlier rendering of the proof by Yu et al. [36] (Lemma 11).

While this paper has been in the publication cycle between the technical report [25, 26] and this final journal version, a lower bound $e^{\Omega(\sqrt[3]{n \ln^2 n})}$ on the UFA–DFA tradeoff, similar to the one in Theorem 3, was independently established by Geffert and Pighizzini [7, 8], who constructed witness languages by multiplying the cycle lengths p_1, \dots, p_k from the definition of Landau’s function by k each, and inferred the asymptotics from Landau’s [16] estimation of $g(n)$.

References

- [1] M. Anselmo, M. Madonia, “Some results on the structure of unary unambiguous automata” *Advances in Applied Mathematics*, 47:1 (2011), 88–101.
- [2] E. Bach, J. Shallit, *Algorithmic Number Theory, Vol. 1: Efficient Algorithms*, MIT Press, 1996.
- [3] J. C. Birget, “Partial orders on words, minimal elements of regular languages, and state complexity”, *Theoretical Computer Science*, 119 (1993), 267–291.
- [4] H. Björklund, W. Martens, “The tractability frontier for NFA minimization”, *Journal of Computer and System Sciences*, 78:1 (2012), 198–210.
- [5] M. Chrobak, “Finite automata and unary languages”, *Theoretical Computer Science*, 47 (1986), 149–158; errata: 302:1–3 (2003), 497–498.
- [6] V. Geffert, C. Mereghetti, G. Pighizzini, “Complementing two-way finite automata”, *Information and Computation*, 205:8 (2007), 1173–1187.
- [7] V. Geffert, G. Pighizzini, “Pairs of complementary unary languages with ‘balanced’ nondeterministic automata”, *LATIN 2010: Theoretical Informatics, 9th Latin American Symposium* (Oaxaca, Mexico, 19–23 April 2010), LNCS 6034, 196–207.
- [8] V. Geffert, G. Pighizzini, “Pairs of complementary unary languages with ‘balanced’ nondeterministic automata”, *Algorithmica*, to appear.
- [9] M. Holzer, M. Kutrib, “Nondeterministic descriptonal complexity of regular languages”, *International Journal of Foundations of Computer Science*, 14 (2003), 1087–1102.
- [10] M. Holzer, M. Kutrib, “Nondeterministic finite automata—Recent results on the descriptonal and computational complexity”, *International Journal of Foundations of Computer Science*, 20:4 (2009), 563–580.
- [11] J. Hromkovič, S. Seibert, J. Karhumäki, H. Klauck, G. Schnitger, “Communication complexity method for measuring nondeterminism in finite automata”, *Information and Computation*, 172:2 (2002), 202–217.
- [12] A. W. Ingleton, “The rank of circulant matrices”, *Journal of the London Mathematical Society*, 31 (1956), 445–460.
- [13] T. Jiang, E. McDowell, B. Ravikumar, “The structure and complexity of minimal NFA’s over a unary alphabet”, *International Journal of Foundations of Computer Science*, 2:2 (1991), 163–182.
- [14] M. Kunc, A. Okhotin, “Describing periodicity in two-way deterministic finite automata using transformation semigroups”, *Developments in Language Theory* (DLT 2011, Milan, Italy, 19–22 July 2011), LNCS 6795, 324–336.
- [15] M. Kunc, A. Okhotin, “State complexity of operations on two-way deterministic finite automata over a unary alphabet”, *Descriptive Complexity of Formal Systems* (DCFS 2011, Limburg, Germany, 25–27 July 2011), LNCS 6808, 222–234.
- [16] E. Landau, “Über die Maximalordnung der Permutationen gegebenen Grades” (On the maximal order of permutations of a given degree), *Archiv der Mathematik und Physik, Ser. 3*, 5 (1903), 92–103.

- [17] H. Leung, “Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata”, *SIAM Journal on Computing*, 27:4 (1998), 1073–1082.
- [18] H. Leung, “Descriptive complexity of NFA of different ambiguity”, *International Journal of Foundations of Computer Science*, 16:5 (2005), 975–984.
- [19] Yu. Lyubich, “Bounds for the optimal determinization of nondeterministic automata”, *Sibirskii Matematicheskii Zhurnal*, 2 (1964), 337–355, in Russian.
- [20] R. Mandl, “Precise bounds associated with the subset construction on various classes of nondeterministic finite automata”, *7th Princeton Conference on Information and System Sciences*, 1973, 263–267.
- [21] A. N. Maslov, “Estimates of the number of states of finite automata”, *Soviet Mathematics Doklady*, 11 (1970), 1373–1375.
- [22] C. Mereghetti, G. Pighizzini, “Optimal simulations between unary automata”, *SIAM Journal on Computing*, 30:6 (2001), 1976–1992.
- [23] W. Miller, “The maximum order of an element of a finite symmetric group”, *American Mathematical Monthly*, 94:6 (1987), 497–506.
- [24] OEIS Foundation, Inc. (2011), *The On-Line Encyclopedia of Integer Sequences*, <http://oeis.org>.
- [25] A. Okhotin, “A study of unambiguous finite automata over a one-letter alphabet”, TUCS Technical Report 951, Turku Centre for Computer Science, September 2009.
- [26] A. Okhotin, “Unambiguous finite automata over a unary alphabet”, *Mathematical Foundations of Computer Science (MFCS 2010, Brno, Czech Republic, 23–27 August 2010)*, LNCS 6281, 556–567.
- [27] G. Pighizzini, J. Shallit, “Unary language operations, state complexity and Jacobsthal’s function”, *International Journal of Foundations of Computer Science*, 13:1 (2002), 145–159.
- [28] S. Ramanujan, “A proof of Bertrand’s postulate”, *Journal of the Indian Mathematical Society*, 11 (1919), 181–182.
- [29] B. Ravikumar, O. H. Ibarra, “Relating the type of ambiguity of finite automata to the succinctness of their representation”, *SIAM Journal on Computing*, 18:6 (1989), 1263–1282.
- [30] J. B. Rosser, “The n th prime is greater than $n \ln n$ ”, *Proceedings of the London Mathematical Society*, 45 (1938), 21–44.
- [31] J. B. Rosser, “Explicit bounds for some functions of prime numbers”, *American Journal of Mathematics*, 63:1 (1941), 211–232.
- [32] E. M. Schmidt, *Succinctness of Description of Context-Free, Regular and Unambiguous Languages*, Ph. D. thesis, Cornell University, 1978.
- [33] J. Sondow, “Ramanujan primes and Bertrand’s postulate”, *American Mathematical Monthly*, 116 (2009), 630–635.
- [34] R. E. Stearns, H. B. Hunt III, “On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata”, *SIAM Journal on Computing*, 14 (1985), 598–611.
- [35] M. Szalay, “On the maximal order in S_n and S_n^* ”, *Acta Arithmetica*, 37 (1980), 321–331.
- [36] S. Yu, Q. Zhuang, K. Salomaa, “The state complexity of some basic operations on regular languages”, *Theoretical Computer Science*, 125 (1994), 315–328.