Alexander Okhotin

A study of unambiguous finite automata over a one-letter alphabet

Turku Centre *for* Computer Science

TUCS Technical Report
No 951, September 2009

A study of unambiguous finite automata over a one-letter alphabet

Alexander Okhotin
Academy of Finland, *and*
Department of Mathematics, University of Turku, *and*
Turku Centre for Computer Science
Turku FIN–20014, Finland
alexander.okhotin@utu.fi

TUCS Technical Report
No 951, September 2009

Abstract

Nondeterministic finite automata (NFA) with at most one accepting computation on every input string are known as unambiguous finite automata (UFA). It is shown that every UFA over a unary alphabet $\Sigma = \{a\}$ can be transformed to the Chrobak normal form without adding any extra states. The normal form is then used to determine the exact number of states in DFAs needed to represent unary languages recognized by n-state UFAs; the growth rate of this function is $e^{\Theta(\sqrt[3]{n \ln^2 n})}$. The conversion of an n-state unary NFA to a UFA requires UFAs with $g(n)+O(n^2) = e^{\sqrt{n \ln n}(1+o(1))}$ states, where $g(n)$ is Landau's function. In addition, it is shown that the complement of n-state unary UFAs requires at least $n^{2-o(1)}$ states in an NFA.

Keywords: Finite automata, unary languages, ambiguity, descriptional complexity, state complexity, Landau's function.

TUCS Laboratory
Discrete Mathematics for Information Technology

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 Apparently, this family was first studied by Schmidt [19], whose unpublished thesis contains a interesting method of proving lower bounds for UFAs based upon the rank of certain matrices, and a $2^{\Omega(\sqrt{n})}$ lower bound on the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 [9, 10] and by Hromkovič et al. [6], who studied degrees of nondeterminism in finite automata. In particular, Leung [10] established a precise $2^n - 1$ UFA–DFA tradeoff. Computational complexity of testing properties of UFAs was studied by Stearns and Hunt [21] and recently by Björklund and Martens [2].

This paper considers UFA in the special case of an alphabet $\Sigma = \{a\}$. The main properties of DFAs and NFAs over a unary alphabet are quite different from the the case of a general alphabet. Lyubich [11] and Chrobak [3] have shown that in the unary case the DFA–NFA tradeoff is $g(n) + O(n^2)$, where $g(n) = e^{(1+o(1))\sqrt{n \ln n}}$ is the maximum order of an element in the group of permutations on n objects, known as *Landau's function* [8]. State complexity of basic operations on unary DFAs was first studied by Yu, Zhuang and K. Salomaa [22], and elaborated by Pighizzini and Shallit [17]. A similar study for unary NFAs was carried out by Holzer and Kutrib [5], and the hardest languages for complementation were further studied by Mera and Pighizzini [14]. Succinctness of two-way automata over a unary alphabet has received particular attention in the works of Chrobak [3], Mereghetti and Pighizzini [15] and Geffert et al. [4].

The first natural question about unary UFAs is whether they are nontrivial: that is, any more succinct than unary DFAs. The smallest example of a nontrivial UFA is presented in Figure 1, left; it 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. This UFA has $1 + 4 + 6 = 11$ states, while the smallest equivalent DFA shown on the right requires $1 + \mathrm{lcm}(4, 6) = 13$ states. This example motivates the study of unary UFAs, which is undertaken in the present paper.

It should be noted that the existing methods of proving lower bounds on the size of UFAs, based upon the matrix methods of Schmidt [19], are quite hard to apply in the case of a unary alphabet. For unary inputs, Schmidt's matrix belongs to a class of *circulant matrices*, and the problem of determining the rank of a circulant matrix of 0s and 1s, studied by Ingleton [7], is surprisingly hard in the general case. Unless the matrix for a particular language happens to be of some special form, finding its rank is difficult.

New methods of analysis are thus required, and they shall be derived from the earlier work on unary NFAs. Perhaps the most important basic

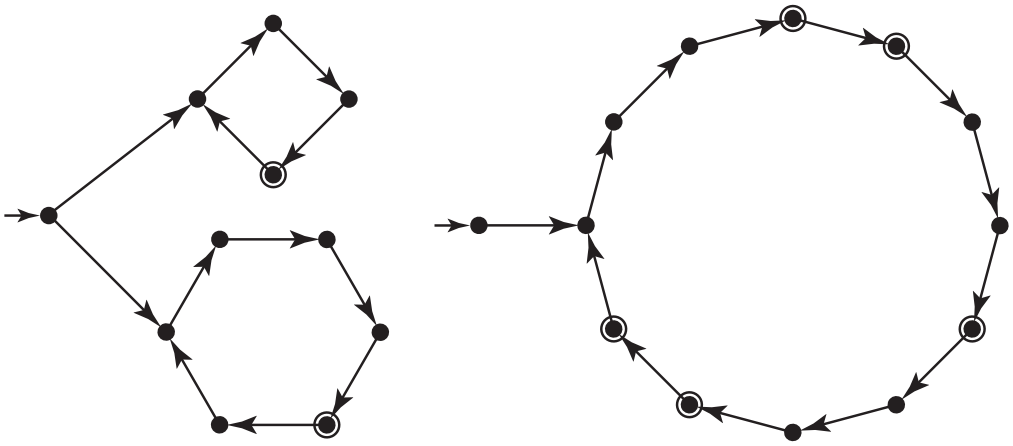

Figure 1: An 11-state unary UFA and the 13-state minimal equivalent DFA.

result on unary NFAs is the *Chrobak normal form*, in which there is one tail of states, ending with transitions into one or more disjoint cycles. It was proved by Chrobak [3] that every n-state NFA can be transformed to this normal form, with the cycles of combined length at most n and with the tail of length $O(n^2)$. This paper begins with refining Chrobak's transformation for the case of a UFA, eventually developing a transformation to the same normal form, but *without increasing the number of states*, and satisfying an additional condition specific to UFAs.

This normal form is then used to determine the precise tradeoff between UFAs and DFAs, which is expressed in terms of a more complicated variant of Landau's function, denoted $\widetilde{g}$. In particular, the UFA–DFA tradeoff is asymptotically equivalent to $\widetilde{g}$, and the growth rate of the latter is determined as $2^{\Theta(\sqrt[3]{n \ln^2 n})}$. A close lower bound on the tradeoff between NFAs and UFAs is established using the matrix methods of Schmidt [19], and the tradeoff is found to be of the order of the original Landau's function, that is, $e^{(1+o(1))\sqrt{n \ln n}}$. Finally, the complexity of operations on UFAs is approached, and an $n^{2-o(1)}$ lower bound for complementation is established, which shows for the first time that the complement of a UFA sometimes requires additional states. The complexity of Kleene star is determined precisely as $(n-1)^2+1$.

2 Simplifying unary 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 nonempty set of states; $Q_0 \subseteq Q$ is the set of initial states; $\delta : Q \times \Sigma \to 2^Q$ 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 shall be extended to $\delta : Q \times \Sigma^* \to 2^Q$ by $\delta(q, \varepsilon) = \{q\}$ and $\delta(q, aw) = \bigcup_{q' \in \delta(q,a)} \delta(q', w)$.

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 $|\delta(q, a)| \leqslant 1$ for all q and a.

An NFA is *unambiguous* if for every $w \in L(A)$ the corresponding sequence of states $r_0, \ldots, r_{|w|}$ in the definition of acceptance is unique. An unambiguous NFA is called an *unambiguous finite automaton* (UFA).

The first lower bound argument for UFAs was given by Schmidt [19, 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 [10]:

Schmidt's Theorem [19, 10]. *Let $L \subseteq \Sigma^*$ be a regular language and let $\{(u_1, v_1), \ldots, (u_n, v_n)\}$ with $n \geqslant 1$ 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 $M_{i,j} = 1$ if $u_i v_j \in L$, and $M_{i,j} = 0$ otherwise. Then every UFA recognizing L has at least* $\operatorname{rank} M$ *states.*

A state q is called *useful* if $q \in \delta(q_0, a^k)$ and $\delta(q, a^\ell) \cap F$ for some $k, \ell \geqslant 0$. A state that is not useful is called *useless*. A state q is a *sink state* if $\delta(q, a) = \varnothing$. Note that a sink state that is not accepting is always useless.

Let $\Sigma = \{a\}$ and consider the following transformation of automata. First, the acceptance is done one step earlier; second, an extra transition by a is added to the beginning of the automaton. Clearly, the transformation preserves the language:

Lemma 1. *Let $A = (\{a\}, Q, Q_0, \delta, F)$ be an NFA with sink states $Q_{sink} \subseteq Q$. Then the NFA $B = (\{a\}, (Q \setminus Q_{sink}) \cup \{q_{-1}\}, \{q_{-1}\}, \delta', F')$ with $\delta'(q, a) = \delta(q, a) \setminus Q_{sink}$ for all $q \in Q \setminus Q_{sink}$, $\delta'(q_{-1}, a) = Q_0 \setminus Q_{sink}$ and $F' = \{\, q \mid \delta(q, a) \cap F \neq \varnothing \,\} \cup \{\, q_{-1} \mid$ if $\varepsilon \in L(A) \,\}$ recognizes the same language. Furthermore, if A is unambiguous, then so is B.*

Proof. If order to see that $L(B) = L(A)$, consider any string a^ℓ with $\ell \geqslant 1$.

If $a^\ell \in L(A)$, then there is a state $q \in \delta(Q_0, a^\ell)$ with $q \in F$. Then there exists a state $q' \in \delta(Q_0, a^{\ell-1})$ with $q \in \delta(q', a)$. The latter means that $q' \notin Q_{sink}$, and therefore $q' \in \delta'(Q_0, a^{\ell-1})$ in the automaton B as well. Using the transition from q_{-1}, $q' \in \delta'(q_{-1}, a^\ell)$. In addition, $q' \in F'$ because of q, and therefore $a^\ell \in L(B)$.

Conversely, if $a^\ell \in L(B)$, then there is a state $q' \in \delta'(q_{-1}, a^\ell)$ with $q' \in F'$. Hence, on one hand, there is $q_0 \in Q_0$ with $q' \in \delta'(q_0, a^{\ell-1}) \subseteq \delta(q_0, a^{\ell-1})$, and

on the other hand, there is $q \in \delta(q', a)$ with $q \in F$. This implies $q \in \delta(q_0, a^\ell)$ and hence $a^\ell \in L(A)$.

Now assume A is unambiguous and suppose B is not. Then there exists a string accepted by B in two different ways, that is, there are states $q \neq q'$ and numbers $\ell, m \geqslant 0$, such that $q, q' \in \delta'(q_{-1}, a^\ell)$, $\delta'(q, a^m) \cap F' \neq \varnothing$ and $\delta'(q', a^m) \cap F' \neq \varnothing$. Consider that $\ell \geqslant 1$, and therefore there are states $q_0, q'_0 \in \delta'(q_{-1}, a) \subseteq Q_0$ satisfying the conditions $q \in \delta'(q_0, a^{\ell-1})$ and $q' \in \delta'(q'_0, a^{\ell-1})$. The same transitions are possible in A, that is, $q \in \delta(q_0, a^{\ell-1})$ and $q' \in \delta(q'_0, a^{\ell-1})$. At the same time, by the construction of F', $\delta(q, a^{m+1}) \cap F \neq \varnothing$ and $\delta(q', a^{m+1}) \cap F \neq \varnothing$. This gives two different accepting computations of A on $a^{\ell+m}$, which is impossible by assumption. □

Note that as long as A has at least one sink state, the number of states in B does not exceed the number of states in A. The purpose of this transformation is to simplify the structure of sink states.

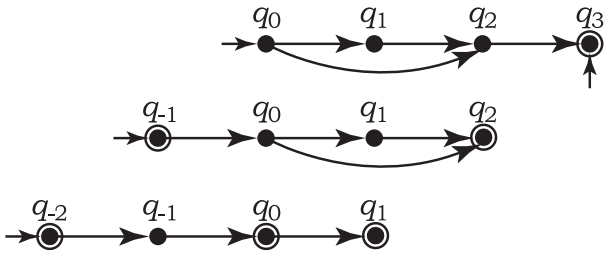

Figure 2: Back-step transformation of unary NFAs.

If the language is finite, the transformation in Lemma 1 can be applied until the NFA is straightened into a chain ending with one sink state, as shown in the example in Figure 2. This gives a proof of the following known result:

Proposition 1 (Mandl [12])**.** *For every NFA recognizing a finite language over $\{a\}$ there exists a partial DFA with the same number of states recognizing the same language.*

This partial DFA will have a unique sink state. In the case of an infinite language, the above transformation leads to a complete elimination of sink states:

Lemma 2. *For every n-state NFA recognizing an infinite unary language there exists an NFA without sink states that has at most n states and recognizes the same language. The construction is effective and preserves unambiguity.*

Proof. Let us say that a state q is *terminal*, if $\delta(q, a^\ell) = \varnothing$ for some ℓ. The statement of the lemma is proved by induction on the number of terminal

states in the NFA. The basis, 0 terminal states, holds because the given NFA already satisfies the condition.

For the induction step, let $A = (\{a\}, Q, Q_0, \delta, F)$ be the given NFA, and let $Q_{sink} \subseteq Q$ be its nonempty set of sink states. Construct the NFA $B = (\{a\}, (Q \setminus Q_{sink}) \cup \{q_{-1}\}, \{q_{-1}\}, \delta', F')$ with $L(B) = L(A)$, as defined in Lemma 1. It is now claimed that every terminal state of B is a terminal state of A.

Assume that $q \in Q$ is not a terminal state of A. Then there exists an infinite sequence of states $q_1, q_2, \ldots$, with $q_1 = q$ and $q_{i+1} \in \delta(q_i, a)$. This makes $\delta(q_i, a^\ell) \neq \varnothing$ for all i and ℓ, and in particular $q_i \notin Q_{sink}$ for all i. Then $q_{i+1} \in \delta'(q_i, a)$, that is, the same sequence of states is preserved in B. This shows that q is not a terminal state of B, which proves the claim.

Accordingly, B has no new terminal states, which would not be terminal in A. Note that since $L(B)$ is infinite, q_{-1} is not a terminal state. At the same time, B has cast away A's terminal states from Q_{sink}, so B has fewer terminal states than A. Then, by the induction hypothesis, B can be transformed to the form without sink states, which is the desired form of A. □

Once sink states are eliminated, and thus all computations of a UFA must eventually end in cycles, it turns out the cyclic part of the UFA has to be deterministic, which is established in the following lemma.

Lemma 3. *Let $A = (\{a\}, Q, Q_0, \delta, F)$ be a UFA recognizing an infinite language. Assume that there are no sink states and no useless states in A. Let q be a cyclic state, that is, with $q \in \delta(q, a^p)$ for some $p \geqslant 0$. Then the outgoing transition from q is unique.*

Proof. Let q be reachable from one of the initial states by a string a^ℓ. One transition from q begins the cycle from q to q by a^p. Suppose there is another transition from q. The path started by this transition eventually, after reading a string a^m, reaches a cyclic state q' with $q' \in \delta(q', a^{p'})$ (or a sink state, which is impossible by assumption). Since the latter state is not useless, it should be possible to reach an accepting state from q' by some string a^n.

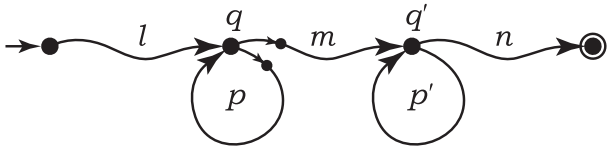

Figure 3: A cyclic state in an NFA.

The above transitions are illustrated in Figure 3. Using these transitions, the automaton can accept all strings in $a^\ell(a^p)^*a^m(a^{p'})^*a^n$, that is, all strings of length $\ell + ip + m + i'p' + n$ by using i iterations over the first cycle and i' iterations over the second cycle. Setting $i = 0$ and $i' = p$, or $i = p'$ and $i' = 0$, one can obtain two distinct accepting computations on the string $a^{\ell+m+n+pp'}$, which contradicts the assumption that the automaton in unambiguous. □

3 Chrobak normal form of unambiguous automata

The study of NFAs over a unary alphabet is founded upon the following normal form:

Definition 1 (Chrobak [3]). *An NFA over $\{a\}$ is said to be in Chrobak normal form if its set of states is $\{q_0, \ldots, q_{\ell-1}\} \cup \bigcup_{i=1}^{k} R_i$, with $\ell \geqslant 0$, $k \geqslant 0$, $R_i = \{r_{i,0}, \ldots, r_{i,p_i-1}\}$ and $1 \leqslant p_1 < p_2 < \ldots < p_k$, the unique initial state is q_0 if $\ell \geqslant 1$ or there is a set of initial states $\{r_{1,0}, \ldots, r_{k,0}\}$ if $\ell = 0$, and the transitions are:*

$$\begin{aligned} \delta(q_i, a) &= \{q_{i+1}\} \quad (0 \leqslant i \leqslant \ell - 2) \\ \delta(q_{\ell-1}, a) &= \{r_{1,0}, r_{2,0}, \ldots, r_{k,0}\} \quad (\text{if } \ell \geqslant 1) \\ \delta(r_{i,j}, a) &= \{r_{i,j+1 \bmod p_i}\} \quad (1 \leqslant i \leqslant k,\ 0 \leqslant j \leqslant p_i - 1), \end{aligned}$$

The set of accepting states may be arbitrary.

It is known from Chrobak [3] that every NFA with n states can be transformed to an equivalent NFA in this normal form, with $\ell = O(n^2)$ and $\sum_{i=1}^{k} p_n \leqslant n$. The growth in the number of states is thus at most quadratic. In contrast, for UFAs such a transformation can be done without increasing the number of states.

Theorem 1. *For every UFA recognizing an infinite language over $\{a\}$ 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.*

Proof. Let $A = (\{a\}, Q, Q_0, \delta, F)$ be a unary UFA. By Lemma 2, there is no loss of generality in the assumption that there are no sink states in the automaton. It can also be assumed that A contains no useless states.

Let $\widehat{Q} \subseteq Q$ be the set of cyclic states of A. By Lemma 3, every state q in $\widehat{Q}$ has a unique outgoing transition, and the graph of transition A from the states in $\widehat{Q}$ is a collection of one or more disjoint simple cycles. Let k be the number of simple cycles in $\widehat{Q}$.

For every $n \geqslant 1$, define $Q_n = \delta(Q_0, a^n) \subseteq Q$. Let $\ell \geqslant 0$ be the least number with $Q_\ell \subseteq \widehat{Q}$: this eventually happens, since all paths lead to simple cycles. The definition of the sets Q_n is illustrated in Figure 4, left, where $k = 2$ and $\ell = 3$.

The idea of the construction is to replace each set Q_n, for $0 \leqslant n \leqslant \ell - 1$, with a new state q_n, and to leave the states in $\widehat{Q}$ as they are. Let p_i be the length of each ith cycle in A, and denote the states in it by $R_i = \{r_{i,0}, \ldots, r_{i,p_i-1}\}$ for all $1 \leqslant i \leqslant k$, and with transitions $\delta(r_{i,j}, a) = r_{i,j+1}$, where the addition is modulo p_i. The new NFA B has the set of states

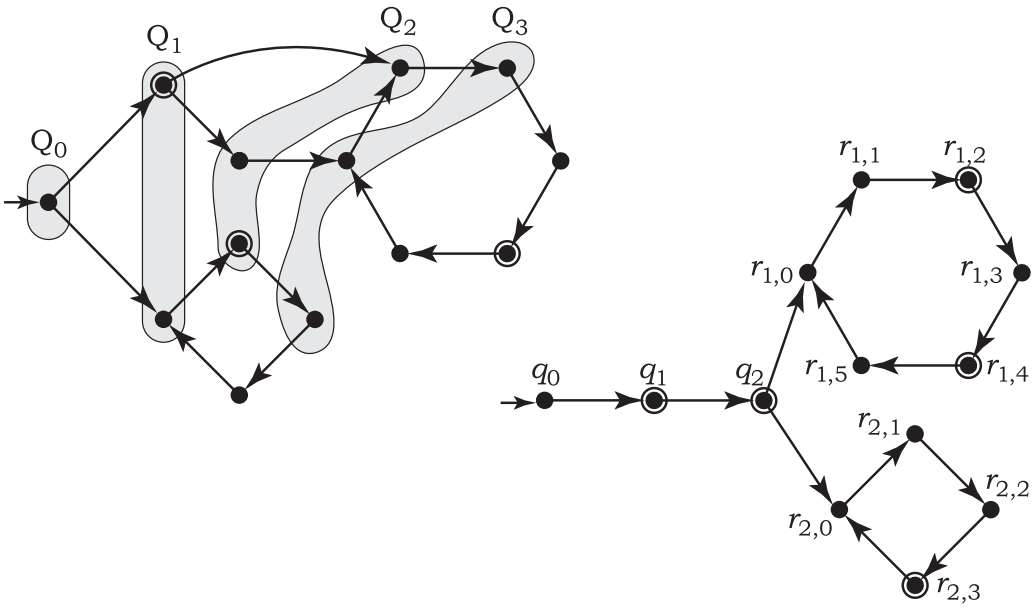

Figure 4: Transforming a UFA to Chrobak normal form.

$Q' = \{q_0, \ldots, q_{\ell-1}\} \cup \bigcup_{i=1}^{k} R_i$. If $\ell \geqslant 1$, its unique initial state is q_0, and in case of $\ell = 0$, the set of initial states is $\{\, r_{i,0} \mid 1 \leqslant i \leqslant k \,\}$. The transitions of B are defined by $\delta'(q_n, a) = q_{n+1}$ for $0 \leqslant n \leqslant \ell-2$, $\delta'(q_{\ell-1}, a) = \{r_{1,0}, \ldots, r_{k,0}\}$ and $\delta'(r_{i,j}, a) = r_{i,j+1} = \delta(r_{i,j}, a)$. The set of accepting states is defined as follows:

$$F' = \{\, q_n \mid 0 \leqslant n \leqslant \ell - 1,\ Q_n \cap F \neq \varnothing \,\} \cup \{\, r_{i,f-s} \mid r_{i,s} \in Q_\ell,\ r_{i,f} \in F \,\},$$

where the subtraction is modulo p_i. The automaton has at most n states, because

$$|Q'| = \ell + \left|\bigcup_{i=1}^{k} R_i\right| = \ell + |\widehat{Q}| \leqslant \left|\bigcup_{n=0}^{\ell-1} Q_n \setminus \widehat{Q}\right| + |\widehat{Q}| = |Q|.$$

Claim 1. *The string $a^{\ell+n}$ in accepted by A in R_i if and only if it is accepted by B in R_i.*

Indeed, the acceptance of $a^{\ell+n}$ by A in R_i is equivalent to the existence of states $r_{i,s} \in Q_\ell$ and $r_{i,f} \in F$ with $f - s = n \pmod{p_i}$. By the definition of F', this holds if and only if $r_{i,n \bmod p_i} \in F'$, which holds if and only if B accepts $a^{\ell+n}$ in R_i. This proves the claim.

One can infer from this claim that $L(B) = L(A)$ as follows. According to the definition of F', every string shorter than a^ℓ is accepted by B if and only if it is accepted by A. A string $a^{\ell+n}$ is accepted by both automata if the ith component mentioned in the claim exists, and is rejected by both automata otherwise.

To show that B is unambiguous, consider that strings up to $a^{\ell-1}$ have a unique computation, and if a string $a^{\ell+n}$ with $n \geqslant 0$ is accepted in two different states, these states must belong to different cycles. Then, by the claim, A should accept $a^{\ell+n}$ in both cycles, which contradicts the assumption that it is unambiguous. □

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

Theorem 2. *An NFA $(\{a\}, Q, q_0, \delta, F)$ in Chrobak normal form recognizing an infinite language over $\{a\}$ 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 proof uses Chinese Remainder Theorem in the following formulation:

Chinese Remainder Theorem. *Let $p, p' \geqslant 1$ and $i, i' \geqslant 0$ be any numbers with $i = i' \pmod{\gcd(p, p')}$. Then there exists a number $n \geqslant 0$ with $n = i \pmod{p}$ and $n = i' \pmod{p'}$.*

Proof of Theorem 2. ⟸ Assume that the conditions on accepting states hold and suppose that the automaton is ambiguous. Then there is a string $a^{\ell+n}$ with $n \geqslant 0$ accepted in cycles R_i and R_j; more precisely, in states $r_{i,f}$ and $r_{j,f'}$. Accordingly, $n = f \pmod{p_i}$ and $n = f' \pmod{p_j}$, and therefore $f = n = f' \pmod{\gcd(p_i, p_j)}$, which contradicts the condition.

⟹ Let the automaton be unambiguous and suppose there exist two states $r_{i,f}, r_{j,f'} \in F$ with $i \neq j$ and $f = f' \pmod{\gcd(p_i, p_j)}$. 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 $n \geqslant 0$ with $n = f \pmod{p_i}$ and $n = f' \pmod{p_j}$. 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. □

Theorem 2, in particular, implies that the lengths of the cycles cannot be primes (unless there is a unique cycle), and that $\gcd(p_i, p_j) \geqslant 2$ 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.

4 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 unary NFA has been established by Lyubich [11]. It is asymptotically equivalent to the maximum order of a permutation on n elements:

$$g(n) = \max\{\, \operatorname{lcm}(p_1, \ldots, p_k) \mid k \geqslant 1,\ p_1 + \ldots + p_k \leqslant n \,\}.$$

This function is known as *Landau's function*, as its $e^{\sqrt{n \ln n}(1+o(1))}$ asymptotics was determined by Landau [8], see also Miller [16] for a more accessible argument.

Twenty years after Lyubich, an asymptotically matching lower bound on the unary NFA to DFA tradeoff was obtained by Chrobak [3], who also gave a new, combinatorial proof of Lyubich's upper bound. These results can be stated as follows:

Proposition 2 (Lyubich [11], Chrobak [3])**.** *For every n-state unary NFA there exists a DFA with at most $g(n) + n^2$ states recognizing the same language. Conversely, for every n there is a language recognized by an n-state NFA, such that every equivalent DFA requires $g(n)$ states.*

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

This analysis of NFAs can be extended to UFAs, if the additional constraints on their Chrobak normal form given in Theorem 2 are embedded into the definition of Landau's function. This leads to the following variant of this function:

$$\begin{aligned}\widetilde{g}(n) = \max \big\{ \operatorname{lcm}(p_1, \ldots, p_k) \;\big|\; & k \geqslant 1,\ p_1 + \ldots + p_k \leqslant n, \\ & \exists f_1, \ldots, f_k \text{ with } f_i \in \{0, \ldots, p_i - 1\} : \\ & \forall i, j\ (i \neq j)\ f_i \neq f_j \pmod{\gcd(p_i, p_j)} \big\}\end{aligned}$$

For n up to 9 the value of $\widetilde{g}(n)$ is n. The next value is $\widetilde{g}(10) = 12$, given by $k = 2$, $p_1 = 4$, $p_2 = 6$, $f_1 = 0$ and $f_2 = 1$ with $0 \neq 1 \pmod{\gcd(4, 6)}$. This function can be asymptotically 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 can be done as follows:

Theorem 3. *For every $n \geqslant 1$, the following number of states is sufficient and in the worst case necessary for a DFA to recognize a language recognized by an n-state UFA with multiple initial states:*

$$f_{\mathit{UFA\text{–}DFA}}(n) = \begin{cases} n + 1, & \text{if } n \leqslant 9 \\ \max\limits_{0 \leqslant \ell < n} \widetilde{g}(n - \ell) + \ell, & \text{if } n \geqslant 10 \end{cases}$$

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

$$f_{\mathit{UFA_1\text{–}DFA}}(n) = \begin{cases} n + 1, & \text{if } n \leqslant 10 \\ \max\limits_{1 \leqslant \ell < n} \widetilde{g}(n - \ell) + \ell, & \text{if } n \geqslant 11 \end{cases}$$

For $n \leqslant 9$, 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 nontrivial values of $\widetilde{g}$, the following witness languages can be represented:

Lemma 4. *Let $k \geqslant 2$, $\ell \geqslant 0$, $p_1, \ldots, p_k \geqslant 2$ and $f_1, \ldots, f_k \geqslant 0$ with $0 \leqslant f_i < p_i$ be any numbers, such that (a) $f_i \neq f_j \pmod{\gcd(p_i, p_j)}$ for all $1 \leqslant i < j \leqslant k$, (b) $\operatorname{lcm}(p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_k)$ is not divisible by p_i for any $1 \leqslant i \leqslant k$, 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 $\ell + p_1 + \ldots + p_k$ states, while every DFA for this language requires $\ell + \operatorname{lcm}(p_1, \ldots, p_m)$ states.

Proof. The construction of a UFA in Chrobak normal form with a tail of length ℓ and with cycles $p_1, \ldots, p_k$, each with a unique accepting state at offset f_i, is entirely obvious. As $f_i \neq f_j \pmod{\gcd(p_i, p_j)}$ by assumption, the condition of Theorem 2 is satisfied.

Let $p = \operatorname{lcm}(p_1, \ldots, p_m)$ and consider any DFA recognizing L. It is sufficient to prove that for any two distinct states $q = \delta(q_0, a^m)$ and $q' = \delta(q_0, a^{m'})$ with $0 \leqslant m < m' < \ell + p$ there exists a string accepted from one of these states and not accepted from the other. If $m' - m = 0 \pmod{p}$, 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 $m' - m \neq 0 \pmod{p}$. 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 $n \in \{0, \ldots, p-1\}$ equal to $f_1 + m' - m$ modulo p_1, such that the string $a^{\ell+p+n-(m'-m)}$ is in L_1, 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 equal to $n_1 = f_1 + m' - m$ modulo p_1, the string $a^{\ell+n}$ is in L. Let

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

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

Claim 2. *Let $2 \leqslant i \leqslant k$ and let n_{i-1} be a number with $0 \leqslant n_{i-1} < \operatorname{lcm}(p_1, \ldots, p_{i-1})$. Assume that $a^{\ell+n} \in L_i \cup L_{i+1} \cup \ldots \cup L_k$ for all $n \geqslant 0$ equal to n_{i-1} modulo $\operatorname{lcm}(p_1, \ldots, p_{i-1})$. Then there exists a number n_i with*

$0 \leqslant n_i < \operatorname{lcm}(p_1, \ldots, p_{i-1}, p_i)$, *such that* $a^{\ell+n} \in L_{i+1} \cup \ldots \cup L_k$ *for every number* $n \geqslant 0$ *equal to* n_i *modulo* $\operatorname{lcm}(p_1, \ldots, p_{i-1}, p_i)$.

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

It remains to prove the claim. Consider two numbers, n_{i-1} and $n_{i-1} + \operatorname{lcm}(p_1, \ldots, p_{i-1})$. It is known that $\operatorname{lcm}(p_1, \ldots, p_{i-1})$ is nonzero modulo p_i (otherwise p_i would divide $\operatorname{lcm}(p_1, \ldots, p_{i-1})$, contradicting assumption (b)). Then $n_{i-1} \neq n_{i-1} + \operatorname{lcm}(p_1, \ldots, p_{i-1}) \pmod{p_i}$, and therefore at least one of these numbers must be different from f_i modulo p_i; denote this number by n_i.

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

The matching upper bound is implied by the following lemma:

Lemma 5. *For every* n*-state UFA in Chrobak normal form with a tail of length* $\ell \geqslant 0$ *there exists a DFA with at most* $\ell + \widetilde{g}(n - \ell)$ *states recognizing the same language.*

Proof. Let $p_1, \ldots, p_k$ be the lengths of the cycles in this UFA. Then it is well-known that there is an equivalent DFA with $\operatorname{lcm}(p_1, \ldots, p_k) + \ell$ states [3, Thm. 4.4].

Consider one accepting state from each cycle: $r_{1,f_1}, r_{2,f_2}, \ldots, r_{k,f_k} \in F$. By Theorem 2, $f_i \neq f_j \pmod{\gcd(p_i, p_j)}$ for all $i \neq j$. Then these numbers satisfy the definition of $\widetilde{g}$, and accordingly $\operatorname{lcm}(p_1, \ldots, p_k) \leqslant \widetilde{g}(n - \ell)$, which shows that the above DFA has at most $\widetilde{g}(n - \ell) + \ell$ states. □

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

Proof of Theorem 3. Note that $\widetilde{g}(10) = \operatorname{lcm}(4, 6) = 12$, and therefore, for every $n \geqslant 11$, $\widetilde{g}(n - \ell) + \ell > n + 1$ for $\ell = n - 10$. It is also easy to verify that the smallest two numbers with a common divisor and with a least common multiple larger than either of the numbers are 4 and 6, and accordingly $\widetilde{g}(n) = n$ for $n < 10$. Then the function stated in the theorem can be equivalently expressed as follows:

$$f_{\text{UFA–DFA}}(n) = \max\big(n + 1, \max_{0 \leqslant \ell < n} \widetilde{g}(n - \ell) + \ell\big).$$

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 Theorem 1, it can be assumed that the UFA is in Chrobak normal form; let ℓ be the length of the tail. Then a DFA with $\widetilde{g}(n-\ell)+\ell$ states recognizing the same language exists due to Lemma 5. In all three cases the number of states is at most $f_{\text{UFA–DFA}}(n)$.

To prove the ***lower bound***, fix $n \geqslant 1$. 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 hence $f_{\text{UFA–DFA}}(n) \geqslant n+1$. It remains to prove that $f_{\text{UFA–DFA}}(n) \geqslant \widetilde{g}(n-\ell)+\ell$ for every $\ell \in \{1,\ldots,n-1\}$.

Choose ℓ so that the number $\widetilde{g}(n-\ell)+\ell$ is the greatest possible, and consider the number $\widetilde{g}(n-\ell)$, which is given by $\operatorname{lcm}(p_1,\ldots,p_k)$ for some $k \geqslant 1$, $p_1,\ldots,p_k \geqslant 2$ and $f_1,\ldots,f_k \geqslant 0$ with $p_1+\ldots+p_k \leqslant n-\ell$ and $f_i \neq f_j \pmod{\gcd(p_i,p_j)}$ for all $i \neq j$. Furthermore, the number $\operatorname{lcm}(p_1,\ldots,p_k)$ 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, $\operatorname{lcm}(p_1,\ldots,p_{i-1},p_{i+1},\ldots,p_k)$ is not divisible by p_i. Indeed, if $\operatorname{lcm}(p_1,\ldots,p_{i-1},p_{i+1},\ldots,p_k)$ is a multiple of p_i, then $\operatorname{lcm}(p_1,\ldots,p_{i-1},p_{i+1},\ldots,p_k) = \operatorname{lcm}(p_1,\ldots,p_k)$, and accordingly $\widetilde{g}(p_1+\ldots+p_k) = \widetilde{g}(p_1+\ldots+p_k-p_i)$, which implies that $\widetilde{g}(n-\ell-p_i)+\ell+p_i > \widetilde{g}(n-\ell)+\ell$. Then $\ell' = \ell+p_i$ leads to a greater value $\widetilde{g}(n-\ell')+\ell'$, which contradicts the choice of ℓ.

The next claim is that the offsets $f_1,\ldots,f_k$ 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 $f'_i \neq f'_j \pmod{\gcd(p_i,p_j)}$ is preserved, because $f'_i-f'_j = f_i-f_j \pmod{\gcd(p_i,p_j)}$.

It has thus been demonstrated that all conditions of Lemma 4 are satisfied, and hence there exists a language representable by an n-state UFA, for which every DFA must have $\operatorname{lcm}(p_1,\ldots,p_k)+\ell = \widetilde{g}(n-\ell)+\ell$ states. □

The values of $\widetilde{g}(n)$ for small values of n, calculated by an exhaustive search, are given in Table 1, along with the computed lengths of cycles $p_1,\ldots,p_k$. Furthermore, the table gives the precise number of states in a DFA needed to simulate an n-state UFA, as well as witness languages on which this bound is reached. These languages and their state complexity are determined on the basis of the values of $\widetilde{g}(n)$ according to Lemma 4 (which does not involve any extensive calculations). The next Table 2 gives similar results for UFAs with a unique initial state.

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

Table 1: Values of $\widetilde{g}(n)$; UFA–DFA tradeoff with witness languages.

n: UFA_1	$\widetilde{g}(n)$	$f(n)$: DFA	witness language
1	1	2	$\{\varepsilon\}$
2	2	3	$\{a\}$
3	3	4	$\{a^2\}$
4	4	5	$\{a^3\}$
5	5	6	$\{a^4\}$
6	6	7	$\{a^5\}$
7	7	8	$\{a^6\}$
8	8	9	$\{a^7\}$
9	9	10	$\{a^8\}$
10	12 =lcm(4,6)	11	$\{a^9\}$
11	12 =lcm(4,6)	13	$a^4(a^4)^* \cup a^5(a^6)^*$
12	12 =lcm(4,6)	14	$a^5(a^4)^* \cup a^6(a^6)^*$
13	13	15	$a^6(a^4)^* \cup a^7(a^6)^*$
14	24 =lcm(6,8)	16	$a^7(a^4)^* \cup a^8(a^6)^*$
15	24 =lcm(6,8)	25	$a^6(a^6)^* \cup a^7(a^8)^*$
16	30 =lcm(6,10)	26	$a^7(a^6)^* \cup a^8(a^8)^*$
17	30 =lcm(6,10)	31	$a^6(a^6)^* \cup a^7(a^{10})^*$
18	40 =lcm(8,10)	32	$a^7(a^6)^* \cup a^8(a^{10})^*$
19	40 =lcm(8,10)	41	$a^8(a^8)^* \cup a^9(a^{10})^*$
20	42 =lcm(6,14)	42	$a^9(a^8)^* \cup a^{10}(a^{10})^*$
21	42 =lcm(6,14)	43	$a^6(a^6)^* \cup a^7(a^{14})^*$
22	60 =lcm(10,12)	44	$a^7(a^6)^* \cup a^8(a^{14})^*$
23	60 =lcm(10,12)	61	$a^{10}(a^{10})^* \cup a^{11}(a^{12})^*$
24	70 =lcm(10,14)	62	$a^{11}(a^{10})^* \cup a^{12}(a^{12})^*$
25	70 =lcm(10,14)	71	$a^{10}(a^{10})^* \cup a^{11}(a^{14})^*$
26	84 =lcm(12,14)	72	$a^{11}(a^{10})^* \cup a^{12}(a^{14})^*$
27	84 =lcm(12,14)	85	$a^{12}(a^{12})^* \cup a^{13}(a^{14})^*$
28	90 =lcm(10,18)	86	$a^{13}(a^{12})^* \cup a^{14}(a^{14})^*$
29	90 =lcm(10,18)	91	$a^{10}(a^{10})^* \cup a^{11}(a^{18})^*$
30	120 =lcm(8,10,12)	92	$a^{11}(a^{10})^* \cup a^{12}(a^{18})^*$
31	120 =lcm(8,10,12)	121	$a^8(a^8)^* \cup a^9(a^{10})^* \cup a^{10}(a^{12})^*$
32	126 =lcm(14,18)	122	$a^9(a^8)^* \cup a^{10}(a^{10})^* \cup a^{11}(a^{12})^*$
33	126 =lcm(14,18)	127	$a^{14}(a^{14})^* \cup a^{15}(a^{18})^*$
34	168 =lcm(8,12,14)	128	$a^{15}(a^{14})^* \cup a^{16}(a^{18})^*$
35	168 =lcm(8,12,14)	169	$a^8(a^8)^* \cup a^{10}(a^{12})^* \cup a^9(a^{14})^*$
36	180 =lcm(9,12,15)	170	$a^9(a^8)^* \cup a^{11}(a^{12})^* \cup a^{10}(a^{14})^*$
37	180 =lcm(9,12,15)	181	$a^9(a^9)^* \cup a^{10}(a^{12})^* \cup a^{11}(a^{15})^*$
38	240 =lcm(10,12,16)	182	$a^{10}(a^9)^* \cup a^{11}(a^{12})^* \cup a^{12}(a^{15})^*$
39	240 =lcm(10,12,16)	241	$a^{10}(a^{10})^* \cup a^9(a^{12})^* \cup a^{11}(a^{16})^*$
40	240 =lcm(10,12,16)	242	$a^{11}(a^{10})^* \cup a^{10}(a^{12})^* \cup a^{12}(a^{16})^*$
41	240 =lcm(10,12,16)	243	$a^{12}(a^{10})^* \cup a^{11}(a^{12})^* \cup a^{13}(a^{16})^*$
42	336 =lcm(12,14,16)	244	$a^{13}(a^{10})^* \cup a^{12}(a^{12})^* \cup a^{14}(a^{16})^*$
43	336 =lcm(12,14,16)	337	$a^{12}(a^{12})^* \cup a^{13}(a^{14})^* \cup a^{14}(a^{16})^*$
44	336 =lcm(12,14,16)	338	$a^{13}(a^{12})^* \cup a^{14}(a^{14})^* \cup a^{15}(a^{16})^*$
45	336 =lcm(12,14,16)	339	$a^{14}(a^{12})^* \cup a^{15}(a^{14})^* \cup a^{16}(a^{16})^*$
46	420 =lcm(12,14,20)	340	$a^{15}(a^{12})^* \cup a^{16}(a^{14})^* \cup a^{17}(a^{16})^*$
47	420 =lcm(12,14,20)	421	$a^{12}(a^{12})^* \cup a^{13}(a^{14})^* \cup a^{14}(a^{20})^*$
48	420 =lcm(12,14,20)	422	$a^{13}(a^{12})^* \cup a^{14}(a^{14})^* \cup a^{15}(a^{20})^*$
49	420 =lcm(12,14,20)	423	$a^{14}(a^{12})^* \cup a^{15}(a^{14})^* \cup a^{16}(a^{20})^*$
50	560 =lcm(14,16,20)	424	$a^{15}(a^{12})^* \cup a^{16}(a^{14})^* \cup a^{17}(a^{20})^*$

Table 2: UFA_1–DFA tradeoff with witness languages.

5 Estimations of $\widetilde{g}$

The function $\widetilde{g}$ 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 unary alphabet.

So what is the asymptotic behaviour of the function $\widetilde{g}$? The first step towards determining its growth rate is estimating the maximum number of cycles k for a given sum of cycle lengths.

Lemma 6. *Let $k \geqslant 1$ and let $\pi_1, \ldots, \pi_k \geqslant 2$ be any integers, for which (a) there exist $f_1, \ldots, f_k \in \mathbb{N}$ with $f_i \neq f_j \pmod{\gcd(\pi_i, \pi_j)}$ for all $i \neq j$, and (b) $\operatorname{lcm}(\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_k)$ is not divisible by p_i for any $1 \leqslant i \leqslant k$. Then $\pi_1 + \ldots + \pi_k > \frac{4}{9}k^3 \ln k - \frac{8}{27}k^3\sqrt{\ln k}$.*

As in Lemma 4, the condition 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 $r_i = \frac{\operatorname{lcm}(\pi_1,\ldots,\pi_k)}{\operatorname{lcm}(\pi_1,\ldots,\pi_{i-1},\pi_{i+1},\ldots,\pi_k)}$ and let $\pi_i = r_i s_i$. Then the numbers $r_1, \ldots, r_k$ 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:

$$\min_{\substack{r_1,\ldots,r_k \geqslant 2 \\ \text{relatively prime}}} \quad \min_{\substack{s_1,\ldots,s_k \in \mathbb{N} \\ \exists f_1,\ldots,f_k \in \mathbb{N} \\ f_i \neq f_j \pmod{\gcd(s_i,s_j)}}} \quad \sum_{i=1}^{k} r_i s_i > \tfrac{4}{9}k^3 \ln k - \tfrac{8}{27}k^3\sqrt{\ln k}.$$

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 condition on $s_1, \ldots, s_k$ involving the numbers $f_1, \ldots, f_k$ with the following simpler consequence of this condition:

Claim 6.1. $\frac{1}{s_1} + \ldots + \frac{1}{s_k} \leqslant 1$.

Proof. Let $s = \operatorname{lcm}(s_1, \ldots, s_k)$. An ith cycle is said to *cover* a number $n \in \{0, \ldots, s-1\}$, if $f_i = n \pmod{s_i}$. 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 $\sum_{i=1}^{k} \frac{s}{s_i} > s$, and accordingly, some number n must be covered by two different cycles, that is, $f_i = n \pmod{s_i}$ and $f_j = n \pmod{s_j}$. Therefore, $f_i = n = f_j \pmod{\gcd(s_i, s_j)}$, which contradicts the assumption. □

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 6.1. For example, for $k = 3$ and $r_1 = 2$, $r_1 = 3$, $r_1 = 7$, 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 6.1, 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 6.2. *Let $a_1, \ldots, a_m > 0$ be any positive real numbers. Then*

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

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

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

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

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

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

Its partial derivative by x_i is

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

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:

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

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

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

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

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

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

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

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 $\sum_{i=1}^{k}\sum_{j=1}^{k}\sqrt{a_i a_j} = (\sqrt{a_1}+\ldots+\sqrt{a_k})^2$, which proves the claim. □

Therefore, a lower bound on the sum $\sum_{i=1}^{k} r_i s_i$ is $(\sqrt{r_1}+\ldots+\sqrt{r_k})^2$, 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 $r_1,\ldots,r_k \geqslant 2$. In fact, the minimum is achieved by taking the first k primes.

Claim 6.3. *Let $2 \leqslant r_1 < \ldots < r_k$ be any pairwise relatively prime natural numbers. Then $p_i \leqslant r_i$, where p_i is the ith 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 $r'_j \leqslant p_{i-1}$. Since the primes $r'_1,\ldots,r'_{i-1}$ must be pairwise distinct, it follows that $\{r'_1,\ldots,r'_{i-1}\} = \{p_1,\ldots,p_{i-1}\}$, and thus every prime factor of r_i must belong to this set, which contradicts the assumption that $r_1,\ldots,r_k$ are relatively prime. □

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

$$(\sqrt{r_1}+\ldots+\sqrt{r_k})^2 \geqslant (\sqrt{p_1}+\ldots+\sqrt{p_k})^2.$$

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

Proposition 3. *$p_n > n\ln n$ for all $n \geqslant 1$.*

It remains to calculate the resulting sum:

Claim 6.4. *$\sum_{n=1}^{k}\sqrt{n\ln n} > \frac{2}{3}k\sqrt{k\ln k} - \frac{2}{9}k\sqrt{k}$ for all $k \geqslant 1$.*

Proof. For $k \leqslant 8$ the inequality can be verified by direct calculations, so assume $k \geqslant 9$. The idea is to approximate the sum with the integral $\int_1^k \sqrt{x\ln x}\,dx$. Integrating by parts,

$$\int \sqrt{x\ln x}\,dx = x\sqrt{x\ln x} - \int x\,d\sqrt{x\ln x} = x\sqrt{x\ln x} - \int x\frac{\ln x+1}{2\sqrt{x\ln x}}\,dx =$$
$$= x\sqrt{x\ln x} - \tfrac{1}{2}\int\sqrt{x\ln x}\,dx - \tfrac{1}{2}\int\sqrt{\frac{x}{\ln x}}\,dx,$$

and solving the resulting equation gives

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

Then, using the facts that $f(x) = \sqrt{x \ln x}$ is increasing, and that $\sqrt{\frac{x}{\ln x}} \leqslant \sqrt{x}$ for all $x \geqslant e$,

$$\sum_{n=1}^{k} \sqrt{n \ln n} > \int_1^k \sqrt{x \ln x}\, dx = \tfrac{2}{3} k \sqrt{k \ln k} - \tfrac{1}{3} \int_9^k \sqrt{\frac{x}{\ln x}}\, dx - \tfrac{1}{3} \int_1^9 \sqrt{\frac{x}{\ln x}}\, dx >$$
$$> \tfrac{2}{3} k \sqrt{k \ln k} - \tfrac{1}{3} \int_9^k \sqrt{x}\, dx - \tfrac{1}{3} \int_1^9 \sqrt{\frac{x}{\ln x}}\, dx =$$
$$= \tfrac{2}{3} k \sqrt{k \ln k} - \tfrac{2}{9} k \sqrt{k} + \tfrac{2}{9} 9 \sqrt{9} - \tfrac{1}{3} \int_1^9 \sqrt{\frac{x}{\ln x}}\, dx.$$

Approximating the latter integral numerically shows that $\frac{1}{3} \int_1^9 \sqrt{\frac{x}{\ln x}}\, dx < 6 = \frac{2}{9} 9 \sqrt{9}$, which completes the proof. □

With all these auxiliary results established, Lemma 6 is proved by the following chain of inequalities.

Proof of Lemma 6.

$$\min_{\substack{r_1,\ldots,r_k \geqslant 2 \\ \text{relatively prime}}} \min_{\substack{s_1,\ldots,s_k \in \mathbb{N} \\ \exists f_1,\ldots,f_k \in \mathbb{N} \\ f_i \neq f_j \pmod{\gcd(s_i,s_j)}}} \sum_{i=1}^{k} r_i s_i \geqslant \min_{\substack{r_1,\ldots,r_k \geqslant 2 \\ \text{relatively prime}}} \min_{\substack{s_1,\ldots,s_k \in \mathbb{N} \\ \frac{1}{s_1}+\ldots+\frac{1}{s_k} \leqslant 1}} \sum_{i=1}^{k} r_i s_i \geqslant$$

$$\geqslant \min_{\substack{r_1,\ldots,r_k \geqslant 2 \\ \text{relatively prime}}} \min_{\substack{x_1,\ldots,x_k \in \mathbb{R}_+ \\ \frac{1}{x_1}+\ldots+\frac{1}{x_k} \leqslant 1}} \sum_{i=1}^{k} r_i x_i = \min_{\substack{r_1,\ldots,r_k \geqslant 2 \\ \text{relatively prime}}} \left(\sqrt{r_1} + \ldots + \sqrt{r_k}\right)^2 =$$

$$= \left(\sqrt{p_1} + \ldots + \sqrt{p_k}\right)^2 > \left(\sum_{i=1}^{k} \sqrt{i \ln i}\right)^2 > \left(\tfrac{2}{3} k \sqrt{k \ln k} - \tfrac{2}{9} k \sqrt{k}\right)^2 >$$
$$> \tfrac{4}{9} k^3 \ln k - \tfrac{8}{27} k^3 \sqrt{\ln k}.$$

□

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

Lemma 7. *Under the assumptions of Lemma 6,* $k < \frac{3}{\sqrt[3]{4}} \sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}$, *where* $n = \pi_1 + \ldots + \pi_k \geqslant 55$.

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

Proof. Suppose $k \geqslant \frac{3}{\sqrt[3]{4}}\sqrt[3]{\frac{n}{\ln n - 2\sqrt{\ln n}}}$. Then $k^3 \geqslant \frac{27}{4}\frac{n}{\ln n - 2\sqrt{\ln n}}$ and $\ln k \geqslant \frac{1}{3}(\ln n - \ln(\ln n - 2\sqrt{\ln n}) + \ln\frac{27}{4})$, 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,

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

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

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 6. The contradiction obtained proves the lemma. □

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

Theorem 4 (Upper bound)**.** $\widetilde{g}(n) < e^{\sqrt[3]{2n\ln^2 n}(1+o(1))}$.

The proof of the theorem relies *only* on the upper bound on k, and otherwise ignores the additional constraints in the definition of $\widetilde{g}$ as compared to g. Using further properties of $\widetilde{g}$ in this proof would likely lead to a better bound.

The first step is to simplify the model by replacing the least common multiple of $\pi_1, \ldots, \pi_k$ with their product, and then allowing the cycle lengths to be real numbers. Then, as it is well-known, the maximum of the product is reached for all factors being identical:

Proposition 4. $\max\limits_{x_1+\ldots+x_k \leqslant x} x_1 \ldots x_k = (\frac{x}{k})^k$ *for every* $k \in \mathbb{N}$ *and* $x \in \mathbb{R}_+$.

Another fact about elementary functions is that $(\frac{n}{k})^k$ reaches its maximum at $k = \frac{n}{e}$, and since the values of k allowed by Lemma 7 are much smaller, one should choose k as large as possible to obtain the greatest value of $(\frac{n}{k})^k$.

Proposition 5. *The function* $f(y) = (\frac{n}{y})^y$ *increases on* $0 < y \leqslant \frac{n}{e}$*, has a maximum at* $y = \frac{n}{e}$ *and decreases on* $\frac{n}{e} \leqslant y$.

Proof of Theorem 4. The upper bound is proved by the following chain of inequalities, which uses Lemma 7, Proposition 4 and Proposition 5:

$$\widetilde{g}(n) = \max_{k\geqslant 1}\{\,\mathrm{lcm}(\pi_1,\ldots,\pi_k) \mid \pi_1+\ldots+\pi_k \leqslant n \text{ and } \langle\ldots\rangle\,\} =$$

$$= \max_{k\geqslant 1}\{\,\mathrm{lcm}(\pi_1,\ldots,\pi_k) \mid \pi_1+\ldots+\pi_k \leqslant n, \tfrac{\mathrm{lcm}(\pi_1,\ldots,\pi_k)}{\mathrm{lcm}(\pi_1,\ldots,\pi_{i-1},\pi_{i+1},\ldots\pi_k)} \geqslant 2, \text{ and } \langle\ldots\rangle\,\} =$$

$$= \max_{1\leqslant k<\frac{3}{\sqrt[3]{4}}\sqrt[3]{\frac{n}{\ln n-2\sqrt{\ln n}}}}\{\,\mathrm{lcm}(\pi_1,\ldots,\pi_k) \mid \pi_1+\ldots+\pi_k \leqslant n \text{ and } \langle\ldots\rangle\,\} \leqslant$$

$$\leqslant \max_{1\leqslant k<\frac{3}{\sqrt[3]{4}}\sqrt[3]{\frac{n}{\ln n-2\sqrt{\ln n}}}}\{\,\pi_1\ldots\pi_k \mid \pi_1+\ldots+\pi_k \leqslant n\,\} \leqslant$$

$$\leqslant \max_{1\leqslant k<\frac{3}{\sqrt[3]{4}}\sqrt[3]{\frac{n}{\ln n-2\sqrt{\ln n}}}} \max_{\substack{x_1,\ldots,x_k\in\mathbb{R}_+\\ x_1+\ldots+x_k\leqslant n}} \prod_{i=1}^{k} x_i = \max_{1\leqslant k<\frac{3}{\sqrt[3]{4}}\sqrt[3]{\frac{n}{\ln n-2\sqrt{\ln n}}}} \left(\frac{n}{k}\right)^k \leqslant$$

$$\leqslant \left(\frac{n}{\frac{3}{\sqrt[3]{4}}\sqrt[3]{\frac{n}{\ln n-2\sqrt{\ln n}}}}\right)^{\frac{3}{\sqrt[3]{4}}\sqrt[3]{\frac{n}{\ln n-2\sqrt{\ln n}}}} = e^{\frac{3}{\sqrt[3]{4}}\sqrt[3]{\frac{n}{\ln n-2\sqrt{\ln n}}}\ln\left(\frac{\sqrt[3]{4}}{3}n^{\frac{2}{3}}\sqrt[3]{\ln n-2\sqrt{\ln n}}\right)} <$$

$$< e^{\frac{3}{\sqrt[3]{4}}\sqrt[3]{\frac{n}{\ln n-2\sqrt{\ln n}}}\ln\left(n^{\frac{2}{3}}\sqrt[3]{\ln n}\right)} = e^{\frac{3}{\sqrt[3]{4}}\frac{\sqrt[3]{n}}{\sqrt[3]{\ln n}}\sqrt[3]{\frac{\ln n}{\ln n-2\sqrt{\ln n}}}\left(\frac{2}{3}\ln n+\frac{1}{3}\ln\ln n\right)} =$$

$$= e^{\frac{3}{\sqrt[3]{4}}\frac{\sqrt[3]{n}}{\sqrt[3]{\ln n}}\sqrt[3]{1+\frac{2\sqrt{\ln n}}{\ln n-2\sqrt{\ln n}}}\frac{2}{3}\ln n\left(1+\frac{\ln\ln n}{2\ln n}\right)} = e^{\sqrt[3]{2}\sqrt[3]{n}(\ln n)^{\frac{2}{3}}(1+o(1))}.$$

□

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

Proposition 6 (Bach and Shallit [1])**.** $\sum_{i=1}^{k} p_i = (1+o(1))\frac{1}{2}k^2\ln k$.

Proposition 7. $\prod_{i=1}^{k} p_i = e^{(1+o(1))k\ln k}$.

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

Theorem 5 (Lower bound)**.** $\widetilde{g}(n) > e^{\sqrt[3]{\frac{2}{9}}\sqrt[3]{n\ln^2 n}(1+o(1))}$.

Proof. For any k, consider the numbers kp_i with $i \in \{1,\ldots,k\}$. These numbers satisfy the definition of $\widetilde{g}$ 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 $\widetilde{g}$ on s_k must be at least $\mathrm{lcm}(kp_1,\ldots,kp_k) = k\prod_{i=1}^{k} p_i$.

By Proposition 6, the argument of $\widetilde{g}$ is estimated as

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

Note that

$$s_{k+1} = (1+o(1))\tfrac{1}{2}(k+1)^3\ln(k+1) = (1+o(1))\tfrac{1}{2}k^3(\ln k)\frac{(k+1)^3}{k^3}\frac{\ln(k+1)}{\ln k} =$$
$$= (1+o(1))\tfrac{1}{2}k^3(\ln k)(1+\tfrac{O(k^2)}{k^3})(1+\tfrac{\ln\frac{k+1}{k}}{\ln k}) = (1+o(1))\tfrac{1}{2}k^3\ln k.$$

Let $f : \mathbb{N} \to \mathbb{R}$ 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 $s_k \leqslant n$. Then

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

Using Proposition 7 to estimate the product of the first k primes, the value of $\widetilde{g}$ is

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

Using the inequality (1), the expression $k\ln k$ can be estimated by the following function of n:

$$\sqrt[3]{\tfrac{2}{9}}\sqrt[3]{n}\ln^{\frac{2}{3}} n < \sqrt[3]{\tfrac{2}{9}}\sqrt[3]{1+f(k)}\frac{1}{\sqrt[3]{2}}k\sqrt[3]{\ln k}\ln^{\frac{2}{3}}\big((1+f(k))\tfrac{1}{2}k^3\ln k\big) =$$
$$= \sqrt[3]{\tfrac{1}{9}}\sqrt[3]{1+f(k)}k\sqrt[3]{\ln k}\big[3\ln k + \ln(1+f(k)) + \ln\tfrac{\ln k}{2}\big]^{\frac{2}{3}} =$$
$$= \sqrt[3]{\tfrac{1}{9}}\sqrt[3]{1+f(k)}k\sqrt[3]{\ln k}\sqrt[3]{9}(\ln^{\frac{2}{3}} k)\big[1+\tfrac{\ln(1+f(k))}{3\ln k}+\tfrac{\ln\frac{\ln k}{2}}{3\ln k}\big]^{\frac{2}{3}} = k\ln k(1+o(1)),$$

which leads to the following lower bound on $\widetilde{g}(n)$:

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

□

According to Theorems 4–5, the values of the function $\widetilde{g}$ are confined within the following bounds:

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

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

Improving this estimation is an interesting theoretical question. Perhaps it could be proved that $\widetilde{g}$ is of the order $e^{C\sqrt[3]{n\ln^2 n}(1+o(1))}$, for some constant C with $0.605 < \sqrt[3]{\frac{2}{9}} \leqslant C \leqslant \sqrt[3]{2} < 1.260$. 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 $\widetilde{g}(n)$. Lemma 6 gives a

lower bound of $\frac{4}{9}(1+o(1))k^3 \ln k$. At the same time, the proof of Theorem 5 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 $\frac{4}{9} \leqslant C' \leqslant \frac{1}{2}$. The gap between $C' = \frac{4}{9}$ and $C' = \frac{1}{2}$ reflects several essential simplifications made in 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 7 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 $\widetilde{g}(n)$ given in Theorem 4 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 5, the lower bound on $\widetilde{g}$ 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 $\widetilde{g}(n) \leqslant f_{\text{UFA–DFA}} \leqslant \widetilde{g}(n) + n$, while in the case of UFAs with a unique initial state, $\widetilde{g}(n-1) \leqslant f_{\text{UFA}_1\text{–DFA}} \leqslant \widetilde{g}(n-1) + n$. Therefore, both functions asymptotically behave as $\widetilde{g}$:

Corollary 2. *$f_{UFA–DFA} = e^{\Theta\left(\sqrt[3]{n(\ln n)^2}\right)}$ and $f_{UFA_1–DFA} = e^{\Theta\left(\sqrt[3]{n(\ln n)^2}\right)}$.*

6 NFA–UFA tradeoff

An NFA can be transformed to an equivalent UFA simply by transforming it to a DFA. It turns out that for some NFAs no better transformation is possible:

Lemma 8. *For all $k \geqslant 1$ and $p_1, \ldots, p_k \geqslant 2$, the language*

$$L = \{\varepsilon\} \cup a\bigcup_{i=1}^{k}\{\varepsilon, a, a^2, \ldots, a^{p_i-2}\}(a^{p_i})^* = \overline{(a^{\operatorname{lcm}(p_1,\ldots,p_k)})^*} \cup \{\varepsilon\}$$

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

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

The smallest DFA for L contains an accepting initial state and a loop of length $\operatorname{lcm}(p_1, \ldots, p_k)$, 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 [19].

Let $n = \operatorname{lcm}(p_1, \ldots, p_k)$ and consider the strings $u_i = v_i = a^{i-1}$ for $1 \leqslant i \leqslant n+1$. The corresponding $(n+1) \times (n+1)$ matrix M is defined by $M_{i,j} = 0$ for $i + j = n + 2$ and for $i = j = n + 1$, and $M_{i,j} = 1$ for the rest of the entries. 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:

$$\det M = \begin{vmatrix} 1 & 1 & 1 & \ldots & 1 & 1 & 0 \\ 1 & 1 & 1 & \ldots & 1 & 0 & 1 \\ 1 & 1 & 1 & \ldots & 0 & 1 & 1 \\ \vdots & \vdots & \vdots & \iddots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & \ldots & 1 & 1 & 1 \\ 1 & 0 & 1 & \ldots & 1 & 1 & 1 \\ 0 & 1 & 1 & \ldots & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & \ldots & 1 & 1 & 0 \\ 0 & 0 & 0 & \ldots & 0 & -1 & 1 \\ 0 & 0 & 0 & \ldots & -1 & 0 & 1 \\ \vdots & \vdots & \vdots & \iddots & \vdots & \vdots & \vdots \\ 0 & 0 & -1 & \ldots & 0 & 0 & 1 \\ 0 & -1 & 0 & \ldots & 0 & 0 & 1 \\ -1 & 0 & 0 & \ldots & 0 & 0 & 0 \end{vmatrix} =$$

$$= \begin{vmatrix} 0 & 0 & 0 & \ldots & 0 & 0 & n-1 \\ 0 & 0 & 0 & \ldots & 0 & -1 & 1 \\ 0 & 0 & 0 & \ldots & -1 & 0 & 1 \\ \vdots & \vdots & \vdots & \iddots & \vdots & \vdots & \vdots \\ 0 & 0 & -1 & \ldots & 0 & 0 & 1 \\ 0 & -1 & 0 & \ldots & 0 & 0 & 1 \\ -1 & 0 & 0 & \ldots & 0 & 0 & 0 \end{vmatrix} = (-1)^n \cdot (n-1).$$

Since the determinant is nonzero, 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. □

Thus $g(n-1)+1$ is a lower bound on the NFA to UFA transformation. An asymptotically matching upper bound of $g(n-1) + O(n^2)$ is given by Chrobak's [3, Thm. 4.4] construction, which begins by converting an n-state NFA to the Chrobak normal form with a tail of length $O(n^2)$ and with at most $n-1$ states in the cycles, and then proceeds by determinizing the cycles, making at most $g(n-1)$ states.

Theorem 6. *For every $n \geqslant 1$, the number of states in a UFA sufficient and, in the worst case, necessary to represent languages recognized by n-state NFAs is $g(n-1) + O(n^2) = e^{\sqrt{n \ln n}(1+o(1))}$.*

7 Complementing unary UFAs

The first operation is complementation: with respect to DFAs, it has state complexity n, since in order to represent the complement of a language recognized by a DFA, it is sufficient to complement its set of accepting states. For unary NFAs, Holzer and Kutrib [5] have shown that complementation may require a blowup of up to $g(n)$ states: that is, complementing some unary NFAs basically requires determinizing them.

The situation with UFAs is quite nontrivial. On one hand, for a substantial class of UFAs, the complement can be constructed by changing the set of accepting states, like in the case of DFAs. On the other hand, it will be proved that complementing some UFAs requires additional states.

The following subclass of UFAs allows efficient complementation:

Lemma 9. *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, such that for every two accepting states $r_{i,f}, r_{j,f'} \in F$ with $i \neq j$, it holds that $f \neq f' \pmod{p}$. 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 $\{0, \ldots, p-1\}$ is partitioned into disjoint sets $S_1, \ldots, S_k$, such that a state $r_{i,f}$ may be accepting only if the number f modulo p is in S_i. That is, 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 Theorem 2 are still met for the new automaton, that is, it remains a 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 construction, this string is accepted by the new automaton if and only if $r_{i,f} \notin F$. □

In particular, this lemma is applicable to all UFAs with $k = 2$ cycles, such as the one in Figure 1. But for $k \geqslant 3$ the lengths of the cycles need not have a common divisor, which gives examples of UFAs not covered by the above lemma. Sometimes the lengths of the cycles may have a common divisor, yet the separation of offsets required by Theorem 2 would not be possible. 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 Theorem 2, as $7 \neq 8 \pmod 2$, $7 \neq 9 \pmod 4$ and $8 \neq 9 \pmod 2$. This leads to a UFA with $1+8+10+12 = 31$ states recognizing the language $a^8(a^8)^* \cup a^9(a^{10})^* \cup a^{10}(a^{12})^*$.*

However, $\gcd(8, 10, 12) = 2$ and $7 = 9 \pmod 2$, and thus Lemma 9 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 whose lengths 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 10. *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*

$$\begin{aligned} L_1 &= a\{a^{p_1}, a^{2p_1}, \ldots, a^{(p_2-1)p_1}\}(a^{p_1p_2})^*, \\ L_2 &= a\{a^{p_2}, a^{2p_2}, \ldots, a^{(p_3-1)p_2}\}(a^{p_2p_3})^* \quad \textit{and} \\ L_3 &= a\{a^{p_3}, a^{2p_3}, \ldots, a^{(p_1-1)p_3}\}(a^{p_1p_3})^*, \end{aligned}$$

has a UFA with $p_1p_2 + p_2p_3 + p_1p_3 + 1$ states, while every NFA for $\overline{L}$ contains at least $p_1p_2p_3$ states.

Proof. The construction of the UFA for L is straightforward. It has a tail of length 1 and three cycles of length p_1p_2, p_2p_3 and p_1p_3, with accepting states $r_{1,\ell_1 p_1}$ for $\ell_1 \in \{1, \ldots, p_2 - 1\}$, $r_{2,\ell_2 p_2}$ for $\ell_2 \in \{1, \ldots, p_3 - 1\}$ and $r_{3,\ell_3 p_3}$ for $\ell_3 \in \{1, \ldots, p_1 - 1\}$. To see that the condition of Theorem 2 is satisfied, consider the offsets of accepting states in the cycles modulo p_1, p_2 and p_3:

	(mod p_1)	(mod p_2)	(mod p_3)
$\{a^{p_1}, a^{2p_1}, \ldots, a^{(p_2-1)p_1}\}$ (mod p_1p_2)	0	$\{1, \ldots, p_2 - 1\}$	$\ldots$
$\{a^{p_2}, a^{2p_2}, \ldots, a^{(p_3-1)p_2}\}$ (mod p_2p_3)	$\ldots$	0	$\{1, \ldots, p_3 - 1\}$
$\{a^{p_3}, a^{2p_3}, \ldots, a^{(p_1-1)p_3}\}$ (mod p_1p_3)	$\{1, \ldots, p_1 - 1\}$	$\ldots$	0

Now the offsets of accepting states in the first and the second cycles are different modulo $p_2 = \gcd(p_1p_2, p_2p_3)$, etc.

In order to show that no UFA for $\overline{L}$ can have fewer than $p_1p_2p_3$ states, it is sufficient to establish the following statement:

Claim 3. *Every infinite regular subset of $\overline{L}$ has period divisible by $p_1p_2p_3$.*

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, and consider the number n modulo p_1:

- If $n = 0 \pmod{p_1}$, then $n = 0 \pmod{p_2}$ (otherwise, if $n \not\equiv 0 \pmod{p_2}$, the string a^{1+n} would be in L_1), and $n = 0 \pmod{p_3}$ as well (otherwise $a^{1+n} \in L_2$). Since a^{1+n} is accepted in a cycle of length p, the string a^{1+n+pp_3} is accepted as well. This string satisfies $n + pp_3 = 0 \pmod{p_3}$, but at the same time $n + pp_3 \not\equiv 0 \pmod{p_1}$, and therefore $a^{1+n+pp_3} \in L_3$, which is a contradiction.

- If $n \not\equiv 0 \pmod{p_1}$, then $n \not\equiv 0 \pmod{p_3}$ (otherwise $a^{1+n} \in L_3$) and $n \not\equiv 0 \pmod{p_2}$ (otherwise $a^{1+n} \in L_2$). Let i be p taken modulo p_1 ($i \not\equiv 0$ by assumption), and consider the string $a^{1+n+(p_1-i)pp_2}$, which should be accepted in the same cycle as a^{1+n}. However, $n + (p_1 - i)pp_2 \not\equiv 0 \pmod{p_2}$ and $n + (p_1 - i)pp_2 = i + (p_1 - i) = 0 \pmod{p_1}$, and accordingly $a^{1+n+(p_1-i)pp_2} \in L_1$. The contradiction obtained completes the proof of the claim.

Now every NFA recognizing $\overline{L}$ should have a tail and one or more cycles, and the combination of tail and each of these cycles is a DFA recognizing some subset of $\overline{L}$. Therefore, the length of each cycle of this NFA must be a multiple of $p_1p_2p_3$, which proves the lemma. □

In particular, applying this lemma for $p_1 = 3$, $p_2 = 5$ and $p_3 = 7$ gives a language recognized by a UFA with $72 = 15 + 35 + 21 + 1$ states, while its complement requires a UFA with at least $105 = 3 \cdot 5 \cdot 7$ states. Witness languages of this form lead to the following fairly modest lower bound.

Theorem 7. *The state complexity of complementation for UFAs over a unary alphabet is at least* $\frac{1}{42}n\sqrt{n}$ *(for all* $n \geqslant 867$*) and at most* $f_{UFA\text{–}DFA}(n)$.

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 [18] that for every $m \geqslant 17$ 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

$$\sqrt{\tfrac{n}{12}} < p_1 < p_2 < p_3 \leqslant \sqrt{\tfrac{n}{3}}.$$

By Lemma 10, there is a language L recognized by a UFA with

$$p_1p_2 + p_2p_3 + p_1p_3 + 1 \leqslant 3\sqrt{\tfrac{n}{3}}\Big(\sqrt{\tfrac{n}{3}} - 2\Big) + 1 = n - 6\sqrt{\tfrac{n}{3}} + 1 \leqslant n$$

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

$$p_1p_2p_3 \geqslant \Big(\sqrt{\tfrac{n}{12}}\Big)^3 = \big(\tfrac{1}{12}\big)^{\frac{3}{2}} n\sqrt{n} \geqslant \tfrac{1}{42}n\sqrt{n}$$

states, which proves the lower bound. □

Better lower bounds can be obtained from the following generalization of Lemma 10 to any number of prime divisors:

Lemma 11. *Let* $k \geqslant 1$ *and let* $p_1, \ldots, p_{2k+1}$ *be any pairwise distinct primes. Then the language* $L = \bigcup_{i=1}^{2k+1} L_i$*, where*

$$L_i = \{\, a^{1+n} \mid n \neq 0 \pmod{p_i},\ n = 0 \pmod{p_{i+1} \ldots p_{i+k}} \,\}$$

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

Proof. A UFA for L has a tail of length 1 and $2k + 1$ cycles, with each ith cycle of length $\widehat{p}_i = p_ip_{i+1} \ldots p_{i+k}$, containing accepting states $r_{i,\ell p_{i+1} \ldots p_{i+k}}$ for $\ell \in \{1, \ldots, p_i - 1\}$.

To see that the condition of Theorem 2 is satisfied, consider any ith and any jth 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 $\{p_{j+1}, \ldots, p_{j+k}\}$, or p_j belongs to $\{p_{i+1}, \ldots, p_{i+k}\}$. Assume, without loss of generality, that the former is the case. Then p_i is a common divisor of $\widehat{p}_i$ and $\widehat{p}_j$, and for every two accepting states $r_{i,\ell p_{i+1}\ldots p_{i+k}}$ and $r_{j,\ell' p_{j+1}\ldots p_{j+k}}$ the number $\ell p_{i+1} \ldots p_{i+k}$ is nonzero modulo p_i, while $\ell' p_{j+1} \ldots p_{j+k}$ is divisible by p_i. Therefore, $\ell p_{i+1} \ldots p_{i+k} \neq \ell' p_{j+1} \ldots p_{j+k} \pmod{\gcd(\widehat{p}_i, \widehat{p}_j)}$.

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

Claim 4. *Every infinite periodic subset of $\overline{L}$ containing any string a^{1+n} with $n = 0 \pmod{p_1 \ldots p_{2k+1}}$ has period divisible by $p_1 \ldots p_{2k+1}$.*

Indeed, if p is the period of such a subset and it is not divisible by some p_i, for any $i \in \{1, \ldots, 2k+1\}$, then the string $w = a^{1+n+p(p_1\ldots p_{i-1}p_{i+1}\ldots p_{2k+1})}$ belongs to this subset as well. Since $n + p(p_1 \ldots p_{i-1}p_{i+1} \ldots p_{2k+1}) = n = 0 \pmod{p_j}$ for every $j \neq i$, but $n + p(p_1 \ldots p_{i-1}p_{i+1} \ldots p_{2k+1}) \neq 0 \pmod{p_i}$, the string w belongs to L_i, which is a contradiction.

Now consider that an NFA for the language $\overline{L}$ should accept all strings in $a(a^{p_1\cdots p_{2k+1}})^*$, and all but finitely many of them are accepted in the periodic part. Then, by the above claim, the period of the periodic part must be a multiple of $p_1 \ldots p_{2k+1}$. □

Lemma 12. *Let $k \geqslant 1$. Then the number of states in an NFA necessary to represent complements of n-state UFAs over a unary alphabet is at least $\frac{1}{2^{2k+1}(2k+1)^2} \cdot n^{2-\frac{1}{k+1}}$ for all $n \geqslant (2k+1)(4(2k+1)\ln 4(2k+1))^{k+1}$.*

Proof. Let r_i denote ith Ramanujan prime, that is, the smallest integer, such that for every $m \geqslant r_i$ 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 [18], 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 $(2k+1) \cdot (r_{2k+1})^{k+1}$ ($k = 1, 2, 3, \ldots$ this means that $n \geqslant 867, 344605, 84821527, \ldots$) Then there exist $2k+1$ primes $p_1, \ldots, p_{2k+1}$ with

$$\tfrac{1}{2}\sqrt[k+1]{\tfrac{n}{2k+1}} < p_1 < \ldots < p_{2k+1} \leqslant \sqrt[k+1]{\tfrac{n}{2k+1}}.$$

By Lemma 10, there is a language L recognized by a UFA with

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

$$= n - 2(2k+1)\sqrt[k+1]{\tfrac{n}{2k+1}} + 1 \leqslant n$$

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

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

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 [20]: $2i \ln 2i < r_i < 4i \ln 4i$. Then $(2k+1) \cdot (r_{2k+1})^{k+1} < (2k+1)(4(2k+1)\ln 4(2k+1))^{k+1}$. □

Theorem 8. *The state complexity of complementation for UFAs over a unary alphabet is at least $n^{2-o(1)}$ and at most $f_{UFA\text{-}DFA}(n)$.*

Proof. According to Lemma 12, the function $f(n)$ defined by

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

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 $\lim_{n\to\infty} h(n) = 0$.

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

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

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

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

and therefore $h(n) \leqslant \varepsilon$. □

8 State complexity of intersection and star

Now consider the operation of *intersection*, which has state complexity mn both for DFAs [13, 22] and for NFAs [5], and both over unary and larger alphabets. It maintains the same complexity for UFAs:

Lemma 13. *For every alphabet Σ and for all $m, n \geqslant 1$, 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 8 (Holzer, Kutrib [5])**.** *For all relatively prime $m, n \geqslant 2$, the language $(a^{mn})^* = (a^m)^* \cap (a^n)^*$ requires an NFA with at least mn states.*

Theorem 9. *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*: its state complexity for unary DFAs is $(n-1)^2+1$, obtained by Yu, Zhuang and Salomaa [22, Thm. 5.3]. An identical result holds for UFAs, in spite of the differences between the two models.

Lemma 14 (Yu, Zhuang and Salomaa [22])**.** *For every language $L \subseteq a^*$ recognized by an n-state unary NFA in Chrobak normal form, there exists a DFA for L^* with $(n-1)^2+1$ states.*

Strictly speaking, Yu, Zhuang and Salomaa [22] established this result for L represented by a DFA, but their argument can be entirely replicated to prove Lemma 14 as stated.

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 as DFAs.

Lemma 15. *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 $k \geqslant 1$ cycles of length $p_1, \ldots, p_k$. It is claimed that every string of length ℓ or more is accepted by A.

Let $n \geqslant \ell$ and consider the string $a^{\ell+m\cdot\mathrm{lcm}(p_1,\ldots,p_k)}$, which is longer than a^m and hence is in L. Let this string be accepted in an ith cycle, that is, in state $r_{i,n-\ell+m\cdot\mathrm{lcm}(p_1,\ldots,p_k)}$, 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, m should be at most $\ell-1$. Therefore, the tail of A contains at least $m+1$ states, while the loops consist of at least one state, which proves the lower bound of $m+2$. □

Theorem 10. *For every $n \geqslant 1$, star of an n-state UFA is representable by a UFA with $(n-1)^2+1$ states, and this number of states is in the worst case necessary.*

Proof. The upper bound is given in Lemma 14.

For the lower bound, consider the language $L = a^{n-1}(a^n)^*$. As noted by Yu, Zhuang and Salomaa [22], its star L^* is co-finite, and the longest string not belonging to it is $a^{(n-2)n}$. Then, by Lemma 15, every UFA for L^* requires at least $(n-2)n+2 = (n-1)^2+1$ states. □

9 Conclusion

The refinement of Chrobak normal form for the unambiguous case has proved to be a useful tool for studying unary UFAs. The main result is that the transformation of unary UFAs to DFAs leads to an exponential blowup, which is, however, smaller than the unary NFA to DFA blowup. The new variant of Landau's function, which characterizes the UFA to DFA blowup, deserves a further study: 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 asymptotic estimation than the given $e^{\Theta(\sqrt[3]{n \ln^2 n})}$, perhaps an estimation of the form $e^{C\sqrt[3]{n \ln^2 n}(1+o(1))}$. Another question of interest is to determine an efficient method of computing the values of $\widetilde{g}$.

The complexity of operations on UFAs, in particular the complexity of complementing them, is left as the main open problem. It is unlikely that complementation could be done using as few as n^2 states. Most probably the complexity is exponential, yet perhaps not of the order of $f_{\text{UFA-DFA}}(n)$. However, proving such stronger lower bounds requires a deeper analysis than in Lemma 10.

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 and to Hermann Gruber for their helpful comments on the manuscript. Research supported by the Academy of Finland under grant 134860.

References

[1] E. Bach, J. Shallit, *Algorithmic Number Theory, Vol. 1: Efficient Algorithms*, MIT Press, 1996.

[2] H. Björklund, W. Martens, "The tractability frontier for NFA minimization", *Automata, Languages and Programming* (ICALP 2008, Reykjavík, Iceland, July 6–13, 2008), part II, LNCS 5126, 27–38.

[3] M. Chrobak, "Finite automata and unary languages", *Theoretical Computer Science*, 47 (1986), 149–158; errata: 302:1–3 (2003), 497–498.

[4] V. Geffert, C. Mereghetti, G. Pighizzini, "Complementing two-way finite automata", *Information and Computation*, 205:8 (2007), 1173–1187.

[5] M. Holzer, M. Kutrib, "Nondeterministic descriptional complexity of regular languages", *International Journal of Foundations of Computer Science*, 14 (2003), 1087–1102.

[6] 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.

[7] A. W. Ingleton, "The rank of circulant matrices", *Journal of the London Mathematical Society*, 31 (1956), 445–460.

[8] 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.

[9] H. Leung, "Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata", *SIAM Journal on Computing*, 27:4 (1998), 1073–1082.

[10] H. Leung, "Descriptional complexity of NFA of different ambiguity", *International Journal of Foundations of Computer Science*, 16:5 (2005), 975–984.

[11] Yu. Lyubich, "Bounds for the optimal determinization of nondeterministic autonomic automata", *Sibirskii Matematicheskii Zhurnal*, 2 (1964), 337–355, in Russian.

[12] 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.

[13] A. N. Maslov, "Estimates of the number of states of finite automata", *Soviet Mathematics Doklady*, 11 (1970), 1373–1375.

[14] F. Mera, G. Pighizzini, "Complementing unary nondeterministic automata", *Theoretical Computer Science*, 330:2 (2005), 349–360.

[15] C. Mereghetti, G. Pighizzini, "Optimal simulations between unary automata", *SIAM Journal on Computing*, 30:6 (2001), 1976–1992.

[16] W. Miller, "The maximum order of an element of a finite symmetric group", *American Mathematical Monthly*, 94:6 (1987), 497–506.

[17] 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.

[18] S. Ramanujan, "A proof of Bertrand's postulate", *Journal of the Indian Mathematical Society*, 11 (1919), 181–182.

[19] E. M. Schmidt, *Succinctness of Description of Context-Free, Regular and Unambiguous Languages*, Ph. D. thesis, Cornell University, 1978.

[20] J. Sondow, "Ramanujan primes and Bertrand's postulate", *American Mathematical Monthly*, 116 (2009), 630–635.

[21] 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.

[22] S. Yu, Q. Zhuang, K. Salomaa, "The state complexity of some basic operations on regular languages", *Theoretical Computer Science*, 125 (1994), 315–328.

Turku
Centre *for*
Computer
Science

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematical Sciences

Åbo Akademi University
- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration
- Institute of Information Systems Sciences

ISBN 978-952-12-2328-0
ISSN 1239-1891