

# Complexity of Unary Exclusive Nondeterministic Finite Automata

Martin Kutrib      Andreas Malcher  
Matthias Wendlandt

Institut für Informatik, Universität Giessen  
Arndtstr. 2, 35392 Giessen, Germany

{kutrib, andreas.malcher, matthias.wendlandt}@informatik.uni-giessen.de

Exclusive nondeterministic finite automata (XNFA) are nondeterministic finite automata with a special acceptance condition. An input is accepted if there is exactly one accepting path in its computation tree. If there are none or more than one accepting paths, the input is rejected. We study the descriptive complexity of XNFA accepting unary languages. While the state costs for mutual simulations with DFA and NFA over general alphabets differ significantly from the known types of finite automata, it turns out that the state costs for the simulations in the unary case are in the order of magnitude of the general case. In particular, the state costs for the simulation of an XNFA by a DFA or an NFA are  $e^{\Theta(\sqrt{n \cdot \ln n})}$ . Conversely, converting an NFA to an equivalent XNFA may cost  $e^{\Theta(\sqrt{n \cdot \ln n})}$  states as well. All bounds obtained are also tight in the order of magnitude. Finally, we investigate the computational complexity of different decision problems for unary XNFAs and it is shown that the problems of emptiness, universality, inclusion, and equivalence are coNP-complete, whereas the general membership problem is NL-complete.

## 1 Introduction

The ability of using nondeterminism for finite automata does not increase their computational power in comparison with the deterministic variant, but the simulation costs for a deterministic finite automaton (DFA) can be exponentially higher in terms of states than for an equivalent nondeterministic finite automaton (NFA) [21, 23].

In the last decades several structural extensions of finite automata have been examined. One such extension is, for example, to give the reading head of the finite automaton the power of two-way motion. Such *two-way* finite automata do also not increase the computational power of finite automata [30], but they are interesting from a descriptive complexity point of view, since the costs for one-way deterministic finite automata for the simulation of two-way deterministic finite automata can be exponential in the number of states [23]. Similar results can also be shown for the nondeterministic case [31].

A more fine-grained look on the range between nondeterministic and deterministic finite automata leads to the model of *unambiguous* finite automata [32]. Here, nondeterminism is allowed, but for every accepted word there has to be exactly one accepting path. From a descriptive complexity perspective it is known that the trade-off from unambiguous finite automata to DFAs is exponential as well [17, 18, 32].

In contrast to these structural extensions, another extension is examined in [13, 14] that is based on the acceptance conditions of the automata and which leads to *exclusive* nondeterministic finite automata (XNFA). In this model, the computation tree of an input is defined in the same way as for nondeterministic finite automata, but its interpretation is different. Namely, an input word  $w$  is accepted, if there is exactly one accepting path for  $w$ . If there is no accepting path for  $w$  or two or more accepting paths for  $w$ , then  $w$  is rejected. Clearly, any unambiguous finite automaton can be considered as an XNFA, but

in comparison to unambiguous finite automata, multiple accepting paths are allowed and lead to non-acceptance in an XNFA. In [13, 14] complexity aspects of XNFAs have been investigated. Concerning the descriptonal complexity, it is shown that  $n$ -state XNFAs can be determinized as well, but the upper bound turns out to be  $3^n - 2^n + 1$  and is shown to be tight. Moreover,  $n \cdot 2^{n-1}$  states are shown to be a tight bound for the simulation of an XNFA by an equivalent NFA. The simulation of an NFA by an equivalent XNFA leads to an upper bound of  $2^n - 1$  which is shown to be tight as well. Concerning the computational complexity, it is shown that the problems of emptiness, universality, inclusion, and equivalence are PSPACE-complete, whereas the general membership problem is NL-complete. It should be noted that a computational model with exactly one accepting computation on every accepted input has already been known in the context of complexity theory as the class US (unique solution). It is defined (see [1]) as the class of languages  $L$  for which there exists a nondeterministic polynomial time Turing machine  $M$  such that  $w \in L$  if and only if  $M$  has on input  $w$  exactly one accepting computation path. A short overview on the properties of the class US may be found in [8].

In this paper, we investigate the descriptonal and computational complexity of XNFAs accepting *unary* languages. The descriptonal complexity of unary regular languages has extensively been studied in the literature. A fundamental result was obtained by Chrobak in [2, 3]. He shows that  $O(F(n))$  is a tight bound for the simulation of an NFA by an equivalent DFA. Here,  $F(n)$  denotes Landau's function [15] that is the maximal order of the cyclic subgroups of the symmetric group on  $n$  elements and can be estimated as  $F(n) \in e^{\Theta(\sqrt{n \cdot \ln n})}$ . Landau's function plays a crucial role in many results on the descriptonal complexity of unary regular languages. One line of research in the past years is that many automata models such as, for example, one-way finite automata, two-way finite automata, pushdown automata, and context-free grammars have been investigated and compared to each other with respect to simulation results and the size costs of the simulation (see, for example, [6, 20, 25, 26, 29]). Another line of research in recent years concerns investigations on the state complexity of operations on unary languages which can be found, for example, in [9, 12, 19, 28].

The paper is structured as follows. In Section 2, we give the basic definitions that are used in the further sections. In Section 3, we study the descriptonal costs for determinizing a given unary XNFA. As a fundamental preparatory step we show that any unary  $n$ -state XNFA can be converted to an equivalent  $O(n^3)$ -state XNFA in Chrobak normal form. This result is in slight contrast to NFAs where the conversion of an arbitrary NFA to Chrobak normal form may induce only a quadratic blow-up of the number of states. Based on the XNFA in Chrobak normal form we can construct an equivalent DFA whose number of states is bounded by  $e^{\Theta(\sqrt{n \cdot \ln n})}$ . This upper bound is also tight in the order of magnitude. In Section 4, we obtain similar upper and lower bounds for the conversion of unary XNFAs to equivalent NFAs and of unary NFAs to equivalent XNFAs. Finally, in Section 5 we study the computational complexity of decidability questions. In particular, we consider general membership, emptiness, universality, inclusion, and equivalence with respect to the unary case and show that for unary XNFAs the general membership problem is NL-complete, whereas the questions of emptiness, finiteness, inclusion, and equivalence are coNP-complete.

## 2 Definitions and Preliminaries

Let  $\Sigma^*$  denote the set of all words over the finite alphabet  $\Sigma$ . The *empty word* is denoted by  $\lambda$ , and  $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$ . The *reversal* of a word  $w$  is denoted by  $w^R$ . For the *length* of  $w$  we write  $|w|$ . We use  $\subseteq$  for *inclusions* and  $\subset$  for *strict inclusions*. We write  $2^S$  for the power set and  $|S|$  for the cardinality of a set  $S$ .

A *nondeterministic finite automaton* (NFA) is a system  $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ , where  $Q$  is the finite set of *states*,  $\Sigma$  is the finite set of *input symbols*,  $q_0 \in Q$  is the *initial state*,  $F \subseteq Q$  is the set of *accepting states*, and  $\delta: Q \times \Sigma \rightarrow 2^Q$  is the *transition function*.

With an eye towards further modes of acceptance, we define the *acceptance of an input* in terms of computation trees. For any input  $w = a_1 a_2 \cdots a_n \in \Sigma^*$  read by some NFA  $M$ , a (*complete*) *path for  $w$*  is a sequence of states  $q_0, q_1, \dots, q_n$  such that  $q_{i+1} \in \delta(q_i, a_{i+1})$ ,  $0 \leq i \leq n-1$ . All possible paths on  $w$  are combined into a *computation tree* of  $M$  on  $w$ . So, a computation tree of  $M$  is a finite rooted tree whose nodes are labeled with states of  $M$ . In particular, the root is labeled with the initial state, and the successor nodes of a node labeled  $q$  are the nodes  $p_1, p_2, \dots, p_m$  if and only if  $\delta(q, a) = \{p_1, p_2, \dots, p_m\}$ , for the current input symbol  $a$ . A path in the computation tree is an *accepting path* if it ends in an accepting state.

Now, an input  $w$  is accepted by an NFA if at least one path in the computation tree of  $w$  is accepting.

An NFA, where for acceptance it is required that *exactly* one path is accepting, is called an *exclusive nondeterministic finite automaton* (XNFA).

The *language accepted* by the XNFA  $M$  is  $L(M) = \{w \in \Sigma^* \mid w \text{ is accepted by } M\}$ .

Finally, an NFA is a *deterministic finite automaton* (DFA) if and only if  $|\delta(q, a)| = 1$ , for all  $q \in Q$  and  $a \in \Sigma$ . In this case we simply write  $\delta(q, a) = p$  for  $\delta(q, a) = \{p\}$  assuming that the transition function is a mapping  $\delta: Q \times \Sigma \rightarrow Q$ . So, any DFA is complete, that is, the transition function is total, whereas for the other automata types it is possible that  $\delta$  maps to the empty set. A finite automaton is called *unary* if its set of input symbols is a singleton. In this case we use  $\Sigma = \{a\}$  throughout the paper.

### 3 Determinization of unary XNFAs

The problem of evaluating the costs of unary automata simulations was raised in [34], and has led to emphasize some relevant differences with the general case. For example, unary NFAs can be much more concise than DFAs, but yet not as much as for the general case. Moreover, the sophisticated studies in [20] reveal tight bounds for many other types of unary finite automata conversions. The paper and the survey [27] are also a valuable source for further references.

For state complexity issues of unary finite automata, Landau's function

$$F(n) = \max\{\text{lcm}(c_1, c_2, \dots, c_l) \mid l \geq 1, c_1, c_2, \dots, c_l \geq 1, c_1 + c_2 + \dots + c_l = n\}$$

which gives the maximal order of the cyclic subgroups of the symmetric group on  $n$  elements, plays a crucial role, where  $\text{lcm}$  denotes the *least common multiple* [15, 16]. It is well known that the  $c_i$  always can be chosen to be relatively prime. Moreover, an easy consequence of the definition is that the  $c_i$  always can be chosen such that  $c_1, c_2, \dots, c_l \geq 2$ ,  $c_1 + c_2 + \dots + c_l \leq n$ , and  $\text{lcm}(c_1, c_2, \dots, c_l) = F(n)$  (cf., for example, [24]).

Since  $F$  depends on the irregular distribution of the prime numbers we cannot expect to express  $F(n)$  explicitly by  $n$ . In [15, 16] the asymptotic growth rate  $\lim_{n \rightarrow \infty} (\ln F(n) / \sqrt{n \cdot \ln n}) = 1$  was determined, which for our purposes implies the (sufficient) rough estimate  $F(n) \in e^{\Theta(\sqrt{n \cdot \ln n})}$  (see also [4, 36] for bounds on  $F$ ).

The asymptotically tight bound of  $F(n)$  for the unary NFA-to-DFA conversion was presented in [2, 3]. The proof is based on a normal form for unary NFAs derived in [2]. Each  $n$ -state unary NFA can effectively be converted into an equivalent  $O(n^2)$ -state NFA in this so-called Chrobak normal form. However, the original proof in [2] contains an error that has been discovered and fixed in [37]. While the correction increases the state costs, their order of magnitude is not affected. In connection with magic

numbers, more precise and improved state bounds have been shown in [5] by a completely different proof.

Let  $t, d \geq 0$  be two integers. An *arithmetic progression with offset  $t$  and period  $d$*  is the set

$$\{t + x \cdot d \mid x \geq 0\}.$$

We recall a well-known useful fact which is related to number theory and Frobenius numbers (see, for example, [33] for a survey).

**Lemma 1.** *Let  $0 < c_1 < c_2 < \dots < c_r \leq n$  be positive integers. Then the set of integers  $z > n^2$  that can be written as a non-negative integer linear combination of the  $c_i$  is  $\{t + x \cdot d \mid x \geq 0\}$ , where  $t$  is the least integer greater than  $n^2$  that is a multiple of  $d = \gcd(c_1, c_2, \dots, c_r)$ .*

A unary XNFA  $M = \langle Q, \{a\}, \delta, q_0, F \rangle$  is in *Chrobak normal form* if, for some  $m \geq 0$  and  $k \geq 0$ ,  $Q = \{q_i \mid 0 \leq i \leq m\} \cup C_1 \cup C_2 \cup \dots \cup C_k$ , where, for each  $1 \leq i \leq k$ ,  $C_i = \{p_{i,0}, p_{i,1}, \dots, p_{i,j_i-1}\}$  for some  $j_i \geq 1$ ,  $\delta(q_i, a) = \{q_{i+1}\}$  for  $0 \leq i \leq m-1$ , and for each  $1 \leq i \leq k$  and  $0 \leq h \leq j_i - 1$ ,  $\delta(p_{i,h}, a) = \{p_{i,(h+1) \bmod j_i}\}$ , and  $\delta(q_m, a) = \{p_{1,0}, p_{2,0}, \dots, p_{k,0}\}$ .

So, an XNFA is in Chrobak normal form if its structure is a deterministic tail from  $q_0$  to  $q_m$ , where the automaton makes only a single nondeterministic decision, which chooses one of the disjoint cycles  $C_i$ .

Next, we show how to convert a unary XNFA into Chrobak normal form. The idea of the construction is along the lines of the construction in [37] but with modifications with respect to the exclusiveness of the XNFA.

**Lemma 2.** *Let  $n \geq 1$ . For every unary  $n$ -state XNFA, an equivalent  $O(n^3)$ -state XNFA in Chrobak normal form can effectively be constructed, such that the sum of the cycle lengths is of order  $O(n)$ .*

*Proof.* Let  $M = \langle Q, \Sigma, \delta, q_0, F \rangle$  be an  $n$ -state XNFA. Since any unary language over some alphabet is completely determined by the lengths of the words in the language, we can safely disregard  $\Sigma$  and consider the state graph of  $M$  only. For  $L(M) = \emptyset$ , the theorem is trivial. So, in the sequel we assume that  $L(M)$  is not empty. Moreover, we may safely assume that all states  $q \in Q$  are reachable and productive, that is, there is a path from  $q_0$  to  $q$  and a path from  $q$  to a final state. Now, by adding states and possibly removing some states and transitions, we modify  $M$  such that there is no incoming transition to the initial state, such that  $F = \{q_+\}$  is a singleton, and such that  $q_+$  is the only state without outgoing transitions. To this end, all unreachable states together with their incoming and outgoing transitions are removed. Similarly, all unproductive states together with their incoming and outgoing transitions are removed as well. Next, if the initial state has incoming transitions, a new state without incoming transitions is added whose outgoing transitions go to the successor states of the initial state. This new state becomes the new initial state. In order to make  $F$  a singleton, we have to take care about words that are accepted on more than one path. So, first a new accepting state  $q_+$  is added. For each pair of old accepting states, if both states do not share a common predecessor state, from each of their predecessor states a transition to  $q_+$  is added. Both states become non-accepting. However, if both states have at least one common predecessor, say  $p$ , then there are two paths via  $p$  to accepting states. This means that inputs following these paths do not belong to  $L(M)$ . In this case, both states become non-accepting, some state  $p'$  is added, and all incoming transitions to  $p$  are doubled and are directed to  $p'$  as well. Furthermore, a transition from  $p$  to  $q_+$  and a transition from  $p'$  to  $q_+$  is added. Similarly, for all common predecessors of the old accepting states. In this way, we obtain an XNFA equivalent to  $M$  that has the desired properties. For convenience, we call it also  $M$ . The modified XNFA has at most  $m = 2n$  states.

From now on, we identify  $M$  with its state graph. Let  $S$  be the set of non-trivial strongly connected components of  $M$ . A *superpath* in  $M$  is a subgraph

$$\alpha = P_1 S_1 P_2 S_2 \cdots P_\ell S_\ell P_{\ell+1},$$

where, for  $1 \leq i \leq \ell$ ,  $S_i \in S$ ; for  $1 \leq i \leq \ell + 1$ ,  $P_i$  is a path in  $M$  whose inner nodes do not belong to non-trivial strongly component components of  $M$ ; the first node of  $P_1$  is  $q_0$ ; the last node of  $P_{\ell+1}$  is  $q_+$ ; for  $1 \leq i \leq \ell$ , the last node of  $P_i$  belongs to  $S_i$ ; for  $2 \leq i \leq \ell + 1$ , the first node of  $P_i$  belongs to  $S_{i-1}$ .

For every superpath  $\alpha$  in  $M$ , let  $L_\alpha$  be the set of all lengths of paths in  $M$  from  $q_0$  to  $q_+$  that are in  $\alpha$ . It follows that the length of any accepting path in  $M$  belongs to  $\bigcup_\alpha L_\alpha$ , where the union ranges over all superpaths in  $M$ .

We define the set  $\Psi_\alpha$  to be the subset of paths from  $q_0$  to  $q_+$  in  $\alpha$  that are simple, that is, no state appears twice. Clearly, the length of any path in  $\Psi_\alpha$  does not exceed  $m$ .

Next, we define  $\Pi_\alpha$  to be another subset of paths from  $q_0$  to  $q_+$  in  $\alpha$ . In particular, for every path  $\sigma$  in  $\Psi_\alpha$ , we put the following extensions  $\sigma'$  of  $\sigma$  into  $\Pi_\alpha$ . Whenever  $\sigma$  enters a strongly connected component  $S_i$  in some state  $v$ , then a Hamiltonian walk in  $S_i$  (that is, a tour that visits all nodes in  $S_i$ ) that cannot be shortened and that starts and ends in  $v$  is inserted into  $\sigma$ . Note that a Hamiltonian walk that cannot be shortened is a path from which no nodes can be removed without obtaining a path that is no longer Hamiltonian. It needs not to be the shortest Hamiltonian walk in  $S_i$ . Since  $S_i$  is strongly connected, such Hamiltonian walks exist. Results in [7] show that the lengths of such Hamiltonian walks in  $S_i$  do not exceed  $|S_i|^2$ , where  $|S_i|$  denotes the number of nodes in  $S_i$ . Therefore, the length of any path in  $\Pi_\alpha$  does not exceed  $m^2 + m$ .

Now we consider a fixed superpath  $\alpha$  in  $M$ . Let  $0 < c_1 < c_2 < \cdots < c_r \leq m$  be the lengths of all simple cycles in  $\alpha$ ,  $\sigma$  in  $\Psi_\alpha$ , and  $\sigma'$  be an extension of  $\sigma$  in  $\Pi_\alpha$ . Since  $\sigma'$  visits each node in  $\alpha$  at least once, the set  $Z_{\alpha, \sigma'}$  of all lengths  $z$  for which  $z = |\sigma'| + x_1 c_1 + x_2 c_2 + \cdots + x_r c_r$  is solvable in non-negative integers is contained in  $L_\alpha$ . By Lemma 1,  $Z_{\alpha, \sigma'} = X_\alpha \cup \{t_{\sigma'} + x \cdot d \mid x \geq 0\}$ , where  $X_\alpha$  contains lengths not larger than  $2m^2 + m$  and  $t_{\sigma'}$  is the least integer greater than  $2m^2 + m$  such that  $t_{\sigma'} \equiv |\sigma'| \pmod{d}$ , where  $d = \gcd(c_1, c_2, \dots, c_r)$ . Since the Hamiltonian walks in  $\sigma'$  are (compound) cycles, that is, linear combinations of  $c_1, c_2, \dots, c_r$ , the number  $d$  divides their lengths and, thus, we have  $t_{\sigma'} \equiv |\sigma'| \pmod{d}$ .

On the other hand, the set of all lengths  $y$  for which there is a  $\sigma$  in  $\Psi_\alpha$  such that

$$y = |\sigma| + x_1 c_1 + x_2 c_2 + \cdots + x_r c_r$$

is solvable in non-negative integers, clearly contains  $L_\alpha$ . Therefore, if  $w \in L_\alpha$  and  $w > 2m^2 + m$  then Lemma 1 implies that there is a  $\sigma$  in  $\Psi_\alpha$  such that  $w \equiv |\sigma| \pmod{d}$ . Since  $\{t_{\sigma'} + x \cdot d \mid x \geq 0\} \subseteq Z_{\alpha, \sigma'}$ , we conclude  $w \in Z_{\alpha, \sigma'}$ .

Altogether, we have  $L_\alpha = N_\alpha \cup \bigcup_{\sigma' \in \Pi_\alpha} \{t_{\sigma'} + x \cdot d \mid x \geq 0\}$ , where  $N_\alpha$  contains lengths not larger than  $2m^2 + m$ .

So far, we have created the prerequisites for constructing the normal form without specifically addressing XNFAs. So, the next task is to assemble an XNFA  $M' = \langle Q', \{a\}, \delta', q'_0, F' \rangle$  equivalent to  $M$  in Chrobak normal form.

To this end, we start with a deterministic tail consisting of the  $m^3 + 2$  states  $\{q'_i \mid 0 \leq i \leq m^3 + 1\}$  with  $\delta'(q'_i, a) = \{q'_{i+1}\}$ , for  $0 \leq i \leq m^3$ . A state  $q'_i$  of the tail becomes accepting if and only if the input of length  $i$  belongs to  $L(M)$ . So, all words whose length does not exceed  $m^3 + 1$  are correctly accepted or rejected.

Next, we want to add the cycles to the initial tail of  $M'$ .

To construct the cycles appropriately, we consider each superpath  $\alpha$  of  $M$  and distinguish three cases, respectively. As before, let  $0 < c_1 < c_2 < \dots < c_r \leq m$  be the lengths of all simple cycles in  $\alpha$  and  $d = \gcd(c_1, c_2, \dots, c_r)$ . We consider all inputs of lengths  $z > m^3 + 1 \in L_\alpha$ .

Case 1: There are at least two simple cycles  $C_1$  and  $C_2$  in  $\alpha$ . Then, each path of length  $z$  in  $\alpha$  that can be shortened to some path in  $\Pi_\alpha$  by deleting cycles, sees at least  $z - (m^2 + m)$  nodes in complete simple cycles of  $\alpha$ . If one of these paths contains at least two different cycles of the same length, then these cycles can replace each other and, thus, there are at least two accepting paths of length  $z$  in  $\alpha$ . Therefore, the input of length  $z$  does not belong to  $L(M)$ . Assume now that all cycles in these paths have different lengths. Then there are at most  $m$  cycles. Assume that each of these cycles is passed through at most  $m - 1$  times. Then,

$$\begin{aligned} z &\leq m^2 + m + \sum_{i=1}^m i(m-1) = m^2 + m + \frac{m^2 + m}{2}(m-1) \\ &= \frac{m^2 + m}{2}(m+1) = \frac{m^3 + 2m^2 + m}{2} \leq m^3 + 1 < z. \end{aligned}$$

From the contradiction we conclude that there is at least one cycle, say  $C_1$ , that is passed through for  $x_1 \geq m$  times. Let  $C_2$  be passed through for  $x_2$  times. We have  $x_1 \geq m \geq |C_2| \geq 1$  and  $|C_1| \geq 1$ . So,  $x_1|C_1| + x_2|C_2| = (x_1 - |C_2|)|C_1| + (x_2 + |C_1|)|C_2|$ . The equality means that passing  $x_1$  times through the cycle  $C_1$  and  $x_2$  times through the cycle  $C_2$  is equivalent to passing  $(x_1 - |C_2|)$  times through the cycle  $C_1$  and  $(x_2 + |C_1|)$  times through the cycle  $C_2$ . So, there are at least two accepting paths of length  $z$  in  $\alpha$ . Therefore, the input of length  $z$  does not belong to  $L(M)$ .

Case 2: There is exactly one simple cycle  $C_1$  in  $\alpha$ . So, there is at most one non-trivial strongly connected component in  $\alpha$  and this strongly connected component is the cycle  $C_1$ . Clearly, in this case we have  $d = |C_1|$  and the input length  $z$  is uniquely accepted along  $\alpha$ .

Case 3: There is no simple cycle in  $\alpha$ . In this case, there is no non-trivial strongly connected component in  $\alpha$  and the unique path of length  $z$  from the initial state ends in the initial tail and, by construction, the input of length  $z$  is correctly accepted or rejected.

Now we are ready to add the cycles for  $\alpha$  to the tail of  $M'$ . To this end, nothing has to be done for Case 3.

For the remaining cases, the cycle length must be  $d$ . If there is no cycle of length  $d$ , we add two disjoint cycles  $A_\alpha$  and  $R_\alpha$  each of length  $d$ . In particular,  $A_\alpha$  consists of states  $\{s_0, s_1, \dots, s_{d-1}\}$  with  $\delta'(s_h, a) = \{s_{(h+1) \bmod d}\}$ , and similarly,  $R_\alpha$  consists of states  $\{r_0, r_1, \dots, r_{d-1}\}$  with  $\delta'(r_h, a) = \{r_{(h+1) \bmod d}\}$ . The cycles are connected to the tail by the transitions  $\delta(q'_{m^3+1}, a) = \{s_0\}$  and  $\delta(q'_{m^3+1}, a) = \{r_0\}$ . If there are already two cycles  $A$  and  $R$  of length  $d$  that have already been constructed for some other superpath, then they are reused and nothing is added.

Next, we identify the accepting states on the cycles.

For Case 1, we consider each  $\sigma \in \Psi_\alpha$  and states  $s_i$  and  $r_i$  become accepting if  $m^3 + 1 + i + 1 \equiv |\sigma| \pmod{d}$ . In this way, Case 1 is treated correctly, since now two different paths in  $M'$  are accepting for the same length.

For case 2, we also consider each  $\sigma \in \Psi_\alpha$ . Here, only state  $s_i$  becomes accepting if  $m^3 + 1 + i + 1 \equiv |\sigma| \pmod{d}$ .

In this way, Case 2 is treated correctly, since only one path is made accepting. However, it may be that  $r_i$  was already accepting. This means that the corresponding inputs are also accepted by another superpath.

This concludes the construction of  $M'$ . Note, if an input is accepted by different superpaths having different cycle length, then it clearly does not belong to  $L(M')$ , but is also does not belong to  $L(M)$ .

Conversely, if an input is accepted unambiguously by  $M$  then it is accepted also unambiguously by  $M'$ . So, we conclude  $L(M) = L(M')$ . Moreover, since the sum of the different cycle lengths is at most  $m$  and each cycle length appears at most twice, the total sum of the cycle lengths is at most  $2m$ .  $\square$

Next, we can utilize the normal form to show that the costs for the determinization of *unary* XNFAs are the same (in the order of magnitude) as for NFAs. This is in strict contrast to XNFAs over a general alphabet. The backbone of the construction is similar to the backbone of the construction given in [2]. However, here we have to treat the cases when inputs are accepted at multiple paths.

**Theorem 3.** *Let  $n \geq 1$  and  $M$  be a unary  $n$ -state XNFA. Then  $e^{\Theta(\sqrt{n \cdot \ln n})}$  states are sufficient for a DFA to accept  $L(M)$ .*

*Proof.* Given a unary  $n$ -state XNFA  $M$ , we first construct an equivalent  $O(n^3)$ -state XNFA  $M'$  in Chrobak normal form as in the proof of Lemma 2. Let  $A_1, R_1, A_2, R_2, \dots, A_k, R_k$ , for  $k \geq 1$ , be the cycles of  $M'$ , where  $|A_i| = |R_i|$ , for  $1 \leq i \leq k$ . We construct the equivalent DFA  $M'' = \langle Q, \Sigma, \delta, q_0, F \rangle$  as follows.

First, we take over the initial deterministic tail of  $M'$ , which has the  $m^3 + 2$  states  $\{q'_i \mid 0 \leq i \leq m^3 + 1\}$ , where  $m = 2n$  as in the proof of Lemma 2. Then we add one big cycle of length  $\ell = \text{lcm}\{|A_1|, |A_2|, \dots, |A_k|\}$  to the tail. To this end the states from the set  $\{p_i \mid 0 \leq i \leq \ell - 1\}$  are cyclically connected and a transition from  $q_{m^3+1}$  to  $p_0$  is added.

Next, we have to identify the accepting states. To this end, all accepting states on the tail remain accepting. So, as for  $M'$  all words up to length  $m^3 + 1$  are treated correctly.

Then, we assume that each state  $p_i$  of the cycle has a counter attached that is initially set to 0. Now, we consider each cycle  $A_i$  of  $M'$  consisting of the states  $\{s_0, s_1, \dots, s_{d-1}\}$ . Whenever a state  $s_j$  is accepting, then the counters of all states  $\{p_t \mid t = j + x \cdot d, \text{ for } 0 \leq x \leq \frac{\ell}{d} - 1\}$  are increased by one. Similarly, for each cycle  $R_i$  of  $M'$  consisting of the states  $\{r_0, r_1, \dots, r_{d-1}\}$ . If a state  $r_j$  is accepting, then the counters of all states  $\{p_t \mid t = j + x \cdot d, \text{ for } 0 \leq x \leq \frac{\ell}{d} - 1\}$  are increased by one.

In a last construction step, all states whose counters are exactly one become accepting, all the others become non-accepting. In this way, all inputs that are accepted by more than one path in  $M'$  are rejected in  $M''$ , and all inputs that are accepted in  $M'$  and, thus, in  $M$  by exactly one path are accepted by  $M''$  as well. So,  $L(M) = L(M'')$  and, clearly,  $M''$  is a DFA. Moreover,  $M''$  has at most

$$m^3 + 2 + \ell \leq (2n)^3 + 2 + \ell \leq (2n)^3 + 2 + F(n) \in e^{\Theta(\sqrt{n \cdot \ln n})}$$

many states.  $\square$

It will turn out after Proposition 5 that the upper bound for the determinization in Theorem 3 is tight in the order of magnitude.

## 4 Converting unary NFAs to XNFAs and Vice Versa

Here, again Landau's function

$$F(n) = \max\{\text{lcm}(c_1, c_2, \dots, c_l) \mid l \geq 1, c_1, c_2, \dots, c_l \geq 1, c_1 + c_2 + \dots + c_l = n\}$$

plays a crucial role. Recall that the  $c_i$  always can be chosen to be relatively prime such that  $c_1, c_2, \dots, c_l \geq 2$ ,  $c_1 + c_2 + \dots + c_l \leq n$ , and  $\text{lcm}(c_1, c_2, \dots, c_l) = F(n)$ . This, for example, means that the  $c_i$  can be prime powers. An interesting and simplifying result in [22] revealed that, instead of prime powers, one can sum up the first prime numbers such that the sum does not exceed the limit  $n$ . More,

precisely, it has been shown in [22] that the following function  $G(n)$  is of the same order of magnitude as  $F(n)$ , that is,  $G(n) \in \Theta(F(n))$ . Let  $p_i$  denote here the  $i$ th prime number with  $p_1 = 2$ .

$$G(n) = \max\{p_1 \cdot p_2 \cdots p_l \mid l \geq 1 \text{ and } p_1 + p_2 + \cdots + p_l \leq n\}$$

In the following theorem we use the function  $G(n)$  to describe the worst case state costs of an NFA simulating a unary XNFA.

**Theorem 4.** *Let  $n \geq 2$ . There exists a unary  $(n+1)$ -state XNFA  $M$  such that every NFA in Chrobak normal form accepting  $L(M)$  has at least  $G(n)$  states.*

*Proof.* For  $n \geq 2$ , let  $G(n)$  be represented by the product  $p_1 \cdot p_2 \cdots p_l$  of the first  $l \geq 1$  prime numbers. We consider the XNFA  $M = \langle Q, \{a\}, \delta, q_0, F \rangle$  whose state graph has  $l$  disjoint cycles. Each cycle  $1 \leq i \leq l$  has length  $p_i$  and consists of the states  $\{r_{i,0}, r_{i,1}, \dots, r_{i,p_i-1}\}$ , where  $\delta(r_{i,h}, a) = \{r_{i,(h+1) \bmod p_i}\}$ , for  $0 \leq h \leq p_i - 1$ . Now, the initial state  $q_0$  is nondeterministically connected to the cycles by  $\delta(q_0, a) = \{r_{1,1}, r_{2,1}, \dots, r_{l,1}\}$ . The set of accepting states is  $F = \{r_{i,0} \mid 1 \leq i \leq l\}$ . By construction,  $M$  has at most  $n+1$  states.

The language  $L(M)$  accepted by  $M$  is

$$\{a^m \mid \text{there is exactly one } i \in \{1, 2, \dots, \ell\} \text{ such that } m \equiv 0 \pmod{p_i}\}.$$

We define the set of all integers that are not divisible by all  $p_i$ ,  $1 \leq i \leq l$ , as

$$K = \{k \in \mathbb{N} \mid k \text{ is not divisible by all } p_i, 1 \leq i \leq l\}.$$

Assume now, that  $L(M)$  is accepted by an NFA  $M'$  in Chrobak normal form with less than  $G(n)$  states, say  $m < G(n)$  states.

Our first goal is to show the claim that for any  $p_i$ ,  $1 \leq i \leq l$ , all cycles in the state graph of  $M'$  on which infinitely many words from  $\{a^{x \cdot p_i} \mid x \in K\}$  are accepted, have a length that is divisible by  $p_i$ .

Since all words from the infinite set  $\{a^{x \cdot p_i} \mid x \in K\}$  belong to  $L(M)$ , cycles on which infinitely many such words are accepted exist. Assume that one of these cycles has a length  $c$  not divisible by  $p_i$  and let  $a^{x_0 \cdot p_i}$  with  $x_0 \in K$  be one of the accepted words. Then, the word  $w = a^{x_0 \cdot p_i + c \cdot p}$  with  $p = \frac{G(n)}{p_i}$  is accepted as well. But since  $c$  and  $p$  are not divisible by  $p_i$ , we have that  $|w|$  is not divisible by  $p_i$ , either. Moreover, since  $x_0 \cdot p_i$  is not divisible by any  $p_j$  with  $i \neq j$  but  $c \cdot p$  is, we have that  $|w|$  is not divisible by any  $p_j$  with  $i \neq j$ , either. So,  $w$  cannot belong to  $L(M')$ . From this contradiction the claim follows.

Since  $m < G(n)$ , there must be two cycles  $C_1$  and  $C_2$ , say of length  $c_1$  and  $c_2$ , such that there are two different prime numbers  $p_i \neq p_j$  with  $1 \leq i, j \leq l$ , where  $c_1$  is divisible by  $p_i$  but not divisible by  $p_j$  and infinitely many words from  $\{a^{x \cdot p_i} \mid x \in K\}$  are accepted in  $C_1$ , and where  $c_2$  is divisible by  $p_j$  but not divisible by  $p_i$  and infinitely many words from  $\{a^{x \cdot p_j} \mid x \in K\}$  are accepted in  $C_2$ . Since  $p_j$  is relatively prime to  $c_1$ , there is an integer  $p$  such that  $p \cdot c_1 \equiv 1 \pmod{p_j}$ . Consider some word  $w = a^{x_0 \cdot p_i}$  with  $x_0 \in K$  that is accepted in  $C_1$ . Then, the word  $a^{x_1 \cdot p \cdot c_1 + |w|}$  with  $(x_1 + |w|) \equiv 0 \pmod{p_j}$  is accepted in  $C_1$  as well. However, this word does not belong to  $L(M)$ , since it is divisible by  $p_i$  and  $p_j$ .

So, from this contradiction we conclude there is no NFA in Chrobak normal form with less than  $G(n)$  states.  $\square$

Clearly the upper bound for the simulation of an XNFA by an NFA is given by determinization. Thus, we have the following proposition.

**Proposition 5.** *Let  $n \geq 2$  and  $M$  be a unary  $n$ -state XNFA. Then  $e^{\Theta(\sqrt{n \cdot \ln n})}$  states are sufficient for an NFA to accept  $L(M)$ .*



The lower bound in Theorem 4 says that there are  $(n + 1)$ -state XNFAs such that any equivalent NFA in Chrobak normal form has at least  $G(n)$  states. Moreover, any  $n$ -state NFA can be converted into an equivalent NFA in Chrobak normal form that has at most  $O(n^2)$  states. So, since  $G(n) \in \Theta(F(n))$  [22], the lower bound for the state costs of the simulation of an  $n$ -state XNFA by an NFA (not necessarily in Chrobak normal form) is

$$\Theta(\sqrt{G(n-1)}) = \Theta(\sqrt{e^{\Theta(\sqrt{(n-1) \cdot \ln(n-1)})}}) = e^{\Theta(\sqrt{n \cdot \ln n})}.$$

So, we conclude that the upper bound for the unary XNFA-to-DFA conversion shown in Theorem 3 and the upper bound for the unary XNFA-to-NFA conversion shown in Proposition 5 are tight in the order of magnitude.

We turn to the simulation of NFAs by XNFAs. In [25] it has been shown that the language

$$L = \{a^n \mid n \not\equiv 0 \pmod{\text{lcm}(c_1, c_2, \dots, c_k)}\} \cup \{\lambda\},$$

for  $k \geq 1$  and  $c_1, c_2, \dots, c_k \geq 2$  is accepted by an NFA with  $1 + \sum_{i=1}^k c_i$  states, while the smallest UFA for  $L$  needs at least  $1 + \text{lcm}(c_1, c_2, \dots, c_k)$  many states. The proof of the lower bound is based on a method given in [32] which is based on a rank argument on certain matrices. After a thorough analysis of the arguments of the method, it turned out that exclusively accepting computations of the UFAs are used. In other words, the arguments can be applied to XNFAs as well. So, we derive that also the smallest XNFA needs at least  $1 + \text{lcm}(c_1, c_2, \dots, c_k)$  states to accept the language  $L$ . So, we have the following lower bound.

**Theorem 6.** *Let  $n \geq 2$ . There exists a unary  $(n + 1)$ -state NFA  $M$  such that every XNFA accepting  $L(M)$  has at least  $F(n) + 1$  states.*

Clearly the upper bound for the simulation of an NFA by an XNFA is given by determinization. Thus, we have the following proposition.

**Proposition 7.** *Let  $n \geq 2$  and  $M$  be a unary  $n$ -state NFA. Then  $e^{\Theta(\sqrt{n \cdot \ln n})}$  states are sufficient for an XNFA to accept  $L(M)$ .*

As before, we also conclude here that the lower bound and upper bound are tight in the order of magnitude.

## 5 Computational Complexity

In this section, we discuss the computational complexity of decidability questions. In particular, we consider general membership, emptiness, universality, inclusion, and equivalence with respect to the unary case. These problems have been studied in [13, 14] in case of general alphabets. It turns out here that the general membership problem in the unary case shares the same computational complexity with the general case, namely, both problems are NL-complete. However, the questions of emptiness, universality, inclusion, and equivalence turn out to be coNP-complete in the unary case, whereas these questions have been shown to be PSPACE-complete in the general case [13, 14].

**Theorem 8.** *The problem of testing the general membership for unary XNFAs is NL-complete.*

*Proof.* To show that the problem is in NL for unary XNFAs we can use the same construction that has been described in [13, 14] for general alphabets. The basic idea is to test whether an input  $w$  is not

accepted by a given XNFA  $A$ . This means that either there is no accepting path in the computation tree for  $w$  or there are at least two accepting paths. In the first case, the input  $w$  is not accepted by  $A$  even if  $A$  is considered as an NFA. Hence, this case can be solved in NL using the known algorithms for NFAs. The second case can be checked by guessing two different accepting paths in the computation tree. To this end, one has to keep track of two states representing the current position on the two paths. Since this can be realized in NL, the general membership problem is in NL in particular for unary XNFAs.

To show the NL-hardness of the general membership problem for unary XNFAs we can in principle apply the reduction that is described in [14] for general alphabets. To adapt it to the unary case we have to use the fact that the membership problem for unary NFAs remains NL-complete (see, e.g., [11]) and we have to observe that the XNFA constructed in the reduction is unary, since the given NFA is unary. Since the reduction described in [14] is not yet published we provide the reduction here for the sake of completeness.

To show the NL-hardness of the general membership problem we reduce the non-membership problem for NFAs which is known to be NL-complete, since the membership problem for NFAs is NL-complete.

Let  $\langle A, w \rangle$  be the encoding of an NFA  $A = \langle Q, \{a\}, \delta, q_0, F \rangle$  and an input word  $w$ . We construct an XNFA  $A' = \langle Q \cup \{p_0, p\}, \{a\}, \delta', p_0, F' \rangle$ , where  $p_0$  and  $p$  are two new states not belonging to  $Q$ . The accepting states  $F'$  are defined as  $F' = F \cup \{p_0, p\}$ , if  $\lambda \in L(A)$ , and  $F' = F \cup \{p\}$  otherwise. The transition function  $\delta'$  is defined as follows. First,  $A'$  has the same behavior as  $A$  on states from  $Q$ . Formally,  $\delta'(q, a) = \delta(q, a)$  for all  $q \in Q$ . Second, from the new initial state  $p_0$  all states are reached that are reached from the initial state  $q_0$  of  $A$ . Additionally, the new state  $p$  is reached from  $p_0$ . Formally,  $q' \in \delta'(p_0, a)$ , if  $q' \in \delta(q_0, a)$ , and  $p \in \delta'(p_0, a)$ . Finally, the state  $p$  acts as an accepting sink state, that is,  $p \in \delta'(p, a)$ .

The reduction from the encoding  $\langle A, w \rangle$  to an encoding  $\langle A', w \rangle$  can be realized by a deterministic logarithmically space-bounded Turing machine.

For the correctness of the reduction we have to show that the XNFA  $A'$  accepts  $w$  if and only if  $w$  is not accepted by the NFA  $A$ . On the one hand, if  $w$  is accepted by  $A'$ , then  $p \in \delta'(p_0, w)$  and  $\delta'(p_0, w) \cap F = \emptyset$ , since otherwise there would be at least two accepting paths for  $w$ . Hence,  $w$  is not accepted by the NFA  $A$ . On the other hand, if  $w$  is not accepted by  $A'$ , then  $p \in \delta'(p_0, w)$  and  $\delta'(p_0, w) \cap F \neq \emptyset$ , since there must be at least two accepting paths for  $w$ . Hence,  $w$  is accepted by the NFA  $A$ . This concludes the correctness of the reduction and shows the NL-hardness of the general membership problem for XNFAs. Altogether, we obtain that the general membership problem for XNFAs is NL-complete.  $\square$

It is known that the emptiness problem for unary NFAs is NL-complete. In contrast, we show the problem becomes coNP-complete for unary XNFAs. In the following proofs we need a result obtained in [13, 14] on the conversion of XNFAs to DFAs in case of general alphabets.

**Theorem 9.** [13, 14] *Let  $n \geq 1$  and  $M$  be an  $n$ -state XNFA. Then  $3^n - 2^n + 1$  states are sufficient for a DFA to accept  $L(M)$ .*

**Theorem 10.** *The emptiness problem for unary XNFAs is coNP-complete.*

*Proof.* We will show that the non-emptiness problem for unary XNFAs is NP-complete which implies that the emptiness problem is coNP-complete. To show that the non-emptiness problem belongs to NP we use a similar approach as described in Theorem 6.1 in [35]. Let  $M$  be an XNFA over a unary alphabet  $\{a\}$  with state set  $Q = \{q_1, q_2, \dots, q_n\}$ , initial state  $q_1$ , and transition function  $\delta$ . By applying Theorem 9 we know that there exists an equivalent DFA that has at most  $3^n$  states. It is clear that  $L(M)$  is not empty if and only if  $M$  accepts a word of length  $m \leq 3^n$ .

Now, the idea is first to guess a length  $m \leq 3^n$  in ternary representation  $m_1 m_2 \cdots m_n$  and to check whether there is exactly one path of length  $m$  in  $M$  leading from the initial state to an accepting state. The latter can be realized by mapping the transition function of  $M$  to its corresponding adjacency matrix  $A_M$  where we set an entry  $A_M[i, j] = 1$  if and only if  $q_j \in \delta(q_i, a)$ , for  $1 \leq i, j \leq n$ . Then,  $a^m \in L(M)$  if and only if the first row of  $A_M^m$  has exactly one entry corresponding to an accepting state with value 1. Thus, we have as second task to compute the matrix product  $A_M^m = A_M^{m_1 \cdot 3^{n-1}} \cdot A_M^{m_2 \cdot 3^{n-2}} \cdots A_M^{m_n}$  by inspecting the ternary counter. The matrix  $A_M^m$  can be computed by successively cubing and multiplying  $A_M$ . For example, let  $m = 22$  and its ternary notation be 211. Then, we have to multiply  $A_M \cdot A_M^3 \cdot A_M^9 \cdot A_M^9$ . In general, we have at most  $3 \cdot 2 \log_3(m) \leq 6n$  matrix multiplications. Since every matrix multiplication can be realized in time  $n^2$ , we obtain that  $A_M^m$  can be computed in deterministic time bounded by a polynomial in  $n$ . Finally, the first row of the resulting matrix  $A_M^m$  has to be inspected. Altogether, these three tasks can be realized in nondeterministic time bounded by a polynomial in  $n$ . Hence, the complete procedure is in NP.

To show that the non-emptiness problem is NP-hard we use again a similar approach as described in Theorem 6.1 in [35]. It is shown there that a given Boolean formula in conjunctive form with exactly three literals per conjunct is satisfiable if and only if a regular unary language  $L$  described by a regular expression is not equal to  $\{a\}^*$ . Moreover, the reduction is computable in logarithmic space. Since a language described by a regular expression can equivalently be described by an NFA of similar size, we let now  $L$  be described by an NFA  $M$ . Moreover, we construct a one-state DFA  $M'$  that accepts  $\{a\}^*$ . Then, we construct an XNFA  $M''$  that initially guesses whether it simulates for the complete input the NFA  $M$  or the DFA  $M'$ . Since  $M''$  is an XNFA we obtain that  $L(M) = \{a\}^*$  if and only if  $L(M'') = \emptyset$ . Hence, we have  $L(M'') \neq \emptyset$  if and only if  $L(M) \neq \{a\}^*$  if and only if the given Boolean formula is satisfiable. Since the constructions of  $M$ ,  $M'$ , and  $M''$  can be realized in logarithmic space, we obtain the NP-hardness of the non-emptiness problem for XNFAs and, thus, the coNP-hardness of the emptiness problem for XNFAs.  $\square$

**Theorem 11.** *The problems of testing universality, inclusion, and equivalence for unary XNFAs are coNP-complete.*

*Proof.* Let us first show that the problems of testing non-universality, non-inclusion, and non-equivalence for unary XNFAs are in NP. We start with the non-universality problem. Let  $M$  be an  $n$ -state XNFA. By applying Theorem 9 we know that there exists an equivalent DFA that has at most  $3^n$  states. Hence,  $L(M) \neq \{a\}^*$  if and only if there is a word of length  $m \leq 3^n$  that is not accepted by  $M$ . Similar to the proof of Theorem 10 we can guess a ternary representation of that word, compute  $A_M^m$ , and check that the guessed word is not accepted by  $M$  by inspecting the first row whether there is no entry corresponding to an accepting state with value 1. According to the considerations made in the proof of Theorem 10 the procedure can be realized in nondeterministic polynomial time and we obtain that the non-universality problem is in NP. Hence, the universality problem is in coNP.

Next, we consider the non-inclusion problem. Let  $M_1$  be an  $n_1$ -state XNFA and  $M_2$  be an  $n_2$ -state XNFA. By applying Theorem 9 we know that there exist equivalent DFAs having at most  $3^{n_1}$  states and  $3^{n_2}$  states, respectively. Hence,  $L(M_1) \not\subseteq L(M_2)$  if and only if  $L(M_1) \cap \overline{L(M_2)} \neq \emptyset$  if and only if there is a word of length  $m \leq 3^{n_1+n_2}$  that is accepted by  $M_1$ , but not accepted by  $M_2$ . Similar to the proof of Theorem 10 and to the above construction for the non-universality problem we obtain that the non-inclusion problem is in NP. Hence, the inclusion problem is in coNP.

Finally, we consider the equivalence problem. Let  $M_1$  and  $M_2$  be two XNFAs. Since the inclusion problem is in coNP, we obtain that the equivalence problem is coNP by testing  $L(M_1) \subseteq L(M_2)$  and

$L(M_2) \subseteq L(M_1)$ .

To show the coNP-hardness of the problems we shortly describe how the reduction given in the proof of Theorem 10 has to be extended. We recall that we have constructed an XNFA  $M''$  such that  $L(M'') \neq \emptyset$  if and only if the given Boolean formula is satisfiable.

For non-universality we construct another XNFA  $A$  that initially guesses whether it simulates for the complete input the XNFA  $M''$  or the DFA  $M'$  accepting  $\{a\}^*$ . Then, we have  $L(A) \neq \{a\}^*$  if and only if  $L(M'') \neq \emptyset$  and obtain the NP-hardness of non-universality. For the equivalence problem we consider  $M'$  as an XNFA and have  $L(A) = L(M') = \{a\}^*$  if and only if  $L(M'') = \emptyset$ , which gives the coNP-hardness of the equivalence problem. Finally, we have  $L(M') \subseteq L(A)$  if and only if  $L(A) = L(M')$  if and only if  $L(M'') = \emptyset$  and obtain the coNP-hardness of the inclusion problem.  $\square$

The computational complexity results in the unary case are summarized in Table 1.

	DFA	NFA	XNFA	AFA
membership	L	NL	NL	P
emptiness	L	NL	coNP	PSPACE
universality	L	coNP	coNP	PSPACE
inclusion	L	coNP	coNP	PSPACE
equivalence	L	coNP	coNP	PSPACE

Table 1: Computational complexity results for the decidability problems in the unary case. All problems are complete with respect to the complexity class indicated. The results for XNFAs are obtained in this paper. The remaining results and pointers to the literature are summarized, for example, in the survey [10].

## References

- [1] Andreas Blass & Yuri Gurevich (1982): *On the Unique Satisfiability Problem*. *Inform. Control* 55, pp. 80–88, doi:10.1016/S0019-9958(82)90439-9.
- [2] Marek Chrobak (1986): *Finite automata and unary languages*. *Theor. Comput. Sci.* 47, pp. 149–158, doi:10.1016/0304-3975(86)90142-8. Errata: [3].
- [3] Marek Chrobak (2003): *Errata to “Finite automata and unary languages”*. *Theor. Comput. Sci.* 302, pp. 497–498, doi:10.1016/S0304-3975(03)00136-1.
- [4] Keith Ellul (2004): *Descriptive Complexity Measures of Regular Languages*. Master’s thesis, University of Waterloo, Ontario, Canada.
- [5] Viliam Geffert (2007): *Magic numbers in the state hierarchy of finite automata*. *Inform. Comput.* 205(11), pp. 1652–1670, doi:10.1016/j.ic.2007.07.001.
- [6] Viliam Geffert, Carlo Mereghetti & Giovanni Pighizzini (2003): *Converting two-way nondeterministic unary automata into simpler automata*. *Theor. Comput. Sci.* 295, pp. 189–203, doi:10.1016/S0304-3975(02)00403-6.
- [7] Yahya Ould Hamidoune (1979): *Sur les parcours hamiltoniens dans les graphes orientés*. *Discrete Mathematics* 26, pp. 227–234, doi:10.1016/0012-365X(79)90028-1.
- [8] Lane A. Hemaspaandra & Mitsunori Ogihara (2002): *The Complexity Theory Companion*. Springer, doi:10.1007/978-3-662-04880-1.

- [9] Markus Holzer & Martin Kutrib (2003): *Unary Language Operations and Their Nondeterministic State Complexity*. In M. Ito & M. Toyama, editors: *Developments in Language Theory (DLT 2002)*, LNCS 2450, Springer, pp. 162–172, doi:10.1007/3-540-45005-X\_14.
- [10] Markus Holzer & Martin Kutrib (2011): *Descriptive and Computational Complexity of Finite Automata – A Survey*. *Inform. Comput.* 209, pp. 456–470, doi:10.1016/J.IC.2010.11.013.
- [11] Neil D. Jones (1975): *Space-Bounded Reducibility among Combinatorial Problems*. *J. Comput. Syst. Sci.* 11, pp. 68–85, doi:10.1016/S0022-0000(75)80050-X.
- [12] Michal Kunc & Alexander Okhotin (2012): *State complexity of operations on two-way finite automata over a unary alphabet*. *Theor. Comput. Sci.* 449, pp. 106–118, doi:10.1016/J.TCS.2012.04.010.
- [13] Martin Kutrib, Andreas Malcher & Matthias Wendlandt (2023): *Complexity of Exclusive Nondeterministic Finite Automata*. In Henning Bordihn, Nicholas Tran & György Vaszil, editors: *Descriptive Complexity of Formal Systems (DCFS 2023)*, LNCS 13918, Springer, pp. 121–133, doi:10.1007/978-3-031-34326-1\_9.
- [14] Martin Kutrib, Andreas Malcher & Matthias Wendlandt (2024): *Complexity of Exclusive Nondeterministic Finite Automata*. submitted for journal publication.
- [15] Edmund Landau (1903): *Über die Maximalordnung der Permutationen gegebenen Grades*. *Archiv der Math. und Phys.* 3, pp. 92–103.
- [16] Edmund Landau (1909): *Handbuch der Lehre von der Verteilung der Primzahlen*. Teubner, Leipzig.
- [17] Hing Leung (1998): *Separating Exponentially Ambiguous Finite Automata from Polynomially Ambiguous Finite Automata*. *SIAM J. Comput.* 27, pp. 1073–1082, doi:10.1137/S0097539793252092.
- [18] Hing Leung (2005): *Descriptive complexity of NFA of different ambiguity*. *Int. J. Found. Comput. Sci.* 16, pp. 975–984, doi:10.1142/S0129054105003418.
- [19] Filippo Mera & Giovanni Pighizzini (2005): *Complementing unary nondeterministic automata*. *Theor. Comput. Sci.* 330, pp. 349–360, doi:10.1016/J.TCS.2004.04.015.
- [20] Carlo Mereghetti & Giovanni Pighizzini (2001): *Optimal Simulations between Unary Automata*. *SIAM J. Comput.* 30, pp. 1976–1992, doi:10.1137/S009753979935431X.
- [21] Albert R. Meyer & Michael J. Fischer (1971): *Economy of Description by Automata, Grammars, and Formal Systems*. In: *Symposium on Switching and Automata Theory (SWAT 1971)*, IEEE, pp. 188–191, doi:10.1109/SWAT.1971.11.
- [22] William Miller (1987): *The maximum order of an element of a finite symmetric group*. *Am. Math. Mon.* 94, pp. 497–506, doi:10.1080/00029890.1987.12000673.
- [23] Frank R. Moore (1971): *On the Bounds for State-Set Size in the Proofs of Equivalence Between Deterministic, Nondeterministic, and Two-Way Finite Automata*. *IEEE Trans. Comput.* 20(10), pp. 1211–1214, doi:10.1109/T-C.1971.223108.
- [24] J.-L. Nicolas (1968): *Sur l'ordre maximum d'un élément dans le groupe  $S_n$  des permutations*. *Acta Arith.* 14, pp. 315–332, doi:10.4064/aa-14-3-315-332.
- [25] Alexander Okhotin (2012): *Unambiguous finite automata over a unary alphabet*. *Inform. Comput.* 212, pp. 15–36, doi:10.1016/J.IC.2012.01.003.
- [26] Giovanni Pighizzini (2009): *Deterministic Pushdown Automata and Unary Languages*. *Int. J. Found. Comput. Sci.* 20(4), pp. 629–645, doi:10.1142/S0129054109006784.
- [27] Giovanni Pighizzini (2015): *Investigations on Automata and Languages Over a Unary Alphabet*. *Int. J. Found. Comput. Sci.* 26, pp. 827–850, doi:10.1142/S012905411540002X.
- [28] Giovanni Pighizzini & Jeffrey Shallit (2002): *Unary Language Operations, State Complexity and Jacobsthal's Function*. *Int. J. Found. Comput. Sci.* 13, pp. 145–159, doi:10.1142/S012905410200100X.
- [29] Giovanni Pighizzini, Jeffrey Shallit & Ming-Wei Wang (2002): *Unary Context-Free Grammars and Pushdown Automata, Descriptive Complexity and Auxiliary Space Lower Bounds*. *J. Comput. Syst. Sci.* 65, pp. 393–414, doi:10.1006/JCSS.2002.1855.

- [30] Michael Oser Rabin & Dana Scott (1959): *Finite Automata and Their Decision Problems*. *IBM J. Res. Dev.* 3, pp. 114–125, doi:10.1147/rd.32.0114.
- [31] William J. Sakoda & Michael Sipser (1978): *Nondeterminism and the size of two way finite automata*. In ACM, editor: *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing (STOC 1978)*, ACM, ACM Press, New York, pp. 275–286, doi:10.1145/800133.804357.
- [32] Erik Meineche Schmidt (1978): *Succinctness of Descriptions of Context-Free, Regular and Finite Languages*. Ph.D. thesis, Cornell University, Ithaca, NY.
- [33] Jeffrey Shallit (2008): *The Frobenius Problem and Its Generalizations*. In Masami Ito & Masafumi Toyama, editors: *Developments in Language Theory (DLT 2008)*, LNCS 5257, Springer, pp. 72–83, doi:10.1007/978-3-540-85780-8\_5.
- [34] Michael Sipser (1980): *Lower Bounds on the Size of Sweeping Automata*. *J. Comput. Syst. Sci.* 21, pp. 195–202, doi:10.1016/0022-0000(80)90034-3.
- [35] Larry J. Stockmeyer & A. R. Meyer (1973): *Word Problems Requiring Exponential Time*. In ACM, editor: *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing (STOC 1973)*, ACM Press, New York, NY, USA, pp. 1–9, doi:10.1145/800125.804029.
- [36] M. Szalay (1980): *On the maximal order in  $S_n$  and  $S_n^*$* . *Acta Arithm.* 37, pp. 321–331, doi:10.4064/aa-37-1-321-331.
- [37] Anthony Widjaja To (2009): *Unary finite automata vs. arithmetic progressions*. *Inform. Process. Lett.* 109, pp. 1010–1014, doi:10.1016/J.IPL.2009.06.005.