A Superpolynomial Lower Bound for the Size of Non-Deterministic Complement of an Unambiguous Automaton

Mikhail Raskin[1]
LaBRI, University of Bordeaux, 351, cours de la Libération F-33405 Talence cedex, France
raskin@mccme.ru
https://orcid.org/0000-0002-6660-5673

Abstract

Unambiguous non-deterministic finite automata (UFA) are non-deterministic automata (over finite words) such that there is at most one accepting run over each input. Such automata are known to be potentially exponentially more succinct than deterministic automata, and non-deterministic automata can be exponentially more succinct than them.

In this paper we establish a superpolynomial lower bound for the state complexity of the translation of an UFA to a non-deterministic automaton for the complement language. This disproves the formerly conjectured polynomial upper bound for this translation. This lower bound only involves a one letter alphabet, and makes use of the random graph methods.

The same proof also shows that the translation of sweeping automata to non-deterministic automata is superpolynomial.

2012 ACM Subject Classification Theory of computation → Models of computation

Keywords and phrases unambiguous automata, language complement, lower bound

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.138

Acknowledgements I would like to thank Gabriele Puppis for numerous useful discussions. I would like to thank anonymous reviewers, Thomas Colcombet and Bruno Courcelle for their advice regarding presentation.

1 Introduction

In many areas of computer science, the relationship between deterministic and non-deterministic devices is a subject of significant interest. An intermediate notion between deterministic and non-deterministic computation devices is the notion of unambiguous device. Such a device can make non-deterministic choices, but it is guaranteed that for every input there is at most one accepting execution trace.

For finite automata it is known that non-deterministic automata can be exponentially more succinct than deterministic automata [10]. It is also known that unambiguous automata can be exponentially more succinct than deterministic automata and in other situations they can be exponentially less succinct than non-deterministic automata [8]. The paper establishing exponential separation also defines several automata classes of limited ambiguity and provides exponential separation between some of them.

Other notions of unambiguity have been considered. Some of them (for example, structural unambiguity [9]: for all input words u and all states p, there is at most one run of the

[1] This work was supported by the French National Research Agency (ANR project GraphEn / ANR-15-CE40-0009)

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 138; pp. 138:1–138:11
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

automaton over u starting in an initial state and ending in p) describe a wider class of automata than unambiguity. Some are more restrictive than simple unambiguity (for example, strong unambiguity [14]: there is a set of result states, for every input there is exactly one way to reach a result state, and the result states can be accepting or rejecting). We do not consider these notions in the present paper.

We study the problem of representing a complement of a language specified by a finite automaton. It is easy to see that replacing the set of accepting states with its complement allows to recognize the complement of a language specified by a deterministic finite automaton without increasing the number of states. Complementing a language specified by a non-deterministic finite automaton may require an exponential number of states [1].

It has been conjectured (see for instance [3]) that every unambiguous non-deterministic one-way finite automaton (*UFA*) recognizing some language L can be converted into an *UFA* recognizing the complement of the original language L with polynomial increase in the number of states. The best known lower bound was quadratic [13], while the upper bounds were exponential [6]. The quadratic lower bound holds even for the single-letter alphabet. One of the arguments in favour of the conjecture was the fact that universality and even containment of the languages recognized by unambiguous finite automata can be decided in polynomial time [15].

The case of the single-letter alphabet has a better upper bound for the state complexity of recognizing the complement of the language of a non-deterministic finite automaton. A one-way non-deterministic finite automaton (*NFA*) with n states can be converted to a one-way deterministic finite automaton (*DFA*) with $e^{\Theta(\sqrt{n \log n})}$ states accepting the same language [11]. As a *DFA* can be converted into a *DFA* for the complement of the language without any increase in the number of states, this conversion provides an upper bound on the state complexity of recognizing the complement of the language recognized by an *NFA*. This upper bound is tight [12].

Recognizing the complement of the language of a two-way non-deterministic automaton (*2NFA*) with n states over the single-letter alphabet can be done using an *2NFA* with at most $O(n^8)$ states [4]. The same paper also shows that recognising the complement of the language of a *2DFA* with n states can be done by a $4n$-states *2DFA* for arbitrary alphabet. For the single-letter alphabet the complement of the language of a *2DFA* can be recognized by a *2DFA* with $2n+3$ states [7].

In the present paper we show a superpolynomial lower bound for the state complexity of recognizing the complement of a language of an unambiguous finite automaton by a non-deterministic finite automaton.

▶ **Theorem 1.** *There exists a sequence of unary UFA $(A_d)_{d\in\mathbb{N}}$ such that every NFA recognising the complement of the language of A_d has size at least $|A_d|^d$.*

The size of A_d is $2^{2^{2^{d^{\Theta(1)}}}}$.

▶ **Corollary 2.** *Worst case, complementing an UFA of size z by an NFA may require more than $z^{(\log\log\log z)^{\Theta(1)}}$ states.*

In other words, complementing an *UFA* requires more than polynomial increase in size regardless of the size of the alphabet, and the bound holds even if we allow the complement to be represented by *NFA*.

We also note that the same languages (and their complements) can be recognized by sweeping deterministic finite automata with a small increase in the state complexity compared to the case of *UFA*.

The proof revolves around a connection between *UFA* and tournaments (orientations of complete graphs), and an observation about existence of tournaments with special properties described in Section 4.

The rest of the paper is structured as follows. In the next section we give the standard definitions. Then we present in Section 3 our construction of the unambiguous automata A_d. It involves the use of tournaments with special properties, and the choice of many relatively close primes. We prove the existence of the suitable tournaments in Section 4, and explain how to chose the prime numbers in Section 5. The Section 6 finishes the proof of Theorem 1. We briefly study the case of sweeping automata in Section 7. In the final section we summarize the results and outline some possible future directions.

2 Definitions

In this section we will remind the definitions of deterministic, unambiguous and non-deterministic finite automata, and their normal forms.

▶ **Definition 3.** An *non-deterministic finite automaton* (*NFA*) is defined by an *alphabet* Σ, a set of *states* Q, a subset of *initial states* $I \subseteq Q$, a subset of *accepting states* $F \subseteq Q$ and the *transition relation* $T \subseteq Q \times \Sigma \times Q$. The *size* of an *NFA* A is the number of its states, and is denoted by $|A|$. A *run* of an *NFA* over a word $u = a_1 \dots a_n$ is a sequence of states $q_0, \dots, q_n$ such that (q_{i-1}, a_i, q_i) belongs to T for all $i = 1 \dots n$ and $q_0 \in I$. The run is *accepting* if its last state is accepting, i.e. $q_n \in F$. A *language* L over alphabet Σ is an subset $L \subseteq \Sigma^*$. The *language recognized by* an automaton A is the set $L(A)$ of all words w such that there exists an accepting run of A on w. An automaton over the single-letter alphabet is called *unary*.

A *deterministic finite automaton* (*DFA*) is an *NFA* such that I is a singleton and for all states q and all letters a there is at most one transition of the form $(q, a, q') \in T$.

An *unambiguous non-deterministic finite automaton* (*UFA*) is an *NFA* such that for every word there is at most one accepting run.

A unary non-deterministic finite automaton is in the *Chrobak normal form* [2] if it consists of a path of states followed by a single nondeterministic choice to a set of disjoint cycles.

An automaton is in *simple Chrobak normal form* if it consists of a disjoint union of cycles, each of them containing exactly one initial state.

The following theorem shows that every *UFA* can be transformed into one in Chrobak normal form without increase in size, and as a consequence we sought the construction that would have this shape.

▶ **Theorem 4** ([5]). *For all regular unary languages, there exists an unambiguous automaton recognizing the language which is minimal in size and is furthermore in Chrobak normal form.*

3 The construction

We present in this section the construction of the automaton A_d involved in the proof of Theorem 1. We also establish the unambiguity of A_d in Lemma 5 and compute its size in Lemma 6.

Parameters

The construction of A_d involves several parts, and the parameters have to be adjusted carefully for the lower bound. In this section, we use many parameters, to be specified in the final proof, in Section 6.

These parameters are the following:

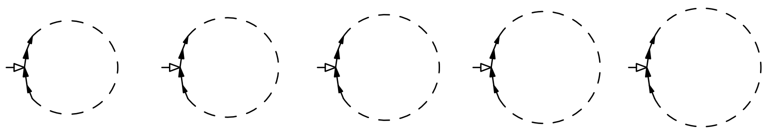
- $n \in \mathbb{N}$ is the number of cycles of the automaton A_d in simple Chrobak normal form.
- R is a tournament of size n: a tournament is an orientation of the edges of the complete undirected graph, see Section 4 for more details. The tournament R will eventually be required to have a special property, established in Lemma 7.
- $b \in \mathbb{N}$ is used as a basis for numbering, and we set $N = b^n$.
- $P = \{p_i \mid i = 0 \dots N-1\}$ is a set of N distinct primes. These will eventually be chosen sufficiently close one from each other thanks to Lemma 10.

The construction

We now construct the automaton A_d as follows.

- It consists of n disjoint cycles $C_1, \dots, C_n$, the cycle C_i having as length m_i which is the product of the primes p_j's such that the ith digit of j in base b is 0 (the digits are numbered from 1 to n). We write that p_j *belongs to* m_i if $p_j | m_i$.
- The 0th state of the cycle C_i is *initial*.
- The rth state of a cycle C_i is *accepting* if it satisfies three conditions:
  1. r is non null,
  2. r modulo p belongs to $\{0, i\}$ for all p belonging to m_i,
  3. if iRj for some j, then there exists a prime p belonging to both m_i and m_j such that $r \mod p = i$.

  And in this case, we call r an *accepting remainder for m_i*.



Let us look more precisely at the structure of this automaton.

We first note that the empty word is not accepted by this automaton, thanks to Item 1 of the definition. One can also note that each cycle is the product of b^{n-1} distinct prime numbers. Furthermore, if one computes the gcd of ℓ different m_i's, the result is the product of $b^{n-\ell}$ prime numbers. Hence there are many primes dividing a cycle, there are many primes dividing simultaneously two cycles, and so on.

Of course, the subtlety in this construction lies in the choice of the accepting remainders for each m_i. This has to respect several constraints. The remainders are chosen to be sufficiently complicated for allowing the lower bound proof, and there should be not too many of them in order to guarantee the unambiguity for A_d. In particular if Condition 3 was omitted, it would be easy find accepting remainders for two distinct m_i's that would yield ambiguity[2]. The Condition 3 is used to resolve these conflictual situations, and when an input would be accepted by two cycles, the tournament is used to "declare the winner".

Concretely, we prove:

▶ **Lemma 5.** *The automaton A_d is unambiguous.*

Proof. Assume that the automaton A_d would be ambiguous. This would mean that there exists a word, of length ℓ, such that it is accepted by two distinct cycles. Let us say by C_i and C_j. This means that $r = \ell \mod m_i$ is an accepting remainder for m_i, and $r' = \ell$

[2] Indeed, let p being a prime of m_i and p' a prime of m_j, consider, by the Chinese remainder theorem an integer ℓ that is equal to i modulo p, equal to j modulo p', and null modulo all other primes. In the absence of Assumption 3, the word of length ℓ would be accepted by both C_i and C_j.

mod m_j is an accepting remainder for m_j. Let us assume without loss of generality that iRj. This implies by Item 3 that there is a prime number p that belongs to both m_i and m_j such that $r \mod p = i$. Hence $\ell \mod p = i$ since p belongs to m_i. Hence $r' \mod p = i$ since p also belongs to m_j. However, we know that r' is an accepting remainder for m_j, therefore Item 2 requires that $r' \mod p \in \{0, j\}$. A contradiction. ◀

We conclude this section by computing the size of this automaton.

▶ **Lemma 6.** *The automaton A_d has between $n(\min P)^{n^{b-1}}$ and $n(\max P)^{n^{b-1}} = n(\max P)^{\frac{N}{b}}$ states.*

Proof. Indeed, the automaton is a union of n cycles and the length of each cycle is a product of $b^{n-1} = \frac{N}{b}$ primes from P. ◀

The rest of the proof is now devoted to showing that there are no small non-deterministic automata for the complement of the language accepted by A_d.

4 Tournaments

A *tournament graph* (or simply a *tournament*) of size n is an orientation of the complete graph. In our case, we see it as a relation over $\{1, \ldots, n\}$ such that for all distinct $i, j = 1 \ldots n$, either iRj and not jRi, or jRi and not iRj. By convention, iRi is assumed to never hold.

As we have seen in the previous section, a tournament is used as a parameter in the construction of the automaton A_d. For our lower bound proof to go through, we use the fact that this tournament has a special technical property, that is shown possible according to the following lemma.

▶ **Lemma 7.** *For all positive integers k, there exists a tournament R such that the following property holds: for all $E \subseteq R$, if for all vertices x there exists a vertex y such that xEy, then E contains at least k distinct edges that do not share an extremity.*

It is possible to chose a tournament with this property of size $n = 12k^2 2^{2k}$.

The rest of this section is devoted to the proof of Lemma 7. Note that this proof involves a probabilistic argument.

The core notion used in the proof, and therefore the notion at the core of the entire proof of Theorem 1, is the notion of inbound-covering sets.

▶ **Definition 8.** A set S is an *inbound-covering set* for a tournament R if for all vertices x outside S, we have xRy for some $y \in S$.

▶ **Lemma 9.** *For every positive integer h there exist a large enough integer n and a tournament of size n such that the smallest inbound-covering set has size larger than h.*

It is enough to take $n = 3h^2 2^h$.

Proof. Consider a uniformly random tournament of size n, i.e., the vertices are fixed as $1, \ldots, n$, and for all $i < j$, one tosses a fair coin in order to chose whether iRj or jRi. Consider an arbitrary set $S \subset V(G)$ of size h. The probability (over the choice of a tournament) that a given vertex $v \in V(G) \setminus S$ has at least one edge from v to S is $1 - 2^{-h}$. For a given set S and $v_1, v_2, \ldots \in V(G) \setminus S$, the existence of an outgoing edge from v_i towards S is independent for the different vertices (indeed for all $i \neq j$, the set of edges from v_i to S and the set of edges from v_j to S are disjoint and thus their orientations are chosen independently). Therefore the probability for a given set S to be inbound-covering is equal to $(1 - 2^{-h})^{(n-h)}$. Note that since $\log(1 - 2^{-h}) < (-2^{-h})$, this quantity is bounded from above by $e^{-2^{-h}(n-h)}$ $(\star)$.

Let us provide now an upper bound on the probability α that a tournament has an inbound covering set of size h. Since there are (less than) n^h sets of size n and using ($\star$), we immediately get that

$$\alpha \leqslant n^h \exp\left(-\frac{n-h}{2^h}\right) = \exp\left(h\log n - \frac{n-h}{2^h}\right) .$$

We shall prove now that for $h \geqslant 8$ and $n = 3h^2 2^h$, this quantity α is smaller than one, which concludes the proof. According to the above inequality, it is sufficient for proving $\alpha < 1$ to establish that $h \log n < \frac{n-h}{2^h}$, which is equivalent to $h2^h \log n < n - h$. We establish this inequality as follows:

$$\begin{aligned} h2^h \log n &= h2^h(h\log 2 + 2\log h + \log 3) \\ &< h2^h \times (2h) \\ &= 2h^2 2^h \\ &< 3h^2 2^h - h \\ &= n - h \end{aligned}$$ ◀

▶ Remark. Note that, as it is customary with probabilistic constructions, our choice of n is in fact enough to ensure that *most* tournaments have no inbound-covering sets of sizes up to h.

Now we can prove Lemma 7.

Proof. By Lemma 9 we can pick a tournament with orientation R that has no inbound-covering sets of size up to $h = 2k$. We can choose such a tournament of size $n = 3h^2 2^h = 12k^2 2^{2k}$.

Assume we have already constructed 2ℓ distinct vertices $x_1, y_1, \ldots, x_\ell, y_\ell$ forming edges $(x_1, y_1), \ldots, (x_\ell, y_\ell)$. Since $S = \{x_1, y_1, \ldots, x_\ell, y_\ell\}$ has cardinality $2\ell < 2h$, it is not an inbound-covering set. Hence, one can find a vertex $x_{\ell+1}$ such that there is an edge from all vertices of S to it. We know that E must contain some edge $(x_{\ell+1}, y_{\ell+1})$ from $x_{\ell+1}$, and this edge cannot lead to S, so the edge $(x_{\ell+1}, y_{\ell+1})$ doesn't share an extremity with any previously chosen edge. Applying this argument by induction on ℓ for $\ell = 0$ to k, we have proved Lemma 7. ◀

5 Choice of primes

▶ **Lemma 10.** *For all large enough N it is possible to select N primes no larger than $4N^2 \log N$ within a factor of $1 + \frac{1}{N}$ of each other.*

Proof. We will take the interval of length $3N \log N$ between $3N^2 \log N$ and $4N^2 \log N$ that contains the most primes. By the Prime number theorem there are

$$\frac{3N^2 \log N}{2\log N + \log\log N + \log 3} + o(N^2) = \frac{3}{2}N^2 + o(N^2)$$

primes no larger than $3N^2 \log N$ and

$$\frac{4N^2 \log N}{2\log N + \log\log N + \log 4} + o(N^2) = 2N^2 + o(N^2)$$

primes no larger than $4N^2 \log N$. Therefore, there are $\frac{1}{2}N^2 + o(1)$ primes between $3N^2 \log N$ and $4N^2 \log N$. If we divide this interval into subintervals of length $3N \log N$, the average subinterval will contain

$$\frac{1}{2}N^2\frac{3N\log N}{N^2\log N}(1+o(1)) = \frac{3}{2}N + o(N)$$

primes, which is enough. ◀

6 The lower bound

In this section we present the main combinatorial argument of the proof, and complete the proof of Theorem 1.

▶ **Lemma 11.** *A non-deterministic automaton that accepts the complement of the language of A_d has to have at least $(\min P)^{N(1-\exp(-\frac{k}{b^2}))}$ states.*

Let us fix ourselves a non-deterministic automaton C_d that accepts the complement of the language of A_d.

The principle of the proof of Lemma 11 is the following: as we have already noticed, the word of length $\prod_{p\in P} p$, since it is congruent to 0 modulo all the m_i's, is not accepted by A_d (this follows from Condition 1 in the definition of accepting remainders). Thus it has to be accepted by C_d. Since this word is very long (the length is much larger than the bound we want to prove), the run of C_d that accepts it has to visit twice some state and perform a cycle in the mean time. We shall look at what are the words obtained by pumping this cycle, that are all accepted by C_d, and obtain from this analysis that this cycle in C_d has to be rather long.

The core combinatorial result justifying this intuition is the following.

▶ **Lemma 12.** *Let A_d be constructed from a tournament of size $n = 12k^2 2^{2k}$ satisfying the conclusion of Lemma 7.*

Let x and y be integers such that

(a) $(\prod_{p\in P} p) = x + y$*, and;*

(b) $(xs + y) \mod m_i$ *is not an accepting remainder modulo m_i for all i and all $s \geqslant 0$,*

then x has to be divisible by at least $N(1-(1-\frac{1}{b}^2)^k)$ distinct primes from P.

Proof. Consider the set $E \subseteq R$ defined as

$$E = \{(i,j) \in R \mid \gcd(m_i, m_j) | x\} \ .$$

The proof then goes in two steps. We shall show in step 1 that the assumption for E in Lemma 7 are fulfilled. Then we will apply Lemma 7 and conclude in step 2.

Step 1: We assume that E does not fulfill the assumptions of Lemma 7, and head toward a contradiction. This means that we assume that there exists an $i = 1 \dots n$ such that whenever iRj then $\gcd(m_i, m_j)$ does not divide x.

According to the Chinese remainder theorem and existence of inverse in $\mathbb{Z}/p\mathbb{Z}$ there exists $s > 0$ such that $(xs + y) \mod p = i$ for all primes $p \in P$ that do not divide x. Note that for a prime p that divides x, since furthermore p divides $x + y$ (assumption (a) of the lemma), we obtain $p|y$, and thus $(xs + y) \mod p = 0$. Overall, for $r = xs + y$, we have that for all primes $p \in P$:

$$r \mod p = \begin{cases} 0 & \text{if } p \text{ divides } x \text{, and} \\ i & \text{otherwise.} \end{cases}$$

Let us show that this r is an accepting remainder:

1 Let j be such that iRj. By assumption, $\gcd(m_i, m_j) \nmid x$. Hence, there exists $p \in P$ that divides m_i but not x. For this p we know that $r \mod p = i$, therefore $r \mod m_i \neq 0$.
2 We have seen above that $r \mod p \in \{0, i\}$.
3 Let j be such that iRj. According to the assumption, $\gcd(m_i, m_j)$ does not divide x. Hence there exists a prime p that divides both m_i and m_j but not x. For this prime, we have seen that $r \mod p = i$.

However, we knew by assumption (b) of the lemma that a number of the form $xs + y$ such as r cannot be an accepting remainder. This is contradiction, and thus terminates the proof of the step 1.

Step 2: Let us now apply Lemma 7. According to the lemma, there are k distinct E-edges $(i_1, j_1), \ldots, (i_k, j_k)$ that do not share an extremity. Let us count the number of primes that divide both m_{i_t} and m_{j_t} for some $t = 1 \ldots k$ (and thus divide x). By construction of the m_i's, it contains all the primes p_v such that both the i_tth and the j_tth digits (in the base-b notation) of v are null for some $t = 1 \ldots k$.

We will first count the primes in P **not** dividing x. These are the primes p_v with v having a nonzero digit in at least one of the two positions i_t and j_t for every t. There are k pairs of positions and there are $b^2 - 1$ combinations of digits that are not $(0, 0)$. There are also $n - 2k$ positions with no such constraints. The total number of possible combinations is $(b^2-1)^k b^{n-2k} = (b^2(1-\frac{1}{b^2}))^k b^{n-2k} = b^n(1-\frac{1}{b^2})^k = N(1-\frac{1}{b^2})^k$.

The primes in P dividing x are all the other primes, and there are $N - N(1-\frac{1}{b^2})^k = N(1-(1-\frac{1}{b^2})^k)$ of them. ◀

Let us prove Lemma 11.

Proof. Let us fix a tournament of size n according to Lemma 7.

Let use fix a set of primes according to Lemma 10.

An *NFA* recognizing the complement of the language has to have a cycle, because the complement is infinite. Consider the word of length $\prod_{p \in P} p$. This length is obviously greater than $|A_d|^d$. If the *NFA* has an accepting run over the word with no cycles, it has to be very large. Otherwise, let C be a cycle of length x occurring in this run, and y be the remaining part of the run length, i.e. $\prod_{p \in P} p = x + y$ (a). The product of all the primes $\prod_{p \in P} p$ has remainder zero modulo every modulus m_i in the construction. By iterating $s \geqslant 0$ times the cycle C, we obtain that the word of length $xs + y$ has to be accepted by C_d. Thus it is not accepted by A_d, and hence $(xs + y) \mod m_i$ is not an accepting remainder for all $s \geqslant 0$ and $i = 1 \ldots n$ (b).

Hence the assumptions (a) and (b) of Lemma 12 are fulfilled. It follows that the cycle C has a length x divisible by $N(1-(1-\frac{1}{b}^2)^k)$ distinct primes from P.

This ensures that the cycle C has a length at least $(\min P)^{N(1-(1-\frac{1}{b}^2)^k)}$. Since furthermore $(1-\frac{1}{b^2})^k < \exp(-\frac{1}{b^2})^k = \exp -\frac{k}{b^2}$, this is at least $(\min P)^{N(1-\exp(-\frac{k}{b^2}))}$.

The size of the *NFA* cannot be less than that. ◀

We can finally conclude the proof of the main theorem of this paper, Theorem 1.

Proof of Theorem 1. Let us fix d. Let $b = 2d, k = b^2, n = 12k^2 2^{2k}, N = b^n$.

By Lemma 6 the size of the automaton A_d is at most $n(\max P)^{\frac{N}{b}}$ states. This automaton is unambiguous by Lemma 5.

By Lemma 11 each *NFA* recognizing the complement of the language of A_d must have at least $(\min P)^{N(1-\exp(-\frac{k}{b^2}))}$ states. As $\frac{k}{b^2} = 1$, the size of the *NFA* cannot be less than $(\min P)^{0.6N}$.

We now only need to verify that $(O(n)(\max P)^{\frac{N}{b}})^d < (\min P)^{0.6N}$. But indeed, for large enough d we have $\min P > N \gg n \gg d$ and

$$(O(n)(\max P)^{\frac{N}{b}})^d < ((O(n)(1+2\frac{1}{N})(\min P))^{\frac{N}{2d}})^d$$
$$< O(n)^{\frac{N}{2}} \exp(\frac{d}{N})(\min P)^{\frac{N}{2}} < (\min P)^{0.6N}$$

In case of d not large enough, we can replace the automaton with the automaton for the smallest large enough d.

Let us estimate the size of A_d. We know that b is linear in d, k is quadratic in d, n is $2^{\Theta(d^2)}$, N is $b^n = b^{2^{\Theta(d^2)}} = 2^{(\log b)2^{\Theta(d^2)}} = 2^{2^{\Theta(d^2)}}$. The primes in P are all $\Theta(N^2 \log N)$. Then the size of the automaton A_d is $\Theta(n(\min P)^{\frac{N}{b}}) = (\min P)^{\Theta(\frac{N}{b})} = (N^2 \log N)^{\Theta(\frac{N}{b})} = 2^{2^{\Theta(d^2)} 2^{2^{\Theta}(d^2)}} = 2^{2^{2^{\Theta}(d^2)}} = 2^{2^{2^{d^{\Theta(1)}}}}$ ◀

7 Sweeping automata

We will now make some additional remarks about the application of the main construction to two-way and sweeping automata.

First we remind the definitions of two-way and sweeping automata.

▶ **Definition 13.** A two-way non-deterministic finite automaton (*2NFA*) is defined by an alphabet Σ, a set of *states* $Q \sqcup \{\top, \bot\}$, a subset of *initial states* I, and the *transition relation* $T \subseteq Q \times (\Sigma \sqcup \{\vdash, \dashv\}) \times (Q \sqcup \{\top, \bot\}) \times \{+1, -1\}$. We call $\vdash$ and $\dashv$ *endpoint markers*.

A *run* of an *2NFA* on an input word $u_1 \dots u_k$ is a list of pairs of positions and states, $(x_0 = 1, q_0 \in I)$, $(x_1, q_1), \dots, (x_n, q_n)$ such that all transitions are allowed and the run ends with one of the special states $\top, \bot$. The exact conditions are as follows:

1. x_0 is 1;
2. q_0 is in I;
3. all x_i are between 0 and $k+1$;
4. $(q_{i-1}, w_{x_{i-1}}, q_i, x_i - x_{i-1}) \in T$ for all $i = 1 \dots n$, in which we assume that $u_0 =\vdash$ and $u_{k+1} =\dashv$;
5. the last state q_n is either $\top$ or $\bot$;
6. $x_i \neq x_{i-1}$ for all $i = 1 \dots n$.

A run is *accepting* if the last state is $\top$.

A *two-way non-deterministic finite automaton* (*2DFA*) is a *2NFA* such that for every state q and every letter a there is at most one transition of the form $(q, s, q', j) \in T$.

A *sweeping two-way deterministic finite automaton* (*swNFA*) is a *2NFA* with exactly one initial state such that for each state q all the transitions of the form (q, s, q', j) where s is in Σ have the same j.

A *swDFA* is an *swNFA* that is also a *2DFA*.

▶ **Lemma 14.** *The languages $L(A_d)$ and $\overline{L(A_d)}$ constructed in the proof of Theorem 1 can also be recognized. by a swDFA of size $|A_d|$.*

Proof. A *swDFA* can go through the word n times calculating the remainder modulo the next modulus each time. This construction requires the same number of states as the *UFA* constructed in the proof of Theorem 1. Such a *swDFA* can be constructed to recognize either the language or its complement. ◀

▶ **Theorem 15.** *Converting a unary sweeping two-way deterministic automaton to a non-deterministic finite automaton for the same language may require a superpolynomial size.*

Proof. Consider the automata constructed in Lemma 14. ◀

8 Conclusion and further directions

We have constructed a counterexample to the conjecture that the complement of a language recognized by an *UFA* can be recognized by an *UFA* with polynomial increase in the number of states. Moreover, in our example the language and its complement are easy to recognize by a *swDFA* with approximately the same number of states, but the complement requires superpolynomial number of states in the recognizing *NFA* even without the requirement of unambiguity. The example only uses the single-letter alphabet.

The construction provides a relatively weak kind of superpolynomial growth. It would be interesting to improve the lower bound. It seems likely that the number of primes used in the construction could be reduced, making the growth faster.

The question about exponential separation in the case of a general alphabet remains open. We hope that our counterexample to the conjectured polynomial upper bound for complementing *UFA* will inspire new results in this area.

References

1 Jean-Camille Birget. Partial orders on words, minimal elements of regular languages and state complexity. *Theor. Comput. Sci.*, 119(2):267–291, 1993. doi:10.1016/0304-3975(93)90160-U.

2 Marek Chrobak. Finite automata and unary languages. *Theor. Comput. Sci.*, 47(3):149–158, 1986. doi:10.1016/0304-3975(86)90142-8.

3 Thomas Colcombet. Unambiguity in automata theory. In Jeffrey Shallit and Alexander Okhotin, editors, *Descriptional Complexity of Formal Systems - 17th International Workshop, DCFS 2015, Waterloo, ON, Canada, June 25-27, 2015. Proceedings*, volume 9118 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 2015. doi:10.1007/978-3-319-19225-3_1.

4 Viliam Geffert, Carlo Mereghetti, and Giovanni Pighizzini. Complementing two-way finite automata. *Inf. Comput.*, 205(8):1173–1187, 2007. doi:10.1016/j.ic.2007.01.008.

5 Tao Jiang, Edward McDowell, and Bala Ravikumar. The structure and complexity of minimal nfa's over a unary alphabet. *Int. J. Found. Comput. Sci.*, 2(2):163–182, 1991. doi:10.1142/S012905419100011X.

6 Jozef Jirásek Jr., Galina Jirásková, and Juraj Sebej. Operations on unambiguous finite automata. In Srecko Brlek and Christophe Reutenauer, editors, *Developments in Language Theory - 20th International Conference, DLT 2016, Montréal, Canada, July 25-28, 2016, Proceedings*, volume 9840 of *Lecture Notes in Computer Science*, pages 243–255. Springer, 2016. doi:10.1007/978-3-662-53132-7_20.

7 Michal Kunc and Alexander Okhotin. State complexity of operations on two-way finite automata over a unary alphabet. *Theor. Comput. Sci.*, 449:106–118, 2012. doi:10.1016/j.tcs.2012.04.010.

8 Hing Leung. Descriptional complexity of nfa of different ambiguity. *Int. J. Found. Comput. Sci.*, 16(5):975–984, 2005. doi:10.1142/S0129054105003418.

9 Hing Leung. Structurally unambiguous finite automata. In Oscar H. Ibarra and Hsu-Chun Yen, editors, *Implementation and Application of Automata, 11th International Conference, CIAA 2006, Taipei, Taiwan, August 21-23, 2006, Proceedings*, volume 4094 of *Lecture Notes in Computer Science*, pages 198–207. Springer, 2006. doi:10.1007/11812128_19.

10 Oleg B Lupanov. A comparison of two types of finite automata. *Problemy Kibernetiki*, 9:321–326, 1963.

11 J.I. Lyubich. Estimates of the number of states that arise in the determinization of a nondeterministic autonomous automaton. *Sov. Math., Dokl.*, 5:345–348, 1964.

12 Filippo Mera and Giovanni Pighizzini. Complementing unary nondeterministic automata. *Theor. Comput. Sci.*, 330(2):349–360, 2005. doi:10.1016/j.tcs.2004.04.015.

13 Alexander Okhotin. Unambiguous finite automata over a unary alphabet. *Inf. Comput.*, 212:15–36, 2012. doi:10.1016/j.ic.2012.01.003.

14 Seppo Sippu and Eljas Soisalon-Soininen. *Parsing Theory. Vol. 1: Languages and Parsing.* Springer-Verlag New York, Inc., New York, NY, USA, 1988.

15 Richard Edwin Stearns and Harry B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM J. Comput.*, 14(3):598–611, 1985. doi:10.1137/0214044.