Lecture Notes in Computer Science 13439

More information about this series at https://link.springer.com/bookseries/558

Yo-Sub Han · György Vaszil (Eds.)

Descriptional Complexity of Formal Systems

24th IFIP WG 1.02 International Conference, DCFS 2022
Debrecen, Hungary, August 29–31, 2022
Proceedings

Springer

Editors
Yo-Sub Han
Yonsei University
Seoul, Korea (Republic of)

György Vaszil
University of Debrecen
Debrecen, Hungary

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-13256-8 ISBN 978-3-031-13257-5 (eBook)
https://doi.org/10.1007/978-3-031-13257-5

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains the papers presented at the 24th International Conference on Descriptional Complexity of Formal Systems (DCFS 2022) which was held at the University of Debrecen, Hungary, during August 29–31, 2022. It was jointly organized by the Working Group 1.02 on Descriptional Complexity of the International Federation for Information Processing (IFIP) and by the Department of Computer Science at the Faculty of Informatics of the University of Debrecen.

The DCFS conference series is an international venue for the dissemination of new results related to all aspects of descriptional complexity including, but not limited to, the following:

- Automata, grammars, languages, and other formal systems; various modes of operations and complexity measures
- Succinctness of description of objects, state-explosion-like phenomena
- Circuit complexity of Boolean functions and related measures
- Size complexity of formal systems
- Structural complexity of formal systems
- Trade-offs between computational models and modes of operation
- Applications of formal systems (e.g., in software and hardware testing, in dialogue systems, in systems modeling or in modeling natural languages) and their complexity constraints
- Cooperating formal systems
- Size or structural complexity of formal systems for modeling natural languages
- Complexity aspects related to the combinatorics of words
- Descriptional complexity in resource-bounded or structure-bounded environments
- Structural complexity as related to descriptional complexity
- Frontiers between decidability and undecidability
- Universality and reversibility
- Nature-motivated (bio-inspired) architectures and unconventional models of computing
- Blum static (Kolmogorov/Chaitin) complexity, algorithmic information

DCFS became an IFIP working conference in 2016, continuing the former Workshop on Descriptional Complexity of Formal Systems, which was a merger in 2002 of two other workshops: Formal Descriptions and Software Reliability (FDSR) and Descriptional Complexity of Automata, Grammars and Related Structures (DCAGRS). DCAGRS was previously held in Magdeburg (1999), London (2000), and Vienna (2001). FDSR was previously held in Paderborn (1998), Boca Raton (1999), and San Jose (2000). Since 2002, DCFS has been successively held in London, Ontario, Canada (2002), Budapest, Hungary (2003), London, Ontario, Canada (2004), Como, Italy (2005), Las Cruces, New Mexico, USA (2006), Nový Smokovec, High Tatras, Slovakia (2007), Charlottetown, Prince Edward Island, Canada (2008), Magdeburg, Germany (2009), Saskatoon, Canada (2010), Giessen, Germany (2011), Braga, Portugal (2012), London, Ontario, Canada (2013), Turku, Finland (2014), Waterloo, Ontario, Canada (2015), Bucharest, Romania (2016), Milan, Italy (2017), Halifax, Nova Scotia, Canada (2018), and Košice, Slovakia (2019). The next DCFS conferences were planned to be held in Vienna, Austria (2020), and in Seoul, South Korea (2021), but both of these events were canceled as in-person meetings due to the COVID-19 pandemic. The accepted papers appeared only in the conference proceedings.

This year 17 papers were submitted by authors from 14 different countries. The number of submissions was less than usual, probably due to the current problems in the world and to the desirable and aspired return to an in-person conference. On the other hand, these submissions were of extraordinary quality. Therefore, after the review of each paper by three referees, the Program Committee were able to accept 14 papers out of the 17 submissions.

The program also included four invited talks by

- Mikołaj Bojańczyk, University of Warsaw, Poland,
- Stefano Crespi Reghizzi, Polytechnic University of Milan, Italy,
- Szabolcs Iván, University of Szeged, Hungary,
- Galina Jirásková, Slovak Academy of Sciences, Košice, Slovakia.

We thank all invited speakers, contributing authors, Program Committee members, and external referees for their valuable contributions towards the realization of DCFS 2022.

We are also grateful to the editorial staff at Springer for their guidance and help during the process of publishing this volume, and for supporting the event through publication in the LNCS series.

Partial financial support for the conference was provided by the Department of Computer Science and by the Faculty of Informatics of the University of Debrecen.

Finally, we would like to thank the members of the organizing committee who worked hard to make this edition successful and all participants who, either in-person or virtually, contributed to the success of the conference.

We are looking forward to DCFS 2023 in Potsdam, Germany.

June 2022

Yo-Sub Han
György Vaszil

Organization

Steering Committee

Cezar Câmpeanu	University of Prince Edward Island, Canada
Erzsébet Csuhaj-Varjú	Eötvös Loránd University, Hungary
Stavros Konstantinidis	Saint Mary's University, Canada
Martin Kutrib (Chair)	Justus Liebig University Giessen, Germany
Giovanni Pighizzini	University of Milan, Italy
Rogério Reis	University of Porto, Portugal
Kai Salomaa	Queen's University, Canada

Program Committee

Henning Bordihn	University of Potsdam, Germany
Johanna Björklund	University of Umeå, Sweden
Cezar Câmpeanu	University of Prince Edward Island, Canada
Erzsébet Csuhaj-Varjú	Eötvös Loránd University, Hungary
Szilárd Zsolt Fazekas	Akita University, Japan
Pawel Gawrychowski	University of Wrocław, Poland
Dora Giammarresi	Tor Vergata University of Rome, Italy
Yo-Sub Han (Co-chair)	Yonsei University, South Korea
Géza Horváth	University of Debrecen, Hungary
Galina Jirásková	Slovak Academy of Sciences, Košice, Slovakia
Stavros Konstantinidis	Saint Mary's University, Canada
Martin Kutrib	Justus Liebig University Giessen, Germany
Ian McQuillan	University of Saskatchewan, Canada
Alexander Okhotin	St. Petersburg State University, Russia
Andrei Păun	University of Bucharest, Romania
Giovanni Pighizzini	University of Milan, Italy
Narad Rampersad	University of Winnipeg, Canada
Rogério Reis	University of Porto, Portugal
Michel Rigo	University of Liège, Belgium
Kai Salomaa	Queen's University, Canada
György Vaszil (Co-chair)	University of Debrecen, Hungary
Matthias Wendlandt	Justus Liebig University Giessen, Germany
Lynette van Zijl	Stellenbosch University, South Africa

Additional Reviewers

Sabine Broda
Jürgen Dassow
Jozef Jirásek
Andreas Malcher
Nelma Moreira
Timothy Ng
Luca Prigioniero
Marek Szykuła
Bianca Truthe

Organizing Committee

Bence Hegedűs	University of Debrecen, Hungary
Géza Horváth	University of Debrecen, Hungary
Arnold Pintér	University of Debrecen, Hungary
György Vaszil	University of Debrecen, Hungary

Abstracts of Invited Talks

Polyregular Functions

Mikołaj Bojańczyk

Institute of Informatics, University of Warsaw, Poland
bojan@mimuw.edu.pl

Transducers are like automata, but instead of accepting/rejecting they produce an output, such as a string or a tree. This talk is about a class of string-to-string functions, called the polyregular functions, which can be seen as a candidate for the notion of regular string-to-string transducers of polynomial growth. The class has many equivalent characterisations, including monadic second-order logic, two-way automata, an imperative programming language with for loops, and functional programming languages.

On Scattered Context-free Order Types (Extended Abstract)

Szabolcs Iván[1]

Department of Informatics, University of Szeged, Hungary
szabivan@inf.u-szeged.hu

1 Introduction

When the alphabet Σ of a language $L \subseteq \Sigma^*$ is linearly ordered, the language itself can be seen as a linearly ordered set, by the lexicographic ordering $<$ in which $xay < xbz$ if $a < b$ and $x < xy$ if $y \in \Sigma^+$. As an example, with $\Sigma = \{a, b\}$ and $a < b$, the order types of the languages a^*, $a^* + b^*$ and b^*a^* are ω, $\omega + \omega$ and ω^2, respectively, with ω denoting the order type of the natural numbers. (For the last one, consider the chain $\varepsilon < a < aa < \ldots < b < ba < baa < \ldots < bb < \ldots$)

Clearly, we can encode any such Σ by a constant-length homomorphism into $\{a, b\}^*$ preserving the order type of the language (e.g. for $\Sigma = \{a, b, c, d\}$ we can use $\{aa, ab, ba, bb\}$ as the image of the letters) so generally it suffices to consider the binary alphabet when we are interested only in the order types. An order type is called *regular* (*context-free*, resp.) if it is the order type of some regular (context-free, resp.) language. Since the set Σ^* of all Σ-words is countable as well, the order type of any language is countable; on the other hand, since every countable order type can be embedded into the order type η of the rationals and $L = \{aa, bb\}^*ab$ has the order type η (since it is a dense ordering without least and greatest elements), every countable order type arises as the order type of some language.

An operational characterization of the regular order types was given in [11]. It was shown in [2] that an ordinal is regular if and only if it is less than ω^ω.

The central topic of the presentation, the study of context-free order types was initiated in [1]. From the model checking aspect of interactive programs, studying *scattered* order types might have its actual usage: an order type is scattered if it does not have a dense subordering. Hausdorff assigned a (countable) ordinal to the (countable) scattered orderings (see e.g. [13]), called its *rank*. In our results, we use a slightly modified definition of the original rank as follows: finite order types have rank 0 and if an order type is a finite sum of ζ-sums of scattered order types each having a rank less then α, then its order type is at most α. Formally we can define for each ordinal α a class H_α of (scattered, countable) order types as H_0 consisting of the finite order types and H_α being the smallest class containing each order type of the form $\sum\limits_{j \in \{1,\ldots,n\}} \sum\limits_{i \in \mathbb{Z}} o_{j,i}$ with

[1] Support of the ITM NKFIA TKP2021 grant is acknowledged.

each $o_{j,i}$ being a member of some H_β with $\beta < \alpha$. Then the *rank* of a (scattered countable) order type o is the least ordinal α with $o \in H_\alpha$. Due to Hausdorff's theorem, every scattered order type has a rank. As examples, ω, ζ, ω^k and ω^ω have ranks 1, 1, k and ω respectively, for the latter one we can write e.g. $\omega^\omega = 1 + \omega + \omega^2 + \omega^3 + \ldots$ which is an ω-sum of order types having a finite rank.

2 Selected Results

It is known [3] that an ordinal is regular if and only if it is less than ω^ω and it is context-free if and only if it is less than ω^{ω^ω}. Also, the rank of any scattered regular (context-free, resp.) order type is less than ω (ω^ω, resp.) [7, 11]. The other reason why it is interesting to study scattered context-free orderings is that it is decidable whether a context-free grammar G generates a scattered language [5] while it is undecidable whether it generates a dense one [6]. For the general case, it is even undecidable whether the order type of a context-free language is η [6]. However, for scattered context-free order types we do have some positive results: it is known [10] that the order type of a well-ordered language generated by a prefix grammar (i.e. in which each nonterminal generates a prefix-free language) is computable, thus the isomorphism problem of context-free ordinals is decidable if the ordinals in question are given as the lexicograpic ordering of *prefix* grammars. Also, the isomorphism problem of regular orderings is decidable as well [4, 14]. It is unknown whether the isomorphism problem of scattered context-free orderings is decidable – a partial result in this direction is that if the rank of such an ordering is at most one (that is, the order type is a finite sum of the terms ω, $-\omega$ and 1), then the order type is effectively computable from a context-free grammar generating the language [8, 9]. Moreover, it is also decidable whether a context-free grammar generates a scattered language of rank at most one. It is a very plausible scenario though that the isomorphism problem of scattered context-free orderings is undecidable in general – the rank 1 is quite low compared to the upper bound ω^ω of the rank of these orderings, and there is no known structural characterization of scattered context-free orderings. Clearly, among the well-orderings, exactly the ordinals smaller than ω^{ω^ω} are context-free but for scattered orderings the main obstacle is the lack of a finite "normal form" – as every ω-indexed sum of the terms ω and $-\omega$ is scattered of rank two, there are already uncountably many scattered orderings of rank two and thus only a really small fraction of them can possibly be context-free. So it makes sense to study language classes lying strictly between the regular and the context-free languages. One candidate can be that of the deterministic context-free languages: for these it is known that their order types are exactly the (general) context-free order types [7].

Another candidate for the next step is the class of the *one-counter languages*: these are the ones that can be recognized by a pushdown automaton having only one stack symbol. In [12], a family of well-ordered languages $L_n \subseteq \{a, b, c\}^*$ was given for each integer $n \geq 0$ so that the order type of L_n is $\omega^{\omega \times n}$ (thus its rank is $\omega \times n$) and Kuske formulated two conjectures: i) the order type of well-ordered one-counter languages is strictly less than ω^{ω^2} and more generally, ii) the rank of scattered one-counter languages is strictly less than ω^2. Of course the second conjecture implies the first.

In the main part of the presentation we aim to prove this second conjecture.

References

1. Bloom, S.L., Ésik, Z.: Regular and algebraic words and ordinals. In: Mossakowski, T., Montanari, U., Haveraaen, M. (eds.) Algebra and Coalgebra in Computer Science, vol. 4624, pp. 1–15. Springer, Berlin, Heidelberg (2007). 10.1007/978-3-540-73859-6_1
2. Bloom, S.L., Choffrut, C.: Long words: the theory of concatenation and omega-power. Theor. Comput. Sci. **259**(1), 533–548 (2001)
3. Bloom, S.L., Ésik, Z.: Algebraic ordinals. Fundam. Inform. **99**(4), 383–407 (2010)
4. Bloom, S.L., Ésik, Z.: The equational theory of regular words. Inform. Comput. **197**(1), 55–89 (2005)
5. Ésik, Z.: Scattered context-free linear orderings. In: Mauri, G., Leporati, A. (eds.) Developments in Language Theory, vol. 6795, pp. 216–227. Springer, Berlin, Heidelberg (2011). 10.1007/978-3-642-22321-1_19
6. Ésik, Z.: An undecidable property of context-free linear orders. Inform. Process. Lett. **111**(3), 107–109 (2011)
7. Ésik, Z., Iván, S.: Hausdorff rank of scattered context-free linear orders. In: Fernández-Baca, D. (eds.) LATIN 2012: Theoretical Informatics, vol. 7256, pp. 291–302. Springer, Berlin, Heidelberg (2012). 10.1007/978-3-642-29344-3_25
8. Gelle, K., Iván, S.: On the order type of scattered context-free orderings. In: The Tenth International Symposium on Games, Automata, Logics, and Formal Verification, 2–3 September 2019, pp. 169–182 (2019)
9. Gelle, K., Iván, S.: The order type of scattered context-free orderings of rank one is computable. In: Alexander, C., et al. (eds.) SOFSEM 2020: Theory and Practice of Computer Science - 46th International Conference on Current Trends in Theory and Practice of Informatics, SOFSEM 2020, Limassol, Cyprus, 20–24 January 2020, Proceedings of Lecture Notes in Computer Science, vol. 12011, pp. 273–284. Springer, Cham (2020). 10.1007/978-3-030-38919-2_23
10. Gelle, K., Iván, S.: The ordinal generated by an ordinal grammar is computable. Theor. Comput. Sci. **793**, 1–13 (2019)
11. Heilbrunner, S.: An algorithm for the solution of fixed-point equations for infinite words. RAIRO – Theor. Inform. Appl. **14**(2), 131–141 (1980)
12. Kuske, D.: Logical aspects of the lexicographic order on 1-counter languages. In: Chatterjee, K., Sgall, J. (eds.) Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, 26–30 August 2013. Proceedings, vol. 8087 of Lecture Notes in Computer Science, vol. 8087, pp. 619–630. Springer, Berlin, Heidelberg (2013). 10.1007/978-3-642-40313-2_55
13. Rosenstein, J.G.: Linear orderings. Pure Appl. Math. (1982)
14. Thomas, W.: On frontiers of regular trees. ITA **20**(4), 371–381 (1986)

Operations on Unambiguous Finite Automata (Extended Abstract)

Galina Jirásková[1]

Mathematical Institute, Slovak Academy of Sciences, Grešákova 6, 040 01, Košice, Slovakia
jiraskov@saske.sk

Abstract. We investigate the complexity of basic regular operations on languages represented by unambiguous finite automata. We get tight upper bounds for intersection (mn), left and right quotients ($2^m - 1$), positive closure $\left(\frac{3}{4} \cdot 2^n - 1\right)$, star $\left(\frac{3}{4} \cdot 2^n\right)$, shuffle ($2^{mn} - 1$), and concatenation $\left(\frac{3}{4} \cdot 2^{m+n} - 1\right)$. To describe witnesses, we use a binary alphabet for intersection and left and right quotients, a ternary alphabet for positive closure and star, a five-letter alphabet for shuffle, and a seven-letter alphabet for concatenation. We also discuss some partial results for complementation (between $2^{\log\log\log n}$ and $\sqrt{n+1} \cdot 2^{n/2}$) and union (between $mn + m + n$ and $m + n \cdot \sqrt{m+1} \cdot 2^{m/2}$ where $m \leq n$).

1 Introduction

A nondeterministic finite automaton (with multiple initial states, NFA) is *unambiguous* (UFA) if it admits at most one accepting computation on every input string. Ambiguity in finite automata was first considered by Schmidt [15] in his unpublished thesis, where he developed a lower bound method for the size of unambiguous automata based on the rank of certain matrices. He also obtained a lower bound of $2^{\Omega(\sqrt{n})}$ on the conversion of unambiguous finite automata into deterministic finite automata (DFAs).

Leung [10] improved the UFA-to-DFA trade-off to the tight upper bound 2^n. He described, for every n, a binary n-state UFA with a unique initial state whose equivalent DFA requires 2^n states. A similar binary example with multiple initial states was given by Leiss [8], and a ternary one was presented already by Lupanov [11]; notice that the reverse of Lupanov's witness for NFA-to-DFA conversion is deterministic. Using an elaborated Schmidt's lower bound method, Leung [11] described, for every n, an n-state NFA, in fact, a DFA with multiple initial states, whose equivalent UFA requires $2^n - 1$ states.

Stearns and Hunt [17] showed that it can be tested in polynomial time whether or not a given nondeterministic finite automaton is unambiguous. They also provided polynomial-time algorithms for the equivalence and containment problems for unambiguous finite automata.

[1] Research supported by VEGA grant 2/0132/19.

Hromkovič et al. [4] further elaborated a lower bound method for UFAs. Using communication complexity they showed that so-called exact cover of all 1's with monochromatic sub-matrices in a communication matrix of a language provides a lower bound on the size of any UFA for this language, and they simplified some proofs presented in [15, 17].

Okhotin [13] examined unambiguous automata over a one-letter alphabet. He proved that the UFA-to-DFA trade-off in the unary case is given by a function in $e^{\Theta(\sqrt[3]{n(\ln n)^2})}$, while the NFA-to-UFA trade-off is $e^{\sqrt{n \ln n}(1+o(1))}$. He also obtained the tight upper bound $(n-1)^2+1$ for star, an upper bound mn, tight if m, n are relatively prime, for concatenation, and a lower bound $n^{2-\varepsilon}$ for complementation of unary unambiguous automata.

Here we discuss the results on the complexity of basic regular operations on languages represented by unambiguous finite automata over an arbitrary alphabet obtained by Jirásek, Jirásková, and Šebej [6]. To get upper bounds, we provide a construction of a UFA recognizing the language resulting from an operation. In the case of intersection, the corresponding product automaton is unambiguous. In all the remaining cases, we first describe a nondeterministic automaton for the resulting language, and then count the number of its reachable non-empty sets. Such a number provides an upper bound on the size of an equivalent partial deterministic, so unambiguous, subset automaton.

To get lower bounds, we first restate the lower bound method from [10, 15]. To any NFA N, we assign a matrix M_N whose rows are indexed by sets that are reachable in N and columns by sets that are co-reachable in N, and whose entry (S, T) includes 0 if S and T are disjoint and it includes 1 otherwise. The rank of such a matrix provides a lower bound on the number of states in any unambiguous automaton recognizing the language $L(N)$. Then, using the known fact that the rank of the matrix is $2^n - 1$ if its rows and columns are indexed by all the non-empty subsets of a set of size n and its entries are as described above, we get an observation that the number of reachable sets in any NFA provides a lower bound on the size of any equivalent UFA if all the non-empty sets are co-reachable in the given NFA.

We use this observation to get lower bounds for quotients, positive closure, shuffle, and concatenation. We describe witness languages in such a way that in an NFA for the resulting language, all the non-empty sets are co-reachable, and the number of reachable sets is as large as possible. In the case of our intersection witnesses, the matrix corresponding to the resulting product automaton is an identity matrix of size mn, while in the case of star, we must inspect carefully the rank of the corresponding matrix.

An upper bound on the complexity of complementation of a language represented by a UFA is given by the number of reachable set in a given UFA, as well as by the number of its co-reachable sets. We show that the minimum of these two numbers is at most $2^{0.79n + \log n}$. This upper bound can be further decreased to $\sqrt{n+1} \cdot 2^{n/2}$ as shown by Indzhev and Kiefer [5]. A superpolynomial lower bound on the complexity of complementation on unambiguous automata has been recently obtained by Raskin [14].

2 Preliminaries

We assume that the reader is familiar with basic notions in formal languages and automata theory. For details and all the unexplained notions, the reader may refer to [3, 16].

A *nondeterministic finite automaton* (NFA) is a 5-tuple $N = (Q, \Sigma, \Delta, I, F)$, where Q is a finite nonempty set of states, Σ is a finite nonempty set of input symbols called the input alphabet, $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation, $I \subseteq Q$ is the set of initial states, and $F \subseteq Q$ is the set of final states. Each element (p, a, q) of Δ is called a *transition* of N. A *computation* of N on an input string $a_1 a_2 \cdots a_n$ is a sequence of transitions $(q_0, a_1, q_1)(q_1, a_2, q_2) \cdots (q_{n-1}, a_n, q_n) \in \Delta^*$. The computation is *accepting* if $q_0 \in I$ and $q_n \in F$; in such a case we say that the string $a_1 a_2 \cdots a_n$ is accepted by N. The *language accepted by* the NFA N is the set of strings $L(N) = \{w \in \Sigma^* \mid w \text{ is accepted by } N\}$.

An NFA $N = (Q, \Sigma, \Delta, I, F)$ is *unambiguous* (UFA) if it has at most one accepting computation on every input string, and it is *(partial) deterministic* (DFA) if $|I| = 1$ and for each state p in Q and each symbol a in Σ, there is at most one state q in Q such that (p, a, q) is a transition of N. It follows immediately from the definition that every (partial) deterministic automaton is unambiguous.

The transition relation Δ may be viewed as a function $\cdot : Q \times \Sigma \to 2^Q$, and it can be extended to the domain $2^Q \times \Sigma^*$ in the natural way. We denote this extended function by $\cdot$ as well. Then $L(N) = \{w \in \Sigma^* \mid I \cdot w \cap F \neq \emptyset\}$.

Every NFA $N = (Q, \Sigma, \cdot, I, F)$ can be converted to an equivalent deterministic automaton $\mathcal{D}(N) = (2^Q, \Sigma, \cdot, I, \{S \in 2^Q \mid S \cap F \neq \emptyset\})$, called the *subset automaton* of N [16]. Removing the empty set from the subset automaton results in an equivalent partial deterministic, so unambiguous, automaton. This gives the following observation.

Proposition 1. *Every language accepted by an n-state* NFA *is recognized by a* UFA *of at most* $2^n - 1$ *states.* □

A subset S of the state set Q of an NFA $N = (Q, \Sigma, \cdot, I, F)$ is *reachable* if $S = I \cdot w$ for some string w, and it is *co-reachable* if it is reachable in the reverse of N obtained from N be reversing all its transitions and by swapping the roles of its initial and final states. Using these notions we get the following characterization of unambiguous automata.

Proposition 2. *A nondeterministic finite automaton is unambiguous if and only if* $|S \cap T| \leq 1$ *for each reachable set S and each co-reachable set T.* □

If the reverse of an NFA is deterministic, then each co-reachable set in N is of size one,which gives the next observation.

Proposition 3. *An nondeterministic finite automaton is unambiguous if its reverse is (partial) deterministic.* □

Now we restate the lower bound method from [10, 15].

Proposition 4 (Lower bound method for UFAs). *Let N be an NFA. Let M be the matrix with rows (columns) indexed by reachable (co-reachable) sets of N, in which the entry (S, T) includes 0 if S and T are disjoint, and 1 otherwise. Then every UFA recognizing $L(N)$ has at least $rank(M)$ states.*

Proof. Let A be a minimal n-state unambiguous automaton recognizing $L(N)$. Consider a matrix M'_A whose rows are indexed by the states of A, and columns are indexed by strings generating the co-reachable sets in N. The entry (q, w) of M'_A is 1 if w^R is accepted by A from the state q, and it is 0 otherwise. Since A is unambiguous, for every column in M'_A there is at most one row that contains a 1. It follows that the row of M_N indexed by a set S is a sum of the rows of M'_A corresponding to the states in S. Thus every row of M_N is a linear combination of rows in M'_A, and therefore $rank(M_N) \leq rank(M'_A) \leq n$. □

Let M_n be a matrix with rows and columns indexed by all the non-empty subsets of a set of size n, and such that the entry (S, T) is 0 if S and T are disjoint, and it is 1 otherwise. Then $rank(M_n) = 2^n - 1$ [9, Lemma 3]. This gives the following corollary.

Proposition 5. *If every non-empty set is co-reachable in a nondeterministic finite automaton, then the number of its reachable sets provides a lower bound on the number of states in any equivalent unambiguous automaton.* □

3 Results

Let us start with the trade-offs between deterministic, nondeterministic, and unambiguous finite automata. Every unambiguous automaton of n states can be simulated by a DFA of at most 2^n states obtained by the subset construction. To get tightness, consider an NFA from from Fig. 1. Since its reverse is deterministic, this NFA is unambiguous. As shown by Leung [10, Theorem 1], every equivalent DFA has at least 2^n states.

Every NFA of n states can be simulated by a partial deterministic, so unambiguous, subset automaton of at most 2^n-1 states. To get tightness of this upper bound, consider the binary NFA from Fig. 2, a witness for complementation on NFAs from [7, Theorem 5]. Every non-empty set is reachable in this NFA, and since the reverse of this

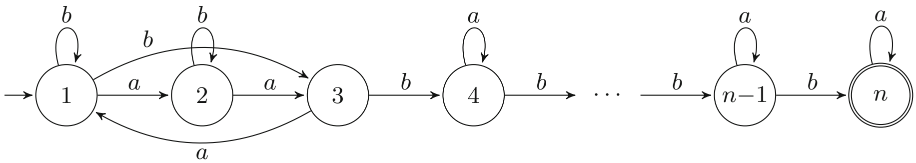

Fig. 1. A binary UFA-to-DFA witness meeting the upper bound 2^n [10].

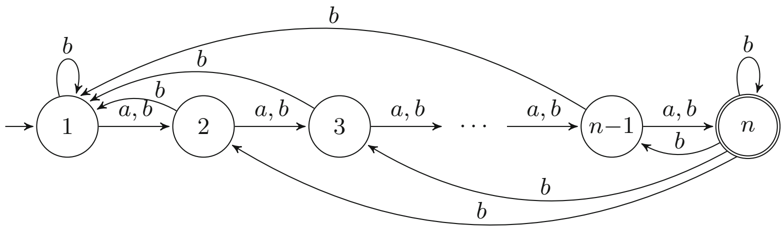

Fig. 2. A binary NFA-to-UFA witness meeting the upper bound 2^n-1.

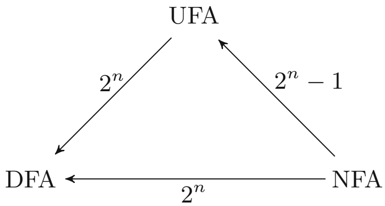

Fig. 3. The trade-offs between deterministic, nondeterministic, and unambiguous finite automata.

NFA is, in fact, the same NFA, every non-empty set is co-reachable as well. Hence every equivalent UFA has at least 2^n-1 states. Moreover, every equivalent DFA has at least 2^n states, since the empty set is reachable in the NFA from Fig. 2. The trade-offs between these three models of automata are shown in Fig. 3.

Now we continue with operational complexity on languages represented by unambiguous finite automata. Table 1 shows the known results on the complexity of basic regular operations on languages represented by deterministic and nondetermin-

Table 1. The complexity of regular operations on languages represented by deterministic and nondeterministic finite automata [2, 7, 12, 18].

Operation	DFA	$\|\Sigma\|$	NFA	$\|\Sigma\|$
Reversal	2^n	2	n	2
Intersection	mn	2	mn	2
Left quotient	$2^m - 1$	2	$m+1$	2
Right quotient	m	1	m	1
Shuffle	?		mn	2
Concatenation	$m \cdot 2^n - 2^{n-1}$	2	$m+n$	2
Positive closure	$\frac{3}{4} \cdot 2^n - 1$	2	n	1
Star	$\frac{3}{4} \cdot 2^n$	2	$n+1$	1
Complementation	n	1	2^n	2
Union	mn	2	$m+n$	2

Table 2. The complexity of regular operations on languages represented by unambiguous finite automata [6].

Operation	UFA	$\|\Sigma\|$
Reversal	n	1
Intersection	mn	2
Left quotient	$2^m - 1$	2
Right quotient	$2^m - 1$	2
Shuffle	$2^{mn} - 1$	5
Concatenation	$\frac{3}{4} \cdot 2^{m+n} - 1$	7
Positive closure	$\frac{3}{4} \cdot 2^n - 1$	3
Star	$\frac{3}{4} \cdot 2^n$	3
Complementation	$\leq 2^{0.8n}$	–
Union ($m \leq n$)	$mn + m + n \leq \cdot \leq m + n \cdot 2^{0.8m}$	4

istic finite automata, while Table 2 summarizes the corresponding results for unambiguous automata from Jirásek Jr., Jirásková, Šebej [6]. Both tables also display the size of alphabet used to describe witness languages. Let us discuss the results for UFAs in more detail.

Reversal. Since the reverse of an unambiguous automaton is unambiguous, the upper bound is n for the reversal operation. This upper bound is met by a one-string unary language a^{n-1} recognized by an n-state partial deterministic, so unambiguous, automaton. Its reversal is the same language which cannot be accepted by any nondeterministic automaton with less than n states.

Intersection. Notice that the product automaton for intersection of two unambiguous automata is unambiguous. This gives an upper bound mn for the intersection operation. The binary languages $\{w \in \{a, b\} \mid |w|_a = m - 1\}$ and $\{w \in \{a, b\} \mid |w|_b = n - 1\}$ meet this upper bound since in the corresponding product automaton each singleton set is reachable and co-reachable, and therefore the corresponding matrix is the identity matrix of size mn.

Left and Right Quotient. The left (right) quotient of a given language is recognized by a nondeterministic automaton obtained from an automaton for the given language by changing the set of initial (final) states. Applying the subset construction to the resulting automaton and omitting the empty set results in an incomplete deterministic, so also unambiguous, automaton for the language resulting from the quotient operation. This gives the upper bound $2^m - 1$ in both cases. To get witness for left quotient, consider the partial deterministic, so unambiguous, automaton from Fig. 4 and its left quotient by the language a^* recognized by a one-state unambiguous automaton. In the corresponding nondeterministic automaton for the left quotient, each non-empty set is reachable and co-reachable; notice that a shifts every subset cyclically by one, and b eliminates the state m. A similar idea works for the right quotient of the language recognized by the automaton from Fig. 4 by the empty string. Let us recall that the

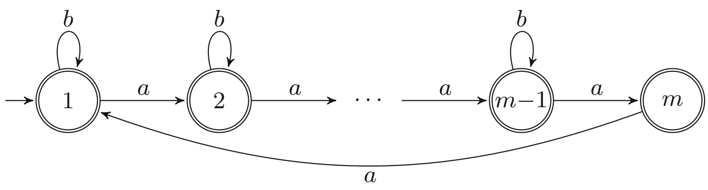

Fig. 4. A binary witness for left quotient (by a*) meeting the upper bound $2^m - 1$.

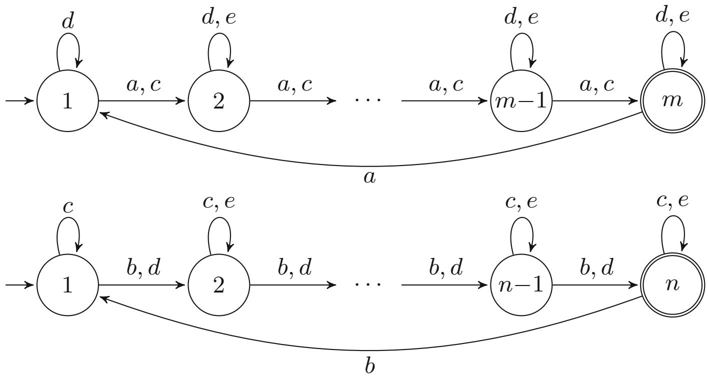

Fig. 5. Quinary witnesses for shuffle meeting the upper bound $2^{mn} - 1$.

upper bound on the complexity of right quotient on DFAs and NFA is just n since changing the set of final states in any DFA or NFA results in a DFA or NFA, respectively. However, changing the set of final states in an unambiguous automaton may not be unambiguous.

Shuffle. The shuffle of two languages represented by UFAs of m and n states is recognized by an mn-state NFA. This gives an upper bound $2^{mn} - 1$ for the shuffle operation on unambiguous automata. To describe witnesses, we use a five-letter alphabet and consider the languages recognized by partial deterministic, so unambiguous, automata shown in Fig. 5; cf.[1]. In the corresponding shuffle automaton, each non-empty set is reachable and co-reachable.

Concatenation. An automaton for the concatenation of two languages can be constructed from the corresponding unambiguous automata by adding the ε-transition from every final state of the first automaton to the initial state of the second automaton. In the resulting automaton, at least 2^{m+n-2} set of states are unreachable – those including a fixed final state of the first automaton and not including the initial state of the second automaton. After excluding the empty set, we get an upper bound $\frac{3}{4} \cdot 2^{m+n} - 1$ for the concatenation operation. For tightness, we consider the languages recognized by unambiguous automata shown in Fig. 6 defined over the seven-letter alphabet

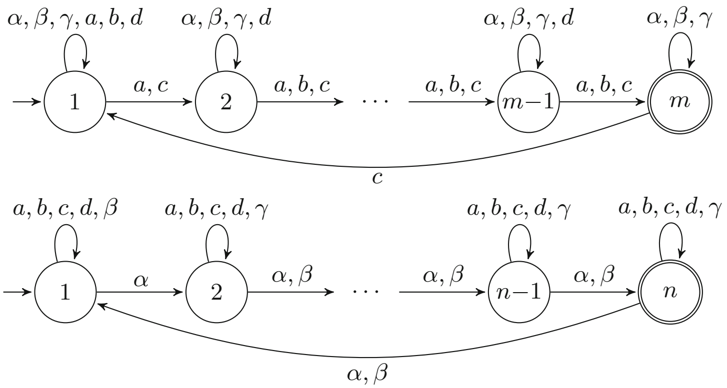

Fig. 6. Septenary witnesses for concatenation meeting the bound $\frac{3}{4} \cdot 2^{m+n} - 1$.

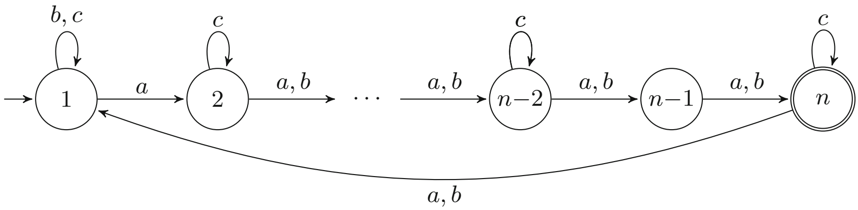

Fig. 7. A ternary witness for positive closure meeting the upper bound $\frac{3}{4} \cdot 2^n - 1$.

$\{a, b, c, d, \alpha, \beta, \gamma\}$; notice that the reverse of the first automaton as well as the second automaton are deterministic. In the corresponding automaton for concatenation of these two languages, each non-empty set is co-reachable, while $\frac{3}{4} \cdot 2^{m+n} - 1$ non-empty sets are reachable.

Positive Closure. To get an automaton for the positive closure of a regular language represented by an unambiguous automaton, we only need to add the ε-transition from every final state of this automaton to its initial state. In the resulting automaton, each set of states that contains a fixed final state and does not contain the initial state is unreachable which, after excluding the empty set, gives the upper bound $\frac{3}{4} \cdot 2^n - 1$. For tightness, we consider the binary witness DFA for star from [18], and we add a loop on a new symbol c in each state, except for the state $n - 1$ to get a ternary partial deterministic, so unambiguous, automaton shown in Fig. 7. The third symbol guarantees the co-reachability of every non-empty subset in the corresponding NFA for positive closure, while by strings over $\{a, b\}$ we get the reachability of $\frac{3}{4} \cdot 2^n - 1$ non-empty sets.

Star. In the case of the star operation, we need to add a new initial (and final) state in the construction from the previous paragraph which increases the upper bound by one.

The witness is the same as for the positive closure, but now we have to inspect carefully the binary matrix corresponding to the automaton for its star since now we cannot have the case when all non-empty sets are co-reachable.

Complementation. The complementation operation looks to be really challenging on unambiguous automata. A lower bound of $\Omega(n^{2-\varepsilon})$ has been obtained by Okhotin [13], while a superpolynomial lower bound has been recently provided by Raskin [14]. Although we are not able to improve these lower bounds, we can decrease the trivial upper bound 2^n to $2^{0.79n+\log n}$. The idea of the proof is to observe that given an n-state unambiguous automaton, the complement of its language is recognized by an unambiguous automaton of $\min\{|\mathcal{R}|, |\mathcal{C}|\}$ states, where $\mathcal{R}$ and $\mathcal{C}$ are the families of reachable and co-reachable sets in a given UFA, respectively. If the maximum of sizes of reachable sets is k, then

$$|\mathcal{R}| \leq \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{k}$$

$$|\mathcal{C}| \leq (k+1) \cdot 2^{n-k}$$

since every co-reachable set may have just one state from a fixed reachable set of size k. If $k \geq n/2$, then $|\mathcal{C}|$ is small enough. Otherwise, $|\mathcal{R}|$ is upper bounded by an increasing function $r(k) = n \cdot \left(\frac{en}{k}\right)^k$ and $|\mathcal{C}|$ is upper bounded by a decreasing function function $c(k) = n \cdot 2^{n-k}$, and we show that $\min\{r(k), c(k)\}$ is at most $2^{0.79n+\log n}$. Recently, Indzhev and Kiefer [5] decreased this upper bound to $\sqrt{n+1} \cdot 2^{n/2}$ by showing that the size of a UFA for the complemented language recognized by an n-state unambiguous automaton is upper bounded by the minimum of the number of cliques and co-cliques (independent sets) of a graph with n vertices, and then by showing that in every such graph the product of the number of its cliques with the number of its cocliques is bounded by $(n+1) \cdot 2^n$.

Union. First, notice that the standard NFA for union is unambiguous if two languages represented by unambiguous automata are disjoint. Without loss of generality, we may assume that $m \leq n$. Since $K \cup L = K \cup (L \cap K^c)$, and the languages K and $L \cap K^c$ are disjoint, we get an upper bound $m + n \cdot 2^{0.79m+\log m}$ for union on unambiguous automata. Taking into account the result from [5], this upper bound can be decreased to $m + n \cdot \sqrt{m+1} \cdot 2^{m/2}$. To get a lower bound of $mn + m + n$, we consider the quaternary partial deterministic, so unambiguous, automata with all states final such that in the first automaton, the symbol a performs a cyclic permutation, while b maps each state, except for the initial one, to itself, and c, d perform the identity. In the second automaton, the symbols a and b perform the identity, while c and d play the same role as a and b in the first automaton. Then, in the NFA for their union, all the non-empty sets are co-reachable, while $mn + m + n$ sets of size one and two are reachable.

4 Open Problems

In this section we state some problems that remain open in the research of the complexity of regular operations on languages represented by unambiguous automata. The problem of finding the exact complexity of complementation seems to be the most challenging.

Open Problem 1. *What is the exact complexity of complementation for unambiguous automata?*

Even some better lower or upper bounds for the complementation operation would be of interest; recall that the known lower bound is $2^{\log\log\log n}$ [14], while the best known upper bound is $\sqrt{n+1} \cdot 2^{n/2}$ [5]. We used the results for complementation to get an upper bound for union. Nevertheless, the gap between lower and upper bound is large in the case of union.

Open Problem 2. *What is the complexity of union for unambiguous automata?*

Our strategy for finding a witness for positive closure was to take the binary witness for the star operation on DFAs, and then define the transition on one more symbol to guarantee the co-reachability of all non-empty subsets in an NFA for positive closure. Perhaps, a new, completely different, witness could be described over a binary alphabet. A similar question arises in the case of the star operation.

Open Problem 3. *What is the complexity of positive closure or star for unambiguous automata in the binary case?*

In the case of shuffle and concatenation, our witnesses are defined over a five-letter and seven-letter alphabet, respectively. Our aim was to have proofs as simple as possible in [6], and we did not consider the possibility of decreasing the size of input alphabet.

Open Problem 4. *Can unambiguous witnesses for shuffle or concatenation be described over a smaller alphabet?*

The research on the complexity of operations for unambiguous automata [6] was really interesting, funny, and exciting for all three of us, and we believe that trying to solve the open problems stated above could be interesting, funny, and exciting as well.

Acknowledgment

I would like to thank my son Jozef and my student Juraj for our Mondays' seminars while working on the topic. I really miss them a lot.

References

1. Câmpeanu, C., Salomaa, K., Yu, S.: Tight lower bound for the state complexity of shuffle of regular languages. J. Autom. Lang. Comb. **7**(3), 303–310 (2002)
2. Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. Int. J. Found. Comput. Sci. **14**(6), 1087–1102 (2003). 10.1142/S0129054103002199

3. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley (1979)
4. Hromkovič, J., Seibert, S., Karhumäki, J., Klauck, H., Schnitger, G.: Communication complexity method for measuring nondeterminism in finite automata. Inf. Comput. **172**(2), 202–217 (2002). 10.1006/inco.2001.3069
5. Indzhev, E., Kiefer, S.: On complementing unambiguous automata and graphs with many cliques and Cocliques. Inf. Process. Lett. **177** (2022). 10.1016/j.ipl.2022.106270. To apper
6. Jirásek Jr, J., Jirásková, G., Šebej, J.: Operations on unambiguous finite automata. Int. J. Found. Comput. Sci. **29**(5), 861–876 (2018). 10.1142/S012905411842008X
7. Jirásková, G.: State complexity of some operations on binary regular languages. Theor. Comput. Sci. **330**(2), 287–298 (2005). 10.1016/j.tcs.2004.04.011
8. Leiss, E.L.: Succint representation of regular languages by Boolean automata. Theor. Comput. Sci. **13**, 323–330 (1981). 10.1016/S0304-3975(81)80005-9
9. Leung, H.: Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata. SIAM J. Comput. **27**(4), 1073–1082 (1998). 10.1137/S0097539793252092
10. Leung, H.: Descriptional complexity of NFA of different ambiguity. Int. J. Found. Comput. Sci. 16(5), 975–984 (2005). 10.1142/S0129054105003418
11. Lupanov, O.B.: Uber den vergleich zweier typen endlicher quellen. Probleme der Kybernetik **6**, 328–335 (1966)
12. Maslov, A.N.: Estimates of the number of states of finite automata. Soviet Math. Doklady **11**, 1373–1375 (1970)
13. Okhotin, A.: Unambiguous finite automata over a unary alphabet. Inf. Comput. **212**, 15–36 (2012), 10.1016/j.ic.2012.01.003
14. Raskin, M.A.: A superpolynomial lower bound for the size of non-deterministic complement of an unambiguous automaton. In: Chatzigiannakis, I., Kaklamanis, C., Marx, D., Sannella, D. (eds.) ICALP 2018. LIPICS, vol. 107, pp. 138:1–138:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018). 10.4230/LIPIcs.ICALP.2018.138
15. Schmidt, E.M.: Succinctness of description of context-free, regular, and finite languages. Ph. D. thesis. Cornell University (1978)
16. Sipser, M.: Introduction to the theory of computation. Cengage Learn. (2013)
17. Stearns, R.E., Hunt, H.B.: On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. SIAM J. Comput. **14**(3), 598–611 (1985). 10.1137/0214044
18. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theor. Comput. Sci. **125**(2), 315–328 (1994). 10.1016/0304-3975(92)00011-F

Contents