COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

NONDETERMINISTIC STATE COMPLEXITY IN SUBREGULAR CLASSES

DISSERTATION THESIS

2017

Mgr. Peter Mlynárčik

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

NONDETERMINISTIC STATE COMPLEXITY IN SUBREGULAR CLASSES

DISSERTATION THESIS

Study programme:	Applied mathematics
Field of study:	9.1.9 Applied mathematics
Supervising institution:	Mathematical Institute, Slovak Academy of Sciences
Supervisor:	RNDr. Galina Jirásková, CSc.

Bratislava, 2017 Mgr. Peter Mlynárčik

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

NEDETERMINISTICKÁ ZLOŽITOSŤ V PODTRIEDACH REGULÁRNYCH JAZYKOV

DIZERTAČNÁ PRÁCA

Študijný program:	Aplikovaná matematika
Študijný odbor:	9.1.9 Aplikovaná matematika
Školiace pracovisko:	Matematický ústav Slovenskej akadémie vied
Školiteľ:	RNDr. Galina Jirásková, CSc.

Bratislava, 2017

Mgr. Peter Mlynárčik

28807588

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Mgr. Peter Mlynárčik
Study programme:	Applied Mathematics (Single degree study, Ph.D. III. deg., full time form)
Field of Study:	Applied Mathematics
Type of Thesis:	Dissertation thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Nondeterministic state complexity in subregular classes

Literature:

1. Birget, J. C.: Intersection and union of regular languages and state complexity. Inform.Process.Lett.43, 185--190 (1992)
2. Birget, J. C.: Partial orders on words, minimal elements of regular languages, and state complexity. Theoret. Comput. Sci. 119, 267--291 (1993) ERRATUM: Partial orders on words, minimal elements of regular languages, and state complexity, 2002. Available at http://clam.rutgers.edu/$\sim$birget/papers.html.
3. Brzozowski, J.A., Jirásková, G., Li, B.: Quotient complexity of ideal languages. Theoret. Comput. Sci. 470, 36--52 (2013)
4. Brzozowski, J.A., Jirásková, G., Zou, C.: Quotient complexity of closed languages. Theory Comput. Syst. 54, 277--292 (2014)
5. Cmorik, R., Jirásková, G. : Binary Operations on Binary Suffix-Free Languages. In: Z. Kotásek et al. (Eds.): MEMICS 2011, LNCS 7119, pp. 94-102. Springer - Verlag Berlin Heidelberg (2012)
6. Câmpeanu, C., Salooma, K., Yu,S.: Tight lower bound for the state complexity of shuffle of regular languages. J.Autom.Lang.Comb.7, 303--310 (2002)
7. Chrobak, M.: Finite automata and unary languages. Theoret. Comput. Sci. 47, 149--158 (1986)
8. Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. Int. J. Found. Comput. Sci. 14, 1087--1102 (2003)
9. Han, Y.-S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410, 2537--2548 (2009)
10. Han, Y.-S., Salomaa, K.: Nondeterministic state complexity for suffix-free regular languages. In: McQuillan, I., Pighizzini, G. (eds) DCFS 2010. EPTCS vol. 31, pp. 189--196 (2010)
11. Han, Y.-S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410, 2537--2548 (2009)
12. Han, Y.-S., Salomaa, K., Wood, D.: Operational state complexity of prefix-free regular languages. Ésik, Z.(ed.). Automata, formal languages, and related topics, pp. 99-115. Institute of Informatics, University of Szeged (2009)
13. Han, Y.-S., Salomaa, K., Wood, D.: Nondeterministic state complexity of basic operations for prefix-free regular languages. Fundam. Inform. 90, 93--106 (2009)

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

14. Han, Y.-S., Salomaa, K., Wood, D.: Operational state complexity of prefix-free regular languages. In: Automata, Formal Languages, and Related Topics, pp. 99--115. Institute of Informatics, University of Szeged (2009)
15. Jirásková, G.: State complexity of some operations on binary regular languages. Theoret. Comput. Sci. 330, 287-298 (2005)
16. Jirásková, G., Masopust, T.: Complexity in union-free regular languages. Internat. J. Found. Comput. Sci. 22, 1639--1653 (2011)
17. Krausová, M.: Prefix-free regular languages: Closure properties, difference and left quotient. In: Kotásek, Z. and Bouda, I. and Černá, I. and Sekanina, L. and Vojnar, T. and Antos, D. (eds.) MEMICS 2011, LNCS, vol. 7119, pp. 114--122, Springer (2012)
18. Maslov,A.N.: Estimates of the number of states of finite automata. Soviet Math. Dokl. 11, 1373--1375 (1970)
19. Mirkin,B.G.: On dual automata. Kibernetika (Kiev) 2, 7--10 (1966)(in Russian). English translation: Cybernetics 2, 6--9 (1966)
20. Rabin, M., Scott, D.: Finite automata and their decision problems. IBM Res. Develop. 3, 114--129 (1959)
21. Sipser, M.: Introduction to the theory of computation. PWS Publishing Company, Boston (1997)
22. Yu, S., Zhuang, Q., Salomaa, K.: The state complexity of some basic operations on regular languages. Theoret. Comput. Sci. 125, 315--328 (1994)

Aim: 1) Summarize known results concerning deterministic and nondeterministic complexity of basic operations in the class of regular languages and its subclasses.
2) Investigate properties of nondeterministic finite automata accepting languages in some special subregular classes (prefix-, suffix-, factor-, and subword-free languages, closed languages, convex languages and ideal languages).
3) Use the properties of nondeterministic automata to get nondeterministic complexity of operations union, intersection, concatenation, star, reversal and complementation in above mentioned subregular classes.

Annotation: We investigate properties of nondeterministic finite automata accepting languages in subregular classes of prefix-, suffix-, factor-, and subword-free languages, closed languages, convex languages and ideal languages. We use these properties to study the nondeterministic complexity of operations of union, intersection, concatenation, star, reversal, and complementation in these subregular classes. Except for complementation on factor-convex and subword-convex languages, we get the precise complexity of each operation in each of considered subregular classes. To describe worst case examples, we use small fixed alphabets in most cases, which are almost always optimal.

Keywords: : regular languages, prefix-, suffix-, factor-, and subword-free languages, closed languages, convex languages and ideal languages, nondeterministic finite automaton, regular operations, nondeterministic complexity.

Tutor: RNDr. Galina Jirásková, CSc.
Department: FMFI.KAMŠ - Department of Applied Mathematics and Statistics

28807588

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Head of department: prof. RNDr. Daniel Ševčovič, DrSc.

Assigned: 22.08.2013

Approved: 24.04.2017 prof. RNDr. Anatolij Dvurečenskij, DrSc.
Guarantor of Study Programme

....

Student

....

Tutor

28807588

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Mgr. Peter Mlynárčik
Študijný program: aplikovaná matematika (Jednoodborové štúdium, doktorandské III. st., denná forma)
Študijný odbor: aplikovaná matematika
Typ záverečnej práce: dizertačná
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Nondeterministic state complexity in subregular classes
Nedeterministická zložitosť v podtriedach regulárnych jazykoch

Literatúra:

1. Birget, J. C.: Intersection and union of regular languages and state complexity. Inform.Process.Lett.43, 185--190 (1992)
2. Birget, J. C.: Partial orders on words, minimal elements of regular languages, and state complexity. Theoret. Comput. Sci. 119, 267--291 (1993) ERRATUM: Partial orders on words, minimal elements of regular languages, and state complexity, 2002. Available at http://clam.rutgers.edu/$\sim$birget/papers.html.
3. Brzozowski, J.A., Jirásková, G., Li, B.: Quotient complexity of ideal languages. Theoret. Comput. Sci. 470, 36--52 (2013)
4. Brzozowski, J.A., Jirásková, G., Zou, C.: Quotient complexity of closed languages. Theory Comput. Syst. 54, 277--292 (2014)
5. Cmorik, R., Jirásková, G. : Binary Operations on Binary Suffix-Free Languages. In: Z. Kotásek et al. (Eds.): MEMICS 2011, LNCS 7119, pp. 94-102. Springer - Verlag Berlin Heidelberg (2012)
6. Câmpeanu, C., Salooma, K., Yu,S.: Tight lower bound for the state complexity of shuffle of regular languages. J.Autom.Lang.Comb.7, 303--310 (2002)
7. Chrobak, M.: Finite automata and unary languages. Theoret. Comput. Sci. 47, 149--158 (1986)
8. Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. Int. J. Found. Comput. Sci. 14, 1087--1102 (2003)
9. Han, Y.-S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410, 2537--2548 (2009)
10. Han, Y.-S., Salomaa, K.: Nondeterministic state complexity for suffix-free regular languages. In: McQuillan, I., Pighizzini, G. (eds) DCFS 2010. EPTCS vol. 31, pp. 189--196 (2010)
11. Han, Y.-S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410, 2537--2548 (2009)
12. Han, Y.-S., Salomaa, K., Wood, D.: Operational state complexity of prefix-free regular languages. Ésik, Z.(ed.). Automata, formal languages, and related topics, pp. 99-115. Institute of Informatics, University of Szeged (2009)
13. Han, Y.-S., Salomaa, K., Wood, D.: Nondeterministic state complexity of basic operations for prefix-free regular languages. Fundam. Inform. 90, 93--106 (2009)

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

14. Han, Y.-S., Salomaa, K., Wood, D.: Operational state complexity of prefix-free regular languages. In: Automata, Formal Languages, and Related Topics, pp. 99--115. Institute of Informatics, University of Szeged (2009)
15. Jirásková, G.: State complexity of some operations on binary regular languages. Theoret. Comput. Sci. 330, 287-298 (2005)
16. Jirásková, G., Masopust, T.: Complexity in union-free regular languages. Internat. J. Found. Comput. Sci. 22, 1639--1653 (2011)
17. Krausová, M.: Prefix-free regular languages: Closure properties, difference and left quotient. In: Kotásek, Z. and Bouda, I. and Černá, I. and Sekanina, L. and Vojnar, T. and Antos, D. (eds.) MEMICS 2011, LNCS, vol. 7119, pp. 114--122, Springer (2012)
18. Maslov,A.N.: Estimates of the number of states of finite automata. Soviet Math. Dokl. 11, 1373--1375 (1970)
19. Mirkin,B.G.: On dual automata. Kibernetika (Kiev) 2, 7--10 (1966)(in Russian). English translation: Cybernetics 2, 6--9 (1966)
20. Rabin, M., Scott, D.: Finite automata and their decision problems. IBM Res. Develop. 3, 114--129 (1959)
21. Sipser, M.: Introduction to the theory of computation. PWS Publishing Company, Boston (1997)
22. Yu, S., Zhuang, Q., Salomaa, K.: The state complexity of some basic operations on regular languages. Theoret. Comput. Sci. 125, 315--328 (1994)

Cieľ: 1) Zhrnúť známe výsledky týkajúce sa deterministickej a nedeterministickej zložitosti základných operácií v triede regulárnych jazykov a jej podtriedach.
2) Skúmať vlastnosti nedeterministických automatov akceptujúcich jazyky v niektorých špeciálnych podtriedach.
3) Využiť vlastnosti nedeterministických automatov pri zisťovaní nedeterministickej zložitosti operácií zjednotenie, prienik,, zreťazenie, Kleeneho uzáver, zrkadlový obraz, doplnok v skúmaných podtriedach

Anotácia: Skúmame vlastnosti nedeterministických konečnostavových automatov akceptujúcich jazyky v niektorých špeciálnych podtriedach regulárnych jazykov. Tieto vlastnosti využijeme pri štúdiu nedeterministickej zložitosti operácií zjednotenie, prienik, zreťa zenie, Kleeneho uzáver, zrkadlový obraz, doplnok v skúmaných podtriedach. S výnimkou doplnku v dvoch podtriedach vždy získavame presnú hodnotu zložitosti každej operácie v každej zo skúmaných podtried. Na definovanie najhorších prípadov používame väčšinou malú konštantnú abecedu, ktorá je takmer vždy optimálna.

Kľúčové slová: regulárne jazyky, podtriedy regulárnych jazykov, nedeterministický konečnostavový automat, regulárne operácie, nedeterministická zložitosť.

Školiteľ: RNDr. Galina Jirásková, CSc.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, DrSc.

Dátum zadania: 22.08.2013

Dátum schválenia: 24.04.2017

prof. RNDr. Anatolij Dvurečenskij, DrSc.
garant študijného programu

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

....

študent školiteľ

Declaration

I here declare that I have produced this dissertation thesis without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such.

This dissertation thesis was conducted from 2013 to 2017 under the supervision of RNDr. Galina Jirásková, CSc. during PhD study at the Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Slovak Republic.

Bratislava, 2017 ....

Mgr. Peter Mlynárčik

Acknowledgements

I would like to thank to my supervisor RNDr. Galina Jirásková, CSc. for all his help, valuable ideas, advice and time she has given me throughout my PhD. studies. She has been a great example how to work with high focusing on the problem with huge enthusiasm. Also she guides me in mathematic world with its correctness and pureness and beauty.

My great thanks belong also to my coleague Michal Hospodár for helping with drawing figures in better system and for discussions on many topics.

Also I thank to my colleague Matúš Palmovský for friendly cooperation during studies.

I would like to thank also to my parents and family, friends for support and life energy.

Abstract

We study the nondeterministic state complexity of the operations of intersection, union, concatenation, star, reversal, and complementation in the classes of prefix-, suffix-, factor-, and subword-free (-closed, and -convex) languages, and in the classes of right (left, two-sided, and all-sided) ideal languages. Except for complementation on factor-convex and subword-convex languages, we obtained tight upper bounds for all considered operations in all considered classes. Most of our witness languages are defined over small fixed alphabets of size at most three, and the size of these alphabets usually cannot be decreased. As for complementation, we show that the corresponding upper bounds cannot be met by any binary prefix-, suffix-, or factor-free language, and on suffix-convex languages, we prove the tightness of the upper bound 2^n, which are the most ineteresting results of this thesis.

Keywords: regular operations, subregular classes of prefix-, suffix-, factor-, and subword-free (-closed, and -convex) languages, and right (left, two-sided, and all-sided) ideal languages, nondeterministic state complexity, fooling-set lower bound method

Abstrakt

V práci študujeme nedeterministickú stavovú zložitosť operácií prienik, zjednotenie, zreťazenie, uzáver, zrkadlový obraz a doplnok v triedach bezpredponových, bezpríponových, bezfaktorových a bezpodslovových jazykov, ďalej v triedach jazykov uzavretých na predpony (prípony, faktory, podslová), predponovo-, príponovo-, faktorovo-, podslovovo-konvexných jazykov a v triedach pravých (ľavých, obojstranných, všetkostranných) ideálnych jazykov. Až na doplnok na faktorovo-konvexných a podslovovo-konvexných jazykoch, získavame vždy presné hodnoty zložitostí všetkých operácií na všetkých uvažovaných podtriedach. Na definovanie jazykov pre najhorší prípad používame skoro vždy malé konštatntné abecedy, veľkosť ktorých obvykle už nejde zmenšiť. Najzaujímavejšími výsledkami práce sú výsledky týkajúce sa doplnku na binárnych bezpredponových, bezpríponových a bezfaktorových jazykoch a na príponovo-konvexných jazykoch.

Kľúčové slová: regulárne operácie, subregulárne triedy bezpredponových, bezpríponových, bezfaktorových a bezpodslovových jazykov, triedy jazykov uzavretých na predpony (prípony, faktory, podslová), predponovo-, príponovo-, faktorovo-, podslovovo-konvexné jazyky, triedy pravých (ľavých, obojstranných, všetkostranných) ideálnych jazykov, nedeterministická stavová zložitosť, metóda klamúcej množiny

Contents

Preface

"You will come to know the truth, and the truth will set you free."
(John 8,32)

A language is a bridge to man's mind or soul. It is one of the most important treasure for people. Understanding a language means a big step to the other heart, as well as to the mine. It is not easy to describe natural language (even perhaps not possible at all), but I believe that attemption to describe formal forms helps us to be closer to language understanding. Those facts are strong motivations for me to deal with formal languages.

I am very thankful for possibility to study and work on such intersting topic with my colleagues. It gave me much in many field in my life.

I have performed a little step to understanding and also I realize how much I still do not know and I have much to do for my next way.

Introduction

Finite automata and regular languages are one of the oldest topics in formal languages theory. The basic properties of this class of languages were investigated in 1950s and 1960s. Although regular languages are the simplest languages in Chomsky hierarchy, some challenging problems are still open. The most famous is the question of how many states are sufficient and necessary for two-way deterministic automata to simulate two-way nondeterministic automata, which is connected to the well-known NLOGSPACE vs. DLOGSPACE problem [2].

In last three decades, we can observe a new interest in regular languages which have applications in software engineering, programming languages, and other areas of computer science. However, they are also interesting from the theoretical point of view [28]. Various properties of this class are now intensively studied. One of them is descriptional complexity which studies the cost of description of languages by formal systems such as deterministic and nondeterministic automata, or grammars.

Rabin and Scott in 1959 [42] defined nondeterministic finite automata (NFAs), described an algorithm known as the "subset construction" which shows that every n-state nondeterministic automaton can be simulated by at most 2^n-state deterministic finite automaton (DFA). In 1962 Yershov [46] then showed that this construction is optimal. Maslov [37] investigated the state complexity of union, concatenation, and star, and also some other operations. Birget in [3, 4] examined intersection and union. He also considered the question of the size of nondeterministic automaton for the complement of a language. The complement of a formal language L over an alphabet Σ is the language $L^c = \Sigma^* \setminus L$, where Σ^* is the set of all strings over an alphabet Σ. The complementation is an easy operation on regular languages represented by deterministic finite automata (DFAs) since to get a DFA for the complement of a regular language, it is enough to interchange the final and non-final states in a DFA for this language.

On the other hand, complementation on regular languages represented by NFAs is an expensive task. We first must apply the subset construction to a given NFA, and only after that, we may interchange the final and non-final states. This gives an upper bound 2^n.

Sakoda and Sipser [43] presented an example of languages over a growing alphabet size meeting this upper bound. Birget claimed the result for a three-letter alphabet in [4], and later corrected this to a four-letter alphabet. Holzer and Kutrib [24] obtained the lower bound 2^{n-2} for a binary n-state NFA language. Finally, binary n-state NFA languages meeting the upper bound 2^n were described by Jirásková in [30]. In the case of a unary alphabet, the complexity of complementation is in $e^{\Theta(\sqrt{n \ln n})}$ [24, 30].

Birget [3] described a lower-bound technique for proving minimality of NFAs. The technique is known as a fooling-set method. Although in some cases there is a large gap between the size of a fooling set and the size of minimal nondeterministic automaton [29], in a many other cases, the fooling sets can be used to prove the minimality of nondeterministic machines, and we successfully use this method throughout our thesis.

The systematic study of the state complexity of operations on regular languages began in the paper by Yu et al. [49]. The nondeterministic state complexity of operations was investigated by Holzer and Kutrib [24], and some improvements of their results can be found in [30]. Some special operations were examined as well: proportional removals in [14], shuffle in [10], and cyclic shift in [33].

Recently, researchers investigated subclasses of regular languages such as, for example, prefix- and suffix-free languages [13,19,22], ideal languages [7], closed languages [8], bifix-, factor-, and subword-free languages [5], union-free languages [31], or star-free languages [9]. In some of these classes, the operations have smaller complexity, while in the others, the complexity of operations is the same as in the general case of regular languages.

Prefix-free languages are used in codes like variable-length Huffman codes or country calling codes. In a prefix-code, there is no codeword which is a proper prefix of any other codeword. Therefore, a receiver can identify each codeword without any special marker between words. This was a motivation for investigating this class of languages in last few years [15, 16, 21, 23, 36].

The non-deterministic state complexity of operations on prefix-free and suffix-free languages was studied by Han *et al.* in [19–21,23]. For the nondeterministic state complexity of complementation, they obtained an upper bound $2^{n-1} + 1$ in both classes, and lower bounds 2^{n-1} and $2^{n-1} - 1$ for prefix-free and suffix-free languages, respectively. The questions of tightness remained open. In the first part of this thesis, we solve both of these open questions, and we prove that in both classes, the tight bound is 2^{n-1}. To prove tightness, we use a ternary alphabet. Hence the nondeterministic state complexity of complementation on prefix- or suffix-free languages defined over an alphabet that contains at least three symbols is given by the function 2^{n-1}. We also show that this upper bound cannot be met by any binary prefix- or suffix-free language. We get a similar result in the class of factor-free languages, and moreover we obtain the tight upper bounds on the nondeterministic complexity of each considered operation in each of the four free classes. We also study the unary free languages, and, besides some other results, we prove that the nondeterministic state complexity of complementation is in $\Theta(\sqrt{n})$ in the each of the four classes of free languages.

Then we deal with the operations of intersection, union, concatenation, star, reversal, and complementation on prefix-, suffix-, factor-, and subword-closed languages, and on right (left, two-sided, and all-sided) ideal languages. In all cases, we get tight upper bounds on the nondeterministic complexity for all operations. Except for three cases, our witnesses are defined over small fixed alphabets.

Finally, we use our results to show that the nondeterministic complexities of basic regular operations, except for complementation, in the classes of prefix-, suffix-, factor-, and subword-convex languages are the same as in the general case of regular languages. As for complementation, the complexity in the class of suffix-convex languages is 2^n which is one of the most interesting results of this thesis. A curious reader is referred to read Chapter 7 to find out why this is the case :).

All the results of this thesis, except for the results on convex languages and some results on free languages, have been already published in three papers (co-authored) by the author of this thesis given in the list of author's publications, and presented at international conferences DCFS 2014, 2015 and CIAA 2016. The paper on the results for convex and free languages has been accepted to the conference CIAA 2017 in Paris, and will be presented by the author (if nothing unexpected happens).

Chapter 1

Preliminaries

We use a standard model of a nondeterministic finite automaton (NFA), as explained, for example, in [44]. For details, the reader may refer to [25, 44, 48].

Let Σ be a finite non-empty alphabet of symbols. Then Σ^* denotes the set of strings over the alphabet Σ including the empty string ε. The length of a string w is denoted by $|w|$, and the number of occurrences of a symbol a in a string w by $|w|_a$. A language is any subset of Σ^*. For a finite set X, the cardinality of X is denoted by $|X|$, and its power-set by 2^X.

For a language L over an alphabet Σ, the complement of L is the language $L^c = \Sigma^* \setminus L$. The *intersection* of languages K and L is the language $K \cap L = \{w \mid w \in K \text{ and } w \in L\}$. The *union* of languages K and L is the language $K \cup L = \{w \mid w \in K \text{ or } w \in L\}$. The *concatenation* of languages K and L is the language $KL = \{uv \mid u \in K \text{ and } v \in L\}$. The *power* of a language L is the language $L^k = LL^{k-1}$, where $L^0 = \{\varepsilon\}$. The *star* of a language L is the language $L^* = \bigcup_{i \geq 0} L^i$. The reversal of a string is defined as $\varepsilon^R = \varepsilon$ and $(wa)^R = aw^R$ for each symbol a and string w. The reversal of a language L is the language $L^R = \{w^R \mid w \in L\}$. The *shuffle* $u \sqcup\!\sqcup v$ of strings $u, v \in \Sigma^*$ is defined as follows:

$$u \sqcup\!\sqcup v = \{u_1 v_1 \cdots u_k v_k \mid u = u_1 \cdots u_k, v = v_1 \cdots v_k, u_1, \ldots, u_k, v_1, \ldots, v_k \in \Sigma^*\}.$$

The shuffle of two languages K and L over Σ is defined by

$$K \sqcup\!\sqcup L = \bigcup_{u \in K, v \in L} u \sqcup\!\sqcup v.$$

A *nondeterministic finite automaton* (NFA) is a quintuple $A = (Q, \Sigma, \cdot, s, F)$, where Q is a finite non-empty set of states, Σ is a finite non-empty input alphabet, $\cdot$ is the transition function that maps $Q \times \Sigma$ to 2^Q, $s \in Q$ is the start (or initial) state, and

$F \subseteq Q$ is the set of final (or accepting) states. In this thesis we also use another notation of transition function, as δ or $\circ$. The transition function is extended to the domain $2^Q \times \Sigma^*$ in the natural way. The language accepted by the NFA A is the set of strings $L(A) = \{w \in \Sigma^* \mid s \cdot w \cap F \neq \emptyset\}$.

An NFA $A = (Q, \Sigma, \cdot, s, F)$ is a (complete) *deterministic finite automaton* (DFA) if for each state q and each input symbol a, the set $q \cdot a$ has exactly one element. In such a case, we write $p \cdot a = q$ instead of $p \cdot a = \{q\}$. If $|q \cdot a| \leq 1$ for each q and a, then A is an *incomplete deterministic finite automaton* (IDFA) Notice that every DFA can be considered to be incomplete. Next, the number of states in the minimal complete and incomplete DFAs for the same language differ by at most one.

Two automata are *equivalent* if they accept the same language. A DFA (an NFA) A is *minimal* if there is no equivalent DFA (NFA) with a smaller number of states than A has. It is known that every regular language has a unique, up to isomorphism, minimal DFA however, this is not true for NFAs.

The *state complexity* of a regular language L, (L), is the number of states in the minimal DFA for L.

Sometimes, we allow an NFA to have multiple initial states and use the notation NNFA (an NFA with a nondeterministic choice of initial states) for this model [48]. A *nondeterministic finite automaton* (NNFA) is a 5-tuple $A = (Q, \Sigma, \cdot, I, F)$, where $Q, \Sigma, \cdot, F$ are the same as for NFA, and $I \subseteq Q$ is the set of initial states. The *language accepted by* NNFA A is the set $L(A) = \{w \in \Sigma^* \mid I \cdot w \cap F \neq \emptyset\}$.

For an easier description of some constructions, we use ε-model of NFA, denoted as ε-NFA, where we also allow the transitions on the empty string. It is known that every ε-NFA can be converted to an equivalent NFA without increasing the number of states. The reader can find more detailed conversion from ε-NFA to NFA in Chapter 2.

We call a state of an NNFA *sink state* if it has a loop on every input symbol. From every final sink state, every string is accepted, but from every non-final sink state, no string is accepted. Notice that every minimal IDFA has no non-final sink states, and every minimal DFA has at most one non-final sink state. A state q of an NFA A is called a *dead state* if no string is accepted by A from q, that is, if $q \cdot w \cap F = \emptyset$ for each string w. An NFA A is a *trim* NFA if each its state q is reachable, that is, there is a string u in Σ^* such that $q \in s \cdot u$, and, moreover, no state of A is dead.

For a symbol a and states p and q, we say that (p, a, q) is a transition in the NNFA A if $q \in p \cdot a$, and for a string w, we write $p \xrightarrow{w} q$ if $q \in p \cdot w$. We also say that the state q has an *in-transition* on symbol a, and the state p has an *out-transition* on symbol a. A *state is non-exiting* if it does not have any out-transitions. An NFA is *non-returning* if its initial state does not have any in-transitions, and it is *non-exiting* if each final state of A does not have any out-transitions.

Every NNFA $A = (Q, \Sigma, \cdot, I, F)$ can be converted to an equivalent deterministic automaton $A' = (2^Q, \Sigma, \circ, I, F')$, where $S \circ a = S \cdot a$ for each S in 2^Q and a in Σ, and $F' = \{S \in 2^Q \mid S \cap F \neq \emptyset\}$. We call the DFA A' the *subset automaton* of the NNFA A. The subset automaton may not be minimal since some of its states may be unreachable or equivalent to other states.

The nondeterministic state complexity of a regular language L, $\text{nsc}(L)$, is the smallest number of states in any NFA for L. To prove the minimality of NFAs, we use a *fooling set* lower-bound technique [3,4,18,27]. We describe this technique in detail in Chapter 2; here we only give the definition of a fooling set.

Definition 1.1. A set of pairs of strings $\{(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)\}$ is called a *fooling set* for a language L if for all i, j in $\{1, 2, \ldots, n\}$,

(F1) $u_i v_i \in L$,

(F2) if $i \neq j$, then $u_i v_j \notin L$ or $u_j v_i \notin L$.

The *reverse of an automaton* $A = (Q, \Sigma, \cdot, I, F)$ is the NNFA A^R obtained from A by swapping the role of initial and final states and by reversing all the transitions. Formally, we have $A^R = (Q, \Sigma, \cdot^R, F, I)$, where $q \cdot^R a = \{p \in Q \mid q \in p \cdot a\}$ for each state q in Q and each symbol a in Σ. The NFA A^R accepts the language $L(A)^R$.

Let $A = (Q, \Sigma, \cdot, I, F)$ be an NNFA and $S, T \subseteq Q$. We say that S is *reachable* in A if there is a string w in Σ^* such that $S = I \cdot w$. Next, we say that T is *co-reachable* in A if T is reachable in A^R.

If $u, v, w, x \in \Sigma^*$ and $w = uxv$, then u is a *prefix* of w, x is a *factor* of w, and v is a *suffix* of w. Both u and v are also factors of w. If $w = u_0 v_1 u_1 \cdots v_n u_n$, where $u_i, v_i \in \Sigma^*$, then $v_1 v_2 \cdots v_n$ is a *subword* of w. For example, let $w = abbacb$. Strings $abac, bbb, bc$ are subwords of w, but string aca is not a subword of w. Every factor of w is also a subword of w. A prefix v (suffix, factor, subword) of w is *proper* if $v \neq w$.

A language L is *prefix-free* if $w \in L$ implies that no proper prefix of w is in L; it is *prefix-closed* if $w \in L$ implies that each prefix of w is in L; and it is *prefix-convex* if $u, w \in L$ and u is a prefix of w imply that each string v such that u is a prefix of v and v is a prefix of w is in L. Suffix-, factor-, and subword-free, -closed, and -convex languages are defined analogously.

A language L is a right (respectively, left, two-sided, all sided) *ideal* if $L = L\Sigma^*$ (respectively, $L = \Sigma^* L, L = \Sigma^* L \Sigma^*, L = L \sqcup\!\sqcup \Sigma^*$).

We say that a regular language is a *free language* if it is either prefix-free, or suffix-free, or factor-free, or subword-free. Let us emphasize that we do not consider star-free, or union-free, or any other free languages in this thesis. In an analoguous way, we use the notions of *closed languages*, *convex languages*, and *ideal languages*.

Notice that the classes of prefix-free, prefix-closed, and ideal languages are subclasses of convex languages and the complement of a closed language is an ideal language.

If languages K and L are accepted by NFAs $A = (\{0, 1, \ldots, m-1\}, \Sigma, \cdot_A, 0, F_A)$ and $B = (\{0, 1, \ldots, n-1\}, \Sigma, \cdot_B, 0, F_B)$, respectively, then the language $K \cap L$ is accepted by the *product automaton*

$$A \times B = (\{0, 1, \ldots, m-1\} \times \{0, 1, \ldots, n-1\}, \Sigma, \cdot, (0, 0), F_A \times F_B),$$

where $(p, q) \cdot a = (p \cdot_A a) \times (q \cdot_B a)$. We call the set of states $\{(i, 0) \mid 0 \leq i \leq m-1\}$ the first column of the product automaton. The first row and the last row/column are the sets $\{(0, j) \mid 0 \leq j \leq n-1\}$ and $\{(m-1, j) \mid 0 \leq j \leq n-1\}$/ $\{(i, n-1) \mid 0 \leq i \leq m-1\}$, respectively.

It is known that every unary n-state NFA can be transformed to *Chrobak normal form* which consist of a simple path containing not more than $2n^2+n$ states, ending with a state in which there is a nondeterministic choice to several pairwise disjoint cycles containing at most n states [12, 45]; see Figure 3.2.

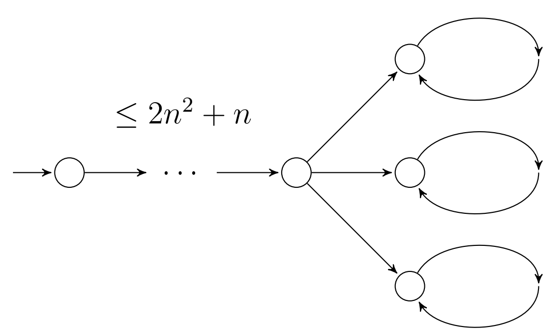

Figure 1.1: Every unary automaton can be transformed to Chrobak normal form

Chapter 2

Upper and Lower Bound Methods

In this thesis we consider unary or binary regular operation as union, intersection, concatenation, star, reversal, complementation. The *nondeterministic state complexity* of a binary regular operation $\circ$ is a function $f\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined as

$$f(m,n) = \max\{\text{nsc}(K \circ L) \mid \text{nsc}(K) = m \text{ and } \text{nsc}(L) = n\}$$

and the nondeterministic state complexity of a unary operation is a function $f\colon \mathbb{N} \to \mathbb{N}$ defined as

$$f(n) = \max\{\text{nsc}(\circ(L)) \mid \text{nsc}(L) = n\}.$$

To find the nondeterministic state complexity of the binary operation $\circ$, it is necessary to prove that:

(1) for all integers m, n and all languages K, L such that $\text{nsc}(K) = m, \text{nsc}(L) = n$ we have $\text{nsc}(K \circ L) \leq f(m,n)$, so we say that f is an *upper bound* on the nondeterministic complexity of $\circ$;

(2) for all integers m, n there exist languages K, L such that $\text{nsc}(K) = m, \text{nsc}(L) = n$ and $\text{nsc}(K \circ L) = f(m,n)$, so we say that f is a *lower bound* on the nondeterministic complexity of $\circ$. The languages K, L are called *wittnes languages* for the operation $\circ$.

For a unary operation, the task to find the nondeterministic state comlexity is described analogously.

The nondeterministic complexity of an operation may depend on the size of alphabet over which languages are considered. We usually try to describe wittnes languages over as small alphabet as possible. Moreover, it is not necessary to verify the minimality of NFAs for our wittnes languages because the nondeterministic complexity of each operation is an increasing function in both m, n.

Sometimes the upper and lower bound may be different, but in this thesis all our upper and lower bounds coincide, except for complementation on factor-convex and subword-convex languages. Thus we almost always obtain the exact nondeterministic state complexity of each operation on each considered class.

We now describe a very useful tool for estimation of a lower bound on the number of states in NNFAs based on fooling set techniques [3, 4, 18, 27]. Recall that set of pairs of strings $\{(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)\}$ is called a *fooling set* for a language L if for all i, j in $\{1, 2, \ldots, n\}$,

(F1) $u_i v_i \in L$,

(F2) if $i \neq j$, then $u_i v_j \notin L$ or $u_j v_i \notin L$.

Lemma 2.1 ([3, Lemma 1], **Lower bound method for NNFAs**). *Let $\mathcal{F}$ be a fooling set for a language L. Then every NNFA for the language L has at least $|\mathcal{F}|$ states.*

Proof. Let $A = (Q, \Sigma, \delta, I, F)$ be NFA, such that $L(A) = L$ and let $\mathcal{F} = \{(x_i, y_i) \mid 1 \leq i \leq n\}$ be the fooling set for language L. Let us assume for contradiction, that $|Q| < |\mathcal{F}|$. Fix an accepting computation of A on every $x_i y_i$, for $1 \leq i \leq n$. Let q_i be the state on this computation reached after reading x_i. So there are n such fixed states $q_1, q_2, \ldots, q_n$. Since $n > |Q|$, there are i, j such that $i \neq j$ and $q_i = q_j$. See Figure. 2.1. So there are two accepted computations $x_i y_j$ and $x_j y_i$ which is contradiction with the property (F2) of the fooling set.

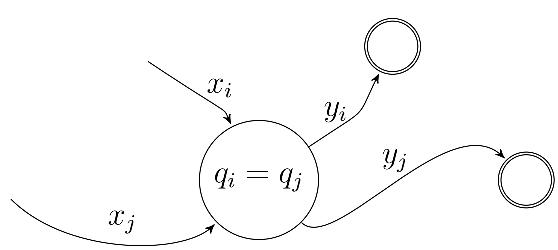

Figure 2.1: The number of states of NFA recocnizing a language L cannot be less than number of pairs in fooling set for L.

□

Let n be an even integer and consider the binary language

$$L_n = \{uv \in \{a, b\}^* \mid |u| = |v| = n, u \neq v or u = ww\}.$$

In [29] Jirásková showed that every fooling set for the language L_n is of size $O(n^2)$, while every NFA for L_n has at least $2^{n/6}$ states. It follows that the fooling set method described by Lemma 2.1 may fail significantly in some cases. Nevertheless, we use a this method successfully throughout this thesis to get the exact nondeterministic complexity of basic operations in subregular classes in most of considered 96 cases.

Let us emphasize that the size of a fooling set for L provides a lower bound on the number of states in any NNFA for L. If we insist on having just one initial state, then the following modification of a fooling set method can be used.

Lemma 2.2 ([31, Lemma 4], **Lower bound method for NFAs**). *Let $\mathcal{A}$ and $\mathcal{B}$ be sets of pairs of strings and let u and v be two strings such that $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, u)\}$, and $\mathcal{B} \cup \{(\varepsilon, v)\}$ are fooling sets for a language L. Then every* NFA *for L has at least $|\mathcal{A}| + |\mathcal{B}| + 1$ states.*

Now we present the known result that for each ε-NFA, there exists an NFA that accepts the same language. We provide a more detailed proof here. First, we give the following definition.

Definition 2.1. Let $A = (Q, \Sigma, \cdot, s, F)$ be a ε-NFA. The *ε-closure* of a state $q \in Q$, denoted $\varepsilon-$closure(q), is the set of all states that are reachable from q by zero or more ε-transitions.

Theorem 2.3 ([47], Lemma 4). *For each ε-NFA A, there exists an NFA A' such that $L(A) = L(A')$. Moreover, the NFA A' has the same number of states as the ε-NFA A.*

Proof. Let $A = (Q, \Sigma, \cdot, s, F)$ be a ε-NFA. We construct an NFA $A' = (Q, \Sigma, \circ, s, F')$ where for each $q \in Q$ and $a \in \Sigma$,

$$p \circ a = \bigcup_{q \in \varepsilon-\text{closure}(p)} q \cdot a$$

and

$$F' = \{q \in Q \mid \varepsilon-\text{closure}(q) \cap F \neq \emptyset\}$$

First, we show $L(A) \subseteq L(A')$.
Let $w \in L(A)$, $w = a_1 a_2 \cdots a_k$ where $a_i \in \Sigma$ for each i. Then there are sequences of states $q_0, q_1, \ldots, q_n$ and symbols or empty strings $x_1, x_2, \ldots, x_n$ such that

$$q_0 \xrightarrow{x_1} q_1 \xrightarrow{x_2} \cdots \xrightarrow{x_{n-1}} q_{n-1} \xrightarrow{x_n} q_n$$

is an accepting computation in A, where each $q_i \in Q$ and $u_i \in \Sigma \cup \{\varepsilon\}$, q_0 is an initial state, $q_n \in F$, and $x_1 x_2 \cdots x_n = a_1 a_2 \cdots a_k$. There is a sequence $1 \leq i_1 < i_2 < \ldots < i_k \leq n$, such that for each i_j, $x_{i_j} = a_j$. Then in the accepting computation mentioned above are parts of the following form $q_0 \cdots q_{i_j - 1} \xrightarrow{x_{i_j}} q_{i_j} \xrightarrow{\varepsilon} \cdots \xrightarrow{\varepsilon} q_{i_{j+1}-1} \xrightarrow{x_{i_{j+1}}} q_{i_{j+1}} \cdots q_n$, except the case that $n > i_k$, when the computation ends with $q_{i_k} \xrightarrow{\varepsilon} \cdots \xrightarrow{\varepsilon} q_n$.

Now, let us find the appropriate accepting computation in A'. Since $q_{i_{j+1}-1}$ is in $\varepsilon-$closure(q_{i_j}) and $q_{i_{j+1}} \in q_{i_{j+1}-1} \cdot a_{j+1}$, in A' there is a transition $q_{i_j} \xrightarrow{a_{j+1}} q_{i_{j+1}}$. Let us consider the marginal cases, the begining of computation and finishing of computation. Since the state $q_{i_1 - 1} \in \varepsilon-$closure$(q_0)$ and $q_{i_1} \in q_{i_1-1} \cdot a_1$, in A' there is transition $q_0 \xrightarrow{a_1} q_{i_1}$. If $n > i_k$, then $q_{i_k} \in F'$. Finally, there is accepting computation in A': $q_0, q_{i_1}, \ldots, q_{i_k}$ such that $q_{i_{j+1}} \in q_{i_j} \circ a_{j+1}$, so $a_1 a_2 \cdots a_k$ is accepted by A'. Hence $w \in L(A')$.

Now, we show $L(A') \subseteq L(A)$. Let $w \in L(A'), w = a_1 a_2 \cdots a_k$. Then there is a sequence of states $q_0, q_1, \ldots, q_k$ such that $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_k} q_k$ is an accepting computation in A'. If any transition $q_i \xrightarrow{a_{i+1}} q_{i+1}$ does not exist in A, there are t_i states $p^i_1, p^i_2, \ldots, p^i_{t_i}$, such that $q_i \xrightarrow{\varepsilon} p^i_1 \xrightarrow{\varepsilon} p^i_2 \xrightarrow{\varepsilon} \cdots \xrightarrow{\varepsilon} p^i_{t_i} \xrightarrow{a_{i+1}} q_{i+1}$. If $q_i \xrightarrow{a_{i+1}} q_{i+1}$ exists in A, then $t_i = 0$. If $q_k \in F'$ and $q_k \notin F$, then there exists $q_{k+1} \in F$ and states $p^k_1, p^k_2, \ldots, p^k_{t_k}$ such that $q_k \xrightarrow{\varepsilon} p^k_1 \xrightarrow{\varepsilon} p^k_2 \xrightarrow{\varepsilon} \cdots \xrightarrow{\varepsilon} p^k_{t_k} \xrightarrow{\varepsilon} q_{k+1}$. So there is accepting computation in A on string $\varepsilon^{t_0} a_1 \varepsilon^{t_1} a_2 \cdots a_k \varepsilon^{t_k}$, hence $w \in L(A)$. □

The next observations are used throughout this paper. Recall that a subset S of the state set of an NNFA $A = (Q, \Sigma, \cdot, I, F)$ is reachable if $S = I \cdot w$ for some string w, and it is co-reachable if it is reachable in the NNFA A^R.

Proposition 2.4. *Let T be a co-reachable set in an* NNFA $A = (Q, \Sigma, \cdot, I, F)$. *Then there is a string w in Σ^* such that w is accepted by A from each state in T and rejected from each state in T^c.*

Proof. Let T be a a co-reachable set in A. Then T is reachable in A^R, so there is a string v such that $F \cdot^R v = T$. Set $w = v^R$. Then w is the desired string. □

Lemma 2.5. *Let A be an* NNFA. *Let for each state q of A, the singleton set $\{q\}$ is reachable as well as co-reachable in A. Then A is minimal.*

Proof. Let $A = (Q, \Sigma, \cdot, I, F)$. Since $\{q\}$ is reachable in A, there is a string u_q such that $I \cdot u_q = \{q\}$. Since $\{q\}$ is co-reachable in A, by Proposition 2.4, there is a string v_q accepted by A from and only from the state q. Then $\{(u_q, v_q) \mid q \in Q\}$ is a fooling set for $L(A)$. By Lemma 2.1, the NNFA A is minimal. □

Recall that an NFA A is trim if every state of A is reachable and useful.

Notice that if A is a trim incomplete DFA, then for each state q of A, the singleton set $\{q\}$ is reachable. If moreover A^R is an incomplete DFA, then $\{q\}$ is co-reachable in A. So we get the following result.

Lemma 2.6. *Let A be a trim* NFA. *If both A and A^R are incomplete* DFAs, *then A and A^R are minimal* NFAs. □

Proposition 2.7. *Let L be a language accepted by an n-state* NFA *in which each subset of the state set is reachable and co-reachable. Then* $\text{nsc}(L^c) = 2^n$.

Proof. Let $A = (Q, \Sigma, \cdot, s, F)$ be an n-state NFA and $S \subseteq Q$. Since S is reachable, there exists a string u_S in Σ^* such that $s \cdot u_S = S$. Next, the set S^c is co-reachable. By Proposition 2.4, there is a string v_S which is accepted by A from each state in S^c, but rejected from each state in S. It follows that $\{(u_S, v_S) \mid S \subseteq Q\}$ is a fooling set for L^c of size 2^n. Hence $\text{nsc}(L^c) \geq 2^n$ by Lemma 2.1. □

Chapter 3

Known Results

A language is regular if it is accepted by some deterministic or nondeterministic finite automaton. In 1959 Rabin and Scott [42] provided an algorithm called "subset construction" for the conversion of an NFA to an equivalent DFA. It follows from this algorithm that if a given NFA has n states, then the resulting DFA has at most 2^n state. A binary n-state witness NFA meeting the upper bound 2^n was presented in 1962 by Yershov [46]. Some other witnesses were described in 1963 by Lupanov, and in the western literature, in 1971 by Moore [40] and Meyer, Fisher [38].

The reverse of an NFA N was defined by Rabin and Scott [42] as the NFA N^R obtained from N by swapping the roles of the initial and final states and by reversing all the transitions, and it was shown by them that the NFA N^R accepts the reverse of the language $L(N)$.

In 1970 Maslov [37] studied the state complexity of union, concatenation, star, and some other regular operations. He also provided a general statement of the problem as follows: Let f be a k-ary regular operation, and let languages $L_1, L_2, \ldots, L_k$ be represented by automata $A_1, A_2, \ldots, A_k$ with $n_1, n_2, \ldots, n_k$ states, respectively. What is the maximal number of states of a minimal automaton recognizing the language $f(L_1, L_2, \ldots, L_k)$ for given $n_1, n_2, \ldots, n_k$?

Birget [3] studied the state complexity of the intersection and union of k $(2 \leq k \leq n)$ languages each of which has an n-state DFA or NFA. He obtained tight upper bounds in both cases. For tightness he used a ternary alphabet in the deterministic case, and a quaternary alphabet in the nondeterministic case. He also provided a useful lower bound method for the number of states in NFAs known as *fooling set method.* In 1993 the same author [4] described a quaternary language accepted by an n-state NFA such that every NFA for its complement has at least 2^n states. The bound 2^n is also an upper bound for complementation of languages represented by NFAs since given an n-state NFA for a language L, we first apply the subset construction to this NFA, and then we interchange the final and non-final states to get a DFA (and therefore also an NFA) of 2^n states for L^c.

In 1994 Yu, Zhuang, and K. Salomaa [49] initiated the systematic study of the state complexity of regular operations. Their paper was followed by several papers examining the state complexity of operations on subregular classes. Unary languages were studied by Pighizzini and Shallit [41], finite languages by Câmpeanu, Salomaa, and Yu [10], prefix-free languages by Han, K. Salomaa, and Wood [22], suffix-free languages by Han and K. Salomaa [19], ideal languages by Brzozowski, Jirásková, and Li [7], closed languages by Brzozowski, Jirásková, and Zou [8], and bifix-, factor-, and subword-free languages by Brzozowski, Jirásková, Li, and Smith [5]. The results for ideal, closed, and free languages are summarized in Tables 3.1 and 3.2, respectively.

	$K \cap L$	$\|\Sigma\|$	$K \cup L$	$\|\Sigma\|$
Right ideal	mn	2	$mn - (m + n - 2)$	2
Left ideal	mn	2	mn	4
Two-sided ideal	mn	2	$mn - (m + n - 2)$	2
All-sided ideal	mn	2	$mn - (m + n - 2)$	2
Prefix-closed	$mn - (m + n - 2)$	2	mn	2
Suffix-closed	mn	4	mn	2
Factor-closed	$mn - (m + n - 2)$	2	mn	2
Subword-closed	$mn - (m + n - 2)$	2	mn	2
Prefix-free	$mn - 2(m + n - 3)$	2	$mn - 2$	2
Suffix-free	$mn - 2(m + n - 3)$	2	$mn - (m + n - 2)$	2
Factor-free	$mn - 3(m + n - 4)$	2	$mn - (m + n)$	3
Subword-free	$mn - 3(m + n - 4)$	$m + n - 7$	$mn - (m + n)$	$m + n - 3$
Regular	mn	2	mn	2
Unary ideal	$\max\{m, n\}$		$\min\{m, n\}$	
Unary closed	$\min\{m, n\}$		$\max\{m, n\}$	
Unary free	n if $m = n$		$\max\{m, n\}$	
Unary regular	mn		mn	

Table 3.1: State complexity of boolean operations on subregular classes from [5, 7, 8].

	KL	$\|\Sigma\|$	L^*	$\|\Sigma\|$	L^R	$\|\Sigma\|$
Right ideal	$m+2^{n-2}$	2	$n+1$		2^{n-1}	2
Left ideal	$m+n-1$	1	$n+1$	2	$2^{n-1}+1$	3
Two-sided ideal	$m+n-1$		$n+1$	2	$2^{n-2}+1$	3
All-sided ideal	$m+n-1$	1	$n+1$	2	$2^{n-2}+1$	$2n-4$
Prefix-closed	$(m+1)2^{n-2}$	3	$2^{n-2}+1$	3	2^{n-1}	2
Suffix-closed	$(m-1)n+1$	3	n	2	$2^{n-1}+1$	3
Factor-closed	$m+n-1$	2	2	2	$2^{n-2}+1$	3
Subword-closed	$m+n-1$	2	2	2	$2^{n-2}+1$	$2n$
Prefix-free	$m+n-2$	1	n	2	$2^{n-2}+1$	3
Suffix-free	$(m-1)2^{n-1}+1$	3	$2^{n-2}+1$	2	$2^{n-2}+1$	3
Factor-free	$m+n-2$	1	$n-1$	2	$2^{n-3}+2$	3
Subword-free	$m+n-2$	1	$n-1$	2	$2^{n-3}+2$	$2^{n-3}-1$
Regular	$m2^n-2^{n-1}$	2	$2^{n-1}+2^{n-2}$	2	2^n	2
Unary ideal	$m+n-1$		n		n	
Unary closed	$m+n-2$		1		n	
Unary free	$m+n-2$		$n-2$		n	
Unary regular	mn		$(n-1)^2$		n	

Table 3.2: State complexity of concatenation, star, and reversal on subregular classes from [5, 7, 8].

In 2010 Brzozowski [6] studied convex languages. He observed that the state complexity of union and intersection in all four convex classes is mn because this is an upper bound for regular languages and it is met by all-sided ideals (so subword-convex languages) for intersection, and by subword-closed (so subword-convex) languages for union. The state complexity of concatenation, star, and reversal on convex languages is not known.

In 2003 Holzer and Kutrib [24] investigated the complexity of basic operations on languages represented by nondeterministic finite automata. They obtained tight upper bounds in most cases. Their results for reversal and complementation were improved by Jirásková [30] by providing binary witness languages for these two operations. All these results on the nondeterministic state complexity of basic operations on regular and unary regular languages are summarized in Table 3.3.

	Regular	$\lvert\Sigma\rvert$	Source	Unary regular	Source
$K \cap L$	mn	2	[24, Theorem 3]	mn; $\gcd(m,n)=1$	[24, Theorem 4]
$K \cup L$	$m+n+1$	2	[24, Theorem 1]	$m+n+1$; $\gcd(m,n)=1$	[24, Theorem 2]
KL	$m+n$	2	[24, Theorem 7]	$\geq m+n-1$ $\leq m+n$	[24, Theorem 8]
L^*	$n+1$	1	[24, Theorem 9]	$n+1$	[24, Theorem 9]
L^R	$n+1$	2	[30, Theorem 2]	n	
L^c	2^n	2	[30, Theorem 5]	$2^{\Theta(\sqrt{n \log n})}$	[12, Theorem 4.5]

Table 3.3: The nondeterministic complexity of operations on regular languages; sources from Chrobak [12], Holzer and Kutrib [24], and Jirásková [30].

We use the following result from [30] several times in this thesis. Therefore we give a sketch of its proof here. A detailed proof can be found in [30, Theorem 5].

Theorem 3.1 ([30, Theorem 5]). *Let G be a binary language accepted by the NFA A shown in Fig. 3.1. Then every NFA for the language G^c requires 2^n states.*

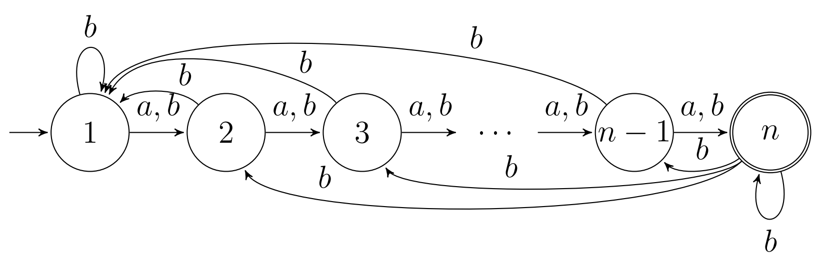

Figure 3.1: The NFA A of the binary regular language G from [30] with $\mathrm{nsc}(G^c) = 2^n$

Sketch of proof. Our aim is to describe a fooling set for the language G^c. Let us consider the set of states $\{1, 2, \ldots, n\}$ in the NFA A. We describe two strings u_S and v_S for every subset S of $\{1, 2, \ldots, n\}$ such that $\mathcal{F} = \{(u_S, v_S) \mid S \subseteq \{1, 2, \ldots, n\}\}$ is a fooling set for L^c.

We first show that every subset of $\{1, 2, \ldots, n\}$ is reachable from 1. Every singleton $\{i\}$ is reached from 1 state by the string a^{i-1} and the emptyset is reached from n by a.

Every set $\{i_1, i_2, \ldots, i_k\}$ of size k, where $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ is reached from the set $\{i_2 - i_1, i_3 - i_1, \ldots, i_k - i_1\}$ of size $k-1$ by the string $ba^{i_1 - 1}$. This proves the reachability of all subsets by induction. It follows that for every subset S of $\{1, 2, \ldots, n\}$, there is a string u_S such that the state 1 goes to the set S after reading u_S in NFA A.

Now we are going to define the strings v_S. If $S = \{1, 2, \ldots, n\}$, then define $v_S = a^n$. Otherwise, let k be the minimal state that is not in S, that is $\{1, 2, \ldots, k-1\} \subseteq S$ and $k \notin S$. Then the string v_S of length $n - k$ is defined as $v_S = v_0 v_1 \cdots v_{n-k-1}$, where

$$v_i = \begin{cases} a, & \text{if } n - i \in S \\ b, & \text{if } n - i \notin S \end{cases}$$

Then:

(1) if $p \in S$, then the string v_S is rejected by the NFA A from the state p.

(2) if $p \notin S$, then string v_S is accepted by the NFA A from the state p.

The proof of given claims is in [30, Theorem 5]. □

Chrobak [12] defined so called *Chrobak normal form* of unary NFAs. A unary NFA is said to be in the Chrobak normal form if it consists of a simple path ending with a state in which there is a nondeterministic choice to several pairwise disjoint cycles; see Figure 3.2 for an illustration.

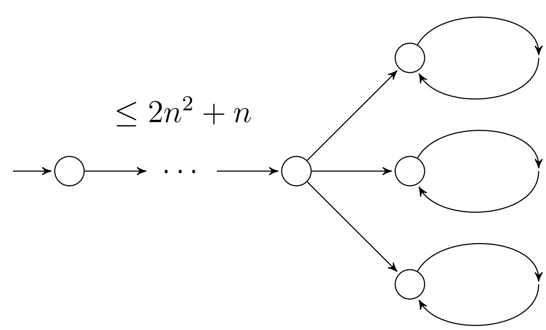

Figure 3.2: Every unary automaton can be transformed to Chrobak normal form

We use the following result from [45] later is in this thesis.

Theorem 3.2 (cf. [45, Theorem 1]). *Every unary n-state NFA can be converted to Chrobak normal form which consist of a simple path containing not more than $2n^2+n$ states, ending with a state in which there is a nondeterministic choice to several pairwise disjoint cycles containing at most n states.* □

Chapter 4

Free Languages

In this chapter we study the nondeterministic complexity of basic operations in the classes of prefix-, suffix-, factor-, and subword-free languages. Recall that a language is prefix-free if it does not contain two distinct strings such that one of them is a prefix of the other. Suffix-, factor-, and subword-free languages are defined analogously. We use the notion of a free language for a language belonging to one of these four classes.

For each considered operation, we obtain tight upper bound in each class. To prove tightness, we always use an optimal fixed alphabet of size at most three, except for intersection and complementation on subword-free languages where we use a growing alphabet.

4.1 Properties of free languages

We start with recalling the characterization of minimal deterministic automata accepting prefix-free languages. Since we use this characterization several times in this chapter, we provide a detailed proof here.

Proposition 4.1 ([22] **Characterization of prefix-free DFAs).** *Let $n \geq 2$ and $A = (Q, \Sigma, \delta, s, F)$ be a minimal n-state DFA for a language L. Then L is prefix-free if and only if A has a dead state q_d and exactly one final state q_f such that $\delta(q_f, a) = q_d$ for each a in Σ.*

Proof. $\Rightarrow$: Let L be a prefix-free language accepted by an DFA A. Let $q \in F$. For the sake of contradiction assume that there is a symbol $a \in \Sigma$ such that $\delta(q, a) = p$ and p is not a dead state. Then there is a string u such that $\delta(p, u) \in F$. On the other hand the state q is reachable from the initial state, so there is a string v such that $\delta(s, v) = q$. Thus we have $v \in L$ and also $vau \in L$, where v is proper prefix of vau and it is contradiction. Hence, from each final state every transition goes to the dead state. Therefore all final states are equivalent. Since A is minimal, there is just one final state in A.

$\Leftarrow$: The single final state going on every input to the dead state indicates that no string can be extended to be accepted, so it cannot be a prefix of some longer string. Now let us prove it more formally. Let w be a string, such that $w \in L$. Since f is the only final state, we have $\delta(s, w) = f$. Let u, v be strings such that $w = uv$ and $v \neq \varepsilon$. The prefix $u \notin L$, because otherwise we would have $\delta(s, u) = f$ and $\delta(f, v) = f$, which is contradiction with assumption that final state goes on every input to the dead state. $\square$

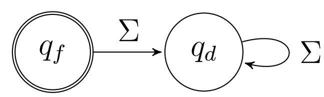

Figure 4.1: Every minimal DFA recognizing a prefix-free language has just one final state, from which every transition goes to a dead state.

Now we state a necessary condition for an NFA to accept a prefix-free language.

Proposition 4.2 (Neccessary conditions for prefix-free NFA). *Let $N = (Q, \Sigma, \delta, s, F)$ be a minimal NFA for a prefix-free language. Then N has exactly one final state q_f and $\delta(q_f, a) = \emptyset$ for each a in Σ.*

Proof. It is not possible to reach some final state from any final state by any a nonempty string. The reason is the same as in deterministic case shown above. So from every final state no nonempty string is accepted, so we can merge all final states to one final state. $\square$

Figure 4.2 shows that the converse of Proposition 4.2 does not hold .

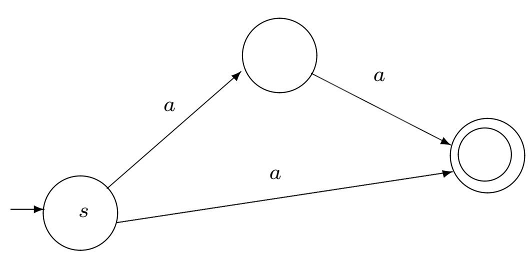

Figure 4.2: An NFA satisfying condition in Proposition 4.2, but it accepts the language $\{a, aa\}$ which is not prefix-free.

We continue with necessary conditions for DFAs accepting suffix-free languages.

Proposition 4.3 **(Neccessary conditions for suffix-free DFA).** *Let* $A = (Q, \Sigma, \delta, s, F)$ *be a minimal DFA for a non-empty suffix-free regular language. Then* A *satisfies the following properties:*

1. A *is non-returning.*

2. A *has a dead state.*

3. *For each symbol* a *in* Σ*, there is a state* $q_a \neq q_d$*, such that* $\delta(q_a, a) = q_d$*.*

4. *Let* $a \in \Sigma$*. There is no state* $q \in Q \setminus \{s\}$ *such that* $\delta(s, a) = \delta(q, a)$*.*

Proof. Let us prove every property. 1) Let us consider for contradiction, that there is a state $q \in Q$ and a symbol $a \in \Sigma$, such that there is a transition $q \xrightarrow{a} s$. Since A is minimal, there are strings u, v such that $\delta(q, v) \in F$ and $\delta(s, u) = q$. Therefore there are two accepted strings uv and $uauv$, where uv is the proper suffix of string $uauv$, hence language of automaton A is not suffix-free which is a contradiction. 2) Let $a \in \Sigma$. Consider the string a^m with $m \geq |Q|$. Let a sequence of states $s, q_1, q_2, \ldots, q_m$ be the computation on the string a^m. Let us show that the state q_m is dead. Let us assume for contradiction that q_m is not dead. Then there is a string w such that $\delta(q_m, w) \in F$. There are i, j such that $i < j$ and $q_i = q_j$, therefore we can omit states $q_{i+1}, \ldots, q_j$ and we get computation $s, q_1, \ldots, q_i, q_{j+1}, \ldots q_m$ on a string a^ℓ, where $\ell = m - (j - i)$. We get two accepted strings $a^m w, a^\ell w$, where $\ell < m$, so a^ℓ is the proper suffix of $a^m w$, which is contradiction with suffix-free property of $L(A)$. 3) Similarly as in the previous consideration we can take arbitrary $a \in \Sigma$, so there is q_a, such that $q_a \neq d$ and $\delta(q_a, a) = d$. 4) For a sake of contradiction let us assume that there are states q, p such that there are transitions $s \xrightarrow{a} p$ and $q \xrightarrow{a} p$, for some $a \in \Sigma$. There is a string $u \neq \varepsilon$, which leads automaton from s to q. Also, there is a string w, which leads automaton from p to some final state $f \in F$. Then there are two accepted computations $s \xrightarrow{a} p \xrightarrow{w} f$ and $s \xrightarrow{u} q \xrightarrow{a} p \xrightarrow{w} f$, so strings aw and uaw are accepted by A. The string aw is a proper suffix of the string uaw, which is a contradiction. □

The example in Figure 4.3 illustrates that the properties of Proposition 4.3 are not sufficient.

The next Cmorik's lemma provides a very comfortable tool for proving the suffix-freeness of a language accepted by an incomplete DFA.

Lemma 4.4 ([13, Lemma 1]). *Let* A *be a non-returning incomplete DFA that has a unique final state. If each state of* A *has at most one in-transition on every input symbol, then* $L(A)$ *is suffix-free.* □

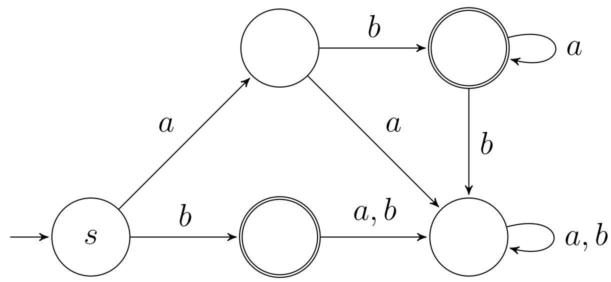

Figure 4.3: A DFA A satisfying every property from Proposition 4.3, but accepted language is not suffix-free since both b and ab are accepted.

4.2 Unary free languages

Every free unary language L can contain only one string. It follows that in the unary case all free classes coincide. Moreover if nondeterministic complexity of L is n, then we must have $L = \{a^{n-1}\}$. The next theorem gives an overview of complexities for basic operations except complementation on unary free languages. The complexity of complementation is analysed in another theorem.

Theorem 4.5. *Let K, L be a unary free languages with* $\text{nsc}(L) = n$ *and* $\text{nsc}(K) = m$.

(1) $\text{nsc}(K \cap L) \leq \max\{m, n\}$,

(2) $\text{nsc}(K \cup L) \leq \max\{m, n\}$,

(3) $\text{nsc}(KL) \leq m + n - 1$,

(4) $\text{nsc}(L^*) \leq n - 1$,

(5) $\text{nsc}(L^R) \leq n$.

Proof. (1) If $m \neq n$, then $K \cap L = \emptyset$, so $\text{nsc}(K \cap L) = 1$. If $m = n$, then $K \cap L = K = L$, so $\text{nsc}(K \cap L) = n$. In both cases $\text{nsc}(K \cap L) \leq \max\{m, n\}$. The languages $K = L = \{a^{m-1}\}$ meet the upper bound.

(2) Let $A = (\{0, \ldots, m-1\}, \{a\}, \delta, 0, \{m-1\})$ be NFA for language K, where $\delta(i, a) = i + 1$ if $0 \leq i < m - 1$, and $B = (\{q_0, \ldots, q_{n-1}\}, \{a\}, \delta', q_0, \{q_{n-1}\})$ be NFA for language L, where $\delta'(q_i, a) = q_{i+1}$ if $0 \leq i < n - 1$. Since union is commutative operation we may assume that $m \leq n$. For union $K \cup L$ we can construct automaton $C = (\{q_0, \ldots, q_{n-1}\}, \{a\}, \delta', q_0, \{q_{m-1}, q_{n-1}\})$ with n states, where $\delta'(q_i, a) = q_{i+1}$ if $0 \leq i < n - 1$. The languages a^{m-1} and a^{n-1} meet the upper bound for union.

(3) Let $A = (\{0, \ldots, m-1\}, \{a\}, \delta, 0, \{m-1\})$ be NFA for language K, where $\delta(i, a) = i+1$ if $0 \leq i < m-1$, and $B = (\{q_0, \ldots, q_{n-1}\}, \{a\}, \delta', q_0, \{q_{n-1}\})$ be NFA for language L, where $\delta'(q_i, a) = q_{i+1}$ if $0 \leq i < n-1$. For concatenation KL we can construct automaton $C = (\{0, \ldots, m+n-1\}, \{a\}, \delta'', 0, \{m+n-1\})$ with $m+n-1$ states, where $\delta''(q_i, a) = q_{i+1}$ if $0 \leq i < m+n-1$. The languages a^{m-1} and a^{n-1} meet the upper bound for concatenation.

(4) Let $A = (\{0, \ldots, n-1\}, \{a\}, \delta, 0, \{n-1\})$ be NFA for L, where $\delta(i, a) = i+1$ if $0 \leq i < n-1$. For L^* we can construct automaton $C = (\{0, \ldots, n-2\}, \{a\}, \delta', 0, \{0\})$ with $n-1$ states, where $\delta'(i, a) = i+1 \mod (n-1)$ if $0 \leq i \leq n-2$. The language a^{n-1} meets the upper bound for star.

(5) The reversal of every unary language is the same language, so $L^R = L$, therefore we have $\text{nsc}(L^R) = n$. $\square$

Now, let us analyse complementation on unary free languages. Recall that if $\text{nsc}(L) = n$ for a unary free language L, then $L = \{a^{n-1}\}$. Hence, the complement of L contains every string w in a^* with $|w| \neq n-1$.

Theorem 4.6. *Let L be unary free language with* $\text{nsc}(L) = n$. *Then* $\text{nsc}(L^c) = \Theta(\sqrt{n})$.

Proof. Let us denote the length of string in L by m. So $m = n-1$. The language L^c contains all strings with length not equal to m. First consider a lower bound, and let us show that every NFA for L^c requires at least $\sqrt{n/3}$ states. Assume for a contradiction that there is an NFA N for L^c with less than $\sqrt{n/3}$ states. Recall, that every unary n-state NFA can be transformed to Chrobak normal form which consist of a simple path containing not more than $2n^2+n$ states, ending with state in which there is a nondeterministic choice to several pairwise disjoint cycles containing at most n states (see Theorem 3.2, Figure 3.2). Thus, the tail in the Chrobak normal form of N is of size less than $3 \cdot (\sqrt{n/3})^2$ [12, 45], thus less than n. Since a^m must be rejected, each cycle in the Chrobak normal form must contain a rejecting state. It follows that infinitely many strings are rejected, which is a contradiction. Now let us prove the upper bound. Let $h = \lfloor \sqrt{m} \rfloor$, and consider relatively prime numbers h and $h+1$. It is known that the maximal integer that cannot be expressed as $xh + y(h+1)$ for non-negative integers x and y is $(h-1)h - 1 = h^2 - h - 1$ [49]. Let $k = m - (h^2 - h - 1)$. Then $0 < k \leq 3\sqrt{m}$. Next, the NFA A shown in Figure 4.4 and consisting of a path of length k and two overlapping cycles of lengths h and $h+1$ does not accept a^m, and accepts all strings a^i with $i \geq m+1$.

It remains to accept the shorter strings. To this aim let $p_1, p_2, \ldots, p_\ell$ be the first ℓ primes such that $p_1 p_2 \cdots p_\ell > m$. Then $\ell \leq \lceil \log m \rceil$. Thus $p_1 + p_2 + \cdots + p_\ell = \Theta(\ell^2 \ln \ell) \leq$

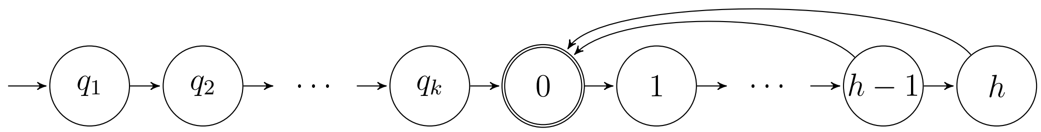

Figure 4.4: The part of NFA accepting every string of length more than $m = n - 1$

$\sqrt{m}$ [1]. Consider an NFA B consisting of an initial state s that is connected to ℓ cycles of lengths $p_1, p_2, \ldots, p_\ell$. Let the states in the $j - th$ cycle be $0, 1, \ldots, p_j - 1$, where s is connected to state 1. The state $m \bmod p_j$ is non-final, and all the other states are final. Then this NFA does not accept a^m, but accepts all strings a^i with $i \leq m-1$ since we have $(i \bmod p_1, i \bmod p_2, \ldots, i \bmod p_\ell) \neq (m \bmod p_1, m \bmod p_2, \ldots, m \bmod p_\ell)$. The NFA B for $m = 24$ is shown in Figure 4.5.

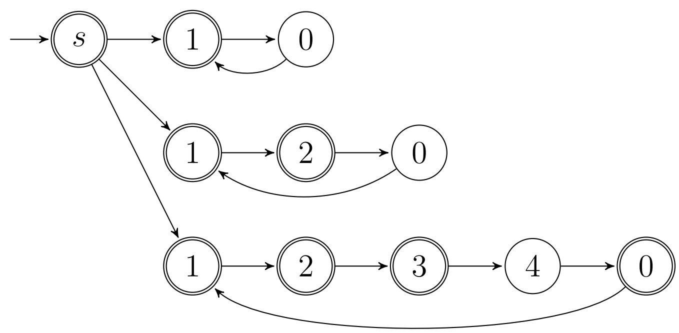

Figure 4.5: An example of NFA accepted every string shorter than $m = 24$

Now we get the resulting NFA for the language L of at most $6\sqrt{m}$ states as the union of NFAs A and B. □

4.3 Operations on free languages

We start with intersection. The nondeterministic complexity of intersection on prefix- and suffix-free languages was studied by Han et al. [20,22], where the tight upper bounds were obtained and a three-letter alphabet was used to prove tightness. The binary witnesses were described by Jirásková and Olejár [34]. Here we obtain the tight upper bounds for intersection on factor- and subword-free languages. To prove tightness, we use a binary alphabet in the factor-free case and a growing alphabet of size $m+n-5$ in the subword-free case.

Lemma 4.7. *Let K and L be languages over Σ with* $\text{nsc}(K) = m$ *and* $\text{nsc}(L) = n$.

(a) If K and L are prefix-free (suffix-free) then $\text{nsc}(K \cap L) \leq mn - (m + n - 2)$*, and the bound is tight if $m \geq 4$, $n \geq 2$, and $|\Sigma| \geq 2$.*

(b) If K and L are factor-free, then $\text{nsc}(K \cap L) \leq mn - 2(m + n - 3)$*, and the bound is tight if $m \geq 5$, $n \geq 3$, and $|\Sigma| \geq 2$.*

Proof. We first prove the upper bounds. Let A and B be minimal NFAs for K and L, respectively. We may assume that the state sets of A and B are $\{0, 1, \ldots, m - 1\}$ and $\{0, 1, \ldots, n - 1\}$, respectively, with the initial state 0 in both automata. Construct the product automaton $A \times B$ for $K \cap L$. If K and L are prefix-free with the final states $m - 1$ and $n - 1$ respectively, then all states in the last row and last column, except for $(m - 1, n - 1)$, are dead, so we can omit them. If K and L are suffix-free, then A and B are non-returning, so all states in the first row and first column, except for $(0, 0)$, are unreachable. Since every factor-free language is both prefix-free and suffix-free, all the three upper bounds follow from these observations.

To prove tightness, we first consider factor-free languages. Let $m \geq 5$, $n \geq 3$. Let K and L be the languages accepted by the NFAs A and B shown in Figure 4.6.

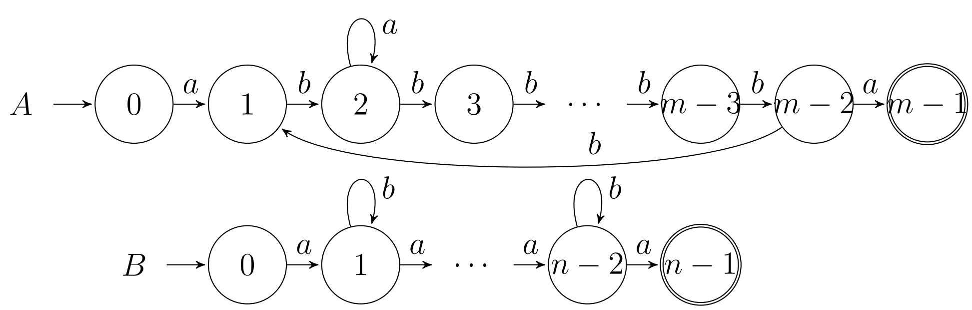

Figure 4.6: Factor-free witnesses for intersection meeting the bound $mn - 2(m + n - 3)$.

Every string w in K begins and ends with a, and $|w|_b \bmod (m - 2) = (m - 3)$. Every proper factor v of w which begins and ends with a has a computation in A which either starts in 0 and ends in 2, or starts and ends in 2, or starts in 2 and ends in $m - 1$. However, in all three cases, $|v|_b \bmod (m - 2) \neq (m - 3)$, so $v \notin L$. Hence the language K is factor-free. Next, every string in L has exactly $n - 1$ a's, but every proper factor of every string in L has less then $n - 1$ a's. Hence L is factor-free.

Construct the product automaton $A \times B$ and remove all the unreachable and dead states to get a trim NFA N for $K \cap L$. Figure 4.7 shows the NFA N in the case of $m = 5$

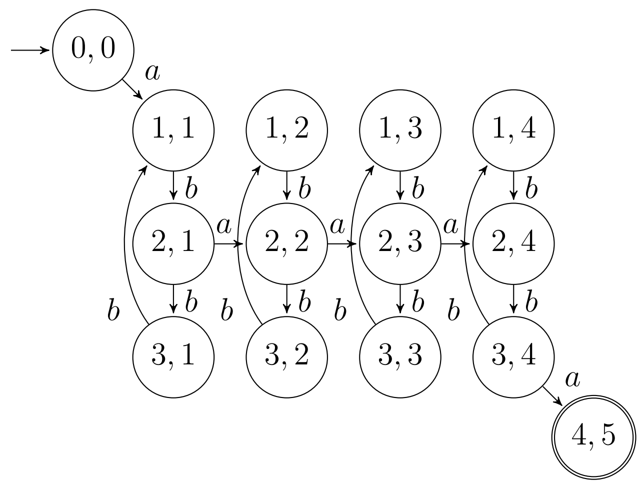

Figure 4.7: The NFA for intersection of the languages from Figure 4.6; $m = 5, n = 6$.

and $n = 6$. Since the NFA N and its reverse N^R are incomplete DFAs, the NFA N is minimal by Lemma 2.5. So we have $\text{nsc}(K \cap L) = mn - 2(m + n - 3)$. Notice that there is no need to prove that NFAs A and B are minimal because the upper bound cannot be met by languages of a smaller nondeterministic complexity. For this reason we do not prove the minimality of witnesses in what follows.

Next, the left quotients of K and L by the string a, that is, the languages $a\backslash K$ and $a\backslash L$, are prefix-free and meet the upper bound $mn - (m + n - 2)$. Similarly, the right quotients K/a and L/a are suffix-free witnesses. $\square$

The next lemma provides a subword-free witness for intersection defined over a growing alphabet.

Lemma 4.8. *Let $m, n \geq 3$. There exist subword-free regular languages K and L over an $(m + n - 5)$-letter alphabet such that $\text{nsc}(K) = m$, $\text{nsc}(L) = n$, and $\text{nsc}(K \cap L) = mn - 2(m + n - 3)$.*

Proof. Let $\Sigma = \{a\} \cup \{b_k \mid 2 \leq k \leq m - 2\} \cup \{c_\ell \mid 2 \leq \ell \leq n - 2\}$. Let K and L be languages accepted by incomplete DFAs $A = (\{0, 1, \ldots, m - 1\}, \Sigma, 0, \cdot, \{m - 1\})$ and $B = (\{0, 1, \ldots, n-1\}, \Sigma, 0, \circ, \{n-1\})$, where for each i $(0 \leq i \leq m-2)$, j $(0 \leq j \leq n-2)$, k $(2 \leq k \leq m - 2)$, and ℓ $(2 \leq \ell \leq n - 2)$, we have

$i \cdot a = i + 1,$	$j \circ a = j + 1,$
$0 \cdot b_k = k$ and $(k - 1) \cdot b_k = m - 1,$	$0 \circ b_k = 1$ and $(n - 2) \circ b_k = n - 1,$
$0 \cdot c_\ell = 1$ and $(m - 2) \cdot c_\ell = m - 1,$	$0 \circ c_\ell = \ell$ and $(\ell - 1) \circ c_\ell = n - 1.$

Figure 4.8 shows the automata A and B in the case of $m = 5$ and $n = 6$.

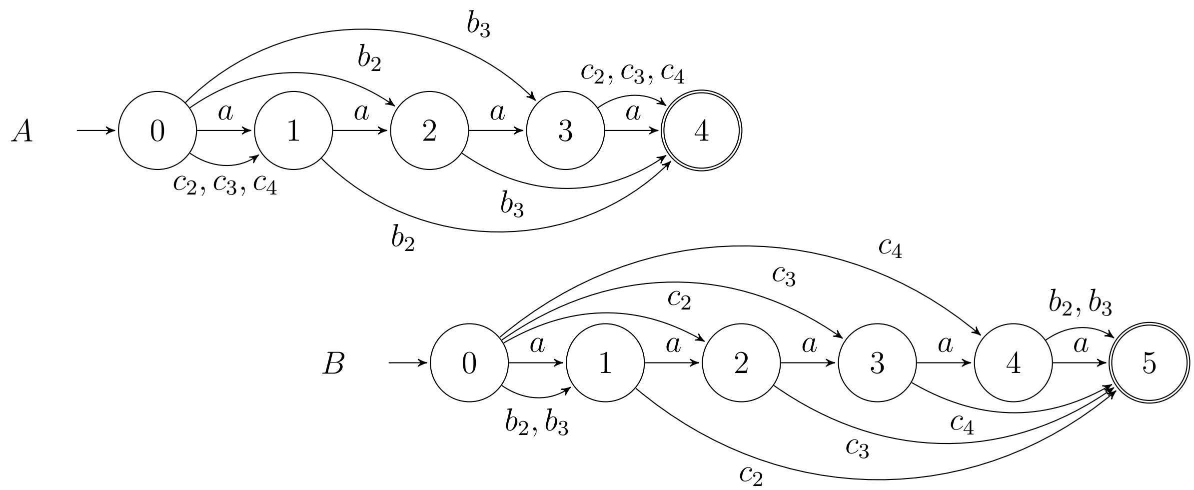

Figure 4.8: Subword-free witnesses for intersection; $m = 5, n = 6$.

To prove that K is subword-free, let $\Sigma_1 = \{a, c_2, c_3, \ldots, c_{m-2}\}$ and $\Sigma_2 = \{b_2, b_3, \ldots, b_{n-2}\}$. Notice that no string in Σ_1^* of length less than $m-1$ is in K. Next, each string in K contains at most two symbols from Σ_2. Let w be a string in K. If w contains no symbol from Σ_2, then $|w| = m-1$ and no proper subword of w is in K. If w contains exactly one symbol from Σ_2, then either $w = ub_k$ for some string u with $u \in \Sigma_1^*$ and $|u| = k-1$, or $w = b_k v$ for some string v with $v \in \Sigma_1^*$ and $|v| = n-k$. In both cases, no proper subword of w is in K. Finally, if w contains two symbols from Σ_2, then $w = b_k a^t b_{k+t+1}$ where $k \geq 0$ and $2 \leq k < k+t+1 \leq m-2$. No proper subword of such string is in K. The proof for L is similar.

Construct the product automaton $A \times B$ for $K \cap L$. To get a trim NFA N, omit all the unreachable and dead states; see Figure 4.9 for an illustration in the case of $m = 4$ and $n = 5$.

The resulting trim NFA has $(m-2)(n-2)+2$ states, it is an incomplete DFA, and its reverse is an incomplete DFA as well. By Lemma 2.6, this NFA is minimal. This concludes the proof. $\square$

We conjecture that the bound mn is asymptotically tight in the binary case if $m = n$.

Conjecture 4.9. *There exist a constant c and binary subword-free languages K and L with* $\mathrm{nsc}(K) = \mathrm{nsc}(L) = n$ *such that* $\mathrm{nsc}(K \cap L) \geq n^2/c$.

As a corollary of the three lemmata above, and taking into account the unary case in Theorem 4.5, we get the following result.

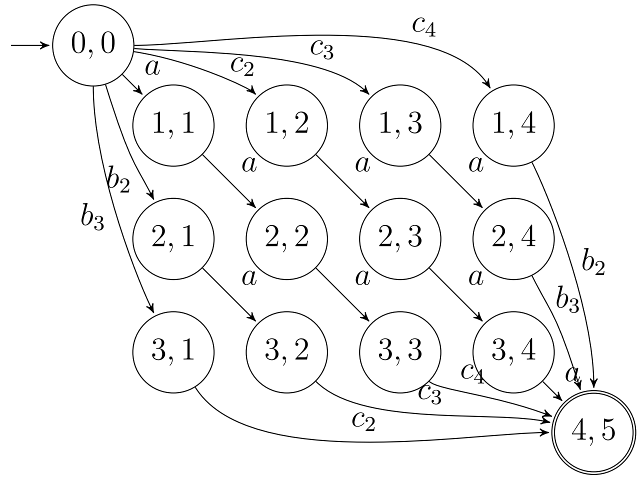

Figure 4.9: The NFA for intersection of languages from Figure 4.8.

Theorem 4.10 (Intersection). *The nondeterministic state complexity of intersection is* $mn - (m+n-2)$ *on prefix-free and suffix-free languages, and it is* $mn - 2(m+n-3)$ *on factor-free and subword-free languages. Except for subword-free witnesses which are defined over an alphabet of size* $m+n-5$*, all the remaining witnesses are binary and, moreover, this binary alphabet cannot be reduced.* □

Now we consider the union operation. In [22] it is claimed that the upper bound $m+n$ is met by the union of prefix-free languages $K = (a^{m-1})^*b$ and $L = (c^{n-1})^*d$, and a set P of pairs of strings of size $m+n$ is described in [22, Proof of Theorem 3.2]. The authors claimed that P is a fooling set for $K \cup L$. However, the language $K \cup L$ is accepted by an NNFA of $m+n-1$ states. Therefore P cannot be a fooling set for $K \cup L$. Here we prove the tightness of the upper bound $m+n$ for union of prefix-free languages using a binary alphabet and Lemma 2.2. Next we get the tight upper bound for union of suffix-, factor-, and subword-free languages. To get tightness, we always use a binary alphabet which is optimal for all four classes.

Lemma 4.11. *Let* K *and* L *be languages over* Σ *with* $\mathrm{nsc}(K) = m$ *and* $\mathrm{nsc}(L) = n$*.*

(a) If K *and* L *are prefix-free then* $\mathrm{nsc}(K \cup L) \leq m+n$*, and the bound is tight if* $m \geq 3, n \geq 3$*, and* $\Sigma \geq 2$*.*

(b) If K *and* L *are suffix-free then* $\mathrm{nsc}(K \cup L) \leq m+n-1$*, and the bound is tight if* $m \geq 3, n \geq 3$*, and* $\Sigma \geq 2$*.*

(c) If K *and* L *are factor-free, then* $\mathrm{nsc}(K \cup L) \leq m+n-2$*, and the bound is met by binary subword-free languages if* $m \geq 2$ *and* $n \geq 2$*.*

Proof. We first prove the upper bounds. Let A and B be minimal NFAs for K and L, respectively, with disjoint state sets, and the initial states s_A and s_B, respectively.

(a) If K and L are prefix-free, then NFAs A and B are non-exiting and have a unique final state. To get an $(m+n)$-state NFA for $K \cup L$ from A and B, add a new initial (non-final) state connected through ε-transitions to s_A and s_B, make the states s_A and s_B non-initial, and merge the final states of A and B.

(b) If K and L are suffix-free, then A and B are non-returning. We can get an $(m+n-1)$-state NFA for $K \cup L$ from A and B by merging their initial states.

(c) If K and L are factor-free, then they are both prefix- and suffix-free. To get an $(m+n-2)$-state NFA for $K \cup L$ from A and B, we merge their initial states, and then we merge their final states.

To prove tightness, consider languages K and L accepted by an m-state and n-state NFAs A and B, respectively, shown in Figure 4.10 (left). Notice that K is prefix-free since every string in K ends with b while every proper prefix of every string in K is in a^*. Similarly, L is prefix-free.

Construct the $(m+n)$-state NFA for their union by adding a new initial state s, by adding transitions (s, a, p_1) and (s, b, q_1), by making states p_0 and q_0 non-initial, and by merging their final states as shown in Figure 4.10 (right). The resulting trim NFA is an incomplete DFA, and its reverse is an incomplete DFA as well. By Lemma 2.6, this NFA is minimal. It follows that $\text{nsc}(K \cup L) \geq m + n$.

Next, the languages K^R and L^R are suffix-free, and they are accepted by m-state and n-state NFAs A^R and B^R, respectively. To get an NFA for $K^R \cup L^R$, we merge the initial states of A^R and B^R. For each state q of the resulting automaton, the singleton set $\{q\}$ is reachable, as well as co-reachable. By Lemma 2.5, this NFA is minimal. Hence we get $\text{nsc}(K^R \cup L^R) \geq m + n - 1$.

Finally, we again use Lemma 2.6 to show that the union of binary subword-free languages $\{a^{m-1}\}$ and $\{b^{n-1}\}$ meets the upper bound $m + n - 2$. $\square$

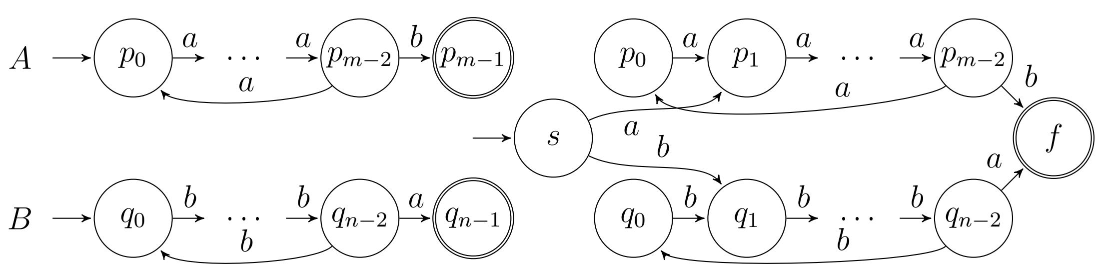

Figure 4.10: Binary prefix-free witnesses for union meeting the upper bound $m + n$.

As a corollary of the lemma above, and taking into account unary case in Theorem 4.5, we get the following result.

Theorem 4.12 (Union). *The nondeterministic state complexity of union is $m + n$ on prefix-free languages, $m + n - 1$ on suffix-free languages, and $m + n - 2$ on factor- or subword-free languages. All the witnesses can be defined over a binary alphabet, and the size of alphabet cannot be reduced.* □

The nondeterministic state complexity of concatenation on regular languages is $m+n$ with binary witnesses [24, Theorem 7]. For prefix-free and suffix-free languages, the upper bound is $m + n - 1$ [20, 22], and to prove tightness, a binary alphabet was used in [22, Theorem 3.1] and [20, Theorem 4]. In this section, we show that this upper bound is tight for all four classes of free languages, and to prove tightness, we use a unary alphabet.

Lemma 4.13. *Let K and L be prefix- or suffix-free languages with* $\mathrm{nsc}(K) = m$ *and* $\mathrm{nsc}(L) = n$. *Then* $\mathrm{nsc}(KL) \leq m + n - 1$, *and this upper bound is met by unary subword-free languages.*

Proof. Let A and B be minimal NFAs for K and L, respectively. In the prefix-free case, we can merge the final state of A and the initial state of B to get an NFA for KL. In the suffix-free case, automata A and B are non-returning. To get an NFA for KL, we add the transition (p, a, q) for each final state p of A and and each transition (s_B, a, q) of B. Next, we make final states of A non-final, and remove the unreachable state s_B. As a result, we get an NFA for KL of $m + n - 1$ states in both cases. This upper bound is met by the concatenation of unary subword-free languages $\{a^{m-1}\}$ and $\{a^{n-1}\}$. □

As an immediate corollary of the lemma above, we get the next result.

Theorem 4.14 (Concatenation). *The nondeterministic state complexity of concatenation on each of the four classes of free languages is $m + n - 1$, with unary witnesses.* □

We next consider the Kleene star and reversal operations. Both operations have nondeterministic complexity $n + 1$ on regular languages with a unary witness for star [24, Theorem 9] and a binary witness for reversal [30, Theorem 2].

In [22, Theorem 4.2] and [20, Theorem 7] it is claimed that for each prefix-free or suffix-free language L with $\mathrm{nsc}(L) = n$ the nondeterministic complexity of L^* is n. However, this is not true since $\{a^{n-1}\}$ is a prefix- and suffix-free language of nondeterministic complexity n and its star, the language $(a^{n-1})^*$, has nondeterministic complexity $n - 1$.

The next lemma provides tight upper bounds for star on all four classes of free languages. To get tightness, we use an optimal binary alphabet in the prefix- and suffix-free case, and a unary alphabet otherwise.

Lemma 4.15. *Let L be a language over an alphabet Σ with* $\operatorname{nsc}(L) = n$.

(a) If L is prefix- or suffix-free then $\operatorname{nsc}(L^*) \le n$. *These upper bounds are tight if $|\Sigma| \ge 2$, and the size of alphabet cannot be decreased.*

(b) If L is factor-free, then $\operatorname{nsc}(L^*) \le n-1$, *and the bound is met by a unary subword-free language.*

Proof. Let $A = (Q, \Sigma, \cdot, s, F)$ be a minimal NFA for L.

(a) If L is prefix-free, then A is non-exiting and has a unique final state q_f. We can construct an n-state ε-NFA for the language L^* from A by making state q_f initial and state s non-initial, and by adding the ε-transition from q_f to s. If L is suffix-free, then A is non-returning. Now we construct an n-state ε-NFA for L^* from A by making the initial state s final, and by adding the ε-transition from every final state to the initial state s.

To get tightness, we first consider the suffix-free case. Let L be the language accepted by the n-state NFA A shown in Figure 4.11 (left). Notice that it is non-returning, has a unique final state, and each of its states has at most one in-transition on each input symbol. By Lemma 4.4, the language L is suffix-free. Next, the set $\{(a^i, a^{n-1-i}b) \mid 0 \le i \le n-1\}$ is a fooling set for L^* since $a^{n-1}b \in L^*$, but for each j with $j < n-1$, the string $a^j b$ is not in L^*.

Now consider the prefix-free language L^R. It is accepted by the n-state NFA A^R shown in Figure 4.11 (right). Construct an NFA N for L^* from A by making state $n-1$ initial and state 0 non-initial, and by adding the transitions $(n-1, a, 1)$ and $(n-1, b, 0)$. Notice that for each state q of N, the singleton set $\{q\}$ is reachable and co-reachable, so N is minimal by Lemma 2.5.

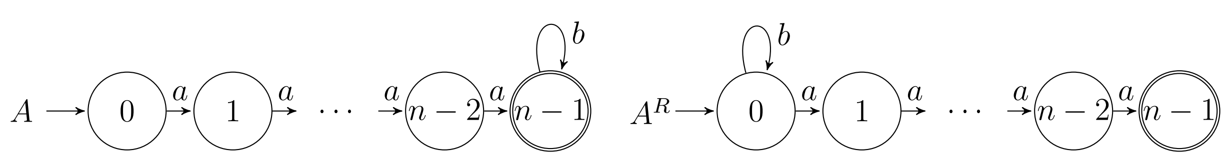

Figure 4.11: Prefix-free and suffix-free witnesses for star meeting the upper bound n.

(b) If L is factor-free, then A is non-returning and non-exiting, and it has a unique final state q_f. We construct an NFA for L^* by making state q_f initial, by adding transition

(q_f, a, q) for each transition (s, a, q), and by omitting the unreachable state s. The unary subword-free language $\{a^{n-1}\}$ meets this upper bound. $\square$

The next theorem summarizes the results of the lemma above.

Theorem 4.16 (Star). *The nondeterministic state complexity of star is n on prefix- and suffix-free languages with binary witnesses, and it is $n-1$ on factor- and subword-free languages with unary witnesses. The binary alphabet in the prefix- and suffix-free case cannot be reduced.* $\square$

Now we turn our attention to the reversal operation. Han et al. obtained tight upper bounds for reversal on prefix-free and suffix-free languages and they provided a binary prefix-free witness [22, Theorem 3.4] and a ternary suffix-free witness [20, Theorem 9]. As shown in the next lemma, the upper bound for reversal on prefix-free languages is n, so it is met by any unary language, in particular, by the subword-free language $\{a^{n-1}\}$.

Lemma 4.17. *Let L be a prefix-free language with* $\operatorname{nsc}(L) = n$*. Then* $\operatorname{nsc}(L^R) \leq n$*, and this upper bound is met by a unary subword-free language.*

Proof. If L is prefix-free, then every minimal NFA for L has a unique final state. Thus $\operatorname{nsc}(L^R) \leq n$. The bound is met by the subword-free language $\{a^{n-1}\}$. $\square$

Now we consider the suffix-free case, and provide a binary witness meeting the upper bound $n+1$. Notice that the reverse of a language accepted by an n-state NFA is accepted by an n-state NNFA. This means that we cannot use a fooling set method to prove the tightness of the bound $n+1$. However, a modified fooling set method described in Lemma 2.2 can be successfully used here.

Lemma 4.18. *Let $n \geq 5$. There exists a binary suffix-free language L such that* $\operatorname{nsc}(L) = n$ *and* $\operatorname{nsc}(L^R) = n+1$.

Proof. Let L be the language accepted by the NFA A shown in Figure 4.12.

Since every string in L contain both a and b, but every proper suffix of every string in L is in $a^* \cup b^*$, the language L is suffix-free. Now we show that every NFA for L^R needs at least $n+1$ states. Let

$\mathcal{A} = \{(a^{n-3}, a^{n-4}b)\} \cup \{(a^i, a^{n-4-i}b) \mid 1 \leq i \leq n-4\} \cup \{(a^{n-4}b, \varepsilon)\}$,

$\mathcal{B} = \{(bb, ba), (b, a)\}$,

$u = ba$,

$v = a^{n-4}b$.

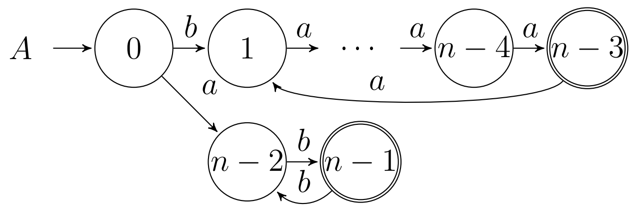

Figure 4.12: A binary suffix-free witness for reversal meeting the upper bound $n + 1$.

Notice that $\{a^{2n-7}b, a^{n-4}b, bbba, ba\} \subseteq L^R$. Moreover, in every string in L^R starting with a, the number of consecutive a's modulo $(n-3)$ is $(n-4)$, and the string ends with a single b. Next, the string bba is not in L^R. Finally, every string in L^R either starts with a and ends with b or starts with b and ends with a. It follows that $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, u)\}$, and $\mathcal{B} \cup \{(\varepsilon, v)\}$ are fooling sets for L^R. By Lemma 2.2, we have $\operatorname{nsc}(L^R) \geq n + 1$. $\square$

We can use the same fooling sets as in the above proof to show that the left ideal language $\{a, b\}^*L$ is a witness for reversal meeting the upper bound $n+1$. This improves the result from [26, Theorem 17] by decreasing the size of alphabet from three to two.

As a corollary of the two lemmata above, and taking into account that the reversal of every unary language is the same language, we get the next result.

Theorem 4.19 (Reversal). *The nondeterministic state complexity of reversal is n on the classes of prefix-, factor-, and subword-free languages, with the witnesses defined over a unary alphabet. The nondeterministic state complexity of reversal is $n + 1$ on the class of suffix-free languages, with the witnesses defined over a binary alphabet which is optimal in this case.* $\square$

We continue with complementation, which is the most interesting and the most difficult part of this thesis. Han and Salomaa in [20] have obtained an upper bound $2^{n-1} + 1$ on the nondeterministic state complexity of complementation on suffix-free languages. Our next result shows that this upper bound can be decreased by one.

Lemma 4.20. *Let $n \geq 3$. Let L be a suffix-free regular language with* $\operatorname{nsc}(L) = n$. *Then* $\operatorname{nsc}(L^c) \leq 2^{n-1}$.

Proof. Let N be a non-returning n-state NFA for a suffix-free language L. The subset automaton $A = (Q, \Sigma, \delta, s, F)$ of the NFA N has at most $1 + 2^{n-1}$ reachable states since the only reachable subset that contains the initial state of N is the initial state of the subset automaton. The initial state of the subset automaton is non-final since L does not contain the empty string.

After interchanging the final and non-final states, we get a DFA $A^c = (Q, \Sigma, \delta, s, Q \setminus F)$ for L^c of $1 + 2^{n-1}$ states. The initial state of A^c is final and has no in-transitions. The state d is final as well, and it accepts every string.

Construct a 2^{n-1}-state NFA N^c from the DFA A^c as follows. Let Q_d be the set of states of A^c different from d and such that they have a transition to the state d, that is, $Q_d = \{q \in Q \setminus \{d\} \mid \text{there is an } a \text{ in } \Sigma \text{ such that } \delta(q, a) = d\}$; remind that by Proposition 4.3, for each symbol a, there is a state q_a in Q_d that goes to d by a. Replace each transition (q, a, d) by transitions (q, a, p) for each p in Q_d, and moreover add the transition (q, a, s). Then, remove the state d. Formally, let $N^c = (Q \setminus \{d\}, \Sigma, \delta', s, (Q \setminus \{d\}) \setminus F)$, where

$$\delta'(q, a) = \begin{cases} \{\delta(q, a)\}, & \text{if } \delta(q, a) \neq d, \\ \{s\} \cup Q_d, & \text{if } \delta(q, a) = d. \end{cases}$$

In a similar way as in the case of prefix-free languages, it can be shown that $L(N^c) = L(A^c)$. □

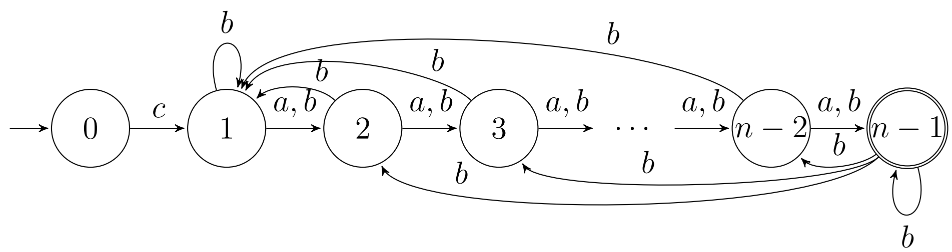

Figure 4.13: An NFA of a ternary suffix-free regular language L with $\text{nsc}(L^c) = 2^{n-1}$

As for a lower bound, Han and Salomaa in [20] claimed that there exists a ternary suffix-free language meeting the bound $2^{n-1} - 1$. In the next lemma, we increase this lower bound by one.

Lemma 4.21. *Let* $n \geq 3$. *There exists a ternary suffix-free language such that* $\text{nsc}(L) = n$ *and* $\text{nsc}(L^c) \geq 2^{n-1}$.

Proof. Let K be the language accepted by the NFA over $\{a, b\}$ shown in Figure 3.1 with $n - 1$ states. Set $L = c \cdot K$. Then L is a suffix-free language recognized by an n-state NFA shown in Figure 4.13. As shown in [30, Theorem 5], there exists a fooling set $\mathcal{F} = \{(x_S, y_S) \mid S \subseteq \{1, 2, \dots, n-1\}\}$ of size 2^{n-1} for the language K^c. Then the set of pairs of strings $\mathcal{F}' = \{(c \cdot x_S, y_S) \mid S \subseteq \{1, 2, \dots, n-1\}\}$ is a fooling set of size 2^{n-1} for the language L^c. □

We can summarize these results in the following theorem which provides the tight bound on the nondeterministic state complexity of complementation on suffix-free languages over an alphabet with at least three symbols.

Theorem 4.22 (Complementation on suffix-free languages). *Let $n \geq 3$. Let L be a suffix-free language over an alphabet Σ with $\operatorname{nsc}(L) = n$. Then $\operatorname{nsc}(L^c) \leq 2^{n-1}$, and the bound is tight if $|\Sigma| \geq 3$.* $\square$

Now, let us turn our attention to investigate bounds for binary alphabet. The lower bound for binary alphabet is a little bit different. We also improve the estimation of upper bound.

Let G be the language accepted by the NFA over $\{a, b\}$ shown in Figure 3.1 with $n-1$ states. Let $L = cG$. The language L is a suffix-free language over $\{a, b, c\}$ recognized by an n-state NFA A shown in Figure 4.13, and we have $\operatorname{nsc}(L^c) \geq 2^{n-1}$ ([32, Lemma 5]). Now, let us define a homomorphism h as follows: $h(c) = 00$, $h(a) = 10$, $h(b) = 11$ (used in [13, Theorem 7]). After applying h on the language L, we have a binary language $K = h(L)$ over $\{0, 1\}$.

Lemma 4.23. *The language K is a suffix-free language.*

Proof. Every string in L contains exactly one symbol c at the begining, so every string in K begins with the string 00 and such substring does not appear further in the string. If there is a string $w = uv$ and $u \neq \varepsilon$, then v does not contain 00 and therefore $v \notin K$. So K is suffix-free. $\square$

Now let us define NFA A' for the language K. We use the description of automaton A for original language L. Let $A = (Q, \{a, b, c\}, \delta, 0, \{n-1\})$ (be NFA shown in Figure 4.13). The idea is replace every transitions $q \xrightarrow{a} q_a$ by adding a new state q' and two transitions $q \xrightarrow{1} q' \xrightarrow{0} q_a$, similarly for symbol b $q \xrightarrow{1} q' \xrightarrow{1} q_b$ and transition $q \xrightarrow{c} q_c$ we replace by adding q' and two transitions $q \xrightarrow{0} q' \xrightarrow{0} q_c$; see Figure 4.14.

More formally: $A' = (Q', \{0, 1\}, \delta', 0, \{n-1\})$ where states $Q' = \bigcup_{q \in Q} \{q, q'\}$ (so we add for every state q from Q a new state q') and transition function is defined as follows:

- for $q \neq 0$ and $q' \neq 0'$, we have $\delta'(q, 1) = \{q'\}$, $\delta'(q', 0) = \delta(q, a)$, $\delta'(q', 1) = \delta(q, b)$
- for $0, 0'$ we have $\delta'(0, 0) = \{0'\}$, $\delta'(0', 0) = \delta(0, c)$

In the following three lemmas, we prove that the NFA A' is a minimal NFA recognizing the language K. Then we show that $\operatorname{nsc}(K^c) \geq 2^{n-1}$

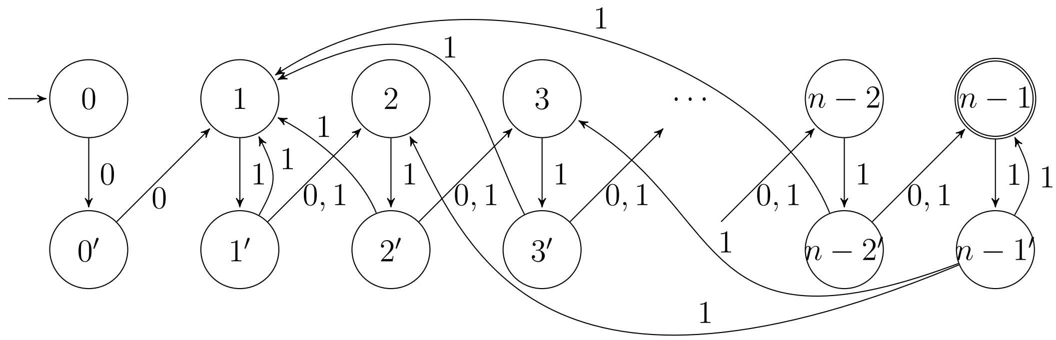

Figure 4.14: An NFA of binary suffix-free regular language K.

Lemma 4.24. *The NFA A' defined above recognizes the language K.*

Proof. We have to prove $L(A') = K$.

The first we show $K \subseteq L(A')$. Let $w \in K$, then there is $u \in L$, where $u = u_1 u_2 \dots u_m$ such that $h(u) = w$, so $w = h(u_1)h(u_2)\cdots h(u_m)$. There is computation in A as follows: $0, q_1, q_2, \dots, q_m$ where $q_1 \in \delta(0, u_1)$ and for every q_i, such that $1 \leq i < m$ $q_{i+1} \in \delta(q_i)$, 0 is initial state and q_m is final state.

We claim that after reading $h(u_1)h(u_2)\cdots h(u_i)$ the automaton A' can be in q_i thus $q_i \in \delta'(0, h(u_1)h(u_2)\cdots h(u_i))$. We prove it by mathematical induction.

The base case is $i = 1$. Every string in L begins with symbol c, so $u_1 = c$, hence $h(u_1) = 00$. By definition of δ', $\delta'(0, 0) = \{0'\}$ and $\delta'(0', 0) = \delta(0, c)$, so $q_1 \in \delta'(0, h(u_1))$

Let us assume that $q_i \in \delta'(0, h(u_1)h(u_2)\cdots h(u_i))$, $1 \leq i \leq m-1$. Symbol u_{i+1} can be equal to a or b. There is $\delta'(q_i, 1) = \{q_i'\}$, so $q_i' \in \delta'(0, h(u_1)h(u_2)\cdots h(u_i)1)$. If $u_{i+1} = a$, then $h(u_{i+1}) = 10$. Since $\delta'(q_i', 0) = \delta(q_i, a)$ and $q_{i+1} \in \delta(q_i, a)$, we have $q_{i+1} \in \delta'(0, h(u_1)\dots h(u_i)10) = \delta'(0, h(u_1)\dots h(u_i)h(u_{i+1}))$. Similarly if $u_{i+1} = b$.

In conclusion $q_m \in \delta'(0, h(u_1)\cdots h(u_m))$ and q_m is final state, so $w \in L(A')$.

Now we show that $L(A') \subseteq K$.

Let $w \in L(A')$. Every computation in A' has an alternate form of states q, q' as follows $00'q_1q_1'q_2q_2'\cdots q_{m-1}q_{m-1}'q_m$ and accepted string has a form $001x_11x_2\cdots 1x_{m-1}$ where $x_i \in \{0, 1\}$ for $1 \leq i \leq m-1$. For such string is possible to find a string $u = u_1u_2\cdots u_m$, where $u_1 = c$ and for every $1 \leq i \leq m$, $u_i \in \{a, b\}$ and $h(u_i) = 1x_{i-1}$.
We claim that after reading $u_1u_2\cdots u_i$, the automaton A can be in q_i, more precisely $q_i \in \delta(0, u_1u_2\cdots u_i)$. We show it by mathematical induction.

The base case is $i = 1$. Then $u_1 = c$ and $\delta'(0, 00) = \delta(0, c)$ and $q_1 \in \delta'(0, 00)$ so $q_1 \in \delta(0, u_1)$.

Let us assume $q_i \in \delta(0, u_1 \cdots u_i)$, where $1 \leq i \leq m-1$. By definition of δ' we have $\delta'(q_i, 1x_i) = \delta(q_i, u_{i+1})$ and since $q_{i+1} \in \delta'(q_i, 1x_i)$, we have $q_{i+1} \in \delta(q_i, u_{i+1})$. So at last $q_m \in \delta(0, u_1 u_2 \ldots u_m)$ and q_m is final state, so $u = u_1 u_2 \ldots u_m \in L$.

Hence: $u \in L, h(u) = w$, so $w \in K$. □

Lemma 4.25. *The NFA A' is a minimal NFA for the language K.*

Proof. For every state q of A', we are going to define a pair of strings (u_q, v_q) such that

(a) by u_q, the initial state of A' goes only to the state q, and

(b) v_q is accepted by A' only from the state q if $q \neq (n-1)'$.

Let $u_0 = \varepsilon$ and $u_{0'} = 0$. Next, if $1 \leq i \leq n-1$, then let $u_i = 00(10)^{i-1}$ and $u_{i'} = 00(10)^{i-1}1$. Then (a) is satisfied for every state q of A'.

Now, let $v_0 = 00(10)^{n-2}$ and $v_{0'} = 0(10)^{n-2}$. Next, if $1 \leq i \leq n-2$, then let $v_i = (10)^{n-1-i}$ and $v_{i'} = 0(10)^{n-2-i}$. Finally, let $v_{n-1} = \varepsilon$ and $v_{(n-1)'} = 1$. Then (b) is satisfied for every state q of A', except for $(n-1)'$. We show that the set of pairs of strings $\mathcal{F} = \{(u_q, v_q) \mid q \text{ is a state of } A'\}$ is a fooling set for K.

(F1) For every pair (u_q, v_q), we have $u_q v_q \in K$.

(F2) Let us consider two distinct pairs (u_q, v_q) and (u_p, v_p), except for $(u_{(n-1)'}, v_{(n-1)'})$. By the string u_q, we reach only the state q, and the string v_p is accepted only from the state p. Thus $u_q v_p \notin K$. Now let us consider (u_p, v_p) and $(u_{(n-1)'}, v_{(n-1)'})$. Since by the string $u_{(n-1)'}$, we can reach only the state $(n-1)'$, and the string v_p is accepted only from the state p, we have $u_{(n-1)'} v_p \notin K$.

Hence $\mathcal{F}$ is a fooling set for K. Since the size of $\mathcal{F}$ is $2n$, the NFA A' is minimal. □

Lemma 4.26. *Let $n \geq 3$ and K be the language defined above. Then* $\operatorname{nsc}(K^c) \geq 2^{n-1}$.

Proof. As it is shown in [32, Lemma 5], the set $\mathcal{F} = \{(cx_S, y_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ is a fooling set for L^c. Let us define $\mathcal{F}' = \{(h(cx_S), h(y_S)) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$. Let us show that the $\mathcal{F}'$ is fooling set for K^c.

(F1) For every pair $(h(cx_S), h(y_S))$, we have $cx_S y_S \in L^c$, so $cx_S y_S \notin L$ and since homomorphism h is a bijection $h(cx_S y_S) \notin K$ so $(h(cx_S), h(y_S)) \in K^c$.

(F2) Let $(h(cx_S), h(y_S))$, $(h(cx_T), h(y_T))$ be two distinct pairs. Without loss of generality, let $cx_S y_T \notin L^c$. So $cx_S y_T \in L$, then $h(cx_S y_T) \in K$, so $h(cx_S y_T) \notin K^c$.

Hence $\mathcal{F}'$ is a fooling set for K^c. Since the size $\mathcal{F}'$ is 2^{n-1}, $\operatorname{nsc}(K^c) \geq 2^{n-1}$. □

We need the following observation later.

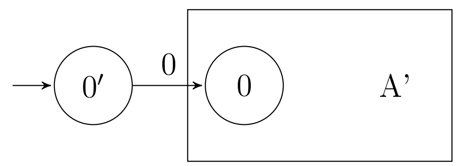

Figure 4.15: An automaton A'' recognizing a binary suffix-free language K_1.

Proposition 4.27. *Let L be a suffix-free language L over alphabet Σ. Then for every $x \in \Sigma$ the language $R = xL$ is suffix-free.*

Proof. For a contradiction let us assume, that there are two strings xu, xv in R, such that xv is suffix of xu. So there exists y such, that $xu = xyxv$. Hence $u = yxv$, it means that v is suffix of u and $u, v \in L$. It is contradiction, that L is suffix-free. □

Above we found a binary language with an even nondeterministic state complexity, and now we want to find a binary language with an odd one. Now let us consider the language $K_1 = 0K$, where K is described above. By Proposition 4.27, K_1 is suffix-free.

Lemma 4.28. *Let K_1 be suffix-free language given above. Then* $\text{nsc}(K_1) = 2n + 1$.

Proof. Let us consider the automaton A' for the language K. Let us construct an automaton A'' from A' by simply adding a new state $0'$ and transition from $0'$ to original initial state 0 on symbol 0. State $0'$ become a new initial state. The NFA A'' is shown in Figure 4.15, and we have $L(A'') = K_1$. Now let us consider the minimality of A''. Let $\mathcal{F}$ be fooling set for K. Let us construct $\mathcal{F}'$ from $\mathcal{F}$ as follows: $\mathcal{F}' = \{(0u, v) \mid (u, v) \in \mathcal{F}\} \cup \{\varepsilon, 000(10)^{n-2}\}$. The set $\mathcal{F}'$ is fooling set for K_1 and $|F'| = 2n + 1$, so $\text{nsc}(K_1) = 2n + 1$. □

Lemma 4.29. *Let $n \geq 3$ and K_1 be the language defined above. Then* $\text{nsc}(K_1^c) \geq 2^{n-1}$.

Proof. Let $\mathcal{F}$ be the fooling set for K^c given by Lemma 4.26. Let $\mathcal{F}' = \{(0u, v) \mid (u, v) \in \mathcal{F}\}$. Let us show that $\mathcal{F}'$ is fooling set for K_1^c.

(F1) If $uv \in K^c$, then $uv \notin K$, then also $0uv \notin K_1$, so $0uv \in K_1^c$.

(F2) If $(u, v), (x, y)$ are two distinct pairs in $\mathcal{F}$. Then without loss of generality, $uy \notin K^c$, so $uy \in K$. Then $0uy \in K_1$ and $0uy \notin K_1^c$.

Hence $\mathcal{F}'$ is fooling set for K_1^c. Since the size of $\mathcal{F}'$ is 2^{n-1}, we have $\text{nsc}(K_1^c) \geq 2^{n-1}$. □

We summarize our results in the following theorem.

Theorem 4.30 (Complementation on binary suffix-free languages; lower bound). *Let $n \geq 6$. There is a binary suffix-free language L such that* $\operatorname{nsc}(L) = n$ *and* $\operatorname{nsc}(L^c) \geq 2^{\lfloor \frac{n}{2} \rfloor - 1}$.

Proof. If n is even, that is, $n = 2k$ for some $k \geq 3$, then we set $L = K$, where K is the language described above with $\operatorname{nsc}(K) = 2k$. By Lemma 4.26, $\operatorname{nsc}(K^c) \geq 2^{k-1}$. Hence $\operatorname{nsc}(L^c) = \operatorname{nsc}(K^c) \geq 2^{k-1} = 2^{\lfloor \frac{n}{2} \rfloor - 1}$. If n is odd. That is, $n = 2k+1$ for some $k \geq 3$, then we said $L = K_1$, where K_1 is the language described above with $\operatorname{nsc}(K_1) = 2k+1$. By Lemma 4.29, $\operatorname{nsc}(K_1^c) \geq 2^{k-1}$. Hence $\operatorname{nsc}(L^c) = \operatorname{nsc}(K_1^c) \geq 2^{k-1} = 2^{\lfloor \frac{n}{2} \rfloor - 1}$. □

In the next consideration we use concept of *prefix-free* languages. Now, we consider an upper bound. Let us recall the following result.

Lemma 4.31. *Let $n \geq 12$. Let L be a binary prefix-free language with* $\operatorname{nsc}(L) = n$. *Then* $\operatorname{nsc}(L^c) \leq 2^{n-1} - 2^{n-3} + 1$.

Notice that the proof at [32, Lemma 9] works also for NFAs with multiple initial states. We are going to use it also for suffix-free languages.

Lemma 4.32. *Let $n \geq 12$. Let L be a binary suffix-free language with* $\operatorname{nsc}(L) = n$. *Then* $\operatorname{nsc}(L^c) \leq 2^{n-1} - 2^{n-3} + 2$.

Proof. After reversing an NFA for L, we obtain an n-state NFA (possibly with multiple initial states) for a prefix-free language L^R. By Lemma 4.31, $\operatorname{nsc}((L^R)^c) \leq 2^{n-1} - 2^{n-3} + 1$. Since $(L^R)^c = (L^c)^R$, we have

$$\operatorname{nsc}((L^c)^R) \leq 2^{n-1} - 2^{n-3} + 1$$

It follows that $(L^c)^R$ is accepted by an NFA N which has at most $2^{n-1} - 2^{n-3} + 1$ states. Now we reverse the the NFA N, and get NFA N^R, possibly with multiple initial states. By adding one more state, we get an NFA for L^c with at most $2^{n-1} - 2^{n-3} + 2$ states and with a unique initial state. Our proof is complete. □

So in the binary case the upper bound does not reach value 2^{n-1} and there is language, such that nondeterministic complexity of its complement is at least $2^{\lfloor \frac{n}{2} \rfloor - 1}$, so complement requires still exponential number of states for nondeterministic automaton.

Han *et al.* in [21] obtained an upper bound $2^{n-1} + 1$ and a lower bound 2^{n-1} on the nondeterministic complexity of complementation on prefix-free languages. Our first result shows that the upper bound can be decreased by one.

Lemma 4.33. *Let $n \geq 3$. Let L be a prefix-free regular language with* $\text{nsc}(L) = n$*. Then* $\text{nsc}(L^c) \leq 2^{n-1}$.

Proof. Let N be an n-state NFA for a prefix-free language L. Construct the subset automaton of the NFA N and minimize it. Then, all the final states are equivalent, and they go to the dead state on each input. Thus L is accepted by a DFA $A = (Q, \Sigma, \delta, s, \{q_f\})$ with at most $2^{n-1} + 1$ states, with a dead state q_d which goes to itself on each symbol, and one final state q_f which goes to the dead state on each symbol, thus $\delta(q_d, a) = q_d$ and $\delta(q_f, a) = q_d$ for each a in Σ.

To get a DFA for the language L^c, we interchange the final and non-final states in the DFA A, thus L^c is accepted by the $(2^{n-1} + 1)$-state DFA $A^c = (Q, \Sigma, \delta, s, Q \setminus \{q_f\})$. We show that using nondeterminism, we can save one state, that is, we describe a 2^{n-1}-state NFA for the language L^c.

Construct a 2^{n-1}-state NFA N^c for L^c from the DFA A^c by omitting state q_d, and by replacing each transition (q, a, q_d) by two transitions (q, a, q_f) and (q, a, s); see the Figure 4.16.
Formally, construct an NFA $N^c = (Q \setminus \{q_d\}, \Sigma, \delta', s, Q \setminus \{q_f, q_d\})$, where

$$\delta'(q, a) = \begin{cases} \{\delta(q, a)\}, & \text{if } \delta(q, a) \neq q_d, \\ \{q_f, s\}, & \text{if } \delta(q, a) = q_d. \end{cases}$$

Let us show that $L(N^c) = L(A^c)$.

Let $w = a_1 a_2 \cdots a_k$ be a string in $L(A^c)$, and let $s, q_1, q_2, \ldots, q_k$ be the computation of the DFA A^c on the string w. If $q_k \neq q_d$, then each q_i is different from q_d since q_d goes to itself on each symbol. It follows that $s, q_1, q_2, \ldots, q_k$ is also a computation of the NFA N^c on the string w. Now assume that $q_k = q_d$. Then there exists an ℓ such that the states $q_\ell, q_{\ell+1}, \ldots, q_k$ are equal to q_d, and the states $s, q_1, \ldots, q_{\ell-1}$ are not equal to q_d. If $\ell = k$, then $\delta(q_{k-1}, a_k) = q_d$, so $s \in \delta'(q_{k-1}, a_k)$. It follows that $s, q_1, q_2, \ldots, q_{k-1}, s$ is an accepting computation of N^c on w. If $\ell < k$, then we have $q_\ell = q_{\ell+1} = \cdots = q_k = q_d$, and therefore the string w is accepted in N^c through the accepting computation $s, q_1, \ldots, q_{\ell-1}, q_f, q_f, \ldots, q_f, s$ since we have $\delta'(q_{\ell-1}, a_\ell) = \{q_f, s\}$, and $\delta'(q_f, a) = \{q_f, s\}$ for each a in Σ.

Now assume that a string $w = a_1 a_2 \cdots a_k$ is rejected by the DFA A^c. Let $s = q_0, q_1, q_2, \ldots, q_k$ be the rejecting computation of the DFA A^c on the string w. Since the only non-final state of the DFA A^c is q_f, we must have $q_k = q_f$. It follows that each state q_i is different from q_d, and therefore in the NFA N^c, we have $\delta'(q_{i-1}, a_i) = \{\delta(q_{i-1}, a_i)\}$.

This means that $s = q_0, q_1, q_2, \ldots, q_k$ is a unique computation of N^c on w. Since this computation is rejecting, the string w is rejected by the NFA N^c. $\square$

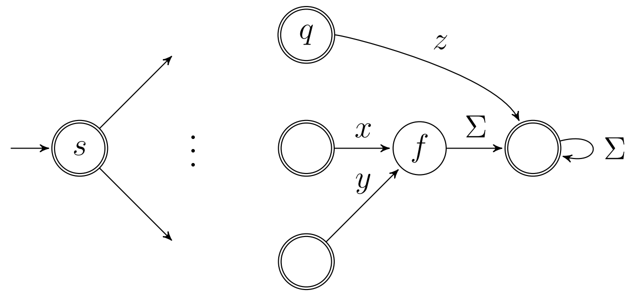

Figure 4.16: A sketch of substitution of a former dead state by new transitions

To prove tightness, we use similar languages as in the case of suffix-free languages, shown in Figure 4.17

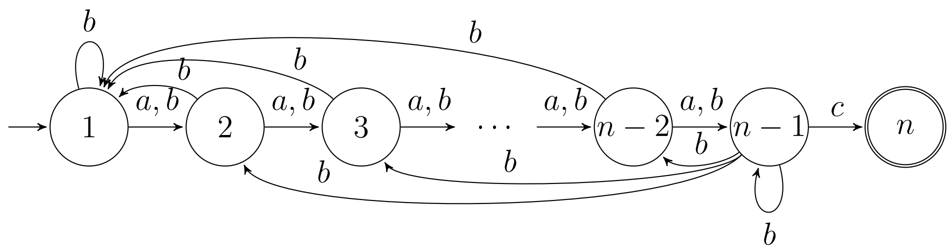

Figure 4.17: An NFA of a ternary prefix-free language L with $\mathrm{nsc}(L^c) = 2^{n-1}$

Lemma 4.34. *Let* $n \geq 3$. *There exists a ternary prefix-free language such that* $\mathrm{nsc}(L) = n$ *and* $\mathrm{nsc}(L^c) \geq 2^{n-1}$.

Proof. Let K be the language accepted by the NFA over $\{a, b\}$ shown in Figure 3.1 with $n-1$ states. Set $L = K \cdot c$. Then L is a prefix-free language recognized by an n-state NFA in Figure 4.17. As shown in [30, Theorem 5], there exists a fooling set $\mathcal{F} = \{(x_S, y_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ of size 2^{n-1} for the language K^c. Then the set of pairs of strings $\mathcal{F}' = \{(x_S, y_S \cdot c) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ is a fooling set of size 2^{n-1} for the L^c. $\square$

We summarize the results given in Lemma 4.33 and Lemma 4.34 in the following theorem which provides the tight bound on the nondeterministic state complexity of complementation on prefix-free languages.

Theorem 4.35 (**Complementation on prefix-free languages**). *Let $n \geq 3$. Let L be a prefix-free regular language over an alphabet Σ with* $\mathrm{nsc}(L) = n$. *Then* $\mathrm{nsc}(L^c) \leq 2^{n-1}$, *and the bound is tight if* $|\Sigma| \geq 3$. □

Now, let us turn our attention to a binary alphabet. Similarly as in the case of suffix-free language, we can apply the same homomorphism h on the ternary prefix-free language L from [32, Lemma 3] shown in Figure 4.17. We only have to be careful with the proof of prefix-free property of the language $h(L)$. Now every string in $h(L)$ ends by 00. The only proper prefix of a string in $h(L)$ which ends with 00 has an odd length. But such a string does not belong to $h(L)$. Therefore $h(L)$ is prefix-free.

We can construct NFA A for $h(L)$ with $2n$ states similarly as in the suffix-free case. The main difference between the automaton in case of binary suffix-free language and automaton for binary prefix-free language is the final state; see the Figure 4.18. Similarly as in suffix-free case we can prove that A is minimal and therefore $\mathrm{nsc}(h(L)) = 2n$. Finally, we use a similar approach to find a binary prefix-free language with an odd size of states, such that we add a new state n' and the transition from original final state n to n' on symbol 0. State n' become a new final state. Such a language is still prefix-free.

Hence we get the following result for binary prefix-free languages.

Theorem 4.36 (**Complementation on binary prefix-free languages; lower bound**). *Let $n \geq 6$. There is a binary prefix-free language L such that* $\mathrm{nsc}(L) = n$ *and* $\mathrm{nsc}(L^c) \geq 2^{\lfloor \frac{n}{2} \rfloor - 1}$. □

Lemma 4.31 and Theorem 4.36 give the following result.

Theorem 4.37 (**Complementation on binary prefix-free, suffix-free languages**). *Let $n \geq 12$. Let L be a binary prefix-free or suffix free language with* $\mathrm{nsc}(L) = n$. *Then* $\mathrm{nsc}(L^c) \leq 2^{n-1} - 2^{n-3} + 2$. *The lower bound is* $2^{\lfloor \frac{n}{2} \rfloor - 1}$. □

After investigation of prefix and suffix free languages we investigate other free classes of languages: *factor-free* and *subword-free languages.*

The next theorem provides a tight bound on the nondeterministic state complexity of complement on factor-free languages.

Theorem 4.38 (**Complementation on factor-free languages**). *Let $n \geq 3$. Let L be a factor-free language over an alphabet Σ such that* $\mathrm{nsc}(L) = n$. *Then* $\mathrm{nsc}(L^c) \leq 2^{n-2} + 1$, *and the bound is tight if* $|\Sigma| \geq 3$.

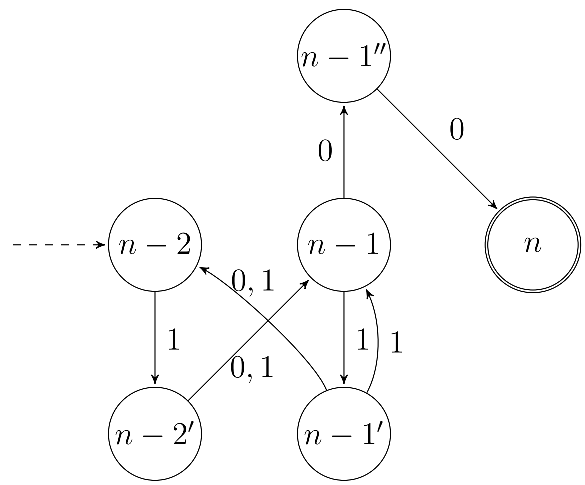

Figure 4.18: The last part of an NFA of prefix-free language $h(L)$

Proof. We first prove the upper bound. Let A be an n-state NFA for L. Since L is factor-free, it is suffix-free and also prefix-free. It follows that no transition goes to the initial state of A, and all the final states in the subset automaton are equivalent. Hence the subset automaton has at most $2^{n-2}+2$ reachable and pairwise distinguishable states. After exchanging the final and non-final states, we get a DFA for L^c of at most $2^{n-2}+2$ states. In the same way as for prefix-free languages in [32, Lemma 2], we can use a nondeterminism to save one state. This gives the upper bound $2^{n-2}+1$.

To prove tightness, consider the binary language G accepted by the $(n-2)$-state NFA N shown in Figure 3.1. Let $L = c \cdot G \cdot c$. Then L is accepted by an n-state NFA A shown in Figure 4.19.

Let $\mathcal{F} = \{(x_S, y_S) \mid S \subseteq \{1, 2, \ldots, n-2\}\}$ be a fooling set for the G^c [30, Theorem 5]. Notice that the strings x_S and y_S have the following properties: (1) by x_S, the initial state goes to the set S; (2) the string y_S is rejected by N from every state in S and it is accepted by N from every state in $\{1, 2, \ldots, n-2\} \setminus S$. Then the set of pairs of strings $\mathcal{F}' = \{(cx_S, y_S c) \mid S \subseteq \{1, 2, \ldots, n-2\}\}$ is a fooling set for L^c. Let

$\mathcal{A} = \{(cx_S, y_S c) \mid S \subseteq \{1, 2, \ldots, n-2\}$ and $S \neq \emptyset\}$,

$\mathcal{B} = \{(ca^{n-3}c, y_\emptyset c)\}$,

$u = y_\emptyset c$,

$v = \varepsilon$.

Let us show that $\mathcal{A}$, $\mathcal{B}$, u and v satisfy the conditions of Lemma 2.2. The set $\mathcal{A}$ is a fooling set for L^c since $\mathcal{A} \subseteq \mathcal{F}'$. The set $\mathcal{B}$ is fooling set for L^c, because the string $ca^{n-3}c \cdot y_\emptyset c$ is in L^c, because it contains three symbols c.

Notice that the string $y_\emptyset c$ is accepted by A from each state in the set $\{1, 2, \ldots, n-2\}$ since $y_\emptyset$ is accepted by N from each state in $\{1, 2, \ldots, n-2\}$ [30, Theorem 5]. Thus, if S is non-empty, then $cx_S \cdot y_\emptyset c \notin L^c$ since by cx_S the NFA A reaches the non-empty set S, from which it accepts $y_\emptyset c$. It follows that $\mathcal{A} \cup \mathcal{B}$ and $\mathcal{A} \cup \{(\varepsilon, u)\}$ are fooling sets for L^c. Also $\mathcal{B} \cup \{(\varepsilon, \varepsilon)\}$ is fooling set for L^c, because $\varepsilon \cdot \varepsilon \in L^c$ and $ca^{n-3}c \cdot \varepsilon \notin L^c$.

It follows that the conditions in Lemma 2.2 are satisfied, and therefore we have $\text{nsc}(L^c) \geq |\mathcal{A}| + |\mathcal{B}| + 1 = 2^{n-2} + 1$. □

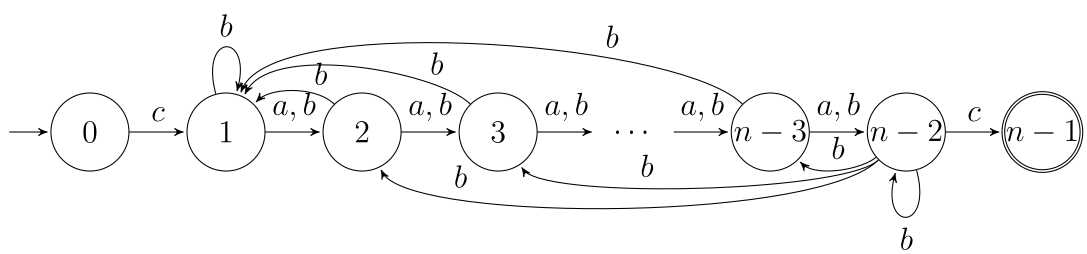

Figure 4.19: An NFA of a ternary factor-free language L with $\text{nsc}(L^c) = 2^{n-2} + 1$

It remains to find the bounds for the binary case.

Let us start with an upper bound. Let L be a binary factor-free language with $\text{nsc}(L) = n$ accepted by an n-state NFA N. The NFA N has to have the same properties as an automaton for a prefix or suffix free language. Thus there is just one final state with no outgoing transition and no transition goes to the initial state. We obtain a similar lemma as in the case of binary prefix-free languages in [32, Lemma 9].

Lemma 4.39. *There is a positive integer n_0 such that for every $n > n_0$, if L is a binary factor-free language with* $\text{nsc}(L) = n$ *then* $\text{nsc}(L^c) \leq 2^{n-2} - 2^{n-4} + 1$.

Proof. Let N be a minimal NFA for L. Let $\{1, 2, \ldots, n\}$ be the state set of N. Let n be the final state and 1 be initial state of N. Without loss of generality, the state n is reached from the state $n-1$ on a in N. Recall that no transition goes to state 1 because L is also a suffix-free language, so at most two subsets of states are reachable from 1. Therefore it is enough to consider subsets of set $\{2, 3, \ldots, n-1\}$.

If there is no transition (i, a, j) with $i, j \in \{2, 3, \ldots, n-1\}$, then the automaton on states $\{2, 3, \ldots, n-1\}$ is unary. It follows that in the subset automaton of N, at most $O(F(n-2))$ distinguishable subsets of $\{1, 2, \ldots, n-1\}$ can be reached. Since, starting from some positive integer n_0, we have $O(F(n-2)) < 2^{n-2} - 2^{n-4}$, the lemma follows in this case.

Now consider a transition (i, a, j) with $i, j \in \{2, 3, \ldots, n-1\}$. Let us show that no subset of $\{2, 3, \ldots, n-1\}$ containing states i and $n-1$ may be reachable. Assume for contradiction, that a set $S \cup \{i, n-1\}$ is reached from the initial state of the subset automaton by a string u. Since N is minimal, the final state n is reached from the state j by a non-empty string v. However, the set $S \cup \{i, n-1\}$ goes to a final set $S' \cup \{j, n\}$ by a, and then to a final set $S'' \cup \{n\}$ by v. It follows that the subset automaton accepts the strings ua and uav, which is a contradiction with the prefix-freeness of the accepted language. Thus at least 2^{n-4} subsets of $\{2, 3, \ldots, n-1\}$ are unreachable. Therefore, the subset automaton has at most $2^{n-2} - 2^{n-4} + 1$ states. After exchanging the accepting and the rejecting states we get a DFA of the same size for the complement of $L(N)$, and the lemma follows. $\square$

For the lower bound, let us consider the language $L = cGc$, where G is accepted by the $n-2$-state NFA shown in Figure 3.1. Then L is accepted by an n-state NFA A shown in Figure 4.19. By a similar strategy as in the binary case of prefix or suffix free language, we apply homomorphism h on the language L. Every string w in $h(L)$ has a form $001u1100$ or $001u1000$ and the string u does not contain string 00. So in the first case, any proper factor belonging to $h(L)$ does not exist. In the second case, every proper factor belonging to $h(L)$ has to have form $001u100$ but it has an odd length, and since every string in $h(L)$ has an even length, such a string is not in $h(L)$. So $h(L)$ is factor-free. We get an NFA A for $h(L)$ in a similar way as in cases suffix-free or prefix-free. The NFA A is minimal and has $2n$ states, so $\mathrm{nsc}(h(L)) = 2n$.

We deal with odd values of n similarly as before. Thus we get the following result.

Lemma 4.40. *Let $n \geq 8$. There is a binary factor-free language L such that* $\mathrm{nsc}(L) = n$ *and* $\mathrm{nsc}(L^c) \geq \Omega(2^{\frac{n}{2}})$. $\square$

We summarize our results on binary factor-free languages in the following theorem.

Theorem 4.41 (Complementation on binary factor-free languages). *There is a positive integer n_0 such that for every $n > n_0$, if L is a binary factor-free language with* $\mathrm{nsc}(L) = n$ *then* $\mathrm{nsc}(L^c) \leq 2^{n-2} - 2^{n-4} + 1$. *The lower bound is $\Omega(2^{\frac{n}{2}})$.* $\square$

The last class investigated is class of *subword-free* languages. We provide upper and lower bounds for nondeterministic state complexity of complement of such a language. We prove that bound is tight if alphabet has exponential size.

First, we prove the following observation.

Proposition 4.42. *Let L be a language. If L is subword-free, then L is finite.*

Proof. Let us assume for a contradiction that L is not finite. Let A be a minimal NFA for L. Since L is infinite, there is a state q in A and a non-empty string u such that $q \in \delta(q, u)$, where δ is a transition function of A and $u \neq \varepsilon$. Since A is minimal, there must be a string x by which q is reachable from the initial state, and a string y which is accepted from q. So the string xy belongs to the language L. The string xuy also belongs to L. However, the string xy is a proper subword of the string xuy. So L is not subword-free, which is contradiction. $\square$

Theorem 4.43 (Complementation on subword-free languages). *Let $n \geq 4$. Let L be a subword-free language over an alphabet Σ such that* $\text{nsc}(L) = n$. *Then* $\text{nsc}(L) \leq 2^{n-2} + 1$, *and the bound is tight if $|\Sigma| \geq 2^{n-2}$.*

Proof. The upper bound is the same as for factor-free languages. To prove tightness, let $\Sigma = \{a_S \mid S \subseteq \{1, 2, \ldots, n-2\}\}$ be an alphabet with 2^{n-2} symbols.

Consider the language L accepted by the NFA $A = (Q, \Sigma, \delta, 0, \{n-1\})$, where $Q = \{0, 1, \ldots, n-1\}$, and the transition function δ is defined as follows: for each symbol a_S in Σ,

$\delta(0, a_S) = S$;
$\delta(i, a_S) = \emptyset$ if $1 \leq i \leq n-2$ and $i \in S$;
$\delta(i, a_S) = \{n-1\}$ if $1 \leq i \leq n-2$ and $i \notin S$; and
$\delta(n-1, a_S) = \emptyset$.

The Figure 4.20 shows the NFA A with $n = 4$.

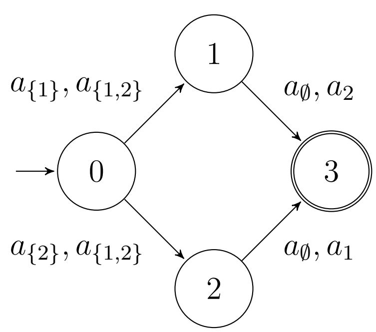

Figure 4.20: Example of a subword-free language with NFA of four states and alphabet $a_\emptyset, a_{\{1\}}, a_{\{2\}}, a_{\{1,2\}}$

Notice that each string in L is of length 2, so L is subword-free. Consider the set of pairs $\mathcal{F} = \{(a_S, a_S) \mid S \subseteq \{1, 2, \ldots, n-2\}\}$. Let us show that the set $\mathcal{F}$ is a fooling set for L^c.

(F1) For each S, the string $a_S a_S$ is in L^c since A goes to S by a_S and a_S is rejected by A from each state in S.

(F2) Let $S \neq T$. Then, without loss of generality, there is a state q in $\{1, 2, \ldots, n-2\}$ such that $q \in S$ and $q \notin T$. Then $a_S a_T$ in not in L^c since A goes to the state q by a_S, and then to the accepting state $n-1$ by a_T.

Hence $\mathcal{F}$ is a fooling set for L^c.

Let

$\mathcal{A} = \{(a_S, a_S) \mid S \subseteq \{1, 2, \ldots, n-2\}$ and $S \neq \emptyset\}$,

$\mathcal{B} = \{(a_{\{1\}} a_{\{2\}}, a_{\emptyset})\}$,

$u = a_{\emptyset}$,

$v = \varepsilon$.

Let us show that L^c, $\mathcal{A}$, $\mathcal{B}$, u, and v satisfy the condition in Lemma 2.2. The set $\mathcal{A}$ is a fooling set for L^c since $\mathcal{A} \subseteq \mathcal{F}$. The set $\mathcal{B}$ is fooling set for L^c, because the string $a_{\{1\}} a_{\{2\}} \cdot a_{\emptyset}$ is in L^c, because it contains three symbols.

Notice, if S is non-empty, then $a_S \cdot a_{\emptyset}$ is accepted by A, so $a_S \cdot a_{\emptyset} \notin L^c$. It follows that $\mathcal{A} \cup \mathcal{B}$ and $\mathcal{A} \cup \{(\varepsilon, a_{\emptyset})\}$ are fooling sets for L^c. Also $\mathcal{B} \cup \{(\varepsilon, \varepsilon)\}$ is fooling set for L^c, because $\varepsilon \cdot \varepsilon \in L^c$ and $a_{\{1\}} a_{\{2\}} \cdot \varepsilon \notin L^c$.

So the conditions in Lemma 2.2 are satisfied, therefore we have $\operatorname{nsc}(L^c) \geq 2^{n-2} + 1$. $\square$

4.4 Concluding remarks and open problems

Table 4.1 provides an overview of complexities of operations on unary-free languages and compares them to the known results on regular unary languages from [24]. Notice that the exact complexity of concatenation in the case of regular languages is still not known.

Table 4.2 summarizes our results on the nondeterministic complexity of operations on prefix-, suffix-, factor-, and subword-free languages and compares them to the results on regular languages which are from [24, 30]. Notice that the complexity of each operation in each class is always smaller than in the general case of regular languages, except for the reversal operation on suffix-free languages. All our wittnes languages are defined over small fixed alphabet which are always optimal, except for intersection and complementation on subword-free languages where it remains open whether the upper bounds can be met by subword-free languages defined over smaller alphabets. We conjecture that the bound mn is assymptotically tight for intersection of binary subword-free languages.

	$K \cap L$	$K \cup L$	KL	L^*	L^c
Unary free	$m = n$	$\max\{m, n\}$	$m + n - 1$	$n - 1$	$\Theta(\sqrt{n})$
Unary regular [24]	mn; $\gcd(m, n) = 1$	$m + n + 1$; $\gcd(m, n) = 1$	$\geq m + n - 1$ $\leq m + n$	$n + 1$	$2^{\Theta(\sqrt{n \log n})}$

Table 4.1: Nondeterministic complexity of operations on unary free languages.

Class	Regular [24, 30]		Prefix-free		Suffix-free		Factor-free		Subword-free	
$K \cap L$	mn	2	$mn-(m+n-2)$	2	.	2	$mn-2(m+n-3)$	2	.	$m+n-5$
$K \cup L$	$m+n+1$	2	$m+n$	2	$m+n-1$	2	$m+n-2$	2	.	2
KL	$m+n$	2	$m+n-1$	1	.	1	.	1	.	1
L^*	$n+1$	1	n	2	.	2	$n-1$	1	.	1
L^R	$n+1$	2	n	1	$n+1$	2	n	1	.	1
L^c	2^n	2	2^{n-1}	3	.	3	$2^{n-2}+1$	3	.	2^{n-2}

Table 4.2: Nondeterministic complexity of operations on free classes. The dot means that the complexity is the same as in the previous column.

Chapter 5

Closed Languages

Recall that a language L is prefix-closed if $w \in L$ implies that every prefix of w is in L. Suffix-, factor-, and subword-closed languages are defined analogously. In the first part of this chapter, we investigate properties of nondeterministic finite automata accepting closed languages. Then we examine the nondeterministic complexity of basic operations on the four classes of closed languages. We also study the unary case. We conclude the chapter by summarizing our results and stating some open problems.

5.1 Properties of closed languages

The next propositions say something about the characterization of NFA recognizing a prefix-closed and suffix-closed language.

Proposition 5.1 (Characterizations of NFA).

(a) A regular language is prefix-closed if and only if it is accepted by some NFA with all states final.

(b) A regular language is suffix-closed if and only if it is accepted by some NNFA with all states initial.

Proof. (a) $\Rightarrow$: Let $A = (Q, \Sigma, \delta, s, F)$ be an trim NFA for a prefix-free language L. If A does not have any non-final state we are done. If there are non-final states, they are not dead states, because A is trim. The non-final states we set as final. More formally, Let $A' = (Q, \Sigma, \delta, s, F')$, where $Q' = M$, $F' = Q$. We show, that, the automaton A' accepts the same language as automaton A.

First, let us to show $L(A') \subseteq L(A)$. Let $u \in L(A')$. Let computation on string u finishes in a state q. Since in A are every transitions and states as in A' (some of them may not be final), the same computation is also in A. If $q \in F$, then $u \in L(A)$. If $q \notin F$, then q is not dead state in A. So there exists a string v, such that it reaches some $p \in F$, from q. Therefore $uv \in L(A)$. But $L(A)$ is prefix-closed, so $u \in L(A)$.

Second, $L(A) \subseteq L(A')$, bacause every accepted string in A is also accepted in A'. Thus $L(A') = L(A)$ and A' has all states final.

$\Leftarrow$: Let A be an automaton with the all states final. Let $uv \in L(A)$. Then there is a computation $s \xrightarrow{u} p \xrightarrow{v} f$, where s is initial state and p, f are final states. Therefore also u belongs to $L(A)$. So, $L(A)$ is prefix-closed.

(b) $\Rightarrow$: Let A be a NFA for a suffix-closed language L. We can set every noninitial state to initial and get NNFA, which is also, accepting L.

$\Leftarrow$: Let A be an NNFA such that all states are initial. Let w be a string accepted by A and $w = uv$. There is a computation $q \xrightarrow{u} p \xrightarrow{v} f$, where f is final state. Since p is also initial state, also suffix v is accepted, hence language accepted by A is suffix-closed. $\square$

Proposition 5.2. *Let NFA $A = (Q, \Sigma, \delta, s, F)$ be minimal accepting language L. Then L is suffix-closed if and only if the next condition is satisfied: if a string w is accepted from a state q, then the w is accepted from the initial state s.*

Proof. Since A is minimal, there is a string u for arbitrary state q in Q, such that $q \in \delta(s, u)$ and a string v, such that $\delta(q, v) \cap F \neq \emptyset$.

$\Leftarrow$: Assume $uv \in L$. Then there is computation such that $s \xrightarrow{u} q \xrightarrow{v} f$, where $f \in F$. So there is computation $q \xrightarrow{v} f$ and by assumption also computation $s \xrightarrow{v} f'$, where $f' \in F$. So $v \in L$, hence L is suffix-closed.

$\Rightarrow$: Assume L is suffix-closed. Assume, there is a computation $q \xrightarrow{v} f$, where $f \in F$. NFA A is minimal, so there exists u, such that $s \xrightarrow{u} q$. Then there is computation $s \xrightarrow{u} q \xrightarrow{v} f$, so $uv \in L$. Since L is suffix-closed also $v \in L$, so there exists a computation $s \xrightarrow{v} f'$, where $f' \in F$. $\square$

In what follows we use several times the following useful observation about factor-closed languages.

Proposition 5.3. *A language L is factor-closed if and only if the L is prefix-closed and suffix-closed.*

Proof. $\Rightarrow$: It follows directly from definition of factor-closed.

$\Leftarrow$: Let $uvy \in L$. Then since L is suffix-closed, $vy \in L$ and since L is prefix-closed, the $v \in L$. $\square$

5.2 Unary closed languages

In this section we pay attention to unary closed languages. Consider prefix-closed language and two cases, finite language and infinite language. In the case of finite language, there is a string with maximum length, so every shorter strings also must be in the language. In the case of infinite language, for arbitrary positive integer i, there is a string w with length at least i and with this string every its prefix, so such a language is a^*. Moreover suffix-closed, factor-closed and subword-closed coincide.

Theorem 5.4. *Let K and L be two unary closed languages with* $\operatorname{nsc}(K) = m$ *and* $\operatorname{nsc}(L) = n$. *Then*

1. $\operatorname{nsc}(K \cup L) \leq \max\{m, n\}$,
2. $\operatorname{nsc}(K \cap L) \leq \min\{m, n\}$,
3. $\operatorname{nsc}(KL) \leq m + n - 1$,
4. $\operatorname{nsc}(L^*) \leq 1$,
5. $\operatorname{nsc}(L^R) \leq n$,
6. $\operatorname{nsc}(L^c) \leq n + 1$.

All these bounds are tight.

Proof. An unary closed language L with $\operatorname{nsc}(L) = 1$ is $\emptyset$ or a^* or $\{\varepsilon\}$. For $n \geq 2$, the unary closed language L with $\operatorname{nsc}(L) = n$ is the set $\{a^i \mid 0 \leq i \leq n-1\}$. This observation helps us to show that if $2 \leq m < n$,

1) the language $K \cup L$ is the same as L, because every string in K is in L;
2) the language $K \cap L$ is the same as K, for the same reason;
3) the language KL is the set of strings with maximal length $m - 1 + n - 1$, hence $\operatorname{nsc}(KL) = m + n - 1$;
4) the language L^* is the same as a^*, because the string a is in L, hence $\operatorname{nsc}(L^*) = 1$;
5) the language L^R is the same as L, what holds true for every unary language;
6) the language L^c is the set of strings with minimal length n, what needs $n+1$ states.

This case is proven by simple observation. Let us transform the unary automaton for L to complete DFA. We need to add a dead state, which is single nonfinal state. After exchanging finality we get automaton with single final state and $n + 1$ states. This is the minimal number. See the Figure 5.1. $\square$

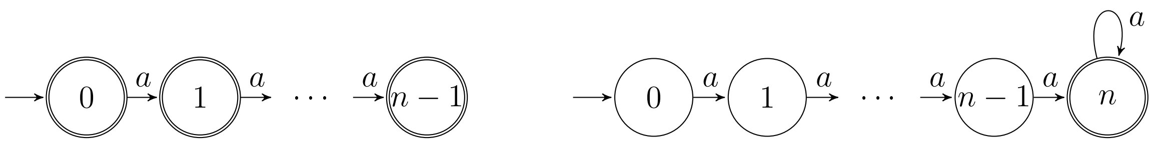

Figure 5.1: On the left is closed language and on the right its complement.

5.3 Operations on closed languages

We start with union and intersection on the class of closed languages.

Theorem 5.5 (Union). *Let* $m, n \geq 2$*. Let* K *and* L *be closed languages with* $\operatorname{nsc}(K) = m$ *and* $\operatorname{nsc}(L) = n$*. Then* $\operatorname{nsc}(K \cup L) \leq m + n + 1$*. The bound is met by binary subword closed languages.*

Proof. The upper bound is the same as for regular languages. To prove tightness, consider the binary languages shown in Figure 5.2. The language K consists of all strings such that each string contains at most $m-1$ symbols a, hence each subword contains also at most $m-1$ symbols a and such string belongs to the language K. Therefore K is subword-closed. Similarly, the language L is also subword-closed.

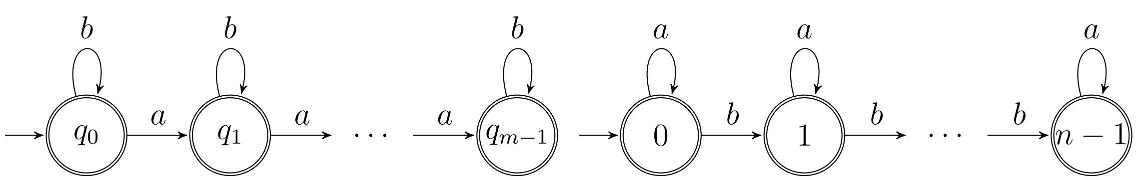

Figure 5.2: The DFAs of subword-closed languages K and L with $\operatorname{nsc}(K \cup L) = m+n+1$.

Consider the following sets of pairs of strings:

$$\mathcal{A} = \{(b^n a^i, a^{m-1-i} b) \mid 0 \leq i \leq m-1\},$$
$$\mathcal{B} = \{(ab^{n-1-j}, b^j a^m) \mid 0 \leq j \leq n-1\}.$$

Let us show that $\mathcal{A} \cup \mathcal{B}$ is a fooling set. Condition (F1) is satisfied since for each i, j, the strings $b^n a^i \cdot a^{m-1-i} b$ and $ab^{n-1-j} \cdot b^j a^m$ are in $K \cup L$. To prove (F2), we consider three cases:

(1) if $0 \leq i < k \leq m-1$, then $b^n a^k \cdot a^{m-1-i} b$ is not in $K \cup L$;
(2) if $0 \leq j < \ell \leq n-1$, then $ab^{n-1-j} \cdot b^\ell a^m$ is not in $K \cup L$;
(3) if $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$, then $b^n a^i \cdot b^j a^m$ is not in $K \cup L$.

In addition, $\mathcal{A} \cup \{(\varepsilon, a^m b^{n-1})\}$ and $\mathcal{B} \cup \{(\varepsilon, a^{m-1} b^n)\}$ are fooling sets for $K \cup L$. By Lemma 2.2, we have that $\operatorname{nsc}(K \cup L) \geq m + n + 1$. $\square$

Theorem 5.6 (Intersection). *Let $m, n \geq 2$. Let K and L be closed languages with* $\operatorname{nsc}(K) = m$ *and* $\operatorname{nsc}(L) = n$. *Then* $\operatorname{nsc}(K \cap L) \leq mn$. *The bound is met by binary subword-closed languages.*

Proof. The upper bound is the same as for regular languages. To prove tightness, consider the binary subword-closed languages shown in Figure 5.2. Consider the following set of pairs of strings:

$$\mathcal{F} = \{(a^i b^j, a^{m-1-i} b^{n-1-j}) \mid 0 \leq i \leq m-1, 0 \leq j \leq n-1\}.$$

Let us show that $\mathcal{F}$ is a fooling set for $K \cap L$. Condition (F1) is satisfied since for each i, j, the string $a^i b^j \cdot a^{m-1-i} b^{n-1-j}$ is in $K \cap L$. To prove (F2), let $(i,j) \neq (k,\ell)$. (1) If $i < k$, then $a^k b^\ell \cdot a^{m-1-i} b^{n-1-j}$ is not in $K \cap L$. (2) If $i = k$ and $j < \ell$, then $a^k b^\ell \cdot a^{m-1-i} b^{n-1-j}$ is not in $K \cap L$. Hence $\mathcal{F}$ is a fooling set for $K \cap L$, so $\operatorname{nsc}(K \cap L) \geq mn$. □

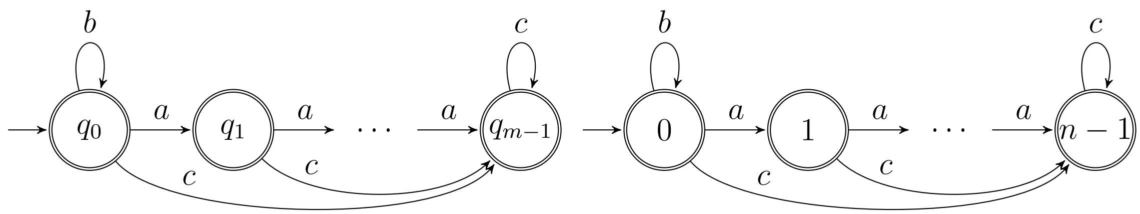

Figure 5.3: The subword-closed witnesses K, L for concatenation meeting the bound $m+n$.

Let us continue with concatenation.

Theorem 5.7 (Concatenation). *Let K and L be closed languages with* $\operatorname{nsc}(K) = m$ *and* $\operatorname{nsc}(L) = n$. *Then* $\operatorname{nsc}(KL) \leq m + n$. *The bound is met by ternary subword-closed languages.*

Proof. The upper bound is the same as for regular languages. To prove tightness, consider the ternary subword-closed languages shown in Figure 5.3. Consider the following set of pairs of strings:

$$\mathcal{F} = \{(a^i, a^{m-1-i} cba^{n-1}) \mid 0 \leq i \leq m-1\} \cup \{(a^{m-1} cba^j, a^{n-1-j}) \mid 0 \leq j \leq n-1\}.$$

Let us show that $\mathcal{F}$ is a fooling set for KL. Condition (F1) is satisfied since for each i, j, the strings $a^i \cdot a^{m-1-i} cba^{n-1}$ and $a^{m-1} cba^j \cdot a^{n-1-j}$ are in KL. To prove (F2), notice that KL is a subset of $b^* a^* c^* b^* a^* c^*$ and every string in KL has at most $m-1+n-1$ letters a. We consider three cases.

(1) If $0 \le i < k \le m-1$, then $a^k \cdot a^{m-1-i}cba^{n-1}$ is not in KL, because it has more than $m-1+n-1$ letters a.

(2) If $0 \le j < \ell \le n-1$, then $a^{m-1}cba^{\ell} \cdot a^{n-1-j}$ is not in KL, because it has more than $m-1+n-1$ letters a.

(3) If $0 \le i \le m-1$ and $0 \le j \le n-1$, then $a^{m-1}cba^j \cdot a^{m-1-i}cba^{n-1}$ is not in KL, because this string is not in the form $b^*a^*c^*b^*a^*c^*$.

Hence $\mathcal{F}$ is a fooling set for KL, so $\operatorname{nsc}(KL) \ge m+n$. □

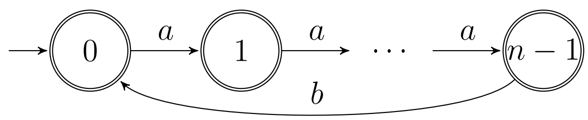

Figure 5.4: The prefix-closed witness language L for star and reversal.

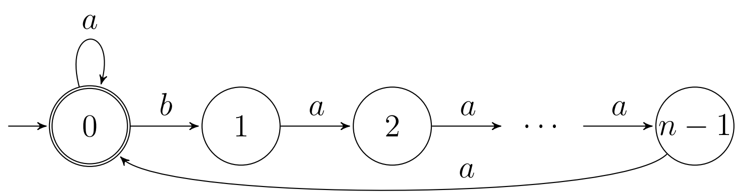

Figure 5.5: The suffix-closed witness language L for star meeting the bound n.

Theorem 5.8 (Star). *Let L be a closed language over Σ with $\operatorname{nsc}(L) = n$. Then*

(a) if L is prefix-closed, then $\operatorname{nsc}(L^*) \le n$, and the bound is tight if $|\Sigma| \ge 2$;

(b) if L is suffix-closed, then $\operatorname{nsc}(L^*) \le n$, and the bound is tight if $|\Sigma| \ge 2$;

(c) if L is factor- or subword-closed, then $\operatorname{nsc}(L^*) = 1$.

Proof. If L is a closed language, then $\varepsilon \in L$. It follows that $\operatorname{nsc}(L^*) \le n$. To prove tightness, consider a prefix-closed language shown in Figure 5.4 and a suffix-closed language shown in Figure 5.5. Lower bound for prefix-closed was proven in [11, Theorem 14], lower bound for suffix-closed is n because $L = L^*$. For factor- or subword-closed, let Γ be set of letters present in any string of L. While $L \subseteq \Gamma^*$, every single-letter string from Γ is in L. It follows that $L^* = \Gamma^*$, hence $\operatorname{nsc}(L^*) = 1$. □

Theorem 5.9 (Reversal). *Let $n \ge 3$ and L be a closed language with $\operatorname{nsc}(L) = n$. Then $\operatorname{nsc}(L^R) \le n+1$. The bound is met by a binary prefix-closed language, by a binary factor-closed language and by a subword-closed language over an alphabet of size $2n-2$.*

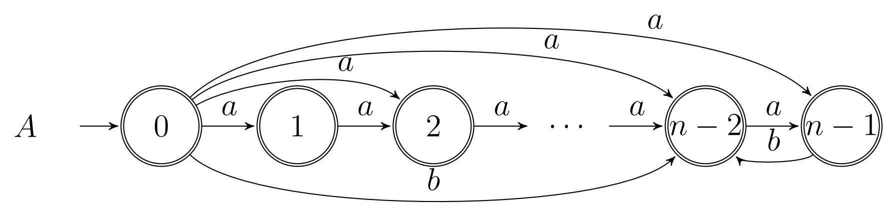

Figure 5.6: A binary factor-closed witness for reversal meeting the bound $n+1$.

Proof. The upper bound is the same as for regular languages. To prove tightness, consider the binary prefix-closed language shown in Figure 5.4. It was shown in [11] that the reversal of this language requires $n+1$ states.

Now we consider factor-closed case. Let L be the language accepted by the NFA $A = (Q, \{a, b\}, \cdot, 0, Q)$, where $Q = \{0, 1, \dots, n-1\}$ and the transitions are as follows:

$0 \cdot a = Q \setminus \{0\}$ and $i \cdot a = \{i+1\}$ if $1 \le i \le n-2$,

$0 \cdot b = \{n-2\}$ and $(n-1) \cdot b = \{n-2\}$,

and all the remaining transitions go to $\emptyset$. The NFA A is shown in Figure 5.6.

First we show that L is factor-closed. Since each state of A is final, the language L is prefix-closed. Next, for each transition (i, σ, j) there is the transition $(0, \sigma, j)$ in A. Hence if a string is accepted by A from a state i, then it is accepted also from the initial state 0. Therefore L is suffix-closed. Since L is prefix-closed and suffix-closed, it is factor-closed.

Now we show that every NFA for L^R needs at least $n+1$ states. Let

$\mathcal{A} = \{(ba^{n-1-i}, a^i) \mid 0 \le i \le n-2\}$,

$\mathcal{B} = \{(b, a^{n-1})\}$,

$u = a^{n-1}$, and $v = a^{n-2}$.

Notice that $\{ba^{n-1}, a^{n-1}, a^{n-2}\} \subseteq L^R$ and no string in L^R has more than $n-1$ consecutive a's. It follows that $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, u)\}$, and $\mathcal{B} \cup \{(\varepsilon, v)\}$ are fooling sets for L^R. By Lemma 2.2, we have $\operatorname{nsc}(L^R) \ge n+1$.

Finally consider the subword-closed language accepted by the DFA shown in Figure 5.7. Consider the following sets:

$\mathcal{A} = \{(b_2 b_3 \cdots b_{n-1}, a_1)\}$, $\mathcal{B} = \{(b_1 \cdots b_{i-1} b_{i+1} \cdots b_{n-1}, a_i) \mid 2 \le i \le n-1\} \cup \{(b_1 a_2, \varepsilon)\}$.

Let us show that $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, a_2)\}$ and $\mathcal{B} \cup \{(\varepsilon, a_1)\}$ are fooling sets for L^R. Condition (F1) for $\mathcal{A} \cup \mathcal{B}$ is satisfied because for every i the string $b_1 \cdots b_{i-1} b_{i+1} \cdots b_{n-1} \cdot a_i$ is in L^R. Next, for every $i \ne j$ the string $b_1 \cdots b_{i-1} b_{i+1} \cdots b_{n-1} \cdot a_j$ is not in L^R, because it has b_j before a_j. Hence (F2) is satisfied. The condition (F1) for $\mathcal{A} \cup \{(\varepsilon, a_2)\}$ and for $\mathcal{B} \cup \{(\varepsilon, a_1)\}$ is satisfied, because the strings a_2 and a_1 are in L^R. The proof of condition (F2) uses the same strings as for $\mathcal{A} \cup \mathcal{B}$. $\square$

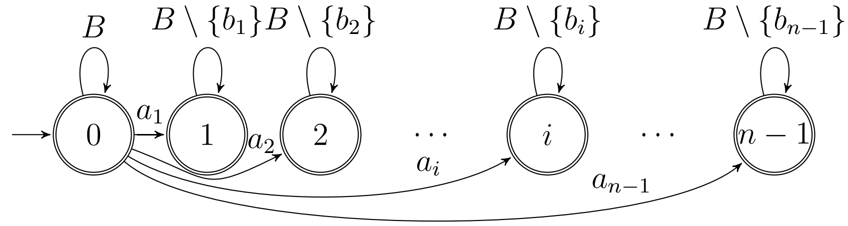

Figure 5.7: The DFA of subword-closed language L where $B = \{b_1, \ldots, b_{n-1}\}$

.

The next lemma provides a binary factor-closed witness language for reversal meeting the bound $n + 1$, which improves the result from [26, Theorem 9] by reducing the size of alphabet from three to two. This language is also a binary factor-convex witness.

We conclude this section with the complementation operation. In [11], a ternary prefix-closed language meeting the upper bound 2^n for complement was described. Now we describe a binary witness language.

Theorem 5.10 (**Complementation**). *Let L be a closed language over Σ with* $\mathrm{nsc}(L) = n$. *Then*

(a) if L is prefix-closed, then $\mathrm{nsc}(L^c) \le 2^n$, *and the bound is tight if* $|\Sigma| \ge 2$;

(b) if L is suffix-closed, then $\mathrm{nsc}(L^c) \le 2^{n-1} + 1$, *and the bound is met by a binary factor-closed language;*

(c) if L is subword-closed, then $\mathrm{nsc}(L^c) \le 2^{n-1} + 1$, *and the bound is tight if* $|\Sigma| \ge 2^n$.

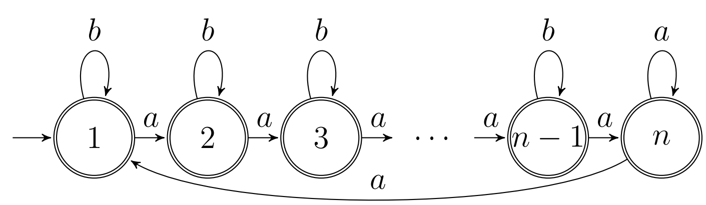

Figure 5.8: The NFA of binary witness prefix-closed language L with $\mathrm{nsc}(L^c) = 2^n$.

Proof. (a) The upper bound is the same as for regular languages. To prove tightness, let L be the binary language accepted by the NFA A shown in Figure 5.8. First, we prove the reachability of every subset of $\{1, 2, \ldots, n\}$ in the subset automaton of A. Notice that we have $\{1\} \xrightarrow{a^{n-1}} \{n\} \xrightarrow{a^{n-1}} \{1, 2, \ldots, n\}$. Next, we can shift cyclically by one every subset

S: we use the string a if $n \notin S$ or if $n \in S$ and $n-1 \in S$, and we use the string ab otherwise. Finally, we can remove state n from any subset containing n by b. It follows that every subset of $\{1, 2, \ldots, n\}$ is reachable. Thus for every set S, there exists a string u_S such that u_S leads the subset automaton from $\{1\}$ to S.

Now, we define a fooling set for complement of L. For every set S we define a string v_S as follows. First we define $\sigma(i)$, where $i \in \{1, 2, \ldots, n\}$ as

$$\sigma(i) = \begin{cases} ba, & \text{if } i \in S, \\ a, & \text{if } i \notin S. \end{cases}$$

Let $v_S = \sigma(n)\sigma(n-1)\ldots\sigma(2)\sigma(1)$. We show, that such a string is rejected by A from every $i \in S$ and accepted from every $i \notin S$. Let $i \notin S$, then $\sigma(i) = a$, and
$i \xrightarrow{\sigma(n)} i+1 \xrightarrow{\sigma(n-1)} i+2 \xrightarrow{\sigma(n-2)} \cdots \xrightarrow{\sigma(i+1)} n \xrightarrow{a} 1 \xrightarrow{\sigma(i-1)\ldots\sigma(1)} i,$
so v_S is accepted since every state is final. If $i \in S$, then $\sigma(i) = ba$, and
$i \xrightarrow{\sigma(n)} \{i+1\} \xrightarrow{\sigma(n-1)} \{i+2\} \xrightarrow{\sigma(n-2)} \cdots \xrightarrow{\sigma(i+1)} \{n\},$
and now A reads the first symbol of $\sigma(i)$ which is b. However, transition on b is not defined in state n, therefore the string v_S is rejected.

Now we show that $\mathcal{F} = \{(u_S, v_S) \mid S \subseteq \{1, 2, \ldots, n\}\}$ is a fooling set for L^c.

(F1) Let $S \subseteq \{1, 2, \ldots, n\}$. The NFA A reaches subset S by u_S, and from every state $q \in S$ the string v_S is rejected. So $u_S v_S$ is rejected by A, so $u_S v_S \in L^c$.

(F2) Let $S, T \subseteq \{1, 2, \ldots, n\}$ and $S \neq T$. Without loss of generality, there exists a state i, such that $i \in S$ and $i \notin T$. So v_T is accepted from i. Hence $u_S v_T$ is accepted by A, and therefore $u_S v_T \notin L^c$. This completes the proof of (a).

(b) We first prove the upper bound. Let $A = (Q, \Sigma, \delta, s, F)$ be an minimal NFA, such that $L(A) = L$. Since A is a minimal NFA, every q in Q is reachable from s and also some final state is reachable from q. Let a state $q \in Q$ be reachable from s by a string u. If a final state is reachable from q by string v, then also uv reaches a final state, so uv is accepted. Since L is suffix-closed, the string v reaches a final state from s. Therefore every subset of Q containing s is equivalent to $\{s\}$ in the subset automaton of NFA A. So subset automaton of A has at most $2^{n-1} + 1$, so $\text{nsc}(L^c) \leq 2^{n-1} + 1$.

To prove tightness, consider a language L accepted by automaton in Figure 5.9. If there is an accepting computation from a state q on a string u such that $q \xrightarrow{a(b)} q' \xrightarrow{u'} f$, where $u = au'$ or $u = bu'$ and f is a final state, then there is a computation $s \xrightarrow{a(b)} q' \xrightarrow{u'} f$. It follows that L is suffix-closed. Therefore L is factor-closed. First, we prove the reachability of every subset of $\{1, 2, \ldots, n-1\}$ in the subset automaton of A. Notice that we have $\{0\} \xrightarrow{a} \{1, 2, \ldots, n-1\}$. Next, we can shift cyclically by one every subset S by using

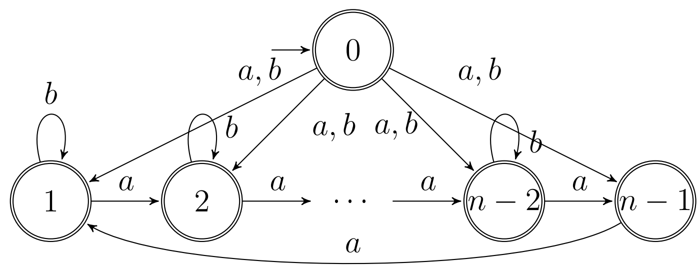

Figure 5.9: The factor-closed witness L for complement, with $\mathrm{nsc}(L^c) = 2^{n-1} + 1$.

the string a. Finally, we can remove state $n-1$ from any subset containing $n-1$ by b. It follows that every subset of $\{1, 2, \ldots, n-1\}$ is reachable. Thus for every set S, there exists a string u_S such that u_S leads the subset automaton from $\{0\}$ to S. Now, we define a fooling set for complement of L. For every set S we define a string v_S as follows. First we define $\sigma(i)$, where $i \in \{1, 2, \ldots, n-1\}$ as $\sigma(i) = ba$ if $i \in S$, and $\sigma(i) = a$ if $i \notin S$. Let $v_S = \sigma(n-1)\sigma(n-2)\cdots\sigma(2)\sigma(1)$. Similarly as in proof in case of prefix-closed in (a) we can show that such a string is rejected by A from every $i \in S$ and accepted from every $i \notin S$. Let $\mathcal{A} = \{(u_S, v_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$. We can show that $\mathcal{F} = \mathcal{A} \cup \{(\varepsilon, (ba)^n)\}$ is a fooling set for L^c.

(c) Since subword-closed language is also factor-closed, the upper bound is $2^{n-1} + 1$. To prove tightness consider an NFA A, defined as follows:
$A = (Q, \Sigma, \delta, s, F)$, where $Q = \{0, 1, 2, \ldots, n-1\}, s = 0, F = Q$ and $\Sigma = \{a_S, b_S \mid S \subseteq \{1, 2, \ldots, n-1\}\}$, $\delta(0, a_S) = S$, for $i > 0$ $\delta(i, a_S) = \emptyset$, $\delta(0, b_S) = 0$, for $i > 0$: if $i \notin S$, then $\delta(i, b_S) = \{i\}$ and if $i \in S$, then $\delta(i, b_S) = \emptyset$. Such an NFA is shown in Figure 5.10.

Consider now the language $L = L(A)$. Let $w \in L$. The string w is accepted in a i$\in S$. Any substring of w is accepted also in the i. Hence L is subword-closed. We can show that $\mathcal{A} = \{(a_S, b_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\} \cup \{(\varepsilon, a_\emptyset\}$ is fooling set for L^c. Therefore $\mathrm{nsc}(L^c) \geq 2^{n-1} + 1$. $\square$

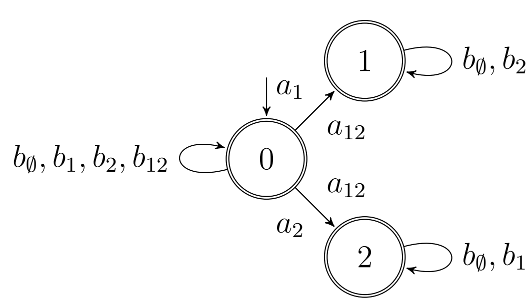

Figure 5.10: The subword-closed witness language L with $\mathrm{nsc}(L) = 3$ and $|\Sigma| = 2^n$.

5.4 Concluding remarks and open problems

We investigated the nondeterministic state complexity of basic regular operations on the classes of closed languages. For each class and for each operation, we obtained the tight upper bounds. To prove tightness we usually used a binary alphabet. In all the cases where we used a larger alphabet for describing witness languages, it remains open whether the obtained upper bounds can be met also by languages defined over smaller alphabets. We also considered the unary case. Our results are summarized in the following tables. The tables also display the size of alphabet used to describe witness languages.

Class	$K \cap L$	$\|\Sigma\|$	$K \cup L$	$\|\Sigma\|$	$K \cdot L$	$\|\Sigma\|$
Prefix-closed	mn	2	$m+n+1$	2	$m+n$	3
Suffix-closed	mn	2	$m+n+1$	2	$m+n$	3
Factor-closed	mn	2	$m+n+1$	2	$m+n$	3
Subword-closed	mn	2	$m+n+1$	2	$m+n$	3
Unary closed	$\min(m,n)$		$\max(m,n)$		$m+n-1$	
Regular	mn	2	$m+n+1$	2	$m+n$	2
Unary regular	mn;		$m+n+1$;		$\geq m+n-1$	
	$\gcd(m,n)=1$		$\gcd(m,n)=1$		$\leq m+n$	

Table 5.1: The nondeterministic complexity of union, intersection, and concatenation on closed languages. The results for regular languages are from [24].

Class	L^*	$\|\Sigma\|$	L^R	$\|\Sigma\|$	L^c	$\|\Sigma\|$
Prefix-closed	n	2	$n+1$	2	2^n	2
Suffix-closed	n	2	$n+1$	3	$1+2^{n-1}$	2
Factor-closed	1	1	$n+1$	3	$1+2^{n-1}$	2
Subword-closed	1	1	$n+1$	$2n-2$	$1+2^{n-1}$	2^n
Unary closed	1		n		$n-1$	
Regular	$n+1$	1	$n+1$	2	2^n	2
Unary regular	$n+1$		n		$2^{\Theta(\sqrt{n \log n})}$	

Table 5.2: The nondeterministic complexity of star, reversal, and complementation on closed languages. The results for regular languages are from [24, 30].

Chapter 6

Ideal Languages

In this section we focus on ideal languages. Recall that a language L over an alphabet Σ is a right (left, two-sided, all-sided) ideal if $L = L\Sigma^*$ ($L = \Sigma^* L$, $L = \Sigma^* L \Sigma^*$, $L = L \sqcup\!\sqcup\, \Sigma^*$, respectively). We again get tight upper bounds on the nondeterministic complexity of basic operations in each of these subclasses.

6.1 Properties of ideal languages

In this section we state and prove some useful propositions about some features of automata for ideal languages.

Proposition 6.1. *Let L be a regular language.*

1. *If L is a left ideal, then there exists a minimal NFA A such that $L(A) = L$ and there is a loop on each symbol in the initial state and no transition goes to the initial state from any other state.*

2. *If L is a right ideal, then there exists a minimal NFA A such that $L(A) = L$ and there is the unique final state in which there is a loop on each symbol and from which no transition goes to any other state.*

Proof. (a) Let A be a minimal NFA for L and s be the initial state.

Construct A' from A by adding loops on every symbol in s and by removing every transition going to s from other states.

If $w \in L(A')$, we can split w to two strings u, v such that $w = uv$ and there is a computation such that after reading u, the initial state s occurs the last time and during reading v no added transition is used. So v is accepted in A. Since $L(A)$ is a left ideal, $uv \in L(A)$. Therefore $w \in L(A)$.

If $w \in L(A)$, we can split w to two strings u, v such that $w = uv$ and there is a computation such that after reading u, the initial state s occurs the last time and during reading v, no transition goes to s. So every used transition is also in A', so $v \in L(A')$. Since there is a loop on every symbol in s in A', string u is possible to read in s and continue by reading v. Therefore $uv \in L(A')$, so $w \in L(A')$.

So, $L(A) = L(A')$ and A' is an NFA with required properties.

(b) Let A be a minimal NFA for L and s be the initial state.

Construct A' from A by adding loops on every symbol in every final states and by removing every transition going out from every final state to other state.

If $w \in L(A)$, then we can split w to two strings u, v such that $w = uv$ and there is a computation such that after reading u a final state occurs the first time, so during reading u, no transition going from some final state is used. So u is accepted also in A'. Since in every final state there is a loop on every symbol, the string v is possible to read in a final state, so $w \in L(A')$.

If $w \in L(A')$, then we can split w to two strings u, v such that $w = uv$ and there is a computation such that after reading u a final state occurs the first time, so during reading u no transition going from final state is used. So u is accepted also in A. Since $L(A)$ is a right ideal, $uv \in L(A)$. So $w \in L(A)$.

So $L(A) = L(A')$.

Notice that A' has only one final state. Otherwise all final states would be equivalent and we could merge them into one. But it would be the contradiction with minimality of A. □

Proposition 6.2. *Let L be a language over Σ and let A be a minimal NFA such that $L(A) = L$. Language L is two-sided ideal if and only if there is a minimal NFA A with initial state with a loop on every input and no in-transition from some other state and just one final state with a loop on every input and no out-transition to some other state.*

Proof. A language L is two-sided ideal if and only if it is left ideal and right ideal, therefore proposition follows from Proposition 6.1. □

Proposition 6.3. *Let L be a language over Σ. Language L is all-sided ideal if and only if there is a minimal NFA A with just one final state and with a loop in every state on every letter of an alphabet Σ, such that $L(A) = L$.*

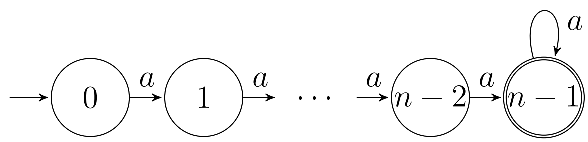

Figure 6.1: Minimal NFA for language $a^k, k \geq n-1$

Proof. $\Rightarrow$: A language L is all-sided ideal, then also it is right ideal, hence by Proposition 6.1 there is just one final state with loop on every input symbol. Every state is reachable and as well as from every state is possible to get to final state. Let us consider a state q. There are strings u, v, such that u leads from initial state to q and v leads from q to final state. Since L is all-sided ideal, every string $u \cdot \Sigma^* \cdot v$ is accepted by A, so we can add a loop in q on every input symbol.

$\Leftarrow$: Since in every state is a loop on every input symbol, we can insert in every position of any accepted string arbitrary string, what means that L is all-sided. □

6.2 Unary ideal languages

In the end of this chapter we pay attention to unary ideal languages. Let $\Sigma = \{a\}$. If L is a right ideal and a^i is its shortest string, then $L = a^i a^*$. Moreover $L = a^* a^i = a^* a^i a^* = a^* \sqcup\!\sqcup\, a^i$, hence left, right, two-sided and all-sided ideals coincide. An NFA for such language L has form shown in Figure 6.1.

Theorem 6.4. *Let $m, n \geq 2$. Let K, L be unary ideals with* $\operatorname{nsc}(K) = m, \operatorname{nsc}(L) = n$. *Then*

(a) $\operatorname{nsc}(K \cap L) = \max\{m, n\}$,

(b) $\operatorname{nsc}(K \cup L) = \min\{m, n\}$,

(c) $\operatorname{nsc}(KL) = m + n - 1$,

(d) $\operatorname{nsc}(L^*) = n - 1$,

(e) $\operatorname{nsc}(L^R) = n$,

(f) $\operatorname{nsc}(L^c) = n - 1$.

Proof. (a) Let $k = \max\{m, n\}$. The string a^{k-1} is the shortest string in $K \cap L$, so $K \cap L = a^{k-1} a^*$, therefore $\operatorname{nsc}(K \cap L) = \max\{m, n\}$.

(b) Let $k = \min\{m, n\}$. The string a^{k-1} is the shortest string in $K \cup L$, so $K \cup L = a^{k-1}a^*$, therefore $\mathrm{nsc}(K \cup L) = \min\{m, n\}$.

(c) It follows directly from Theorem 6.7, because witness languages were over unary alphabet.

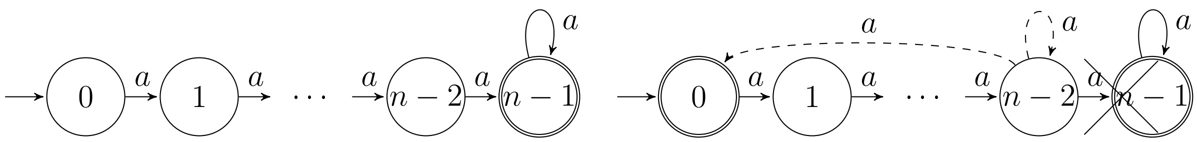

Figure 6.2: The construction an unary NFA for L^*

(d) Let A be minimal NFA for L. We can get NFA C for L^* from A by applying three next steps: (1) omit state $n-1$ with all connected transitions, (2) state 0 set as final, (3) add transitions $(n-2, a, n-2), (n-2, a, 0)$. The construction an NFA for L^* is shown in in Figure 6.2, where dashed lines are added transitions, state 0 is initial and final and crossed state and transitions are omited from A.

NFA C has $n-1$ states, therefore $\mathrm{nsc}(L^*) \le n-1$. To prove tightness consider set of pairs $\mathcal{F} = \{(a^i, a^{n-1-i}) \mid 0 \le i \le n-2\}$. Notice that every string in L^* has length 0 or at least $n-1$. The set $\mathcal{F}$ is fooling set for L^*, so $\mathrm{nsc}(L^*) \ge n-1$. Hence $\mathrm{nsc}(L^*) = n-1$.

(e) A reversal of unary string is the same, so $L^R = L$, therefore $\mathrm{nsc}(L^R) = n$.

(f) Let A be minimal NFA for L. The form of such automaton is shown in Figure 6.1. The automaton is also deterministic, so we can interchange final and non-final states to get NFA A' for complement L^c. In A' we can omit the state $n-1$, because it is dead state. So $\mathrm{nsc}(A') = n-1$.

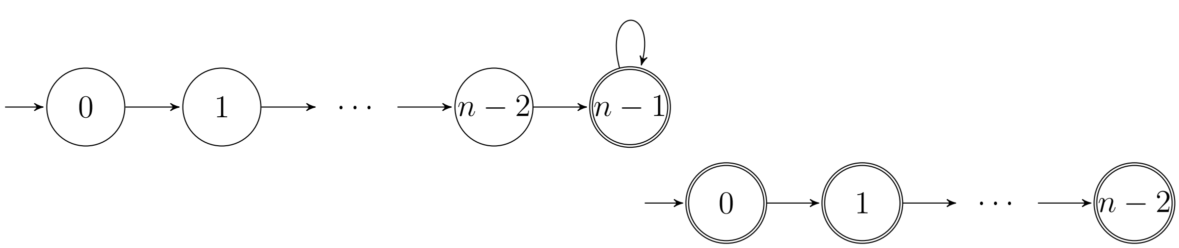

Figure 6.3: Unary ideal language and its complement

$\square$

6.3 Operations on ideal languages

First we consider the intersection operation on ideal languages.

Theorem 6.5 **(Intersection).** *Let $m, n \geq 1$. Let K and L be ideal languages with* $\mathrm{nsc}(K) = m$ *and* $nsc(L) = n$. *Then* $\mathrm{nsc}(K \cap L) \leq mn$. *The bound is met by binary all-sided ideals.*

Proof. The upper bound mn holds since it holds for regular languages. For tightness, consider the binary all-sided ideals

$K = \{w \in \{a,b\}^* \mid \#_a(w) \geq m-1\}$ and

$L \ = \{w \in \{a,b\}^* \mid \#_b(w) \geq \ n-1\}$.

with $\mathrm{nsc}(K) = m$ and $nsc(L) = n$. Let $\mathcal{F} = \{(a^i b^j, a^{m-1-i} b^{n-1-j} \mid 0 \leq i \leq m-1$ and $0 \leq j \leq n-1\}$ be a set of mn pairs. To prove the theorem, we only need to show that $\mathcal{F}$ is a fooling set for $K \cap L$. The concatenation of the first and the second component of each pair in $\mathcal{F}$ gives a string w with $\#_a(w) = m-1$ and $\#_b(w) = n-1$. Since all such strings are in $K \cap L$, condition (F1) is satisfied. To prove (F2), let $(i,j) \neq (k,\ell)$. If $i < k$, then $a^i b^j \cdot a^{m-1-k} b^{n-1-\ell}$ has less then $m-1$ a's, so it is not in $K \cap L$. If $i = k$ and $j < \ell$, then the string $a^i b^j \cdot a^{m-1-i} b^{n-1-\ell}$ has less than $n-1$ b's, so it is not in $K \cap L$. Hence $\mathcal{F}$ is a fooling set for $K \cap L$. $\square$

We continue with the union operation.

Theorem 6.6 **(Union).** *Let $m, n \geq 3$. Let K and L be ideal languages over an alphabet Σ with* $\mathrm{nsc}(K) = m$ *and* $nsc(L) = n$. *Then*

(a) if K, L are right ideals, then $\mathrm{nsc}(K \cup L) \leq m + n$,

(b) if K, L are left ideals, then $\mathrm{nsc}(K \cup L) \leq m + n - 1$,

(c) if K, L are two-sided or all-sided ideals, then $\mathrm{nsc}(K \cup L) \leq m + n - 2$, *and all the bounds are tight if $|\Sigma| \geq 2$.*

Proof. (a) We first prove the upper bound. Let A be a minimal m-state NFA for K and B be a minimal n-state NFA for L. Since K and L are right ideals, A and B have exactly one final state which goes to itself on each symbol. We can get an ε-NFA for $K \cup L$ from NFAs A and B by merging the final states of A and B and by adding a new initial state connnected to the initial states of A and B by ε-transitions. The resulting ε-NFA has $m + n$ states, so the corresponding NFA for $K \cup L$ has also $m + n$ states.

To prove tightness, consider the binary right ideals K and L shown in Figure 6.4.

Now we show that minimal NFA for $K \cup L$ needs $m + n$ states.

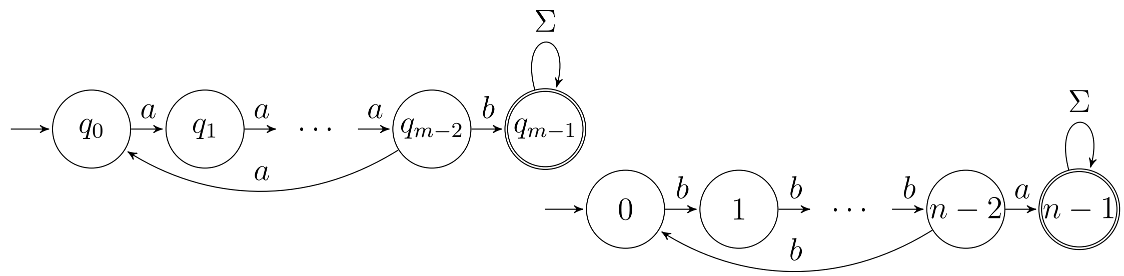

Figure 6.4: Witnesses right ideals for union.

To this aim let
$\mathcal{A} = \{(a^{m-1+i}, a^{m-2-i}b) \mid 0 \leq i \leq m-2\} \cup \{(a^{m-2}b, \varepsilon)\}$, and
$\mathcal{B} = \{(b^{n-1+j}, b^{n-2-j}a) \mid 0 \leq j \leq n-2\}$.
The sets $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, b^{n-2}a)\}$ and $\mathcal{B} \cup \{(\varepsilon, a^{m-2}b)\}$ are fooling sets. We first prove that $\mathcal{A}$ is fooling set. Since $a^{m-1+i}a^{m-2-i}b = a^{m-1}a^{m-2}b \in K$, and $a^{m-2}b \in K$, condition (F1) is satisfied. To prove (F2) we have two cases:

(1) Consider two pairs of forms $(a^{m-1+i}, a^{m-2-i}b)$ and $(a^{m-1+j}, a^{m-2-j}b)$ where $0 \leq i < j \leq m-2$. Then $a^{m-1+i} \cdot a^{m-2-j}b = a^{m-1+(m-2-(j-i))}b$. After reading the string $a^{m-1+(m-2-(j-i))}$ NFA A is in the state $m-2-(j-i)$, in which there is no transition on b since $m-2-(j-i) < m-2$. So that string is rejected by NFA A. The string is rejected also by NFA B, since in the initial state of B, there is no transition on a. Hence, $a^{m-1+i} \cdot a^{m-2-j}b \notin K \cup L$.

(2) Consider a pair of a form $(a^{m-1+i}, a^{m-2-i}b)$ with $0 \leq i \leq m-2$ and the pair $(a^{m-2}b, \varepsilon)$. Then $a^{m-1+i} \cdot \varepsilon \notin K \cup L$ because it does not contain any symbol b.

Hence $\mathcal{A}$ is fooling set.

A proof that $\mathcal{B}$ is fooling set is symmetric to case (1).

Now consider one pair from $\mathcal{A}$ and one pair from $\mathcal{B}$. To prove (F2) we have two cases:

(1) Consider two pairs of forms $(a^{m-1+i}, a^{m-2-i}b)$ and $(b^{n-1+j}, b^{n-2-j}a)$ where $0 \leq i \leq m-2$ and $0 \leq j \leq n-2$. If $i < m-2$, then $a^{m-1+i} \cdot b^{n-2-j}a$ is rejected by NFA A, because in state i there is no transition on b, and in NFA B it is rejected immediately in the initial state, in which there is no transition on a. In the case $j = n-2$, the string does not contain any b. Hence $a^{m-1+i} \cdot b^{n-2-j}a \notin K \cup L$. If $i = m-2$, then $b^{n-1+j} \cdot a^{m-2-(m-2)}b = b^{n+j}$ does not contain any a, so it is rejected by both NFA A, B. Hence $b^{n-1+j} \cdot a^{m-2-(m-2)}b = b^{j+1} \notin K \cup L$.

(2) For pairs $(a^{m-2}b, \varepsilon)$ and $(b^{n-1+j}, b^{n-2-j}a)$ with $0 \leq j \leq n-2$, we have $b^{n-1+j} \cdot \varepsilon \notin K \cup L$ because this string does not contain any symbol a.

Hence $\mathcal{A} \cup \mathcal{B}$ is a fooling set.

In case $\mathcal{A} \cup \{(\varepsilon, b^{n-2}a)\}$, we have $\varepsilon \cdot b^{n-2}a \in K \cup L$, so condition (F1) is satisfied. Now we prove condition (F2). For a pair $(a^{m-1+i}, a^{m-2-i}b)$ with $0 \leq i \leq m-2$, the string $\varepsilon \cdot a^{m-2-i}b$ is not in $K \cup L$. For the pair $(a^{m-2}b, \varepsilon)$, the string $\varepsilon \cdot \varepsilon$ is not in $K \cup L$. Hence, $\mathcal{A} \cup \{(\varepsilon, b^{n-2}a)\}$ is a fooling set. In the case of $\mathcal{B} \cup \{(\varepsilon, a^{m-2}b)\}$ the situation is similar.

By Lemma 2.2 we have $\operatorname{nsc}(K \cup L) \geq |\mathcal{A}| + |\mathcal{B}| + 1 = m + n$.

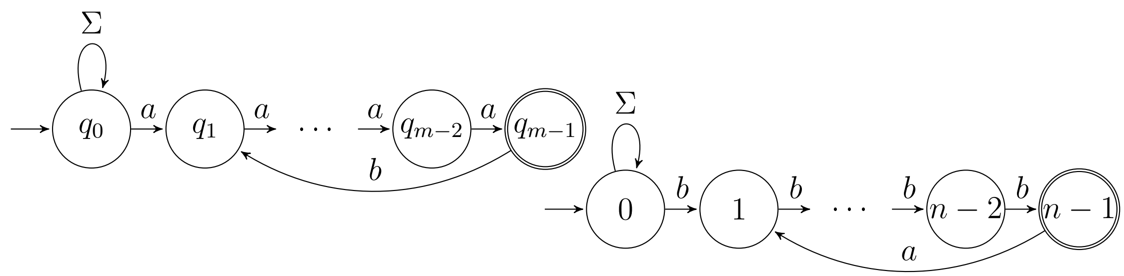

Figure 6.5: Witnesses left ideals for union.

(b) We first prove the upper bound. Let A be a minimal m-state NFA for K and B be a minimal n-state NFA for L. Since K and L are left ideals, we may assume by Proposition 6.1 that A and B have a loop on each symbol in the initial state, and no transition from some other state goes to the initial state.

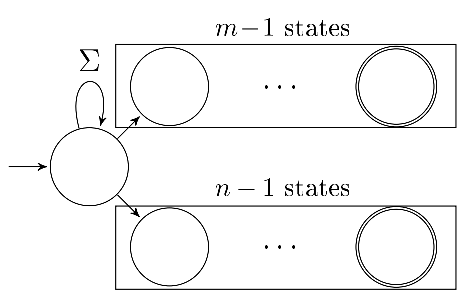

Figure 6.6: General construction of automaton for union of left ideals

We can get an NFA for $K \cup L$ from NFAs A and B by merging the initial states.

All original transitions from initial states of NFAs A, B go from new merged state to states as before merging. See Figure 6.6. The resulting NFA has $m + n - 1$ states, so $\operatorname{nsc}(K \cup L) \leq m + n - 1$.

To prove tightness, consider two left ideals shown in Figure 6.5. Now we show that minimal NFA for $K \cup L$ needs $m + n - 1$ states. To this aim let $\mathcal{A} = \{(a^i, a^{m-1-i}) \mid 0 \leq$

$i \leq m-1\}$ and $\mathcal{B} = \{(b^j, b^{n-1-j}) \mid 1 \leq j \leq n-2\} \cup \{(b^{n-1}, ab^{n-2})\}$. The set $\mathcal{A} \cup \mathcal{B}$ is fooling set for $K \cup L$, so $\text{nsc}(K \cup L) \geq m+n-1$, therefore $\text{nsc}(K \cup L) = m+n-1$.

(c) For upper bound, let A be a minimal m-state NFA for K and B be a minimal n-state NFA for L. Since K and L are left ideals and also right ideals, we may assume by Proposition 6.1 that A and B have properties claimed there. We can get an NFA for $K \cup L$ from NFAs A and B by merging the initial states, and by merging the final states of A and B. The resulting NFA has $m+n-2$ states and we leave to the reader to verify the corectness of the construction. To prove tightness, consider languages $K = \{w \in \{a,b\}^* \mid \#_a(w) \geq m-1\}$ and $L = \{w \in \{a,b\}^* \mid \#_b(w) \geq n-1\}$, so K and L are all-sided ideals. Notice that each string in $K \cup L$ has at least $m-1$ symbols a or at least $n-1$ symbols b. Let $\mathcal{A} = \{(a^i, a^{m-1-i}) \mid 0 \leq i \leq m-1\}$ and $\mathcal{B} = \{(b^j, b^{n-1-j}) \mid 1 \leq j \leq n-2\}$. The set $\mathcal{A} \cup \mathcal{B}$ is fooling set for $K \cup L$ and contains $m+n-2$ pairs, so $\text{nsc}(K \cup L) \geq n+m-2$. $\square$

In the next theorem we consider the concatenation operation and we use unary ideals to prove tightness.

Theorem 6.7 (Concatenation). *Let $m, n \geq 3$. Let K and L be ideal languages over Σ with $\text{nsc}(K) = m$ and $\text{nsc}(L) = n$. Then $\text{nsc}(KL) \leq m+n-1$ and the bound is tight if $|\Sigma| \geq 1$.*

Proof. First, let K, L be left ideals. Let $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ be minimal NFAs for K, L. Since K and L are left ideals, we may assume by Proposition 6.1 that A and B have a loop on each symbol in the initial state, and no transition from some other state goes to the initial state. We can get an NFA C for KL from NFAs A and B as follows: For every f in F_A add a loop on every symbol and add transitions (f, a, q) when there is a transition (s_B, a, q) in B, where $f \in F_A, a \in \Sigma, q \in Q_B \setminus \{s_B\}$. Set $F_C = F_B, Q_C = Q_A \cup Q_B \setminus \{s_B\}$. The resulting NFA has $m+n-1$ states, so $\text{nsc}(KL) \leq m+n-1$.

Now, let K, L be right ideals. Let $A = (Q_A, \Sigma, \delta_A, s_A, \{q_f\})$ be a minimal m-state NFA for K and $B = (Q_B, \Sigma, \delta_B, s_B, \{p_f\})$ be a minimal n-state NFA for L. Since K and L are right ideals, we may assume by Proposition 6.1 that A and B have a loop on each symbol in the unique final state, and no transition goes from the final state to some other state. We can get an NFA C for KL from NFAs A and B by merging final state of A with initial state of B and excluding of merged state from set of final states as follows: $C = (Q_C, \Sigma, \delta_C, s_A, \{p_f\})$, where $Q_C = (Q_A \setminus \{q_f\}) \cup (Q_B \setminus \{s_B\}) \cup \{n_{AB}\}$ and for every a in Σ we have $\delta_C(n_{AB}, a) = \delta_A(q_f, a) \cup \delta_B(s_B, a)$. The resulting NFA has $m+n-1$ states, so $\text{nsc}(KL) \leq m+n-1$.

Two-sided and all-sided ideals are also right ideals, so upper bound is the same as in that cases. To prove tightness, consider all-sided ideal languages $K = \{a^{m-1}a^k \mid k \geq 0\}$ and $L = \{a^{n-1}a^k \mid k \geq 0\}$, with $\text{nsc}(K) = m$ and $\text{nsc}(L) = n$. The set $\mathcal{F} = \{(a^i, a^{m+n-2-i}) \mid 0 \leq i \leq m+n-2\}$ is fooling set for KL, so $\text{nsc}(KL) \geq |\mathcal{F}| = m+n-1$. □

Let us continue with star and reversal.

Theorem 6.8 **(Star).** *Let $n \geq 2$. Let L be ideal languages over Σ with $nsc(L) = n$. Then $\text{nsc}(L^*) \leq n+1$ and the bound is met by a binary all-sided ideal.*

Proof. The upper bound $n+1$ holds since it holds for regular languages. For tightness, consider the binary all-sided ideal $L = \{w \in \{a,b\}^* \mid \#_a(w) \geq n-1\}$ with $nsc(L) = n$. Let $\mathcal{F} = \{(ba^i, a^{n-1-i}b) \mid 0 \leq i \leq n-1\} \cup \{(\varepsilon, \varepsilon)\}$ be a set of $n+1$ pairs. To prove the theorem, we only need to show that $\mathcal{F}$ is a fooling set for L^*. Since $\varepsilon \cdot \varepsilon \in L^*$ and $ba^i \cdot a^{n-1-i}b \in L^*$, where $0 \leq i \leq n-1$, condition (F1) is satisfied. To prove (F2), consider two cases: (1) Pairs of forms $(ba^i, a^{n-1-i}b)$ and $(ba^j, a^{n-1-j}b)$, where $0 \leq i < j \leq n-1$. Then $ba^i \cdot a^{n-1-j}b = ba^{n-1-(j-i)}b$, which is the string not equal to ε and with small number of a, so $ba^i \cdot a^{n-1-j}b \notin L^*$. (2) Pairs $(ba^i, a^{n-1-i}b)$ and $(\varepsilon, \varepsilon)$, where $0 \leq i \leq n-1$. Then if $i < n-1$, the string $ba^i \cdot \varepsilon \notin L^*$ and if $i = n-1$, the string $\varepsilon \cdot a^{n-1-i}b = b \notin L^*$. Hence $\mathcal{F}$ is fooling set for L^*, so $\text{nsc}(L^*) \geq |\mathcal{F}| = n+1$. □

Theorem 6.9 **(Reversal).** *Let $n \geq 3$. Let L be ideal languages over Σ with $\text{nsc}(L) = n$.*

(a) If L is right or two-sided or all-sided ideal, then $\text{nsc}(L^R) \leq n$ and the bound is tight if $|\Sigma| \geq 1$.

(b) If L is left ideal, then $\text{nsc}(L^R) \leq n+1$ and the bound is tight if $|\Sigma| \geq 3$.

Proof. (a) Let L be a right ideal. We first prove the upper bound. Let A be a minimal n-state NFA for L. We can construct an NFA A^R for L^R by reverse all transition and setting initial state of A to final state and every final state of A to initial state. Since by Proposition 6.1, NFA A has unique final state, the A^R has unique initial state and therefore $\text{nsc}(L^R) \leq n$. Two-sided and all sided ideals are also right sided, so the upper bound is also n. To prove tightness, consider unary language L with $\text{nsc}(L) = n$. Such a language is the same for L^R, so $\text{nsc}(L^R) = n$.

(b) Let L be a left ideal. Let A be a minimal n-state NFA for L. After construction of A^R described in case (a) above we get NNFA with possible more initial states, so after adding new extra initial state $\text{nsc}(L^R) \leq n+1$. To prove tightness, consider a language $L = (a+b+c)^*b(a^{n-2}c)^*(\varepsilon + a + a^2)$ shown in Figure 6.7.

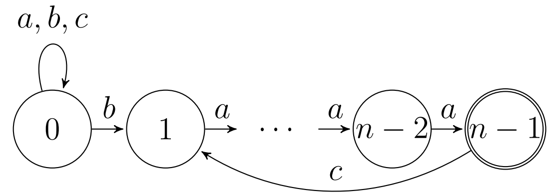

Figure 6.7: The left ideal language L with $\mathrm{nsc}(L^R) = n+1$.

Let $\mathcal{A} = \{(ca^i, a^{n-2-i}b) \mid 0 \leq i \leq n-3\} \cup \{(ca^{n-2}b, \varepsilon)\}$, and $\mathcal{B} = \{(ca^{n-2}, b)\}$. Minimal NFA for L^R needs $n+1$ states, because $\mathcal{A} \cup B$, $A \cup \{(\varepsilon, b)\}$ and $B \cup \{(\varepsilon, ab)\}$ are fooling sets. We show that $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, b)\}$ and $\mathcal{B} \cup \{(\varepsilon, ab)\}$ are fooling sets. Consider three cases: (1) The set $\mathcal{A} \cup \mathcal{B}$. Since reversals of $ca^i \cdot a^{n-2-i}b$ for $0 \leq i \leq n-2$ are in L and also reversal of $ca^{n-2}b \cdot \varepsilon$ is in L, the condition (F1) is satisfied. To prove (F2) we have two cases:

(1.1) Consider two pairs $(ca^i, a^{n-2-i}b)$ and $(ca^j, a^{n-2-j}b)$ where $0 \leq i < j \leq n-2$. Then in the string $ca^i a^{n-2-j}b$ are less a's than $n-2$, so $ca^i \cdot a^{n-2-j}b \notin L^R$.

(1.2) Consider pair $(ca^i, a^{n-2-i}b)$, where $0 \leq i \leq n-2$ and $(ca^{n-2}b, \varepsilon)$. Then $ca^i \cdot \varepsilon$ does not contain any symbol b, so $ca^i \cdot \varepsilon \notin L^R$.

Hence, the set $\mathcal{A} \cup \mathcal{B}$ is fooling set.

(2) The set $\mathcal{A} \cup \{(\varepsilon, b)\}$. The condition (F1) for $\mathcal{A}$ was proved in case (1) and for (ε, b) is also satisfied. To prove (F2) we have two cases:

(2.1) Consider $(ca^i, a^{n-2-i}b)$, where $0 \leq i \leq n-3$ and (ε, b). Then $ca^i \cdot b$ contains less than $n-2$ a's between c and b, so $ca^i \cdot b \notin L^R$.

(2.2) Consider $(ca^{n-2}b, \varepsilon)$ and (ε, b). Then $\varepsilon \cdot \varepsilon \notin L^R$

Hence, the set $\mathcal{A} \cup \{(\varepsilon, b)\}$ is fooling set.

(3) The set $\mathcal{B} \cup \{(\varepsilon, ab)\}$. The reversals $ca^{n-2} \cdot b$ and $\varepsilon \cdot ab$ are in L so condition (F1) is satisfied. Concatenation $ca^{n-2} \cdot ab$ has more than $n-2$ a's between b and c, so $ca^{n-2} \cdot ab \notin L^R$, which prove (F2).

Hence, the set $\mathcal{B} \cup \{(\varepsilon, ab)\}$ is fooling set. By Lemma 2.2 we have $\mathrm{nsc}(L^R) \geq |\mathcal{A}| + |\mathcal{B}| + 1 = n+1$. $\square$

As the last operation, we study complementation on ideal languages.

Theorem 6.10 (Complementation). *Let $n \geq 3$. Let L be a language over Σ with $nsc(L) = n$.*

(a) If L is a right or left ideal, then $\mathrm{nsc}(L^c) \leq 2^{n-1}$. The bound is tight if $|\Sigma| \geq 2$.

(b) If L is a two-sided ideal, then $\mathrm{nsc}(L^c) \leq 2^{n-2}$. The bound is tight if $|\Sigma| \geq 2$.

(c) If L is an all-sided ideal, then $\mathrm{nsc}(L^c) \le 2^{n-2}$. *The bound is tight if* $|\Sigma| \ge 2^{n-2}$.

Proof. (a) First, let us consider right ideal languages.

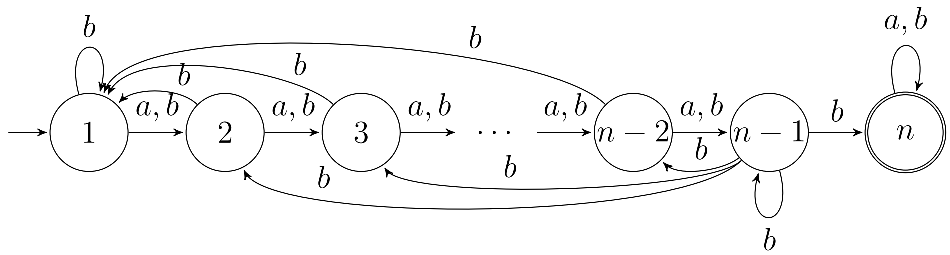

Figure 6.8: An NFA of a binary right ideal language L with $\mathrm{nsc}(L^c) = 2^{n-1}$

Let $A = (Q, \Sigma, \delta, s, F)$ be a minimal n-state NFA for a right ideal L. Then by Proposition 6.1 the NFA A has a unique final state f which goes to itself on every input symbol, that is, we have $\delta(f, a) = \{f\}$ for each a in Σ. It follows that in the subset automaton of the NFA A, all final states are equivalent since they accept all the strings in Σ^*. Hence the subset automaton has at most $2^{n-1} + 1$ reachable and pairwice distinguishable states. By interchanging the final and non-final states, we get a DFA B for L^c. The DFA B has a dead state. After removing the dead state, we get an NFA N for L^c of at most 2^{n-1} states.

To prove tightness, let $L = G \cdot b \cdot (a + b)^*$, where G is the language accepted by the binary $(n-1)$-state NFA N shown in Figure 3.1. Then L is accepted by the n-state NFA N shown in Figure 6.8 and by Proposition 6.1 it is a right ideal. The NFA N is minimal because $\mathcal{F} = \{(a^i, a^{n-2-i}b) \mid 0 \le i \le n-2\} \cup \{(a^{n-2}b, \varepsilon)\}$ is a fooling set for L.

Let $\mathcal{F} = \{(u_S, v_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ be a fooling set for G^c as described in [30, Theorem 5]. We prove that the set $\mathcal{F}' = \{(u_S, v_S \cdot b) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ is a fooling set for L^c.

(F1) For each S, the string $u_S v_S$ is in G^c, so it is not accepted by N. It follows that the string $u_S v_S b$ is not accepted by A. Thus $u_S v_S b$ is in L^c.

(F2) Let $S \ne T$. Then $u_S v_T \notin G^c$ or $u_T v_S \notin G^c$. In the former case, the string $u_S v_T$ is accepted by the NFA N, and therefore the string $u_S v_T b$ is accepted by A. Hence $u_S v_T b \notin L^c$. The latter case is symmetric.

Hence $\mathcal{F}'$ is a fooling set for L^c, which means that $\mathrm{nsc}(L) = 2^{n-1}$.

Second, let us consider left ideal languages.

Let $A = (Q, \Sigma, \delta, s, F)$ be a minimal n-state NFA for a left ideal L. By Proposition 6.1 we can add a loop in the initial state s on every input symbol, we get an NFA N which is

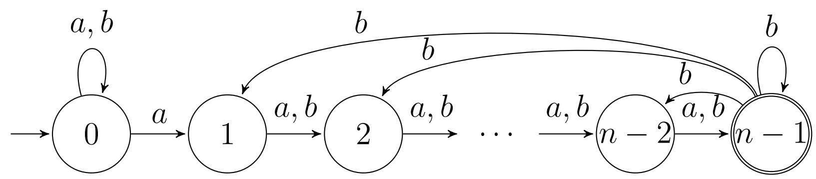

Figure 6.9: An NFA of a binary left ideal language L with $\text{nsc}(L^c) = 2^{n-1}$

equivalent to A. Since the initial state s of N goes to itself on every input symbol, each reachable subset of the subset automaton of N contains the initial state s, so the number of all reachable subsets is at most 2^{n-1}.

To prove tightness, let the language L be accepted by NFA A in Figure 6.9. Then L is by Proposition 6.1 binary left ideal. The NFA A is minimal because $\mathcal{F} = \{(a^i, a^{n-1-i}) \mid 0 \leq i \leq n-1\}$ is fooling set for L.

We are going to consider L^c. Let us consider set of states $\{1, 2, \ldots, n-1\}$ in NFA A. Our aim is to find two strings u_S and v_S for every subset S of $\{1, 2, \ldots, n-1\}$ such that $\mathcal{F} = \{(u_S, v_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ would be a fooling set for L^c. Such strings are described in Preliminaries in Theorem 3.1, for a little different automaton but description is the same unless the size of set of states. Summarize the property of strings u_S, v_S:
(1) string u_S is such that the state 1 goes to the set S after reading u_S
(2) if $p \in S$, then the string v_S is rejected by the NFA A from the state p.
(3) if $p \notin S$, then string v_S is accepted by the NFA A form the state p.
The proof is almost the same as in [30, Theorem 5] and we omit it.

Now, we prove that the set $\mathcal{F}' = \{(a \cdot u_S, v_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ is a fooling set for L^c.

(F1) For each S, the string $u_S v_S$ is not accepted from state 1, so it follows that the string $au_S v_S$ is not accepted by A. Thus $au_S v_S$ is in L^c.

(F2) Let $S \neq T$. Then $u_S v_T \notin L^c$ or $u_T v_S \notin L^c$. Let $u_S v_T$ be accepted by the NFA A, and therefore the string $au_S v_T$ is accepted by A. Hence $au_S v_T \notin L^c$. The latter case is symmetric.

Hence $\mathcal{F}'$ is a fooling set for L^c, which means that $\text{nsc}(L) = 2^{n-1}$.

(b)

Let $A = (Q, \Sigma, \delta, s, F)$ be a minimal n-state NFA for a two-sided ideal L. Then by Proposition 6.2 A has a unique final state f which goes to itself on every input symbols. We can also by Proposition 6.2 add a loop in initial state s for every input symbol. That is, we have $\delta(f, a) = \{f\}$ for each a in Σ and $\delta(s, a) = \{s\}$ for each a in Σ. It follows that in the subset automaton of the NFA A are at most 2^{n-1} reachable subsets, but every

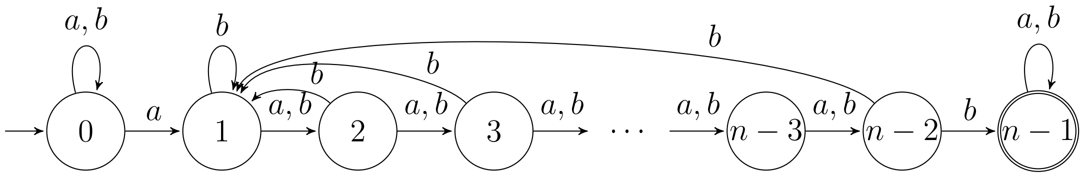

Figure 6.10: An NFA of a binary two-sided ideal language L with $\text{nsc}(L^c) = 2^{n-2}$

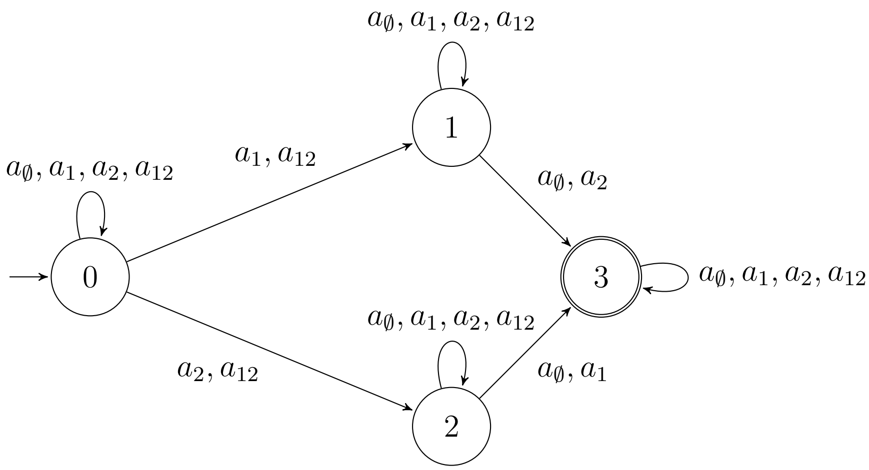

Figure 6.11: Example of a all-sided ideal language with NFA of four states and alphabet $\Sigma = \{a_\emptyset, a_1, a_2, a_{12}\}$

subset containing final state f are equivalent, hence the subset automaton has at most $2^{n-2} + 1$ reachable and pairwise distinguishable states. By interchanging the final and non-final states, we get a DFA B for L^c. The DFA B has a dead state. After removing the dead state, we get an NFA N for L^c of at most 2^{n-2} states.

To prove tightness, let the language L be accepted by NFA A in Figure 6.10. Then L is by Proposition 6.2 binary two-sided ideal. The NFA A is minimal because $\mathcal{F} = \{(a^i, a^{n-2-i}b) \mid 0 \leq i \leq n-2\} \cup \{(a^{n-2}b, \varepsilon)\}$ is fooling set for L. Let us define set of pairs as $\mathcal{F}' = \{(a \cdot u_S, v_S \cdot b) \mid S \subseteq \{1, 2, \ldots, n-2\}\}$, where strings u_S, v_S are define the same way as in Preliminaries in Theorem 3.1. The set $\mathcal{F}'$ is fooling set for L^c with 2^{n-2} elements, thus $\text{nsc}(L^c) \geq 2^{n-2}$.

(c)

The upper bound is the same as for two-sided ideals. To prove tightness, let $\Sigma = \{a_S \mid S \subseteq \{1, 2, \ldots, n-2\}\}$ be an alphabet with 2^{n-2} symbols. Consider the language L accepted by the NFA $A = (\{0, 1, \ldots, n-1\}, \Sigma, \delta, 0, \{n-1\})$ where for each symbol a_S, we have

$\delta(0, a_S) = \{0\} \cup S$;
$\delta(i, a_S) = \{i\}$ if $i \in S$;
$\delta(i, a_S) = \{i, n-1\}$ if $i \in \{1, 2, \dots, n-2\} \setminus S$;
$\delta(n-1, a_S) = \{n-1\}$.

At the Figure 6.11 is shown NFA with $n = 4$.

Since in each state of A, we have a loop on every input symbol, the language L is an all-sided ideal by Proposition 6.3.

Let $\mathcal{F} = \{(a_S, a_S) \mid S \subseteq \{1, 2, \dots, n-2\}\}$. Let us show that $\mathcal{F}$ is a fooling set for L^c.

(F1) For each S, the NFA A reaches the set $\{0\} \cup S$ by a_S. By the next a_S, the NFA A remains in the set $\{0\} \cup S$, and rejects. Thus $a_S a_S \in L^c$.

(F2) Let S and T be two subsets of $\{1, 2, \dots, n-2\}$ with $S \neq T$. Without loss of generality, there is a state i with $i \in S$ and $i \notin T$. By a_S, the NFA A goes to $\{0\} \cup S$. Since $i \in S$, the NFA A goes to i by a_S. Then it goes to the state $n-1$ by a_T since $i \notin T$. Hence A accepts $a_S a_T$, and therefore $a_S a_T \notin L^c$.

Thus $\mathcal{F}$ is a fooling set for L^c. It follows that $\mathrm{nsc}(L^c) \geq 2^{n-2}$. □

6.4 Concluding remarks and open problems

We investigated the nondeterministic state complexity of basic regular operations on the classes of ideal languages. For each class and for each operation, we obtained the tight upper bounds. These bounds are the same as in the general case of regular languages for intersection and star on all four classes, and reversal on left ideals, while in the remaining cases the complexity is always smaller than for regular languages.

To prove tightness we usually used a binary alphabet which is always optimal. In all the cases where we used a larger alphabet for describing witness languages, It remains open whether the obtained upper bounds can be met also by languages defined over smaller alphabets. We also considered the unary case. Our results are summarized in the following tables.

Class	$K \cap L$	$\lvert\Sigma\rvert$	$K \cup L$	$\lvert\Sigma\rvert$	$K \cdot L$	$\lvert\Sigma\rvert$
Right ideal	mn	2	$m+n$	2	$m+n-1$	1
Left ideal	mn	2	$m+n-1$	2	$m+n-1$	1
Two-sided ideal	mn	2	$m+n-2$	2	$m+n-1$	1
All-sided ideal	mn	2	$m+n-2$	2	$m+n-1$	1
Unary ideal	$\max(m,n)$		$\min(m,n)$		$m+n-1$	
Regular	mn	2	$m+n+1$	2	$m+n$	2
Unary regular	mn; $\gcd(m,n)=1$		$m+n+1$; $\gcd(m,n)=1$		$\geq m+n-1$ $\leq m+n$	

Table 6.1: The nondeterministic complexity of intersection, union, and concatenation on ideal languages. The results for regular languages are from [24].

Class	L^*	$\lvert\Sigma\rvert$	L^R	$\lvert\Sigma\rvert$	L^c	$\lvert\Sigma\rvert$
Right ideal	$n+1$	2	n	1	2^{n-1}	2
Left ideal	$n+1$	2	$n+1$	3	2^{n-1}	2
Two-sided ideal	$n+1$	2	n	1	2^{n-2}	2
All-sided ideal	$n+1$	2	n	1	2^{n-2}	2^{n-2}
Unary ideal	$n-1$		n		$n-1$	
Regular	$n+1$	1	$n+1$	2	2^n	2
Unary regular	$n+1$		n		$2^{\Theta(\sqrt{n \log n})}$	

Table 6.2: The nondeterministic complexity of star, reversal, and complementation on ideal languages. The results for regular languages are from [24].

Chapter 7

Convex Languages

In this chapter we study the nondeterministic complexity of basic operations on convex-languages. Recall that a language L is prefix-convex if $u, w \in L$ and u is a prefix of w imply that each string v such that u is a prefix of v and v is a prefix of w is in L. Suffix-, factor-, and subword-convex languages are defined analogously. Except for complementation on factor- and subword-convex languages, we always obtain tight upper bounds.

7.1 Properties of convex languages

Our first proposition provides a sufficient condition on a DFA to accept a prefix-convex language.

Proposition 7.1. *Let $D = (Q, \Sigma, \cdot, s, F)$ be a* DFA. *If for each final state q and each symbol a in Σ, the state $q \cdot a$ is final or dead, then $L(D)$ is prefix-convex.*

Proof. Let u and w be strings in $L(D)$ such that u is a prefix of w, that is, $w = uv$ for a string v. In the accepting computation on uv, the state reached after reading u is final. It follows that all the following states in this computation must be final because otherwise w would be rejected. Hence $L(D)$ is prefix-convex. $\square$

7.2 Unary convex languages

In this chapter we examine unary convex languages and nondeterministic state complexity of operations on them. Notice that if $i \leq j$, then a^i is a prefix, suffix, factor, and subword of a^j. It follows that in the unary case all convex classes coincide.

Let L be a unary convex language and k be the length of the shortest string in L. If L is infinite, then $L = \{a^i \mid i \geq k\}$. If L is finite and ℓ is the length of the longest string in L, then $L = \{a^i \mid k \leq i \leq \ell\}$. In the first case the set $\{(a^i, a^{k-i}) \mid 0 \leq i \leq k\}$ is a fooling set for L. In the second case the set $\{(a^i, a^{\ell-i}) \mid 0 \leq i \leq \ell\}$ is a fooling set for L. It follows that the minimal incomplete DFA for L, which has $k+1$ states if L is infinite, and $\ell + 1$ states if L is finite, is a minimal NFA for L.

The next theorem provides the tight upper bounds for unary convex languages. All the results, except for the intersection, hold true for free languages too; notice that witness languages for all operations, except for intersection, are free.

Theorem 7.2 (Operations on unary convex languages). *Let $m, n \geq 2$. Let K and L be unary convex languages with $\mathrm{nsc}(K) = m$ and $\mathrm{nsc}(L) = n$. Then*

(1) $\mathrm{nsc}(K \cap L), \mathrm{nsc}(K \cup L) \leq \max\{m, n\}$,

(2) $\mathrm{nsc}(KL) \leq m + n - 1$,

(3) $\mathrm{nsc}(L^*) \leq n - 1$, $\mathrm{nsc}(L^R) \leq n$, *and* $\mathrm{nsc}(L^c) \leq n + 1$,

and all these upper bounds are tight.

Proof. The upper bound for intersection and union can be verified by the case analysis, where K and L can be final or non-final. The upper bounds for concatenation and complement follow from the fact that the minimal NFAs can be incomplete deterministic. The upper bound for reversal follows from the fact that $L^R = L$.

Now we prove an upper bound for star. Let L be a unary convex language with $\mathrm{nsc}(L) = n$. If L is infinite, then $L = a^{n-1}a^*$, and the language L^* is accepted by the $(n-1)$-state NFA $N = (\{0, 1, \ldots, n-2\}, \{a\}, \cdot, 0, \{0\})$ where $i \cdot a = \{i+1\}$ if $i < n-2$ and $i \cdot a = \{0, n-2\}$ if $i = n-2$.

If L is finite, then there is an integer k such that $L = \{a^i \mid k \leq i \leq n-1\}$. Then the $(n-1)$-state NFA for the language L^* can be constructed from a minimal incomplete DFA $(\{0, 1, \ldots, n-1\}, \{a\}, \cdot, 0, \{k, k+1, \ldots, n-1\})$ for L by making the state $n-1$ initial, adding the transition $(n-1, a, 1)$, and removing the state 0.

The languages $a^{m-1}a^*$ and $a^{n-1}a^*$ meet the upper bound for intersection, the languages a^{m-1} and a^{n-1} meet the upper bound for union and concatenation, the language a^{n-1} meets the upper bound for square, star, reversal, and the language $\{a^i \mid 0 \leq i \leq n-1\}$ meets the upper bound for complementation. $\square$

7.3 Operations on convex languages

We start with the operations of intersection and union.

Theorem 7.3 (Intersection). *The nondeterministic state complexity of intersection on all the four classes of convex languages is mn. The upper bound is met by binary subword-convex languages, and it cannot be met in the unary case.*

Proof. The upper bound is the same as for regular languages. Binary subword-closed, so also subword-convex, languages meeting the bound mn for their intersection are given in Theorem 5.6. By Theorem 7.2, the complexity of intersection in the unary case is $\max\{m, n\}$, so the bound mn cannot be met in the unary case. □

Theorem 7.4 (Union). *The nondeterministic state complexity of union on all the four classes of convex languages is $m+n+1$. The upper bound is met by binary subword-convex languages, and it cannot be met in the unary case.*

Proof. The upper bound is the same as for regular languages. Binary subword-closed, so also subword-convex, witnesses are described in Theorem 5.5. In the unary case, the complexity of union is $\max\{m, n\}$ by Theorem 7.2. □

Let us continue with concatenation, star, and reversal.

Theorem 7.5 (Concatenation). *The nondeterministic state complexities of concatenation on each of the four classes of convex languages is $m + n$. The upper bound is met by ternary subword convex languages.*

Proof. The upper bound is the same as for regular languages. Ternary subword-closed, so also subword-convex, languages meeting the bound $m+n$ for concatenation are described in Theorem 5.7. □

Theorem 7.6 (Star). *The nondeterministic state complexity of star on all the four convex classes is $n + 1$. The upper bound is met by a binary subword-convex language, and it cannot be met in the unary case.*

Proof. The upper bound is the same as for regular languages. The binary all-sided ideal, so subword-convex language, meeting the upper bound is described in Theorem 5.8. As shown in Theorem 7.2, case (3), the upper bound cannot be met by any unary convex language. □

Theorem 7.7 (Reversal). *The nondeterministic state complexity of reversal on all the four classes of convex languages is* $n+1$*. All the witnesses are binary, except for subword-convex languages, where the witness is defined over an alphabet of size* $2n-2$*. The upper bound cannot be met by any unary convex language.*

Proof. The upper bound is the same as for regular languages. The subword-closed, so also subword-convex, witness defined over an alphabet of size $2n-2$ is described in Theorem 5.9. Binary factor-closed, so also factor-convex, language given by Theorem 5.9, proves the tightness for the remaining convex classes. The binary alphabet is optimal since $L = L^R$ for every unary language L. □

Now we turn our attention to the complementation operation. To get an automaton for the complement of a language L represented by an n-state NFA, we first apply the subset construction to this NFA. Then, we interchange the final and non-final states. This gives an upper bound 2^n. The binary witness is provided in [30, Theorem 5], and binary prefix-closed language meeting the bound 2^n is described in Theorem 5.10. The same theorem provides tight upper bound $2^{n-1}+1$ for complement on suffix-, factor-, and subword-closed languages, with a binary suffix- and factor-closed witness and a subword-closed witness defined over an alphabet of size 2^n.

The aim of the next part is to describe a suffix-convex language meeting the upper bound 2^n for complementation. Notice that it must be so called proper suffix-convex language, that is, a suffix-convex language which is neither suffix-free nor suffix-closed nor left-ideal, since as mentioned above, the nondeterministic complexity of complementation on suffix-closed and suffix-free languages is less than 2^n; cf. Theorems 5.10, 4.22, and 4.37, and the same is true for left ideal languages; cf. Theorem 6.10.

Lemma 7.8. *Let* $n \geq 3$*. There exists a suffix-convex language* L *over a* 5*-letter alphabet such that* $\operatorname{nsc}(L) = n$ *and* $\operatorname{nsc}(L^c) = 2^n$*.*

Proof. Let L be the language accepted by the nondeterministic finite automaton $A = (\{0,1,\ldots,n-1\}, \{a,b,c,d,e\}, 0, \cdot, \{1,2,\ldots,n-1\})$, where the transitions on a and b are shown in Figure 7.1, the transitions on c, d, e are as follows:

$0 \cdot c = \{0,1,\ldots,n-1\}$,

$0 \cdot d = \{1,2,\ldots,n-1\}$,

$q \cdot e = \{n-1\}$ for each state q of A,

and all the remaining transitions go to the empty set. In the NFA A^R, the final state 0 goes to itself on a, b, c and to the empty set on d and e. Next, every other state of A^R goes to 0 on d, and the state $n-1$ goes to $\{0,1,\ldots,n-1\}$ on e.

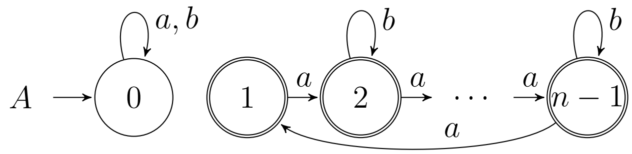

Figure 7.1: Transitions on a and b in suffix-convex witness for complementation.

Thus in the subset automaton of A^R, each final subset, that is, a subset containing the state 0, goes either to a final subset containing 0 or to the empty set on each input symbol. By Proposition 7.1, L^R is prefix-convex, so L is suffix-convex.

Let us show that each subset of the state set of A is reachable and co-reachable. Notice that

$\{0\} \cdot a = 0, \{0\} \cdot b = \{0\}$,

$0 \cdot c = \{0, 1, \dots, n-1\}$, and

$0 \cdot d = \{1, 2, \dots, n-1\}$.

Moreover, we can shift each subset of $\{1, 2, \dots, n-1\}$ cyclically by one using the symbol a. Next, we can eliminate the state 1 from each subset containing 1 by b. It follows that each subset is reachable.

To prove co-reachability, notice that the initial subset of A^R is $\{1, 2, \dots, n-1\}$ and it goes to $\{0, 1, \dots, n-1\}$ on e. We again use symbol a to shift subsets of $\{1, 2, \dots, n-1\}$ and symbol b to eliminate the state 1. It follows that every subset is co-reachable. By Proposition 2.7, we have $\text{nsc}(L^c) = 2^n$. $\square$

The next theorem summarizes our results on the nondeterministic complexity of complementation on the classes of convex languages.

Theorem 7.9 (Complementation). *The nondeterministic state complexity of complementation is 2^n on prefix- and suffix-convex languages. The prefix-convex witness is binary, and the suffix-convex witness is defined over a 5-letter alphabet. On factor- and subword-convex languages, the complexity of complementation is at least $2^{n-1} + 1$ and at most 2^n.*

Proof. Binary prefix-closed witness is given in 5.10. Lemma 7.8 provides a suffix-convex witness over a 5-letter alphabet. The lower bound for factor- and subword-convex languages follows from 5.10 as well. $\square$

7.4 Concluding remarks and open problems

Tables 7.1 and 7.2 summarizes our results on convex languages. In the second table, the $\cdot$ means that the complexity is the same as in the previous column. This table also displays the sizes of alphabet used for describing wittnes languages. Whenever the alphabet is binary or unary, it is always optimal, otherwise we do not know whether the upper bounds are tight also for smaller alphabets. The exact complexity of complementation in the classes of factor-convex and subword-convex languages remains open.

	$K \cap L$	$K \cup L$	KL	L^*	L^c
Unary convex	$\max\{m, n\}$	$\max\{m, n\}$	$m + n - 1$	$n - 1$	$n + 1$
Unary regular [24]	mn; $\gcd(m, n) = 1$	$m + n + 1$; $\gcd(m, n) = 1$	$\geq m + n - 1$ $\leq m + n$	$n + 1$	$2^{\Theta(\sqrt{n \log n})}$

Table 7.1: Nondeterministic complexity of operations on unary convex classes.

	Regular [24, 30]		Prefix-		Suffix-		Factor-		Subword-convex	
$K \cap L$	mn	2	$\cdot$	2	$\cdot$	2	$\cdot$	2	$\cdot$	2
$K \cup L$	$m + n + 1$	2	$\cdot$	2	$\cdot$	2	$\cdot$	2	$\cdot$	2
KL	$m + n$	2	$\cdot$	3	$\cdot$	3	$\cdot$	3	$\cdot$	3
L^*	$n + 1$	1	$\cdot$	2	$\cdot$	2	$\cdot$	2	$\cdot$	2
L^R	$n + 1$	2	$\cdot$	2	$\cdot$	2	$\cdot$	2	$\cdot$	$2n - 2$
L^c	2^n	2	$\cdot$	2	$\cdot$	5	$\geq 2^{n-1} + 1$ $\leq 2^n$	2	$\geq 2^{n-1} + 1$ $\leq 2^n$	2^n

Table 7.2: Nondeterministic complexity of operations on convex classes. The $\cdot$ means that the complexity is the same as in the previous column.

Conclusions

In this thesis, we studied the nondeterministic state complexity of basic unary and binary operations on the subregular classes of free, closed, ideal, and convex languages. After providing basic definitions and notations, we summarized the known results concerning the complexity of basic operations on the above mentioned classes in the deterministic case, and on the class of regular languages in the nondeterministic case in Chapter 3. In the next chapter, we described upper and lower bound methods used throughout this thesis.

In Chapter 4 we examined the operations on the classes of prefix-, suffix-, factor-, and subword-free languages, and we obtained tight upper bounds in each case. The most interesting result of this part of the thesis is obtaining the complexity of complementation for prefix-, suffix-, and factor-free languages. In each of these three classes, we described witness languages over a ternary alphabet, and we were able to show that the upper bounds cannot be met by any binary languages.

In Chapters 5 and 6 we studied closed and ideal languages. For each of these eight subclasses, we again found the exact nondeterministic complexity of each considered operation. Except for three cases, all our witness languages are desribed over a fixed alphabet of size at most three, and moreover binary alphabets are always optimal.

In Chapter 7 we used our previous results to show that the complexity of each operation, except for complementation, in the class of convex languages is the same as in the general case of regular languages. A careful reader might notice that the classes of prefix-free, prefix-closed, and right ideal languages are subclasses of the class of prefix-convex languages; and we have similar inclusions in the other three convex classes. In the case of complementation on suffix-convex languages, we obtained another very interesting result of this thesis. We described a proper suffix-convex language, that is, a suffix-convex language which is neither suffix-free, nor suffix-closed, nor left ideal, meeting the upper bound 2^n for its complementation. We had to find such a special language because the complexity of complementation on the classes of suffix-free, suffix-closed, and left ideal languages is less than 2^n.

Some problems remained open. For complementation on subword-free languages, we defined witnesses over a growing alphabet. It is open whether the upper bound is tight for some fixed alphabet. In the classes of closed and ideal languages, some of our witnesses were described over a ternary alphabet. We do not know whether or not a binary alphabet can be used to describe the corresponding witnesses. The exact nondeterministic state complexity of complementation in the classes of factor-convex and subword-convex languages remains open as well.

Appendix

7.5 The list of my published papers

This part contains the list of publications.

(a) Mlynárčik,P.: *On average complexity of InsertSort.* ITAT 2005, Information Technologies - Applications and Theory, Proceedings, Slovakia, 117-122

(b) Čevorová, K., Jirásková, G., Mlynárčik, P., Palmovský, M., Šebej, J.: *Operations on Automata with All States Final.* Z. Ésik and Z. Fülöp (Eds.): Automata and Formal Languages 2014 (AFL 2014) EPTCS 151, 2014, pp. 201Ű215, doi:10.4204/EPTCS.151.14

(c) Jirásek, J., Jirásková, G., Krausová, M., Mlynárčik, P., Šebej, J.: *Prefix-Free Languages: Right Quotient and Reversal* In: H. Jürgensen et al. (Eds.): DCFS 2014, LNCS 8614, pp. 210-221. Springer International Publishing Switzerland (2014)

(d) Jirásková, G., Mlynárčik, P.: *Complement on Prefix-Free, Suffix-Free, and Non-Returning NFA Languages.* In: H. Jürgensen et al. (Eds.): DCFS 2014, LNCS 8614, pp. 222-233. Springer International Publishing Switzerland (2014)

(e) Mlynárčik,P.: *Complement on Free and Ideal Languages.* In: Shallit, Okhotin (Eds.): DCFS 2015, LNCS 9118, pp. 185-196. Springer International Publishing Switzerland (2015)

(f) Hospodár, M., Jirásková, G., and Mlynárčik: *Nondeterministic Complexity of Operations on Closed and Ideal Languages.* In: Han YS., Salomaa K. (eds) Implementation and Application of Automata. CIAA 2016. LNCS 9705, pp. 125-137. Springer (2016)

(g) Hospodár, M., Jirásková, G., and Mlynárčik: Nondeterministic Complexity of Operations on Free and Convex Languages. Accepted in CIAA 2017.

7.6 The list of given talks

This part contains the list of my talks in significant conferences concerning to the topic of my thesis.

(a) *16th International Workshop on Descriptional Complexity of Formal Systems.*
August 5-8, 2014, Turku, Finland

(b) *DCFS 2015 Descriptional Complexity of Formal Systems*
June 25-27, 2015, Waterloo, Ontario, Canada

(c) *CIAA 2016 21st International Conference on Implementation and Application of Automata*
July 19-22, 2016, Seoul, South Korea

Bibliography

[1] Bach, E., Shallit, J.: 2.7 in Algorithmic Number Theory, Vol. 1: Efficient Algorithms. Cambridge, MA: MIT Press (1996)

[2] Berman, P., Lingas, A.: On the complexity of regular languages in terms of finite automata. Technical Report 304 (Polish Academy of Sciences, 1977)

[3] Birget, J.: Intersection and union of regular languages and state complexity. Inf. Process. Lett. 43(4), 185–190 (1992), `http://dx.doi.org/10.1016/0020-0190(92)90198-5`

[4] Birget, J.: Partial orders on words, minimal elements of regular languages and state complexity. Theoretical Computer Science 119(2), 267–291 (1993), `http://dx.doi.org/10.1016/0304-3975(93)90160-U`

[5] Brzozowski, J., Jirásková, G., Li, B., Smith, J.: Quotient complexity of bifix-, factor-, and subword-free regular languages. Acta Cybernetica 21, 507–527 (2014)

[6] Brzozowski, J.A.: Complexity in convex languages. In: Dediu, A., Fernau, H., Martín-Vide, C. (eds.) Language and Automata Theory and Applications, 4th International Conference, LATA 2010, Trier, Germany, May 24-28, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6031, pp. 1–15. Springer (2010), `http://dx.doi.org/10.1007/978-3-642-13089-2_1`

[7] Brzozowski, J.A., Jirásková, G., Li, B.: Quotient complexity of ideal languages. Theor. Comput. Sci. 470, 36–52 (2013), `http://dx.doi.org/10.1016/j.tcs.2012.10.055`

[8] Brzozowski, J.A., Jirásková, G., Zou, C.: Quotient complexity of closed languages. Theory Comput. Syst. 54(2), 277–292 (2014), `http://dx.doi.org/10.1007/s00224-013-9515-7`

[9] Brzozowski, J.A., Liu, B.: Quotient complexity of star-free languages. Int. J. Found. Comput. Sci. 23(6), 1261–1276 (2012), `http://dx.doi.org/10.1142/S0129054112400515`

[10] Câmpeanu, C., Salomaa, K., Yu, S.: Tight lower bound for the state complexity of shuffle of regular languages. Journal of Automata, Languages and Combinatorics 7(3), 303–310 (2002)

[11] Čevorová, K., Jirásková, G., Mlynárčik, P., Palmovský, M., Sebej, J.: Operations on automata with all states final. In: Ésik, Z., Fülöp, Z. (eds.) Proceedings 14th International Conference on Automata and Formal Languages, AFL 2014, Szeged, Hungary, May 27-29, 2014. EPTCS, vol. 151, pp. 201–215 (2014), http://dx.doi.org/10.4204/EPTCS.151.14

[12] Chrobak, M.: Finite automata and unary languages. Theor. Comput. Sci. 47(3), 149–158 (1986), http://dx.doi.org/10.1016/0304-3975(86)90142-8

[13] Cmorik, R., Jirásková, G.: Basic operations on binary suffix-free languages. In: Kotásek et al. [35], pp. 94–102, http://dx.doi.org/10.1007/978-3-642-25929-6_9

[14] Domaratzki, M.: State complexity of proportional removals. Journal of Automata, Languages and Combinatorics 7(4), 455–468 (2002)

[15] Eom, H., Han, Y., Salomaa, K.: State complexity of k-union and k-intersection for prefix-free regular languages. In: Descriptional Complexity of Formal Systems — 15th International Workshop, DCFS 2013, London, ON, Canada, July 22-25, 2013. Proceedings. pp. 78–89 (2013), http://dx.doi.org/10.1007/978-3-642-39310-5_9

[16] Eom, H., Han, Y., Salomaa, K., Yu, S.: State complexity of combined operations for prefix-free regular languages. In: Paun, G., Rozenberg, G., Salomaa, A. (eds.) Discrete Mathematics and Computer Science. In Memoriam Alexandru Mateescu (1952-2005). pp. 137–151. The Publishing House of the Romanian Academy (2014)

[17] Ésik, Z., Fülöp, Z. (eds.): Automata, Formal Languages, and Related Topics - Dedicated to Ferenc Gécseg on the occasion of his 70th birthday. Institute of Informatics, University of Szeged, Hungary (2009)

[18] Glaister, I., Shallit, J.: A lower bound technique for the size of nondeterministic finite automata. Inf. Process. Lett. 59(2), 75–77 (1996), http://dx.doi.org/10.1016/0020-0190(96)00095-6

[19] Han, Y., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theor. Comput. Sci. 410(27-29), 2537–2548 (2009), http://dx.doi.org/10.1016/j.tcs.2008.12.054

[20] Han, Y., Salomaa, K.: Nondeterministic state complexity for suffix-free regular languages. In: McQuillan, I., Pighizzini, G. (eds.) Proceedings Twelfth Annual Workshop on Descriptional Complexity of Formal Systems, DCFS 2010, Saskatoon, Canada, 8-10th August 2010. EPTCS, vol. 31, pp. 189–196 (2010), http://dx.doi.org/10.4204/EPTCS.31.21

[21] Han, Y., Salomaa, K., Wood, D.: Nondeterministic state complexity of basic operations for prefix-free regular languages. Fundam. Inform. 90(1-2), 93–106 (2009), http://dx.doi.org/10.3233/FI-2009-0008

[22] Han, Y., Salomaa, K., Wood, D.: Operational state complexity of prefix-free regular languages. In: Ésik and Fülöp [17], pp. 99–115

[23] Han, Y., Salomaa, K., Wood, D.: Operational state complexity of prefix-free regular languages. In: Ésik and Fülöp [17], pp. 99–115

[24] Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages 14(6), 1087–1102 (2003), http://dx.doi.org/10.1142/S0129054103002199

[25] Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley (1979)

[26] Hospodár, M., Jirásková, G., Mlynárčik, P.: Nondeterministic complexity of operations on closed and ideal languages. In: Han, Y., Salomaa, K. (eds.) Implementation and Application of Automata - 21st International Conference, CIAA 2016, Seoul, South Korea, July 19-22, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9705, pp. 125–137. Springer (2016), http://dx.doi.org/10.1007/978-3-319-40946-7_11

[27] Hromkovič, J.: Communication Complexity and Parallel Computing. Texts in Theoretical Computer Science. An EATCS Series, Springer (1997), http://dx.doi.org/10.1007/978-3-662-03442-2

[28] Hromkovič, J.: Descriptional complexity of finite automata: Concepts and open problems. Journal of Automata, Languages and Combinatorics 7(4), 519–531 (2002)

[29] Jirásková, G.: Note on minimal automata and uniform communication protocols. In: Martín-Vide, C., Mitrana, V. (eds.) Grammars and Automata for String Processing: From Mathematics and Computer Science to Biology, and Back: Essays in Honour of Gheorghe Paun. Topics in Computer Mathematics, vol. 9, pp. 163–170. Taylor and Francis (2003)

[30] Jirásková, G.: State complexity of some operations on binary regular languages. Theoretical Computer Science 330(2), 287–298 (2005), http://dx.doi.org/10.1016/j.tcs.2004.04.011

[31] Jirásková, G., Masopust, T.: Complexity in union-free regular languages. Int. J. Found. Comput. Sci. 22(7), 1639–1653 (2011), http://dx.doi.org/10.1142/S0129054111008933

[32] Jirásková, G., Mlynárčik, P.: Complement on prefix-free, suffix-free, and non-returning NFA languages. In: Jürgensen, H., Karhumäki, J., Okhotin, A. (eds.) Descriptional Complexity of Formal Systems - 16th International Workshop, DCFS 2014, Turku, Finland, August 5-8, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8614, pp. 222–233. Springer (2014), http://dx.doi.org/10.1007/978-3-319-09704-6_20

[33] Jirásková, G., Okhotin, A.: State complexity of cyclic shift. ITA 42(2), 335–360 (2008), http://dx.doi.org/10.1051/ita:2007038

[34] Jirásková, G., Olejár, P.: State complexity of intersection and union of suffix-free languages and descriptional complexity. In: Bordihn, H., Freund, R., Holzer, M., Kutrib, M., Otto, F. (eds.) Workshop on Non-Classical Models for Automata and Applications - NCMA 2009, Wroclaw, Poland, August 31 - September 1, 2009. Proceedings. books@ocg.at, vol. 256, pp. 151–166. Austrian Computer Society (2009)

[35] Kotásek, Z., Bouda, J., Černá, I., Sekanina, L., Vojnar, T., Antos, D. (eds.): Mathematical and Engineering Methods in Computer Science - 7th International Doctoral Workshop, MEMICS 2011, Lednice, Czech Republic, October 14-16, 2011, Revised Selected Papers, Lecture Notes in Computer Science, vol. 7119. Springer (2012), http://dx.doi.org/10.1007/978-3-642-25929-6

[36] Krausová, M.: Prefix-free regular languages: Closure properties, difference, and left quotient. In: Kotásek et al. [35], pp. 114–122, http://dx.doi.org/10.1007/978-3-642-25929-6_11

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

[38] Meyer, A.R., Fischer, M.J.: Economy of description by automata, grammars, and formal systems. In: 12th Annual Symposium on Switching and Automata Theory, East Lansing, Michigan, USA, October 13-15, 1971. pp. 188–191. IEEE Computer Society (1971), http://dx.doi.org/10.1109/SWAT.1971.11

[39] Mlynárčik, P.: Complement on free and ideal languages. In: Shallit, J., Okhotin, A. (eds.) Descriptional Complexity of Formal Systems - 17th International Workshop, DCFS 2015, Waterloo, ON, Canada, June 25-27, 2015. Proceedings. Lecture Notes in Computer Science, vol. 9118, pp. 185–196. Springer (2015), http://dx.doi.org/10.1007/978-3-319-19225-3_16

[40] Moore, F.R.: On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. IEEE Trans. Computers 20(10), 1211–1214 (1971), http://doi.ieeecomputersociety.org/10.1109/T-C.1971.223108

[41] Pighizzini, G., Shallit, J.: Unary language operations, state complexity and jacobsthal's function 13(1), 145–159 (2002), http://dx.doi.org/10.1142/S012905410200100X

[42] Rabin, M.O., Scott, D.: Finite automata and their decision problems. IBM J. Res. Dev. 3(2), 114–125 (Apr 1959), http://dx.doi.org/10.1147/rd.32.0114

[43] Sakoda, W.J., Sipser, M.: Nondeterminism and the size of two way finite automata. In: Lipton, R.J., Burkhard, W.A., Savitch, W.J., Friedman, E.P., Aho, A.V. (eds.) Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA. pp. 275–286. ACM (1978), http://doi.acm.org/10.1145/800133.804357

[44] Sipser, M.: Introduction to the theory of computation. PWS Publishing Company (1997)

[45] To, A.W.: Unary finite automata vs. arithmetic progressions. Inf. Process. Lett. 109(17), 1010–1014 (2009), http://dx.doi.org/10.1016/j.ipl.2009.06.005

[46] Yershov, Y.L.: On a conjecture of V. A. Uspenskii. Algebra i logika (seminar) 1, 45–48 (1962 (in Russian))

[47] Yu, S.: Regular languages. Department of Computer Science, University of Western Ontario, London, Ontario, Canada

[48] Yu, S.: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Word, Language, Grammar, Handbook of Formal Languages, vol. 1, pp. 41–110. Springer-Verlag (1997)

[49] Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoretical Computer Science 125(2), 315–328 (1994), http://dx.doi.org/10.1016/0304-3975(92)00011-F