Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Nondeterministic complexity in subclasses of convex languages ☆

Michal Hospodár *, Galina Jirásková, Peter Mlynárčik

Mathematical Institute, Slovak Academy of Sciences, Grešákova 6, 040 01 Košice, Slovakia

ARTICLE INFO

Article history:
Received 1 January 2018
Received in revised form 21 November 2018
Accepted 27 December 2018
Available online xxxx

Keywords:
Nondeterministic finite automata
Descriptional complexity
Intersection
Union
Concatenation
Kleene star
Reversal
Complementation
Convex languages
Free languages
Ideal languages
Closed languages
Operational state complexity

ABSTRACT

We study the nondeterministic state complexity of basic regular operations on the classes of prefix-, suffix-, factor-, and subword-free, -closed, and -convex regular languages and on the classes of right, left, two-sided, and all-sided ideal regular languages. For the operations of concatenation, intersection, union, reversal, star, and complementation, we get tight upper bounds for all considered classes except for complementation on factor- and subword-convex languages. Most of our witnesses are described over optimal alphabets. The description of a proper suffix-convex language over a five-letter alphabet meeting the upper bound 2^n for complementation, and obtaining an asymptotically tight bound $\Theta(\sqrt{n})$ for complementation of unary prefix-free languages are among the most interesting results of this paper.

1. Introduction

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 [1–3]. 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 [4–6], or grammars [7,8].

Rabin and Scott in 1959 [9] 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 [10] then showed that this construction is optimal. Maslov [4] investigated the state complexity of union, concatenation, and star, and also some other operations.

☆ This work was conducted while Peter Mlynárčik and Michal Hospodár studied on FMFI UK in Bratislava. Some parts of it were presented at the conferences DCFS 2014, DCFS 2015, CIAA 2016, and CIAA 2017, and extended abstracts appeared in the conference proceedings. The research was supported by grant VEGA 2/0084/15 and grant APVV-15-0091.

* Corresponding author.
E-mail addresses: hosmich@gmail.com (M. Hospodár), jiraskov@saske.sk (G. Jirásková), mlynarcik1972@gmail.com (P. Mlynárčik).

https://doi.org/10.1016/j.tcs.2018.12.027
0304-3975/© 2019 Elsevier B.V. All rights reserved.

The systematic study of the state complexity of operations on regular languages began in the paper by Yu et al. [6]. Some special operations were examined as well: proportional removals by Domaratzki [11], shuffle by Câmpeanu et al. [12], and cyclic shift by Jirásková and Okhotin [13]. Operational state complexity has been also studied in subclasses of regular languages. Prefix- and suffix-free languages were investigated by Han et al. [14,15] and by Jirásková et al. [16,13,17]. Brzozowski et al. examined the classes of ideal languages [18], closed languages [19], bifix-, factor-, and subword-free languages [20], convex languages [21,22], and star-free languages [23]. Union-free languages were considered by Jirásková and Masopust [24]. 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.

The nondeterministic state complexity of operations was investigated by Holzer and Kutrib [5], and some improvements of their results can be found in [25]. Holzer et al. [26] also considered nondeterministic operational state complexity in the class of star-free languages, Jirásková and Masopust [24] examined the class of union-free languages, and Han et al. [14,15] studied the classes of prefix-free and suffix-free languages.

In this paper, we continue this research and study the nondeterministic state complexity of basic regular operations in the subclasses of convex languages. In particular, we consider the operations of concatenation, intersection, union, reversal, star, and complementation, and we examine the nondeterministic state complexity of these operations in the classes of prefix-, suffix-, factor-, and subword-free, -closed, and -convex languages, as well as in the classes of right, left, two-sided, and all-sided ideal languages. Thus we study the same classes as Brzozowski et al. [18–21], but from the point of view of *nondeterministic* operational state complexity.

Our results are as follows. The upper bound $m+n$ for concatenation is tight in all four classes of closed (so also convex) languages, while in the remaining cases, the complexity of concatenation is $m+n-1$. The upper bound mn for intersection is tight in all four classes of ideal and closed (so also convex) languages, and its complexity is $mn-m-n+2$ on prefix- and suffix-free languages, and $mn-2m-2n+6$ on factor- and subword-free languages. The upper bound $m+n+1$ for union is tight in all four classes of closed (so also convex) languages, and its complexity is $m+n$ on prefix-free and right-ideal languages, $m+n-1$ on suffix-free and left ideal languages, and $m+n-2$ on the remaining subclasses. The upper bound $n+1$ for reversal is tight in all four classes of closed (so also convex) languages, as well as in the classes of suffix-free and left ideal languages, and in all the remaining classes, the complexity of reversal is n. The upper bound $n+1$ for star is tight in all four classes of ideal (so also convex) languages, its complexity is n on prefix- and suffix-free and -closed languages, and it is 1 on factor- and subword-closed languages. Most of our witnesses are defined over an optimal alphabet, that is, we are able to prove that the corresponding upper bound cannot be met by any language defined over a smaller alphabet.

The most interesting results of this paper are those on the complementation operation. Complementation is an easy operation on regular languages represented by deterministic finite automata since to get a DFA for the complement of a regular language, it is enough to invert the finality of all 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 invert the finality of all states. This gives an upper bound 2^n. Sakoda and Sipser [27] presented an example of languages over a growing alphabet meeting this upper bound. Birget claimed the result for a three-letter alphabet in [28], and later corrected this to a four-letter alphabet. Holzer and Kutrib [5] 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 [25]. In the case of a unary alphabet, the complexity of complementation is in $e^{\Theta(\sqrt{n \ln n})}$ [5,25].

For the nondeterministic state complexity of complementation, Han et al. [14,15] obtained lower bounds 2^{n-1} and $2^{n-1}-1$ for prefix-free and suffix-free languages, respectively, and an upper bound $2^{n-1}+1$ in both classes. The questions of tightness remained open. In this paper, we solve both of these open questions, and we prove that in both classes, the tight upper bound is 2^{n-1}. To prove tightness, we use a ternary alphabet, and we prove that the ternary alphabet is optimal here. Except for factor- and subword-convex languages, we get the exact complexity of complementation in all the remaining classes. We describe binary prefix-closed (so prefix-convex) and a suffix-convex language over a five-letter alphabet which meet the upper bound 2^n. In all the other cases, the complexity of complementation is between 2^{n-2} and $2^{n-1}+1$. This means that to find our witness suffix-convex language, we required it to be neither free, nor closed, nor ideal, that is, we had to define a so-called *proper* suffix-convex language. Another interesting result on the complementation operation is obtaining asymptotically tight bound $\Theta(\sqrt{n})$ for complementation on unary prefix-free languages.

To get lower bounds, we use a lower bound technique described by Birget [29,28]. 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 [30,31], in many other cases, the fooling sets can be used to prove the minimality of nondeterministic finite automata, and we successfully use this method throughout our paper. In the case of union and reversal, where using multiple initial states may save one state, this fooling-set method cannot be used. Instead, its modification given by [24, Lemma 4] is used in this paper. Moreover, we state some sufficient conditions for NFAs to guarantee their minimality. This allows us to avoid tedious description of fooling sets in several cases.

2. Preliminaries

We assume that the reader is familiar with basic notions in formal languages and automata theory. For details and all the unexplained notions, the reader may refer to [32–34].

Please cite this article in press as: M. Hospodár et al., Nondeterministic complexity in subclasses of convex languages, Theoret. Comput. Sci. (2019), https://doi.org/10.1016/j.tcs.2018.12.027

Let Σ be a finite non-empty alphabet of symbols (or letters). 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 $K \cup L = \{w \mid w \in K \text{ or } w \in L\}$. The *concatenation* of languages K and L is the language $KL = \{uv \mid u \in K \text{ and } v \in L\}$. The *(Kleene) star* of a language L is the language $L^* = \bigcup_{i \geq 0} L^i$ where $L^0 = \{\varepsilon\}$ and $L^i = LL^{i-1}$ if $i \geq 1$.

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, the function $\cdot$ 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. 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 $L(A) = \{w \in \Sigma^* \mid s \cdot w \cap F \neq \emptyset\}$.

In an ε-NFA, we also allow the transitions on the empty string. It is known that the ε-transitions can be removed without increasing the number of states in the resulting NFA [32, Theorem 2.2], [34, Theorem 2.3]. 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; cf. [34].

A state of an NNFA is called a *sink state* if it has a loop on each symbol. 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 every string w. An NFA A is a *trim* NFA if each state q of A 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.

We say that (p, a, q) is a transition in NFA A if $q \in p \cdot a$. 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. An NFA is *non-returning* if its initial state does not have any in-transitions, and it is *non-exiting* if each its final state does not have any out-transitions. To *omit* a state in an NFA means to remove it from the set of states and to remove all its in-transitions and out-transitions from the transition function. To *merge* two states means to replace them by a single state with all in-transitions and out-transitions of the original states.

An NFA A 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 q instead of $\{q\}$. If $|q \cdot a| \leq 1$ for each q and a, then A is a *partial* DFA.

Every NNFA $A = (Q, \Sigma, \cdot, I, F)$ can be converted to an equivalent DFA $A' = (2^Q, \Sigma, \cdot, I, \{S \in 2^Q \mid S \cap F \neq \emptyset\})$ by the *subset construction* [9]. The DFA A' is called the *subset automaton* of the NFA A.

The nondeterministic state complexity of a regular language L, $\operatorname{nsc}(L)$, is the smallest number of states in any NFA for L. The nondeterministic state complexity of a unary regular operation f is a function $\operatorname{nsc}_f \colon \mathbb{N} \to \mathbb{N}$ defined as $\operatorname{nsc}_f(n) = \max\{\operatorname{nsc}(f(L)) \mid L \text{ is accepted by an } n\text{-state NFA}\}$. The nondeterministic state complexity of a binary regular operation f is a function $\operatorname{nsc}_f \colon \mathbb{N}^2 \to \mathbb{N}$ defined as $\operatorname{nsc}_f(m, n) = \max\{\operatorname{nsc}(f(K, L)) \mid K \text{ and } L$ are accepted by m-state and n-state NFAs, respectively$\}$. To prove the minimality of NFAs, we use a fooling set lower-bound technique [28,35].

Definition 1. A set of pairs of strings $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is called a *fooling set* for a language L if for all i, j in $\{1, 2, \ldots, n\}$,

(F1) $x_i y_i \in L$, and

(F2) if $i \neq j$, then $x_i y_j \notin L$ or $x_j y_i \notin L$.

Lemma 2 *([28,35]). Let $\mathcal{F}$ be a fooling set for a language L. Then every* NNFA *for the language L has at least $|\mathcal{F}|$ states.* □

The *reverse of a string* w in Σ^* is defined by $\varepsilon^R = \varepsilon$ and $(wa)^R = aw^R$ for each a in Σ and each w in Σ^*. The *reverse of a language* L is the language $L^R = \{w^R \mid w \in L\}$. The *reverse of an* NNFA $A = (Q, \Sigma, \cdot, I, F)$ is the NNFA $A^R = (Q, \Sigma, \cdot^R, F, I)$ where $q \cdot^R a = \{p \in Q \mid q \in p \cdot a\}$; notice that A^R is obtained from A by reversing all the transitions, and by swapping the roles of the initial and final states.

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. Immediately from the definition, we get the following result.

Proposition 3. *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 $Q \setminus T$.* □

We use the following lemma to prove a lower bound on the size of NFAs in some cases.

Lemma 4 *(cf. [36, Lemma 4]). Let A be an* NNFA. *If, for each state q of A, the singleton set $\{q\}$ is reachable and 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 3, 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, the NNFA A is minimal. □

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

Lemma 5 *(cf. [36, Lemma 5]). Let A be a trim* NFA. *If both A and A^R are partial* DFA*s, then A and A^R are minimal* NFA*s.* □

The next observation and its corollary are very useful to get lower bounds for complementation.

Proposition 6. *Let L be a language accepted by an* NFA *in which k subsets of the state set are reachable and each of their complements is co-reachable. Then* $\operatorname{nsc}(L^c) \geq k$.

Proof. Let $A = (Q, \Sigma, \cdot, s, F)$ be an NFA and let the subsets $S_1, S_2, \ldots, S_k$ of Q be reachable, and their complements be co-reachable. Since S_i is reachable, there exists a string u_i in Σ^* such that $s \cdot u_i = S_i$. Since the set $Q \setminus S_i$ is co-reachable, there is a string v_i which is accepted by A from each state in $Q \setminus S_i$, but rejected from each state in S_i. It follows that $\{(u_i, v_i) \mid 1 \leq i \leq k\}$ is a fooling set for L^c of size k. Hence $\operatorname{nsc}(L^c) \geq k$ by Lemma 2. □

Corollary 7. *Let L be a language accepted by an n-state* NFA *in which each subset of the state set is reachable as well as co-reachable. Then* $\operatorname{nsc}(L^c) = 2^n$. □

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. 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. 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, $L = \Sigma^* L$, $L = \Sigma^* L \Sigma^*$, $L = L \sqcup\!\sqcup\, \Sigma^*$, where $L \sqcup\!\sqcup\, \Sigma^*$ is the language obtained from L by inserting any number of symbols to any string in L). Notice that the classes of free, closed, and ideal languages are subclasses of convex languages, and the complement of a prefix- (suffix-, factor-, subword-) closed language is a right (left, two-sided, all-sided, respectively) ideal language.

3. Properties of finite automata accepting subclasses of convex languages

The following properties of NFAs for prefix- and suffix-free languages are known.

Proposition 8 *([14,15]). Let A be a minimal* NFA *for a language L. If L is prefix-free (suffix-free), then A is non-exiting (non-returning).* □

Proposition 9 *(Necessary conditions for suffix-free* DFA*, cf. [37, Proposition 4]). Let $A = (Q, \Sigma, \cdot, s, F)$ 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 q_d;
(3) for each symbol a in Σ, there is a state $q_a \neq q_d$, such that $q_a \cdot a = q_d$;
(4) for each a in Σ, there is no state q in $Q \setminus \{s\}$ such that $s \cdot a = q \cdot a$.

Proof. We only prove case (3); the remaining three proofs are straightforward. Let $|Q| = n$ and $a \in \Sigma$. We must have $s \cdot a^n = q_d$ because otherwise some string w would be accepted from the state $s \cdot a^n$, so the DFA A would accept strings $a^n w$ and $a^\ell w$ with $\ell < n$, which would be a contradiction with suffix-freeness of $L(A)$. Since $s \neq q_d$, there is a state q_a with $q_a \neq q_d$ such that $q_a \cdot a = q_d$. □

The next lemma provides a sufficient condition for a language to be suffix-free.

Lemma 10 *([16, Lemma 1]). Let A be a non-returning partial* 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.* □

A simple method for constructing prefix-, suffix-, and factor-free languages follows from the following observation.

Proposition 11 *([20, Proposition 1]). Let $L \subseteq \Sigma^*$ and $\# \notin \Sigma$. Then the language $L\#$ is prefix-free, $\#L$ is suffix-free, and $\#L\#$ is factor-free.* □

The next proposition from [38, Proposition 12] shows some features of automata for left and right ideals. We provide a proof here.

Proposition 12. *Let L be a regular language accepted by an n-state* NFA.

(1) If L is a left ideal, then L is accepted by an NFA *of at most n states such that there is a loop on each symbol in its initial state, and no transition goes to the initial state from any other state.*

(2) If L is a right ideal, then L is accepted by an NFA *of at most n states such that there is a unique final sink state from which no transition goes to any other state.*

Proof. (1) Let A be an NFA for L and s be the initial state. Construct A' from A by adding a loop on s for each symbol 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', we can read the string u 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.

(2) Let A be an NFA for L and s be the initial state. Construct A' from A by adding loops on every symbol in every final state and removing every other transition going out of a final 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 for the first time. While reading u, no transition leaving a 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 can be read in a final state, so $w \in L(A')$. If $w \in L(A')$, then we can split w into two strings u, v such that $w = uv$ and there is a computation such that after reading u, a final state occurs for 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 $L(A) = L(A')$. If A' has more final states, we can merge them into one to get a desired NFA. □

The following properties of automata accepting closed languages are also known; cf. [39,22]. The proofs are straightforward.

Proposition 13. *Let L be a language accepted by* NFA *A with the initial state s.*

(1) If all the states of A are final, then L is prefix-closed.

(2) If every string which is accepted from a state q of A is also accepted from s, then L is suffix-closed.

(3) If L is prefix-closed and suffix-closed, then L is factor-closed. □

4. Operations on unary convex languages

In this section we investigate the nondeterministic state complexity of basic regular operations on unary free, ideal, closed, and convex languages. Notice that in the unary case the prefix-, suffix-, factor-, and subword-free languages coincide. The same is true for closed and convex languages. The classes of right, left, two-sided, and all-sided unary ideals coincide as well.

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 $L = \{a^i \mid k \leq i \leq \ell\}$. Let $t = k$ for infinite L and $t = \ell$ for finite L. Then the set $\{(a^i, a^{t-i}) \mid 0 \leq i \leq t\}$ is a fooling set for L. Thus the minimal partial DFA for L, which has $t+1$ states, is a minimal NFA for L.

The next theorem provides tight upper bounds for every considered operation and every considered subclass of unary convex languages. It summarizes the results of [38, Theorems 11 and 19] and [36, Theorem 7]. All proof details are provided here. We use this theorem several times throughout the paper to get the optimality of binary alphabets in our witnesses.

Theorem 14. *Let K and L be unary languages accepted by m-state and n-state* NFA, *respectively. Then*

(1) if K and L are free then $\operatorname{nsc}(K \cup L) \leq \max\{m, n\}$, $\operatorname{nsc}(K \cap L) \leq \min\{m, n\}$, $\operatorname{nsc}(KL) \leq m + n - 1$, $\operatorname{nsc}(L^*) \leq n - 1$, *and* $\operatorname{nsc}(L^c) \in \Theta(\sqrt{n})$*;*

(2) if K and L are ideal then $\operatorname{nsc}(K \cup L) \leq \min\{m, n\}$, $\operatorname{nsc}(K \cap L) \leq \max\{m, n\}$, $\operatorname{nsc}(KL) \leq m + n - 1$, $\operatorname{nsc}(L^*) \leq n - 1$, *and* $\operatorname{nsc}(L^c) \leq n - 1$*;*

(3) if K and L are closed then $\operatorname{nsc}(K \cup L) \leq \max\{m, n\}$, $\operatorname{nsc}(K \cap L) \leq \min\{m, n\}$, $\operatorname{nsc}(KL) \leq m + n - 1$, $\operatorname{nsc}(L^*) \leq 1$, *and* $\operatorname{nsc}(L^c) \leq n + 1$*;*

(4) if K and L are convex then $\operatorname{nsc}(K \cup L) \leq \max\{m, n\}$, $\operatorname{nsc}(K \cap L) \leq \max\{m, n\}$, $\operatorname{nsc}(KL) \leq m + n - 1$, $\operatorname{nsc}(L^*) \leq n - 1$, *and* $\operatorname{nsc}(L^c) \leq n + 1$.

All these upper bounds are tight.

Proof. (1)–(3) For every integer n with $n \geq 1$ there exist the unary free language $\{a^{n-1}\}$, the unary ideal language $\{a^i \mid i \geq n-1\}$, and the unary closed language $\{a^i \mid i \leq n-1\}$, all of them with complexity n. If $n \geq 2$, then they are the only languages with complexity n in those classes. If $n = 1$, then also the language $\emptyset$ is free, ideal, and closed, and $\{a\}^*$ is closed. The upper and lower bounds for complexities of operations, except for intersection and complementation of free languages, follow.

To get the result for intersection on unary free languages, let K and L be unary free languages accepted by m-state and n-state NFAs, respectively. Then $K = \{a^k\}$ with $k \le m-1$ and $L = \{a^\ell\}$ with $\ell \le n-1$. Hence $K \cap L = \emptyset$ or $K \cap L = \{a^{\min\{k,l\}}\}$. This gives the upper bound. For tightness, let $K = L = \{a^{\min\{m,n\}-1}\}$.

To get the result for complementation on unary free languages, let $n \ge 3$ and $L = \{a\}^* \setminus \{a^n\}$. We show that $\sqrt{n/3} \le \operatorname{nsc}(L) \le 6\sqrt{n}$. First consider a lower bound, and let us show that every NFA for L requires at least $\sqrt{n/3}$ states. Assume for a contradiction that there is an NFA N for L with less than $\sqrt{n/3}$ states. Then the tail in the Chrobak normal form [40, Definition 4.2] of N is of size less than $3 \cdot (\sqrt{n/3})^2$ [40,41], thus less than n. Since a^n 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 $m = \lfloor \sqrt{n} \rfloor$, and consider relatively prime numbers m and $m+1$. It is known that the maximal integer that cannot be expressed as $xm + y(m+1)$ for non-negative integers x and y is $(m-1)m - 1 = m^2 - m - 1$ [6]. Let $k = n - (m^2 - m - 1)$. Then $0 < k \le 3\sqrt{n}$. Next, the NFA A shown in Fig. 1 and consisting of a path of length k and two overlapping cycles of lengths m and $m+1$ does not accept a^n and accepts all strings a^i with $i \ge n+1$. It remains to accept the shorter strings. To this aim let $\ell = \lceil \log n \rceil$ and let $p_1, p_2, \ldots, p_\ell$ be the first ℓ primes. Then $p_1 p_2 \cdots p_\ell > n$ and $p_1 + p_2 + \cdots + p_\ell = \Theta(\ell^2 \ln \ell) \le \sqrt{n}$ [42]. Consider an NFA B consisting of an initial state that is connected to ℓ cycles of lengths $p_1, p_2, \ldots, p_\ell$. Let the states in the jth cycle be $0, 1, \ldots, p_j - 1$, where the initial state of B is connected to state 1. The state $n \bmod p_j$ is non-final, and all the other states are final. Then this NFA does not accept a^n, but accepts all strings a^i with $i \le n-1$ since we have $(i \bmod p_1, i \bmod p_2, \ldots, i \bmod p_\ell) \ne (n \bmod p_1, n \bmod p_2, \ldots, n \bmod p_\ell)$. Now we get the resulting NFA for the language L of at most $6\sqrt{n}$ states as the union of NFAs A and B. This gives the result for complementation on unary free languages.

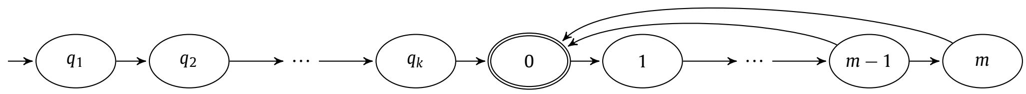

Fig. 1. The part of the NFA accepting every string of length more than n.

(4) The upper bound for intersection and union can be verified by the case analysis, where K and L can be finite or infinite. The upper bounds for concatenation and complementation follow from the fact that the minimal NFAs can be partial DFAs. Now we prove an upper bound for star. Let L be a unary convex language with $\operatorname{nsc}(L) = n$. 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 $L = \{a^i \mid k \le i \le n-1\}$. Then the $(n-1)$-state NFA for L^* can be constructed from a minimal partial 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 star, and the language $\{a^i \mid i \le n-1\}$ meets the upper bound for complementation. $\square$

5. Concatenation

The nondeterministic state complexity of concatenation on regular languages is $m+n$ and the witness language is defined over a binary alphabet [5, Theorem 7]. In the unary case, a lower bound is $m+n-1$. It is not known whether the upper bound $m+n$ can be met by concatenation of unary languages. We prove that the upper bound $m+n$ is met by ternary subword-closed (so subword-convex) languages. We do not know whether or not the ternary alphabet is optimal here. In the classes of free and ideal languages, the complexity of concatenation is $m+n-1$ with unary witnesses. This summarizes our results from the conference papers [38, Theorems 6 and 15] and [36, Theorem 10].

Theorem 15. *Let K and L be convex languages accepted by an m-state* NFA *and an n-state* NFA*, respectively. Then KL is accepted by an* NFA *with at most $m+n$ states, and this bound is met by ternary subword-closed languages.*

Proof. We prove that the upper bound $m+n$ is met by the concatenation of ternary subword-closed languages. Consider the ternary subword-closed languages K and L accepted by NFAs shown in Fig. 2. Consider the set of pairs $\mathcal{F} = \{(a^i, a^{m-1-i}cba^{n-1}) \mid 0 \le i \le m-1\} \cup \{(a^{m-1}cba^j, a^{n-1-j}) \mid 0 \le j \le n-1\}$. We have $a^{m-1}cba^{n-1} \in KL$ and $KL \subseteq b^*a^*c^*b^*a^*c^*$. Next, no string which has more than $m+n-2$ occurrences of a is in KL. Hence the set $\mathcal{F}$ is a fooling set for KL, so $\operatorname{nsc}(KL) \ge m+n$ by Lemma 2. $\square$

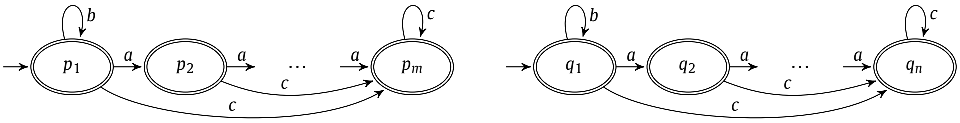

Fig. 2. Ternary subword-closed witnesses for concatenation meeting the upper bound $m+n$.

The concatenation operation on the classes of prefix- and suffix-free languages was investigated by Han et al. [14, Theorem 3.1], [15, Theorem 4] where the upper bound $m+n-1$ was established and a binary prefix-free witness was provided. Since no proof for the suffix-free case was given, we provide this proof here. Moreover, we describe a unary subword-free witness. We also discuss the case of ideal languages in the next theorem.

Theorem 16. *Let K and L be free or ideal languages accepted by an m-state* NFA *and an n-state* NFA, *respectively. Then KL is accepted by an* NFA *with at most $m+n-1$ states, and this bound is met by unary subword-free and unary all-sided ideal languages.*

Proof. In the prefix-free case (cf. [14, Proof of Theorem 3.1]), we may assume that automata A and B are non-exiting and have a unique final state. 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, we may assume that automata A and B are non-returning. To get an NFA for KL from A and B, we add the transition (p,a,q) for each final state p of A and each transition (s_B,a,q) of B. Next, we make final states of A non-final, the initial state of B non-initial, 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 $\{a^{n-1}\}$.

Let K,L be right ideals accepted by NFAs $A=(Q_A,\Sigma,\cdot_A,s_A,\{f_A\})$ and $B=(Q_B,\Sigma,\cdot_B,s_B,\{f_B\})$, respectively; by Proposition 12, we may assume that A and B have a loop on each symbol in the unique final state which has no other out-transitions. Construct an $(m+n-1)$-state NFA for KL from NFAs A and B by merging the final state of A with the initial state of B; the initial state of this NFA is s_A and its unique final state is f_B.

Let K and L be left ideals accepted by NFAs $A=(Q_A,\Sigma,\cdot_A,s_A,F_A)$ and $B=(Q_B,\Sigma,\cdot_B,s_B,F_B)$, respectively. By Proposition 12, we may assume that A and B have a loop on each symbol in the initial state which has no other in-transitions. Construct the NFA $((Q_A\cup Q_B)\setminus\{s_B\},\Sigma,\cdot,s_A,F_B)$ for KL from NFAs A and B as follows: For every state p in F_A, add a loop on every symbol and add the transitions (p,a,q) when there is a transition (s_B,a,q) in B.

To prove tightness, consider all-sided ideal languages $K=\{a^k \mid k\ge m-1\}$ and $L=\{a^k \mid k\ge n-1\}$, accepted by m-state and n-state NFA, respectively. Then the set $\{(a^i,a^{m+n-2-i}) \mid 0\le i\le m+n-2\}$ is a fooling set for KL, so $\operatorname{nsc}(KL)\ge m+n-1$ by Lemma 2. □

6. Intersection

The tight upper bound on the nondeterministic state complexity of intersection is mn and it is met by binary languages [5, Theorem 3], as well as by unary languages if $\gcd(m,n)=1$ [5, Theorem 4]. In this section, we show that the upper bound mn is tight in all subclasses of ideal and closed (so also convex) languages, and it is less than mn in all subclasses of free languages. Except for subword-free witness that is defined over a growing alphabet, all the remaining witnesses are binary. Moreover, the binary alphabet is always optimal. Some results of this section were stated in our conference papers [38] and [36]. Here we provide all proofs which were not given before. Moreover, we discuss the binary subword-free case.

Theorem 17. *Let K and L be convex languages accepted by an m-state* NFA *and an n-state* NFA, *respectively. Then $K\cap L$ is accepted by an* NFA *with at most mn states, and this bound is met by binary subword-closed and binary all-sided ideal languages. The binary alphabet is optimal.*

Proof. The upper bound is the same as in the case of regular languages. For tightness, consider the binary all-sided ideals $K=\{w\in\{a,b\}^* \mid |w|_a\ge m-1\}$ and $L=\{w\in\{a,b\}^* \mid |w|_b\ge n-1\}$ accepted by m-state and n-state DFAs, respectively. Let $\mathcal{F}=\{(a^ib^j,a^{m-1-i}b^{n-1-j}) \mid 0\le i\le m-1 \text{ and } 0\le j\le n-1\}$. Since each string w with $|w|_a=m-1$ and $|w|_b=n-1$ is in $K\cap L$, while no string with $|w|_a<m-1$ or $|w|_b<n-1$ is in $K\cap L$, the set $\mathcal{F}$ is a fooling set for $K\cap L$, so $\operatorname{nsc}(K\cap L)\ge mn$.

Next, consider the binary subword-closed languages $K=\{w\in\{a,b\}^* \mid |w|_a\le m-1\}$ and $L=\{w\in\{a,b\}^* \mid |w|_b\le n-1\}$ accepted by an m-state and n-state partial DFA, respectively. In a similar way as in the case of ideal languages, we can show that the same set $\mathcal{F}$ is a fooling set for $K\cap L$.

By Theorem 14, the binary alphabet is optimal in both cases. □

The nondeterministic complexity of intersection on prefix- and suffix-free languages was studied by Han et al. [14,15], 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 [43]. Here we obtain the tight upper bounds for 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. To show subword-freeness of our witnesses, we use the following proposition.

Proposition 18. *Let $A = (\{0, 1, \dots, n-1\}, \Sigma, \cdot\, 0, \{n-1\})$ be an n-state partial* DFA *such that for all states i and j and each input symbol σ in Σ, we have*

(1) $i \cdot \sigma > i$ whenever $i \cdot \sigma$ is defined;

(2) if $i < j$, then $i \cdot \sigma < j \cdot \sigma$ whenever $i \cdot \sigma$ and $j \cdot \sigma$ are defined.

Then $L(A)$ is subword-free.

Proof. By induction on $|w|$, we prove the properties (1) and (2) for an arbitrary non-empty string w. Since A is a partial DFA, every string w in $L(A)$ has exactly one computation in A. Consider a string $w = v_0u_1v_1u_2 \cdots v_{k-1}u_kv_k$ and its proper subword $v = v_0v_1 \cdots v_k$ with $|v_0| \ge 0$, $|v_k| \ge 0$, and $|u_i|, |v_i| > 0$ if $1 \le i \le k-1$. Assume that both w and v are in L. This means that both w and v have accepting computations in A:

$$0 \xrightarrow{v_0} p_1 \xrightarrow{u_1} q_1 \xrightarrow{v_1} p_2 \xrightarrow{u_2} q_2 \xrightarrow{v_2} \cdots \xrightarrow{u_k} q_k \xrightarrow{v_k} p_{k+1} = n-1 \text{ and } 0 \xrightarrow{v_0} p_1 \xrightarrow{v_1} p_2' \xrightarrow{v_2} \cdots \xrightarrow{v_k} p_{k+1}' = n-1.$$

Since u_1 is non-empty, $p_1 < q_1$, so $p_2' = p_1 \cdot v_1 < q_1 \cdot v_1 = p_2$, and by induction, $p_{k+1}' < p_{k+1} = n-1$, a contradiction. Hence $L(A)$ is subword-free. □

Theorem 19. *Let $m \ge 5$ and $n \ge 3$. Let K and L be languages accepted by an m-state* NFA *and an n-state* NFA*, respectively.*

(1) If K and L are prefix- or suffix-free, then $K \cap L$ is accepted by an NFA *with at most $mn - (m+n-2)$ states, and this bound is met by binary languages in both cases.*

(2) If K and L are factor- or subword-free, then the language $K \cap L$ is accepted by an NFA *with at most $mn - 2(m+n-3)$ states, and this bound is met by binary factor-free languages and subword-free languages over an alphabet of size $m+n-5$ if $m \ge 5$ and $n \ge 3$.*

The binary alphabet is optimal in the cases of prefix-, suffix-, and factor-free languages.

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 $\{0, 1, \dots, m-1\}$ and $\{0, 1, \dots, n-1\}$, 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 $(m-1, j)$ with $0 \le j \le n-2$ and all states $(i, n-1)$ with $0 \le i \le m-2$ 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.

We first describe binary factor-free witnesses. Then, prefix- and suffix-free witnesses will be obtained from them. Let K and L be the languages accepted by the NFAs A and B shown in Fig. 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 starts or ends in 2. However, in such a case, $|v|_b \bmod (m-2) \ne (m-3)$, so $v \notin L$. Hence the language K is factor-free. Next, every string in L has exactly $n-1$ occurrences of a, but every proper factor of every string in L has less than $n-1$ occurrences of a. Hence L is factor-free. Construct the product automaton $A \times B$ and omit all the unreachable and dead states to get a trim NFA N for $K \cap L$. Since the NFA N and its reverse N^R are partial DFAs, the NFA N is minimal by Lemma 5. So we have $\text{nsc}(K \cap L) = mn - 2(m+n-3)$.

Next, let K and L be the languages accepted by $(n+1)$-state and $(m+1)$-state NFA from Fig. 3. Then the left quotients of K and L by the string a, that is, the languages $a^{-1}K$ and $a^{-1}L$, are prefix-free and meet the upper bound $mn-(m+n-2)$. Similarly, the right quotients Ka^{-1} and La^{-1} are suffix-free witnesses meeting the same upper bound.

To describe a subword-free witness, let $\Sigma = \{a\} \cup \{b_k \mid 2 \le k \le m-2\} \cup \{c_\ell \mid 2 \le \ell \le n-2\}$. Let K and L be languages accepted by partial DFAs $A = (\{0, 1, \dots, m-1\}, \Sigma, \cdot, 0, \{m-1\})$ and $B = (\{0, 1, \dots, n-1\}, \Sigma, \circ, 0, \{n-1\})$, where for each i $(0 \le i \le m-2)$, j $(0 \le j \le n-2)$, k $(2 \le k \le m-2)$, and ℓ $(2 \le \ell \le n-2)$, 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 $(\ell-1) \circ c_\ell = n-1$.

Fig. 4 shows the automata A and B in the case of $m = 5$ and $n = 6$. By Proposition 18, K and L are subword-free. Construct the product automaton $A \times B$ for $K \cap L$. To get a trim NFA, omit all the unreachable and dead states; see Fig. 5 for an

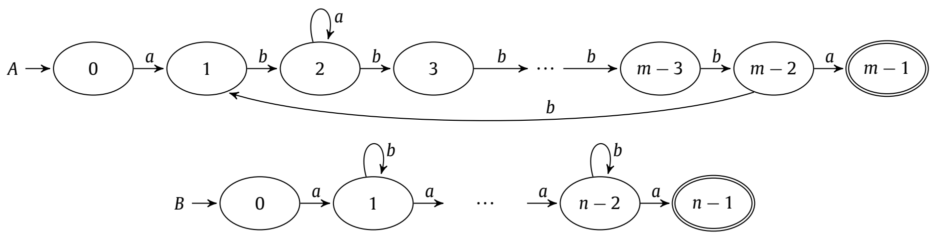

Fig. 3. Binary factor-free witnesses for intersection meeting the upper bound $mn - 2(m+n-3)$.

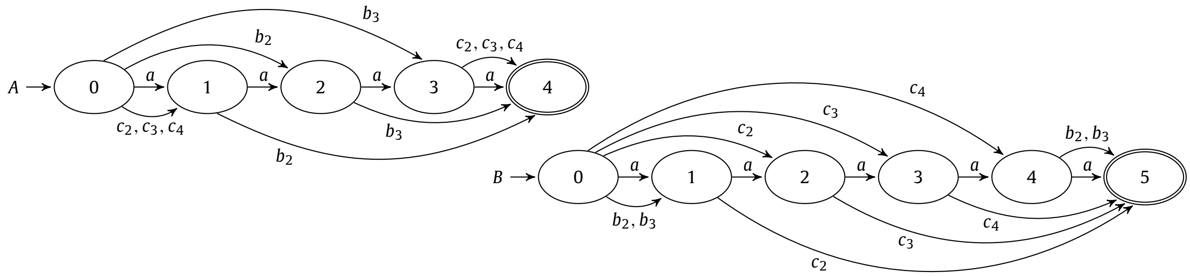

Fig. 4. Subword-free witnesses for intersection meeting the upper bound $mn - 2(m+n-3)$; $m = 5, n = 6$, $|\Sigma| = m + n - 5$.

illustration in the case of $m = 5$ and $n = 6$. The resulting trim NFA has $mn - 2(m+n-3)$ states, it is a partial DFA, and its reverse is a partial DFA as well. By Lemma 5, this NFA is minimal. $\square$

Now we show that the upper bound mn is asymptotically tight for binary subword-free languages in the case of $m = n$.

Proposition 20. *Let $n \geq 4$. There exist binary subword-free languages K and L accepted by n-state* NFA*s such that every* NFA *for $K \cap L$ has at least $(n^2 + 3n)/6$ states.*

Proof. Let $k = \lfloor (n-1)/3 \rfloor$. Let K be the language accepted by NFA $A = (\{0, 1, \dots, n-1\}, \{a, b\}, \cdot, 0, \{n-1\})$, with transitions defined as follows:

$i \cdot a = i + 1$ if $0 \leq i \leq n - 2$,
$i \cdot b = i + 2$ if $0 \leq i \leq 2k - 2$ and i is even,
$i \cdot b = i + 1$ if $n - k - 1 \leq i \leq n - 2$.

Fig. 6 shows the NFA A in the case of $n = 13$. By Proposition 18, K is subword-free. Let L be accepted by the NFA B obtained from A^R by renaming each state i to $n - 1 - i$ if $0 \leq i \leq n - 1$. We have $L = K^R$, so L is subword-free as well. To prove that $\operatorname{nsc}(K \cap L) \geq (n^2 + 3n)/6$, consider the product automaton $A \times B$. Construct an NFA N from this product automaton by omitting all the unreachable and dead states. The state $(0, 0)$ is the initial state of N, and each state (j, j) on the main diagonal is reached from $(0, 0)$ by the string a^j. Next, each state $(2i, i)$ with $1 \leq i \leq k$ is reached from $(2i-2, i-1)$ by b and the state $(2i + j, i + j)$ with $0 \leq j \leq n - 1 - 3i$ is reached from $(2i, i)$ by a string a^j. Notice that N is a partial DFA and $N = N^R$ since $K \cup L = L^R \cup K^R$. By Lemma 5, the NFA N is minimal. The total number of singleton sets that are reachable and co-reachable is $\sum_{i=0}^{k} n - 3i = (k+1)(2n - 3k)/2 \geq (n^2 + 3n)/6$. $\square$

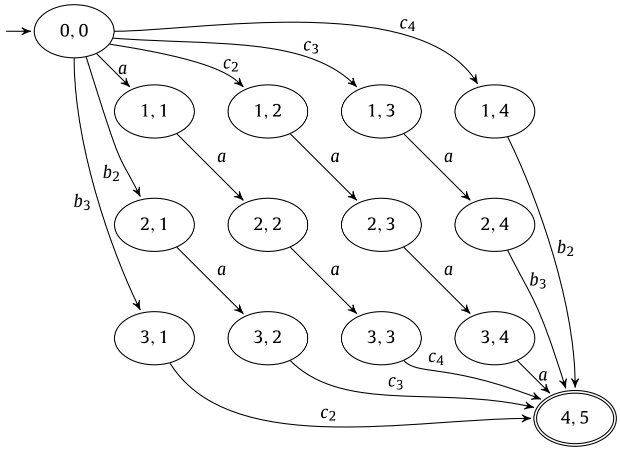

Fig. 5. The product NFA for intersection of languages from Fig. 4.

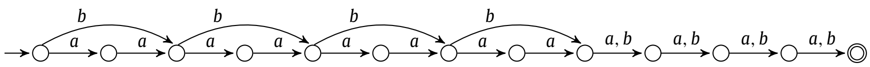

Fig. 6. A binary subword-free language K meeting the bound $(n^2 + 3n)/6$ on $\operatorname{nsc}(K \cap K^R)$; $n = 13$.

Please cite this article in press as: M. Hospodár et al., Nondeterministic complexity in subclasses of convex languages, Theoret. Comput. Sci. (2019), https://doi.org/10.1016/j.tcs.2018.12.027

7. Union

The union of languages K and L accepted by an m-state NFA and an n-state NFA, respectively, is accepted by an NFA with at most $m+n+1$ states, and this upper bound is known to be tight in the binary case [5, Theorem 1]. Let us emphasize that the union can be accepted by an NNFA of $m+n$ states, and therefore the fooling set method described in Lemma 2 cannot be used here because 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 21 *([24, Lemma 4]). Let $\mathcal{A}$ and $\mathcal{B}$ be disjoint 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.* □

In this section, we show that the upper bound $m+n+1$ can be met by the union of binary subword-closed (so subword-convex) languages. Next we prove that the complexity of union is $m+n$ on the classes of prefix-free and right ideal languages, it is $m+n-1$ on suffix-free and left ideal languages, and it is $m+n-2$ on the remaining subclasses of convex languages. All our witnesses are defined over a binary alphabet which is always optimal. The next result is from our conference paper [38]. Here we provide a simplified proof.

Theorem 22. *Let K and L be convex languages accepted by an m-state* NFA *and an n-state* NFA*, respectively. Then the language $K\cup L$ is accepted by an* NFA *of at most $m+n+1$ states, and this bound is met by binary subword-closed languages. The binary alphabet is optimal.*

Proof. The upper bound is the same as for regular languages. For tightness, consider subword-closed, so subword-convex, languages $K=\{w\in\{a,b\}^*\mid |w|_a\le m-1\}$ and $L=\{w\in\{a,b\}^*\mid |w|_b\le n-1\}$. Let
$\mathcal{A}=\{(b^na^i,a^{m-1-i}b)\mid 0\le i\le m-1\}$,
$\mathcal{B}=\{(ab^{n-1-j},b^ja^m)\mid 0\le j\le n-1\}$,
$u=a^mb^{n-1}$, and
$v=a^{m-1}b^n$.
In each string in $K\cup L$, either the number of occurrences of a is at most $m-1$ or the number of occurrences of b is at most $n-1$. It follows that $\mathcal{A}\cup\mathcal{B}$, $\mathcal{A}\cup\{(\varepsilon,u)\}$, and $\mathcal{B}\cup\{(\varepsilon,v)\}$ are fooling sets for $K\cup L$. By Lemma 21, $\mathrm{nsc}(K\cup L)\ge m+n-1$. The binary alphabet is optimal by Theorem 14, cases (3) and (4). □

The results in the next theorem are from our conference papers [38,36]. Here we fix an error in the proof of [36, Theorem 9]. For the sake of completeness, we also give the other proofs, simplified if possible.

Theorem 23. *Let $m,n\ge 3$. Let K and L be languages accepted by an m-state* NFA *and an n-state* NFA*, respectively.*

(1) If K and L are prefix-free or right ideal languages, then $K\cup L$ is accepted by an NFA *of at most $m+n$ states, and this bound is tight in the binary case.*

(2) If K and L are suffix-free or left ideal languages, then $K\cup L$ is accepted by an NFA *of at most $m+n-1$ states, and this bound is tight in the binary case.*

(3) If K and L are factor-free or two-sided ideal languages, then $K\cup L$ is accepted by an NFA *of at most $m+n-2$ states, and this bound is met by binary subword-free and binary all-sided ideal languages.*

The binary alphabet is always optimal.

Proof. (1) If K and L are prefix-free, then we may assume that NFAs A and B are non-exiting and have a unique final state. To get an $(m+n)$-state ε-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.

In [14] 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 [14, 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 $K\cup L$; indeed, both mismatched concatenations of the pairs $(\varepsilon,a^{m-1}b)$ and (a^{m-1},b) are in K, so in $K\cup L$.

To prove tightness correctly, consider languages K and L accepted by m-state and n-state NFAs A and B, respectively, shown in Fig. 7. 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. Let
$\mathcal{A}=\{(a^{m-1+i},a^{m-2-i}b)\mid 0\le i\le m-2\}\cup\{(a^{m-2}b,\varepsilon)\}$,
$\mathcal{B}=\{(b^{n-1+j},b^{n-2-j}a)\mid 0\le j\le n-2\}$.
$u=b^{n-2}a$, and
$v=a^{m-2}b$.
We have $\{a^{2m-3}b,a^{m-2}b,b^{2n-3}a,b^{n-2}a\}\subseteq K\cup L$. Next, every string in $K\cup L$ starting with a has $(m-2)\bmod(m-1)$ consecutive occurrences of a followed by b, and every string starting with b has $(n-2)\bmod(n-1)$ consecutive occurrences of b followed by a. This means that the sets $\mathcal{A}\cup\mathcal{B}$, $\mathcal{A}\cup\{(\varepsilon,u)\}$, and $\mathcal{B}\cup\{(\varepsilon,v)\}$ are fooling sets for $K\cup L$. By Lemma 21, $\mathrm{nsc}(K\cup L)\ge m+n$.

Fig. 7. Binary prefix-free witnesses for union meeting the upper bound $m+n$.

If K and L are right ideals, then we may assume that NFAs A and B have exactly one final sink state. 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 connected to the initial states of A and B through ε-transitions. To prove tightness, let K' and L' be the languages over $\{a,b\}$ accepted by NFAs A and B from Fig. 7. Set $K = K' \cdot \{a,b\}^*$ and $L = L' \cdot \{a,b\}^*$. Then K and L are right ideals accepted by m-state and n-state NFAs obtained from A and B, respectively, by adding the loop on a and b in the final states p_{m-1} and q_{n-1}. We prove that $\mathrm{nsc}(K \cup L) \geq m+n$ exactly the same way as in the case of prefix-free languages.

(2) If K and L are suffix-free, then we may assume that NFAs 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. Consider the same prefix-free languages K and L as in case (1). Then 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 4, this NFA is minimal. Hence we get $\mathrm{nsc}(K^R \cup L^R) \geq m+n-1$.

If K and L are left ideals, we may assume by Proposition 12 that NFAs A and B have a loop on each symbol in the initial state which has no other in-transitions. We can get an $(m+n-1)$-state NFA for $K \cup L$ from NFAs A and B by merging the initial states. To prove tightness, consider two left ideals shown in Fig. 8. Let

$$\mathcal{F} = \{(a^i, a^{m-1-i}) \mid 0 \leq i \leq m-1\} \cup \{(b^j, b^{n-1-j}) \mid 1 \leq j \leq n-2\} \cup \{(b^{n-1}, ab^{n-2})\}.$$

We have $\{a^{m-1}, b^{n-1}, b^{n-1}ab^{n-2}\} \subseteq K \cup L$. Next, no string in a^* (b^*) of length different from $m-1$ ($n-1$) is in $K \cup L$, and if $0 \leq i \leq m-1$ and $1 \leq j \leq n-2$, then $a^i \cdot b^{n-1-j} \notin K \cup L$ and $a^i \cdot ab^{n-2} \notin K \cup L$. It follows that the set $\mathcal{F}$ is a fooling set for $K \cup L$, so $\mathrm{nsc}(K \cup L) \geq m+n-1$.

(3) 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. The upper bound $m+n-2$ is met by binary subword-free languages $\{a^{m-1}\}$ and $\{b^{n-1}\}$ since the NFA for their union, resulting from the construction above, as well as its reverse, are partial trim DFAs, so this NFA is minimal by Lemma 5.

If K and L are two-sided ideals, then they are left ideals and also right ideals, and we may assume by Proposition 12 that A and B have properties claimed there. To get an $(m+n-2)$-state NFA for $K \cup L$ from NFAs A and B, merge their initial and final states. For tightness, consider all-sided ideals $K = \{w \in \{a,b\}^* \mid |w|_a \geq m-1\}$ and $L = \{w \in \{a,b\}^* \mid |w|_b \geq n-1\}$. Notice that the set of pairs of strings $\{(a^i, a^{m-1-i}) \mid 0 \leq i \leq m-1\} \cup \{(b^j, b^{n-1-j}) \mid 1 \leq j \leq n-2\}$ is a fooling set for $K \cup L$.

The optimality of a binary alphabet follows from Theorem 14 for all classes. □

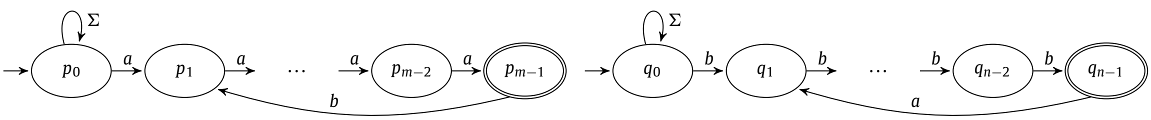

Fig. 8. Binary left ideal witnesses for union meeting the upper bound $m+n-1$.

8. Reversal

The upper bound on the nondeterministic state complexity of reversal is $n+1$ [5, Theorem 10] and it is met by the binary language shown in Fig. 13 (left) [25, Proof of Theorem 2]. In this section, we prove that the upper bound $n+1$ is tight for suffix-free languages, left ideals, and all four classes of closed (so, also convex) languages. Otherwise, the tight upper bound is n. Except for subword-closed case where we use a growing alphabet of size $2n-2$, all our witnesses are defined over binary or unary alphabets which are always optimal. Similarly as for the union operation, we need to use a modified fooling set method described in Lemma 21 in some cases.

The results in the next theorem, except for binary factor-closed witness, are from our conference papers [38,36]. Here we provide a detailed proof for free languages, cf. [36, Theorem 12], the proof for ideal languages which was missing in [38, Theorem 17], we improve the result for factor-closed languages [38, Theorem 9] by describing a binary witness, and give a simplified proof for subword-closed languages.

Theorem 24. *Let $n \geq 5$. Let L be a suffix-free, left ideal, or closed language accepted by an n-state* NFA. *Then L^R is accepted by an* NFA *with at most $n+1$ states, and this bound is met by a binary suffix-free, binary left ideal, binary factor-closed language, and a subword-closed language over an alphabet of size $2n-2$. The binary alphabet is always optimal.*

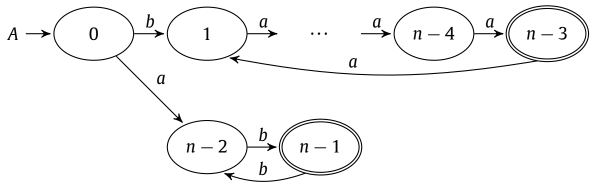

Fig. 9. A binary suffix-free witness for reversal meeting the upper bound $n+1$.

Proof. The upper bound is the same as for regular languages. To get tightness in the suffix-free case, let L be the language accepted by the NFA A shown in Fig. 9. Since every string in L contains 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
$\mathcal{A} = \{(a^{n-3}, a^{n-4}b)\} \cup \{(a^i, a^{n-4-i}b) \mid 1 \le i \le n-4\} \cup \{(a^{n-4}b, \varepsilon)\}$,
$\mathcal{B} = \{(bb, ba), (b, a)\}$,
$u = ba$, and $v = a^{n-4}b$.
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 continues with b. Next, the string bba is not in L^R. Finally, every string in L^R contains both a and b, and no string in L^R starts with bba. 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 21, we have $\text{nsc}(L^R) \ge n+1$.

Now, we describe a left ideal witness. Let K be the language accepted by the NFA A shown in Fig. 9. Set $L = \{a,b\}^*K$. Then L is a left ideal accepted by the n-state NFA obtained from A by adding the loop on a and b in the initial state. We prove that $\text{nsc}(L^R) \ge n+1$ exactly the same way as in the case of suffix-free languages.

Next, let us describe a factor-closed witness. Let L be the language accepted by the NFA shown in Fig. 10 where the only two transitions on b are $(0, b, n-2)$ and $(n-1, b, n-2)$. Since each state of A is final, the language L is prefix-closed by Proposition 13 (1). 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 by Proposition 13 (2). Since L is prefix-closed and suffix-closed, it is factor-closed by Proposition 13 (3). 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 $\{ba^{n-1}, a^{n-1}, a^{n-2}\} \subseteq L^R$, no string in L^R has more than $n-1$ consecutive a's, and $ba^{n-2} \notin L^R$. 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 21, $\text{nsc}(L^R) \ge n+1$. The binary alphabet is optimal

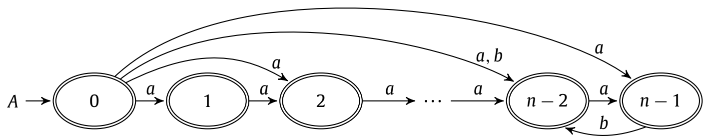

Fig. 10. A binary factor-closed witness for reversal meeting the upper bound $n+1$.

in all cases since the reversal of every unary language is the same language.

Finally, we describe a subword-closed witness. Let L be the language accepted by the NFA shown in Fig. 11. Let
$\mathcal{A} = \{(b_1 \cdots b_{i-1} b_{i+1} \cdots b_{n-1}, a_i) \mid 2 \le i \le n-1\} \cup \{(b_1 a_2, \varepsilon)\}$,
$\mathcal{B} = \{(b_2 b_3 \cdots b_{n-1}, a_1)\}$,
$u = a_1$, and $v = a_2$.
Since $\{a_1, a_2, b_1 a_2, b_2 b_3 \cdots b_{n-1} a_1\} \cup \{b_1 \cdots b_{i-1} b_{i+1} \cdots b_{n-1} a_i \mid 2 \le i \le n-1\} \subseteq L^R$, no string in L^R has b_i before a_i, and no string in L^R has two symbols in $\{a_1, \ldots, a_{n-1}\}$, the sets $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, u)\}$, and $\mathcal{B} \cup \{(\varepsilon, v)\}$ are fooling sets for L^R. Thus by Lemma 21, $\text{nsc}(L^R) \ge n+1$. □

The reversal operation on the other classes of convex languages is a trivial operation, as shown in the next observation.

Proposition 25. *Let L be a prefix-free or right ideal language accepted by an n-state* NFA. *Then L^R is accepted by an* NFA *of at most n states, and this bound is met by a unary subword-free and a unary all-sided ideal language.*

Proof. If L is prefix-free or right ideal, then it is accepted by an NFA with a unique final state by Propositions 8 and 12. This gives the upper bound. The bound is met by the unary subword-free language $\{a^{n-1}\}$ and by the unary all-sided ideal $\{a^i \mid i \ge n-1\}$ since the set $\{(a^i, a^{n-1-i}) \mid 0 \le i \le n-1\}$ is a fooling set for the reversal of both languages. □

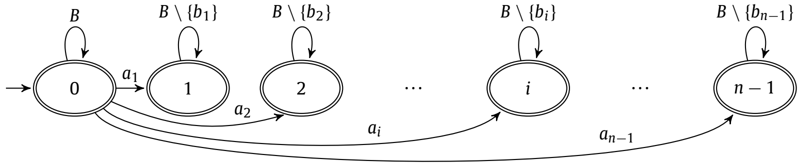

Fig. 11. The DFA of the subword-closed language L where $B = \{b_1, \dots, b_{n-1}\}$.

9. Star

The upper bound on the nondeterministic state complexity of the star operation is $n+1$ and it is known to be tight already in the unary case [5, Theorem 9]. In this section we show that the upper bound $n+1$ is tight in all the classes of ideal (so also convex) languages with binary witnesses. Next we prove that the complexity of the star operation is n in the classes of prefix- and suffix-free and prefix- and suffix-closed languages with binary witnesses, it is $n-1$ in the classes of factor- and subword-free languages with unary witnesses, and it is one for every factor-closed language. Moreover, the binary alphabet is always optimal. This summarizes our results stated in our conference papers [38,36]. Here we provide a proof for ideal languages which was missing in [38, Theorem 16] and a more detailed proof for suffix-closed languages. For the sake of completeness, we give all the remaining proofs which are slightly modified here.

Proposition 26. *Let $n \geq 2$. Let L be a convex language accepted by an n-state* NFA. *Then L^* is accepted by an* NFA *with at most $n+1$ states, and this bound is met by a binary all-sided ideal language. The binary alphabet is optimal.*

Proof. The upper bound is the same as for regular languages. To prove tightness, consider the binary all-sided ideal language $L = \{w \in \{a,b\}^* \mid |w|_a \geq n-1\}$. Let $\mathcal{F} = \{(ba^i, a^{n-1-i}b) \mid 0 \leq i \leq n-1\} \cup \{(\varepsilon, \varepsilon)\}$. We have $\{\varepsilon, ba^{n-1}b\} \subseteq L^*$, while $\{ba^jb \mid j \leq n-2\} \cup \{ba^j \mid j \leq n-2\} \cup \{b\} \subseteq (L^*)^c$. It follows that $\mathcal{F}$ is a fooling set for L^*, so $\text{nsc}(L^*) \geq n+1$. The binary alphabet is optimal by Theorem 14 (4). □

Theorem 27. *Let $n \geq 2$. Let L be a prefix- or suffix-free or prefix- or suffix-closed language accepted by an n-state* NFA. *Then L^* is accepted by an* NFA *with at most n states, and this bound is tight in the binary case. The binary alphabet is optimal.*

Proof. If L is prefix-free, then it is accepted by an NFA A which is non-exiting and has a unique final state q_f by Proposition 8. 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. For tightness, let consider the prefix-free language accepted by the n-state NFA A shown in Fig. 12 (left). The set $\{(ba^i, a^{n-1-i}) \mid 0 \leq i \leq n-1\}$ is a fooling set for L^* since $ba^{n-1} \in L^*$, but for each j with $j < n-1$, the string ba^j is not in L^*.

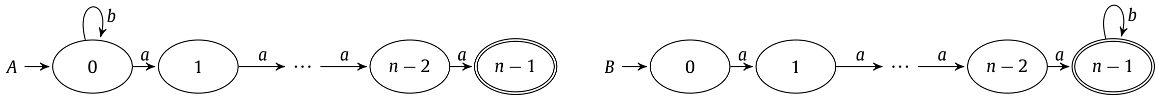

Fig. 12. Binary prefix-free (left) and suffix-free (right) witnesses for star meeting the upper bound n.

If L is suffix-free, then it is accepted by an NFA A which is non-returning by Proposition 8. Construct an n-state NFA for L^* from A by making the initial state s final, and by adding the transition (p, a, q) for each final state p, each symbol a, and each transition (s, a, q) of A. For tightness, let L be the suffix-free language accepted by the n-state NFA B shown in Fig. 12 (right). The set $\{(a^i, a^{n-1-i}b) \mid 0 \leq i \leq n-1\}$ is a fooling set for L^* since $a^{n-1}b \in L^*$, but for each j with $j < n-1$, the string a^jb is not in L^*.

If L is a closed language accepted by an n-state NFA, then $\varepsilon \in L$, so $\text{nsc}(L^*) \leq n$. This upper bound is met by the binary prefix-closed language from [44, Theorem 17] accepted by the NFA shown in Fig. 13 (left) since $\{(a^i, a^{n-1-i}b) \mid 0 \leq i \leq n-1\}$ is a fooling set for its star. The suffix-closed language accepted by the NFA shown in Fig. 13 (right) meets the upper bound as well; here the set $\{(ba^i, a^{n-1-i}) \mid 0 \leq i \leq n-1\}$ is a fooling set for its star.

The binary alphabet is optimal in all four cases by Theorem 14. □

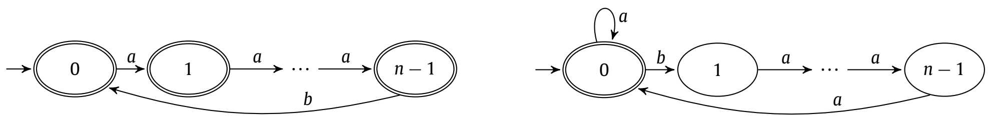

Fig. 13. Binary prefix-closed (left) and suffix-closed (right) witnesses for star meeting the upper bound n.

Proposition 28. *Let $n \geq 2$. Let L be a language accepted by an n-state* NFA.

(1) If L is factor-free, then L^* is accepted by an NFA *with at most $n-1$ states, and this bound is met by a unary subword-free language.*

(2) If L is factor-closed, then L^* is accepted by an NFA *with one state.*

Proof. (1) If L is factor-free, then it is prefix-free and suffix-free, and we may assume that it is accepted by an NFA A which is non-returning and non-exiting, and it has a unique final state q_f by Proposition 8. Let s be the initial state of A. Construct an $(n-1)$-state NFA for L^* by making the initial state s non-initial and the state q_f initial, by adding transition (q_f, a, q) for each transition (s, a, q) in A, and by omitting the unreachable state s. The unary subword-free language $\{a^{n-1}\}$ meets this upper bound.

(2) For a factor- or subword-closed language L, let Γ be the set of symbols present in at least one string of L. Then $L \subseteq \Gamma^*$, and since L is factor-closed, $\Gamma \cup \{\varepsilon\} \subseteq L$. It follows that $L^* = \Gamma^*$, hence $\mathrm{nsc}(L^*) = 1$. □

10. Complementation

The upper bound on the nondeterministic state complexity of complementation is 2^n [5, Corollary 1] and it is met by the binary language K from [25, Theorem 5] accepted by the NFA shown in Fig. 14.

Notice that in this NFA, each singleton set $\{i\}$ is reached by a^i, the empty set is reached from $\{n-1\}$ by a, and every k-element set $\{i_1, i_2, \ldots, i_k\}$, where $2 \leq k \leq n$ and $0 \leq i_1 < i_2 < \cdots < i_k \leq n-1$, is reached from the $(k-1)$-element set $\{i_2 - i_1 - 1, \ldots, i_k - i_1 - 1\}$ by the string ba^{i_1}. Hence, by induction, every subset of the state set is reachable in this NFA. Moreover, this NFA is isomorphic to its reversal, and therefore it has 2^n co-reachable subsets. This means that $\mathrm{nsc}(K^c) = 2^n$ by Corollary 7, which significantly simplifies the proof in [25, Theorem 5]. The language K plays a crucial role in our investigation of the complementation operation in the subclasses of convex languages.

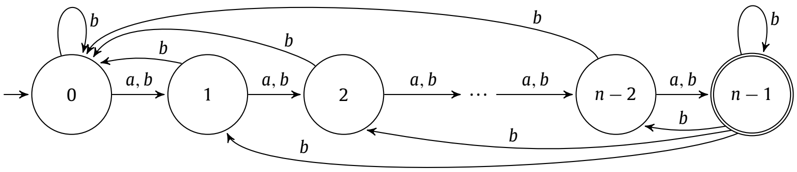

Fig. 14. An NFA of a binary regular language K with $\mathrm{nsc}(K) = 2^n$.

In this section, we show that the nondeterministic complexity of complementation is 2^n on prefix-closed (so prefix-convex) and suffix-convex languages. The exact complexity of complementation on factor- and subword-convex remains open, and it is between 2^{n-2} and $2^{n-1}+1$ in all the remaining subclasses. Let us emphasize that except for growing alphabets in three cases, and a five-letter alphabet in the suffix-convex case, all the remaining alphabets used for describing witnesses are optimal.

A ternary prefix-closed witness was described in [44]. We decreased the alphabet size in [38, Theorem 10 (a)]. We also described a suffix-convex witness in [36, Theorem 13]. Here we present a simplified proof in the prefix-closed case and a more detailed proof in the suffix-convex case. To show that a DFA accepts a prefix-convex language, we use the following sufficient condition.

Proposition 29. *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. □

Theorem 30. *Let $n \geq 3$. Let L be a convex language accepted by an n-state* NFA. *Then L^c is accepted by an* NFA *with at most 2^n states, and this bound is met by a binary prefix-closed language and a suffix-convex language defined over an alphabet of size 5. The binary alphabet in the prefix-closed case is optimal.*

Proof. The upper bound is the same as for regular languages. Consider the binary prefix-closed language accepted by the NFA N shown in Fig. 15. Notice that we can move every subset S of $\{0, 1, \ldots, n-1\}$ cyclically by one by reading a if $n-1 \notin S$ or if both of $n-1$ and $n-2$ are in S, and by reading ab otherwise. Next, we can eliminate the state $n-1$ from every set containing $n-1$ by reading b. Finally, the set $\{0, 1, \ldots, n-1\}$ is reachable by a^{2n}. It follows that every subset of the state

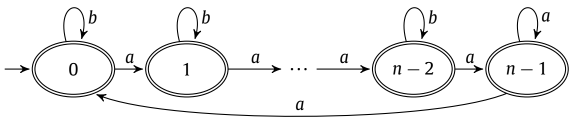

Fig. 15. A binary prefix-closed witness for complementation meeting the upper bound 2^n.

set is reachable in N. The initial state in N^R is $\{0, 1, \dots, n-1\}$, and we show reachability of all the subsets in N^R by the similar arguments as for N. By Corollary 7, $\text{nsc}(L^c) = 2^n$. The binary alphabet is optimal by Theorem 14 (3).

Now, let us describe a suffix-convex witness. Let L be the language accepted by the nondeterministic finite automaton $A = (\{0, 1, \dots, n-1\}, \{a, b, c, d, e\}, \cdot, 0, \{1, 2, \dots, n-1\})$, where the transitions on a and b are shown in Fig. 16, the transitions on c, d, e are as follows:

$0 \cdot c = \{0, 1, \dots, n-1\}$,
$0 \cdot d = \{1, 2, \dots, n-1\}$,
$q \cdot e = \{n-1\}$ 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 $\{0, 1, \dots, n-1\}$ on e. 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 29, the language L^R is prefix-convex, so L is suffix-convex. Now we 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 6, we have $\text{nsc}(L^c) = 2^n$. $\square$

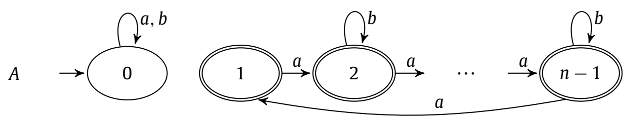

Fig. 16. Transitions on a and b in a suffix-convex witness for complementation meeting the upper bound 2^n.

The following theorem recalls the results from our conference paper [38, Theorem 10 (b, c)]. We present a simplified proof for factor-closed case which uses reachable and co-reachable sets given by Proposition 6.

Theorem 31. *Let $n \geq 3$. Let L be a suffix-closed language accepted by an n-state* NFA. *Then L^c is accepted by an* NFA *with at most $2^{n-1} + 1$ states. This upper bound is met by a binary factor-closed language and by a subword-closed language defined over an alphabet of size 2^n. The binary alphabet is optimal.*

Proof. Let $A = (Q, \Sigma, \cdot, s, F)$ be a minimal NFA for L. Then every state q in Q is reachable from s by a string u and also some final state is reachable from q by a string v. Thus uv is accepted. Since L is suffix-closed, the string v must reach a final state from s. Therefore every subset of Q containing s is equivalent to $\{s\}$ in the subset automaton of A. It follows that L^c is accepted by an NFA of at most $2^{n-1} + 1$ states.

For tightness, consider the binary language accepted by the NFA N shown in Fig. 17. Since all states of N are final, $L(N)$ is prefix-closed by Proposition 13 (1). Notice that if a string w is accepted from any state of N, then w is also accepted from the initial state 0. By Proposition 13 (2), the language $L(N)$ is suffix-closed, hence it is factor-closed by Proposition 13 (3). The subset $\{0, 1, \dots, n-1\}$ is reached from the initial state by a. Next, the 2^{n-1} subsets of $\{1, 2, \dots, n-1\}$ are reachable in N since the set $\{1, 2, \dots, n-1\}$ is reached by b, we can shift every subset of this set cyclically by one by a, and we can eliminate state $n-1$ by b. Moreover, complements of these subsets are co-reachable. By Proposition 6, $\text{nsc}(L^c) \geq 2^{n-1} + 1$. The binary alphabet is optimal by Theorem 14 (3).

To describe a subword-closed witness, let $A = \{a_S \mid S \subseteq \{1, 2, \dots, n-1\}\}$, $B = \{b_S \mid S \subseteq \{1, 2, \dots, n-1\}\}$, and $\Sigma = A \cup B$. Let L be the language accepted by the 2^n-letter and n-state NFA $N = (\{0, 1, \dots, n-1\}, \Sigma, \cdot, 0, \{0, 1, \dots, n-1\})$ where for every a_S and b_S we have

Please cite this article in press as: M. Hospodár et al., Nondeterministic complexity in subclasses of convex languages, Theoret. Comput. Sci. (2019), https://doi.org/10.1016/j.tcs.2018.12.027

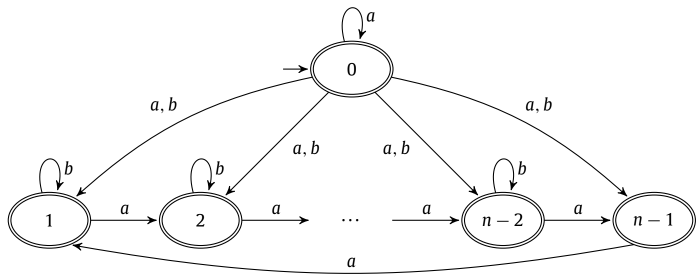

Fig. 17. A binary factor-closed witness for complementation meeting the upper bound $2^{n-1}+1$.

$0 \cdot b_S = S$ and $0 \cdot b_S = \{0\}$,
$i \cdot b_S = \{i\}$ if $i \in \{1, 2, \ldots, n-1\} \setminus S$,
and all the remaining transitions go to the empty set. Notice that each string in L is either in B^*, or in B^*A, or it is in the form $w = b_{R_1} \cdots b_{R_k} a_S b_{T_1} \cdots b_{T_\ell}$ with $S \cap T_1^c \cap \cdots \cap T_\ell^c \neq \emptyset$. Since N accepts every string in $B^* \cup B^*A$, and removing a symbol b_{T_j} from w results in a string in the same form as w, every subword of every string in L is in L as well. Hence L is subword-closed. Let us show that the set of pairs of strings $\mathcal{F} = \{(\varepsilon, a_\emptyset)\} \cup \{(a_S, b_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ is a fooling set for L^c. The string $a_\emptyset$ as well as each string $a_S b_S$ is not accepted by N. On the other hand, each string b_S is accepted by N. Let $S \neq T$, and without loss of generality, there is a state i in $\{1, 2, \ldots, n-1\}$ such that $i \in S \setminus T$. Then the string $a_S b_T$ is accepted by N. It follows that $\mathcal{F}$ is a fooling set for L^c. Hence we have $\operatorname{nsc}(L^c) \geq 2^{n-1}+1$ by Lemma 2. □

Han et al. [14] provided an upper bound $2^{n-1}+1$ and lower bound 2^{n-1} on complexity of complementation on prefix-free languages. Let us show that 2^{n-1} is also an upper bound; cf. our results in [37, Lemma 2, Theorem 1].

Proposition 32. *Let $n \geq 3$. Let L be a prefix-free language accepted by an n-state* NFA. *Then L^c is accepted by an* NFA *of at most 2^{n-1} states. This upper bound is tight in the ternary case.*

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 to the DFA A. Then A has a unique final state q_f which goes to the dead state q_d on each input symbol. To get a DFA for the language L^c, we invert the finality of every state in the DFA A, thus L^c is accepted by the $(2^{n-1}+1)$-state DFA $A^c = (Q, \Sigma, \cdot, s, Q \setminus \{q_f\})$. 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 replacing each transition (q, a, q_d) with two transitions (q, a, q_f) and (q, a, s) and by omitting the state q_d; see Fig. 18. Formally, construct an NFA $N^c = (Q \setminus \{q_d\}, \Sigma, \circ, s, Q \setminus \{q_f, q_d\})$, where

$$q \circ a = \begin{cases} q \cdot a, & \text{if } q \cdot a \neq q_d, \\ \{q_f, s\}, & \text{if } q \cdot a = q_d. \end{cases}$$

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

The empty string is accepted in both automata. Let $w = a_1 a_2 \cdots a_k$, where $a_i \in \Sigma$, be a non-empty 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 $q_{k-1} \cdot a_k = q_d$, so $s \in q_{k-1} \circ a_k$. 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 $q_{\ell-1} \circ a_\ell = \{q_f, s\}$, and $q_f \circ a = \{q_f, s\}$ 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 $s, q_1, q_2, \ldots, q_k$ 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

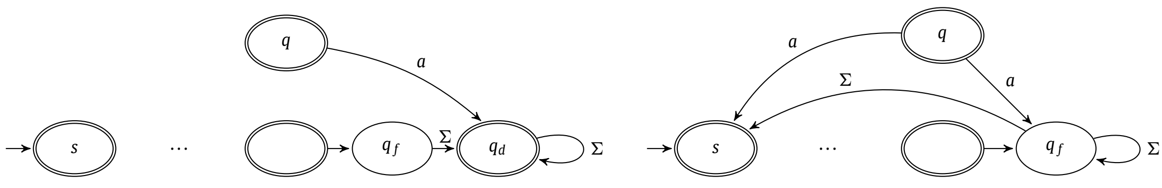

Fig. 18. The substitution of a transition (q, a, q_d) by the transitions (q, a, q_f) and (q, a, s).

state q_i is different from q_d, and therefore in the NFA N^c, we have $q_{i-1} \circ a_i = \{q_{i-1} \cdot a_i\}$. This means that $s, q_1, q_2, \dots, q_k$ is a unique computation of N^c on w. Since this computation is rejecting, the string w is rejected by the NFA N^c.

Notice that the witness from [14, Lemma 4.6] is actually the ternary language $K \cdot c$ where K is the witness language on $n-1$ states from Fig. 14. The language is prefix-free by Proposition 11. □

We proved in our conference paper [37, Lemma 9] that the upper bound 2^{n-1} cannot be met by any binary prefix-free language; here we present a slightly modified proof.

Proposition 33. *Let L be a binary prefix-free language accepted by an n-state* NFA. *Then, for all sufficiently large n, the language L^c is accepted by an* NFA *with at most $2^{n-1} - 2^{n-3} + 1$ states.*

Proof. Let N be a minimal NFA for L with the state set $\{1, 2, \dots, m\}$ where $m \le n$. Let m be the final state of N. Without loss of generality, the state m is reached from the state $m-1$ on a in N. If there is no transition (i, a, j) with $i, j \in \{1, 2, \dots, m-1\}$, then the automaton on states $\{1, 2, \dots, m-1\}$ is unary. It follows that in the subset automaton of N, at most $O(F(m-1))$ subsets of $\{1, 2, \dots, m-1\}$ can be reached by Chrobak's result in [40, Proof of Theorem 4.4], here $F(n)$ is the Landau function defined by $F(n) = \max\{\mathrm{lcm}(x_1, \dots, x_k) \mid n = x_1 + \cdots + x_k\}$. Since $F(n) \approx e^{\sqrt{n \ln n}}$, the lemma follows in this case.

Now consider a transition (i, a, j) with $i, j \in \{1, 2, \dots, m-1\}$. We show that no subset of $\{1, 2, \dots, m-1\}$ containing states i and $m-1$ may be reachable. Assume for contradiction, that a set $S \cup \{i, m-1\}$ is reached from the initial state of the subset automaton by a string u. Since N is minimal, the final state m is reached from the state j by a non-empty string v. However, the set $S \cup \{i, m-1\}$ goes to a final set $S' \cup \{j, m\}$ by a, and then to a final set $S'' \cup \{m\}$ 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^{m-3} subsets of $\{1, 2, \dots, m-1\}$ are unreachable. Therefore, the subset automaton has at most $2^{m-1} - 2^{m-3} + 1$ states. After inverting the finality of its states, we get a DFA of the same size for the complement of $L(N)$, and the lemma follows. □

Consider the witness language $K \cdot c$ from the proof of Proposition 32 and apply the homomorphism $h(a) = 10$, $h(b) = 11$, and $h(c) = 00$ to it. The resulting language is a binary prefix-free language accepted by a $2n$-state NFA whose complement requires 2^{n-1} states. Hence the complexity of complementation on binary prefix-free languages is still exponential. The next theorem summarizes all our results for prefix-free languages.

Theorem 34. *Let L be a prefix-free language accepted by an n-state* NFA. *Then L^c is accepted by an* NFA *with at most 2^{n-1} states, and this bound is tight in the ternary case. The ternary alphabet is optimal. In the binary case, an upper bound is $2^{n-1} + 2^{n-3} + 1$ and a lower bound is $2^{\lfloor \frac{n}{2} \rfloor - 1}$.* □

Han et al. [15] provided an upper bound $2^{n-1} + 1$ and a lower bound $2^{n-1} - 1$ for complementation on suffix-free languages. We show that the tight upper bound is 2^{n-1} here. The following theorem summarizes our results on suffix-free languages from [37, Theorem 2] and [45, Theorem 13].

Theorem 35. *Let $n \ge 3$. Let L be a suffix-free language accepted by an n-state* NFA. *Then L^c is accepted by an* NFA *of at most 2^{n-1} states. This upper bound is tight in the ternary case. The ternary alphabet is optimal. In the binary case, an upper bound is $2^{n-1} + 2^{n-3} + 2$ and a lower bound is $2^{\lfloor \frac{n}{2} \rfloor - 1}$.*

Proof. Let N be a non-returning n-state NFA for a suffix-free language L. The subset automaton $A = (Q, \Sigma, \cdot, s, F)$ of the NFA N has at most $1 + 2^{n-1}$ reachable states including the dead state q_d 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 inverting the finality of all states, we get a DFA $A^c = (Q, \Sigma, \cdot, s, Q \setminus F)$ for L^c of $1 + 2^{n-1}$ states. The initial state of A^c is final and has no in-transitions. The state q_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 q_d and such that they have a transition to the state q_d, that is, $Q_d = \{q \in Q \setminus \{q_d\} \mid \text{there is an } a \text{ in } \Sigma \text{ such that } q \cdot a = q_d\}$. Recall that by Proposition 9, for each symbol a, there is a state q_a in Q_d that goes to q_d by a. Replace each transition (q, a, q_d) by transitions (q, a, p) for each p in Q_d, and moreover, add the transition (q, a, s). Then, remove the state q_d. Let $\circ$ denote the transition function of N^c. Let us show that $L(N^c) = L(A^c)$.

The empty string is accepted by both automata. Let $w = a_1 a_2 \cdots a_k$ be a non-empty string in $L(A^c)$, and let $s, q_1, q_2, \dots, q_k$ be the computation of the DFA A^c on the string w. If $q_k \ne q_d$, then each q_i is different from q_d since q_d goes to itself on each symbol. It follows that $s, q_1, q_2, \dots, q_k$ 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 $q_\ell, q_{\ell+1}, \dots, q_k$ are equal to q_d, and the states $s, q_1, \dots, q_{\ell-1}$ are not equal to q_d. If $\ell = k$, then $q_{k-1} \cdot a_k = q_d$, so $s \in (q_{k-1} \circ a_k)$. It follows that $s, q_1, q_2, \dots, q_{k-1}, s$ is an accepting computation of N^c on w. If $\ell < k$, then we have $q_\ell = q_{\ell+1} = \cdots = q_k = q_d$. Let p_i be a state in Q_d with $(p_i \cdot a_i) = q_d$. Then the string w is accepted in N^c through the following accepting computation

$$s \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \cdots \xrightarrow{a_{\ell-1}} q_{\ell-1} \xrightarrow{a_\ell} p_{\ell+1} \xrightarrow{a_{\ell+1}} p_{\ell+2} \xrightarrow{a_{\ell+2}} \cdots \xrightarrow{a_{k-1}} p_k \xrightarrow{a_k} s.$$

Now assume that a string $w = a_1a_2 \cdots a_k$ is rejected by the DFA A^c. Let $s, q_1, q_2, \dots, q_k$ be the rejecting computation of the DFA A^c on the string w. Since q_d is a final state in A^c, and it goes to itself on each symbol, all the states q_i must be different from q_d. Therefore in the NFA N^c, we have $(q_{i-1} \circ a_i) = \{(q_{i-1} \cdot a_i)\}$. This means that $s, q_1, q_2, \dots, q_k$ is a unique computation of N^c on w. Since this computation is rejecting, the string w is rejected by N^c.

For tightness, let $n \geq 3$ and K be the language over $\{a, b\}$ accepted by the $(n-1)$-state NFA N from Fig. 14, and let $L = c \cdot K$. Then L is a suffix-free language by Proposition 11, and it is accepted by an n-state NFA with the state set $\{q_0\} \cup \{0, 1, \dots, n-2\}$, the initial state q_0, the unique final state $n-2$, the transitions on a, b given by Fig. 14, and the transition $(q_0, c, 0)$. The initial state q_0 goes to the state 0 by c, and then, as shown in the beginning of this section, every subset of $\{0, 1, \dots, n-2\}$ is reachable and co-reachable in N. By Corollary 7, every NFA for L^c has at least 2^{n-1} states.

Now consider the binary case. By applying the homomorphism $h(a) = 10$, $h(b) = 11$, and $h(c) = 00$ to the language $c \cdot K$, we get a binary suffix-free language accepted by a $2n$-state NFA whose complement requires 2^{n-1} states. This gives a lower bound $2^{\lfloor \frac{n}{2} \rfloor - 1}$ in the binary case.

To get the upper bound, we reverse an NFA for L, and obtain an n-state NNFA for a prefix-free language L^R. By Proposition 33, the language $(L^R)^c$ is accepted by an NFA with at most $2^{n-1} - 2^{n-3} + 1$ states. Since $(L^R)^c = (L^c)^R$, we have that the reverse of L^c is accepted by an NFA N which has at most $2^{n-1} - 2^{n-3} + 1$ states. Now we reverse the NFA N, and get an NNFA N^R for L^c. By adding one more state, we get an NFA for L^c with at most $2^{n-1} - 2^{n-3} + 2$ states. Our proof is complete. □

We continue with the complementation operation on factor- and subword-free languages and summarize the results from the conference paper [45] by the third author; cf. [45, Theorems 15, 18, 20].

Theorem 36. *Let $n \geq 4$. Let L be a factor-free language accepted by an n-state* NFA. *Then L^c is accepted by an* NFA *of at most $2^{n-2} + 1$ states. This upper bound is met by a ternary factor-free language and by a subword-free language defined over an alphabet of size 2^{n-2}. For binary factor-free languages, an upper bound is $2^{n-2} - 2^{n-4} + 1$ and a lower bound is $2^{\lfloor \frac{n}{2} \rfloor - 2} + 1$. The upper bound $2^{n-2} + 1$ cannot be met by any subword-free language defined over a fixed alphabet.*

Proof. Let A be an n-state NFA for a factor-free language L. We may assume that A is non-returning, 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 inverting the finality of all 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, we can use a nondeterminism to save one state. This gives the upper bound $2^{n-2} + 1$.

To prove tightness, let K be the witness language over $\{a, b\}$ accepted by the NFA N on $n-2$ states from Fig. 14. Let $L = c \cdot K \cdot c$. Then L is a factor-free language by Proposition 11 and it is accepted by an n-state NFA. Let $\mathcal{F} = \{(u_S, v_S) \mid S \subseteq \{1, 2, \dots, n-2\}\}$ be a fooling set for the language K^c. Notice that the strings u_S and v_S have the following properties: (1) on u_S, the initial state goes to the set S; (2) the string v_S is rejected by N from every state in S and it is accepted by N from every state in $\{1, 2, \dots, n-2\} \setminus S$. Let

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

$\mathcal{B} = \{(cu_\emptyset, v_\emptyset c)\}$,

$u = v_\emptyset c$, and $v = \varepsilon$.

Then $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, u)\}$, and $\mathcal{B} \cup \{(\varepsilon, v)\}$ are fooling sets for L^c, so by Lemma 21, $\operatorname{nsc}(L^c) \geq 2^{n-2} + 1$.

Now consider the binary case. Applying the homomorphism $h(a) = 10$, $h(b) = 11$, and $h(c) = 00$ to the language $c \cdot K \cdot c$ results in a binary factor-free language accepted by a $2n$-state NFA whose complement requires 2^{n-2} states, and the lower bound $2^{\lfloor \frac{n}{2} \rfloor - 2}$ follows.

Let N be a minimal NFA for a factor-free language L over $\{a, b\}$ with the state set $\{1, 2, \dots, n\}$, the initial state 1, and the unique final state 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 the state 1 because L is suffix-free. Therefore it is enough to consider subsets of set $\{2, 3, \dots, n-1\}$. If there is no transition (i, a, j) with $i, j \in \{2, 3, \dots, n-1\}$, then the automaton on states $\{2, 3, \dots, n-1\}$ is unary, and the number of reachable sets is much smaller than the desired upper bound. Otherwise, consider a transition (i, a, j) with $i, j \in \{2, 3, \dots, n-1\}$. We continue in the same way as in the binary prefix-free case and show that no subset of $\{2, 3, \dots, n-1\}$ containing states i and $n-1$ may be reachable. Thus at least 2^{n-4} subsets of $\{2, 3, \dots, n-1\}$ are unreachable. Therefore, the subset automaton has at most $2^{n-2} - 2^{n-4} + 1$ states. After inverting the finality of all states, we get a DFA of the same size for the complement of $L(N)$.

Finally, we show that the upper bound $2^{n-2} + 1$ is met by a subword-free language defined over an alphabet of size 2^{n-2}. Let $\Sigma = \{a_S \mid S \subseteq \{1, 2, \dots, n-2\}\}$. Let $L = \{a_S a_T \mid S, T \subseteq \{1, 2, \dots, n-2\}$ and $S \neq T\}$ be accepted by the NFA $A = (Q, \Sigma, \cdot, 0, \{n-1\})$, where $Q = \{0, 1, \dots, n-1\}$ and for each symbol a_S in Σ,

$0 \cdot a_S = S$;

$i \cdot a_S = \emptyset$ if $1 \leq i \leq n-2$ and $i \in S$;

$i \cdot a_S = \{n-1\}$ if $1 \leq i \leq n-2$ and $i \notin S$; and

$(n-1) \cdot a_S = \emptyset$.

Notice that each string in L is of length 2, so L is subword-free. Let

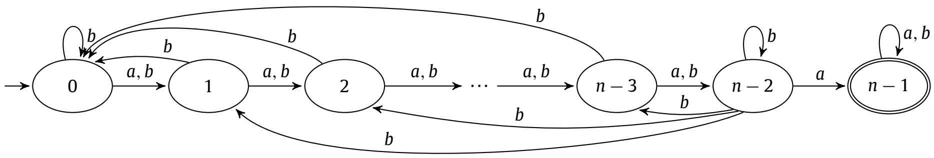

Fig. 19. A binary right ideal witness for complementation meeting the upper bound 2^{n-1}.

$\mathcal{A} = \{(a_S, a_S) \mid S \subseteq \{1, 2, \ldots, n-2\}$ and $S \neq \emptyset\}$,
$\mathcal{B} = \{(a_{\{1\}}a_{\{2\}}, a_\emptyset)\}$,
$u = a_\emptyset$, and $v = \varepsilon$.
Then $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, u)\}$, and $\mathcal{B} \cup \{(\varepsilon, v)\}$ are fooling sets for L^c, so $\mathrm{nsc}(L^c) \geq 2^{n-2} + 1$. Every subword-free regular language is finite [46], so the upper bound $O(k^{\frac{n}{\log_2(k)+1}})$ where $k = |\Sigma|$ on determinization for finite languages [47,48] implies the same upper bound for complementation on subword-free languages. It follows that the upper bound $2^{n-2} + 1$ cannot be met by a subword-free language defined over a fixed alphabet. □

The last theorem of this paper is about nondeterministic state complexity of complementation on classes of ideal languages. The proof for the two-sided ideal case was missing in the third author's conference paper [45, Theorem 28], so we provide it here. We also give all other proofs for the sake of completeness, cf. [45, Theorems 24, 26, 30].

Theorem 37. *Let $n \geq 4$. Let L be an ideal language accepted by an n-state* NFA.
(1) If L is right or left ideal, then L^c is accepted by an NFA *of at most 2^{n-1} states, and the bound is tight in the binary case.*
(2) If L is two-sided ideal, then L^c is accepted by an NFA *of at most 2^{n-2} states. This upper bound is met by a binary two-sided ideal and by an all-sided ideal defined over an alphabet of size 2^{n-2}.*
The binary alphabet is always optimal.

Proof. Let A be an NFA for a right ideal L. By Proposition 12, we may assume that A has a unique final sink state. It follows that in the subset automaton of the NFA A, all final states are equivalent since they accept all the strings. Hence the subset automaton has at most $2^{n-1} + 1$ reachable and pairwise distinguishable states. By inverting the finality of all states, we get a DFA B for L^c. The DFA B has a dead state. After omitting the dead state, we get an NFA for L^c of at most 2^{n-1} states. To prove tightness, let $L = K \cdot a \cdot (a+b)^*$, where K is the language accepted by the binary $(n-1)$-state NFA shown in Fig. 14. Then L is a right ideal accepted by the n-state NFA N shown in Fig. 19. Let S be a subset of $\{0, 1, \ldots, n-2\}$. Then S is reachable and S^c is co-reachable in N, as shown in the beginning of this section. By Proposition 6, $\mathrm{nsc}(L^c) \geq 2^{n-1}$.

Next, consider left ideal languages. Let A be an n-state NFA for a left ideal L. We can add a loop in the initial state on every input symbol to get an NFA N which is equivalent to A. In the subset automaton of N, every reachable subset contains the initial state of N, so the subset automaton has at most 2^{n-1} reachable states. This gives the upper bound. To prove tightness, let $L = (a+b)^* \cdot a \cdot K$ where K is the language accepted by the binary $(n-1)$-state NFA shown in Fig. 14. Then L is a left ideal accepted by the n-state NFA N shown in Fig. 20. For each subset S of $\{1, 2, \ldots, n-1\}$, the set S is co-reachable and the set S^c reachable in N, so by Proposition 6, $\mathrm{nsc}(L^c) \geq 2^{n-1}$.

We continue with two-sided ideal languages. Let A be an n-state NFA for a two-sided ideal L. We can add a loop on every symbol in the initial state of A and also in every final state of A to get an equivalent n-state NFA N for L. In the subset automaton of N, every reachable subset contains the initial state of N, and all final subsets are equivalent and accept all the strings. Hence L is accepted by a DFA of at most $2^{n-2} + 1$ states which contains a final sink state. It follows that L^c is accepted by an NFA of at most 2^{n-2} states. For tightness, let $L = (a+b)^* a \cdot K \cdot a(a+b)^*$ where K is the language accepted by the binary $(n-2)$-state NFA shown in Fig. 14. Then L is a two-sided ideal accepted by the n-state NFA N shown in Fig. 21. Let $S \subseteq \{1, 2, \ldots, n-2\}$. Then the set $\{0\} \cup S$ is reachable and its complement is co-reachable in N, so $\mathrm{nsc}(L^c) \geq 2^{n-2}$ by Proposition 6.

The binary alphabet is optimal in all three cases since the nondeterministic complexity of complementation on unary ideals is $n-1$ by Theorem 14 (2).

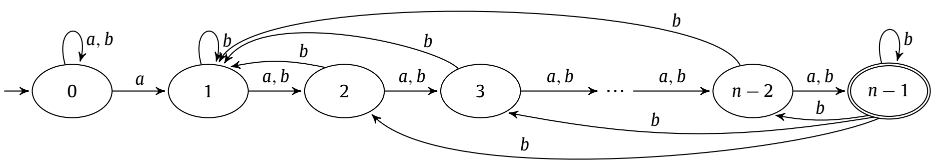

Fig. 20. A binary left ideal witness for complementation meeting the upper bound 2^{n-1}.

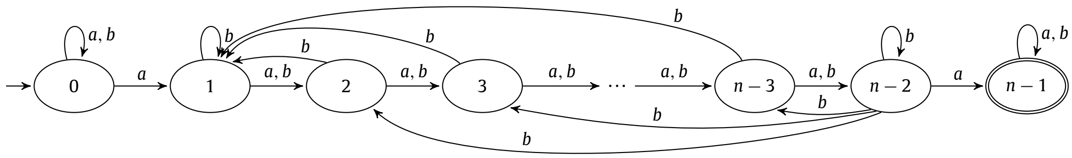

Fig. 21. A binary two-sided ideal witness for complementation meeting the upper bound 2^{n-2}.

Finally, we prove that the upper bound 2^{n-2} is met by an all-sided ideal defined over a growing alphabet of size 2^{n-2}. Let $\Sigma = \{a_S \mid S \subseteq \{1, 2, \dots, n-2\}\}$. Consider the language L accepted by the NFA $A = (\{0, 1, \dots, n-1\}, \Sigma, \cdot, 0, \{n-1\})$ where for each symbol a_S, we have

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

Since in each state of A, we have a loop on every input symbol, the language L is an all-sided ideal. Since the set of pairs $\{(a_S, a_S) \mid S \subseteq \{1, 2, \dots, n-2\}\}$ is a fooling set for L^c, we have $\text{nsc}(L^c) = 2^{n-2}$. $\square$

Notice that on suffix-free, left ideal, and suffix-closed languages, the complexity of complementation is 2^{n-1}, 2^{n-1}, and $2^{n-1} + 1$, respectively. Therefore, every language meeting the upper bound 2^n for complementation on suffix-convex languages must be so-called *proper suffix-convex*, that is, it can be neither suffix-free, nor left ideal, nor suffix-closed.

11. Conclusions

We investigated the nondeterministic state complexity of concatenation, intersection, union, reversal, star, and complementation in the subclasses of convex languages. We considered the classes of prefix-, suffix-, factor-, and subword-free, -closed, and -convex languages, and the classes of right, left, two-sided, and all-sided ideals. All our results are summarized in Table 1 which also compares them to the known results for regular and unary regular languages. The table also displays the size of alphabets used for describing witness languages.

The results in the unary case are given by Theorem 14. They imply that a binary alphabet used to describe witnesses is always optimal in the sense that no unary languages can meet the corresponding upper bounds.

The complexity of concatenation on free and ideal languages is $m + n - 1$ by Theorem 16 where we described unary witnesses. A ternary subword-closed, so also subword-convex witness meeting the upper bound $m + n$ has been described in Theorem 15. We do not know whether this upper bound can be met by binary subword-convex languages. The tight upper bounds on the complexity of intersection on free languages have been established in Theorem 19 and except for the subword-free case, in which the witness is defined over a growing alphabet of size $m + n - 5$, all our remaining witnesses are binary. Moreover, in Proposition 20, we described a binary subword-free language meeting the upper bound in $\Omega(\min\{m, n\}^2)$. A binary all-sided ideal and a binary subword-closed language meeting the upper bound mn for intersection have been described in Theorem 17. The results for union on free and ideal languages have been obtained in Theorem 23, and a binary subword-closed, so also subword-convex, language meeting the upper bound $m + n + 1$ has been described in Theorem 22.

The complexity of the reversal operation is $n + 1$ on the classes of suffix-free, left ideal, and closed languages by Theorem 24 where all witnesses are binary, and it is n on all the remaining classes by Proposition 25 with unary witnesses. Binary all-sided ideal, so also subword-convex, language meeting the upper bound $n + 1$ for star has been given in Proposition 26. On prefix- and suffix-free and -closed languages, the complexity of star is n with binary witnesses by Theorem 27, while it is $n - 1$ on factor-free languages with a unary subword-free witness, and it is one on factor-closed languages by Proposition 28. The results for complementation on free languages follow from Theorems 34, 35, and 36. Except for the subword-free case, the witnesses are ternary, and this size of alphabet is optimal, although, the complexity of complementation in the binary case is still exponential. Our subword-free witness is defined over an alphabet of size 2^{n-2} and we proved that the corresponding upper bound cannot be met by any language defined over a fixed alphabet. Theorem 37 provides the results for complementation on ideal languages. Our all-sided witness is defined over a growing alphabet of size 2^{n-2}, while all the remaining ideal witnesses are binary. The upper bound on the complexity of complementation on suffix-closed languages is $2^{n-1} + 1$, and it is met by a binary factor-closed language and a subword-closed language defined over an alphabet of size 2^n by Theorem 31. In Theorem 30, we described a binary prefix-closed, so also prefix-convex, and a (proper) suffix-convex language over a five-letter alphabet, both meeting the upper bound 2^n for complementation.

Hence, we found the exact complexity of each operation in each of the considered classes, except for complementation on factor- and subword-convex languages, where a lower bound is $2^{n-1} + 1$ and a trivial upper bound is 2^n. The exact complexity of complementation in these two classes remains open, and we conjecture that the upper bound 2^n cannot be met by any factor-convex language.

Table 1
The nondeterministic state complexity of operations in subclasses of convex languages. The dot means that the complexity is the same as in the above row with the same witness. The results for regular and unary regular languages are from [5,25].

Class \ Operation	KL	$\|\Sigma\|$	$K \cap L$	$\|\Sigma\|$	$K \cup L$	$\|\Sigma\|$
Prefix-free	$m+n-1$	1	$mn-(m+n-2)$	2	$m+n$	2
Suffix-free	·	1	$mn-(m+n-2)$	2	$m+n-1$	2
Factor-free	·	1	$mn-2(m+n-3)$	2	$m+n-2$	2
Subword-free	·	1	$mn-2(m+n-3)$	$m+n-5$	·	2
Unary free	$m+n-1$		$\min\{m,n\}$		$\max\{m,n\}$	
Right ideal	$m+n-1$	1	mn	2	$m+n$	2
Left ideal	·	1	·	2	$m+n-1$	2
Two-sided ideal	·	1	·	2	$m+n-2$	2
All-sided ideal	·	1	·	2	·	2
Unary ideal	$m+n-1$		$\max\{m,n\}$		$\min\{m,n\}$	
Prefix-closed	$m+n$	3	mn	2	$m+n+1$	2
Suffix-closed	·	3	·	2	·	2
Factor-closed	·	3	·	2	·	2
Subword-closed	·	3	·	2	·	2
Unary closed	$m+n-1$		$\min\{m,n\}$		$\max\{m,n\}$	
Prefix-convex	$m+n$	3	mn	2	$m+n+1$	2
Suffix-convex	·	3	·	2	·	2
Factor-convex	·	3	·	2	·	2
Subword-convex	·	3	·	2	·	2
Unary convex	$m+n-1$		$\max\{m,n\}$		$\max\{m,n\}$	
Regular	$m+n$	2	mn	2	$m+n+1$	2
Unary regular	$\geq m+n-1$		mn; $\gcd(m,n)=1$		$m+n+1$; $\gcd(m,n)=1$	

Class \ Operation	L^R	$\|\Sigma\|$	L^*	$\|\Sigma\|$	L^c	$\|\Sigma\|$
Prefix-free	n	1	n	2	2^{n-1}	3
Suffix-free	$n+1$	2	n	2	2^{n-1}	3
Factor-free	n	1	$n-1$	1	$2^{n-2}+1$	3
Subword-free	·	1	·	1	$2^{n-2}+1$	2^{n-2}
Unary free	n		$n-1$		$\Theta(\sqrt{n})$	
Right ideal	n	1	$n+1$	2	2^{n-1}	2
Left ideal	$n+1$	2	·	2	2^{n-1}	2
Two-sided ideal	n	1	·	2	2^{n-2}	2
All-sided ideal	·	1	·	2	2^{n-2}	2^{n-2}
Unary ideal	n		$n-1$		$n-1$	
Prefix-closed	$n+1$	2	n	2	2^n	2
Suffix-closed	·	2	n	2	$2^{n-1}+1$	2
Factor-closed	·	2	1	1	·	2
Subword-closed	$n+1$	$2n-2$	·	1	$2^{n-1}+1$	2^n
Unary closed	n		1		$n+1$	
Prefix-convex	$n+1$	2	$n+1$	2	2^n	2
Suffix-convex	·	2	·	2	2^n	5
Factor-convex	·	2	·	2	$\geq 2^{n-1}+1$	2
Subword-convex	$n+1$	$2n-2$	·	2	$\geq 2^{n-1}+1$	2^n
Unary convex	n		$n-1$		$n+1$	
Regular	$n+1$	2	$n+1$	1	2^n	2
Unary regular	n		$n+1$		$2^{\Theta(\sqrt{n \log n})}$	

Acknowledgement

We would like to thank Jozef Jirásek Jr. for his help with finding the suffix-convex witness for complementation.

References

[1] J. Hromkovič, Descriptional complexity of finite automata: concepts and open problems, J. Autom. Lang. Comb. 7 (2002) 519–531.
[2] Y. Gao, N. Moreira, R. Reis, S. Yu, A survey on operational state complexity, J. Autom. Lang. Comb. 21 (4) (2017) 251–310, https://doi.org/10.25596/jalc-2016-251.
[3] S. Yu, State complexity: recent results and open problems, Fund. Inform. 64 (1–4) (2005) 471–480, http://content.iospress.com/articles/fundamenta-informaticae/fi64-1-4-39.
[4] A.N. Maslov, Estimates of the number of states of finite automata, Sov. Math., Dokl. 11 (1970) 1373–1375.
[5] M. Holzer, M. Kutrib, Nondeterministic descriptional complexity of regular languages, Internat. J. Found. Comput. Sci. 14 (6) (2003) 1087–1102, https://doi.org/10.1142/S0129054103002199.
[6] S. Yu, Q. Zhuang, K. Salomaa, The state complexities of some basic operations on regular languages, Theoret. Comput. Sci. 125 (2) (1994) 315–328, https://doi.org/10.1016/0304-3975(92)00011-F.
[7] J. Dassow, R. Stiebe, Nonterminal complexity of some operations on context-free languages, Fund. Inform. 83 (1–2) (2008) 35–49, http://content.iospress.com/articles/fundamenta-informaticae/fi83-1-2-05.

[8] G. Pighizzini, J. Shallit, M. Wang, Unary context-free grammars and pushdown automata, descriptional complexity and auxiliary space lower bounds, J. Comput. System Sci. 65 (2) (2002) 393–414, https://doi.org/10.1006/jcss.2002.1855.
[9] M.O. Rabin, D. Scott, Finite automata and their decision problems, IBM J. Res. Develop. 3 (1959) 114–125, https://doi.org/10.1147/rd.32.0114.
[10] Yu.L. Yershov, On a conjecture of V. A. Uspenskii, Algebra Logika 1 (1962) 45–48 (in Russian).
[11] M. Domaratzki, State complexity of proportional removals, J. Autom. Lang. Comb. 7 (4) (2002) 455–468.
[12] C. Câmpeanu, K. Salomaa, S. Yu, Tight lower bound for the state complexity of shuffle of regular languages, J. Autom. Lang. Comb. 7 (3) (2002) 303–310.
[13] G. Jirásková, A. Okhotin, State complexity of cyclic shift, ITA 42 (2) (2008) 335–360, https://doi.org/10.1051/ita:2007038.
[14] Y.S. Han, K. Salomaa, D. Wood, Nondeterministic state complexity of basic operations for prefix-free regular languages, Fund. Inform. 90 (1–2) (2009) 93–106, https://doi.org/10.3233/FI-2009-0008.
[15] Y.S. Han, K. Salomaa, Nondeterministic state complexity for suffix-free regular languages, in: [49], 2010, pp. 189–196, https://doi.org/10.4204/EPTCS.31.21.
[16] R. Cmorik, G. Jirásková, Basic operations on binary suffix-free languages, in: Z. Kotásek, et al. (Eds.), MEMICS 2011, in: LNCS, vol. 7119, Springer, 2011, pp. 94–102.
[17] G. Jirásková, M. Krausová, Complexity in prefix-free regular languages, in: [49], 2010, pp. 197–204, https://doi.org/10.4204/EPTCS.31.22.
[18] J.A. Brzozowski, G. Jirásková, B. Li, Quotient complexity of ideal languages, Theoret. Comput. Sci. 470 (2013) 36–52, https://doi.org/10.1016/j.tcs.2012.10.055.
[19] J.A. Brzozowski, G. Jirásková, C. Zou, Quotient complexity of closed languages, Theory Comput. Syst. 54 (2) (2014) 277–292, https://doi.org/10.1007/s00224-013-9515-7.
[20] J. Brzozowski, G. Jirásková, B. Li, J. Smith, Quotient complexity of bifix, factor, and subword-free regular languages, Acta Cybernet. 21 (4) (2014) 507–527, https://doi.org/10.14232/actacyb.21.4.2014.1.
[21] J.A. Brzozowski, Complexity in convex languages, in: A. Dediu, H. Fernau, C. Martín-Vide (Eds.), LATA 2010, in: LNCS, vol. 6031, Springer, 2010, pp. 1–15.
[22] J.A. Brzozowski, C. Sinnamon, Complexity of proper prefix-convex regular languages, in: [50], 2017, pp. 52–63, https://doi.org/10.1007/978-3-319-60134-2_5.
[23] J.A. Brzozowski, B. Liu, Quotient complexity of star-free languages, Internat. J. Found. Comput. Sci. 23 (6) (2012) 1261–1276, https://doi.org/10.1142/S0129054112400515.
[24] G. Jirásková, T. Masopust, Complexity in union-free regular languages, Internat. J. Found. Comput. Sci. 22 (7) (2011) 1639–1653, https://doi.org/10.1142/S0129054111008933.
[25] G. Jirásková, State complexity of some operations on binary regular languages, Theoret. Comput. Sci. 330 (2) (2005) 287–298, https://doi.org/10.1016/j.tcs.2004.04.011.
[26] M. Holzer, M. Kutrib, K. Meckel, Nondeterministic state complexity of star-free languages, Theoret. Comput. Sci. 450 (2012) 68–80, https://doi.org/10.1016/j.tcs.2012.04.028.
[27] W.J. Sakoda, M. Sipser, Nondeterminism and the size of two way finite automata, in: R.J. Lipton, W.A. Burkhard, W.J. Savitch, E.P. Friedman, A.V. Aho (Eds.), Proceedings of the 10th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, May 1–3, 1978, ACM, 1978, pp. 275–286.
[28] J.C. Birget, Partial orders on words, minimal elements of regular languages and state complexity, Theoret. Comput. Sci. 119 (2) (1993) 267–291, https://doi.org/10.1016/0304-3975(93)90160-U.
[29] J.C. Birget, Intersection and union of regular languages and state complexity, Inform. Process. Lett. 43 (4) (1992) 185–190, https://doi.org/10.1016/0020-0190(92)90198-5.
[30] G. Jirásková, Note on minimal automata and uniform communication protocols, in: C. Martín-Vide, V. Mitrana (Eds.), Grammars and Automata for String Processing: From Mathematics and Computer Science to Biology, and Back: Essays in Honour of Gheorghe Paun, in: Topics in Computer Mathematics, vol. 9, Taylor and Francis, 2003, pp. 163–170.
[31] M. Hospodár, G. Jirásková, P. Mlynárčik, A survey on fooling sets as effective tools for lower bounds on nondeterministic complexity, in: H. Böckenhauer, D. Komm, W. Unger (Eds.), Adventures Between Lower Bounds and Higher Altitudes — Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday, in: Lecture Notes in Computer Science, vol. 11011, Springer, 2018, pp. 17–32.
[32] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 1979.
[33] M. Sipser, Introduction to the Theory of Computation, Cengage Learning, Boston, 2012.
[34] S. Yu, Regular languages, Chapter 2, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages – Vol. I, Springer, Heidelberg, 1997, pp. 41–110.
[35] I. Glaister, J. Shallit, A lower bound technique for the size of nondeterministic finite automata, Inform. Process. Lett. 59 (2) (1996) 75–77, https://doi.org/10.1016/0020-0190(96)00095-6.
[36] M. Hospodár, G. Jirásková, P. Mlynárčik, Nondeterministic complexity of operations on free and convex languages, in: [50], 2017, pp. 138–150, https://doi.org/10.1007/978-3-319-60134-2_12.
[37] G. Jirásková, P. Mlynárčik, Complement on prefix-free, suffix-free, and non-returning NFA languages, in: H. Jürgensen, J. Karhumäki, A. Okhotin (Eds.), DCFS 2014, in: LNCS, vol. 8614, Springer, 2014, pp. 222–233.
[38] M. Hospodár, G. Jirásková, P. Mlynárčik, Nondeterministic complexity of operations on closed and ideal languages, in: Y.S. Han, K. Salomaa (Eds.), CIAA 2016, in: LNCS, vol. 9705, Springer, 2016, pp. 125–137.
[39] J. Kao, N. Rampersad, J. Shallit, On NFAs where all states are final, initial, or both, Theoret. Comput. Sci. 410 (47–49) (2009) 5010–5021, https://doi.org/10.1016/j.tcs.2009.07.049.
[40] M. Chrobak, Finite automata and unary languages, Theoret. Comput. Sci. 47 (1986) 149–158, Corrigendum: Theoret. Comput. Sci. 302 (2003) 497–498.
[41] A.W. To, Unary finite automata vs. arithmetic progressions, Inform. Process. Lett. 109 (17) (2009) 1010–1014, https://doi.org/10.1016/j.ipl.2009.06.005.
[42] E. Bach, J. Shallit, Algorithmic Number Theory, Vol. 1: Efficient Algorithms, MIT Press, Cambridge, MA, 1996.
[43] G. Jirásková, P. Olejár, State complexity of intersection and union of suffix-free languages and descriptional complexity, in: H. Bordihn, et al. (Eds.), NCMA 2009, vol. 256, Austrian Computer Society, 2009, pp. 151–166.
[44] K. Čevorová, G. Jirásková, P. Mlynárčik, M. Palmovský, J. Šebej, Operations on automata with all states final, in: Z. Ésik, Z. Fülöp (Eds.), AFL 2014, in: EPTCS, vol. 151, 2014, pp. 201–215.
[45] P. Mlynárčik, Complement on free and ideal languages, in: DCFS 2015, 2015, pp. 185–196.
[46] L.H. Haines, On free monoids partially ordered by embedding, J. Combin. Theory 6 (1) (1969) 94–98.
[47] H. Bordihn, M. Holzer, M. Kutrib, Determination of finite automata accepting subregular languages, Theoret. Comput. Sci. 410 (2009) 3209–3222.
[48] K. Salomaa, S. Yu, NFA to DFA transformation for finite languages over arbitrary alphabets, J. Autom. Lang. Comb. 2 (3) (1997) 177–186.
[49] I. McQuillan, G. Pighizzini (Eds.), DCFS 2010, EPTCS, vol. 31, 2010.
[50] A. Carayol, C. Nicaud (Eds.), Implementation and Application of Automata — 22nd International Conference, CIAA 2017, Marne-la-Vallée, France, June 27–30, 2017, Proceedings, Lecture Notes in Computer Science, vol. 10329, Springer, 2017.

Please cite this article in press as: M. Hospodár et al., Nondeterministic complexity in subclasses of convex languages, Theoret. Comput. Sci. (2019), https://doi.org/10.1016/j.tcs.2018.12.027