NONDETERMINISTIC COMPLEXITY OF POWER AND POSITIVE CLOSURE ON SUBCLASSES OF CONVEX LANGUAGES

Michal Hospodár Matúš Palmovský

Mathematical Institute, Slovak Academy of Sciences,
Grešákova 6, 040 01 Košice, Slovakia
hosmich@gmail.com matp93@gmail.com

Abstract
We study the nondeterministic state complexity of the k-th power and positive closure 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 languages. We show that the upper bound kn on the complexity of the k-th power in the class of regular languages is tight for closed and convex classes, while in the remaining classes, the tight upper bound is $k(n-1)+1$. Next we show that the upper bound n on the complexity of the positive closure operation in the class of regular languages is tight in all considered classes except for classes of factor-closed and subword-closed languages, where the complexity is one. All our worst-case examples are described over a unary or binary alphabet, except for witnesses for the k-th power on subword-closed and subword-convex languages which are described over a ternary alphabet. Moreover, whenever a binary alphabet is used for describing a worst-case example, it is optimal in the sense that the corresponding upper bounds cannot be met by a language over a unary alphabet. The most interesting result is the description of a binary factor-closed language meeting the upper bound kn for the k-th power. To get this result, we use a method which enables us to avoid tedious descriptions of fooling sets.

1. Introduction

The nondeterministic state complexity of a regular language is the smallest number of states in any nondeterministic finite automaton (with a unique initial state) recognizing this language. The nondeterministic state complexity of a regular operation is the number of states that are sufficient and necessary in the worst case to accept the language resulting from this operation, considered as a function of the nondeterministic state complexities of the operands.

The nondeterministic state complexity of basic regular operations such as union, intersection, concatenation, and positive closure, has been investigated by Holzer and Kutrib [8].

Research supported by VEGA grant 2/0084/15 and grant APVV-15-0091.

The binary witnesses for complementation and reversal were described by Jirásková [12]. The k-th power operation on nondeterministic automata was studied by Domaratzki and Okhotin [5]. The nondeterministic state complexity of operations on prefix-free and suffix-free languages was examined by Han et al. [6, 7] and by Jirásková et al. [13, 15]. The results of these papers were improved and new results on nondeterministic complexity were obtained in a series of papers by Mlynárčik et al. In [14], complementation on prefix-free, suffix-free, and non-returning languages was investigated. Complementation on factor-free, subword-free, and ideal languages was considered in [17], basic operations (intersection, union, concatenation, Kleene star, reversal, complementation) on closed and ideal languages in [10], and basic operations on free and convex languages in [11]. Let us mention that the (deterministic) state complexity of basic operations on all above mentioned classes were considered by Brzozowski et al. [2, 3, 4].

In this paper, we investigate the nondeterministic state complexity of the k-th power and positive closure operations on subclasses of convex languages. For both operations and all considered subclasses, we provide a tight upper bound on its nondeterministic state complexity. Except for two cases in which our witnesses are ternary, all the witnesses are described over a binary or unary alphabet. Moreover, whenever a binary alphabet is used, it is always optimal in the sense that the corresponding upper bound cannot be met by any unary language.

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 [9, 19, 20]. Let Σ be a finite non-empty alphabet of symbols. Then Σ^* denotes the set of strings over the alphabet Σ including the empty string ε. A language is any subset of Σ^*. The *concatenation* of two languages K and L is the language $KL = \{uv \mid u \in K \text{ and } v \in L\}$. The *$k$-th power* of a language L is the language $L^k = LL^{k-1}$ where $L^0 = \{\varepsilon\}$. The *Kleene star* of a language L is the language $L^* = \bigcup_{i\geq 0} L^i$. The *positive closure* of a language L is the language $L^+ = \bigcup_{i\geq 1} L^i$.

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*, $s \in Q$ is the *initial* state, $F \subseteq Q$ is the set of *final* (or *accepting*) states, and $\cdot\colon Q \times \Sigma \to 2^Q$ is the *transition function* which can be extended to the domain $2^Q \times \Sigma^*$ in the natural way.

The *language accepted* (or *recognized*) by the NFA A is defined as $L(A) = \{w \in \Sigma^* \mid s \cdot w \cap F \neq \emptyset\}$. An NFA is a (partial) *deterministic finite automaton* (DFA) if $|q \cdot a| \leq 1$ for each q in Q and each a in Σ.

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.

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* NFA $A = (Q, \Sigma, \cdot, s, F)$ is the NFA $A^R = (Q, \Sigma, \cdot^R, F, \{s\})$ with possibly multiple initial states 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 $A = (Q, \Sigma, \cdot, s, F)$ be an NFA and $X, Y \subseteq Q$. We say that X is *reachable* in A if there is a string w in Σ^* such that $X = s \cdot w$. Next, we say that Y is *co-reachable* in A if Y is reachable in A^R.

The nondeterministic state complexity of a regular language L, denoted $\text{nsc}(L)$, is the smallest number of states in any NFA for L. To provide lower bounds on nondeterministic state complexity, we use the fooling set method described below.

Definition 2.1 *A set of pairs of strings* $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$ *is called a* fooling set for a language L *if for each* i, j *in* $\{1, 2, \ldots, n\}$, $x_i y_i \in L$, *and if* $i \neq j$, *then* $x_i y_j \notin L$ *or* $x_j y_i \notin L$.

Lemma 2.2 (cf. [1, Lemma 1]) *Let* $\mathcal{F}$ *be a fooling set for a regular language* L. *Then every* NFA *for* L *has at least* $|\mathcal{F}|$ *states.*

The next lemma provides a useful way to prove the minimality of a given NFA.

Lemma 2.3 *Let* $n \geq 2$. *Let* A *be an* NFA *with the state set* $Q = \{1, 2, \ldots, n\}$ *and let* $\{(X_i, Y_i) \mid i \in Q\}$ *be a set of pairs of subsets of* Q *such that for each* i *in* Q

(1) X_i *is reachable and* Y_i *is co-reachable in* A,
(2) $i \in X_i \cap Y_i$, *and*
(3) $X_i \subseteq \{i, i+1, \ldots, n\}$ *and* $Y_i \subseteq \{1, 2, \ldots, i\}$.

Then every NFA *for* $L(A)$ *has at least* n *states.*

Proof. Since X_i is reachable, there is a string x_i which sends the initial state of A to the set X_i. Since Y_i is co-reachable, there is a string y_i which is accepted by A from every state in Y_i and rejected from every other state. Since $X_i \cap Y_i = \{i\}$, the string $x_i y_i$ is in $L(A)$. Let $i \neq j$. Without loss of generality, we have $i > j$. Then $X_i \cap Y_j = \emptyset$, so $x_i y_j$ is not in $L(A)$. It follows by Definition 2.1 that the set $\{(x_i, y_i) \mid i \in Q\}$ is a fooling set for $L(A)$, so every NFA for $L(A)$ has at least n states by Lemma 2.2. □

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 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.

It is known that if a language is prefix-free, then every minimal NFA for it is non-exiting [18, Proposition 4.2], and if a language is suffix-free, then every minimal NFA for it is non-returning [18, Proposition 4.3]. Next, if a language is a right (left) ideal, then it is accepted by a minimal NFA such that its unique final (initial) state has a loop on each symbol and no other out-transitions (in-transitions) [10, Proposition 12], [18, Proposition 6.1]. Finally, an NFA with all states final accepts a prefix-closed language [18, Proposition 5.1], an NFA with all states initial accepts a suffix-closed language, and if a language is prefix-closed and suffix-closed, then it is factor-closed [18, Proposition 5.3].

3. Results

In this section, we examine the nondeterministic state complexity of the k-th power and positive closure on subclasses of convex languages. To get upper bounds, we use automata characterizations of languages in considered classes. To get lower bounds, we use the fooling set method given by Lemma 2.2 or, in the case of binary factor-closed languages, its simplification given by Lemma 2.3.

The nondeterministic state complexity of the k-th power on regular languages is kn if $k \geq 2$ and $n \geq 2$ [5, Theorem 3]. The next theorem shows that the complexity of the k-th power on all the classes of free, ideal, and unary convex languages is $k(n-1)+1$, while in all the remaining classes, it is kn. To describe a subword-closed witness, we use a ternary alphabet. All the remaining witnesses are described over binary or unary alphabets, and moreover, the binary alphabet is always optimal.

Theorem 3.1 (k-th Power) *Let $k \geq 2$ and $n \geq 2$. Let L be a language with* $\mathrm{nsc}(L) \leq n$.

(1) If L is prefix- or suffix-free, then $\mathrm{nsc}(L^k) \leq k(n-1)+1$*, and this upper bound is met by a unary subword-free language.*

(2) If L is right or left ideal, then $\mathrm{nsc}(L^k) \leq k(n-1)+1$*, and this upper bound is met by a unary all-sided ideal language.*

(3) If L is a unary convex language, then $\mathrm{nsc}(L^k) \leq k(n-1)+1$*, and this upper bound is met by a unary subword-closed language.*

(4) If L is prefix- or suffix-closed, then $\mathrm{nsc}(L^k) \leq kn$*, and this upper bound is met by a binary factor-closed language and by a ternary subword-closed language.*

Proof. (1) We may assume that a minimal NFA N for a prefix-free language L is non-exiting and has a unique final state. To get an NFA for L^k, we take k copies of N and we merge the final state in the j-th copy with the initial state in the $(j+1)$-th copy if $1 \leq j \leq k-1$. The initial state of the resulting NFA is the initial state in the first copy, and its final state is the final state in the k-th copy. If L is suffix-free, then we may assume that a minimal NFA N for L is non-returning. To get an NFA for L^k, we take k copies of N. For every symbol a and every final state p in the j-th copy with $1 \leq j \leq k-1$, we make the state p non-final and add the transitions (p,a,q) whenever there is a transition on a to q from the initial state in the $(j+1)$-th copy. Next, we omit the unreachable initial state in the $(j+1)$-th copy.

For tightness, let $L = \{a^{n-1}\}$. Then L is subword-free and it is accepted by an n-state NFA. We have $L^k = \{a^{k(n-1)}\}$ and the set $\{(a^i, a^{k(n-1)-i}) \mid 0 \le i \le k(n-1)\}$ is a fooling set for L^k of size $k(n-1)+1$. By Lemma 2.2, every NFA for L^k has at least $k(n-1)+1$ states. Hence $\mathrm{nsc}(L^k) = k(n-1)+1$.

(2) We may assume that a minimal NFA for a right ideal language L has a loop on every input symbol in its unique final state which has no other out-transitions. The construction of an NFA for L^k is the same as for prefix-free languages in case (1). Next, we may assume that a minimal NFA for a left ideal language L has a loop on every input symbol in its initial state which has no other in-transitions. The construction of an NFA for L^k is the same as for suffix-free languages in case (1), except that we add a loop on every symbol on p.

For tightness, let $L = \{a^i \mid i \ge n-1\}$. Then L is an all-sided ideal language and L is accepted by an n-state NFA. We have $L^k = \{a^i \mid i \ge k(n-1)\}$ and the same fooling set as in case (1) is a fooling set for L^k.

(3) Let L be a unary convex language accepted by a minimal n-state NFA. If L is infinite, then $L = \{a^i \mid i \ge n-1\}$, so $L^k = \{a^i \mid i \ge k(n-1)\}$ and $\mathrm{nsc}(L^k) = k(n-1)+1$. If L is finite, then the length of the longest string in L is $n-1$, so the length of the longest string in L^k is $k(n-1)$ and $\mathrm{nsc}(L^k) = k(n-1)+1$. This upper bound is met by the unary subword-closed language $\{a^i \mid 0 \le i \le n-1\}$.

(4) The upper bound is the same as in the general case of regular languages. Let us describe binary factor-closed and ternary subword-closed witnesses.

Let L be the language accepted by the NFA A shown in Figure 1. Since A has all states final

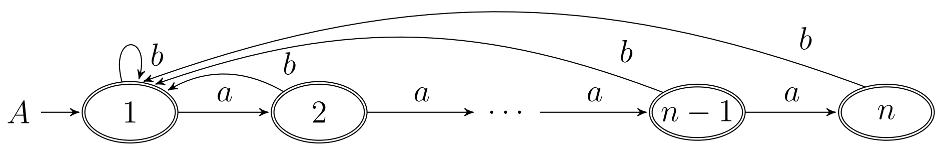

Figure 1: A binary factor-closed witness language meeting the upper bound kn for the k-th power.

and $L = L^R$, the language L is prefix-closed and suffix-closed, and therefore also factor-closed. The reader can verify that the language L^k is accepted by the kn-state partial DFA D consisting of k copies of A connected through the transitions on a going from the last state of the j-th copy to the second state of the $(j+1)$-th copy if $1 \le j \le k-1$.

For $i = 1, 2, \ldots, kn$, let $X_i = \{i\}$ and $Y_i = \{1, 2, \ldots, i\}$. Notice that

- each set X_i with $i \notin \{jn+1 \mid 1 \le j \le k-1\}$ is reachable in D by a word in a^*;
- each set X_i with $i \in \{jn+1 \mid 1 \le j \le k-1\}$ is reachable in D by a word in a^*b;
- each set Y_i with $i \notin \{jn \mid 1 \le j \le k-1\}$ is co-reachable in D since it is reachable in D^R by a word in a^*;
- each set Y_i with $i \in \{jn \mid 1 \le j \le k-1\}$ is co-reachable in D since it is reachable in D^R by a word in a^*b.

Moreover, $i \in X_i \cap Y_i$, $X_i \subseteq \{i, i+1, \ldots, kn\}$, and $Y_i \subseteq \{1, 2, \ldots, i\}$, so the sets X_i and Y_i satisfy the conditions of Lemma 2.3. Hence every NFA for L^k has at least kn states, which together with the upper bound gives $\operatorname{nsc}(L^k) = kn$.

Next, let $L = \{b^* a^i c^* \mid 0 \leq i \leq n-1\}$, which is accepted by an n-state NFA. Since every subword of a string $b^\ell a^i c^m$ in L is of the form $b^{\ell'} a^{i'} c^{m'}$ where $i' \leq i \leq n-1$, the language L is subword-closed. For each j with $1 \leq j \leq k$, consider the set of pairs of strings

$$\mathcal{F}_j = \{((ba^{n-1}c)^{j-1} ba^i, a^{n-1-i} c (ba^{n-1}c)^{k-j}) \mid 0 \leq i \leq n-1\}.$$

We have $(ba^{n-1}c)^k \in L^k$. Next, we have $L^k \subseteq (b^* a^* c^*)^k$, and moreover, no string with more than $k(n-1)$ occurrences of a is in L^k. It follows that the set $\bigcup_{j=1}^{k} \mathcal{F}_j$ is a fooling set for L^k of size kn, so every NFA for L^k has at least kn states by Lemma 2.2. □

Notice that in the theorem above, we must have $n \geq 2$ since for every positive integer k, the k-th power of a language accepted by a 1-state NFA is the same language. The theorem also shows that two symbols are necessary to meet the bound kn for the k-th power on closed languages.

Now we consider the operation of positive closure. The upper bound on nondeterministic state complexity of positive closure on regular languages is n [8, Theorem 9] since we can get an NFA for L^+ from an NFA for L by adding the transition (q, a, s) whenever there is a transition (q, a, f) for a final state f. The next theorem showsthat this upper bound is tight in all the classes of free and ideal, so also convex languages, and on the classes of prefix-closed and suffix-closed languages. It also proves that the positive closure of every factor-closed language is of complexity one.

Theorem 3.2 (Positive Closure) *Let n be a positive integer.*

(1) There exists a unary subword-free language L with $\operatorname{nsc}(L) \leq n$ *and* $\operatorname{nsc}(L^+) = n$.
(2) There exists a unary all-sided ideal language L with $\operatorname{nsc}(L) \leq n$ *and* $\operatorname{nsc}(L^+) = n$.
(3) There exists a binary prefix-closed language L with $\operatorname{nsc}(L) \leq n$ *and* $\operatorname{nsc}(L^+) = n$.
(4) There exists a binary suffix-closed language L with $\operatorname{nsc}(L) \leq n$ *and* $\operatorname{nsc}(L^+) = n$.
(5) If L is factor-closed, then $\operatorname{nsc}(L^+) = 1$.

Proof. (1) Let $L = \{a^{n-1}\}$, which is accepted by an n-state NFA. Then $L^+ = \{a^{k(n-1)} \mid k \geq 1\}$ and the set $\{(a^i, a^{n-1-i}) \mid 0 \leq i \leq n-1\}$ is a fooling set for L^+ of size n. By Lemma 2.2, every NFA for L^+ has at least n states. Hence $\operatorname{nsc}(L^+) = n$.

(2) Let $L = \{a^i \mid i \geq n-1\}$, which is accepted by an n-state NFA. We have $L^+ = L$ and the same set as above is a fooling set for L^+ of size n.

(3) Let L be the language accepted by the NFA shown in Figure 2. Notice that each state of this NFA is final, hence L is prefix-closed. Consider the set $\mathcal{F} = \{(a^i, a^{n-1-i}b) \mid 0 \leq i \leq n-1\}$. We have $a^i a^{n-1-i} b = a^{n-1} b$, which is in L^+. Let $0 \leq i < j \leq n-1$. Then $a^i a^{n-1-j} b$ is not in L^+. Hence the set $\mathcal{F}$ is a fooling set for L^+ of size n.

(4) Let L be the language accepted by the NFA shown in Figure 3. Notice that if we make all states of this NFA initial, then we get an equivalent finite automaton. Hence L is suffix-closed.

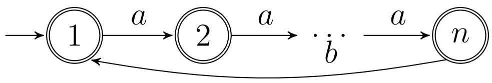

Figure 2: A binary prefix-closed witness language meeting the upper bound n for positive closure.

Moreover, $L^+ = L$ since the initial state is the unique final state. Consider the set of pairs of strings $\mathcal{F} = \{(ba^i, a^{n-1-i}) \mid 0 \le i \le n-1\}$. Since $ba^{n-1} \in L$, while ba^k with $k \le n-2$ is not in L, the set $\mathcal{F}$ is a fooling set for L, so also for L^+, of size n.

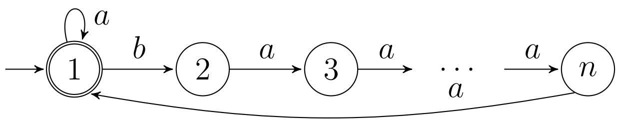

Figure 3: A binary suffix-closed witness language meeting the upper bound n for positive closure.

(5) 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 $L^+ = \Gamma^*$, which is accepted by a one-state NFA. □

Notice that two symbols are necessary to meet the bound n for positive closure on prefix- and suffix-closed languages since every unary prefix- or suffix-closed language is also factor-closed.

4. Conclusions

We investigated the nondeterministic state complexity of the k-th power and positive closure 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. We found the exact complexities of both operations in each of the above mentioned classes. For describing witness languages for the k-th power on subword-closed and subword-convex languages, we used a ternary alphabet. All the remaining witness languages are described over a binary or unary alphabet. Moreover, if a binary alphabet is used, it is optimal in the sense that the corresponding upper bound cannot be met by any unary language.

References

[1] J. BIRGET, Intersection and union of regular languages and state complexity. *Information Processing Letters* 43 (1992) 4, 185–190.

[2] J. BRZOZOWSKI, G. JIRÁSKOVÁ, B. LI, J. SMITH, Quotient complexity of bifix, factor, and subword-free regular languages. *Acta Cybernetica* 21 (2014) 4, 507–527.

[3] J. A. BRZOZOWSKI, G. JIRÁSKOVÁ, B. LI, Quotient complexity of ideal languages. *Theoretical Computer Science* 470 (2013), 36–52.

[4] J. A. BRZOZOWSKI, G. JIRÁSKOVÁ, C. ZOU, Quotient complexity of closed languages. *Theory of Computing Systems* 54 (2014) 2, 277–292.

[5] M. DOMARATZKI, A. OKHOTIN, State complexity of power. *Theoretical Computer Science* 410 (2009) 24-25, 2377–2392.

[6] Y. HAN, K. SALOMAA, Nondeterministic state complexity for suffix-free regular languages. In: MCQUILLAN and PIGHIZZINI [16], 2010, 189–196.

[7] Y. HAN, K. SALOMAA, D. WOOD, Nondeterministic state complexity of basic operations for prefix-free regular languages. *Fundamenta Informaticae* 90 (2009) 1–2, 93–106.

[8] M. HOLZER, M. KUTRIB, Nondeterministic descriptional complexity of regular languages. *International Journal of Foundations of Computer Science* 14 (2003) 6, 1087–1102.

[9] J. E. HOPCROFT, J. D. ULLMAN, *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[10] M. HOSPODÁR, G. JIRÁSKOVÁ, P. MLYNÁRČIK, Nondeterministic complexity of operations on closed and ideal languages. In: Y. HAN, K. SALOMAA (eds.), *Implementation and Application of Automata - 21st International Conference, CIAA 2016, Seoul, South Korea, July 19-22, 2016, Proceedings*. Lecture Notes in Computer Science 9705, Springer, 2016, 125–137.

[11] M. HOSPODÁR, G. JIRÁSKOVÁ, P. MLYNÁRČIK, Nondeterministic complexity of operations on free and convex languages. In: 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 10329, Springer, 2017, 138–150.

[12] G. JIRÁSKOVÁ, State complexity of some operations on binary regular languages. *Theoretical Computer Science* 330 (2005) 2, 287–298.

[13] G. JIRÁSKOVÁ, M. KRAUSOVÁ, Complexity in prefix-free regular languages. In: MCQUILLAN and PIGHIZZINI [16], 2010, 197–204.

[14] G. JIRÁSKOVÁ, P. MLYNÁRČIK, Complement on prefix-free, suffix-free, and non-returning NFA languages. In: H. JÜRGENSEN, J. KARHUMÄKI, A. OKHOTIN (eds.), *Descriptional Complexity of Formal Systems - 16th International Workshop, DCFS 2014, Turku, Finland, August 5-8, 2014. Proceedings*. Lecture Notes in Computer Science 8614, Springer, 2014, 222–233.

[15] G. JIRÁSKOVÁ, P. OLEJÁR, State complexity of intersection and union of suffix-free languages and descriptional complexity. In: H. BORDIHN, R. FREUND, M. HOLZER, M. KUTRIB, F. OTTO (eds.), *Workshop on Non-Classical Models for Automata and Applications (NCMA 2009)*. books@ocg.at 256, Österreichische Computer Gesellschaft, 2009, 151–166.

[16] I. MCQUILLAN, G. PIGHIZZINI (eds.), *Proceedings Twelfth Annual Workshop on Descriptional Complexity of Formal Systems, DCFS 2010, Saskatoon, Canada, 8-10th August 2010*. EPTCS 31, 2010.

[17] P. MLYNÁRČIK, Complement on free and ideal languages. In: J. SHALLIT, A. OKHOTIN (eds.), *Descriptional Complexity of Formal Systems – 17th International Workshop, DCFS 2015, Waterloo, ON, Canada, June 25–27, 2015. Proceedings*. Lecture Notes in Computer Science 9118, Springer, 2015, 185–196.

[18] P. MLYNÁRČIK, Nondeterministic state complexity in subregular classes. *Dissertation thesis.* Faculty of Mathematics, Physics and Informatics of the Comenius University, Bratislava (2017). `http://im.saske.sk/%7Ejiraskov/students/phd_thesis_mlynarcik.pdf`

[19] M. SIPSER, *Introduction to the Theory of Computation*. Cengage Learning, Boston, 2012.

[20] S. YU, Chapter 2: Regular Languages. In: G. ROZENBERG, A. SALOMAA (eds.), *Handbook of Formal Languages*. Vol. 1, Springer, Heidelberg, 1997, 41–110.