For submission to the Journal of Automata, Languages and Combinatorics
Created on January 2, 2018

A SURVEY ON OPERATIONAL STATE COMPLEXITY

Yuan Gao[(A)] Nelma Moreira[(B)] Rogério Reis[(B)] Sheng Yu[(A)]

[(A)] *Department of Computer Science, University of Western Ontario*
London, Ontario, Canada
gaoyuan012591@gmail.com syu@csd.uwo.ca

[(B)] *CMUP & DCC, Faculdade de Ciências da Universidade do Porto*
Rua do Campo Alegre, 4169-007 Porto, Portugal
{nam,rvr}@dcc.fc.up.pt

ABSTRACT

Descriptional complexity is the study of the conciseness of the various models representing formal languages. The state complexity of a regular language is the size, measured by the number of states of the smallest, either deterministic or nondeterministic, finite automaton that recognises it. Operational state complexity is the study of the state complexity of operations over languages. In this survey, we review the state complexities of individual regularity preserving language operations on regular and some subregular languages. Then we revisit the state complexities of the combination of individual operations. We also review methods of estimation and approximation of state complexity of more complex combined operations.

This is the author's version of the following article: Y. Gao, N. Moreira, R. Reis, S. Yu, A Survey on Operational State Complexity. *Journal of Automata, Languages and Combinatorics* **21** (2016) 4, 251–310.
URL `https://doi.org/10.25596/jalc-2016-251`.

1. Introduction

Automata theory is one of the oldest research areas in computer science. Much research has been done on automata theory since 1950's. Work in many subareas of automata theory is still ongoing these days due to its new applications in areas such as software engineering, programming languages, parallel programming, network security, formal verification and natural language and speech processing [164, 175, 166, 195, 151, 201].

Descriptional complexity and, in particular, state complexity is one of such active subareas. Generally speaking, the study of complexity mainly focuses on the following two kinds of issues: time and space complexity issues, i.e. time and space needed for the execution of the processes; or descriptional complexity issues, i.e. the succinctness of the model representations [203]. In general, having succinct objects will improve our control on software, which may become smaller, more efficient and easier to certify.

State complexity is a type of descriptional complexity based on the finite machine model, and, in the domain of regular languages, it is related to the basic question of how to measure the size of a finite automaton. For the deterministic finite automaton (DFA) case, the three usual answers are: the number of states, the number of transitions, or a combination of the two [203]. For a complete DFA, whose transition function is defined for every state and every possible input symbol, the number of transitions is linear with the number of states, for each fixed alphabet. Thus, the number of states becomes the key measure for the size of a complete DFA. When considering the descriptional complexity of nondeterministic finite automata (NFA), because this notion of completeness is not present, the measures based on the number of states and on the number transitions, are much more loosely related.

Since a regular language can be accepted by many DFAs with a different number of states but only by one unique minimal, complete DFA, the deterministic state complexity of a regular language is defined as the number of states of the minimal, complete DFA accepting it. If we replace the minimal, complete DFA with minimal NFA, we have the definition of nondeterministic state complexity. Since state complexity is used as a natural abbreviation of deterministic state complexity by most researchers working in the area, we also follow the convention in this paper.

Complexity can be studied in two different flavours: in the worst case [203] and in the average case [169]. The worst-case complexity of a class of regular languages is the supremum of the complexities of all the languages in the class [203] whereas the average-case complexity is the average value of the complexities of those languages. In spite of its evident practical importance, there is still very few research on average-case state complexity. For that reason, in this paper, we mainly review worst-case results.

Results on descriptional complexity can be, roughly, divided into representational (or transformational) and operational. Representational complexity studies the complexity of transformations between models, by comparing the sizes of different representations of formal languages [191]. For example, given an n-state NFA for a regular language, the DFA which is equivalent to it has at most 2^n states, and this result, established in 1957, is considered the first state complexity result [181]. Operational state complexity studies the state complexity of operations on languages. When we speak about the state complexity of an operation on regular languages, we mean the state complexity of the class of resulting languages from the operation [203]. For example, when we say that the state complexity of the intersection operation on two regular languages, accepted by m-state and n-state DFAs, respectively, is mn, we mean that mn is the worst-case state complexity of the class of regular languages that can be represented as the intersection of an m-state DFA language and an n-state DFA language. Note that this implies that the intersection of any m-state DFA language L_1 and n-state DFA L_2 language has a DFA with at most mn states (upper bound) and that there exist languages L_1 and L_2 such that the minimal DFA for $L_1 \cap L_2$ has exactly mn states (lower bound).

In this survey, we mostly concentrate in operational state complexity results. Although first studies go back to the 1960's and 1970's, research in the area has been most active in the last two decades. This can be partially explained by the fact that back then, descriptional complexity issues were not a priority for applications, as they are today. But, also, due to its combinatorial nature many of the current research is only possible with the help of new high-performance symbolic manipulation software and powerful computers [78].

The paper is organized as follows. After some preliminares in the next section, the notions of deterministic and nondeterministic state complexity are considered in Section 3. To better understand the possible gap between both measures is a main topic of research. In Section 4, we review the state complexities of individual regularity preserving language operations, like, Boolean operations, catenation, star, reversal, shuffle, orthogonal catenation, proportional removal, and cyclic shift, etc. These individual operations are fundamental and important in formal languages and automata theory research and applications. Results in these two sections are given for different classes of (sub)regular languages, e.g. general infinite, finite, unary, star-free, etc. In Section 5, we revisit the state complexities of combined operations which are combinations of individual operations, e.g., star of union, star of intersection, star of catenation, star of reversal, union of star, intersection of star, etc. The state complexities of most of these combined operations are much lower than the mathematical composition of the state complexities of their component individual operations. We also review the methods of estimation and approximation of state complexity of combined operations which can be used for very complex combined operations. Section 6 concludes this survey with some discussion on the results presented, highlighting some open problems and directions of future research.

2. Preliminaries

Here we recall some basic definitions related to finite automata and regular languages. For a more complete presentation the reader is referred to [202].

The set of natural numbers is denoted by $\mathbb{N}$ and for $i, j \in \mathbb{N}$, $[i, j] = \{x \in \mathbb{N} \mid i \leq x \leq j\}$. The power set of a set S is denoted by 2^S and the cardinality of a finite set S is $|S|$. In the following, Σ stands always for a finite alphabet, the empty word is represented by ε and the set of all words over Σ by $\Sigma^\star$. A language is a subset of

$\Sigma^\star$. We say that $L \subseteq \Sigma^\star$ is a unary (respectively, binary, ternary) language if $|\Sigma| = 1$ (respectively, $|\Sigma| = 2$, $|\Sigma| = 3$). Note this definition does not require that all symbols of Σ actually appear in words of L and hence every unary language is also a binary language and a binary language is always a ternary language. A language L is said to be finite if L is a finite subset of $\Sigma^\star$.

A *nondeterministic finite automaton* (NFA) is a tuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to 2^Q$ is the (multi-valued) transition function, $q_0 \in Q$ is the initial state and $F \subseteq Q$ is the set of final (accepting) states. The transition function is extended as a function $\widehat{\delta} : Q \times \Sigma^\star \to Q$ by setting $\widehat{\delta}(q, \varepsilon) = q$ for $q \in Q$ and for $w \in \Sigma^\star$, $x \in \Sigma$, $\widehat{\delta}(q, wx) = \delta(\widehat{\delta}(q, w), x)$. To simplify the notation, we denote $\widehat{\delta}$ by δ. The language recognized by the NFA A is $L(A) = \{w \in \Sigma^\star \mid \delta(q_0, w) \cap F \neq \emptyset\}$.

An NFA $A = (Q, \Sigma, \delta, q_0, F)$ is a *complete deterministic finite automaton* (DFA) if the transition function δ is one-valued, that is, δ is a function $Q \times \Sigma \to Q$. An *incomplete* DFA allows the possibility that some transitions may be undefined, that is, δ is a partial function $Q \times \Sigma \to Q$.

Both the DFAs and the NFAs define the class of regular languages [202]. It is well known that any regular language has a unique minimal (complete or incomplete) DFA, that is, a unique DFA with the smallest number of states. For a given regular language the sizes of the minimal, complete DFA and minimal, incomplete DFA differ by at most one state. Furthermore, for a given DFA there exists an $n \log n$ time algorithm to compute the minimal DFA [202]. On the other hand, for a given regular language there may be more than one minimal NFA and NFA minimization is PSPACE-hard [108, 202].

3. State Complexity and Nondeterministic State Complexity

The *state complexity* of a regular language L, $sc(L)$, is the number of states of its minimal (complete) DFA. The *nondeterministic state complexity* of a regular language L, $nsc(L)$, is the number of states of a minimal NFA that accepts L. Since a DFA is in particular an NFA, for any regular language L one has $sc(L) \leq nsc(L)$. It is well known that any m-state NFA can be converted, via the *subset construction*, into an equivalent DFA with at most 2^m states [181] (we call this conversion *determination*). Thus, $sc(L) \leq 2^{nsc(L)}$. To show that this upper bound is tight one must exhibit a family of languages $(L_m)_{m\geq 1}$ such that $nsc(L_m) = m$ and $sc(L_m) = 2^m$, for every $m \geq 1$. In 1963, Lupanov [152] showed that this upper bound is tight using a family of ternary languages. In 1971, Moore [165] and Meyer and Fischer [161] presented different families of binary languages. All three families of NFAs are represented in Figure 1. However, for unary languages that upper bound is not achievable [153, 52, 53]. Chrobak [52, 53] proved that if L is a unary language with $nsc(L) = m$, then $sc(L) = O(F(m))$ where

$$F(m) = \max\{\operatorname{lcm}(x_1, \ldots, x_l) \mid x_1, \ldots, x_l \geq 1 \text{ and } x_1 + \cdots + x_l = m\} \tag{1}$$

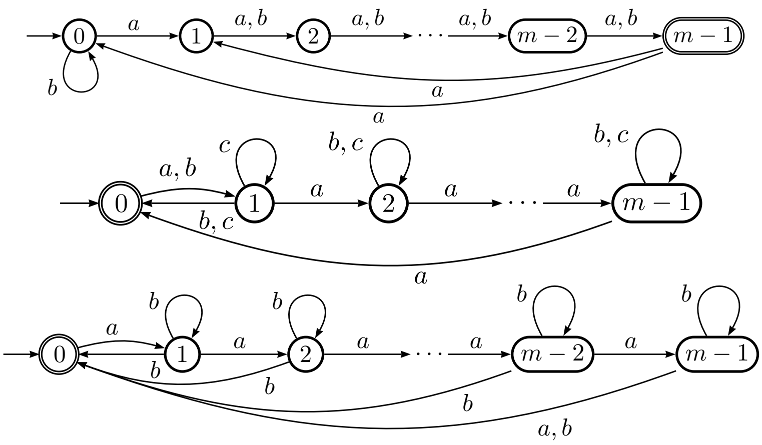

Figure 1: Moore (i), Lupanov (ii), and Meyer & Fischer (iii) minimal m-state NFAs with equivalent minimal 2^m-state DFAs

is the Landau's function and lcm denotes the least common multiple. It is known that $F(m) = e^{\Theta(\sqrt{m \ln m})}$, so $sc(L) = e^{\Theta(\sqrt{m \ln m})}$. This asymptotic bound is tight, i.e., for every m there exists a unary language L_m such that $nsc(L_m) \leq m$ and $sc(L_m) = F(m-1)$. Other related bounds were studied by Meregethi and Pighizzini [160].

For a general finite language L, if $nsc(L) = m$ then $sc(L) = \Theta(k^{\frac{m}{1+\log k}})$, $k = |\Sigma| > 1$, and this bound is tight [192]. In the case of finite binary languages, $\Theta(2^{\frac{m}{2}})$ is a tight bound. In 1973, Mandl [155] had already proved that, for any finite binary language L, if $nsc(L) = m$ then $sc(L) \leq 2 \cdot 2^{m/2} - 1$ if m is even, and $sc(L) \leq 3 \cdot 2^{\lfloor m/2 \rfloor} - 1$ if m is odd, and that these bounds are tight. Finally, for finite unary languages, nondeterminism does not lead to significant improvements. If L is a finite unary language with $nsc(L) = m$, then $sc(L) \leq m + 1$ [155, 192].

In Section 4.3 the state complexity of determination of other subregular languages is reviewed. As it will be evident from the results in the following sections, the complexity of determination plays a fundamental role in the operational complexity and thus the importance of its study *per se*.

The possible gap between state complexity and nondeterministic state complexity for general regular languages leads to the notion of *magic number* introduced in 2000 by Iwama *et al.* [115, 116]. A number α, such that $\alpha \in [m, 2^m]$, is *magic* for m with respect to a given alphabet of size k, if there is no minimal m-state NFA whose equivalent minimal DFA has α states. This notion has been extensively researched in the last decade and has been extended to other gaps between two state complexity values [153, 52, 121, 81, 82, 83, 120, 125, 127, 102]. We summarize here some of the obtained results. The general observation is that, apart from unary languages, magic numbers are hard to find. For binary languages, it was shown that if $\alpha = 2^m - 2^n$ or $\alpha = 2^m - 2^n - 1$, for $n \in [0, m/2 - 2]$ [115], and $\alpha = 2^m - n$ for $n \in [5, 2m - 2]$ and some coprimality condition holds for n [116], then α is not magic. Also, for a binary alphabet, all numbers $\alpha \in [m, m + 2^{\lfloor m/3 \rfloor}]$ have been shown to be non-magic [123], which improves previous results, $[m, m^2/2]$ [121] and

$[m, 2^{\sqrt[3]{m}}]$ [81]. For ternary or quaternary regular languages, and for languages over an alphabet of exponential growing size there are no magic numbers [121, 120, 125, 127]. For the unary case, however, trivially all numbers between $e^{(1+o(1))\sqrt{m \ln m}}$ and 2^m are magic [153, 52, 82]. Moreover, it has been shown that there are much more magic than non-magic numbers in the range from m to $e^{(1+o(1))\sqrt{m \ln m}}$ [82]. In the case of finite languages, partial results were obtained by Holzer *et al.* [102]. All numbers $\alpha \in [m+1, (\frac{m}{2})^2 + \frac{m}{2} + 1]$, if m even, and $\alpha \in [m+1, (\frac{m-1}{2})^2 + m + 1]$, if m is odd, are non-magic. Moreover, all numbers of the form $3 \cdot 2^{\frac{m}{2}-1} + 2^i - 1$, with m even, and $2^{\frac{m+1}{2}} + 2^i - 1$, with m odd, for some integer $i \in [1, \lceil \frac{m-1}{2} \rceil]$ are non-magic. In the same paper, the magic number problem is also studied for other subregular language classes.

3.1. State Complexity versus Quotient Complexity

Quotient complexity, introduced in 2009 by Brzozowski [14, 16], coincides, for regular languages, with the notion of state complexity but it is defined in terms of languages and their (left) quotients. The *left quotient* of a language L by a word w is defined as the language $w^{-1}L = \{x \in \Sigma^\star \mid wx \in L\}$. The *quotient complexity* of L, denoted by $\kappa(L)$, is the number of distinct languages that are left quotients of L by some word. It is well known that, for a regular language L, the number of left quotients is finite and is exactly the number of states of the minimal DFA accepting L. So, in the case of regular languages, state complexity and quotient complexity coincide. Considering that quotient complexity is given in terms of languages, and their left quotients, some language algebraic properties can be used in order to obtain upper bounds for the complexity of operations over languages. Actually, the proof that the set of (left) quotients of a regular language is finite [13] was one of the earliest studies of state complexity. Quotient complexity can also be useful to show that an upper bound is tight. If a given operation can be expressed as a function of other operations (for example, $L_1 - L_2 = L_1 \cap \overline{L_2}$), then, witnesses for the worst-case complexity of those operations can be used to provide a witness for the complexity of the first operation.

4. State Complexity of Individual Operations

The *state complexity of an operation* (or *operational state complexity*) on regular languages is the worst-case state complexity of a language resulting from the operation, considered as a function of the state complexities of the operands. Adapting a formulation from Holzer and Kutrib [107], given a binary operation $\circ$, the $\circ$-*language operation state complexity problem* can be stated as follows:

- Given an m-state DFA A_1 and an n-state DFA A_2.
- How many states are sufficient and necessary, in the worst case, to accept the language $L(A_1) \circ L(A_2)$ by a DFA?

This formulation can be generalized for operations with other arities, other kinds of automata and classes of languages. An upper bound can be obtained by providing an algorithm that, given DFAs for the operands, constructs a DFA that accepts the resulting language. The number of states of the resulting DFA is an upper bound for the state complexity of the referred operation. To show that an upper bound is tight, for each operand a family of languages (one language, for each possible value of the state complexity) must be given such that the resulting automata achieve that bound. We can call those families *witnesses*. The same approach is used to obtain the nondeterministic state complexity of an operation on regular languages. No proofs are here presented for the stated results, although several examples of families of languages, for which the operations achieve a certain upper bound, are given. There are very few results of the study of state complexity on the average case. However, whenever some results are known they are mentioned together with the corresponding worst-case analysis.

In this section, the following notation is used. When considering unary operations, let L be regular language with $sc(L) = m$ ($nsc(L) = m$) and let $A = (Q, \Sigma, \delta, q_0, F)$ be the complete minimal DFA (a minimal NFA) such that $L = L(A)$. Furthermore, $|\Sigma| = k$ or $|\Sigma| = f(m)$ if a growing alphabet is taken into account, $|F| = f$, and $|F - \{q_0\}| = l$. In the same way, for binary operations let L_1 and L_2 be regular languages over the same alphabet with $sc(L) = m$ ($nsc(L) = m$) and $sc(L_2) = n$ ($nsc(L_2) = n$), and let $A_i = (Q_i, \Sigma, \delta_i, q_i, F_i)$ be complete minimal DFAs (minimal NFAs) such that $L_i = L(A_i)$, for $i \in [1, 2]$. Furthermore, $|\Sigma| = k$ or $|\Sigma| = f(m, n)$ if a growing alphabet is taken into account, $|F_i| = f_i$, and $|F_i - \{q_i\}| = l_i$, for $i \in [1, 2]$.

4.1. Basic Operations

In this section we review the main results related with state complexity (and nondeterministic state complexity) of some basic operations on regular languages: Boolean operations (mainly union, intersection, and complement), catenation, star (and plus), and reversal. For some classes of languages, left and right quotients are also considered. Because their particular characteristics, that were already pointed out in Section 3, for each operation the languages are divided into *regular* ($k \geq 2$ and infinite), *finite* ($k \geq 2$), *unary* (infinite) and *finite unary*. Some other subregular languages are considered in Section 4.3. Whenever known, results on the range of complexities that can be reached for each operation are also presented. This extension of the notion of magic number to operational state complexity is now an active topic of research.

There are some other survey papers that partially review the results here presented and that were a reference to our presentation [202, 203, 204, 114, 205, 190, 107, 14, 106, 108].

4.1.1. General Regular Languages

Table 1 summarizes the results for general regular languages. The (fifth) third column contains the smallest alphabet size of the witness languages for the (nondeterministic) state complexity given in the (fourth) second column, respectively. Columns with this kind of information also appear in several tables to follow.

In 1994, Yu *et al.* [208] studied the state complexity of catenation, star, reversal,

Table 1: State complexity and nondeterministic state complexity for basic operations on regular languages.

	Regular			
	sc	$\|\Sigma\|$	nsc	$\|\Sigma\|$
$L_1 \cup L_2$	mn	2	$m+n+1$	2
$L_1 \cap L_2$	mn	2	mn	2
$\overline{L}$	m	1	2^m	2
$(L_1 - L_2)$	mn	2		
$(L_1 \oplus L_2)$	mn	2		
L_1L_2	$m2^n - f_1 2^{n-1}$, if $n > 1$	2	$m+n$	2
	m, if $n = 1$	1		
$L^\star$	$2^{m-1} + 2^{m-l-1}$, if $m > 1$, $l > 0$	2	$m+1$	2
	m, if $m > 1$, $l = 0$	1		
	$m+1$, if $m = 1$	1		
L^+	$2^{m-1} + 2^{m-l-1} - 1$	2	m	2
L^R	2^m	2	$m+1$	2
$L_2 \setminus L_1$	$2^m - 1$	2		
L_1 / L_2	m	1		
$w^{-1}L$	m	1	$O(m+1)$	
Lw^{-1}	m	1	m	1

union, intersection, and left and right quotients. They also studied the state complexity of some operations for unary languages. More than two decades before, in 1970, Maslov [156] had presented some estimates for union, catenation, and star. Although Maslov considered possible incomplete DFAs, and the paper has some incorrections, the binary languages presented are tight witnesses for the upper bounds for that three operations [14]. Rabin and Scott [181] indicated the upper bound mn for the intersection (that also applies to union). Maslov and Yu *et al.* gave similar witnesses of tightness, both for union and intersection. The families of languages given by Yu *et al.* for intersection are $\{x \in \{a,b\}^\star \mid \#_a(x) = 0 \pmod{m}\}$ and $\{x \in \{a,b\}^\star \mid \#_b(x) = 0 \pmod{n}\}$. Their complements are witnesses for union. Hricko *et al.* [112] showed that for any integers $m \geq 2$, $n \geq 2$, and $\alpha \in [1, mn]$, there exist binary languages L_1 and L_2 such that $sc(L_1) = m$, $sc(L_2) = n$, and $sc(L_1 \cup L_2) = \alpha$. Thus, there are no magic numbers for the union. The same holds for intersection.

Complementation for DFAs is trivial (one has only to exchange the final states) and thus, the state complexity of the complement is the same one as of the original language, i.e., $sc(\overline{L}) = sc(L)$. For other Boolean operations (set difference, symmetric difference, exclusive disjunction, etc.) the state complexity can be obtained by expressing them as a function of union, intersection and complement [14].

For catenation, Yu *et al.* gave the upper bounds $m2^n - f_1 2^{n-1}$, if $m \geq 1, n \geq 2$; and m, if $m \geq 1, n = 1$. They presented binary languages tight bound witnesses for $m \geq 1$, $n = 1$ and $m = 1$, $n \geq 2$, but ternary languages tight bound witnesses for $m > 1$, $n \geq 2$. However, the bound is tight for the following binary language families presented by Maslov: $\{w \in \{a,b\}^\star \mid \#_a(w) = (m-1) \pmod{m}\}$ and $L((a^\star b)^{n-2}(a+b)(b+a(a+b))^\star)$, for all $m, n \geq 2$ and $f_1 = 1$. Other families of binary languages for which the catenation achieves the upper bound were presented by Jirásková [122]. Concerning the possible existence of magic numbers, the same author [124, 126] proved that, for all m, n and α such that either $n = 1$ and $\alpha \in [1, m]$, or $n \geq 2$ and $\alpha \in [1, m2^n - 2^{n-1}]$, there exist languages L_1 and L_2 with $sc(L_1) = m$ and $sc(L_2) = n$, defined over a growing alphabet, such that $sc(L_1 L_2) = \alpha$. This result was improved by showing that a linear alphabet is enough to produce all complexity values [141]. Moreover, Jirásek *et al.* [119] showed that the upper bound $m2^n - f_1 2^{n-1}$ on the catenation of two languages L_1 and L_2, with $sc(L_1) = m \geq 2$ and $sc(L_2) = n \geq 2$ respectively, are tight for any integer f_1 with $f_1 \in [1, m-1]$. The witness language families are binary and accepted by the DFAs presented in Figure 2.

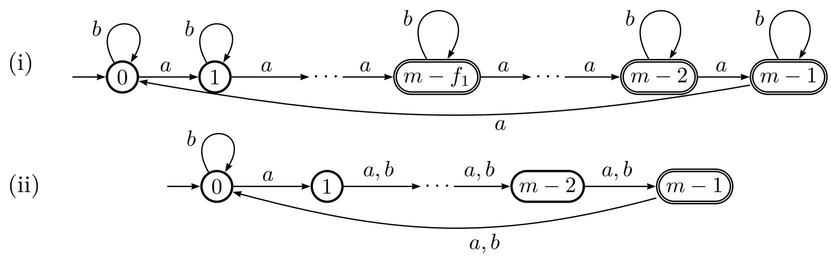

Figure 2: Witness DFAs for all range of state complexities of the catenation

The state complexity for the star on a regular language L was studied by Yu *et al.*. A lower bound of 2^{m-1} was presented before, by Ravikumar and Ibarra [185, 184]. If $sc(L) = 1$ then either $L = \Sigma^\star$, and $sc(L^\star) = 1$, or $L = \emptyset$, and $sc(L^\star) = 2$. If $sc(L) = m > 1$, but $l = 0$, i.e., the minimal DFA accepting L has the initial state as the only final state, then $sc(L^\star) = m$, as $L = L^\star$. Finally, if $sc(L) = m > 1$, and $l > 0$, then $sc(L^\star) \leq 2^{m-1} + 2^{m-l-1}$. The upper bound $2^{m-1} + 2^{m-2}$ is achieved for the language $\{w \in \{a,b\}^\star \mid \#_a(w) \text{ is odd}\}$, if $m = 2$; if $m > 2$, for the family of binary languages accepted by the DFAs presented in Figure 2:(ii). We note that although the upper bound given by Maslov is not correct ($\frac{3}{4}2^m - 1$ instead of $\frac{3}{4}2^m$), the family of languages he presented are witnesses for the above bound (for $m > 2$). Those languages are accepted by the DFAs presented in Figure 3.

Jirásková [123] proved that for all integers m and α with either $m = 1$ and $\alpha \in [1, 2]$, or $m \geq 2$ and $\alpha \in [1, 2^{m-1} + 2^{m-2}]$, there exists a language L over an alphabet of size 2^m such that $sc(L) = m$ and $sc(L^\star) = \alpha$. This result was improved by Jirásková *et*

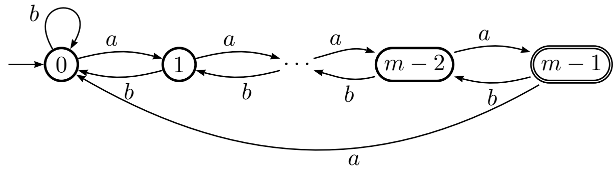

Figure 3: Maslov's witness DFAs for the state complexity of the star

al. [138] by using an alphabet of size atmost $2m$. Again, no gaps or magic numbers exist for the Kleene star operation.

The state complexity for the plus on a regular language L ($L^+ = LL^\star$) coincides with the one for star in the first two cases, but for $m > 1$ and $l > 0$ one state is saved (as a new initial state is not needed).

In 1966, Mirkin [162] pointed out that the reversal of the NFAs given by Lupanov as an example of a tight bound for determination (see Figure 1:(ii)), were deterministic. This yields that 2^m is a tight upper bound for the state complexity of reversal of a (at least ternary) language L such that $sc(L) = m$. Leiss [149] studied also this problem and proved the tightness of the bound for another family of ternary languages. Yu *et al.* presented also (independently) the Lupanov example. Salomaa *et al.* [189] studied several classes of languages where the upper bound is achieved. Nevertheless, a family of binary languages therein presented as meeting the upper bound for $m \geq 5$ was later proved to be wrong [128]. A family of binary languages for which the upper bound for reversal is tight was given by Jirásková and Sěbej [199, 140] and their minimal DFAs are represented in Figure 4.

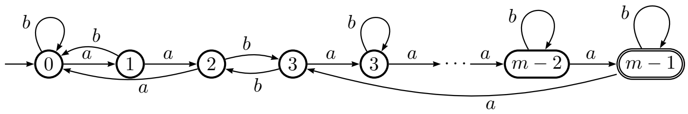

Figure 4: Witness DFAs for the state complexity of the reversal

In the paper cited above [123], Jirásková has shown that for all m and α with $2 \leq m \leq \alpha \leq 2m$, there exists a binary languague L such that $sc(L) = m$ and $sc(L^R) = \alpha$. Allowing alphabets of size 2^m and $m \geq 3$, the reversal operation has no magic numbers in the range $[\log m, 2^m]$. This result was improved by Sěbej [196] considering an alphabet of size $2m-2$. Sěbej gives also some enhanced partial results for the binary case.

Yu *et al.* showed that the state complexity for the left quotient of a regular language L_1 by an arbitrary language L_2, $L_2 \setminus L_1$, is less or equal to $2^m - 1$, with $sc(L_1) = m$, and that this bound is tight for the family of binary languages given in Figure 2:(ii) and considering $L_2 = \Sigma^\star$. In 1971, Conway [54] had already stated that if L_2 is a regular language then $sc(L_2 \setminus L_1) \leq 2^m$. For the right quotient of a regular language L_1 by an arbitrary language L_2 one has $sc(L_1/L_2) \leq m$. The minimal DFA accepting L_1/L_2 coincides with the one for L_1, except that the set of final states is the set of states $q \in Q_1$ such that there exists a word of $w \in L_2$ such that $\delta_1(q, w) \in F_1$. The bound is tight for $L_2 = \{\varepsilon\}$. For the left and the right quotients of a regular language L by a word $w \in \Sigma^\star$ it is then easy to see that $sc(w^{-1}L) = sc(Lw^{-1}) \leq m$. As a family of languages for which the upper bound is tight consider $(a^m)^\star$ and $w \in \{a\}^\star$ [64].

The state complexity of basic operations on NFAs was first studied by Holzer and Kutrib [104], and also by Ellul [64]. We note that for state complexity purposes it is tantamount to consider NFAs with or without ε-transitions. NFAs are considered with only one initial state and trimmed, i.e., all states are accessible from the initial state and from all states a final state is reached.

For union, only a new initial state with ε transitions for each of the operands initial states is needed, thus $sc(L_1 \cup L_2) \leq m + n + 1$. To see that the upper bound is tight, consider the families $(a^m)^\star$ and $(b^n)^\star$ over a binary alphabet. For intersection, a product construction is needed.

The nondeterministic state complexity of the complementation is, trivially, at most 2^m. That this upper bound is tight even for binary languages was proved by Jirásková [122], using a *fooling-set lower-bound technique* [7, 85, 113]. Those languages are accepted by the NFAs presented in Figure 5 (for $m > 2$).

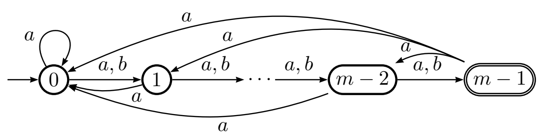

Figure 5: Witness NFAs for the nondeterministic state complexity of complementation

See Holzer and Kutrib [107] for other witness languages. Using the same techniques, Jirásková and Szabari [119] proved that for all integers $m \geq 1$ and $\alpha \in [\log m, 2^m]$, there exists a language L over an alphabet of exponential growing size, such that $nsc(L) = m$ and $nsc(\overline{L}) = \alpha$. This result was improved to a five-symbol alphabet by Jirásková [123].

Mera and Pighizzini [159] proved a related *best case* result, i.e., for every $m \geq 2$ there exists a language L such that $nsc(L) = m$, $nsc(\overline{L}) \leq m+1$ and $sc(L) = sc(\overline{L}) = 2^m$. However, as we will see below, this result does not hold if unary languages are considered.

The upper bound for the nondeterministic state complexity of catenation is $m + n$ and this bound can be reached considering the witness binary languages given for union. All the values $\alpha \in [1, m + n]$ can be obtained as nondeterministic state complexity of catenation of unary languages [124].

For the plus of a regular language L, we have $nsc(L^+) \leq nsc(L) = m$: an NFA accepting L^+ coincides with one accepting L except that each final state has also the transitions to the initial state. In the case of the star, one more state can be needed (if L does not accept the empty word), i.e., $sc(L^\star) \leq m+1$. Witness languages of the tightness of these bounds are $\{w \in \{a, b\}^\star \mid \#_a(w) = (m-1) \pmod m\}$. All range of values $\alpha \in [1, m + 1]$ can be reached for the nondeterministic state complexity of the star of binary languages [123].

For the reversal, at most one more state will be needed, so $nsc(L^R) \leq m + 1$. Witness ternary languages were presented by Holzer and Kutrib, but the bound is tight even for the family of binary languages ($m > 1$) which minimal NFAs are presented in Figure 6 [122]. If $nsc(L) = m \geq 3$ the possible values for $nsc(L^R)$ are $m-1$, m or $m+1$ [123]. The first value is reached by the reversals of the above binary languages and the second considering the languages $\{w \in \{a,b\}^\star \mid |w| = 0 \pmod{m}\}$.

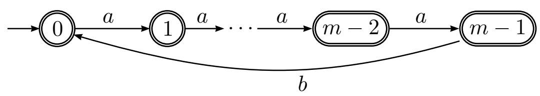

Figure 6: Witness NFAs for the nondeterministic state complexity of reversal

The nondeterministic state complexity of left and right quotients by a word were studied by Ellul [64]. Given a minimal NFA $A = (Q, \Sigma, \delta, q_0, F)$ accepting L, an NFA C accepting Lw^{-1}, for $w \in \Sigma$, coincides with A except that the set of final states is $\{q \in Q \mid \delta(q,w) \cap F \neq \emptyset\}$. Thus $nsc(Lw^{-1}) \leq nsc(L)$. The witness languages used for the state complexity of right quotient show that the bound is tight. An upper bound for $nsc(w^{-1}L)$ can be obtained by considering an NFA C with one new initial state q_0' and ε-transitions from q_0' to each state of A reached when inputing w.

Universal Witnesses Brzozowski [17, 18] identified a ternary family of languages $U_m(a,b,c)$ over $\Sigma = \{a,b,c\}$ which provides witnesses for the state complexity of all operations considered in the previous section. The family, presented in Figure 7 and called *universal witness*, fulfills also other conditions that, according to the same author, should be verified by *the most complex (regular) languages*. For a language L_m the suggested conditions are:

(1) The state complexity should be m.

(2) The state complexity of each quotient of L_m should be m.

(3) The number of atoms of L_m should be 2^m. An atom of a regular language with quotients $K_0, \ldots, K_{m-1}$ is a non-empty intersection of the form $\widetilde{K}_0 \cap \cdots \cap \widetilde{K}_{m-1}$, where $\widetilde{K}_i$ is either K_i or $\overline{K}_i$. Thus the number of atoms is bounded from above by 2^m, and it was proved by Brzozowski *et al.* [39, 41] that this bound is tight[1]. Every quotient of L_m is a union of atoms.

(4) The state complexity of each atom of L_m should be maximal. It was shown [40] that the complexity of the atoms with 0 or m complemented quotients is bounded from above by $2^m - 1$, and the complexity of any atom with r complemented quotients, where $1 \leq r \leq m-1$, by

$$f(m,r) = 1 + \sum_{k=1}^{r} \sum_{h=k+1}^{m-r+k} \binom{m}{h}\binom{h}{k}.$$

(5) The syntactic semigroup of L_m should have cardinality m^m, which is well known to be a tight upper bound [156]. This measure, which is called the *syntactic*

[1] We also notice that the number of atoms of a language L is equal to the state complexity of L^R.

complexity of a language, has been recently studied for many classes of subregular languages [103, 147, 31, 42, 29, 30, 36, 22, 38, 37].

The following result [17, 18] can be considered a milestone in the operational state complexity for regular languages:

> $(U_m(a,b,c) \mid m \geq 3)$ *meets conditions 1–5 and is a witness for the reversal and the star. The families* $(U_m(a,b,c) \mid m \geq 3)$ *and* $(U_n(b,a,c) \mid n \geq 3)$ *are witnesses for the Boolean operations, whereas* $(U_m(a,b,c) \mid m \geq 3)$ *and* $(U_n(a,b,c) \mid n \geq 3)$ *are witnesses for catenation.*

Variants (or dialects) of the universal witness were also given for several combined operations. The question of whether there are universal witnesses for other operations, classes of subregular languages or other complexity measures is a topic of recent research (see page 29). However, when searching for witnesses for a given upper bound, to ensure that the above conditions (or, at least, some of them) are verified, can be a good starting point. Moreover, the study of properties that may enforce (some of) the conditions (1) – (5) is fundamental for a better understanding of the operational state complexity [20].

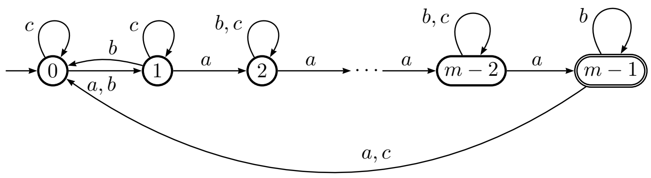

Figure 7: Universal witness DFAs, $U_m(a,b,c)$.

4.1.2. Unary Regular Languages

Table 2 presents the main state complexity results of the basic operations on unary languages. Given the constraints on both DFAs and NFAs over a one symbol alphabet, and the results presented in Section 3, the state complexity for several operations on unary languages is much lower than what is predicted by the general results of state complexity. Some results on the average-case state complexity of operations on unary languages were presented by Nicaud [169, 170].

A DFA that accepts a unary language is characterized by a noncyclic part (the tail) and a cyclic part (the loop). A characterization and the enumeration of minimal unary DFAs was given by Nicaud [169].

The state complexity of the reversal of a unary language L is trivially equal to the state complexity of L. The state complexities of Boolean operations on unary languages coincide asymptotically with the ones on general regular languages. Yu [203] has shown that the bound was tight for union (and thus, for intersection) if m and n are coprimes and the witness languages are $(a^m)^\star$ and $(a^n)^\star$. The state complexity of catenation and star was proved by Yu *et al.* [208] and the tightness for the first was

Table 2: State complexity (*sc*), nondeterministic state complexity (*nsc*) and average state complexity (*asc*) of basic operations on unary languages. The $\sim$ symbol means that the complexities are asymptotically equal to the given values. The upper bounds of state complexity for union, intersection and catenation are exact if the greatest common divisor of m and n, (m,n) is 1. For the average state complexity of intersection and union, $\zeta(n)$ is the function ζ of Riemman. For the average state complexity of catenation, n must be bounded by a polynomial P in m.

	Unary Regular		
	sc	nsc	asc
$L_1 \cup L_2$	$\sim mn$	$m+n+1$, if $m \neq \dot{n}$	$\sim \frac{3\zeta(3)}{2\pi^2}mn$
$L_1 \cap L_2$	$\sim mn$	mn, if $(m,n)=1$	$\sim \frac{3\zeta(3)}{2\pi^2}mn$
$\overline{L}$	m	$e^{\Theta(\sqrt{m \log m})}$	
L_1L_2	$\sim mn$	$[m+n-1, m+n]$, if $m,n>1$	$O(1)$, if $n<P(m)$
$L^\star$	$(m-1)^2+1$, if $m>1$, $l>1$	$m+1$, if $m>2$	$O(1)$
L^+	$(m-1)^2$	m, if $m>2$	
L^R	m	m	
$w^{-1}L$	m	m	
Lw^{-1}	m	m	

also shown for m and n coprimes. The witnesses for the catenation are $(a^m)^\star a^{m-1}$ and $(a^n)^\star a^{n-1}$. For the star, if $m=2$ a witness is $(aa)^\star$, and for each $m>2$ a witness is $(a^m)^\star a^{m-1}$. The state complexity when m and n are not necessarily coprimes was studied by Pighizzini and Shallit [176, 177]. In this case, the tight bounds are given by the number of states in the tail and in the loop of the resulting automata. The state complexity for left and right quotient by a word on unary languages coincides with the general case.

Nicaud [169, 170] proved that the state complexity of union, intersection and catenation on two languages L_1 and L_2 are asymptotically equivalent to mn, where $m = sc(L_1)$ and $n = sc(L_2)$. Let D_n be the set of unary (complete and initially connected) DFAs with n states. The *average state complexity* (asc) of a binary operation $\circ$ on regular languages is given by

$$\frac{\sum_{A_1 \times A_2 \in D_m \times D_n} sc(L(A_1) \circ L(A_2))}{|D_m \times D_n|}$$

This definition can be generalized to operations with other arities, other kinds of automata and classes of languages.

As shown in Table 2, the average state complexities of catenation and star on unary languages are bounded by a constant, and for intersection (and union) note that $\frac{3\zeta(3)}{2\pi^2} \approx 0.1826907423$. Magical numbers for the star operation on unary languages were studied by Čevorová [48]. Considering the gap between the worst-case upper bound, $n^2 - 2n + 2$, and the average case (less than a constant), it is not a surprise that for every n no more than 4 complexities are attainable between $n^2 - 4n + 6$ and the upper bound. In the same paper, the author also establishes a relation between this problem and the Frobenius problem.

The nondeterministic state complexity of basic operations on unary languages was studied by Holzer and Kutrib [105], and also by Ellul [64]. For union and intersection, the upper bound coincides with the general case. However, it was proved to be achievable for union if m is not a divisor or multiple of n. As in the deterministic case, the witnesses for intersection are $(a^m)^\star$ and $(a^n)^\star$, if m and n are coprimes. The nondeterministic state complexity of the complementation is $O(F(m))$ (where F is the Landau's function of equation (1)), which is directly related with the state complexity of determination. Holzer and Kutrib [105] proved that this upper bound is tight in order of magnitude, i.e., for any integer $m > 1$ there exists a unary language L such that $nsc(L) = m$ and $nsc(\overline{L}) = \Omega(F(m))$. Moreover, Mera and Pighizzini [159] have shown that for each $m \geq 1$ and unary language L, such that $nsc(L) = m$ and $sc(L) = sc(\overline{L}) = e^{O(\sqrt{m \log m})}$, then $nsc(\overline{L}) \geq m$. The upper bound $m + n$ for the catenation of two unary languages is not known to be tight. The known lower bound is $m+n-1$ achieved by the catenation of $\{a^l \mid l = (m-1) \pmod m\}$ and $\{a^l \mid l = (n-1) \pmod n\}$ [105]. The same languages can be used to show the tightness of the bound $m + 1$ for the star (and the plus) operation. For the left and right quotients, notice that in the unary case $w^{-1}L = Lw^{-1}$, and the results for the general case apply.

4.1.3. Finite Languages

Finite languages are an important subset of regular languages. They are accepted by complete DFAs that are acyclic apart from a loop on the *sink* (or dead) state, for all alphabetic symbols. Minimal DFAs have also special graph properties that lead to a linear time minimization algorithm [186], and where the length of the longest word accepted by the language plays an important role. Table 3 shows that the (nondeterministic) state complexity of operations on finite languages are, in general, lower than in the general case.

Câmpeanu *et al.* [43] presented the first formal study of state complexity of operations on finite languages. Yu [203] presented upper bounds of $O(mn)$ for the union

Table 3: State complexity and nondeterministic state complexity of basic operations on finite languages.

	Finite			
	sc	$\|\Sigma\|$	nsc	$\|\Sigma\|$
$L_1 \cup L_2$	$mn - (m+n)$	$f(m,n)$	$m+n-2$	2
$L_1 \cap L_2$	$mn - 3(m+n) + 12$	$f(m,n)$	$O(mn)$	2
$\overline{L}$	m	1	$\Theta(k^{\frac{m}{1+\log k}})$	2
L_1L_2	$(m-n+3)2^{n-2} - 1$, $m+1 \geq n$	2	$m+n-1$	2
	$m+n-2$, if $l_1 = 1$	1		
$L^\star$	$2^{m-3} + 2^{m-l-2}$, $l \geq 2$, $m \geq 4$	3	$m-1$, $m > 1$	1
	$m-1$, if $f = 1$	1		
L^+	m	1	m, $m > 1$	1
L^R	$O(k^{\frac{m}{1+\log k}})$	2	m	2

and the intersection. The tight upper bounds were given by Han and Salomaa [91] using growing size alphabets. The upper bound for union and intersection cannot be reached with a fixed alphabet when m and n are arbitrarily large. Câmpeanu *et al.* gave tight upper bounds for catenation, star and reversal. For catenation the bound $(m-n+3)2^{n-2}-1$ is tight for binary languages, if $m+1 \geq n > 2$. The DFAs of the witness languages are presented in Figure 8.

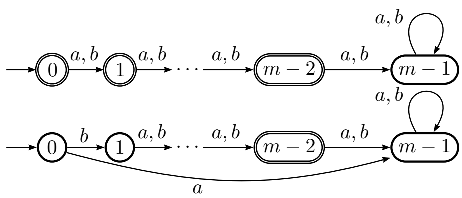

Figure 8: Witness DFAs for the state complexity of catenation on finite languages

For star, Câmpeanu *et al.* have shown that the bound $2^{m-3} + 2^{m-4}$ is tight for ternary languages. The tight upper bound for the reversal of a finite binary language is $3 \cdot 2^{p-1} - 1$, if $m = 2p$, and $2^{p-1} - 1$ if $m = 2p - 1$.

Nondeterministic state complexity of basic operations on finite languages were studied by Holzer and Kutrib [104]. Minimal NFAs accepting finite languages without the empty word can be assumed to have only a final state (with no transitions); and if the empty word is in the language, the initial state is also final. Because there are no cycles, for the union of two finite languages three states can be avoided: no new initial state is needed, and the initial states and the final states can be merged. The upper

Table 4: State complexity and nondeterministic state complexity of basic operations on finite unary languages.

	Finite Unary	
	sc	nsc
$L_1 \cup L_2$	$\max\{m,n\}$	$\max\{m,n\}$
$L_1 \cap L_2$	$\min\{m,n\}$	$\min\{m,n\}$
$\overline{L}$	m	$m+1$
$(L_1 - L_2)$	m	
$(L_1 \oplus L_2)$	$\max\{m,n\}$	
L_1L_2	$m+n-2$	$m+n-1$
$L^\star$	2, if $m=3$ $m-1$, if $f=1$ $m^2-7m+13$, if $m>4$, $f\geq 3$	$m-1$
L^+	m	m
L^R	m	m

bound $m+n-2$ is tight for the languages a^{m-1} and b^{n-1}, for $m,n \geq 2$. Considering the upper bound of determination for finite languages, the nondeterministic state complexity for complement is bounded by $O(k^{\frac{m}{1+\log k}})$. The lower bound $\Omega(k^{\frac{m}{2\log k}})$ is reached for alphabets $\Sigma = \{a_1,\ldots,a_k\}$ of size $k \geq 2$, and the languages $\Sigma^j a_1 \Sigma^i y$, where $i \geq 0$, $0 \leq j \leq i$, $y \in \Sigma \setminus \{a_1\}$, and $m > 2$. However, a tighter lower bound can be achieved by the determination lower bound of $\Omega(k^{\frac{m}{1+\log k}})$. For catenation of finite languages represented by NFAs, one state can be saved. Witness languages for the tightness of the bound $m+n-1$ can be the ones used for union. Two states are also saved for the star, and for plus the nondeterministic state complexity coincides with the one for the general case. Witness languages are a^m and a^{m-1}, respectively. NFAs for the reversal are exponentially more succinct than DFAs. In the case of finite languages, and like other operations, one state can be spared. Witness languages are $\{a,b\}^{m-1}$.

4.1.4. Finite Unary Languages

Table 4 summarizes the state complexity and nondeterministic state complexity results of basic operations on finite unary languages [43, 203, 105]. State complexity of union, intersection and catenation on finite unary languages are linear, while they are quadratic for general unary languages. In this setting, nondeterminism is only relevant for the star (and plus), as unary regular languages are obtained. As already stated, for a finite unary language L, one has $sc(L) \leq nsc(L) + 1$, and $sc(L) - 2$ is the length of the longest word in the language. If an operation preserves finiteness, for state complexity only the longest words must be considered.

4.2. Other Regularity Preserving Operations

Table 5 presents the results for the state complexity of some regularity preserving operations, that are detailed in the next paragraphs.

Proportional removals The *proportional removals* preserving regularity were studied by Hartmanis [198] and were full characterized by Seiferas and McNaughton [197]. For any binary relation $r \subseteq \mathbb{N} \times \mathbb{N}$ and any language $L \subseteq \Sigma^\star$, let the language $P(r, L)$ be defined as

$$P(r, L) = \{x \in \Sigma^\star \mid \exists y \in \Sigma^\star \text{ such that } xy \in L \wedge r(|x|, |y|)\}.$$

A relation r is *regularity-preserving* if $P(r, L)$ is regular for every regular language L. Seiferas and McNaughton [197] gave sufficient and necessary conditions of regularity preservation in this context. For the special case where r is the identity, the correspondent language is denoted by $\frac{1}{2}(L)$. Domaratzki [61] proved that for a regular language L, $sc(\frac{1}{2}(L)) = O(sc(L)F(sc(L)))$ (where F is the Landau's function of equation (1)) and this bound is tight for ternary languages. In the case of L be a unary language, one gets $sc(\frac{1}{2}(L)) = sc(L)$. Following Nicaud's work on average-case complexity, mentioned above, Domaratzki showed that the average state complexity of the $\frac{1}{2}(\cdot)$ operation on an m-state unary automaton is asymptotically equivalent to $\frac{5}{8}m + c$, for some constant c. Domaratzki also studied the state complexity of polynomial removals. Let $f \in \mathbb{Z}[x]$ be a strictly monotonic polynomial such that $f(\mathbb{N}) \subset \mathbb{N}$. Then, the relation $r_f = \{(n, f(n)) \mid n \geq 0\}$ preserves regularity, and $sc(P(r_f, L)) \leq O(sc(L)F(sc(L)))$.
In 1970, Maslov [156] had already studied the language $\frac{p}{q}(L)$, i.e., a language $P(r, L)$ such that r is defined by $\{(m, n) \mid mq = pn\}$ with $p, q \in \mathbb{N}$. An open problem is to obtain the state complexity of $P(r, L)$ where r belongs to the broader class of regularity preserving relations studied by Seiferas and McNaughton.

Nondeterministic state complexity of polynomial removals was studied by Goč *et al.* [86]. The authors showed an $O(n^2)$ upper bound and a matching lower bound in the case where the polynomial is a sum of monomials and a constant, or when the polynomial has rational roots.

Power Given a regular language L and $i \geq 2$, an upper bound of the state complexity of the language L^i is given by considering the state complexity of catenation. However, a tight upper bound is obtained if this operation is studied individually. Domaratzki and Okhotin [62] proved that $sc(L^i) = \Theta(m2^{(i-1)m})$, for $i \geq 2$. The bound is tight for a family of languages over a six-symbol alphabet. In the case $i = 3$, $sc(L^3) = \frac{6m-3}{8}4^m - (m-1)2^m - m$, for $m \geq 3$, and the tightness is witnessed by a family of languages over a four-symbol alphabet. For the square, i.e. if $i = 2$, the upper bound is the one given by the state complexity of catenation, $sc(L^2) = m2^m - 2^{m-1}$ and it is met by a language accepted by a m-state DFA with only one final state. In the case

Table 5: State complexity and nondeterministic state complexity of some regularity preserving operations: proportional removals for the identity relation ($\frac{1}{2}(L)$); power L^i where $i \geq 2$; cyclic shift L^{CS}; shuffle $L_1 \sqcup\!\sqcup L_2$; and orthogonal catenation $L_1 \odot_\perp L_2$.

	Regular			
	sc	$\lvert\Sigma\rvert$	nsc	$\lvert\Sigma\rvert$
$\frac{1}{2}(L)$	$me^{\Theta(\sqrt{m \log m})}$	3	$O(m^2)$	
	m	1		
L^i	$\Theta(m2^{(i-1)m})$	6		
	$im - i + 1$	1	im	2
L^3	$\frac{6m-3}{8}4^m - (m-1)2^m - m$	4		
L^{CS}	$2^{m^2+m\log m - O(m)}$	4	1, if $m = 1$ $2m^2+1$, if $m \geq 2$	2
	$2^{\Theta(m^2)}$	2,3		
	m	1	m	1
$L_1 \sqcup\!\sqcup L_2$	$\leq 2^{mn-1} + 2^{(m-1)(n-1)}(2^{m-1}-1)(2^{n-1}-1)$		$O(mn)$	5
$L_1 \odot_\perp L_2$	$m2^{n-1} - 2^{n-2}$, if $m \geq 3$, $n \geq 4$	4	$m+n$	2

of multiple l final states, the upper bound is $(m-l)2^m + l2^{m-1}$. Čevorová *et al.* [51] proved that those upper bounds are tight in the ternary case for every $l \in [1, m-2]$. The nondeterministic state complexity of L^i is proved to be im. This bound is shown to be tight over a binary alphabet, for $m \geq 2$. The power of unary languages was studied by Rampersad [182]. If L is a unary language with $sc(L) = m \geq 2$, then $sc(L^i) = im - i + 1$. For the square, Čevorová *et al.* showed that all the complexities in the range $[1, 2m-1]$ can be attained for $m \geq 5$.

Cyclic Shift The *cyclic shift* of a language L is defined as $L^{CS} = \{vu \mid uv \in L\}$. Maslov [156] gave an upper bound of $(m2^m - 2^{m-1})^m$ for the state complexity of cyclic shift and an asymptotic lower bound of $(m-3)^{m-3} \cdot 2^{(m-3)^2}$, by considering languages over a growing alphabet (if complete DFAs are considered). Jirásková and Okhotin [135] reviewed and improved Maslov results. Using a fixed four-symbol alphabet, they obtained a lower bound of $(m-1)! \cdot 2^{(m-1)(m-2)}$, $m \geq 3$, which shows that $sc(L^{CS}) = 2^{m^2+m\log m - O(m)}$ for alphabets of size greater than 3. For binary and ternary languages, they proved that the state complexity is $2^{\Theta(m^2)}$. As this function grows faster than the number of DFAs for a given m, there must exist some *magic* *numbers* for the state complexity of the cyclic shift over languages of a fixed alphabet.

The nondeterministic state complexity of this operation was shown to be $2^{m^2}+1$, for $m \geq 2$, and the upper bound is tight for binary languages. Although the hardness of this operation on the deterministic case, its nondeterministic state complexity is relatively low. For a unary language L, as $L^{CS} = L$, one gets $sc(L^{CS}) = nsc(L^{CS}) = sc(L)$.

Shuffle The *shuffle* operation of two words $w_1, w_2 \in \Sigma^\star$ is defined by

$$w_1 \sqcup\!\sqcup w_2 = \{u_1 v_1 \cdots u_m v_m \mid u_i, v_i \in \Sigma^\star,\ i \in [1,m],\ w_1 = u_1 \cdots u_m \text{ and } w_2 = v_1 \cdots v_m\}.$$

This operation is extended trivially to languages. If two regular languages are regular, their shuffle is also a regular language. Câmpeanu *et al.* [46] showed that the state complexity of the shuffle of two regular languages L_1 and L_2 is less or equal to $2^{mn}-1$. They proved that this bound is tight for witness languages over a five symbols alphabet and if minimal incomplete DFAs are considered. Thus, $sc(L_1 \sqcup\!\sqcup L_2)$ is at least $2^{(sc(L_1)-1)(sc(L_2)-1)}$. In 2016, Brzozowski *et al.* [25] improved these results and proved that $2^{mn-1} + 2^{(m-1)(n-1)}(2^{m-1}-1)(2^{n-1}-1)$ is an upper-bound. The tightness was proved for $2 \leq m \leq 5$ and $n \geq 2$ and also for $m = n = 6$. They have also shown that an alphabet of size mn is needed for the bound to be reached provided that $m, n \geq 2$.

Various restrictions and generalizations of the shuffle operation have been studied. Mateescu *et al.* [157] introduced the shuffle operation of two languages L_1 and L_2 on a set of trajectories $T \subseteq \{0,1\}^\star$, $L_1 \sqcup\!\sqcup_T L_2$. When L_1, L_2, and T are regular languages $L_1 \sqcup\!\sqcup_T L_2$ is a regular language. In particular, if $T = \{0,1\}^\star$, then $L_1 \sqcup\!\sqcup_T L_2 = L_1 \sqcup\!\sqcup L_2$; and if $T = \{0\}^\star\{1\}^\star$, then $L_1 \sqcup\!\sqcup_T L_2 = L_1 L_2$. Domaratzki and Salomaa [63] studied the state complexity of the shuffle on regular trajectories. In general, $sc(L_1 \sqcup\!\sqcup_T L_2) \leq 2^{nsc(L_1)nsc(L_2)nsc(T)}$. If T belongs to special families of regular languages, tight bounds were also presented.

Orthogonal Catenation A language L is the *orthogonal catenation* of L_1 and L_2, and denoted by $L = L_1 \odot_\perp L_2$, if every word w of L can be obtained in just one way as a catenation of a word of L_1 and a word of L_2. If catenation uniqueness is not verified for every word of L, orthogonal catenation of L_1 and L_2 is undefined, otherwise L_1 and L_2 are *orthogonal.* Daley *et al.* [59] studied the state complexity of orthogonal catenation and generalized orthogonality to other operations. Although it is a restricted operation, its state complexity is only half of the one for the general catenation, i.e., $m2^{n-1} - 2^{n-2}$ for $m \geq 3$ and $n \geq 4$. The tight bound was obtained for languages over a four-symbol alphabet. Concerning nondeterministic state complexity, one has $nsc(L_1 \odot_\perp L_2) = nsc(L_1) + nsc(L_2)$, which coincides with the one for (general) catenation. Witness languages presented for the catenation are orthogonal (see page 12), thus apply to orthogonal catenation.

Unique Regular Operations Similar to orthogonality is the concept of *unique operation* introduced by Rampersad *et al.* [183]. However, instead of demanding that every pair of words of the operand languages lead to a distinct word on the resulting language, the

Table 6: State complexity of unique operations: for unique star L°, $\varepsilon \notin L$; for the nondeterministic state complexity of $L_1 \circ L_2$, the combined state complexity of L_1 and L_2 is $O(h)$, for $h \geq 0$.

Unique Regular Operations			
	sc	$\lvert\Sigma\rvert$	nsc
$L_1 \overset{\circ}{\cup} L_2$	mn	2	
$L_1 \circ L_2$	$O(m3^n - f_1 3^{n-1})$		$\geq 2^{O(h)}$
$L^{\circ 2}$	$m3^m - 3^{m-1}$	2	
L°	$O(3^{m-1} + (f+2)3^{m-f-1}$ $-(2^{m-1} + 2^{m-f-1} - 2))$		

language resulting from a *unique operation* only contains the words that are uniquely obtained through the given operation. Rampersad *et al.* studied several properties of unique operations and of their *poly* counterpart (i.e. where each resulting word must be obtained in more than one way), such as closure, ambiguity, and membership and non-emptiness decision problems. Results on state complexity and nondeterministic state complexity were obtained for *unique union* ($L_1 \overset{\circ}{\cup} L_2$), *unique catenation* ($L_1 \circ L_2$), *unique square* ($L \circ L = L^{\circ 2}$), and *unique star* ($L^\circ$). The state complexity of $L_1 \overset{\circ}{\cup} L_2$ is mn, and witness binary languages are $\{x \in \{a,b\}^\star \mid \#_a(x) = (m-1) \pmod{m}\}$ and $\{x \in \{a,b\}^\star \mid \#_b(x) = (n-1) \pmod{n}\}$, for $m, n \geq 3$ (that were also used by Maslov [156] for general union). For unique catenation, $sc(L_1 \circ L_2) \leq m3^n - f_1 3^{n-1}$ which is much higher than the one for general catenation. It is an open problem to know if this bound is tight, although several examples, for specific values of m and n, were presented. However, for the unique square $sc(L^{\circ 2}) = m3^m - 3^{m-1}$, and the bound is tight for binary languages and $m \geq 3$. For the nondeterministic state complexity of unique catenation, an exponential lower bound was provided. An upper bound for the state complexity of the unique star is $3^{m-1} + (f+2)3^{m-f-1} - (2^{m-1} + 2^{m-f-1} - 2)$. But, again, it is an open problem to know if this upper bound is tight. Table 6 summarizes the known results.

4.3. Other Subregular Languages

Besides finite and unary languages, several other subregular languages are used in many applications and are now theoretically well studied. Prefix-free or suffix-free languages are examples of codes that are fundamental in coding theory [144, 6]. Prefix-closed, factor-closed, or subword-closed languages were studied by several authors [90, 200, 60, 84]. These languages belong to a broader set of languages, the *convex languages*, for which a general framework has been recently addressed by Ang and Brzozowski [2] and Brzozowski *et al.* [33]. A detailed survey on complexity topics was presented by Brzozowski [15]. Partially based on that survey, here we summarize some of the results concerning the state complexity of preserving regularity operations over some of the convex subregular languages. Star-free languages are another family of subregular languages well studied [194, 158]. We briefly address recent results on the (nondeterministic) state complexity of basic regular operations on these languages.

4.3.1. Convex Subregular Languages

We begin by some definitions and results on determination for these languages. Let $\trianglelefteq$ be a partial order on $\Sigma^\star$, and let $\trianglerighteq$ be its converse. A language L is $\trianglelefteq$-convex if $u \trianglelefteq v$ and $v \trianglelefteq w$ with $u, w \in L$ implies $v \in L$. It is $\trianglelefteq$-free if $v \trianglelefteq w$ and $w \in L$ implies $v \notin L$. It is $\trianglelefteq$-closed if $v \trianglelefteq w$ and $w \in L$ implies $v \in L$. It is $\trianglerighteq$-closed if $v \trianglerighteq w$ and $w \in L$ implies $v \in L$. The closure and the converse closure operations are:

$$_{\trianglelefteq}L = \{v \mid v \trianglelefteq w \text{ for some } w \in L\},$$

$$L_{\trianglelefteq} = \{v \mid w \trianglelefteq v \text{ for some } \ w \in L\}.$$

The *freeness* operation can defined for a language L, by

$$L^{\trianglelefteq} \subseteq L \text{ and } \forall w \in L^{\trianglelefteq}, \forall v \in \Sigma^\star,\ v \lhd w \text{ implies } v \notin L^{\trianglelefteq}.$$

The following proposition is from [2], except for the last item.

Proposition 1. *Let $\trianglelefteq$ be an arbitrary relation on $\Sigma^\star$. Then*
(I) *A language is $\trianglelefteq$-convex if and only if it is $\trianglerighteq$-convex.*
(II) *A language is $\trianglelefteq$-free if and only if it is $\trianglerighteq$-free.*
(III) *Every $\trianglelefteq$-closed language and every $\trianglerighteq$-closed language are $\trianglelefteq$-convex.*
(IV) *A language is $\trianglelefteq$-closed if and only if its complement is $\trianglerighteq$-closed.*
(V) *A language L is $\trianglelefteq$-closed ($\trianglerighteq$-closed) if and only if $L = _{\trianglelefteq} L$ ($L = L_{\trianglelefteq}$).*
(VI) *A language L is $\trianglelefteq$-free if and only if $L = L^{\trianglelefteq}$.*

We consider $\trianglelefteq$ to be:

- $\leq$: if $u, v, w \in \Sigma^\star$ and $w = uv$, then u is *prefix* of w, and we write $u \leq w$.
- $\preceq$: if $u, v, w \in \Sigma^\star$ and $w = uv$, then v is *suffix* of w, and we write $v \preceq w$
- $\sqsubseteq$: if $u, v, w \in \Sigma^\star$ and $w = uxv$, then x is *factor* of w, and we write $x \sqsubseteq w$. Note that a prefix or suffix of w is also a factor of w. This relation is also called *infix*.
- $\Subset$: if $w = w_0 a_1 w_1 \cdots a_n w_n$, where $a_1, \ldots, a_n \in \Sigma$, and $w_0, \ldots, w_n \in \Sigma^\star$, then $v = a_1 \cdots a_n$ is a *subword* of w; and we write $v \Subset w$. Note that every factor of w is a subword of w.

If a language is both prefix- and suffix-convex it is *bifix-convex*. *Bifix-free* and *bifix-closed* languages are defined in the same manner. Ideals are languages directly related with closed languages. A non-empty language $L \subseteq \Sigma^\star$ is a

- *right ideal* if $L = L\Sigma^\star$ (also called *ultimate definite* [174]); the complement is prefix converse-closed.
- *left ideal* if $L = \Sigma^\star L$ (also called *reverse ultimate definite* [174]); the complement is suffix converse-closed.
- *two-sided ideal* if $L = \Sigma^\star L \Sigma^\star$ (also called *central definite*); the complement is bifix converse-closed.
- *all-sided ideal* if $L = \Sigma^\star \sqcup\!\sqcup L$; the complement is subword converse-closed; also studied by Haines [90] and Thierrin [200].

Table 7: State complexity of determination of free, closed and ideal languages considering prefix, suffix and factor partial orders, respectively. For free and closed languages, the range of correspondent non-magic numbers appears on the second row.

		Free			
$\leq$	$\|\Sigma\|$	$\preceq$	$\|\Sigma\|$	$\sqsubseteq$	$\|\Sigma\|$
$2^{m-1}+1$	3	$2^{m-1}+1$	3	$2^{m-2}+2$	3
$]m, 2^{m-1}+1]$		$]m, 2^{m-1}+1]$		$]m, 2^{m-2}+2]$	
		Closed			
$\leq$	$\|\Sigma\|$	$\preceq$	$\|\Sigma\|$	$\sqsubseteq$	$\|\Sigma\|$
2^m	3	$2^{m-1}+1$	4	$2^{m-1}+1$	4
$]m, 2^m]$		$[m, 2^{m-1}+1]$		$]m, 2^{m-1}+1]$	
		Ideal			
right	$\|\Sigma\|$	left	$\|\Sigma\|$	two-sided	$\|\Sigma\|$
2^{m-1}	2	$2^{m-1}+1$	3	$2^{m-2}+1$	3

Some of the languages defined above are also characterized in terms of properties of the finite automata that accept them. In particular: prefix-closed languages are accepted by NFAs where all states are final; suffix-closed languages are accepted by NFAs where all states are initial; factor-closed languages are accepted by NFAs where all states are initial and final; prefix-free languages are accepted by non-exiting NFAs (i.e. there are no transitions from the final states); suffix-free languages are accepted by non-returning NFAs (i.e. there are no transitions to the initial state); and factor-free languages are accepted by non-returning and non-exiting NFAs.

The state complexity of the determination on some subregular languages (or for the kind of NFAs they are defined by) was recently studied by Bordihn *et al.* [9], Jui-Yi Kao *et al.* [145], and Jirásková *et al.* [128].

Table 7 presents some of the values for the languages considered above. The existence of magic numbers for some subregular languages was studied by Holzer *et*

Table 8: State complexity and nondeterministic state complexity of some operations on prefix-free languages.

	Prefix-free			
	sc	$\|\Sigma\|$	nsc	$\|\Sigma\|$
$L_1 \cup L_2$	$mn-2$	2	$m+n$	2
$L_1 \cap L_2$	$mn-2(m+n-3)$	2	$mn-(m+n)+2$	2
			2^{m-1}	3
$\overline{L}$	m	1	$[2^{\lfloor \frac{m}{2} \rfloor -1}, 2^{m-1}+2^{m-3}+1]$	2
			$\Theta(\sqrt{m})$	1
$(L_1 - L_2)$	$mn-m-2n+4$	3	$(m-1)2^{n-1}+1$	4
$(L_1 \oplus L_2)$	$mn-2$	2		
L_1L_2	$m+n-2$	1	$m+n-1$	1
L_1/L_2	$n-1$	2		
	$n-m+2$	1		
$L^\star$	m	2	m	1
	$m-2$	1		
L^R	$2^{m-2}+1$	3	m	1
L^{CS}	$(2m-3)^{m-2}$	6	$2m^2-4m+3$	2

al. [102]. As can be seen in Table 7, m is the only magic number for all free languages and for both prefix- and factor-closed languages (except if $m = 1$, where m is non-magic). Suffix-closed languages have no magic numbers.

Free languages Table 8 summarizes state complexity results of individual operations on prefix-free languages [94, 95, 128, 24, 138, 134, 118, 65, 163, 49]. In the case of state complexity, the results are valid for Boolean operations if $m, n \geq 3$; for catenation if $m, n \geq 2$; for star if $k = 1$, then $m \geq 3$, if $k = 2$ then $m \neq 3$, and else $m \geq 2$; and for reversal if $m \geq 4$ and the tight bound cannot be reached if $k = 2$ [128]. The state complexty of right quotient is 1, if $k = 1$ and $m = 1$ or $m > n$, and if $k = 2$ and $m = 1$ or $n = 1$; furthermore, if $m = 2$ then $sc(L_1/L_2) = n$ [118].

Note that here the state complexities of the catenation and the star are much lower than on general regular languages. For catenation, witness languages are a^{m-2} and a^{n-2}. Moreover, for the star, the only complexities attained are $m-2$, $m-1$, and m [138].

Nondeterministic state complexity of complementation on free languages was stud-

Table 9: State complexity and nondeterministic state complexity of some operations on suffix-free languages.

	Suffix-free			
	sc	$\|\Sigma\|$	nsc	$\|\Sigma\|$
$L_1 \cup L_2$	$mn-(m+n-2)$	2	$m+n-1$	2
$L_1 \cap L_2$	$mn-2(m+n-3)$	2	$mn-(m+n-2)$	2
			2^{m-1}	3
$\overline{L}$	m	1	$[2^{\lfloor \frac{m}{2} \rfloor -1}, 2^{m-1}+2^{m-3}+2]$	2
			$\Theta(\sqrt{m})$	1
$L_1 - L_2$	$mn-(m+2n-4)$	4		
$L_1 \oplus L_2$	$mn-(m+n-2)$	5		
L_1L_2	$(m-1)2^{n-2}+1$	4		
L^2	$m2^{m-3}+1$	3		
$L^\star$	$2^{m-2}+1$	4		
L^R	$2^{m-2}+1$	3		

ied by Jirásková and Mlynárcik [134, 163]. Comparing with the general case, one state is saved for union, catenation, star and reversal, and half of the sates are saved for complementation.

Table 9 summarizes the state complexity of some regular operations on suffix-free languages. Han and Salomaa showed that all bounds, except for complementation, difference, and symmetric difference, are tight [92, 93]. Jirásková and Olejár [137] provided binary witnesses for intersection and union. They also proved that for all integer α between 1 and the respective bound there are languages L_1 and L_2 such that $(n)sc(L_1 \circ L_2) = \alpha$, for $\circ \in \{\cap, \cup\}$ (and witnesses are ternary, except for $nsc(L_1 \cap L_2)$ for which the witnesses are over a four-symbol alphabet). The bounds for difference and symmetric difference are from Brzozowski *et al.* [24]. Čevorová [49] studied the square. Jirásková *et al.* [134] proved the results for complementation.

If a language is subword-free then it is factor-free, and if it is factor-free then it is bifix-free. Table 10 summarizes the state complexity of some regular operations on bifix-, factor-, and subword-free languages [24]. The tight upper bounds for the state complexity of these operations on the three classes of languages coincide. Complementation for these languages was also studied in [134, 163].

Nondeterministic state complexity for factor-free and subword-free languages is $2^{m-2}+1$, but for the former it is tight for ternary languages and the latter when the alphabet size is $\geq 2^{m-2}$. For binary alphabets and factor-languages, the upper bound is $2^{m-2} - 2^{m-4} + 1$ and the lower bound $\Omega(2^{\frac{m}{2}})$.

Closed Languages and Ideals Table 11 shows the state complexity of some basic operations on prefix-, suffix-, factor-, and subword-closed languages. A language is factor-closed if and only if it is subword-closed. So the state-complexity results of operations are the same for those classes. For prefix and suffix closed languages, the bounds for the square operation are not the ones of catenation [49, 50].

The state complexity of the closure on the respective partial orders is also considered. Subword and converse subword closures were first studied by Gruber *et al.* [88, 89] and Okhotin [172]. Brzozowski *et al.* [26, 27] presented the tight upper bound, by using a growing alphabet. Karandikar and Schoebelen [146] have shown that the exponential blown up is also required in the binary case. Given a regular language L with $sc(L) = m$, $nsc({}_{\Subset}L) = nsc(L_{\Subset}) = m$ and these upper bounds are tight for witness binary languages.

Prefix, suffix, and factor closures (respectively, ${}_{\leq}L$, ${}_{\preceq}L$, and ${}_{\sqsubseteq}L$) were studied by Kao *et al.* [145]. If L does not have $\emptyset$ as a quotient, Brzozowski *et al.* have shown that the state complexity of the suffix closure is $2^m - 1$ (instead of 2^{m-1}).

If L is a right (respectively, left, two-sided, all-sided) ideal, any language $G \subseteq \Sigma^\star$ such that $L = G\Sigma^\star$ (respectively, $L = \Sigma^\star G$, $L = \Sigma^\star G \Sigma^\star$,$L = \Sigma^\star \sqcup\!\sqcup G$) is a *generator* of L. Brzozowski and Jirásková [23] studied state complexity on ideals. Table 12 presents the state complexity of basic operations on ideals. As stated before closed languages and ideals are related. In particular, the state complexity of basic operations on two-sided and all-sided ideals coincide. Brzozowski [15] observed that for the four types of convex languages (prefix, suffix, factor and subword) the state complexity of the Boolean operations is mn. Čevorová [49] showed that the state complexity of the square on ideals coincides with one for catenation.

Operational nondeterministic state complexity on closed and ideal languages are presented in Table 13 [111]. Again, only for complementation the bounds are signifi-

Table 10: State complexity of basic operations on bifix-, factor-, and subword-free languages.

	Free			
	sc	$\|\Sigma\|$		
		$\leq \cup \preceq$	$\sqsubseteq$	$\Subset$
$L_1 \cup L_2$	$mn - m - n$	5	5	$< m + n - 3$
$L_1 \cap L_2$	$mn - 3m - 3n + 12$, $m, n \geq 4$	3	3	$m + n - 7$
$L_1 - L_2$	$mn - 2m - 3n + 9$	4	4	$< m + n - 6$
$L_1 \oplus L_2$	$mn - m - n$	5	5	$m + n - 3$
$L_1 L_2$	$m + n - 2$, $m, n > 1$	1	1	1
$L^\star$	$m - 1$, $m > 2$	2	2	2
L^R	$2^{m-3} + 2$, $m \geq 3$	2	2	$2^{m-3} - 1$

cantly smaller than the ones for general regular languages.

Unary convex languages In the case of unary languages, prefix, suffix, factor, and subword partial orders coincide. Table 14 summarizes the state complexity of basic operations on unary free, unary closed, unary ideals and unary convex languages. For nondeterministic state complexity, some results were given in Table 13.

Freeness Operations Here we analyse the state complexity of freeness operations for prefix, suffix, bifix and factor orders that were studied by Pribavkina and Rodaro [179]. Given a regular language L, the $\trianglelefteq$-free language $L^{\trianglelefteq}$ for $\trianglelefteq \in \{\leq, \preceq, \sqsubseteq\}$, is respectively[2]:

- prefix: $L^{\leq} = L - L\Sigma^{+}$
- suffix: $L^{\preceq} = L - \Sigma^{+}L$
- factor: $L^{\sqsubseteq} = L - (\Sigma^{+}L\Sigma^{\star} \cup \Sigma^{\star}L\Sigma^{+})$

The bifix operation is defined by $L^{b} = L^{\leq} \cap L^{\preceq}$. If L is an ideal, prefix, suffix and factor operations were studied by Brzozowski and Jirásková [23]. In this case, the resulting languages are minimal generators for left, right and two sided ideals,

[2] In [179] the superscripts for prefix, suffix and factor operations were respectively p, s and ι.

Table 11: State complexity of some operations on prefix-, suffix-, factor-, and subword-closed languages. The last two columns correspond to factor and subword, respectively. The last but one row contains the state complexity of the closure of prefix, suffix, and factor respectively. The last row contains the state complexity of the subword closure, considering unbounded and binary alphabets, respectively.

	Closed						
	$\leq$	$\|\Sigma\|$	$\preceq$	$\|\Sigma\|$	$\sqsubseteq,\Subset$	$\|\Sigma\|_{\sqsubseteq}$	$\|\Sigma\|_{\Subset}$
$L_1 \cup L_2$	mn	2	mn	4	mn	2	2
$L_1 \cap L_2$	$mn-m-n+2$	2	mn	2	$mn-m-n+2$	2	2
$L_1 - L_2$	$mn-n+1$	2	mn	4	$mn-n+1$	2	2
$L_1 \oplus L_2$	mn	2	mn	2	mn	2	2
L_1L_2	$m2^{n-2}+2^{n-2}$	3	$mn-fn+f$	3	$m+n-1$	2	2
L^2	$(m+4)2^{m-3}-1$	2	$\frac{1}{2}(m^2+m)-1$	3	$2m-1$	2	2
$L^{\star}$	$2^{m-2}+1$	3	m	2	2	2	2
L^R	2^{m-1}	2	$2^{m-1}+1$	3	$2^{m-2}+1$	3	$2m$
${}_{\trianglelefteq}L$	m	1	2^{m-1}	2	2^m-1	2	
${}_{\Subset}L$					$2^{m-2}+1$		$m-2$
					$2^{\Omega(\frac{m}{3})}$		2

Table 12: State complexity of basic operations on ideals. The last two columns correspond to two-sided and all-sided ideals, respectively.

	Ideal						
	right	$\|\Sigma\|$	left	$\|\Sigma\|$	-sided	$\|\Sigma\|_{\text{two}}$	$\|\Sigma\|_{\text{all}}$
$L_1 \cup L_2$	$mn-m-n+2$	2	mn	4	$mn-m-n+2$	2	2
$L_1 \cap L_2$	mn	2	mn	2	mn	2	2
$L_1 - L_2$	$mn-m+1$	2	mn	4	$mn-m+1$	2	2
$L_1 \oplus L_2$	mn	2	mn	2	mn	2	2
$L_1 L_2$	$m+2^{n-2}$	1	$m+n-1$	1	$m+n-1$	1	3
$L^\star$	$m+1$	2	$m+1$	2	$m+1$	2	2
	If $\varepsilon \in L$, then $L = \Sigma^\star$ and $sc(L^\star) = 1$.						
L^R	2^{m-1}	2	$2^{m-1}+1$	3	$2^{m-2}+1$	3	$2m-4$

respectively. Table 15 presents the state complexity of prefix, suffix, factor and bifix operations on regular languages (and correspondent ideals). The state complexity of this operations is much lower in the case of right and two-sided ideals than for general regular languages.

Universal Witnesses Most complex languages, i.e. that are "universal witnesses", for some classes of convex languages were investigated recently. Brzozowski *et al.* [21, 22] presented the most complex languages for right, left and two-sided ideals, respectively. For suffix-free languages with state complexity greater than 4 no universal witnesses can exist [37]. This is because it is not possible to satisfy simultaneously all the conditions (1)–(5) of page 13 (adapted for this class of languages). Most complex languages for prefix-convex languages (right-ideals, prefix-closed, prefix-free and proper) were deeper studied by Brzozowski and Sinnamon [34]. For binary operations, it was observed that if the operands are languages with different alphabets, larger complexity bounds can be obtained. The same generalization was considered for general regular languages and ideal regular languages [19, 35]. In this situation, universal witnesses may be slightly different than the ones we presented before.

4.3.2. Star-free Languages

Star-free languages are the smallest class containing the finite languages and closed under Boolean operations and catenation. This class of languages corresponds exactly to the regular languages of star height 0. The minimal DFAs of star-free languages are *permutation-free* (i.e. no word performs a non-trivial permutation of a subset of its states). This is the famous Schützenberger result that a language is star free if and only if its syntactic monoid is aperiodic [178]. Bordhin *et al.* [9] showed that the

Table 13: Nondeterministic state complexity of some operations on closed and ideal languages. Empty cells in column two (five) indicate that bounds are for all closed (ideal) languages.

		Closed			Ideal	
	$\trianglelefteq$	nsc	$\|\Sigma\|$		nsc	$\|\Sigma\|$
				right	$m+n$	2
		$m+n+1$	2	left	$m+n-1$	2
$L_1 \cup L_2$		$\max\{m,n\}$	1	sided	$m+n-2$	2
					$\min\{m,n\}$	1
$L_1 \cap L_2$		mn	2		mn	2
		$\min\{m,n\}$	1		$\max\{m,n\}$	1
$L_1 L_2$		$m+n$	3		$m+n-1$	1
		$m+n-1$	1			
L^2		$2m$	3		$2m-1$	1
		$2m-1$	1			
$L^\star$	$\leq$, $\preceq$	m	2		$m+1$	2
	$\sqsubseteq$,$\Subset$	1	1		$m-1$	1
L^R	$\leq$	$m+1$	2	right	m	1
	$\preceq$,$\sqsubseteq$	$m+1$	3	left	$m+1$	3
	$\Subset$	$m+1$	$2m-2$	sided	m	1
		m	1			
$\overline{L}$				right	2^{m-1}	2
	$\leq$	2^m	2	left	2^{m-1}	2
	$\preceq$,$\sqsubseteq$	$1+2^{m-1}$	2	2-sided	2^{m-2}	2
	$\Subset$	$1+2^{m-1}$	2^m	-sided	2^{m-2}	2^{m-2}
		$m+1$	1		$m-1$	1

Table 14: State complexity of basic operations on unary convex languages.

	Unary			
	Free	Closed	Ideal	Convex
$L_1 \cup L_2$	$\max\{m,n\}$	$\max\{m,n\}$	$\min\{m,n\}$	$\max\{m,n\}$
$L_1 \cap L_2$	$m = n$	$\min\{m,n\}$	$\max\{m,n\}$	$\max\{m,n\}$
$L_1 - L_2$	m	m	n	$\max\{m,n\}$
$L_1 \oplus L_2$	$\max\{m,n\}$	$\max\{m,n\}$	$\max\{m,n\}$	$\max\{m,n\}$
L_1L_2	$m+n-2$	$m+n-2$	$m+n-1$	$m+n-1$
$L^\star$	$m-2$	2	$m-1$	$n^2-7n+13$
L^R	m	m	m	m

Table 15: State complexity of prefix, suffix, factor and bifix operations on regular languages and on ideals (right, left and two sided, respectively).

	Regular		Ideal	
	sc	$\lvert\Sigma\rvert$	sc	$\lvert\Sigma\rvert$
$L^{\leq}$	$m+1$	2	$m+1$	2
$L^{\preceq}$	$(m-1)2^{m-2}+2$, $m \geq 4$	4	$\frac{n(n-1)}{2}+2$	1
$L^{\sqsubseteq}$	$(m-2)2^{m-3}+3$, $m \geq 4$	3	$n+1$	1
L^b	$(m-2)2^{m-2}+3$, $m \geq 4$	4		

state complexity of the determination of a star-free language L is $2^{nsc(L)}$. Figure 9 presents a family of ternary NFAs for which the bound is tight. Holzer *et al.* [102] showed that star-free languages have no magic numbers.

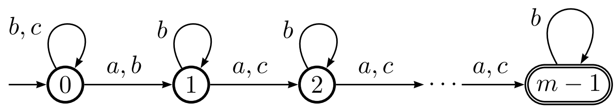

Figure 9: Minimal m-state NFAs with equivalent minimal 2^m-state DFA for star-free languages

Brzozowski and Liu [32] and Brzozowski and Szykuła [38] studied the state complexity of the basic regular operations on star-free languages, and their results are summarized in Table 16. The bounds obtained for general regular languages are reached except in the catenation for $n = 2$, the reversal, and operations on unary languages. Holzer *et al.* [109, 110] studied the same languages for the operational nondeterministic state complexity. The bounds coincide with the ones for general regular languages and are tight for binary languages. The witness languages for union and catenation are $a^{m-1}(ba^{m-1})^\star$ and $b^{n-1}(ab^{n-1})^\star$. For intersection, witnesses are $b^\star(ab^\star)^{m-1}$ and $a^\star(ba^\star)^{n-1}$. The first witness for union is also a witness for the star operation. The language family presented in Figure 6 is star-free and thus a witness for the reversal operation. On unary star-free languages, the upper bounds for operational nondeterministic state-complexity coincide with general case, except for the complementation. Holzer *et al.* [110] showed that for reversal and star the bounds are tight. For union, the presented lower bound misses the upper bound by one state. For intersection, the presented bound is tight in the order of magnitude ($\Theta(mn)$) and the bound for complementation is $\Theta(n^2)$. The lower bound for catenation misses the upper bound for unary general languages by one state.

Table 16: State complexity of basic regular operations on star-free regular and unary languages, where $\circ \in \{\cup, \cap, \setminus, \oplus\}$. For non-unary star-free languages and $n = 2$, $m \geq 2$. For non-unary star-free languages if $m \in [1, 2]$, the bound for reversal is tight for $|\Sigma| \geq m$, and if $m \geq 3$, for $|\Sigma| \geq m - 1$.

	Star-free		
	sc	$\|\Sigma\|$	Unary
$L_1 \circ L_2$	mn	2	$\max\{m, n\}$
$L_1 L_2$	$(m-1)2^n + 2^{n-1}$, if $n \geq 3$	4	$m + n - 1$
	$3m - 2$, if $n = 2$	3	
$L^\star$	2, if $m = 1$	1	2, if $m = 1$
	$2^{m-1} + 2^{m-2}$, if $m \geq 2$	4	m, if $m \in [2, 5]$
			$m^2 - 7m + 13$, if $m > 5$
L^R	$2^m - 1$	$m - 1$	m

There are some important subclasses of star-free languages besides the finite ones. The piecewise-testable languages are finite Boolean combinations of languages of the form $\Sigma^\star a_1 \Sigma^\star a_2 \Sigma^\star \cdots \Sigma^\star a_k \Sigma^\star$ where $k \geq 0$ and $a_i \in \Sigma$. And a subclass of these is the set of languages thar are finite unions of languages of the form $\Sigma_1^\star a_1 \Sigma_2^\star a_2 \Sigma_3^\star \cdots \Sigma_k^\star a_k \Sigma^\star$ where $k \geq 0$, $a_i \in \Sigma$ and $\Sigma_i \subseteq \Sigma \setminus \{a_i\}$, for $1 \leq i \leq k$. Jirásková and Masopust [132, 133] studied the state complexity of the reversal on these two classes. In both cases, the bound is 2^{m-1}. For the latter subclass the bound is met by a ternary language, and for the piecewise-testable languages an alphabet with $m - 1$ elements is needed.

4.4. Some More Results

We briefly cite some more work on operational state complexity. Câmpeanu and Ho [44] and Brzozowski and Konstantinidis [28] considered uniform finite languages.

Krieger *et al.* studied decimations of languages [148]. Câmpeanu and Konstantinidis [45] analysed a subword closure operation. Union-free languages were considered by Jirásková and Masopust [129, 130]. The same authors studied the state complexity of projected languages [131]. Determination and operational state complexity of basic operations on (strongly) bounded regular languages were investigated by Herrmann *et al.* [98]. The chop (or *fusion*) of two words is their catenation where the touching symbols are merged if equal, or is undefined otherwise. The chop operation and its iterated variants (star and plus) where studied by Holzer *et al.* [101, 99, 100]. The (nondeterministic) state complexity results are similar to the ones for catenation, star and plus, with the exception of chop-star where the complexities also depend on the alphabet size. This comes as a surprise as chop based regular expressions are known to be exponentially more succinct than classical catenation based ones. Bassino *et al.* [4] provided upper bounds of the state complexity of basic operations on cofinite languages as a function of the size the of complementary finite language (taken as the summation of the lengths of all its words). The average state complexity on finite languages is addressed in two works. Gruber and Holzer [87] analysed the average state complexity of DFAs and NFAs based on a uniform distribution over finite languages whose longest word is of length at most n. Based on the size of finite languages as the summation of the lengths of all its words and a correspondent uniform distribution, Bassino *et al.* [3] establish that the average state complexities of the basic regular operations are asymptotically linear.

5. State Complexity of Combined Operations

The number of standard individual operations on regular languages is clearly limited and almost all of their state complexities have been already obtained. However, in many practical cases, not only these individual operations but also their combinations are used, for example, the operations expressed by the regular expressions in the programming language Perl. These combinations are called combined operations.

In 2011, Salomaa *et al.* [188] proved that there cannot exist an algorithm such that, for a given composition of basic regularity preserving operations, computes the state complexity of the corresponding composed operation. The undecidability result holds already for arbitrary compositions of intersection and marked concatenation and the proof relies on a reduction from Hilbert's Tenth Problem. Although the composition of state complexities of individual component operations of a combined operation would give an upper bound for the state complexity of the combined operation, the upper bound is usually too high to be meaningful [150, 187, 206]. For example, for two regular languages L_1 and L_2 accepted by an m-state and an n-state DFA, respectively, the exact state complexity of $(L_1 \cup L_2)^*$ is actually $2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$, while the composition of their individual state complexities is $2^{mn-1} + 2^{mn-2}$. Clearly, $O(2^{m+n})$ and $O(2^{mn})$ are totally different.

Since the number of combined operations is unlimited and the state complexities of many of them are very difficult to compute, it would be good if we have a general estimation method that generates close upper bounds of the state complexities of combined operations which are good enough to use in practice. Such an estimation method has been proposed by Ésik *et al.* [66], and Salomaa and Yu [193]. A further concept in this direction, approximation of state complexity has been introduced Gao and Yu [78].

In the following, we will survey both the results of state complexities of combined operations and the results of estimations and approximations of state complexities of combined operations.

5.1. State Complexity of Combined Operations on Regular Languages

The state complexities of a number of basic combined operations on regular languages have been studied. Most of these combined operations are composed of two basic individual operations. The results are shown in Table 17.

In 1996, Birget [8] obtained the the state complexity of $\overline{\Sigma^\star \overline{L}}$, where L is a regular language. This combination of complementation, catenation and star is the first combined operation composed of different individual operations whose state complexity was established. In 2007, Salomaa *et al.* [187] pointed out that the mathematical composition of state complexities of individual component operations of a combined operation is usually much higher than the state complexity of the combined operation. This is because the result of a component operation of the combined operation may not be among the worst-cases of the succeeding component operation. They established the state complexity of $(L_1 \cup L_2)^\star$ and indicated that the state complexity of $(L_1 \cap L_2)^\star$ should be at least reasonably close to the mathematical composition of state complexities of intersection and star. Later, Jirásková and Okhotin [136] proved that the state complexity of $(L_1 \cap L_2)^\star$ is exactly the same as the mathematical composition of state complexities of intersection and star.

Gao *et al.* [75], in 2008, established the state complexities of $(L_1L_2)^\star$ and $(L_1^R)^\star$, where L_1 and L_2 are regular languages. The state complexity of $(L_1L_2)^\star$ is $2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$ which is lower than the mathematical composition of the state complexities of catenation and star. Interestingly, the state complexity of $(L_1^R)^\star$ is the same as that of L_1^R which is 2^m. The worst-case example over a three-letter alphabet for L_1^R [208] also works for $(L_1^R)^\star$.

In 2008, Liu *et al.* [150] studied the state complexities of $(L_1 \cup L_2)^R$, $(L_1 \cap L_2)^R$, and $(L_1L_2)^R$, where L_1 and L_2 are regular languages. The tight bounds for $(L_1 \cup L_2)^R$ were proved and the state complexity of $(L_1 \cap L_2)^R$ is the same as that of $(L_1 \cup L_2)^R$ because of De Morgan's laws and $\overline{L^R} = \overline{L}^R$. They also gave an upper bound for the last combined operation which was proved to be tight, in 2012, by Cui *et al.* [57].

Cui *et al.* [56] established the state complexities of $L_1(L_2 \cup L_3)$ and $L_1(L_2 \cap L_3)$ in 2011. The state complexity of $L_1(L_2 \cup L_3)$ is lower than the mathematical composition of the state complexities of union and catenation, whereas the state complexity of $L_1(L_2 \cap L_3)$ is the same as the corresponding composition.

In 2012, Jirásková and Shallit [139] proved the state complexity of the combined operation $\overline{\overline{L_1^\star}}^\star$ to be $2^{\Theta(m \log m)}$, where L_1 is a regular language accepted by an m-state DFA. A seven-letter alphabet was used in the proof for the lower bound. The boundary of a language L is the language $L^\star \cap (\overline{L})^\star$. Jirásek and Jirásková studied the state complexity of the boundary operation and the trivial upper bound of $\frac{9}{16}4^m$

Table 17: State complexities of some basic combined operations on regular languages.

	Regular	
	sc	$\lvert\Sigma\rvert$
$\overline{\Sigma^\star \overline{L_1}}$	2^{m-1} ([8])	2
$\overline{L_1^\star}^\star$	$2^{\Theta(m \log m)}$ ([139])	7
$L_1^\star \cap (\overline{L_1})^\star$	$\frac{3}{8}4^m + 2^{m-2} - 2 \cdot 3^{m-2} - m + 2$ ([117])	5
$(L_1 \cup L_2)^\star$	$2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$ ([136, 187])	2
$(L_1 \cap L_2)^\star$	$2^{mn-1} + 2^{mn-2}$ ([136])	6
$(L_1 L_2)^\star$	$2^{m+n-1} + 2^{m+n-4} - 2^{m-1} - 2^{n-1} + m + 1$ ([75])	4
$(L_1^R)^\star = (L_1^\star)^R$	2^m ([75])	3
$(L_1 \cup L_2)^R$	$2^{m+n} - 2^m - 2^n + 2$ ([150])	3
$(L_1 \cap L_2)^R$	$2^{m+n} - 2^m - 2^n + 2$ ([150])	3
$(L_1 L_2)^R$	$3 \cdot 2^{m+n-2} - 2^n + 1$ ([57, 150])	4
$L_1^\star L_2$	$5 \cdot 2^{m+n-3} - 2^{m-1} - 2^n + 1$ ([57])	4
$L_1 L_2^\star$	$(3m-1)2^{n-2}$ ([58])	3
$L_1^R L_2$	$3 \cdot 2^{m+n-2}$ ([57])	4
$L_1 L_2^R$	$m2^n - 2^{n-1} - m + 1$ ([58])	3
$L_1(L_2 \cup L_3)$	$(m-1)(2^{n+p} - 2^n - 2^p + 2) + 2^{n+p-2}$ ([56])	4
$L_1(L_2 \cap L_3)$	$m2^{np} - 2^{np-1}$ ([56])	4
$L_1^\star \cup L_2$	$3 \cdot 2^{m-2} \cdot n - n + 1$ ([79])	3
$L_1^\star \cap L_2$	$3 \cdot 2^{m-2} \cdot n - n + 1$ ([79])	3
$L_1^R \cup L_2$	$2^m \cdot n - n + 1$ ([79])	4
$L_1^R \cap L_2$	$2^m \cdot n - n + 1$ ([79])	4
$(L_1 \cup L_2)L_3$	$mn2^p - (m+n-1)2^{p-1}$ ([57])	4
$(L_1 \cap L_2)L_3$	$mn2^p - 2^{p-1}$ ([57])	4
$L_1 L_2 \cup L_3$	$(m2^n - 2^{n-1})p$ ([57])	4
$L_1 L_2 \cap L_3$	$(m2^n - 2^{n-1})p$ ([57])	3
$L_1 L_2 L_3$	$m2^{n+p} - 2^{n+p-1} - (m-1)2^{n+p-2}$ $-2^{n+p-3} - (m-1)(2^p - 1)$ ([66])	5

was improved for $m \geq 5$ and an alphabet of size at least 5 (see Table 17). For binary and ternary alphabets, the state complexity of the boundary is $\Theta(4^n)$. Combined operations with complement, extending the Birget result cited above, were studied by Moreira *et al.* [167]. There some nondeterministic state complexity upper bounds were obtained.

Gao *et al.* presented the state complexities of four combined operations: $L_1^\star \cup L_2$, $L_1^\star \cap L_2$, $L_1^R \cup L_2$, and $L_1^R \cap L_2$, where L_1 and L_2 are regular languages accepted by m and n-state DFAs, respectively. The state complexities of the four combined operations are all $n-1$ less than the mathematical composition of the state complexities of their component operations. Although the gaps are the same, the reasons causing them are different. For $L_1^\star \cup L_2$ and $L_1^\star \cap L_2$, the gap $n-1$ exists because there are $n-1$ unreachable states in the constructions of resulting DFAs. For $L_1^R \cup L_2$ and $L_1^R \cap L_2$, it is because n states are equivalent and can be merged into one in the constructions.

Cui *et al.* [58, 57] gave the state complexities of a number of combined operations including: $L_1^\star L_2$, $L_1 L_2^\star$, $L_1^R L_2$, $L_1 L_2^R$, $(L_1 \cup L_2)L_3$, $(L_1 \cap L_2)L_3$, $L_1L_2 \cup L_3$, and $L_1L_2 \cap L_3$. The state complexities of the first five combined operations are less than the corresponding mathematical compositions and the state complexities of the others are the same as the compositions. The state complexity of $L_1L_2^R$ is equal to that of catenation combined with antimorphic involution $(L_1\theta(L_2))$ in biology [58]. Up to now, the state complexities of all the combined operations composed of two basic individual operations have been obtained. These results may serve as the basis of the research on the state complexities of combined operations with more complex structures in the future.

Besides these basic combined operations, a few combined operations on k operand regular languages have also been investigated, e.g. $(\bigcup_{i=1}^{k} L_i)^\star$, $k \geq 2$. These results are summarized in Table 18. The state complexity of $L_1 \cap L_2 \cap \cdots \cap L_k$, $k \geq 2$ was shown to be $n_1 n_2 \cdots n_k$ by Birget [7], and Yu and Zhuang [207] in 1991, where L_i is a regular language accepted by an n_i-state DFA, $1 \leq i \leq k$. Ésik *et al.* [66] later extended the result to combined Boolean operations. A combined Boolean operation $f(L_1, L_2, \ldots, L_k)$ is a function which can be constructed from the projection functions and the binary union, intersection and the complementation operations by function composition, e.g. $\overline{L_1} \cup L_2 \cap L_2 \cap \cdots \cap L_k$. Its state complexity was proved to be also $n_1n_2\cdots n_k$. Ésik *et al.* [66] presented the state complexities of $L_1L_2L_3$ and $L_1L_2L_3L_4$ in the same paper. The worst-case examples for the two combined operations are modifications of the worst-case examples proposed by Yu *et al.* [208] for catenation. On the basis of these results, Gao [70] established the state complexity of $L_1L_2\cdots L_k$, which formula is too complex to figure here. Using algebraic combinatorics and Brzozowski universal witness dialects, Caron *et al.* [47] gave a recursive definition for the state complexity for multiple catenation and improved the size of the alphabet of the set of witnesses from $2k-1$ to $k+1$.

In 2012, Gao *et al.* [72] gave the state complexities of a series of combined operations composed of arbitrarily many individual operations, including: $(\bigcup_{i=1}^{k} L_i)^\star$, $(\bigcup_{i=1}^{k} L_i)^2$,

$\bigcup_{i=1}^{k} L_i^{\star}$, $\bigcap_{i=1}^{k} L_i^{\star}$, $\bigcup_{i=1}^{k} L_i^{2}$, $\bigcap_{i=1}^{k} L_i^{2}$, $\bigcup_{i=1}^{k} L_i^{R}$, and $\bigcap_{i=1}^{k} L_i^{R}$. Tight bounds were established for all these combined operations.

In Table 18, we can see that all the results on the state complexities of combined operations on k operand languages were proved with increasing alphabets. Clearly, it is comparatively easier to design worst-case examples with increasing alphabets than fixed ones. However, the most crucial reason is that it is impossible to design a worst-case example for a combined operation on arbitrary k operand languages which are over a fixed alphabet and accepted by arbitrary n_1, n_2, …, n_k-state DFAs, respectively. This is because there exists only a limited number of different DFAs with a fixed number of states if the alphabet is fixed. Therefore, when k is large enough and n_i is an arbitrary positive integer, $1 \leq i \leq k$, some of the DFAs may have the same number of states and some of them may be indeed the same according to pigeonhole principle [72]. Thus, the research on the state complexities of combined operations on k operand languages uses increasing alphabets in general.

5.2. State Complexity of Combined Operations on Prefix-free Regular Languages

Since the research history of combined operations is much shorter than that of individual operations, there remains a lot of work to be done on state complexity of combined operations for subregular language classes. The state complexities of several combined operations on prefix-free regular languages were obtained by Han *et al.* [96], in 2010. In 2015, Palmovský snd Šebej studied the star-complement-star. These results are shown in Table 19. The bounds for the results with intersection and concatenation combined with star are specially small but the surprise could be the complexity for the star-of-union, as the complexity of star is linear on prefix-free languages. But, of course, prefix-freeness is not closed under union.

5.3. Estimation and Approximation of State Complexity of Combined Operations

We can summarize at least two problems concerning the state complexities for combined operations. First, the state complexities of combined operations composed of large numbers of individual operations are extremely difficult to compute. Second, a large proportion of results that have been obtained are pretty complex and impossible to comprehend [77]. For example, Ésik et al. [66] have shown that the state complexity of the catenation for four regular languages with state complexities m, n, p, q, respectively, is

$$9(2m-1)2^{n+p+q-5} - 3(m-1)2^{p+q-2} - (2m-1)2^{n+q-2} + (m-1)2^{q} + (2m-1)2^{n-2}.$$

Clearly, in these situations, close estimations and approximations of state complexities are usually good enough to use.

5.3.1. Estimation of State Complexity of Combined Operations

An estimation method through nondeterministic state complexity to obtain the upper bound was first introduced by Salomaa and Yu [193]. Assume we are considering the

Table 18: State complexities of some combined operations on k regular languages, $k \geq 2$. Here f denotes any k-ary Boolean operation.

	Regular sc	$\|\Sigma\|$
$(\bigcup_{i=1}^{k} L_i)^\star$	$\prod_{i=1}^{k}(2^{n_i-1}-1)+2^{\sum_{j=1}^{k} n_j-k}$ ([71])	$2k+1$
$(\bigcup_{i=1}^{k} L_i)^2$	$\prod_{h=1}^{k}(n_h-1)[\prod_{i=1}^{k}(2^{n_i}-1)+1]$ $+[\prod_{j=1}^{k} n_j - \prod_{l=1}^{k}(n_l-1)]2^{\sum_{m=1}^{k} n_m-k}$ ([71])	$2k+1$
$\bigcup_{i=1}^{k} L_i^\star$	$(\frac{3}{4})^k 2^g - \sum_{i=1}^{k}[\prod_{j=1}^{i-1}(\frac{3}{4}2^{n_j}-1)\prod_{t=i+1}^{k}(\frac{3}{4}2^{n_t})]+1$ ([73])	$2k$
$\bigcap_{i=1}^{k} L_i^\star$	$(\frac{3}{4})^k 2^g - \sum_{i=1}^{k}[\prod_{j=1}^{i-1}(\frac{3}{4}2^{n_j}-1)\prod_{t=i+1}^{k}(\frac{3}{4}2^{n_t})]+1$ ([73])	$2k$
$\bigcup_{i=1}^{k} L_i^2$	$\prod_{i=1}^{k}(n_i 2^{n_i}-2^{n_i-1})$ ([72])	$2k$
$\bigcap_{i=1}^{k} L_i^2$	$\prod_{i=1}^{k}(n_i 2^{n_i}-2^{n_i-1})$ ([72])	$2k$
$\bigcup_{i=1}^{k} L_i^R$	$\prod_{i=1}^{k}(2^{n_i}-1)+1$ ([72])	$3k$
$\bigcap_{i=1}^{k} L_i^R$	$\prod_{i=1}^{k}(2^{n_i}-1)+1$ ([72])	$3k$
$f(L_1,\ldots,L_k)$	$n_1 n_2 \cdots n_k$ ([7, 66, 207])	$2k$
$L_1 L_2 \cdots L_k$	see details in [47, 70, 66, 77]	$k+1$

Table 19: State complexities of some combined operations on prefix-free regular languages.

	Prefix-Free Regular sc	$\|\Sigma\|$
$(L_1 \cup L_2)^\star$	$5 \cdot 2^{m+n-6}$ ([96])	4
$(L_1 \cap L_2)^\star$	$mn-2(m+n)+6$ ([96])	4
$(L_1 L_2)^\star$	$m+n-2$ ([96])	2
$(L_1^R)^\star = (L_1^\star)^R$	$2^{m-2}+1$ ([96])	3
$\overline{L^\star}^\star$	$2^{m-3}+2$ ([173])	2

combination of a language operation g_1 with k arguments together with operations g_2^i, $i = 1, \ldots, k$. The *nondeterministic estimation upper bound,* or *NEU-bound* for the deterministic state complexity of the combined operation $g_1(g_2^1, \ldots, g_2^k)$ is calculated as follows:

(i) Let the arguments of the operation g_2^i be DFAs A_j^i with m_j^i states, $i = 1, \ldots, k$, $j = 1, \ldots, r_i$, $r_i \geq 1$.

(ii) The nondeterministic state complexity of the combined operation is at most the composition of the individual state complexities, and hence the language

$$g_1(g_2^1(L(A_1^1), \ldots, L(A_{r_1}^1)), \ldots, g_2^k(L(A_1^k), \ldots, L(A_{r_k}^k)))$$

has an NFA with at most

$$\mathrm{nsc}(g_1)(\mathrm{nsc}(g_2^1)(m_1^1, \ldots, m_{r_1}^1), \ldots, \mathrm{nsc}(g_2^k)(m_1^k, \ldots, m_{r_k}^k))$$

states, where $\mathrm{nsc}(g)$ is the nondeterministic state complexity (as a function) of the language operation g.

(iii) Consequently, the deterministic state complexity of the combined operation $g_1(g_2^1, \ldots, g_2^k)$ is upper bounded by

$$2^{\mathrm{nsc}(g_1)(\mathrm{nsc}(g_2^1)(m_1^1, \ldots, m_{r_1}^1), \ldots, \mathrm{nsc}(g_2^k)(m_1^k, \ldots, m_{r_k}^k))} \tag{2}$$

Table 20 shows the state complexities and their corresponding NEU-bounds of the four combined operations [193]: (1) star of union, (2) star of intersection, (3) star of catenation, and (4) star of reversal. This method works well when a combined operation ends with the star operation. However, it does not work well in general for combined operations that are ended with reversal [66, 193]. For example, the state complexity of $(L(A) \cap L(B))^\star$ is $2^{m+n} - 2^m - 2^n + 2$, where A and B are m-state and n-state DFAs, respectively. But using the above method, we would obtain an estimate 2^{mn+1}. We note that in this particular case if reversal is distributed over intersection we can again recover a good estimate. Thus, it may be possible to have a

Table 20: State complexities of four combined operations and their corresponding NEU-bounds on regular languages [78].

	Regular	
	sc	NEU-bound
$(L_1 \cup L_2)^\star$	$2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$	2^{m+n+2}
$(L_1 \cap L_2)^\star$	$3/4\ 2^{mn}$	2^{mn+1}
$(L_1 L_2)^\star$	$2^{m+n-1} + 2^{m+n-4} - 2^{m-1} - 2^{n-1} + m + 1$	2^{m+n+1}
$(L_1^R)^\star$	2^m	2^{m+2}

general estimation method that takes in account algebraic properties of the considered model.[3]

5.3.2. Approximation of State Complexity of Combined Operations

Although an estimation of the state complexity of a combined operation is simpler and more convenient to use, it does not show how close it is to the state complexity. To solve this problem, the concept of approximation of state complexity was proposed by Gao and Yu [77]. The idea of approximation of state complexity comes from the notion of approximation algorithms [80, 142, 143]. A large number of polynomial-time approximation algorithms have been proposed for many NP-complete problems, e.g. the traveling-salesman problem, the set-covering problem, and the subset-sum problem, etc. Since it is considered intractable to obtain an optimal solution for an NP-complete problem, near optimal solutions obtained by approximation algorithms are often good enough to use in practice. Assume there is a maximization or a minimization problem. An approximation algorithm is said to have a ratio bound of $\rho(n)$ if for any input of size n, the cost C of the solution produced by the algorithm is within a factor of $\rho(n)$ of the cost $C^\star$ of an optimal solution [55]:

$$\max\left(\frac{C}{C^\star}, \frac{C^\star}{C}\right) \leq \rho(n).$$

The concept of approximation of state complexity is similar to that of approximation algorithms. An approximation of state complexity of an operation is a close estimation of the state complexity of the operation with a ratio bound showing the error range of the approximation [77]. In spite of similarities, there are some fundamental differences between an approximation algorithm and approximation of state complexity. The efforts in the area of approximation algorithms are in designing polynomial algorithms for NP-complete problems such that the results of the algorithms approximate the optimal results whereas the efforts in approximation of state complexity are in searching directly for the estimations of state complexities such that they are within some certain ratio bounds [77]. The aim of designing an approximation algorithm is to transform an intractable problem into one that is easier to compute and the result is not optimal but still acceptable. In comparison, an approximation of state complexity may have two different effects:

(1) it gives a reasonable estimation of a certain state complexity, with some bound, the exact value of which is difficult or impossible to compute; or
(2) it gives a simpler and more comprehensible formula that approximates a known state complexity [78].

Gao *et al.* gave a formal definition of approximation of state complexity in [78]. Let ξ be a combined operation on k regular languages. Assume that the state complexity of ξ is θ. We say that α is an approximation of the state complexity of the operation ξ with the ratio bound ρ if, for any large enough positive integers $n_1, \ldots, n_k$, which

[3] This observation was made to us by an annonymous referee.

are the numbers of states of the DFAs that accept the argument languages of the operation, respectively,

$$\max\left(\frac{\alpha(n_1,\ldots,n_k)}{\theta(n_1,\ldots,n_k)},\frac{\theta(n_1,\ldots,n_k)}{\alpha(n_1,\ldots,n_k)}\right)\le\rho(n_1,\ldots,n_k).$$

Note that in many cases, ρ is a constant. Some examples of approximation of state complexity of combined operations are shown in Table 21.

Table 21: Approximations of state complexities of six combined operations and their corresponding ratio bounds on regular languages.

	Regular Approximation	Ratio bound
$(L_1 \cup L_2)^\star$	2^{m+n+2}	≈ 8 [78]
$(L_1 \cap L_2)^\star$	2^{mn+1}	$8/3$ [78]
$(L_1L_2)^\star$	2^{m+n+1}	≈ 4 [78]
$(L_1^R)^\star$	2^{m+2}	4 [78]
$(L_1\backslash R)^\star$	$2^{m-1}+2^{m-2}$	$\frac{4}{3}$ [77]
$L_1\backslash R^\star$	2^{m+1}	$\frac{8}{3}$ [77]

6. Conclusions

In the last two decades, a huge amount of results were obtained on the operational state complexity of regular languages. Results are roughly split between: individual and combined operations; regular and different classes of subregular languages; deterministic and nondeterministic complexity; different alphabet sizes; and worst case versus average case. All this work also suggests new directions of research, as well as several open problems.

As it is evident by this survey, many results on this area are functions parametrized by some measures, mostly the state complexities of the operation arguments. Given the amount and diversity of these functions, it is useful to have a software tool that helps to structurally organize, visualize and manipulate this information. Towards this goal, a first step was taken by the development of DesCo, a Web-based information system for descriptional complexity results [180, 168]. DesCo keeps information about language classes, languages operations, models of computation, measures of complexity and complexity functions (both operational and transformational). For instance, given an operation, it is possible to obtain the complexity functions for all language classes and all complexity measures (that are registered in the database).

To obtain a witness for a tight upper bound, many authors performed experiments using computer software. The reason why some witnesses would work for several (or almost all) complexity bounds only recently has been addressed. Universal witnesses (and their variants) for operational state complexity of regular languages can be considered a major breakthrough. Conditions for a family of languages to be *universal* include also other measures as the syntactic complexity and the number of atoms. The study of necessary or sufficient conditions for the maximality of all these measures is a new direction of research. This approach was also extended to other classes of subregular languages and in some cases universal witnesses, i.e. most complex languages may not exist. It is an open problem whether this approach extends to other complexity measures, in particular to nondeterministic state complexity and transition complexity.

Besides the worst-case complexity of an operation, researchers also studied the range of possible values that can be achieved, as a function of the complexities of the arguments and the alphabet size. A magic value is a value that cannot occur (for that kind of complexity, operation and alphabet size). In general, if growing alphabet sizes are allowed no magic numbers exist (and even for binary alphabets they are rare). The distribution of possible complexity values and the density of languages (or tuples of languages) that achieve that values can also be valuable for average-case analysis.

Witnesses with alphabets of increasing size were used in the quest of magic numbers, for the state complexity of certain operations over subregular languages, and almost for all results on combined operations with an arbitrary number of operands. This questions if the alphabet size should be a parameter of the complexity under study. In particular, it should be investigated which situations cannot be characterized without increasing alphabets, and the ones for which languages with fixed alphabets can exist but are not yet known.

For binary operations, it is natural to consider that operand languages may have different alphabets. However, in state complexity results it is usually assumed, and we assumed here, that operands are over the same alphabet and DFAs are complete. In the case of incomplete DFAs, transition complexity is an interesting measure and some operational complexity results were studied [76, 154]. In those papers operational state complexity considering incomplete DFAs was also presented. Note that, an incomplete DFA can be completed by adding at most one more state. For binary operations, considering complete DFAs with different alphabets corresponds to consider incomplete DFAs over the union of the alphabets. State complexity of (basic) binary operations over languages with different alphabets was studied by Brzozowski [19], and called *unrestricted (state) complexity*. Results in that case can be slightly different from the ones presented in this survey.

For many automata applications, a major direction of research is average-case state complexity. An essential question for average results is the probability distribution that is chosen for the models. The few results that exist use a uniform distribution, and even in this case the problem is very difficult. Recently, using the framework of analytic combinatorics, some average-case results were obtained for the size of NFAs equivalent to a given regular expression [171, 10, 11, 12]. It is also worthwhile to mention the average-case computational complexity analysis of the Brzozowski minimization algorithm carried on by Felice and Nicaud [67, 69]. This work can be specially relevant for the operational state complexity because the authors give some characterizations of the state complexity of reversal. Another approach for average-case analysis is to consider experimental results based on samples of uniformly random generated automata. There are some random generators for non-isomorphic DFAs [1, 5, 68]. Uniform random generators for non-isomorphic NFAs using Markov chains were presented in [97]. However for NFAs, the fact that there is no known generic polynomial algorithm for graph isomorphism, the problem seems unfeasible in general. A promising line of research is to consider random generators for interesting subclasses of NFAs.

7. Acknowledgements

This work was partially supported by the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT under projects PEst-C/MAT/UI0144/ 2013 and PTDC/EIA-CCO/101904/2008. The writing of this paper started in the fall of 2010 when Nelma Moreira and Rogério Reis were visiting Sheng Yu at University of Western Ontario. When Sheng passed away in January 2012, the structure of the paper was already in the current form. We deeply thank Kai Salomaa for his encouragement, suggestions, and corrections. We also thank Davide Nabais and Eva Maia for proofreading. Comments and criticisms of the anonymous referee were most valuable for improving the final version of the paper.

References

[1] M. Almeida, N. Moreira, R. Reis, Enumeration and generation with a string automata representation. *Theoret. Comput. Sci.* **387** (2007) 2, 93–102.

[2] T. Ang, J. A. Brzozowski, Languages Convex with Respect to Binary Relations, and Their Closure Properties. *Acta Cybern.* **19** (2009) 2, 445–464.

[3] F. Bassino, L. Giambruno, C. Nicaud, The Average State Complexity of Rational Operations on Finite Languages. *Int. J. Found. Comput. Sci.* **21** (2010) 4, 495–516.

[4] F. Bassino, L. Giambruno, C. Nicaud, Complexity of Operations on Cofinite Languages. In: A. López-Ortiz (ed.), *9th LATIN 2010*. LNCS 6034, Springer, 2010, 222–233.

[5] F. Bassino, C. Nicaud, Enumeration and random generation of accessible automata. *Theoret. Comput. Sci.* **381** (2007) 1-3, 86–104.

[6] J. Berstel, D. Perrin, C. Reutenauer, *Codes and Automata*. Number 129 in Encyclopedia of Mathematics and its Applications, CUP, 2009.

[7] J.-C. Birget, Intersection of Regular Languages and State Complexity. *ACM SIGACT News* **22** (1991) 2.

[8] J.-C. Birget, The State Complexity of $\overline{\Sigma^{\star}\overline{L}}$ and its Connection with Temporal Logic. *Inf. Process. Lett.* **58** (1996) 4, 185–188.

[9] H. Bordihn, M. Holzer, M. Kutrib, Determination of Finite Automata Accepting Subregular Languages. *Theor. Comput. Sci.* **410** (2009) 35, 3209–3222.

[10] S. BRODA, A. MACHIAVELO, N. MOREIRA, R. REIS, On the average state complexity of partial derivative automata. *Int. J. Found. Comput. Sci.* **22** (2011) 7, 1593–1606.

[11] S. BRODA, A. MACHIAVELO, N. MOREIRA, R. REIS, On the Average Size of Glushkov and Partial Derivative Automata. *Int. J. Found. Comput. Sci.* **23** (2012) 5, 969–984.

[12] S. BRODA, A. MACHIAVELO, N. MOREIRA, R. REIS, A Hitchhiker's Guide to Descriptional Complexity through Analytic Combinatorics. *Theor. Comput. Sci.* **528** (2014), 85–100.

[13] J. A. BRZOZOWSKI, Derivatives of Regular Expressions. *J. ACM* **11** (1964) 4, 481–494.

[14] J. A. BRZOZOWSKI, Quotient Complexity of Regular Languages. In: J. DASSOW, G. PIGHIZZINI, B. TRUTHE (eds.), *11th DCFS 2009*. EPTCS 3, 2009, 17–28.

[15] J. A. BRZOZOWSKI, Complexity in Convex Languages. In: A. H. DEDIU, H. FERNAU, C. MARTÍN-VIDE (eds.), *4th LATA 2010*. LNCS 6031, Springer, 2010, 1–15.

[16] J. A. BRZOZOWSKI, Quotient Complexity of Regular Languages. *Journal of Automata, Languages and Combinatorics* **15** (2010) 1/2, 71–89.

[17] J. A. BRZOZOWSKI, In Search of Most Complex Regular Languages. In: N. MOREIRA, R. REIS (eds.), *17th CIAA 2012*. LNCS 7381, Springer, 2012, 5–24.

[18] J. A. BRZOZOWSKI, In Search of Most Complex Regular Languages. *Int. J. Found. Comput. Sci.* **24** (2013) 6, 691–708.

[19] J. A. BRZOZOWSKI, Unrestricted State Complexity of Binary Operations on Regular Languages. In: C. CÂMPEANU, F. MANEA, J. SHALLIT (eds.), *18th DCFS 2016*. LNCS 9777, Springer, 2016, 60–72.

[20] J. A. BRZOZOWSKI, G. DAVIES, Maximally Atomic Languages. In: Z. ÉSIK, Z. FÜLÖP (eds.), *14th AFL 2014*. 2014, 151–161.

[21] J. A. BRZOZOWSKI, G. DAVIES, Most Complex Regular Right-Ideal Languages. In: H. JÜRGENSEN, J. KARHUMÄKI, A. OKHOTIN (eds.), *16th DCFS 2014*. LNCS 8614, Springer, 2014, 90–101.

[22] J. A. BRZOZOWSKI, S. DAVIES, B. Y. V. LIU, Most Complex Regular Ideals. *CoRR* **abs/1511.00157** (2015).

[23] J. A. BRZOZOWSKI, G. JIRÁSKOVÁ, B. LI, Quotient Complexity of Ideal Languages. In: A. LÓPEZ-ORTIZ (ed.), *9th LATIN 2010*. LNCS 6034, Springer, 2010, 208–221.

[24] J. A. BRZOZOWSKI, G. JIRÁSKOVÁ, B. LI, J. SMITH, Quotient Complexity of Bifix-, Factor-, and Subword-Free Regular Languages. In: P. DÖMÖSI, S. IVÁN (eds.), *13th AFL 2011*. 2011, 123–137.

[25] J. A. BRZOZOWSKI, G. JIRÁSKOVÁ, B. LIU, A. RAJASEKARAN, M. SZYKUŁA, On the State Complexity of the Shuffle of Regular Languages. In: C. CÂMPEANU, F. MANEA, J. SHALLIT (eds.), *18th DCFS 2016*. LNCS 9777, Springer, 2016, 73–86.

[26] J. A. BRZOZOWSKI, G. JIRÁSKOVÁ, C. ZOU, Quotient Complexity of Closed Languages. In: F. M. ABLAYEV, E. W. MAYR (eds.), *5th CSR 2010*. LNCS 6072, Springer, 2010, 84–95.

[27] J. A. BRZOZOWSKI, G. JIRÁSKOVÁ, C. ZOU, Quotient Complexity of Closed Languages. *Theory Comput. Syst.* **54** (2014) 2, 277–292.

[28] J. A. BRZOZOWSKI, S. KONSTANTINIDIS, State-complexity Hierarchies of Uniform Languages of Alphabet-size Length. *Theor. Comput. Sci.* **410** (2009) 35, 3223–3235.

[29] J. A. Brzozowski, B. Li, Syntactic Complexities of Some Classes of Star-Free Languages. In: M. Kutrib, N. Moreira, R. Reis (eds.), *14th DCFS 2012*. LNCS 7386, Springer, 2012, 117–129.

[30] J. A. Brzozowski, B. Li, Syntactic Complexity of R- and J-Trivial Regular Languages. In: H. Jürgensen, R. Reis (eds.), *15th DCFS 2013*. LNCS 8031, Springer, 2013, 160–171.

[31] J. A. Brzozowski, B. Li, Y. Ye, Syntactic complexity of prefix-, suffix-, bifix-, and factor-free regular languages. *Theor. Comput. Sci.* **449** (2012), 37–53.

[32] J. A. Brzozowski, B. Liu, Quotient Complexity of Star-Free Languages. In: P. Dömösi, S. Iván (eds.), *13th AFL 2011*. 2011, 138–152.

[33] J. A. Brzozowski, J. Shallit, Z. Xu, Decision Problems for Convex Languages. In: A. H. Dediu, A.-M. Ionescu, C. Martín-Vide (eds.), *3rd LATA 2009*. LNCS 5457, Springer, 2009, 247–258.

[34] J. A. Brzozowski, C. Sinnamon, Complexity of Prefix-Convex Regular Languages. *CoRR* **abs/1605.06697** (2016).

[35] J. A. Brzozowski, C. Sinnamon, Unrestricted State Complexity of Binary Operations on Regular and Ideal Languages. *CoRR* **arXiv:1602.01387v3** (2016).

[36] J. A. Brzozowski, M. Szykuła, Upper Bounds on Syntactic Complexity of Left and Two-Sided Ideals. In: A. M. Shur, M. V. Volkov (eds.), *18th DLT 2014*. LNCS 8633, Springer, 2014, 13–24.

[37] J. A. Brzozowski, M. Szykuła, Complexity of Suffix-Free Regular Languages. In: A. Kosowski, I. Walukiewicz (eds.), *20th FCT 2015*. LNCS 9210, Springer, 2015, 146–159.

[38] J. A. Brzozowski, M. Szykuła, Large Aperiodic Semigroups. *Int. J. Found. Comput. Sci.* **26** (2015) 7, 913–932.

[39] J. A. Brzozowski, H. Tamm, Theory of Átomata. In: G. Mauri, A. Leporati (eds.), *15th DLT 2011*. LNCS 6795, Springer, 2011, 105–116.

[40] J. A. Brzozowski, H. Tamm, Quotient Complexities of Atoms of Regular Languages. In: H.-C. Yen, O. H. Ibarra (eds.), *16th DLT 2012*. LNCS 7410, Springer, 2012, 50–61.

[41] J. A. Brzozowski, H. Tamm, Theory of Átomata. *Theor. Comput. Sci.* **539** (2014), 13–27.

[42] J. A. Brzozowski, Y. Ye, Syntactic Complexity of Ideal and Closed Languages. In: G. Mauri, A. Leporati (eds.), *15th DLT 2011*. LNCS 6795, Springer, 2011, 117–128.

[43] C. Câmpeanu, K. Culik II, K. Salomaa, S. Yu, State Complexity of Basic Operations on Finite Languages. In: O. Boldt, H. Jürgensen (eds.), *4th WIA'99*. LNCS 2214, Springer, 2001, 60–70.

[44] C. Câmpeanu, W. H. Ho, The Maximum State Complexity for Finite Languages. *Journal of Automata, Languages and Combinatorics* **9** (2004) 2/3, 189–202.

[45] C. Câmpeanu, S. Konstantinidis, State Complexity of the Subword Closure Operation with Applications to DNA Coding. *Int. J. Found. Comput. Sci.* **19** (2008) 5, 1099–1112.

[46] C. Câmpeanu, K. Salomaa, S. Yu, Tight Lower Bound for the State Complexity of Shuffle of Regular Languages. *Journal of Automata, Languages and Combinatorics* **7** (2002) 3, 303–310.

[47] P. Caron, J. Luque, B. Patrou, State complexity of multiple catenation. *CoRR* **abs/1607.04031** (2016).

[48] K. Cevorová, Kleene Star on Unary Regular Languages. In: H. Jürgensen, R. Reis (eds.), *15th DCFS 2013*. LNCS 8031, Springer, 2013, 277–288.

[49] K. Cevorová, Square on Ideal, Closed and Free Languages. In: J. Shallit, A. Okhotin (eds.), *17th DCFS 2015*. LNCS 9118, Springer, 2015, 70–80.

[50] K. Cevorová, Square on Closed Languages. In: H. Bordihn, R. Freund, B. N. andGyörgy Vaszil (eds.), *7th NCMA 2016*. 321, Österreichische Computer Gesellschaft, 2016, 121–130.

[51] K. Cevorová, G. Jirásková, I. Krajnáková, On the Square of Regular Languages. In: M. Holzer, M. Kutrib (eds.), *19th CIAA 2014*. LNCS 8587, Springer, 2014, 136–147.

[52] M. Chrobak, Finite Automata and Unary Languages. *Theor. Comput. Sci.* **47** (1986) 3, 149–158.

[53] M. Chrobak, Errata to: "Finite automata and unary languages". *Theor. Comput. Sci.* **302** (2003) 1-3, 497–498.

[54] J. H. Conway, *Regular Algebra and Finite Machines*. William Clowes & Sons, Great Britain, 1971.

[55] T. H. Cormen, C. E. Leiserson, R. L. Rivest, *Introduction to algorithms*. The MIT Press, Massachusetts, USA, 1990.

[56] B. Cui, Y. Gao, L. Kari, S. Yu, State complexity of two combined operations: catenation-union and catenation-intersection. *Int. J. Found. Comput. Sci.* **22** (2011) 8, 1797–1812.

[57] B. Cui, Y. Gao, L. Kari, S. Yu, State complexity of combined operations with two basic operations. *Theor. Comput. Sci.* **437** (2012), 82–102.

[58] B. Cui, Y. Gao, L. Kari, S. Yu, State complexity of two combined operations: catenation-star and catenation-reversal. *Int. J. Found. Comput. Sci.* **23** (2012) 1, 51–56.

[59] M. Daley, M. Domaratzki, K. Salomaa, Orthogonal Concatenation: Language Equations and State Complexity. *J. UCS* **16** (2010) 5, 653–675.

[60] A. de Luca, S. Varricchio, Some combinatorial properties of factorial languages. In: R. Capocelli (ed.), *Sequences*. Springer, 1990, 258–266.

[61] M. Domaratzki, State Complexity of Proportional Removals. *Journal of Automata, Languages and Combinatorics* **7** (2002) 4, 455–468.

[62] M. Domaratzki, A. Okhotin, State Complexity of Power. *Theor. Comput. Sci.* **410** (2009) 24-25, 2377–2392.

[63] M. Domaratzki, K. Salomaa, State Complexity of Shuffle on Trajectories. *Journal of Automata, Languages and Combinatorics* **9** (2004) 2/3, 217–232.

[64] K. Ellul, *Descriptional Complexity Measures of Regular Languages*. Master's thesis, University of Waterloo, 2003.

[65] H.-S. Eom, Y.-S. Han, K. Salomaa, State Complexity of k-Union and k-Intersection for Prefix-Free Regular Languages. In: H. Jürgensen, R. Reis (eds.), *15th DCFS 2013*. LNCS 8031, Springer, 2013, 78–89.

[66] Z. Ésik, Y. Gao, G. Liu, S. Yu, Estimation of state complexity of combined operations. *Theor. Comput. Sci.* **410** (2009) 35, 3272–3280.

[67] S. D. Felice, C. Nicaud, Brzozowski Algorithm Is Generically Super-Polynomial for Deterministic Automata. In: M.-P. Béal, O. Carton (eds.), *17th DLT 2013*. LNCS 7907, Springer, 2013, 179–190.

[68] S. D. Felice, C. Nicaud, Random Generation of Deterministic Acyclic Automata Using the Recursive Method. In: A. A. Bulatov, A. M. Shur (eds.), *8th CSR 2013*. LNCS 7913, Springer, 2013, 88–99.

[69] S. D. Felice, C. Nicaud, On the Average Complexity of Brzozowski's Algorithm for Deterministic Automata with a Small Number of Final States. In: A. M. Shur, M. V. Volkov (eds.), *18th DLT 2014*. LNCS 8633, Springer, 2014, 25–36.

[70] Y. Gao, *Advanced topics on state complexity of combined operations*. Ph.D. thesis, University of Western Ontario, 2010.

[71] Y. Gao, L. Kari, State complexity of star and square of union of k regular languages. In: M. Kutrib, N. Moreira, R. Reis (eds.), *14th DCFS 2012*. LNCS 7386, Springer, 2012, 155–168.

[72] Y. Gao, L. Kari, S. Yu, State complexity of union and intersection of square and reversal on k regular languages. *Theor. Comput. Sci.* **454** (2012), 164–171.

[73] Y. Gao, L. Kari, S. Yu, State complexity of union and intersection of star on k regular languages. *Theor. Comput. Sci.* **429** (2012), 98–107.

[74] Y. Gao, N. Moreira, R. Reis, S. Yu, A Survey on Operational State Complexity. *Journal of Automata, Languages and Combinatorics* **21** (2016) 4, 251–310. URL `https://doi.org/10.25596/jalc-2016-251`

[75] Y. Gao, K. Salomaa, S. Yu, The state complexity of two combined operations: star of catenation and star of reversal. *Fundam. Inform.* **83** (2008) 1-2, 75–89.

[76] Y. Gao, K. Salomaa, S. Yu, Transition Complexity of Incomplete DFAs. *Fundam. Inform.* **110** (2011) 1-4, 143–158.

[77] Y. Gao, S. Yu, State complexity approximation. In: J. Dassow, G. Pighizzini, B. Truthe (eds.), *11th DCFS 2009*. 2009, 163–174.

[78] Y. Gao, S. Yu, State complexity and approximation. *Int. J. Found. Comput. Sci.* **23** (2012) 5, 1085–1098.

[79] Y. Gao, S. Yu, State complexity of combined operations with union, intersection, star and reversal. *Fundam. Inform.* **116** (2012) 1-4, 79–92.

[80] M. R. Garey, R. L. Graham, J. D. Ullman, Worst-case analysis of memory allocation algorithms. In: *4th ACM Symposium on the Theory of Computing*. 1972, 143–150.

[81] V. Geffert, (Non)determinism and the Size of One-way Finite Automata. In: C. Mereghetti, B. Palano, G. Pighizzini, D. Wotschke (eds.), *7th DCFS 2005*. University of Milano, Italy, 2005, 23–37.

[82] V. Geffert, Magic Numbers in the State Hierarchy of Finite Automata. *Inf. Comput.* **205** (2007) 11, 1652–1670.

[83] V. Geffert, State Hierarchy for One-Way Finite Automata. *Journal of Automata, Languages and Combinatorics* **12** (2007) 1-2, 139–145.

[84] A. Gill, L. T. Kou, Multiple-entry Finite Automata. *J. Comput. System Sci.* **9** (1974), 1–19.

[85] I. Glaister, J. Shallit, A Lower Bound Technique for the Size of Nondeterministic Finite Automata. *Inf. Process. Lett.* **59** (1996) 2, 75–77.

[86] D. Goc, A. Palioudakis, K. Salomaa, Nondeterministic State Complexity of Proportional Removals. In: H. Jürgensen, R. Reis (eds.), *15th DCFS 2013*. LNCS 8031, Springer, 2013, 102–111.

[87] H. Gruber, M. Holzer, On the Average State and Transition Complexity of Finite Languages. *Theor. Comput. Sci.* **387** (2007) 2, 155–166.

[88] H. Gruber, M. Holzer, M. Kutrib, The Size of Higman-Haines Sets. *Theor. Comput. Sci.* **387** (2007) 2, 167–176.

[89] H. Gruber, M. Holzer, M. Kutrib, More on the Size of Higman-Haines Sets: Effective Constructions. *Fundam. Inform.* **91** (2009) 1, 105–121.

[90] L. H. Haines, On Free Monoids Partially Ordered by Embedding. *J. Combin. Theory* **6** (1969), 94–98.

[91] Y.-S. Han, K. Salomaa, State Complexity of Union and Intersection of Finite Languages. *Int. J. Found. Comput. Sci.* **19** (2008) 3, 581–595.

[92] Y.-S. Han, K. Salomaa, State Complexity of Basic Operations on Suffix-free Regular Languages. *Theor. Comput. Sci.* **410** (2009) 27-29, 2537–2548.

[93] Y.-S. Han, K. Salomaa, Nondeterministic State Complexity for Suffix-Free Regular Languages. In: I. McQuillan, G. Pighizzini (eds.), *12th DCFS 2010*. EPTCS 31, 2010, 189–196.

[94] Y.-S. Han, K. Salomaa, D. Wood, Nondeterministic State Complexity of Basic Operations for Prefix-Free Regular Languages. *Fundam. Inform.* **90** (2009) 1-2, 93–106.

[95] Y.-S. Han, K. Salomaa, D. Wood, Operational State Complexity of Prefix-Free Regular Languages. In: Z. Ésik, Z. Fülöp (eds.), *Automata, Formal Languages, and Related Topics - Dedicated to Ferenc Gécseg on the occasion of his 70th birthday*. Institute of Informatics, University of Szeged, Hungary, 2009, 99–115.

[96] Y.-S. Han, K. Salomaa, S. Yu, State complexity of combined operations for prefix-free regular languages. In: A. H. Dediu, A.-M. Ionescu, C. Martín-Vide (eds.), *3rd LATA 2009*. LNCS 5457, Springer, 2009, 398–409.

[97] P. Héam, J. Joly, On the Uniform Random Generation of Non Deterministic Automata Up to Isomorphism. In: F. Drewes (ed.), *20th CIAA, 2015*. LNCS 9223, Springer, 2015, 140–152.

[98] A. Herrmann, M. Kutrib, A. Malcher, M. Wendlandt, Descriptional Complexity of Bounded Regular Languages. In: C. Câmpeanu, F. Manea, J. Shallit (eds.), *18th DCFS 2016*. LNCS 9777, Springer, 2016, 138–152.

[99] M. Holzer, S. Jakobi, Chop Operations and Expressions: Descriptional Complexity Considerations. In: G. Mauri, A. Leporati (eds.), *15th DLT 2011*. LNCS 6795, Springer, 2011, 264–275.

[100] M. Holzer, S. Jakobi, State Complexity of Chop Operations on Unary and Finite Languages. In: M. Kutrib, N. Moreira, R. Reis (eds.), *14th DCFS 2012*. LNCS 7386, Springer, 2012, 169–182.

[101] M. Holzer, S. Jakobi, M. Kutrib, The Chop of Languages. In: P. Dömösi, S. Iván (eds.), *13th AFL 2011*. 2011, 197–210.

[102] M. Holzer, S. Jakobi, M. Kutrib, The Magic Number Problem for Subregular Language Families. *Int. J. Found. Comput. Sci.* **23** (2012) 1, 115–131.

[103] M. Holzer, B. König, On deterministic finite automata and syntactic monoid size. *Theor. Comput. Sci.* **327** (2004) 3, 319–347.

[104] M. Holzer, M. Kutrib, State Complexity of Basic Operations on Nondeterministic Finite Automata. In: J.-M. Champarnaud, D. Maurel (eds.), *7th CIAA 2002*. LNCS 2608, Springer, 2003, 148–157.

[105] M. Holzer, M. Kutrib, Unary Language Operations and Their Nondeterministic State Complexity. In: M. Ito, M. Toyama (eds.), *6th DLT 2002*. LNCS 2450, Springer, 2003, 162–172.

[106] M. Holzer, M. Kutrib, Descriptional and Computational Complexity of Finite Automata. In: A. H. Dediu, A.-M. Ionescu, C. Martín-Vide (eds.), *3rd LATA 2009*. LNCS 5457, Springer, 2009, 23–42.

[107] M. Holzer, M. Kutrib, Nondeterministic Finite Automata - Recent Results on the Descriptional and Computational Complexity. *Int. J. Found. Comput. Sci.* **20** (2009) 4, 563–580.

[108] M. Holzer, M. Kutrib, Descriptional and Computational complexity of finite automata - A survey. *Inf. Comput.* **209** (2011) 3, 456–470.

[109] M. Holzer, M. Kutrib, K. Meckel, Nondeterministic State Complexity of Star-Free Languages. In: B. Bouchou-Markhoff, P. Caron, J.-M. Champarnaud, D. Maurel (eds.), *16th CIAA 2011*. LNCS 6807, Springer, 2011, 178–189.

[110] M. Holzer, M. Kutrib, K. Meckel, Nondeterministic State Complexity of Star-Free Languages. *Theor. Comput. Sci.* **450** (2012), 68–80.

[111] M. Hospodár, G. Jirásková, P. Mlynárcik, Nondeterministic Complexity of Operations on Closed and Ideal Languages. In: Y. Han, K. Salomaa (eds.), *21st CIAA 2016*. LNCS 9705, Springer, 2016, 125–137.

[112] M. Hricko, G. Jirásková, A. Szabari, Union and Intersection of Regular Languages and Descriptional Complexity. In: C. Mereghetti, B. Palano, G. Pighizzini, D. Wotschke (eds.), *7th DCFS 2005*. University of Milano, Milano, Italy, 2005, 170–181.

[113] J. Hromkovic, *Communication Complexity and Parallel Computing*. Springer, 1997.

[114] J. Hromkovic, Descriptional Complexity of Finite Automata: Concepts and Open Problems. *Journal of Automata, Languages and Combinatorics* **7** (2002) 4, 519–531.

[115] K. Iwama, Y. Kambayashi, K. Takaki, Tight Bounds on the Number of States of DFAs that are Equivalent to n-state NFAs. *Theor. Comput. Sci.* **237** (2000) 1-2, 485–494.

[116] K. Iwama, A. Matsuura, M. Paterson, A Family of NFA's Which Need $2^n - \alpha$ Deterministic States. In: M. Nielsen, B. Rovan (eds.), *25th MFCS 2000*. LNCS 1893, Springer, 2000, 436–445.

[117] J. Jirásek, G. Jirásková, On the boundary of regular languages. *Theor. Comput. Sci.* **578** (2015), 42–57.

[118] J. Jirásek, G. Jirásková, M. Krausová, P. Mlynárcik, J. Sebej, Prefix-Free Languages: Right Quotient and Reversal. In: H. Jürgensen, J. Karhumäki, A. Okhotin (eds.), *16th DCFS 2014*. LNCS 8614, Springer, 2014, 210–221.

[119] J. Jirásek, G. Jirásková, A. Szabari, State Complexity of Concatenation and Complementation of Regular Languages. *Int. J. Found. Comput. Sci.* **16** (2005) 3, 511–529.

[120] J. Jirásek, G. Jirásková, A. Szabari, Deterministic Blow-ups of Minimal Nondeterministic Finite Automata over a Fixed Alphabet. *Int. J. Found. Comput. Sci.* **19** (2008) 3, 617–631.

[121] G. Jirásková, Note on Minimal Finite Automata. In: J. Sgall, A. Pultr, P. Kolman (eds.), *26th MFCS 2001*. LNCS 2136, Springer, 2001, 421–431.

[122] G. Jirásková, State Complexity of some Operations on Binary Regular Languages. *Theor. Comput. Sci.* **330** (2005) 2, 287–298.

[123] G. Jirásková, On the State Complexity of Complements, Stars, and Reversals of Regular Languages. In: M. Ito, M. Toyama (eds.), *12th DLT 2008*. LNCS 5257, Springer, 2008, 431–442.

[124] G. Jirásková, Concatenation of Regular Languages and Descriptional Complexity. In: A. E. Frid, A. Morozov, A. Rybalchenko, K. W. Wagner (eds.), *4th CSR 2009*. LNCS 5675, Springer, 2009, 203–214.

[125] G. Jirásková, Magic Numbers and Ternary Alphabet. In: V. Diekert, D. Nowotka (eds.), *13th DLT 2009*. LNCS 5583, Springer, 2009, 300–311.

[126] G. Jirásková, Concatenation of Regular Languages and Descriptional Complexity. *Theory Comput. Syst.* **49** (2011) 2, 306–318.

[127] G. Jirásková, Magic Numbers and Ternary Alphabet. *Int. J. Found. Comput. Sci.* **22** (2011) 2, 331–344.

[128] G. Jirásková, M. Krausová, Complexity in Prefix-Free Regular Languages. *Electron. Proc. Theor. Comput. Sci.* **31** (2010), 197–204.

[129] G. Jirásková, T. Masopust, Complexity in Union-Free Regular Languages. In: Y. Gao, H. Lu, S. Seki, S. Yu (eds.), *14th DLT 2010*. LNCS 6224, Springer, 2010, 255–266.

[130] G. Jirásková, T. Masopust, Complexity in Union-Free Regular Languages. *Int. J. Found. Comput. Sci.* **22** (2011) 7, 1639–1653.

[131] G. Jirásková, T. Masopust, State Complexity of Projected Languages. In: M. Holzer, M. Kutrib, G. Pighizzini (eds.), *13th DCFS 2011*. LNCS 6808, Springer, 2011, 198–211.

[132] G. Jirásková, T. Masopust, On the State and Computational Complexity of the Reverse of Acyclic Minimal DFAs. In: N. Moreira, R. Reis (eds.), *17th CIAA 2012*. LNCS 7381, Springer, 2012, 229–239.

[133] G. Jirásková, T. Masopust, On the State Complexity of the Reverse of R- and J-Trivial Regular Languages. In: H. Jürgensen, R. Reis (eds.), *15th DCFS 2013*. LNCS 8031, Springer, 2013, 136–147.

[134] G. Jirásková, P. Mlynárcik, Complement on Prefix-Free, Suffix-Free, and Non-Returning NFA Languages. In: H. Jürgensen, J. Karhumäki, A. Okhotin (eds.), *16th DCFS 2014*. LNCS 8614, Springer, 2014, 222–233.

[135] G. Jirásková, A. Okhotin, State Complexity of Cyclic Shift. *Rairo - Theor. Inf. and Applic.* **42** (2008), 335–360.

[136] G. Jirásková, A. Okhotin, On the State Complexity of Star of Union and Star of Intersection. *Fundam. Inform.* **109** (2011) 2, 161–178.

[137] 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.), *1st NCMA 2009*. books@ocg.at 256, Austrian Computer Society, 2009, 151–166.

[138] G. Jirásková, M. Palmovský, J. Sebej, Kleene Closure on Regular and Prefix-Free Languages. In: M. Holzer, M. Kutrib (eds.), *19th CIAA 2014*. LNCS 8587, Springer, 2014, 226–237.

[139] G. Jirásková, J. Shallit, The State Complexity of Star-Complement-Star. In: H.-C. Yen, O. H. Ibarra (eds.), *16th DLT 2012*. LNCS 7410, Springer, 2012, 380–391.

[140] G. Jirásková, J. Sěbej, Note on Reversal of Binary Regular Languages. In: M. Holzer, M. Kutrib, G. Pighizzini (eds.), *13th DCFS 2011*. LNCS 6808, Springer, 2011, 212–221.

[141] G. Jirásková, A. Szabari, J. Sebej, The Complexity of Languages Resulting from the Concatenation Operation. In: C. Câmpeanu, F. Manea, J. Shallit (eds.), *18th DCFS 2016*. LNCS 9777, Springer, 2016, 153–167.

[142] D. S. Johnson, Fast allocation algorithms. In: *13th Annual IEEE Symposium on Switching and Automata Theory*. IEEE Computer Society, 1972, 144–154.

[143] D. S. Johnson, *Near-optimal bin packing algorithms*. Ph.D. thesis, Massachusetts Institute of Technology, 1993.

[144] H. Jürgensen, S. Konstantinidis, Codes. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages*. 1, Springer, 1997, 511–607.

[145] J.-Y. Kao, N. Rampersad, J. Shallit, On NFAs where all states are final, initial, or both. *Theor. Comput. Sci.* **410** (2009) 47-49, 5010–5021.

[146] P. Karandikar, P. Schnoebelen, On the State Complexity of Closures and Interiors of Regular Languages with Subwords. In: H. Jürgensen, J. Karhumäki, A. Okhotin (eds.), *16th DCFS 2014*. LNCS 8614, Springer, 2014, 234–245.

[147] B. Krawetz, J. Lawrence, J. Shallit, State complexity and the monoid of transformations of a finite set. *Int. J. Found. Comput. Sci.* **16** (2005) 3, 547–563.

[148] D. Krieger, A. Miller, N. Rampersad, B. Ravikumar, J. Shallit, Decimations of languages and state complexity. *Theor. Comput. Sci.* **410** (2009) 24-25, 2401–2409.

[149] E. L. Leiss, Succinct Representation of Regular Languages by Boolean Automata II. *Theor. Comput. Sci.* **38** (1985), 133–136.

[150] G. Liu, C. Martin-Vide, A. Salomaa, S. Yu, State complexity of basic language operations combined with reversal. *Inf. Comput.* **206** (2008) 9-10, 1178–1186.

[151] H. Lu, S. Yu, Are statecharts finite automata? In: S. Maneth (ed.), *14th CIAA 2009*. LNCS 5642, Springer, 2009, 258–261.

[152] O. B. LUPANOV, A comparison of two types of finite sources. *Problemy Kibernetiki* **9** (1963), 321–326. In Russian. German translation: Über den Vergleich zweier Typen endlicher Quellen. Probleme der Kybernetik 6 (1966), 328-335.

[153] Y. I. LYUBICH, Estimates for optimal determinization of nondeterministic autonomous automata. *Sibirskii Matematichskii Zhurnal* **5** (1964), 337–355. In Russian.

[154] E. MAIA, N. MOREIRA, R. REIS, Incomplete Transition Complexity on Finite and Infinite Regular Languages. *Information and Computation* **244** (2015), 1–22.

[155] R. MANDL, Precise bounds associated with the subset construction on various classes of nondeterministic finite automata. In: *Princeton Conference on Information and System Sciences*. 1973, 263–267.

[156] A. N. MASLOV, Estimates of the number of states of finite automata. *Dokllady Akademii Nauk SSSR* **194** (1970), 1266–1268. In Russian. English translation in Soviet Mathematics Doklady, 11, 1373-1375 (1970).

[157] A. MATEESCU, G. ROZENBERG, A. SALOMAA, Shuffle on Trajectories: Syntactic Constraints. *Theor. Comput. Sci.* **197** (1998) 1-2, 1–56.

[158] R. MCNAUGHTON, S. PAPERT, *Counter-free Automata*. Research Monographs 65, The MIT Press, 1971.

[159] F. MERA, G. PIGHIZZINI, Complementing unary nondeterministic automata. *Theor. Comput. Sci.* **330** (2005) 2, 349–360.

[160] C. MEREGHETTI, G. PIGHIZZINI, Optimal Simulations between Unary Automata. *SIAM J. Comput.* **30** (2000) 6, 1976–1992.

[161] A. R. MEYER, M. J. FISCHER, Economy of Description by Automata, Grammars, and Formal Systems. In: *Conference Record 1971 12th Annual Symposium on Switching and Automata Theory*. IEEE Society Press, Los Alamitos, 1971, 188–191.

[162] B. G. MIRKIN, On Dual Automata. *Kibernetika* **2** (1966), 7–10. In Russian. English translation in Cybernetics 2, 6-9 (1966).

[163] P. MLYNÁRCIK, Complement on Free and Ideal Languages. In: J. SHALLIT, A. OKHOTIN (eds.), *17th DCFS 2015*. LNCS 9118, Springer, 2015, 185–196.

[164] M. MOHRI, On some applications of finite-state automata theory to natural language processing. *Natural Language Engineering* **2** (1996), 61–80.

[165] F. R. MOORE, On the Bounds for State-Set Size in the Proofs of Equivalence Between Deterministic, Nondeterministic, and Two-Way Finite Automata. *IEEE Transactions on computers* **2** (1971), 1211–1214.

[166] D. MORALES, F. ALMEIDA, C. RODRIGUEZ, J. RODA, I. COLOMA, A. DELGADO, Parallel dynamic programming and automata theory. *Parallel Computing* **26** (2000), 113–134.

[167] N. MOREIRA, G. PIGHIZZINI, R. REIS, Universal Disjunctive Concatenation and Star. In: J. SHALLIT, A. OKHOTIN (eds.), *17th DCFS 2015*. LNCS 9118, Springer, 2015, 197–208.

[168] D. NABAIS, N. MOREIRA, R. REIS, *DesCo: a Knowledge Based System for Descriptional Complexity of Formal Languages*. Technical Report DCC-2013-12, DCC-FC Universidade do Porto, 2013.

[169] C. Nicaud, Average State Complexity of Operations on Unary Automata. In: M. Kutylowski, L. Pacholski, T. Wierzbicki (eds.), *24th MFCS'99*. LNCS 1672, Springer, 1999, 231–240.

[170] C. Nicaud, *Étude du comportement en moyenne des automates finis et des languages rationels*. Ph.D. thesis, Université de Paris 7, 2000.

[171] C. Nicaud, On the Average Size of Glushkov's Automata. In: A. H. Dediu, A.-M. Ionescu, C. Martín-Vide (eds.), *3rd International Conference on Language and Automata Theory and Applications, LATA 2009. Proceedings*. LNCS 5457, Springer, 2009, 626–637.

[172] A. Okhotin, On the State Complexity of Scattered Substrings and Superstrings. *Fundam. Inform.* **99** (2010) 3, 325–338.

[173] M. Palmovský, J. Sebej, Star-Complement-Star on Prefix-Free Languages. In: J. Shallit, A. Okhotin (eds.), *17th DCFS 2015*. LNCS 9118, Springer, 2015, 231–242.

[174] A. Paz, B. Peleg, Ultimate-Definite and Symmetric-Definite Events and Automata. *J. ACM* **12** (1965) 3, 399–410.

[175] F. Pereira, M. Riley, Speech recognition by composition of weighted finite automata. In: E. Roche, Y. Schabes (eds.), *Finite-State Language Processing*. The MIT Press, 1996, 431–453.

[176] G. Pighizzini, Unary Language Concatenation and Its State Complexity. In: S. Yu, A. Paun (eds.), *5th CIAA 2000*. LNCS 2088, Springer, 2001, 252–262.

[177] G. Pighizzini, J. Shallit, Unary Language operations, state complexity and Jacobsthal's function. *Int. J. Found. Comput. Sci.* **13** (2002) 1, 145–159.

[178] J.-É. Pin, Syntactic semigroups. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages*. 1, Springer, 1997, 679–746.

[179] E. V. Pribavkina, E. Rodaro, State Complexity of Prefix, Suffix, Bifix and Infix Operators on Regular Languages. In: Y. Gao, H. Lu, S. Seki, S. Yu (eds.), *16th DLT 2012*. LNCS 6224, Springer, 2010, 376–386.

[180] Project DesCo, DesCo: a Web based information system for descriptional complexity. `http://desco.up.pt/`, Access date: 1.10.2013.

[181] M. O. Rabin, D. Scott, Finite automata and their decision problems. *IBM J. Research Development* **3** (1959) 2, 114–129.

[182] N. Rampersad, The state complexity of L^2 and L^k. *Inf. Process. Lett.* **98** (2006), 231–234.

[183] N. Rampersad, J. Shallit, B. Ravikumar, State complexity of unique rational operations. *Theor. Comput. Sci.* **410** (2009), 2431–2441.

[184] B. Ravikumar, Some applications of a technique of Sakoda and Sipser. *SIGACT News* **21** (1990) 4, 73–77.

[185] B. Ravikumar, O. H. Ibarra, Relating the Type of Ambiguity of Finite Automata to the Succinctness of Their Representation. *SIAM J. Comput.* **18** (1989) 6, 1263–1282.

[186] D. Revuz, Minimisation of Acyclic Deterministic Automata in Linear Time. *Theor. Comput. Sci.* **92** (1992) 1, 181–189.

[187] A. Salomaa, K. Salomaa, S. Yu, State complexity of combined operations. *Theor. Comput. Sci.* **383** (2007) 2-3, 140–152.

[188] A. Salomaa, K. Salomaa, S. Yu, Undecidability of the state complexity of composed regular operations. In: A. H. Dediu, S. Inenaga, C. Martín-Vide (eds.), *5th LATA 2011*. LNCS 6638, Springer, 2011, 489–498.

[189] A. Salomaa, D. Wood, S. Yu, On the state complexity of reversals of regular languages. *Theor. Comput. Sci.* **320** (2004) 2-3, 315–329.

[190] K. Salomaa, Descriptional Complexity of Nondeterministic Finite Automata. In: T. Harju, J. Karhumäki, A. Lepistö (eds.), *11th DLT 2007*. LNCS 4588, Springer, 2007, 31–35.

[191] K. Salomaa, State and transition complexity of finite automata. *Invited talk at University of Western Ontario* (2009).

[192] K. Salomaa, S. Yu, NFA to DFA Transformation for Finite Languages over Arbitrary Alphabets. *Journal of Automata, Languages and Combinatorics* **2** (1997) 3, 177–186.

[193] K. Salomaa, S. Yu, On the state complexity of combined operations and their estimation. *Int. J. Found. Comput. Sci.* **18** (2007) 4, 683–698.

[194] M. P. Schützenberger, On finite monoids having only trivial subgroups. *Information and Control* **8** (1965), 190–194.

[195] M. Scott, *Programming Language Pragmatics*. Elsevier, 2006.

[196] J. Sebej, Reversal on Regular Languages and Descriptional Complexity. In: H. Jürgensen, R. Reis (eds.), *15th DCFS 2013*. LNCS 8031, Springer, 2013, 265–276.

[197] J. I. Seiferas, R. McNaughton, Regularity-Preserving Relations. *Theor. Comput. Sci.* **2** (1976) 2, 147–154.

[198] R. E. Stearns, J. Hartmanis, Regularity Preserving Modifications of Regular Expressions. *Information and Control* **6** (1963) 1, 55–69.

[199] J. Sĕbej, *Reversal of Regular Languages and State Complexity*. Master's thesis, P. J. Šafárik, University of Košice, Slovakia, 2010.

[200] G. Thierrin, Convex Languages. In: M. Nivat (ed.), *ICALP 1972*. North-Holland, 1972, 481–492.

[201] J. Wang (ed.), *Handbook of Finite State Based Models and Applications*. CRC Press, 2013.

[202] S. Yu, Regular languages. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages*. 1, Springer, 1997, 41–110.

[203] S. Yu, State Complexity of Regular Languages. *Journal of Automata, Languages and Combinatorics* **6** (2001) 2, 221–234.

[204] S. Yu, State Complexity of Finite and Infinite Regular Languages. *Bulletin of the EATCS* **76** (2002), 142–152.

[205] S. Yu, State Complexity: Recent Results and Open Problems. *Fundam. Inform.* **64** (2005) 1-4, 471–480.

[206] S. Yu, On the State Complexity of Combined Operations. In: O. H. Ibarra, H.-C. Yen (eds.), *11th CIAA 2006*. LNCS 4094, Springer, 2006, 11–22.

[207] S. Yu, Q. Zhuang, On the State Complexity of Intersection of Regular Languages. *ACM SIGACT News* **22** (1991) 7, 52–54.

[208] S. Yu, Q. Zhuang, K. Salomaa, The State Complexities of Some Basic Operations on Regular Languages. *Theor. Comput. Sci.* **125** (1994) 2, 315–328.