Quotient Complexity of Ideal Languages ⋆

Janusz Brzozowski [a,1] Galina Jirásková [b] Baiyu Li [a]

[a] *David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1*

[b] *Mathematical Institute, Slovak Academy of Sciences, Grešákova 6, 040 01 Košice, Slovakia*

Abstract

A language L over an alphabet Σ is a right (left) ideal if it satisfies $L = L\Sigma^*$ ($L = \Sigma^* L$). It is a two-sided ideal if $L = \Sigma^* L \Sigma^*$, and an all-sided ideal if $L = \Sigma^* \sqcup\!\sqcup L$, the shuffle of Σ^* with L. Ideal languages are not only of interest from the theoretical point of view, but also have applications to pattern matching. We study the state complexity of common operations in the class of regular ideal languages, but prefer to use the equivalent term "quotient complexity", which is the number of distinct left quotients of a language. We find tight upper bounds on the complexity of each type of ideal language in terms of the complexity of an arbitrary generator and of the minimal generator, and also on the complexity of the minimal generator in terms of the complexity of the language. Moreover, tight upper bounds on the complexity of union, intersection, set difference, symmetric difference, concatenation, star, and reversal of ideal languages are derived.

Key words: finite automaton, ideal, operation, quotient complexity, regular language, state complexity

⋆ This work was supported by the Natural Sciences and Engineering Research Council of Canada grant OGP0000871 and by VEGA grant 2/0111/09.

Email addresses: brzozo@uwaterloo.ca (Janusz Brzozowski), jiraskov@saske.sk (Galina Jirásková), b5li@student.cs.uwaterloo.ca (Baiyu Li).

URLs: http://maveric.uwaterloo.ca/∼brzozo/ (Janusz Brzozowski), http://im3.saske.sk/∼jiraskov/ (Galina Jirásková), http://www.student.cs.uwaterloo.ca/∼b5li/ (Baiyu Li).

[1] Corresponding author.

1 Introduction

A language is a right ideal if it is closed under concatenation on the right with an arbitrary word. Left ideals and two-sided ideals are defined in a similar way. A language is an all-sided ideal if it is closed under the insertion of an arbitrary word in any position in any word of the language. Ideal languages need not be regular, but our interest is in regular ideals only.

Ideals are studied for several reasons. They are fundamental objects in semigroup theory [26,37]. They appear in the theoretical computer science literature as early as 1965 [32] and continue to be of interest in the present [2,3,12,13]. Ideal languages are complements of prefix-, suffix-, factor-, and subword-closed languages, and closed languages constitute another interesting class [2,10]. Ideal languages are closed with respect to the "has a word as a prefix" (respectively, suffix, factor, subword) relation [2]. They are special cases of convex languages [2,38], which form a much larger class. Finally, besides being of theoretical interest, ideals also play a role in algorithms for pattern matching.

Left and right ideals were studied by Paz and Peleg [32] in 1965 under the names "ultimate definite" and "reverse ultimate definite events"; their results include closure properties, decision procedures, and canonical representations for these languages. All-sided ideals were used by Haines [18] (not under that name) in 1969 in connection with subword-free and subword-closed languages, and by Thierrin [38] in 1973 in connection with subword-convex languages. De Luca and Varricchio [27] showed in 1990 that a language is factor-closed (or "factorial") if and only if it is the complement of a two-sided ideal. The 1994 work of Yu, Zhuang and Salomaa in [42] contains two results about left and right ideals. In 2001 Shyr [37] studied right, left, and two-sided ideals and their generators in connection with codes. In 2007 Okhotin [31] presented a result concerning all-sided ideals. Complexity issues of conversion of nondeterministic finite automata (nfa's) to deterministic finite automata (dfa's), where these automata recognize right, left, and two-sided ideals were studied in 2009 by Bordihn, Holzer, and Kutrib [3], who used the names "ultimate definite", "reverse ultimate definite", and "central definite" languages, respectively. The sizes of all-sided ideals were studied in 2009 by Gruber, Holzer and Kutrib [17]. In 2009 all four types of ideals and their closure properties were considered by Ang and Brzozowski [2] in the framework of languages convex with respect to arbitrary binary relations. Decision problems for various classes of convex languages, including ideals, were addressed in 2011 by Brzozowski, Shallit and Xu [12]. The sizes of the syntactic semigroups of right, left, and two-sided ideals were studied in 2011 by Brzozowski and Ye [13].

As mentioned above, ideals also appear in the important area of pattern matching. For this application, a *text* is represented by a word w over some alphabet Σ. A *pattern* can be as simple as a word or a finite set of words, or it can be an arbitrary language L over Σ described by a regular expression. An occurrence of a pattern represented by L in text w is a triple (u, x, v) such that $w = uxv$ and x is in L. Searching text w for words in L is equivalent to looking for prefixes of w that belong to the language $\Sigma^* L$, which is the left ideal generated by L [15].

Algorithms such as that of Aho and Corasick [1] can be used to determine all possible occurrences of words from a finite set L in a given input w. For example, in a Unix-style editor, such as *sed*, if L is just a single word x, then `/.*x$/`, `/^x.*/`, and `/.*x.*/`, or their simplified versions `/x$/`, `/^x/`, and `/x/`, find all the words ending in x (that is, all the words of the left ideal $\Sigma^* x$) that occur in w; all the words beginning with x (that is, all the words of the right ideal $x\Sigma^*$) that occur in w; and all the words that have x as a factor (that is, all the words of the two-sided ideal $\Sigma^* x \Sigma^*$) that occur in w, respectively. The language $\Sigma^* \sqcup\!\sqcup x$ can be used to find subsequences occurring in the given text, for example, to determine whether a report has all the required sections and that they are in the correct order. For more details we refer the reader to [1,15,16].

For another example of applications of pattern matching with regular languages see the recent work of Yu, Chen, Diao, Lakshman and Katz [39]. They consider the problem of scanning at high speed the content of packets, which are units of binary data routed through a computer communication network; this is crucial for network monitoring and security applications. In such cases nfa's are often used, because the exponential size of the naive dfa's requires excessive memory. Rewriting techniques on regular expressions are used to make fast dfa-based pattern matching feasible. Ideals, though not so named, appear often is this work.

In this paper we study ideal languages from the descriptional complexity point of view. The fact that the four classes of ideals are related to each other permits us to obtain many complexity results using similar methods.

1.1 State Complexity versus Quotient Complexity

The study of state complexity of operations on regular languages is a well-established area of research in theoretical computer science. The *state complexity of a regular language* L is the number of states in the (complete) minimal dfa recognizing L.

For subclasses $\mathbf{C}$ and $\mathbf{C}'$ of regular languages, the *state complexity of an op-*

eration $f : \mathbf{C} \times \mathbf{C} \to \mathbf{C}'$ is a function of the state complexities of languages K and L from $\mathbf{C}$ that returns the maximal state complexity of the language $f(K, L)$. If f is an operation $f : \mathbf{C} \to \mathbf{C}'$, then the state complexity of f returns the maximal state complexity of $f(L)$, for L in $\mathbf{C}$, as a function of the state complexity of L.

A notion equivalent to state complexity is that of the quotient complexity of a language. For a language L over a finite alphabet Σ and a word $w \in \Sigma^*$, the *(left) quotient* of L by w is the language $L_w = \{x \mid wx \in L\}$. The *quotient complexity of a language* L is the number of distinct quotients of L. A language L is regular if and only if it has a finite number of quotients, and this number is precisely the state complexity of L. The *quotient complexity of an operation* is a function, similar to the state complexity function defined above, but here it returns the maximal number of quotients of $f(K, L)$ or of $f(L)$, which, of course, is the same as the state complexity of $f(K, L)$ or of $f(L)$.

Although the two concepts—of state and quotient complexity—are equivalent in the numerical sense, they provide different points of view for the same basic idea. The state complexity approach is automaton-oriented and leads to constructions of automata recognizing the language resulting from an operation. The quotient approach is language-oriented and leads to operations on languages. In particular applications one approach may be more convenient than the other. For example, it is often easy to derive an upper bound on the state/quotient complexity of an operation using quotients. On the other hand, to show that the state/quotient complexity of an operation meets an upper bound it is often more appropriate to use automata. These techniques are illustrated several times in the present paper.

In this paper we use either one approach or the other, as is convenient. The terminology is a question of personal preference, and we prefer to use "quotient complexity", since it is a language property defined in language-theoretic terms. However, since we do not discuss any other type of complexity in this work, we will simply use the term *complexity*.

1.2 Previous Work on Complexity

The bound mn on the complexity of intersection was noted in 1959 by Rabin and Scott [36]. In 1963 Lupanov [28] proved that the bound 2^n for the conversion of nfa's to dfa's is tight. In 1966 Mirkin [30] showed that the 2^n bound for the reversal of a dfa is attainable. The state complexities of union, product, and star were first studied in 1970 by Maslov [29]. He stated upper bounds on the complexity of these operations and gave examples of languages meeting these bounds, but provided no proofs.

In 1994 Yu, Zhuang, and Salomaa [42] examined in detail the complexities of concatenation, star, left and right quotients, reversal, intersection, and union in the class of regular languages. Since then, there have been many papers on this subject; see, for example, the 2001 survey by Yu [41] and the reference lists in that work. Quotient complexity was introduced in 2010 by Brzozowski [6]; that paper also contains a short updated survey.

There has also been a considerable amount of work done on the complexity of operations in *proper subclasses of regular languages*: in unary languages in 1994 by Yu, Zhuang and Salomaa [42] and in 2002 by Pighizzini and Shallit [34]; in finite languages in 2001 by Yu [41] and Câmpeanu, Culik, Salomaa and Yu [14]; in prefix-free languages in 2009 by Han, Salomaa and Wood [20]; in suffix-free languages in 2009 by Han and Salomaa [19]; in closed languages in 2010 by Brzozowski, Jirásková and Zou [10]; in union-free languages in 2010 by Jirásková and Masopust [23]; in bifix-, factor- and subword-free languages in 2011 by Brzozowski, Jirásková, Li and Smith [9]; and in star-free languages in 2011 by Brzozowski and Liu [11]. In general, these studies of subclasses show that the complexity can be significantly lower in a subclass than in the general case. There are, however, some surprises: Brzozowski and Liu [11] showed that all the bounds on operations on regular languages, with some small exceptions, are also met by star-free languages—a very restricted class of regular languages. Analogous results were proved by Holzer, Kutrib and Meckel [21] who showed that, in most cases, exactly the same tight state-complexity bounds are reached by operations on nfa's recognizing star-free languages as on general nfa's. This motivates us to study subclasses of regular languages to determine their complexity characteristics. Here we continue this study in four related classes of regular languages: right, left, two-sided, and all-sided ideals.

1.3 Outline

In Section 2 we define our terminology and notation. The complexities of ideal languages in terms of their generators and minimal generators, and the complexities of generators in terms of ideals are studied in Section 3. The complexities of basic operations on ideals are then examined in Section 4. The special case of unary languages is treated in Section 5, and Section 6 concludes the paper.

An earlier version of this work appeared at arXiv [7], and a much shorter version was published in the LATIN 2010 conference proceedings [8].

2 Preliminaries

We assume that the reader is familiar with basic concepts of regular languages and finite automata, as described in [33,40], for example, or in many textbooks. For general properties of ideal languages we refer the reader to [26,37].

If Σ is a non-empty finite alphabet, then Σ^* is the free monoid generated by Σ. A *word* is any element of Σ^*, and the empty word is ε. The length of a word $w \in \Sigma^*$ is $|w|$. A *language* over Σ is any subset of Σ^*.

The following set operations are defined on languages: *complement* ($\overline{L} = \Sigma^* \setminus L$), *union* ($K \cup L$), *intersection* ($K \cap L$), *difference* ($K \setminus L$), and *symmetric difference* ($K \oplus L$). To indicate any one of these four boolean operations with two arguments we use $K \circ L$. We also define the *product*, usually called *concatenation* or *catenation*, $K \cdot L = \{w \in \Sigma^* \mid w = uv, u \in K, v \in L\}$, *positive closure* $L^+ = \bigcup_{i \geq 1} L^i$, and *star* $L^* = \bigcup_{i \geq 0} L^i$. The reverse w^R of a word w in Σ^* is defined as follows: $\varepsilon^R = \varepsilon$, and $(wa)^R = aw^R$ for a letter a and a word w. The *reverse* of a language L is defined as $L^R = \{w^R \mid w \in L\}$.

Regular languages over an alphabet Σ are languages that can be obtained from the *basic languages* $\emptyset$, $\{\varepsilon\}$, and $\{a\}$, $a \in \Sigma$, using a finite number of operations of union, product, and star. Such languages are usually denoted by regular expressions. For example, $E = (\varepsilon \cup a)^* b$ denotes $L = (\{\varepsilon\} \cup \{a\})^* \{b\}$. We use the symbols $\cup$, $\cdot$ (usually omitted), and * for union, product, and star of both regular expressions and languages, rather than using $+$ in expressions and $\cup$ for languages.

A *deterministic finite automaton (dfa)* is a quintuple $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite, non-empty set of *states*, Σ is a finite, non-empty *alphabet*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial state*, and $F \subseteq Q$ is the set of *final states*. The transition function is naturally extended to the domain $Q \times \Sigma^*$. The *language accepted by* dfa $\mathcal{D}$ is $L(\mathcal{D}) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$. The *language accepted from a state* q of a dfa is the language accepted by the dfa $\mathcal{D}_q = (Q, \Sigma, \delta, q, F)$. Two states of a dfa are *distinguishable* if there exists a word w which is accepted from one of the states and rejected from the other. Otherwise, the two states are *equivalent*. A dfa is *minimal* if all of its states are reachable from the initial state and no two states are equivalent.

A *nondeterministic finite automaton* (nfa) is a quintuple $\mathcal{N} = (Q, \Sigma, \delta, S, F)$, where Q, Σ, and F are defined the same way as in a dfa, S is the set of initial states[2], and δ is the nondeterministic transition function that maps $Q \times \Sigma$ to

[2] In contrast to some authors, we use a *set* of initial states, since we require the reverse of an nfa to be an nfa.

2^Q. The transition function is extended to $2^Q \times \Sigma^*$. The *language accepted by* nfa $\mathcal{N}$ is $L(\mathcal{N}) = \{w \in \Sigma^* \mid \delta(S, w) \cap F \neq \emptyset\}$. The *language accepted from a state* q of an nfa is the language accepted by the nfa $(Q, \Sigma, \delta, \{q\}, F)$.

Every nfa $(Q, \Sigma, \delta, S, F)$ can be converted to an equivalent dfa $(2^Q, \Sigma, \delta', S, F')$ by the well-known subset construction [36]: The transition function δ' is defined by $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$, and a state R in 2^Q is in F' if $R \cap F \neq \emptyset$. We call the resulting dfa the *subset automaton* corresponding to the given nfa. This automaton need not be minimal, since some of its states may be unreachable or equivalent.

The following two lemmata are used often to show reachability and distinguishability of states of a subset automaton.

Lemma 1 (Reachability) *Consider an nfa with initial state q_0, in which there are transitions on inputs a and b in states $q_0, q_1, \ldots, q_{n-2}$ as shown in Fig. 1. Then each subset of $\{q_0, q_1, \ldots, q_{n-1}\}$ containing q_0 is reachable in the corresponding subset automaton.*

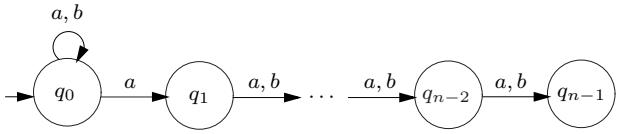

Fig. 1. Reachability of all the subsets of $\{q_0, q_1, \ldots, q_{n-1}\}$ containing state q_0.

PROOF. The proof is by induction on the size of subsets. Subset $\{q_0\}$ is the initial state of the subset automaton. Every subset $\{q_0, q_{i_2}, q_{i_3} \ldots, q_{i_k}\}$ of size k, where $2 \leq k \leq n$ and $1 \leq i_2 < i_3 < \cdots < i_k \leq n-1$, is reached from the subset $\{q_0, q_{i_3 - i_2}, \ldots, q_{i_k - i_2}\}$ of size $k-1$ by $ab^{i_2 - 1}$. □

Lemma 2 (Distinguishability) *If for every state q of an nfa there exists a word w_q that is accepted by the nfa from state q and rejected from any other state, then all the states of the corresponding subset automaton are pairwise distinguishable.*

PROOF. Two distinct subsets of the subset automaton must differ in some state q of the given nfa. These two subsets are distinguished by word w_q, which is accepted by the nfa only from state q. □

Next, we recall some properties of quotients. The *quotient complexity* of L is the number of distinct quotients of L, and is denoted by $\kappa(L)$.

The quotients of a regular language can be computed as follows. First, the *ε-function* L^ε of a regular language L is defined by $L^\varepsilon = \emptyset$ if $\varepsilon \notin L$ and $L^\varepsilon = \varepsilon$ if $\varepsilon \in L$. The quotient by a letter a in Σ is computed by induction:

$$b_a = \begin{cases} \emptyset, \text{ if } b \in \{\emptyset, \varepsilon\}, \text{ or } b \in \Sigma \text{ and } b \neq a, \\ \varepsilon, \text{ if } b = a; \end{cases}$$

$$(\overline{L})_a = \overline{L_a};$$

$$(K \circ L)_a = K_a \circ L_a;$$

$$(KL)_a = K_a L \cup K^\varepsilon L_a;$$

$$(K^*)_a = K_a K^*.$$

The quotient by a word $w \in \Sigma^*$ is computed by induction on the length of w:

$$L_\varepsilon = L; \quad L_{wa} = (L_w)_a.$$

Quotients computed this way are indeed the left quotients of a regular language [5,6]. A quotient L_w is *final* if $\varepsilon \in L_w$; otherwise it is *non-final*.

We use the following formulas [5,6] for quotients of regular languages to establish upper bounds on quotient complexity:

Proposition 3 *If K and L are regular languages, and u and v are in Σ^+, then*

$$(\overline{L})_w = \overline{L_w}; \quad (K \circ L)_w = K_w \circ L_w; \tag{1}$$

$$(KL)_w = K_w L \cup K^\varepsilon L_w \cup \left(\bigcup_{w=uv} K_u^\varepsilon L_v \right). \tag{2}$$

The formulas for boolean operations are obvious. The quotient of a product KL by w consists of the quotient of K by w concatenated with L, of the quotient of L by w if the empty word is in K, and of the quotients of L by non-empty suffixes v of w, where the quotients of K by the corresponding prefixes u of w contain the empty word.

The *quotient dfa* of a regular language L is $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{L_w \mid w \in \Sigma^*\}$, $\delta(L_w, a) = L_{wa}$, $q_0 = L_\varepsilon = L$, and $F = \{L_w \mid \varepsilon \in L_w\}$. So the number of states in the quotient automaton of L is the quotient complexity of L. The quotient dfa of L is isomorphic to the complete minimal dfa of L, and these terms are used here synonymously.

If $u, v, w \in \Sigma^*$ and $w = uxv$, then u is a *prefix* of w, v is a *suffix* of w, and x is a *factor* of w. If $w = u_1v_1u_2v_2 \cdots u_kv_ku_{k+1}$, where the u_i and v_i are in Σ^*, then $v_1v_2 \cdots v_k$ is a *subword* of w. A prefix (suffix, factor, subword) of w is *proper* if it not equal to w.

A language L is *prefix-free* (*prefix-closed*) if $w \in L$ implies that no proper prefix of w is in L (that every prefix of w is in L). In the same way, we define *suffix-free*, *factor-free*, and *subword-free* languages, and the corresponding closed versions.

The *shuffle* $u \sqcup\!\sqcup v$ *of two words* $u, v \in \Sigma^*$ is defined as follows:

$$u \sqcup\!\sqcup v = \{u_1v_1 \cdots u_kv_k \mid u = u_1 \cdots u_k, v = v_1 \cdots v_k, u_1, \ldots, u_k, v_1, \ldots, v_k \in \Sigma^*\}.$$

The *shuffle of two languages* K and L is defined by

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

Note that these operations are commutative.

3 Ideals, Generators, and Minimal Generators

A language $L \subseteq \Sigma^*$ is a *right ideal* (*left ideal*, *two-sided ideal*, *all-sided ideal*) if it is non-empty and satisfies $L = L\Sigma^*$ ($L = \Sigma^*L$, $L = \Sigma^*L\Sigma^*$, $L = \Sigma^* \sqcup\!\sqcup L$, respectively). We refer to all four types as *ideal languages* or simply *ideals*.

If L is a right (respectively, left, two-sided, all-sided) ideal, any language $G \subseteq \Sigma^*$ such that $L = G\Sigma^*$ (respectively, $L = \Sigma^*G$, $L = \Sigma^*G\Sigma^*$, $L = \Sigma^* \sqcup\!\sqcup G$) is a *generator* of L. The quotients of ideals $G\Sigma^*$, Σ^*G, and $\Sigma^*G\Sigma^*$ are derived from Equation (2) and given below, where words u, v, x, and y are in Σ^+:

$$(G\,\Sigma^*)_w = (G_w \cup G^\varepsilon \cup \bigcup_{w=uv} G_u^\varepsilon)\Sigma^*; \tag{3}$$

$$(\Sigma^*G)_w = \Sigma^*G \cup G_w \cup \bigcup_{w=uv} G_v; \tag{4}$$

$$(\Sigma^*G\,\Sigma^*)_w = \Sigma^*(G\,\Sigma^*) \cup (G\,\Sigma^*)_w \cup \bigcup_{w=uv} (G\,\Sigma^*)_v = [\Sigma^*G \cup (G_w \cup \bigcup_{w=uv} G_v) \cup \bigcup_{w=uv} [G_u^\varepsilon \cup (\bigcup_{v=xy} G_x^\varepsilon)]]\Sigma^*. \tag{5}$$

We use these formulas to establish upper bounds on the complexity of the ideals $G\,\Sigma^*$, Σ^*G, and $\Sigma^*G\,\Sigma^*$ generated by G.

Theorem 4 (Complexity of Ideals in Terms of Generators) *Let G be any generator of the right ideal $G\Sigma^*$ (left ideal Σ^*G, two-sided ideal $\Sigma^*G\Sigma^*$, or all-sided ideal $\Sigma^* \sqcup\!\sqcup G$) with $\kappa(G) = n$. Then*

(1) For $n \geq 1$, $\kappa(G\Sigma^*) \leq n$, and the bound is tight if $|\Sigma| \geq 1$;
(2) $\kappa(\Sigma^*G) \leq 2^{n-1}$, and the bound is tight if $|\Sigma| = 1$ for $n = 1$, and $|\Sigma| \geq 2$, otherwise;
(3) For $n = 1$, $\kappa(\Sigma^*G\Sigma^*) = 1$, and for $n \geq 2$, $\kappa(\Sigma^*G\Sigma^*) \leq 2^{n-2} + 1$ and the bound is tight if $|\Sigma| \geq 2$;
(4) For $n = 1$, $\kappa(\Sigma^* \sqcup\!\sqcup G) = 1$, and for $n \geq 2$, $\kappa(\Sigma^* \sqcup\!\sqcup G) \leq 2^{n-2} + 1$ and the bound is tight if $|\Sigma| \geq n - 2$, and cannot be met using any smaller alphabet.

PROOF. The first two items follow from the results in [42]. We give short proofs using quotients.

1. If $n = 1$, then $G = \Sigma^*$ and $\kappa(G\Sigma^*) = 1$. If $n \geq 2$, then G is non-empty. From Equation (3), if w has no prefix in G, then $(G\Sigma^*)_w = G_w\Sigma^*$. As $\kappa(G) = n$, there can be at most $n - 1$ such quotients, for there must be at least one quotient G_w with $w \in G$. However, for every word w with a prefix x in G, we have $(G\Sigma^*)_w = (G\Sigma^*)_x = \Sigma^*$. Hence there are at most n different quotients.

The unary language $G = a^{n-1}a^*$ meets the bound.

2. One of the n quotients of G, namely $G_\varepsilon = G$, always appears on the right-hand side of Equation (4). Thus there are at most 2^{n-1} subsets of quotients of G to be added to Σ^*G, and so Σ^*G has at most 2^{n-1} distinct quotients.

For tightness, if $n = 1$, then $G = \Sigma^*$ meets the bound. For $n \geq 2$, let G be the language accepted by the dfa in Fig. 2. To get an nfa for Σ^*G, add a loop on a in the initial state 0. By Lemma 1, every subset of $\{0, 1, \ldots, n-1\}$ containing state 0 is reachable in the corresponding subset automaton. Since for each state i, the word a^{n-1-i} is accepted by the nfa only from state i, distinguishability follows by Lemma 2.

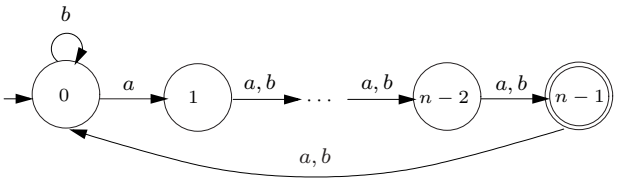

Fig. 2. The dfa of G with $\kappa(G) = n$ and $\kappa(\Sigma^*G) = 2^{n-1}$.

3. Since quotient G is always present in the expression in Equation (5), there are at most 2^{n-1} distinct unions of quotients of language G. Since G is non-empty, it has at least one final quotient. If the final quotient is G, then the

resulting language is Σ^*. Otherwise, at least 2^{n-2} unions contain a final quotient of G, and the corresponding quotients of two-sided ideal $\Sigma^* G \Sigma^*$ are Σ^*. Thus $2^{n-2}+1$ is an upper bound.

To prove the tightness of the bounds, for $n = 1$ use Σ^* and for $n = 2$ use the language $a^*b(a \cup b)^*$. For $n \geq 3$ consider the language G defined by the dfa in Fig. 3. To get an nfa for $\Sigma^* G \Sigma^*$, add a loop on a in the initial state 0. By Lemma 1, every subset of $\{0, 1, \ldots, n-2\}$ containing state 0 is reachable in the corresponding subset automaton. For each state i in $\{0, 1, \ldots, n-2\}$, the word $a^{n-2-i}b$ is accepted by the nfa only from states i and $n-1$. It follows that the subsets of $\{0, 1, \ldots, n-2\}$ are pairwise distinguishable. All of them are non-final states of the subset automaton. Also, the final state $\{0, n-1\}$ is reached from $\{0, n-2\}$ by b. The lower bound $2^{n-2}+1$ follows.

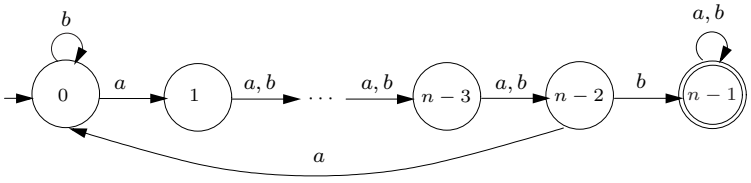

Fig. 3. The dfa of G with $\kappa(G) = n$ and $\kappa(\Sigma^* G \Sigma^*) = 2^{n-2}+1$.

4. If $n = 1$, then $G = \Sigma^*$ and $\kappa(\Sigma^* \sqcup\!\sqcup G) = 1$. For $n \geq 2$, since an all-sided ideal is a two-sided ideal, the upper bound $2^{n-2}+1$ applies. For $n = 2$, the language aa^* meets the bound. For $n \geq 3$, Okhotin [31] used the alphabet $\Sigma = \{a_1, \ldots, a_{n-2}\}$ and the language $G = \bigcup_{i=1}^{n-2} a_i \Sigma^* a_i \Sigma^*$ to prove the tightness of the bound. He also showed that the bound cannot be met if $n-3$ letters are used. □

In Theorem 4, notice the lack of symmetry in the complexities of right and left ideals, and the equality of complexities of two-sides and all-sided ideals. Note also that a provably growing alphabet is required for all-sided ideals [31]. As an anonymous referee correctly points out, in such a situation it is no longer clear whether state complexity is an appropriate measure of complexity. This result should perhaps be restated as follows.

Let G be any generator of the all-sided ideal $\Sigma^* \sqcup\!\sqcup G$ with $\kappa(G) = n \geq 2$, and $|\Sigma| \geq n-2$. Then $\kappa(\Sigma^* \sqcup\!\sqcup G) \leq 2^{n-2}+1$, and this bound is tight.

Here the measure of complexity of G should be some function of both the state complexity and the alphabet size—perhaps their product—because the bound cannot be reached if the alphabet size is bounded by a constant. Conversely, if the size of the alphabet is fixed, then the bound for that alphabet—though it may be hard to find—surely exists, and is guaranteed to be smaller than $2^{n-2}+1$ [31].

Next, we consider the complexities of ideals in terms of their minimal generators. The following are well-known properties of ideals [26].

(1) If L is a right ideal, the *minimal generator* of L is $M = L \setminus (L\Sigma^+)$, and M is prefix-free. If M is prefix-free, then it is the minimal generator of $M\Sigma^*$.
(2) If L is a left ideal, the *minimal generator* of L is $M = L \setminus (\Sigma^+ L)$, and M is suffix-free. If M is suffix-free, then it is the minimal generator of $\Sigma^* M$.
(3) If L is a two-sided ideal, the *minimal generator* of L is $M = L \setminus (\Sigma^+ L\Sigma^* \cup \Sigma^* L\Sigma^+)$, and M is factor-free. If M is factor-free, then it is the minimal generator of $\Sigma^* M\Sigma^*$.
(4) If L is an all-sided ideal, the *minimal generator* of L is the set M of all words of L that have no proper subwords in L, and thus M is subword-free. If M is subword-free, then it is the minimal generator of $\Sigma^* \sqcup\!\sqcup M$.

Theorem 5 (Complexity of Ideals in Terms of Minimal Generators) *Let M be the minimal generator of the right ideal $M\Sigma^*$, (left ideal $\Sigma^* M$, two-sided ideal $\Sigma^* M\Sigma^*$, or all-sided ideal $\Sigma^* \sqcup\!\sqcup M$) with $\kappa(M) = n \geq 3$. Then*

(1) $\kappa(M\Sigma^*) \leq n$ and the bound is tight if $|\Sigma| \geq 2$;
(2) $\kappa(\Sigma^* M) \leq 2^{n-2}$ and the bound is tight if $|\Sigma| \geq 2$;
(3) $\kappa(\Sigma^* M\Sigma^*) \leq 2^{n-3} + 1$ and the bound is tight if $|\Sigma| \geq 2$;
(4) $\kappa(\Sigma^* \sqcup\!\sqcup M) \leq 2^{n-3} + 1$, and the bound is tight if $|\Sigma| \geq n - 3$.

PROOF. 1. The upper bound n follows from Theorem 4. Let $\Sigma = \{a, b\}$, and let $M = a\Sigma^{n-3}$. The dfa for M is shown in Fig. 4. Then M has n quotients and generates the right ideal $L = a\Sigma^{n-3}\Sigma^*$, which also has n quotients. Since M is prefix-free, it is the minimal generator.

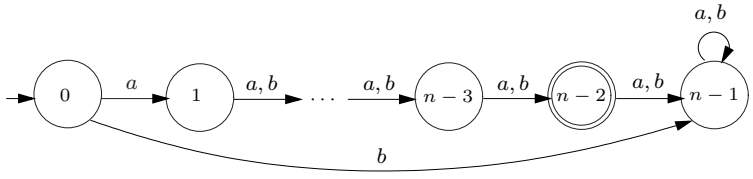

Fig. 4. The dfa of a minimal generator M with $\kappa(M\Sigma^*) = n$ and $\kappa(\Sigma^* M) = 2^{n-2}$.

2. Replace G by M in Equation (4). One of the n quotients of M, namely $M_\varepsilon = M$, always appears in the union. Thus there are at most 2^{n-1} subsets of quotients of M to be added to $\Sigma^* M$. Moreover, since M is suffix-free, M has the empty quotient [19]. Consider the $n - 1$ quotients other than M. Each union of a subset of such quotients that contains the empty quotient is equivalent to a union without the empty quotient; hence there are at most 2^{n-2} quotients of $\Sigma^* M$.

For tightness, let $\Sigma = \{a, b\}$, and consider the suffix-free language $M = a\Sigma^{n-3}$

accepted by the dfa in Fig. 4. To get an nfa for the generated left ideal $\Sigma^* M$, omit the dead state $n-1$ and all transitions incident to it, and add a loop in state 0 on letters a, b. By Lemma 1, all the subsets of $\{0, 1, \ldots, n-2\}$ containing state 0 are reachable in the corresponding subset automaton. These reachable states are pairwise distinguishable since for each state i, the word a^{n-2-i} is accepted by the nfa only from state i. This gives 2^{n-2} reachable and pairwise distinguishable states, and proves the lower bound.

3. Replace G by M in Equation (5). Since $M_\varepsilon = M$ is always present, there are at most 2^{n-1} subsets of quotients of M to add to M_ε. Since M is the minimal generator of L, it is factor-free, and hence prefix-free. Thus it has only one final quotient, ε, and also has the empty quotient, and so we have at most 2^{n-2} subsets. Finally, half of those 2^{n-2} subsets contain Σ^*, and hence are equivalent to Σ^*. This leaves $2^{n-3}+1$ subsets, and so $\kappa(L) \leq 2^{n-3}+1$.

For $n = 3$, let $\Sigma = \{a\}$ and $M = a$; then M is the minimal generator of a^*aa^* and meets the bound. For $n \geq 4$, consider the factor-free language $M = a\Sigma^{n-4}a$ given by the dfa of Fig. 5. To get an nfa for the generated two-sided ideal $\Sigma^* M \Sigma^*$, omit the dead state $n-1$ and add loops on letters a, b in states 0 and $n-2$. By Lemma 1, all the subsets of $\{0, 1, \ldots, n-3\}$ containing state 0 are reachable in the corresponding subset automaton. For states $0, 1, \ldots, n-3$, the word a^{n-2-i} is accepted by the nfa only from state i. Therefore, the non-final subsets of $\{0, 1, \ldots, n-3\}$ are pairwise distinguishable. The final subset $\{0, 1, n-2\}$ is reached from $\{0, n-3\}$ by a, and the lower bound $2^{n-3}+1$ follows.

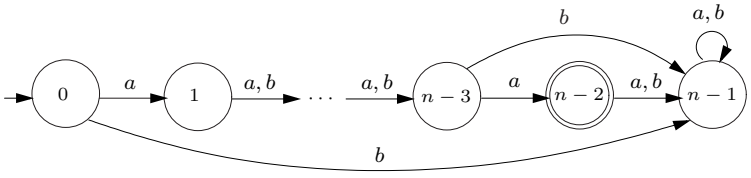

Fig. 5. The dfa of a minimal generator M with $\kappa(\Sigma^* M \Sigma^*) = 2^{n-3}+1$.

4. Since an all-sided ideal is a two-sided ideal, the bound of $2^{n-3}+1$ applies. If $n = 3$, then $M = a$ meets the bound. For $n \geq 4$, consider the subword-free language $M = a_1 a_1 \cup \cdots \cup a_{n-3} a_{n-3}$ over the alphabet $\{a_1, \ldots, a_{n-3}\}$. Figure 6 shows the dfa for M; all the undefined transitions go to the dead state $n-1$ (not shown in the figure).

To get an nfa for $\Sigma^* \sqcup\!\sqcup M$, add loops on every a_i in every state. In the corresponding subset automaton, each subset $\{0, i_1, \ldots, i_k\}$ of $\{0, 1, \ldots, n-3\}$ is reached from the initial state $\{0\}$ by $a_{i_1} \cdots a_{i_k}$. Since a_i with $1 \leq i \leq n-3$ is accepted by the nfa only from states i and $n-2$, the non-final subsets of $\{0, 1, \ldots, n-3\}$ are pairwise distinguishable. One of the final subsets, $\{0, 1, n-2\}$, is reached from $\{0, 1\}$ by $a_1 a_1$. This proves the reachability and

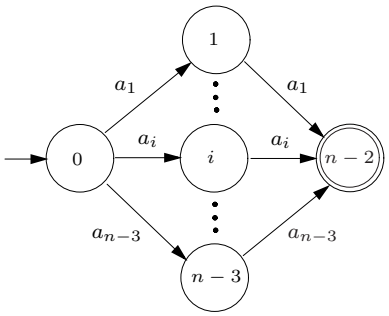

Fig. 6. The dfa of minimal generator M of all-sided ideal N with $\kappa(N) = 2^{n-3}+1$.

distinguishability of $2^{n-3}+1$ subsets, and concludes the proof. □

Theorem 5 shows that using the minimal generator does not affect the complexity of the resulting right ideal, but reduces the complexity of the other ideals roughly by a factor of 2 for large n. There is still lack of left-right symmetry, and the bounds for two-sided and all-sided ideals are again equal.

We now consider the converse problem: Given an ideal L of quotient complexity n, what is the quotient complexity of its minimal generator? We will need the next observation about left ideals which follows from the fact that $vw \in L$ implies $uvw \in L$ if L is a left ideal.

Remark 1 *If L is a left ideal and $u, v \in \Sigma^*$, then $L_v \subseteq L_{uv}$.*

Theorem 6 (Complexity of Minimal Generators) *Let L be an ideal with $\kappa(L) = n$, and let M be its minimal generator.*

(1) If L is a right ideal, then $\kappa(M) \leq n+1$, and the bound is tight if $|\Sigma| \geq 1$.
(2) If L is a left ideal, then $\kappa(M) \leq n(n-1)/2+2$, and the bound is tight if $|\Sigma| \geq 2$.
(3) If L is a two-sided ideal and $n = 1$, then $\kappa(M) = 2$. Otherwise[3] $\kappa(M) \leq 3+(n-1)(n-2)/2$; the bound is tight if $|\Sigma| \geq 1$ when $n \in \{2,3\}$, and if $|\Sigma| \geq 3$ when $n \geq 4$.

PROOF. If $n = 1$, then $L = \Sigma^*$, $M = \varepsilon$, $\kappa(M) = 2$, and the bounds are satisfied in all three cases. Assume from now on that $n > 1$, which implies that $\varepsilon \notin L$.

1. Let L be a right ideal and M its minimal generator. Then M is prefix-free, and therefore the minimal dfa for M has exactly one final state, which goes

[3] We are grateful to Marcus Holzer and Sebastian Jakobi for pointing out two errors in an earlier version of our paper and for providing the witness that satisfies the bound stated here.

to the dead state under each letter. To get a dfa for $L = M\Sigma^*$, we remove all the transitions going from the final state to the dead state, and add a loop on each letter in the final state. In the resulting dfa, the dead state may be unreachable; however, all the remaining states are reachable and pairwise distinguishable. It follows that $\kappa(M) \leq n+1$. The bound is met by the right ideal $L = a^{n-1}a^*$ with $\kappa(L) = n$. The minimal generator is $M = a^{n-1}$ and $\kappa(M) = n+1$.

2. If L is a left ideal and $u, v \in \Sigma^*$, then $L_v \subseteq L_{uv}$ by Remark 1. Since $L = \Sigma^* L$, we have $\Sigma L = \Sigma^+ L$, showing that $M = L \setminus \Sigma L$. Let L have quotients $L_1, L_2, \ldots, L_n$. If $w = av$ is a nonempty word, then $M_w = L_{av} \setminus L_v$, which is a difference of two quotients of L. Next, we have $L_v \subseteq L_{av}$. This means, that if $i \neq j$, then at most one of $L_i \setminus L_j$ and $L_j \setminus L_i$ may be a non-empty quotient of M. Also, $L_i \setminus L_i = \emptyset$ for all i. Hence there are at most $n(n-1)/2+2$ quotients of M: M_ε, at most one quotient for each $i \neq j$, and $\emptyset$.

If $n = 2$, the unary language a^*a meets the bound. For $\Sigma = \{a, b\}$, $n \geq 3$, let $L = (b \cup ab)^* a (ab^*)^{n-3} a \Sigma^*$. The dfa for L is shown in Fig. 7(a). Note that $w \in L$ if and only if $w = xa(ab^*)^{n-3}ay$ for some words x and y, because every quotient of L contains $a(ab^*)^{n-3}a$, or, equivalently, the language $a(ab^*)^{n-3}a$ is accepted from every state of the dfa. Thus L is a left ideal with $\kappa(L) = n$.

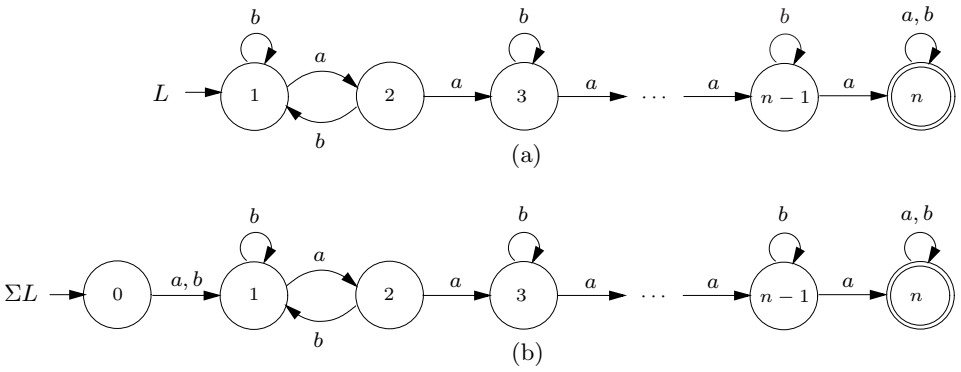

Fig. 7. The dfa's of a left ideal L and of the language ΣL.

The dfa of ΣL is shown in Fig. 7(b). Let $M = L \setminus \Sigma L$, and construct the cross-product automaton for M; see Fig. 8 for $n = 5$. The initial state $(1, 0)$ goes by b to state $(1, 1)$, which turns out to be a dead state, and by a to state $(2, 1)$. Every state $(i, 1)$ with $i \geq 3$ in column 1 is reached from state $(2, 1)$ by $(ab)^{i-2}$. Then every state (i, j) with $i > j \geq 2$ is reached from a state in column 1 by a word in a^*, and there are $n(n-1)/2$ such states. Adding the initial state and the dead state we get $n(n-1)/2 + 2$ reachable states.

Now consider only the above mentioned reachable states. Two states (i, j) and (k, ℓ) with $i < k$, that is, states in different rows, are distinguished by a^{n-k},

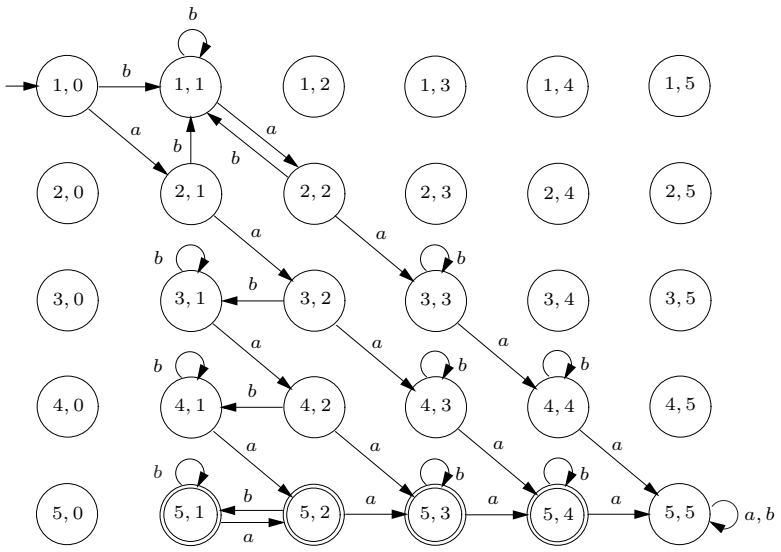

Fig. 8. The dfa's of a left ideal L and of the language ΣL.

which is accepted from (k, ℓ) and rejected from (i, j). Two states (i, j) and (i, ℓ) with $j < \ell$, that is, two distinct states in the same row, go to two states (n, j') and (n, ℓ') with $j' < \ell'$ in row n by a^{n-i}. The latter states are distinguished by $a^{n-\ell'}$, which is rejected from (n, ℓ') and accepted from (n, j'). Thus the reachable states are pairwise distinguishable, and our proof is complete.

3. Since $M = L \setminus (\Sigma^+ L\Sigma^* \cup \Sigma^* L\Sigma^+)$ and $L = \Sigma^* L\Sigma^*$, the minimal generator is $M = L \setminus (\Sigma L \cup L\Sigma)$, and $M_w = L_w \setminus \Big((\Sigma L)_w \cup (L\Sigma)_w\Big)$ for every w in Σ^*. If $w = \varepsilon$, then $M_w = M$; otherwise, $w = av = ub$ for some words u, v in Σ^* and letters a, b in Σ. We have

$$(\Sigma L)_w = (\Sigma L)_{av} = \{x \mid avx \in \Sigma L\} = \{x \mid vx \in L\} = L_v. \tag{6}$$

Next,

$$(L\Sigma)_w = (L\Sigma)_{ub} = \{x \mid ubx \in L\Sigma\}. \tag{7}$$

There are now two cases:

(1) If $u \in L$, then $(L\Sigma)_{ub} = \Sigma^*$ since L is a two-sided ideal and therefore $u \in L$ implies $uz \in L$ for every word z.
(2) If $u \notin L$, then $(L\Sigma)_{ub} = \{x = x'c \mid ubx' \in L \text{ and } c \in \Sigma\} = L_w\Sigma$.

Let us now return to M_w. If $u \in L$, then $(L\Sigma)_w = \Sigma^*$ and $M_w = L_w \setminus \Sigma^* = \emptyset$. If $u \notin L$, then $M_w = L_w \setminus (L_v \cup L_w\Sigma)$. Since L is also a left-ideal, we have $L_v \subseteq L_{av} = L_w$. Suppose the quotients of L are $L_1, \ldots, L_n$, where $L_n = \Sigma^*$. Then for any pair (i, j), $i \neq j$, at most one of $L_i \setminus (L_j \cup L_i\Sigma)$ and $L_j \setminus (L_i \cup L_j\Sigma)$ may be nonempty. In particular, for any $j \neq n$, since $L_j \neq \Sigma^*$, we must have $\varepsilon \notin L_j$; so $L_n \setminus (L_j \cup L_n\Sigma) = \Sigma^* \setminus (L_j \cup \Sigma^+) = \varepsilon$. We also have $L_j \setminus (L_n \cup L_j\Sigma) = \emptyset$. Therefore, there are at most $3 + (n-1)(n-2)/2$ distinct quotients of M: M_ε, ε, $\emptyset$, and at most one quotient for each pair (i, j), with $i \neq j$ and $i, j \neq n$.

If $n = 2$ ($n = 3$), the unary ideal $L = a^*aa^*$ ($L = a^*aaa^*$) has minimal generator $M = a$ ($M = aa$), which meets the bound 3 (4). It was conjectured by Marcus Holzer and Sebastian Jakobi [4] that the language L accepted by the dfa in Fig. 9 might have the highest complexity; we now prove this conjecture.

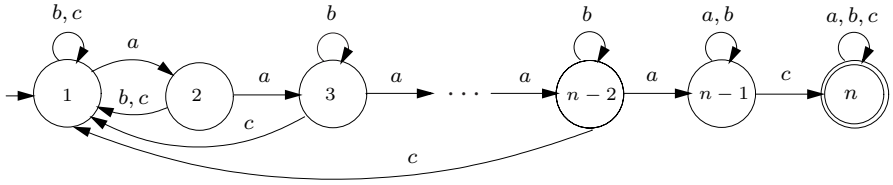

Fig. 9. The dfa of two-sided ideal L meeting the bound for minimal generator.

Construct the dfas for the languages ΣL and $L\Sigma$ as shown in Fig. 10. The minimal generator of L is $M = L \setminus (\Sigma L \cup L\Sigma) = \overline{\Sigma L} \cap L \cap \overline{L\Sigma}$.

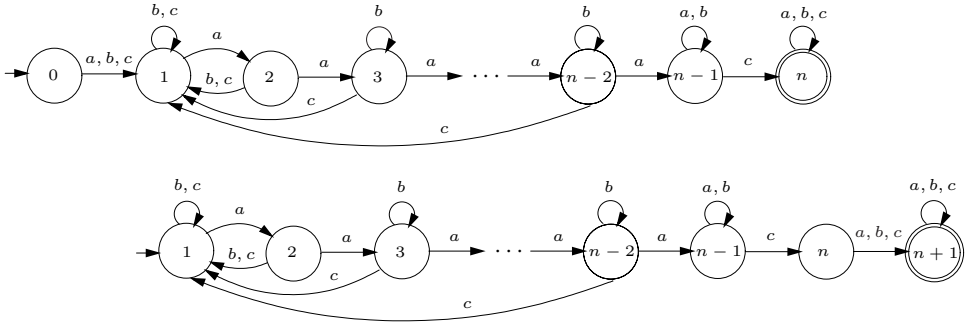

Fig. 10. The dfa's for ΣL and $L\Sigma$.

Construct the direct product of $\overline{\Sigma L}$, L, and $\overline{L\Sigma}$ with initial state $(0, 1, 1)$; this is the only state with first component 0. From state $(0, 1, 1)$ we reach state $(1, 2, 2)$ by a. By applying $(ab)^{j-2}$, we reach $(1, j, j)$ for $j = 3, \ldots, n-1$. Thus we can reach $n-2$ states of the form $(1, j, j)$. From $(1, 2, 2)$ we reach $(2, 3, 3)$ by a, and then $(2, j, j)$ by $(ba)^{j-3}$, for $j = 4, \ldots, n-1$. Thus we can reach $n-3$ states of the form $(2, j, j)$. Having reached (i, j, j), we reach $(i+1, j+1, j+1)$ by a for $i = 2, \ldots, n-3$ and $j = 3, \ldots, n-2$.

So far, we have reached state $(0, 1, 1)$ and $(n-1)(n-2)/2$ states of the form (i, j, j) with $i = 1, \ldots, n-2$ and $j = i + \ 1, \ldots, n-1$. All these states are non-final, because the second component is less than n, which is the only final state of the dfa for L. From $(n-2, n-1, n-1)$, we reach $(1, n, n)$, which is final, because state 1 is non-final in the dfa for L, state n is final in in the dfa for ΣL, and non-final in the dfa for $L\Sigma$.

From the initial state, we reach $(1, 1, 1)$ by b. From any state of the form (i, i, i), we can only reach another state with all three components equal or the state $(n, n, n+1)$. All these states are non-final, and hence empty.

[4] personal communication

In summary, we have shown that the following $3 + (n-1)(n-2)$ states are reachable: the initial state $(0,1,1)$, the final state $(1,n,n)$, the $(n-1)(n-2)/2$ states given above, and an empty state.

We now prove that all these states are distinguishable. The initial state $(0,1,1)$ is the only state accepting $a(ab)^{n-3}c$, $(1,1,1)$ is empty, and $(1,n,n)$ is the only final state. Thus we are left with the remaining $(n-1)(n-2)/2$ states.

For any two different states (i,j,j) and (i',j',j'), where $1 \leq i < j \leq n-1$ and $1 \leq i' < j' \leq n-1$, we have two cases:

(1) $j \neq j'$: Assume that $j < j'$ without loss of generality. Then state (i',j',j') accepts $w = a^{n-1-j'}c$, but state (i,j,j) rejects w.
(2) $j = j'$ and $i \neq i'$: Assume that $i < i'$. Then state (i',j',j') accepts $w = a^{n-2-i'}c$, but state (i,j,j) rejects w.

Thus (i,j,j) and (i',j',j') are distinguishable in both cases.

Therefore the quotient complexity of M_n is $(n-1)(n-2)/2+3$. □

The construction of the minimal generator of L can be viewed as an operation on L, as has been done by Pribavkina and Rodaro [35]. They define the following operators on an arbitrary regular language $L \subseteq \Sigma^+$ and derive their complexities:

(1) The *prefix operator* $L^p = L \setminus L\Sigma^+$; complexity $n+1$.
(2) The *suffix operator* $L^s = L \setminus \Sigma^+ L$; complexity $(n-1)2^{n-2}+2$.
(3) The *infix operator* $L^i = L \setminus (\Sigma^+ L \Sigma^* \cup \Sigma^* L \Sigma^+)$; complexity $(n-2)2^{n-2}+3$.
(4) The *hypercode operator* $L^h = L \setminus \bigcup_{a_1a_2\cdots a_n \in L} \Sigma^* a_1 \Sigma^* a_2 \cdots \Sigma^* a_n \Sigma^*$; complexity $(n-2)2^{n-2}+3$.

Our results show that the complexity is also $n+1$ if L is a right ideal. However, the results differ considerably for left and two-sided ideals, since the complexity of the suffix operation is only $n(n-1)/2+2$ for left ideals, and that of the infix is only $3+(n-1)(n-2)/2$ for two-sided ideals. We do not know the complexity of the hypercode operator for all-sided ideals.

4 Basic Operations on Ideals

We now examine the complexity of common operations on ideal languages. For regular languages, the bounds are known, and they are tight in the binary case; references will be given for each operation later. In this section, we show that the bounds for ideals are generally lower, and tight for languages over

small fixed alphabets, except for reversal of all-sided ideals, which requires a growing alphabet.

4.1 Boolean Operations

For the boolean operations of union [29,42], intersection [36,42], difference [6] and symmetric difference [6], the bound for regular languages is mn, and it is tight for all four operations for binary alphabets.

We show first that the bounds for right, two-sided, and all-sided ideals are still mn for intersection and symmetric difference. However, the bound for union is decreased by $(m+n-2)$, and that for difference, by $m-1$. To prove the tightness of these bounds, the same two languages can be used for all four operations, as is shown in the next theorem.

Theorem 7 (Boolean Operations: Right, 2-Sided, All-Sided Ideals) *Let K and L be right ideals (respectively, two-sided ideals, or all-sided ideals) over an alphabet Σ with $\kappa(K) = m \geq 1$ and $\kappa(L) = n \geq 1$. Then*

(1) $\kappa(K \cap L), \kappa(K \oplus L) \leq mn$,
(2) $\kappa(K \cup L) \leq mn - (m+n-2)$,
(3) $\kappa(K \setminus L) \leq mn - (m-1)$,

and all the bounds are tight if $|\Sigma| \geq 2$.

PROOF. The upper bound mn for intersection and symmetric difference holds since it holds for regular languages. Since K and L both have Σ^* as a quotient, $\kappa(K \cup L) \leq mn - (m+n-2)$ and $\kappa(K \setminus L) \leq mn - (m-1)$ by Theorem 6 (iv) of [6].

For tightness of all four bounds, consider the all-sided ideals K and L accepted by dfa's in Fig. 11. Construct the corresponding cross-product automaton with state set $\{0, \ldots, m-1\} \times \{0, \ldots, n-1\}$, with $(0,0)$ as the initial state. By a, each state (i,j) goes to $(i+1, j)$, except for states $(m-1, j)$ that go to themselves. By b, each state (i,j) goes to $(i, j+1)$, except for states $(i, n-1)$ that go to themselves. In this cross-product automaton, each state (i,j) is reached from the initial state $(0,0)$ by $a^i b^j$. The cross-product automaton for the symmetric difference of K and L is shown in Fig. 12 for $m = 4$ and $n = 5$. For the other operations only the final states change.

In the case of intersection, the sole final state is $(m-1, n-1)$. Consider two states in different rows, that is states (i,j) and (k,l) with $i < k$. By word b^n,

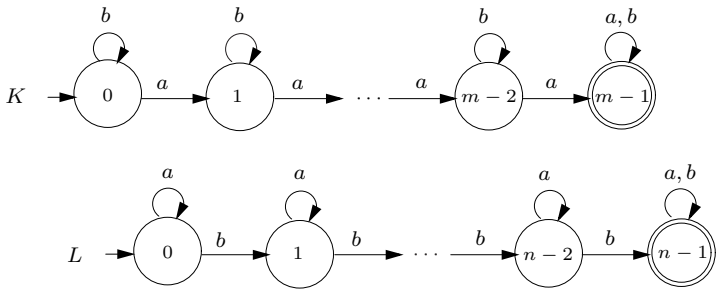

Fig. 11. The all-sided ideals meeting the upper bounds for boolean operations.

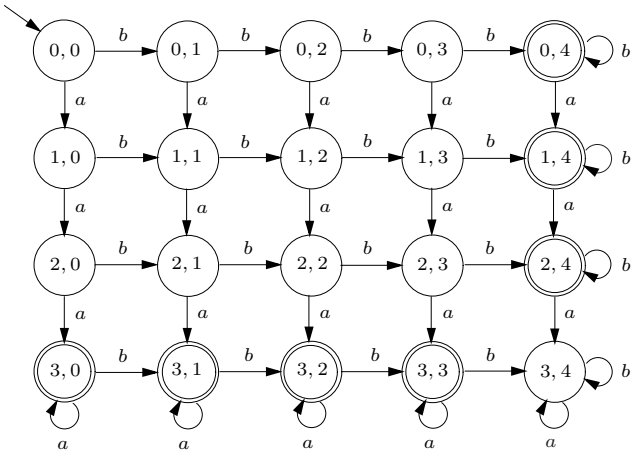

Fig. 12. Cross-product automaton for symmetric difference; $m = 4, n = 5$.

they go to distinct states $(i, n-1)$ and $(k, n-1)$ in the column $n-1$. The latter states are distinguished by word a^{m-1-k}, since it is accepted from $(k, n-1)$ but rejected from $(i, n-1)$. Symmetrically, two states in different columns are distinguished by a word in $a^m b^*$.

In the case of symmetric difference, all the states in row $m-1$ and column $n-1$, except for state $(m-1, n-1)$ are final. Consider two states in different rows, that is states (i, j) and (k, l) with $i < k$. Then the word $b^n a^{m-1-k}$ is rejected from (k, ℓ) and accepted from (i, j). Symmetrically, two states in different columns are distinguished by a word in $a^m b^*$.

In the case of union, all the states in row $m-1$ and in column $n-1$ are final. All of them accept Σ^*. Therefore, these final states are equivalent. The non-final states are distinguished by a word in $a^* \cup b^*$.

In the case of difference, all the states in row $m-1$, except for state $(m-1, n-1)$ are final. All the states in column $n-1$ are equivalent to the dead state $(m-1, n-1)$. Consider the remaining states. The states in different rows are distinguished by a word in a^*. States (i, j) and (i, ℓ) with $j < \ell$ are distinguished by $a^m b^{n-1-\ell}$ since it is rejected from (i, ℓ) but accepted from (i, j). $\square$

Now we turn to left ideals. The next theorem shows that the complexity of all four operations is the same as for regular languages, and binary alphabets suffice for tightness for intersection and symmetric difference. However, an alphabet of four letters is needed for union and three, for difference.

Theorem 8 (Boolean Operations: Left Ideals) *Let K and L be left ideals over an alphabet Σ with $\kappa(K) = m \geq 1$, $\kappa(L) = n \geq 1$, Then*

(1) $\kappa(K \cap L), \kappa(K \oplus L) \leq mn$, and the bound is tight if $|\Sigma| \geq 2$;
(2) $\kappa(K \cup L) \leq mn$, and the bound is tight if $|\Sigma| \geq 4$;
(3) $\kappa(K \setminus L) \leq mn$, and the bound is tight if $|\Sigma| \geq 3$.

PROOF. All the upper bounds hold since they hold for regular languages. Let us prove the lower bounds.

1. Since languages K and L accepted by dfa's in Fig. 11 are all-sided ideals, the lower bounds for intersection and symmetric difference follow by Theorem 7.

2. Consider left ideals K and L accepted by the dfa's in Fig. 13. In the corresponding cross-product automaton for union, each state (i, j) is reached from the initial state $(0, 0)$ by $a^i b^j$. Notice that each state (i, j) goes to state $(i, 0)$ by c, and to state $(0, j)$ by d. All the states in row $m-1$ and in column $n-1$ are final. Two distinct states in different rows are distinguished by a word in ca^*, and two distinct states in different columns are distinguished by a word in db^*.

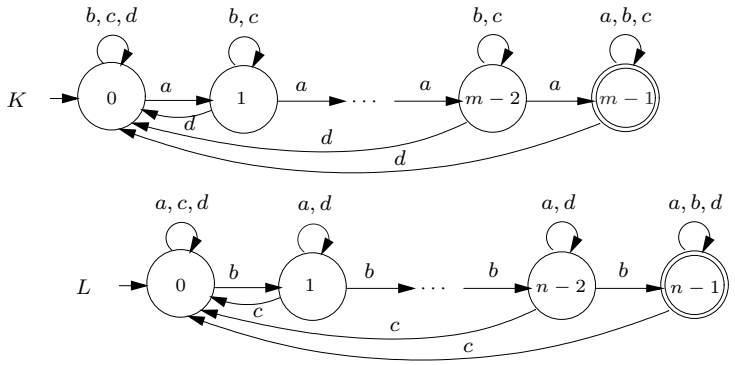

Fig. 13. The left ideals meeting the bound mn for union.

3. Consider left ideals K and L accepted by dfa's in Fig. 13, but restricted to letters a, b, c. In the corresponding cross-product automaton for difference, each state (i, j) is reached from the initial state $(0, 0)$ by $a^i b^j$. All the states in row $m-1$, except for state $(m-1, n-1)$, are final. Two distinct states in different rows are distinguished by a word in ca^*. Two states (i, j) and (i, ℓ) with $j < \ell$ are distinguished by $b^{n-1-\ell} a^{m-1-i}$ since state (i, ℓ) goes to non-final

state $(m-1, n-1)$ by this word, while state (i, j) goes to a final state in row $m-1$. $\square$

4.2 Product

The bound for product (catenation, concatenation) of regular languages is $(m-1)2^n + 2^{n-1}$, and it is tight in the binary case [22,29,42]. We show that the bound for right ideals is still exponential in n, and can be met by binary languages. In contrast to this, the bound for the other three ideals is only $m+n-1$ and it is met by unary languages.

Theorem 9 (Product) *Let K and L be ideals of the same type with $\kappa(K) = m \geq 1$ and $\kappa(L) = n \geq 1$. Then*

(1) If K and L are left, two-sided, or all-sided ideals, then $\kappa(KL) \leq m+n-1$;
(2) If K and L are right ideals and $n \geq 2$, then $\kappa(KL) \leq m + 2^{n-2}$.

The first bound is tight if $|\Sigma| \geq 1$, and the second, if $|\Sigma| \geq 2$.

PROOF. 1. If $m = 1$, then $K = \Sigma^*$, and $\kappa(KL) = \kappa(\Sigma^* L) = n = m+n-1$. Hence suppose that $m \geq 2$, K and L are left ideals, and $\mathcal{A}$ and $\mathcal{B}$ are the dfa's for K and L, respectively. Construct a dfa $\mathcal{C}$ from dfa's $\mathcal{A}$ and $\mathcal{B}$ by omitting all the final states of $\mathcal{A}$ and all the transitions going from the final states, and by redirecting all the transitions that go from a non-final state to a final state of $\mathcal{A}$ to the initial state of $\mathcal{B}$.

Let us show that dfa $\mathcal{C}$ accepts KL. If a word w is accepted by $\mathcal{C}$, then it is in KL. Now let w be a word in KL. Then $w = uv$ for some words u and v such that dfa $\mathcal{A}$ accepts u and dfa $\mathcal{B}$ accepts v. Let u' be the shortest prefix of u such that dfa $\mathcal{A}$ is in a final state after reading u'. Then $u = u'u''$ for some word u''. Since L is a left ideal and v is in L, the word $u''v$ is in L as well, and therefore $\mathcal{B}$ accepts $u''v$. It follows that dfa $\mathcal{C}$ accepts $u'u''v$ since the accepting computation of $\mathcal{C}$ on $u'u''v$ consists of the computation of $\mathcal{A}$ on u', in which the last transitions is redirected to the initial state of $\mathcal{B}$, and of the accepting computation of $\mathcal{B}$ on $u''v$. Hence $\mathcal{C}$ accepts KL and has at most $m+n-1$ states.

Since every all-sided or two-sided ideal is also a left ideal, the upper bound applies in these cases as well. The bound is met by unary all-sided ideals $a^{m-1}a^*$ and $a^{n-1}a^*$.

2. If K and L are right ideals, then $KL = K\Sigma^* L\Sigma^* = K \cdot \Sigma^* L\Sigma^*$, where $\Sigma^* L\Sigma^*$ is a left ideal. The quotient complexity of this left ideal is at most $2^{n-2}+1$ by

Theorem 4. In the same way as above, we can construct a dfa for $K \cdot \Sigma^* L \Sigma^*$ of at most $m + 2^{n-2}$ states.

For tightness, consider the right ideals K and L given by dfa's in Fig. 14; if $m = 1$, then $K = (a \cup b)^*$ and if $n = 2$, then $L = b^* a (a \cup b)^*$. To get an nfa for KL, add a loop on a in state p_0, and redirect transitions on a, b from state $m - 2$ to state p_0.

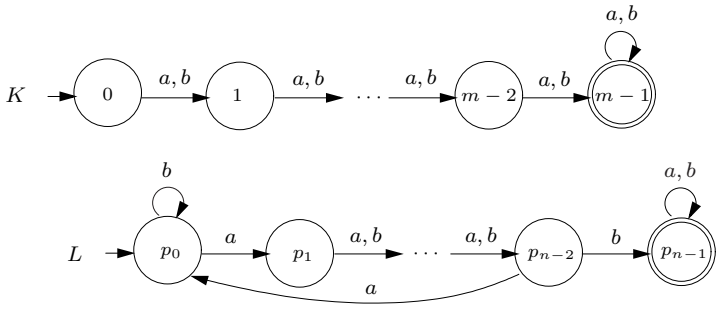

Fig. 14. The right ideals meeting the bound $m + 2^{n-2}$ for product.

In the corresponding subset automaton, all the subsets of $\{p_0, p_1, \ldots, p_{n-2}\}$ containing state p_0 are reachable by Lemma 1, and two such distinct subsets are distinguished by a word in a^*b. The singleton sets $\{0\}, \{1\}, \ldots, \{m-2\}$ are reachable as well, and are distinguished by a word in a^*b. The singleton set $\{i\}$ with $0 \leq i \leq n-2$ and a subset of $\{p_0, p_1, ..., p_{n-2}\}$ containing state p_0 are distinguished by the word $a^{n-2}b$ that is rejected from $\{i\}$ but accepted from any such subset, since state p_0 goes to the accepting state p_{n-1} by $a^{n-2}b$. All these subsets are non-final states of the subset automaton. The final state $\{p_0, p_{n-1}\}$ is reached from $\{p_0, p_{n-2}\}$ by b. This gives $m + 2^{n-2}$ reachable and pairwise distinguishable states. □

4.3 Star

For the star operation, the bound for regular languages [29,42] is $2^{n-1} + 2^{n-2}$, and it is met by a binary language. In sharp contrast to this, the corresponding bound for ideals is only $n + 1$, and it is also met by a binary language.

Theorem 10 (Star) *Let L be an ideal language with $\kappa(L) = n \geq 2$. Then $\kappa(L^*) \leq n + 1$, and the bound is tight if $|\Sigma| \geq 2$.*

PROOF. If L is an ideal and $i \geq 1$, then $L^i \subseteq L$. It follows that $L^* = \{\varepsilon\} \cup L$. To get a dfa for L^* from the quotient automaton for L, we only need to add a new initial and final state going by every letter a to state L_a corresponding to the quotient of L by a. Therefore, $\kappa(L^*) \leq n + 1$.

For tightness, consider the binary all-sided ideal accepted by the dfa of Fig. 15. Construct a dfa for L^* by adding a new initial and final state going to state 1 by a and to state 0 by b. The resulting $(n+1)$-state dfa is minimal since the new initial and final state is distinguished from final state $n-1$ by b, while two distinct non-final states are distinguished by a word in a^*. □

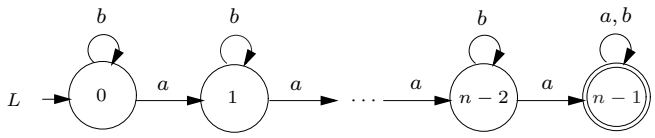

Fig. 15. The all-sided ideal meeting the bound $n+1$ for star.

4.4 Reversal

To deal with reversal, we start with the quotient dfa of L and reverse it by making all the final states initial, making the initial state into a final state, and reversing all the transitions. We then use the subset construction to obtain a dfa for L^R with at most 2^n states.

By a theorem of Brzozowski [4], if a dfa has no unreachable states then the subset construction applied to its reverse and restricted to the reachable states always yields a minimal dfa. Therefore, the complexity of the reverse of a regular language is the same as the number of reachable states in the subset construction for its reverse, and distinguishability need not be verified.

In the case of regular languages, the bound for reversal [25,30] is 2^n, and it is met by a binary language. The bounds for ideals are still exponential, but with somewhat reduced exponents. The witness is binary for right ideals, ternary for left and two-sided ideals, and requires a growing alphabet of $2n-4$ letters for all-sided ideals. Since the reverse of any language recognized by a 1-state dfa is the same language, we assume that $n \geq 2$ in the next theorem.

Theorem 11 (Reversal) *Let L be a language with $\kappa(L) = n \geq 2$.*

(1) If L is a right ideal, then $\kappa(L^R) \leq 2^{n-1}$.
(2) If L is a left ideal, then $\kappa(L^R) \leq 2^{n-1}+1$.
(3) If L is a two-sided ideal, then $\kappa(L^R) \leq 2^{n-2}+1$.
(4) If L is an all-sided ideal, then $\kappa(L^R) \leq 2^{n-2}+1$.

The bound is tight for right ideals if $|\Sigma| = 1$ for $n = 2$ and if $|\Sigma| \geq 2$ otherwise, for left and two-sided ideals if $|\Sigma| \geq 3$, and for all-sided ideals if $|\Sigma| \geq 2n-4$.

PROOF. 1. Since L is a right ideal, it has only one final quotient Σ^*. This quotient becomes the initial state of the nfa for L^R. Since this initial state goes to itself by every letter, it appears in every reachable subset of the corresponding subset automaton. Hence there are at most 2^{n-1} reachable states in the corresponding subset automaton.

For tightness, if $n = 2$, then $L = aa^*$ meets the bound. For $n \geq 3$, consider the binary right ideal accepted by the dfa in Fig. 16. In the subset automaton corresponding to the reverse of this dfa, every subset of $\{0, 1, \ldots, n-1\}$ containing state $n-1$ is reachable by Lemma 1. This gives 2^{n-1} reachable subsets and proves the lower bound.

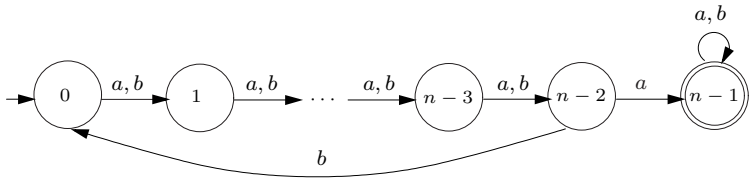

Fig. 16. The right ideal meeting the bound 2^{n-1} for reversal.

2. The initial state of the quotient automaton of a left ideal L is the only final state in the nfa for L^R. In the corresponding subset automaton, this state appears in 2^{n-1} subsets. All these subsets are final states of the subset automaton and all accept Σ^*, since L^R is a right ideal. Hence $\kappa(L^R) \leq 2^{n-1}+1$.

Let us prove the tightness of the bound. If $n = 2$, then the bound is met by the language $(a \cup b)^* a$. If $n \geq 5$, consider the ternary left ideal accepted by the dfa shown in Fig. 17, where all the transitions under c from states $1, 2, \ldots, n-1$ go to state 1. Notice that the automaton restricted to states $1, 2, \ldots, n-1$ and inputs a and b is Šebej's $(n-1)$-state automaton [24] meeting the upper bound for reversal. Therefore, every subset of $\{1, 2, \ldots, n-1\}$ is reachable in the subset automaton corresponding to the reverse of the dfa in Fig 17. Next, state $\{0, 1, \ldots, n-1\}$ is reached from state $\{1\}$ by c.

For $n = 4$, input b maps state 3 to itself in the dfa of Fig. 17 and the remaining transitions are not changed. For $n = 3$, input a maps 0 to itself and transposes 1 and 2, and input b is unchanged.

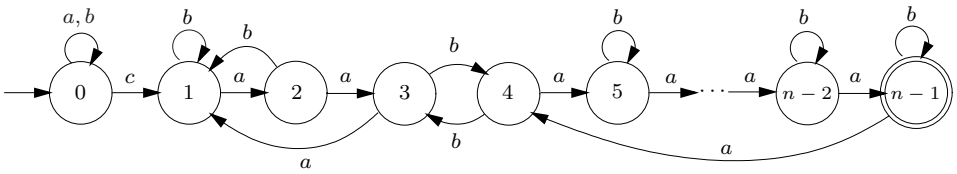

Fig. 17. The left ideal meeting the bound $2^{n-1}+1$ for reversal. States $1, 2, \ldots, n-1$ go to state 1 under c.

3. Since L is a right ideal, its quotient automaton has exactly one final state, which accepts Σ^*. Therefore the subset automaton for L^R has at most 2^{n-1} reachable states. Since L is also a left ideal, all final states of the subset automaton for L^R accept Σ^*. Hence $\kappa(L^R) \leq 2^{n-2}+1$.

If $n = 2$, the bound is met by unary two-sided ideal a^*aa^*. For $n \geq 3$, consider the ternary two-sided ideal accepted by the dfa in Fig. 18. By Lemma 1, every subset of $\{1, 2, \ldots, n-1\}$ containing state $n-1$ is reachable in the subset automaton corresponding to the reverse of the dfa. Moreover, state $\{0, 1, \ldots, n-1\}$ is reached from state $\{1, n-1\}$ by c. This gives $2^{n-2}+1$ reachable subsets and proves the lower bound.

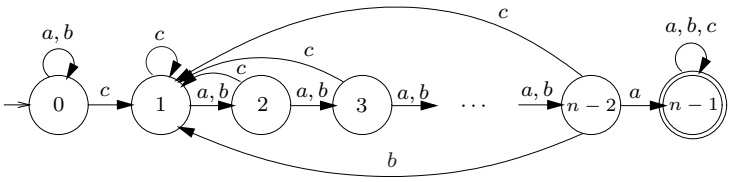

Fig. 18. The two-sided ideal meeting the bound $2^{n-2}+1$ for reversal.

4. Since an all-sided ideal is a two-sided ideal, the bound $2^{n-2}+1$ applies.

If $n = 2$, then the bound is 2, and it is met by the unary all-sided ideal a^*aa^*. If $n \geq 3$, consider the language L over the alphabet $\{a_1, \ldots, a_{n-2}, b_1, \ldots, b_{n-2}\}$ accepted by the dfa in Fig. 19. Here the initial state 0 goes to state i by a_i and to itself by all b_j's. Every state i goes to state $n-1$ by b_i and by all a_j's, and to itself by every other letter. The sole final state $n-1$ goes to itself by every letter. After adding loops on every letter in every state of the dfa, and applying the subset construction and minimization to the resulting nfa, we get a dfa isomorphic to the original one. It follows that L is an all-sided ideal.

In the subset automaton corresponding to the reverse of the dfa for L, $\{n-1\}$ is the initial state, every one of the $2^{n-2}-1$ subsets $\{n-1, i_1, i_2, \ldots, i_k\}$, where $1 \leq k \leq n-2$ and $1 \leq i_1 < i_2 < \cdots < i_k \leq n-2$, is reached from the initial state $\{n-1\}$ by word $b_{i_1}b_{i_2}\cdots b_{i_k}$. The set $\{0, 1, \ldots, n-1\}$ is reached from $\{n-1\}$ by a_1a_1. This completes the proof. □

5 Unary Languages

Unary languages have special properties because the product of unary languages is commutative. Let $\Sigma = \{a\}$. If L is a unary right ideal, let a^i be its shortest word. Then $L \supseteq a^ia^*$, and so $L = a^ia^*$, and every unary right ideal is principal (generated by a single element). In fact, $L = a^ia^* = a^*a^i = a^*a^ia^* = a^* \sqcup\!\sqcup a^i$; hence left, right, two-sided and all-sided ideals coincide.

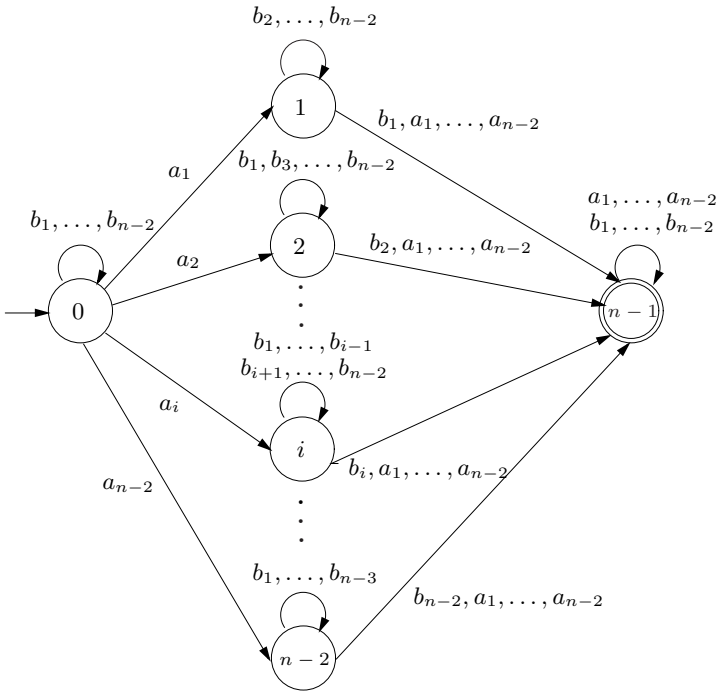

Fig. 19. The dfa of the all-sided ideal meeting the bound $2^{n-2}+1$ for reversal.

Proposition 12 *Let K and L be unary ideals of any type, with $\kappa(K) = m \geq 1$, $\kappa(L) = n \geq 1$. Let M be the minimal generator of L. Then*

$$
\begin{aligned}
\kappa(M) &= n+1;\\
\kappa(L) &= \kappa(M)-1;\\
\kappa(K \cup L) &= min(m,n);\\
\kappa(K \cap L) &= max(m,n);\\
\kappa(K \setminus L) &= \begin{cases} n, \textit{ if } m < n,\\ 1, \textit{ otherwise}; \end{cases}\\
\kappa(K \oplus L) &= \begin{cases} max(m,n), \textit{ if } m \neq n,\\ 1, \qquad\qquad \textit{otherwise}; \end{cases}\\
\kappa(KL) &= m+n-1;\\
\kappa(L^*) &= \begin{cases} 1, \textit{ if } n \in \{1,2\},\\ n, \textit{ otherwise}; \end{cases}\\
\kappa(L^R) &= n.
\end{aligned}
$$

PROOF. If L is a unary ideal, then $L = a^{n-1}a^*$ for some $n \geq 1$, and its minimal generator is a^{n-1}.

We prove the result for L^*. If $n = 1$ or $n = 2$, then $L^* = a^*$, and so $\kappa(L^*) = 1$. If $n \geq 3$, then $L^* = \varepsilon \cup a^{n-1}a^*$, and $\kappa(L^*) = n$. $\square$

6 Conclusions

We have presented a rather complete picture of the state/quotient complexity of operations on regular ideal languages. Tables 1–3 summarize our results. The complexities for regular languages are from [6,29,42]. The minimal alphabet sizes required to meet the bounds are shown in parentheses. As can be seen from the tables, binary alphabets cannot be replaced by unary alphabets. Also, the alphabet of $n-2$ letters for the ideal generated by a language cannot be decreased. We do not know whether the other alphabets can be decreased.

Table 1
Bounds on quotient complexity of generation and of generators.

	$f(G)$	$f(M)$	$\kappa(M)$
unary ideals	n	$n-1$	$n+1$
right	n (1)	n (2)	$n+1$ (1)
left	2^{n-1} (2)	2^{n-2} (2)	$n(n+1)/2+2$ (2)
2-sided	$2^{n-2}+1$ (2)	$2^{n-3}+1$ (2)	$3+(n-1)(n-2)/2$ (3)
all-sided	$2^{n-2}+1$ $(n-2)$	$2^{n-3}+1$ $(n-3)$	open

Table 2
Bounds on quotient complexity of boolean operations.

	$K \cup L$	$K \cap L$	$K \setminus L$	$K \oplus L$
unary ideals	$\min(m,n)$	$\max(m,n)$	n	$\max(m,n)$
right, 2-, all-sided	$mn-(m+n-2)$ (2)	mn (2)	$mn-(m-1)$ (2)	mn (2)
left ideals	mn (4)	mn (2)	mn (3)	mn (2)
regular languages	mn (2)	mn (2)	mn (2)	mn (2)

Table 3
Bounds on quotient complexity of product, star and reversal.

	KL	L^*	L^R
unary ideals	$m+n-1$	n	n
right	$m+2^{n-2}$ (2)	$n+1$ (2)	2^{n-1} (2)
left	$m+n-1$ (1)	$n+1$ (2)	$2^{n-1}+1$ (3)
2-sided	$m+n-1$ (1)	$n+1$ (2)	$2^{n-2}+1$ (3)
all-sided	$m+n-1$ (1)	$n+1$ (2)	$2^{n-2}+1$ $(2n-4)$
regular	$m2^n-2^{n-1}$ (2)	$2^{n-1}+2^{n-2}$ (2)	2^n (2)

Acknowledgments

We are grateful to the anonymous referees for their constructive comments. We also thank Maxime Crochemore and Frank Tompa for providing references to pattern matching.

References

[1] A. Aho, M. J. Corasick, Efficient string matching: An aid to bibliographic search, Communications of the ACM 18 (6) (1975) 333–340.

[2] T. Ang, J. Brzozowski, Languages convex with respect to binary relations, and their closure properties, Acta Cybernet. 19 (2) (2009) 445–464.

[3] H. Bordihn, M. Holzer, M. Kutrib, Determination of finite automata accepting subregular languages, Theoret. Comput. Sci. 410 (2009) 3209–3249.

[4] J. Brzozowski, Canonical regular expressions and minimal state graphs for definite events, in: Proceedings of the Symposium on Mathematical Theory of Automata, vol. 12 of MRI Symposia Series, Polytechnic Press, Polytechnic Institute of Brooklyn, N.Y., 1963, pp. 529–561.

[5] J. Brzozowski, Derivatives of regular expressions, J. ACM 11 (4) (1964) 481–494.

[6] J. Brzozowski, Quotient complexity of regular languages, J. Autom. Lang. Comb. 15 (1/2) (2010) 71–89.

[7] J. Brzozowski, G. Jirásková, B. Li, Quotient complexity of ideal languages, `http://arxiv.org/abs/0908.2083` (2009).

[8] J. Brzozowski, G. Jirásková, B. Li, Quotient complexity of ideal languages, in: A. López-Ortiz (ed.), Proceedings of the 9th Latin American Theoretical Informatics Symposium, (LATIN), vol. 6034 of LNCS, Springer, 2010, pp. 208–211.

[9] J. Brzozowski, G. Jirásková, B. Li, J. Smith, Quotient complexity of bifix-, factor-, and subword-free regular languages, in: P. Dömösi, I. Szabolcs (eds.), Proceedings of the 13th International Conference on Automata and Formal Languages, (AFL), Institute of Mathematics and Informatics, College of Nyíregyháza, Nyíregyháza, Hungary, 2011, pp. 123–137, full paper at `http://arxiv.org/abs/1006.4843`.

[10] J. Brzozowski, G. Jirásková, C. Zou, Quotient complexity of closed languages, in: F. Ablayev, E. Mayr (eds.), Proceedings of the 5th International Computer Science Symposium in Russia, (CSR), vol. 6072 of LNCS, Springer, 2010, pp. 84–95.

[11] J. Brzozowski, B. Liu, Quotient complexity of star-free languages, in: P. Dömösi, I. Szabolcs (eds.), 13th International Conference on Automata and Formal Languages (AFL), Institute of Mathematics and Informatics, College of Nyíregyháza, Debrecen, Hungary, 2011, pp. 138–152.

[12] J. Brzozowski, J. Shallit, Z. Xu, Decision problems for convex languages, Inform. and Comput. 209 (2011) 353–367.

[13] J. Brzozowski, Y. Ye, Syntactic complexity of ideal and closed languages, in: G. Mauri, A. Leporati (eds.), Proceedings of the 15th International Conference on Developments in Language Theory (DLT), vol. 6795 of LNCS, Springer, 2011, pp. 117–128.

[14] C. Câmpeanu, K. Culik II, K. Salomaa, S. Yu, State complexity of basic operations on finite languages, in: O. Boldt, H. Jürgensen (eds.), Revised Papers from the 4th International Workshop on Automata Implementation, (WIA), vol. 2214 of LNCS, Springer, 2001, pp. 60–70.

[15] M. Crochemore, C. Hancart, Automata for pattern matching, in: G. Rozenberg, A. Salomaa (eds.), Handbook of Formal Languages, vol. 2, Springer, 1997, pp. 399–462.

[16] M. Crochemore, C. Hancart, T. Lecroq, Algorithms on Strings, Cambridge University Press, 2007, 392 pages.

[17] G. Gruber, M. Holzer, M. Kutrib, More on the size of Higman-Haines sets: effective constructions., Fundam. Inform. 91 (1) (2009) 105–121.

[18] L. H. Haines, On free monoids partially ordered by embedding, J. Combin. Theory 6 (1) (1969) 94–98.

[19] Y.-S. Han, K. Salomaa, State complexity of basic operations on suffix-free regular languages, Theoret. Comput. Sci. 410 (27-29) (2009) 2537–2548.

[20] 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, University of Szeged, Hungary, 2009, pp. 99–115.

[21] 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 International Conference on Implementation and Application of Automata (CIAA), vol. 6807 of LNCS, Springer, Berlin Heidelberg, 2011, pp. 178–189.

[22] G. Jirásková, State complexity of some operations on binary regular languages, Theoret. Comput. Sci. 330 (2005) 287–298.

[23] G. Jirásková, T. Masopust, Complexity in union-free regular languages, Internat. J. Found. Comput. Sci. 22 (7) (2011) 1639–1653.

[24] G. Jirásková, J. Šebej, Reversal of binary regular languages, Theoret. Comput. Sci. 449 (2012) 85–92.

[25] E. Leiss, Succinct representation of regular languages by boolean automata, Theoret. Comput. Sci. 13 (2009) 323–330.

[26] A. de Luca, S. Varricchio, Finiteness and Regularity in Semigroups and Formal Languages, Springer, 1990.

[27] A. de Luca, S. Varricchio, Some combinatorial properties of factorial languages, in: R. Capocelli (ed.), Sequences: Combinatorics, Compression, Security, and Transmission, Springer, 1990, pp. 258–266.

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

[29] A. N. Maslov, Estimates of the number of states of finite automata, Dokl. Akad. Nauk SSSR 194 (1970) 1266–1268 (Russian), English translation: Soviet Math. Dokl. 11 (1970), 1373–1375.

[30] B. G. Mirkin, On dual automata, Kibernetika (Kiev) 2 (1966) 7–10 (Russian), English translation: Cybernetics 2, (1966) 6–9.

[31] A. Okhotin, On the state complexity of scattered substrings and superstrings, Tech. Rep. 849, Turku Centre for Computer Science (2007).

[32] A. Paz, B. Peleg, Ultimate-definite and symmetric-definite events and automata, J. ACM 12 (3) (1965) 399–410.

[33] D. Perrin, Finite automata, in: J. van Leewen (ed.), Handbook of Theoretical Computer Science, vol. B, Elsevier, 1990, pp. 1–57.

[34] G. Pighizzini, J. Shallit, Unary language operations, state complexity and Jacobsthal's function, Internat. J. Found. Comput. Sci. 13 (2002) 145–159.

[35] E. Pribavkina, E. Rodaro, State complexity of code operators, Internat. J. Found. Comput. Sci. 22 (7) (2011) 1669–1681.

[36] M. Rabin, D. Scott, Finite automata and their decision problems, IBM J. Res. and Dev. 3 (1959) 114–129.

[37] H. J. Shyr, Free Monoids and Languages, Hon Min Book Co, Taiwan, 2001.

[38] G. Thierrin, Convex languages, in: M. Nivat (ed.), Automata, Languages and Programming, North-Holland, 1973, pp. 481–492.

[39] F. Yu, Z. Chen, Y. Diao, T. V. Lakshman, R. H. Katz, Fast and memory-efficient regular expression matching for deep packet inspection, in: Proceedings of the 2006 ACM/IEEE Symposium on Architecture for Networking and Communications Systems (ANCS), ACM, New York, 2006, pp. 93–102.

[40] S. Yu, Regular languages, in: G. Rozenberg, A. Salomaa (eds.), Handbook of Formal Languages, vol. 1, Springer, 1997, pp. 41–110.

[41] S. Yu, State complexity of regular languages, J. Autom. Lang. Comb. 6 (2001) 221–234.

[42] S. Yu, Q. Zhuang, K. Salomaa, The state complexities of some basic operations on regular languages, Theoret. Comput. Sci. 125 (2) (1994) 315–328.