# Syntactic Complexity of Ideal and Closed Languages *

Janusz Brzozowski[1] and Yuli Ye[2]

[1] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1
{brzozo@uwaterloo.ca }

[2] Department of Computer Science, University of Toronto
Toronto, ON, Canada M5S 3G4
{y3ye@cs.toronto.edu}

**Abstract.** The state complexity of a regular language is the number of states in the minimal deterministic automaton accepting the language. The syntactic complexity of a regular language is the cardinality of its syntactic semigroup. The syntactic complexity of a subclass of regular languages is the worst-case syntactic complexity taken as a function of the state complexity $n$ of languages in that class. We prove that $n^{n-1}$ is a tight upper bound on the complexity of right ideals and prefix-closed languages, and that there exist left ideals and suffix-closed languages of syntactic complexity $n^{n-1}+n-1$, and two-sided ideals and factor-closed languages of syntactic complexity $n^{n-2}+(n-2)2^{n-2}+1$.

**Keywords:** automaton, closed, complexity, ideal, language, monoid, regular, reversal, semigroup, syntactic.

## 1 Introduction

There are two fundamental congruence relations in the theory of regular languages: the Nerode congruence [16], and the Myhill congruence [15]. In both cases, a language is regular if and only if it is a union of congruence classes of a congruence of finite index. The Nerode congruence leads to the definitions of left quotients of a language and the minimal deterministic finite automaton (DFA) recognizing the language. The Myhill congruence leads to the definitions of the syntactic semigroup and the syntactic monoid of the language.

The *state complexity* of a language is the number of states in the minimal DFA recognizing the language. This concept has been studied quite extensively; for surveys and references see [4,25]. Syntactic complexity of various types of graph languages has been studied by Bozapalidis and Kalampakas [1,2,3,11] in the framework of magmoids. However, in spite of suggestions that syntactic

* This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant No. OGP0000871 and a Postgraduate Scholarship, and by a Graduate Award from the Department of Computer Science, University of Toronto.

G. Mauri and A. Leporati (Eds.): DLT 2011, LNCS 6795, pp. 117–128, 2011.

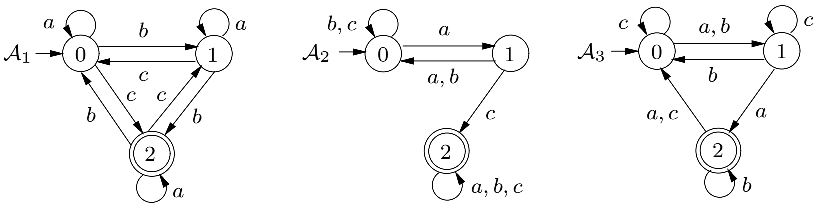


**Fig. 1.** Automata with various syntactic complexities

semigroups of subsets of free monoids deserve to be studied [9,13], little has been done on the syntactic complexity of a regular language, which we define as the cardinality of its syntactic semigroup. This semigroup is isomorphic to the semigroup of transformations of the set of states of the minimal DFA recognizing the language, where these transformations are performed by non-empty words.

The following example illustrates the significant difference between state complexity and syntactic complexity. The DFA's in Fig. 1 have the same alphabet, are all minimal, and all have the same state complexity. However, the syntactic complexity of $\mathcal{A}_1$ is 3, that of $\mathcal{A}_2$ is 9, and that of $\mathcal{A}_3$ is 27. This shows that syntactic complexity is a much finer measure of complexity than state complexity. The question then arises: Is it possible to find upper bounds to the syntactic complexity of a regular language from its properties or from the properties of its minimal DFA? We shed some light on this question for ideal and closed regular languages.

The set of all $n^n$ transformations of a set $Q$ is a monoid under composition of transformations, with identity as the unit element. The set of all $n!$ permutations of $Q$ is a group, the *symmetric* group of degree $n$. The fact that two generators are sufficient to form a basis for the set of all permutations of a set of $n$ elements has been known for many years. For example, such a result is stated in the 1895 paper by Hoyer [10], and later in the 1938 paper by Piccard [18]. The fact that three functions suffice to generate all transformations of a set of $n$ elements was proved in 1935 by Piccard [17]. Also in 1935, Eilenberg showed that no two functions suffice, as reported by Sierpiński [24]. Related later work includes that of Salomaa (1960-63) [20,21,22] and Dénes (1968) [8]. In 1970, Maslov [14] dealt with the generators of the semigroup of all transformations in the setting of finite automata. Holzer and König [9], and independently Krawetz, Lawrence and Shallit [12] studied the syntactic complexity of automata with unary and binary alphabets. See also the recent work in [13,23].

The state complexity in prefix-, suffix-, factor-, and subword-closed languages was studied in 2010 by Brzozowski, Jirásková and Zou [6]. A study of state complexity in ideal languages was done in 2010 by Brzozowski, Jirásková and Li [5]. We refer the reader to these papers for a discussion of past work on this topic and additional references. Closed languages are related to ideal languages as follows: A non-empty language is a right (left, two-sided) ideal if and only its complement is a prefix(suffix, factor)-closed language. Since syntactic complexity is preserved under complementation, our proofs use ideals only.

In Section 2 we define our terminology and notation, and some basic properties of syntactic complexity are given in Section 3. The syntactic complexity of right, left, and two-sided ideals is treated in Sections 4–6, some comments about reversal are in Section 7, and Section 8 concludes the paper. For a full version of this paper and omitted proofs see [7].

## 2 Preliminaries

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

If $w = uxv$ for some $u, v, x \in \Sigma^*$, then $u$ is a *prefix* of $w$, $v$ is a *suffix* of $w$, and $x$ is a *factor* of $w$. A prefix or suffix of $w$ is also a factor of $w$. A language $L$ is *prefix-closed* if $w \in L$ implies that every prefix of $w$ is also in $L$. In an analogous way, we define *suffix-closed* and *factor-closed*. A language $L \subseteq \Sigma^*$ is a *right ideal* (respectively, *left ideal*, *two-sided ideal*) if it is non-empty and satisfies $L = L\Sigma^*$ (respectively, $L = \Sigma^* L$, $L = \Sigma^* L \Sigma^*$). We refer to all three types as *ideal languages* or simply *ideals*.

A *transformation* of a set $Q$ is a mapping of $Q$ *into* itself, whereas a *permutation* of $Q$ is a mapping of $Q$ *onto* itself. In this paper we consider only transformations of finite sets, and we assume without loss of generality that $Q = \{0, 1, \dots, n-1\}$. An arbitrary transformation has the form

$$t = \begin{pmatrix} 0 & 1 & \cdots & n-2 & n-1 \\ i_0 & i_1 & \cdots & i_{n-2} & i_{n-1} \end{pmatrix},$$

where $i_k \in Q$ for $0 \leqslant k \leqslant n-1$. The image of element $i$ under transformation $t$ will be denoted by $it$. The *identity* transformation maps each element to itself, that is, $it = i$ for $i = 0, \dots, n-1$. A transformation $t$ contains a *cycle* of length $k$ if there exist elements $i_1, \dots, i_k$ such that $i_1 t = i_2, i_2 t = i_3, \dots, i_{k-1} t = i_k, i_k t = i_1$. A cycle is denoted by $(i_1, i_2, \dots, i_k)$. For $i < j$, a *transposition* is the cycle $(i, j)$, and $(i, i)$ is the identity. A *singular* transformation, denoted by $\binom{i}{j}$, has $it = j$, and $ht = h$ for all $h \neq i$, and $\binom{i}{i}$ is the identity. A *constant* transformation, denoted by $\binom{Q}{j}$, has $it = j$ for all $i$. The following facts are well-known:

**Theorem 1.** *The complete transformation monoid $T_n$ of size $n^n$ can be generated by any cyclic permutation of $n$ elements together with a transposition and a "returning" transformation $r = \binom{n-1}{0}$. In particular, $T_n$ can be generated by $c = (0, 1, \dots, n-1)$, $t = (0, 1)$ and $r = \binom{n-1}{0}$.*

The *left quotient*, or simply *quotient,* of a language $L$ by a word $w$ is the language $L_w = \{x \in \Sigma^* \mid wx \in L\}$. An equivalence relation $\sim$ on $\Sigma^*$ is a *left congruence* if, for all $x, y \in \Sigma^*$, $x \sim y \Leftrightarrow ux \sim uy$, for all $u \in \Sigma^*$. It is a *right congruence* if, for all $x, y \in \Sigma^*$, $x \sim y \Leftrightarrow xv \sim yv$, for all $v \in \Sigma^*$.

It is a *congruence* if it is both a left and a right congruence. Equivalently, $\sim$ is a congruence if $x \sim y \Leftrightarrow uxv \sim uyv$, for all $u, v \in \Sigma^*$.

For any language $L \subseteq \Sigma^*$, define the *Nerode congruence* [16] $\rightarrow_L$ of $L$ by

$$x \rightarrow_L y \text{ if and only if } xv \in L \Leftrightarrow yv \in L, \text{ for all } u, v \in \Sigma^*. \quad (1)$$

Evidently, $L_x = L_y$ if and only if $x \rightarrow_L y$. Thus, each equivalence class of this congruence corresponds to a distinct quotient of $L$.

The *Myhill congruence* [15] $\leftrightarrow_L$ of $L$ is defined by

$$x \leftrightarrow_L y \text{ if and only if } uxv \in L \Leftrightarrow uyv \in L \text{ for all } u, v \in \Sigma^*. \quad (2)$$

This congruence is also known as the *syntactic congruence* of $L$. The semigroup $\Sigma^+/\leftrightarrow_L$ of equivalence classes of the relation $\leftrightarrow_L$, is the *syntactic semigroup* of $L$, and $\Sigma^*/\leftrightarrow_L$ is the *syntactic monoid* of $L$. The *syntactic complexity* $\sigma(L)$ of $L$ is the cardinality of its syntactic semigroup. The *monoid complexity* $\mu(L)$ of $L$ is the cardinality of its syntactic monoid. If the equivalence class of $\varepsilon$ is a singleton in the syntactic monoid, then $\sigma(L) = \mu(L) - 1$; otherwise, $\sigma(L) = \mu(L)$.

A *(deterministic) semiautomaton* is a triple, $\mathcal{S} = (Q, \Sigma, \delta)$, where $Q$ is a finite, non-empty set of *states*, $\Sigma$ is a finite non-empty *alphabet*, and $\delta : Q \times \Sigma \rightarrow Q$ is the *transition function*. A *deterministic finite automaton (DFA)* is a quintuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where $Q$, $\Sigma$, and $\delta$ are as defined in the semiautomaton $\mathcal{S} = (Q, \Sigma, \delta)$, $q_0 \in Q$ is the *initial state*, and $F \subseteq Q$ is the set of *final states*. By the *language of a state* $q$ of $\mathcal{A}$ we mean the language $L(\mathcal{A}_q)$ accepted by the automaton $\mathcal{A}_q = (Q, \Sigma, \delta, q, F)$. States $p$ and $q$ are *equivalent* if $L(\mathcal{A}_p) = L(\mathcal{A}_q)$. A DFA is *minimal* if every state is reachable from the initial state, and no two states are equivalent.

The *$\varepsilon$-function* $L^\varepsilon$ of a regular language $L$ is $L^\varepsilon = \emptyset$ if $\varepsilon \notin L$; $L^\varepsilon = \{\varepsilon\}$ if $\varepsilon \in L$. The *quotient automaton* of a regular language $L$ is $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{L_w \mid w \in \Sigma^*\}$, $\delta(L_w, a) = L_{wa}$, $q_0 = L_\varepsilon = L$, $F = \{L_w \mid L_w^\varepsilon = \{\varepsilon\}\}$, and $L_w^\varepsilon = (L_w)^\varepsilon$. The number of states in the quotient automaton of $L$ is the *quotient complexity* $\kappa(L)$ of $L$. The quotient complexity is the same as the state complexity, but there are advantages to using quotients [4].

In terms of automata, each equivalence class $[w]_{\rightarrow_L}$ is the set of words $w$ that take the automaton to the same state from the initial state. In terms of quotients, it is the set of words $w$ that can all be followed by the same quotient $L_w$. In terms of automata, each equivalence class $[w]_{\leftrightarrow_L}$ of the syntactic congruence is the set of all words that perform the same transformation on the set of states.

## 3 Basic Properties of Syntactic Complexity

The *transformation semigroup* of an automaton is the set of transformations performed by words of $\Sigma^+$ on the set of states. The transformation semigroup of the quotient automaton of $L$ is isomorphic to the syntactic semigroup of $L$.

**Proposition 1.** *For any $L \subseteq \Sigma^*$ with $\kappa(L) = n > 1$, $n - 1 \leqslant \sigma(L) \leqslant n^n$.*

*Proof.* Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be the quotient automaton of $L$. Since every state other than $q_0$ has to be reachable from the initial state by a non-empty word, there must be at least $n-1$ transformations. If $\Sigma = \{a\}$ and $L = a^{n-1}a^*$, then $\kappa(L) = n$, and $\sigma(L) = n-1$; so the lower bound $n-1$ is achievable. The upper bound is $n^n$, and by Theorem 1 this upper bound is achievable if $|\Sigma| \geqslant 3$. □

If one of the quotients of $L$ is $\emptyset$ (respectively, $\{\varepsilon\}$, $\Sigma^*$, $\Sigma^+$), then we say that *$L$ has $\emptyset$* (respectively, $\{\varepsilon\}$, $\Sigma^*$, $\Sigma^+$). A quotient $L_w$ of a language $L$ is *uniquely reachable (ur)* [4] if $L_x = L_w$ implies that $x = w$. If $L_{wa}$ is uniquely reachable for $a \in \Sigma$, then so is $L_w$. Thus, if $L$ has a uniquely reachable quotient, then $L$ itself is uniquely reachable by $\varepsilon$, *i.e.*, the minimal automaton of $L$ is *non-returning.*

**Theorem 2 (Special Quotients).** *Let $L \subseteq \Sigma^*$ and let $\kappa(L) = n \geqslant 1$.*
*1. If $L$ has $\emptyset$ or $\Sigma^*$, then $\sigma(L) \leqslant n^{n-1}$.*
*2. If $L$ has $\{\varepsilon\}$ or $\Sigma^+$, then $\sigma(L) \leqslant n^{n-2}$.*
*3. If $L$ is uniquely reachable, then $\sigma(L) \leqslant (n-1)^n$.*
*4. If $L_a$ is uniquely reachable for some $a \in \Sigma$, then $\sigma(L) \leqslant 1 + (n-2)^n$.*
*Moreover, all the bounds shown in Table 1 hold.*

**Table 1.** Upper bounds on syntactic complexity for languages with special quotients

| $\emptyset$ | $\Sigma^*$ | $\{\varepsilon\}$ | $\Sigma^+$ | | $L$ is ur | $L_a$ is ur |
|---|---|---|---|---|---|---|
| √ | | | | $n^{n-1}$ | $(n-1)^{n-1}$ | $1+(n-3)^{n-2}$ |
| | √ | | | $n^{n-1}$ | $(n-1)^{n-1}$ | $1+(n-3)^{n-2}$ |
| √ | | √ | | $n^{n-2}$ | $(n-1)^{n-2}$ | $1+(n-4)^{n-2}$ |
| | √ | | √ | $n^{n-2}$ | $(n-1)^{n-2}$ | $1+(n-4)^{n-2}$ |
| √ | √ | | | $n^{n-2}$ | $(n-1)^{n-2}$ | $1+(n-4)^{n-2}$ |
| √ | √ | | √ | $n^{n-3}$ | $(n-1)^{n-3}$ | $1+(n-5)^{n-2}$ |
| √ | √ | √ | | $n^{n-3}$ | $(n-1)^{n-3}$ | $1+(n-5)^{n-2}$ |
| √ | √ | √ | √ | $n^{n-4}$ | $(n-1)^{n-4}$ | $1+(n-6)^{n-2}$ |

*Proof.* Suppose that $L \subseteq \Sigma^*$, $n \geqslant 1$, and $\kappa(L) = n$.

1. Since $\emptyset_a = \emptyset$ for all $a \in \Sigma$, there are only $n-1$ states in the quotient automaton with which one can distinguish two transformations. Hence there are at most $n^{n-1}$ transformations. If $L$ has $\Sigma^*$, then $(\Sigma^*)_a = \Sigma^*$, for all $a \in \Sigma$, and the same argument applies.

2. Since $\{\varepsilon\}_a = \emptyset$ for all $a \in \Sigma$, $L$ has $\emptyset$ if $L$ has $\{\varepsilon\}$. Now there are two states that do not contribute to distinguishing among different transformations. Dually, $(\Sigma^+)_a = \Sigma^*$ for all $a \in \Sigma$, and the same argument applies.

3. If $L$ is uniquely reachable then $L_w = L$ implies $w = \varepsilon$. Thus $L$ does not appear in the image of any transformation by a word in $\Sigma^+$, and there remain only $n-1$ choices for each of the $n$ states.

4. If $L_a$ is uniquely reachable, then so is $L$. Hence $L$ never appears and $L_a$ appears only in one transformation. Therefore there can be at most $(n-2)^n$ other transformations.

The remaining entries in Table 1 are easily verified. □

# 4 Right Ideals and Prefix-Closed Languages

In this section we characterize the syntactic complexity of right ideals. The DFA of Fig. 2 plays an important role here. For $n \geqslant 4$, let $\mathcal{A}_n = (Q_n, \Sigma, \delta, 0, \{n-1\})$, where $Q_n = \{0, 1, \dots, n-1\}$, $\Sigma = \{a, b, c, d\}$, $a = (0, 1, \dots, n-2)$, $b = (0, 1)$, $c = \binom{n-2}{0}$, and $d = \binom{n-2}{n-1}$. This DFA accepts a right ideal and is minimal.

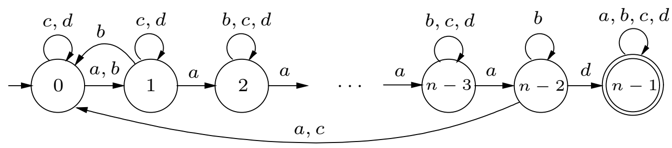


**Fig. 2.** Automaton $\mathcal{A}_n$ of a right ideal with $n^{n-1}$ transformations

**Theorem 3 (Right Ideals and Prefix-Closed Languages).** *Suppose that $L \subseteq \Sigma^*$ and $\kappa(L) = n$. If $L$ is a right ideal or a prefix-closed language, then $\sigma(L) \leqslant n^{n-1}$. Moreover, this bound is tight for $n = 1$ if $|\Sigma| \geqslant 1$, for $n = 2$ if $|\Sigma| \geqslant 2$, for $n = 3$ if $|\Sigma| \geqslant 3$, and for $n \geqslant 4$ if $|\Sigma| \geqslant 4$.*

*Proof.* If $L$ is a right ideal, it has $\Sigma^*$ as a quotient. By Theorem 2, $\sigma(L) \leqslant n^{n-1}$.

For $n = 1$, $n = 2$, and $n = 3$, the languages $a^*$, $b^*a\{a, b\}^*$ and $\mathcal{A}_3$ restricted to alphabet $\{a, c, d\}$, respectively, meet the bound.

Next we prove that the language $L(\mathcal{A}_n)$ accepted by the automaton of Fig. 2 meets this bound for $n \geqslant 4$. Consider any transformation $t$ of the form

$$t = \begin{pmatrix} 0 & 1 & 2 & \cdots & n-3 & n-2 & n-1 \\ i_0 & i_1 & i_2 & \cdots & i_{n-3} & i_{n-2} & n-1 \end{pmatrix},$$

where $i_k \in \{0, 1, \dots, n-1\}$ for $0 \leqslant k \leqslant n-2$. There are two cases:
1. Suppose $i_k \neq n-1$ for all $k$, $0 \leqslant k \leqslant n-2$. By Theorem 1, since all the images of the first $n-1$ states are in $\{0, 1, \dots, n-2\}$, $\mathcal{A}_n$ can do $t$.
2. If $i_h = n-1$ for some $h$, $0 \leqslant h \leqslant n-2$, then there exists some $j$, $0 \leqslant j \leqslant n-2$ such that $i_k \neq j$ for all $k$, $0 \leqslant k \leqslant n-2$. Define $i'_k$ for all $0 \leqslant k \leqslant n-2$ as follows: $i'_k = j$ if $i_k = n-1$, and $i'_k = i_k$ if $i_k \neq n-1$. Then let

$$s = \begin{pmatrix} 0 & 1 & 2 & \cdots & n-3 & n-2 & n-1 \\ i'_0 & i'_1 & i'_2 & \cdots & i'_{n-3} & i'_{n-2} & n-1 \end{pmatrix}.$$

Also, let $r = (j, n-2)$. Since all the images of the first $n-1$ states in $s$ and $r$ are in $\{0, 1, \dots, n-2\}$, by Theorem 1, $s$ and $r$ can be performed by $\mathcal{A}_n$.

We show now that $t = srdr$, which implies that $t$ can also be performed by $\mathcal{A}_n$. If $kt = n-1$, then $ks = j$, $jr = n-2$, $(n-2)d = n-1$, and $(n-1)r = n-1$. If $kt = n-2$, then $n-2 \neq j$. Now $ks = n-2$, $(n-2)r = j$, $jd = j$, and $jr = n-2$. If $kt = i_k < n-2$, then also $k(srdr) = i_k$. In all cases $t = srdr$.

Since there are $n^{n-1}$ transformations like $t$, $L(\mathcal{A}_n)$ meets the bound. □

**Table 2.** Syntactic complexity bounds for right ideals

| | $n=1$ | $n=2$ | $n=3$ | $n=4$ | $n=5$ | ... | $n=n$ |
|---|---|---|---|---|---|---|---|
| $\|\Sigma\|=1$ | **1** | **1** | **2** | **3** | **4** | ... | $\mathbf{n-1}$ |
| $\|\Sigma\|=2$ | – | **2** | **7** | **31** | **167** | ... | ? |
| $\|\Sigma\|=3$ | – | – | **9** | **61** | **545** | ... | ? |
| $\|\Sigma\|=4$ | – | – | – | **64** | **625** | ... | $\mathbf{n^{n-1}}$ |

Table 2 summarizes our result for right ideals. All the bounds are tight. The bounds for $n \leqslant 5$ have all been verified by a computer program.

## 5 Left Ideals and Suffix-Closed Languages

We provide strong support for the following conjecture:

**Conjecture 1 (Left Ideals and Suffix-Closed Languages).** *If $L$ is a left ideal or a suffix-closed language with $\kappa(L) = n \geqslant 1$, then $\sigma(L) \leqslant n^{n-1} + n - 1$.*

We show that the bound can be reached, but first we recall a result from [19]. Consider a semiautomaton $\mathcal{S} = (P \cup \{0\}, \Sigma, \delta)$, where 0 is a sink state, meaning that $\delta(0, a) = 0$ for all $a \in \Sigma$, and $P$ is strongly connected. Such a semiautomaton is *uniformly minimal* if the automaton $\mathcal{A} = (P \cup \{0\}, \Sigma, \delta, q_0, F)$ is minimal for every $q_0 \in P$ and every $F$ such that $\emptyset \neq F \subseteq P$. One can test whether a semiautomaton is uniformly minimal by the method of [19].

Let $n \geqslant 3$, and $\mathcal{S}_n = (Q, \Sigma, \delta)$, where $Q = \{0, \ldots, n-1\}$, $\Sigma = \{a, b, c, d, e\}$, $a = (1, 2, \ldots, n-1)$, $b = (1, 2)$, $c = \binom{n-1}{1}$, $d = \binom{n-1}{0}$, and $e = \binom{Q}{1}$; see Fig. 3. For $n = 3$, $a$ and $b$ coincide; then we use $\Sigma = \{b, c, d, e\}$. Let $\Sigma' = \Sigma \setminus \{e\}$ and let $\mathcal{R}_n$ be the semiautomaton $\mathcal{R}_n = (Q, \Sigma', \delta')$, where $Q = P \cup \{0\}$, $P = \{1, \ldots, n-1\}$, and $\delta'$ is the restriction of $\delta$ to $Q \times \Sigma'$. Note that 0 is a sink state of $\mathcal{R}_n$.

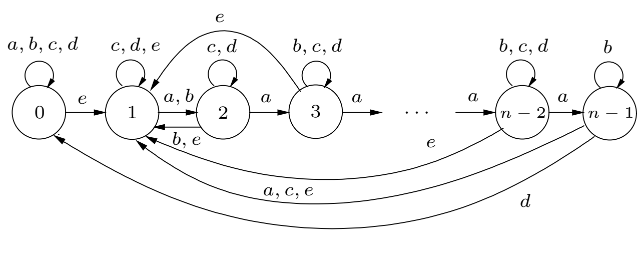


**Fig. 3.** Semiautomaton $\mathcal{S}_n$ with $n^{n-1} + n - 1$ transformations

**Lemma 1.** *The set $P$ is strongly connected and $\mathcal{R}_n$ is uniformly minimal.*

*Proof.* Since $a$ is a cycle of the states in $P$, $\mathcal{R}_n$ is strongly connected. It can be shown that $\mathcal{R}_n$ is uniformly minimal by using a state-pair graph as in [19]. $\square$

**Theorem 4 (Left Ideals and Suffix-Closed Languages).** *For $n \geqslant 3$, let $\mathcal{B}_n = (Q, \Sigma, \delta, 0, F)$, where $(Q, \Sigma, \delta) = \mathcal{S}_n$ of Fig. 3, and $F$ is any non-empty subset of $Q \setminus \{0\}$. Then $\mathcal{B}_n$ is minimal, and the language $L = L(\mathcal{B}_n)$ accepted by $\mathcal{B}_n$ is a left ideal and has syntactic complexity $\sigma(L) = n^{n-1} + n - 1$.*

*Proof.* Since semiautomaton $\mathcal{R}_n$ is uniformly minimal, automaton $\mathcal{B}_n$ is minimal for every choice of $F$. Hence $L$ has $n$ quotients.

To prove that $L$ is a left ideal it suffices to show that, for any $w \in L$, we also have $hw \in L$ for every $h \in \Sigma$. This is obvious if $h \in \Sigma \setminus \{e\}$, since all transitions from state 0 under $h$ lead to state 0. If $w \in L$, then $w$ has the form $w = uev$, where $\delta(0, u) = 0$, $\delta(0, ue) = 1$, and $v \in L_e$. But $\delta(0, eue) = 1$, since $\delta(i, eue) = 1$ for all $i \in Q$, and $v \in L_e$ gives us $euev = ew \in L$. Thus $L$ is a left ideal.

Consider any transformation $t$ of the form

$$t = \begin{pmatrix} 0 & 1 & 2 & \cdots & n-3 & n-2 & n-1 \\ 0 & i_1 & i_2 & \cdots & i_{n-3} & i_{n-2} & i_{n-1} \end{pmatrix},$$

where $i_k \in \{0, \ldots, n-1\}$ for $1 \leqslant k \leqslant n-1$; there are $n^{n-1}$ such transformations. We have two cases:

1. If $i_k \neq 0$ for all $k$, $1 \leqslant k \leqslant n-1$, then all the images of the last $n-1$ states are in the set $\{1, \ldots, n-1\}$. By Theorem 1, $t$ can be performed by $\mathcal{B}_n$.
2. If $i_h = 0$ for some $h$, $1 \leqslant h \leqslant n-1$, then there exists some $j$, $1 \leqslant j \leqslant n-1$ such that $i_k \neq j$ for all $k$, $1 \leqslant k \leqslant n-1$. Define $i'_k$ for all $1 \leqslant k \leqslant n-1$ as follows: $i'_k = j$ if $i_k = 0$, and $i'_k = i_k$, otherwise. Let

$$s = \begin{pmatrix} 0 & 1 & 2 & \cdots & n-3 & n-2 & n-1 \\ 0 & i'_1 & i'_2 & \cdots & i'_{n-3} & i'_{n-2} & i'_{n-1} \end{pmatrix},$$

and $r = (j, n-1)$. By Theorem 1, $s$ and $r$ can be performed by $\mathcal{B}_n$.

Now consider $srdr$. If $kt = 0$, then $ks = j$, $jr = n-1$, $(n-1)d = 0$, and $0r = 0$. If $kt = n-1$, then $ks = n-1$, $(n-1)r = j$, $jd = j$, and $jr = n-1$. Finally, if $kt$ is a state other than 0 or $n-1$, then $srdr$ maps $k$ to that same state. Hence we have $t = srdr$, and $t$ can be performed by $\mathcal{B}_n$ as well.

Now consider any transformation $t = \binom{Q}{j}$ that maps all the states to some state $j \neq 0$; there are $n-1$ such transformations. We have two cases:

1. If $j = 1$, then $t = e$; therefore $t$ can be performed by $\mathcal{B}_n$.
2. Otherwise, let $s = (1, j)$. By Theorem 1, $s$ can be performed by $\mathcal{B}_n$. Since $t = es$, $t$ can also be performed by $\mathcal{B}_n$ as well.

There are no other transformations, since $e$ maps all the states to 1. □

Before considering the cases $n \leqslant 3$, we require some auxiliary results. Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be the quotient automaton of a left ideal. For every word $w \in \Sigma^*$, consider the sequence $q_0 = p_0, p_1, p_2 \ldots$ of states obtained by applying powers of $w$ to the initial state $q_0$, that is, let $p_i = \delta(q_0, w^i)$. Since $\mathcal{A}$ has $n$ states, we must eventually have a repeated state in that sequence, that is, we must have some $i$ and $j > i$ such that $p_0, p_1, \ldots, p_i, p_{i+1}, \ldots p_{j-1}$ are distinct and $p_j = p_i$. The sequence $q_0 = p_0, p_1, \ldots, p_i, p_{i+1}, \ldots p_{j-1}$ of states with $p_j = p_i$ is called the

**Table 3.** Syntactic complexity bounds for left ideals

| | $n = 1$ | $n = 2$ | $n = 3$ | $n = 4$ | $n = 5$ | | $n = n$ |
|---|---|---|---|---|---|---|---|
| $\|\Sigma\| = 1$ | **1** | **1** | **2** | **3** | **4** | ... | $\mathbf{n-1}$ |
| $\|\Sigma\| = 2$ | – | **2** | **7** | 17 | 34 | ... | ? |
| $\|\Sigma\| = 3$ | – | **3** | **9** | 25 | 65 | ... | ? |
| $\|\Sigma\| = 4$ | – | – | **11** | 64 | 453 | ... | ? |
| $\|\Sigma\| = 5$ | – | – | – | 67 | 629 | ... | $n^{n-1} + n - 1$ |

*behavior of* $w$ *on* $\mathcal{A}$, and the integer $j - i$ is the *period* of that behavior. We will use the notation $\langle p_0, p_1, \ldots, p_i, p_{i+1}, \ldots p_{j-1}; p_j = p_i\rangle$ for such behaviors. If the period of $w$ is 1, then its behavior is *aperiodic*; otherwise, it is *periodic*.

**Lemma 2.** *If* $\mathcal{A}$ *is the quotient automaton of a left ideal* $L$*, then the behavior of every word* $w \in \Sigma^*$ *is aperiodic. Moreover,* $L$ *does not have the empty quotient.*

*Proof.* Suppose that $w$ has the behavior $\langle q_0 = p_0, p_1, \ldots, p_i, p_{i+1}, \ldots p_{j-1}; p_j = p_i\rangle$, where $j - i \geqslant 2$; then $j - 1 \geqslant i + 1$. Since $\mathcal{A}$ is minimal, states $p_i$ and $p_{j-1}$ must be distinguishable, say by word $x \in \Sigma^*$. If $w^i x \in L$, then $w^{j-1}x = w^i w^{j-i-1} x = w^{j-i-1}(w^i x) \notin L$, contradicting the assumption that $L$ is a left ideal. If $w^{j-1}x \in L$, then $w^j x = w(w^{j-1}x) \notin L$, again a contradiction.

For the second claim, a left ideal is non-empty by definition. If $w \in L$ and $L$ has the empty quotient, say $L_x = \emptyset$, then $xw \notin L$, which is a contradiction. □

**Theorem 5 (Left Ideals and Suffix-Closed Languages for $\mathbf{n \leqslant 3}$).** *If* $1 \leqslant n \leqslant 3$ *and* $L$ *is a left ideal or a suffix-closed language with* $\kappa(L) = n$*, then* $\sigma(L) \leqslant n^{n-1} + n - 1$*. Moreover, the bound is tight for* $n = 1$ *if* $|\Sigma| \geqslant 1$*, for* $n = 2$ *if* $|\Sigma| \geqslant 3$*, and for* $n = 3$ *if* $|\Sigma| \geqslant 4$*.*

Table 3 summarizes our results. The figures in bold type are tight bounds, verified by a computer program.

## 6 Two-Sided Ideals and Factor-Closed Languages

**Conjecture 2 (Two-Sided Ideals and Factor-Closed Languages).** *If* $L$ *is a two-sided ideal or a factor-closed language with* $\kappa(L) = n \geqslant 2$*, then* $\sigma(L) \leqslant n^{n-2} + (n-2)2^{n-2} + 1$*.*

We show that this complexity can be met. For $n = 2$ and $\Sigma = \{a, b\}$, $\Sigma^* a \Sigma^*$ meets the bound. For $n = 3$ and $\Sigma = \{a, b, c\}$, $(b + c + ac^*b)^* ac^* a \Sigma^*$ meets the bound. Now let $n \geqslant 4$, and let $\mathcal{C}_n = (Q, \Sigma, \delta, 0, \{n-1\})$, where $Q = \{0, \ldots, n-1\}$, $\Sigma = \{a, b, c, d, e, f\}$, $a = (1, 2, \ldots, n-2)$, $b = (1, 2)$, $c = \binom{n-2}{1}$, $d = \binom{n-2}{0}$, $\delta(i, e) = 1$ for $i = 0, \ldots, n-2$ and $\delta(n-1, e) = n-1$, and $f = \binom{1}{n-1}$. For $n = 4$, $a$ and $b$ coincide.

**Theorem 6 (Two-Sided Ideals and Factor-Closed Languages).** *DFA* $\mathcal{C}_n$ *is minimal and* $L = L(\mathcal{C}_n)$ *is a two-sided ideal with* $\sigma(L) = n^{n-2} + (n-2)2^{n-2} + 1$*.*

*Proof.* For $i = 1, \ldots, n-2$, state $i$ is the only non-final state that accepts $a^{n-1-i}f$; hence all these states are distinguishable. State 0 is distinguishable from these states because it does not accept any words in $a^*f$. Hence $\mathcal{C}_n$ is minimal. The proof that $\mathcal{C}_n$ is a left ideal is like that in Theorem 4. Since $L_{ef} = \Sigma^*$ is the only accepting quotient, $L$ is a right ideal. Hence it is two-sided.

First consider any transformation $t$ of the form

$$t = \begin{pmatrix} 0 & 1 & 2 & \cdots & n-3 & n-2 & n-1 \\ 0 & i_1 & i_2 & \cdots & i_{n-3} & i_{n-2} & n-1 \end{pmatrix},$$

where $i_k \in \{0, 1, 2, \ldots, n-2, n-1\}$ for $1 \leqslant k \leqslant n-2$; there are $n^{n-2}$ such transformations. We have two cases:

1. If $i_k \neq n-1$ for all $k$, $1 \leqslant k \leqslant n-2$, then all the images of the first $n-2$ states are in the set $\{0, \ldots, n-2\}$. Without input $f$ and state $n-1$ we have the semiautomaton of Theorem 4. By that theorem, $t$ can be done by $\mathcal{C}_n$.

2. If $i_h = n-1$ for some $h$, $1 \leqslant h \leqslant n-2$, then there exists some $j$, $1 \leqslant j \leqslant n-2$ such that $i_k \neq j$ for all $k$, $1 \leqslant k \leqslant n-2$. Define $i'_k$ for all $1 \leqslant k \leqslant n-2$ as follows: $i'_k = j$ if $i_k = n-1$, and $i'_k = i_k$ if $i_k \neq n-1$. Let

$$s = \begin{pmatrix} 0 & 1 & 2 & \cdots & n-3 & n-2 & n-1 \\ 0 & i'_1 & i'_2 & \cdots & i'_{n-3} & i'_{n-2} & n-1 \end{pmatrix},$$

and let $r = (1, j)$. By Theorem 4, $s$ and $r$ can be performed by $\mathcal{C}_n$.

Now consider $srfr$. If $kt = n-1$, then $ks = j$, $jr = 1$, $1f = n-1$, and $(n-1)r = n-1$. If $kt = 1$, then $ks = 1$, $1r = j$, $jf = j$, and $jr = 1$. Finally, if $t$ maps $k$ to a state other than 1 or $n-1$, then $srfr$ maps $k$ to the same state. Hence we have $t = srfr$, and $t$ can be performed by $\mathcal{C}_n$ as well.

Next, refer to states in $\{1, \ldots, n-2\}$ as the *middle* states. Take any transformation $t$ that maps 0 to $h \in \{1, \ldots, n-2\}$, and every middle state to either $\{n-1\}$ or to $h$. There are $(n-2)2^{n-2}$ such transformations. First consider any middle entry $i$ that is mapped to $n-1$ by $t$. We can map $i$ to $n-1$ without changing any other states. First, apply $a^{n-1-i}$ to "rotate all the middle states clockwise", so that $i$ is mapped to 1. Then apply $f$ to map $i$ to $n-1$, and then $a^i$ to return all the states other than $n-1$ to their original positions. This is repeated for all the middle states that are mapped to $n-1$ by $t$. After this is done, apply $e$ to replace all the remaining middle states by 1, and apply $a^{h-1}$ to change 1 to $h$. Hence $t$ can be done.

Finally, the constant transformation $\binom{Q}{n-1}$ is done by $ef$.

In summary, if $L = L(\mathcal{C}_n)$ then $\sigma(L) \geqslant n^{n-2} + (n-2)2^{n-2} + 1$.

If 0 is mapped to a middle state $i$, then the input word must contain an $e$. But every word of the form $xe$ leaves the automaton in a state in $\{1, n-1\}$. Applying any other word can only result in a state in $\{i, n-1\}$, for some middle state $i$. Hence no transformations other than the ones we have considered can be done by $\mathcal{C}_n$, and the syntactic complexity of the language accepted by $\mathcal{C}_n$ is precisely $n^{n-2} + (n-2)2^{n-2} + 1$. □

Table 4 summarizes our results for two-sided ideals.

**Table 4.** Syntactic complexity bounds for two-sided ideals

| | $n=1$ | $n=2$ | $n=3$ | $n=4$ | $n=5$ | | $n=n$ |
|---|---|---|---|---|---|---|---|
| $\|\Sigma\|=1$ | **1** | **1** | **2** | **3** | **4** | ... | $\mathbf{n-1}$ |
| $\|\Sigma\|=2$ | – | **2** | 5 | 11 | 19 | ... | ? |
| $\|\Sigma\|=3$ | – | – | 6 | 16 | 47 | ... | ? |
| $\|\Sigma\|=4$ | – | – | – | 23 | 90 | ... | ? |
| $\|\Sigma\|=5$ | – | – | – | 25 | 147 | ... | ? |
| $\|\Sigma\|=6$ | – | – | – | – | 150 | ... | $n^{n-2}+(n-2)2^{n-2}+1$ |

## 7 Reversal

It is interesting to note that, for our ideals with maximal syntactic complexity, the reverse languages have maximal state complexity. It was shown in [5] that the reverse of a right (left, two-sided) ideal with $n$ quotients has at most $2^{n-1}$ ($2^{n-1}+1$, $2^{n-2}+1$) quotients, and that these bounds can be met.

**Theorem 7.** *If $L(\mathcal{A}_n)$, $L(\mathcal{B}_n)$ and $L(\mathcal{C}_n)$ are the languages in Theorems 3, 4, and 6, then their reverses have $2^{n-1}$, $2^{n-1}+1$, and $2^{n-2}+1$ quotients, respectively.*

## 8 Conclusions

Despite the fact that the Myhill congruence has left-right symmetry, there are significant differences between left and right ideals. The major open problem is the upper bound for left ideals. Also, the relation between syntactic complexity and reversal deserves further study.

**Acknowledgment.** We thank Arto Salomaa for providing some early references.

## References

1. Bozapalidis, S., Kalampakas, A.: Recognizability of graph and pattern languages. Acta Inform. 42(8/9), 553–581 (2006)
2. Bozapalidis, S., Kalampakas, A.: On the complexity of the syntax of tree languages. In: Bozapalidis, S., Rahonis, G. (eds.) CAI 2009. LNCS, vol. 5725, pp. 189–203. Springer, Heidelberg (2009)
3. Bozapalidis, S., Kalampakas, A.: On the syntactic complexity of tree series. RAIRO-Theor. Inf. Appl. 44, 257–279 (2010)
4. Brzozowski, J.: Quotient complexity of regular languages. In: Dassow, J., Pighizzini, G., Truthe, B. (eds.) Proceedings of the 11th International Workshop on Descriptional Complexity of Formal Systems, DCFS, pp. 25–42 (2009), Extended abstract at `http://arxiv.org/abs/0907.4547`
5. Brzozowski, J., Jirásková, G., Li, B.: Quotient complexity of ideal languages. In: López-Ortiz, A. (ed.) LATIN 2010. LNCS, vol. 6034, pp. 208–221. Springer, Heidelberg (2010), Full paper at `http://arxiv.org/abs/0908.2083`

6. Brzozowski, J., Jirásková, G., Zou, C.: Quotient complexity of closed languages. In: Ablayev, F., Mayr, E.W. (eds.) CSR 2010. LNCS, vol. 6072, pp. 84–95. Springer, Heidelberg (2010)
7. Brzozowski, J., Ye, Y.: Syntactic complexity of ideal and closed languages (October 2010), `http://arxiv.org/abs/1010.3263`
8. Dénes, J.: On transformations, transformation semigroups and graphs. In: Erdös, P., Katona, G. (eds.) Theory of Graphs. Proceedings of the Colloquium on Graph Theory held at Tihany, 1966, pp. 65–75. Akadémiai Kiado (1968)
9. Holzer, M., König, B.: On deterministic finite automata and syntactic monoid size. Theoret. Comput. Sci. 327, 319–347 (2004)
10. Hoyer, M.: Verallgemeinerung zweier sätze aus der theorie der substitutionengruppen. Math. Ann. 46(4), 539–544 (1895)
11. Kalampakas, A.: The syntactic complexity of eulerian graphs. In: Bozapalidis, S., Rahonis, G. (eds.) CAI 2007. LNCS, vol. 4728, pp. 208–217. Springer, Heidelberg (2007)
12. Krawetz, B., Lawrence, J., Shallit, J.: State complexity and the monoid of transformations of a finite set (2003), `http://arxiv.org/abs/math/0306416`
13. Krawetz, B., Lawrence, J., Shallit, J.: State complexity and the monoid of transformations of a finite set. Internat. J. Found. Comput. Sci. 16(3), 547–563 (2005)
14. Maslov, A.N.: Estimates of the number of states of finite automata. Dokl. Akad. Nauk SSSR 194, 1266–1268 (1970) (Russian); English translation: Soviet Math. Dokl. 11 , 1373–1375 (1970)
15. Myhill, J.: Finite automata and representation of events. Wright Air Development Center Technical Report 57–624 (1957)
16. Nerode, A.: Linear automaton transformations. Proc. Amer. Math. Soc. 9, 541–544 (1958)
17. Piccard, S.: Sur les fonctions définies dans les ensembles finis quelconques. Fund. Math. 24, 298–301 (1935)
18. Piccard, S.: Sur les bases du groupe symétrique et du groupe alternant. Commentarii Mathematici Helvetici 11(1), 1–8 (1938)
19. Restivo, A., Vaglica, R.: Automata with extremal minimality conditions. In: Gao, Y., Seki, S., Yu, S. (eds.) DLT 2010. LNCS, vol. 6224, pp. 399–410. Springer, Heidelberg (2010)
20. Salomaa, A.: A theorem concerning the composition of functions of several variables ranging over a finite set. J. Symbolic Logic 25, 203–208 (1960)
21. Salomaa, A.: Some completeness criteria for sets of functions over a finite domain. Ann. Univ. Turkuensis, Ser. AI 53 (1962)
22. Salomaa, A.: On basic groups for the set of functions over a finite domain. Ann. Acad. Scient. Fenn., Ser. A 338 (1963)
23. Salomaa, A.: Composition sequences for functions over a finite domain. Theoret. Comput. Sci. 292, 263–281 (2003)
24. Sierpiński, X.: Sur les suites infinies de fonctions définies dans les ensembles quelconques. Fund. Math. 24, 209–212 (1935)
25. Yu, S.: State complexity of regular languages. J. Autom. Lang. Comb. 6, 221–234 (2001)