Complexity of Left-Ideal, Suffix-Closed and Suffix-Free Regular Languages*

Janusz A. Brzozowski and Corwin Sinnamon

David R. Cheriton School of Computer Science, University of Waterloo,
Waterloo, ON, Canada N2L 3G1
brzozo@uwaterloo.ca, sinncore@gmail.com

Abstract. A language L over an alphabet Σ is suffix-convex if, for any words $x, y, z \in \Sigma^*$, whenever z and xyz are in L, then so is yz. Suffix-convex languages include three special cases: left-ideal, suffix-closed, and suffix-free languages. We examine complexity properties of these three special classes of suffix-convex regular languages. In particular, we study the quotient/state complexity of boolean operations, product (concatenation), star, and reversal on these languages, as well as the size of their syntactic semigroups, and the quotient complexity of their atoms.

Keywords: different alphabets, left ideal, most complex, quotient/state complexity, regular language, suffix-closed, suffix-convex, suffix-free, syntactic semigroup, transition semigroup, unrestricted complexity

1 Introduction

Suffix-Convex Languages Convex languages were introduced in 1973 [27], and revisited in 2009 [1]. For $w, x, y \in \Sigma^*$, if $w = xy$, then y is a *suffix* of w. A language L is *suffix-convex* if, whenever z and xyz are in L, then yz is also in L, for all $x, y, z \in \Sigma^*$. Suffix-convex languages include three well-known subclasses: left-ideal, suffix-closed, and suffix-free languages. A language L is a *left ideal* if it is non-empty and satisfies the equation $L = \Sigma^* L$. Left ideals play a role in pattern matching: If one is searching for all words ending with words in some language L in a given text (a word over Σ^*), then one is looking for words in $\Sigma^* L$. Left ideals also constitute a basic concept in semigroup theory. A language L is *suffix-closed* if, whenever w is in L and x is a suffix of w, then x is also in L, for all $w, x \in \Sigma^*$. The complement of every suffix-closed language not equal to Σ^* is a left ideal. A language is *suffix-free* if no word in the language is a suffix of another word in the language. Suffix-free languages (with the exception of $\{\varepsilon\}$, where ε is the empty word) are suffix codes. They have many applications, and have been studied extensively; see [2] for example.

Quotient/State Complexity If Σ is an alphabet and $L \subseteq \Sigma^*$ is a language such that every letter of Σ appears in some word of L, then the *(left) quotient*

* This work was supported by the Natural Sciences and Engineering Research Council of Canada grant No. OGP0000871.

of L by a word $w \in \Sigma^*$ is $w^{-1}L = \{x \mid wx \in L\}$. A language is regular if and only if it has a finite number of distinct quotients. So the number of quotients of L, the *quotient complexity* $\kappa(L)$ [3] of L, is a natural measure of complexity for L. A concept equivalent to quotient complexity is the *state complexity* [28] of L, which is the number of states in a complete minimal deterministic finite automaton (DFA) with alphabet Σ recognizing L. We refer to quotient/state complexity simply as *complexity*.

If L_n is a regular language of complexity n, and $\circ$ is a unary operation, then the *complexity of* $\circ$ is the maximal value of $\kappa(L_n^\circ)$, expressed as a function of n, as L_n ranges over all regular languages of complexity n. Similarly, if L'_m and L_n are regular languages of complexities m and n respectively, $\circ$ is a binary operation, then the *complexity of* $\circ$ is the maximal value of $\kappa(L'_m \circ L_n)$, expressed as a function of m and n, as L'_m and L_n range over all regular languages of complexities m and n. The complexity of an operation is a lower bound on its time and space complexities, and has been studied extensively; see [3, 4, 28].

In the past the complexity of a binary operation was studied under the assumption that the arguments of the operation are *restricted* to be over the same alphabet, but this restriction was removed in [5]. We study both the restricted and unrestricted cases.

Witnesses To find the complexity of a unary operation we find an upper bound on this complexity, and languages that meet this bound. We require a language L_n for each $n \geqslant k$, that is, a sequence $(L_k, L_{k+1}, \dots)$, where k is a small integer, because the bound may not hold for small values of n. Such a sequence is a *stream* of languages. For a binary operation we require two streams. Sometimes the same stream can be used for both operands; in general, however, this is not the case. For example, the bound for union is mn, and it cannot be met by languages from one stream if $m = n$ because $L_n \cup L_n = L_n$ and the complexity is n instead of n^2.

Dialects For all common binary operations on regular languages the second stream can be a "dialect" of the first, that is it can "differ only slightly" from the first, and all the bounds can still be met [4]. Let $\Sigma = \{a_1, \dots, a_k\}$ be an alphabet ordered as shown; if $L \subseteq \Sigma^*$, we denote it by $L(a_1, \dots, a_k)$ to stress its dependence on Σ. A *dialect* of L is obtained by deleting letters of Σ in the words of L, or replacing them by letters of another alphabet Σ'. More precisely, for a partial injective map $\pi\colon \Sigma \mapsto \Sigma'$, we obtain a dialect of L by replacing each letter $a \in \Sigma$ by $\pi(a)$ in every word of L, or deleting the word entirely if $\pi(a)$ is undefined. We write $L(\pi(a_1), \dots, \pi(a_k))$ to denote the dialect of $L(a_1, \dots, a_k)$ given by π, and we denote undefined values of π by "$-$". For example, if $L(a, b, c) = \{a, ab, ac\}$, then $L(b, -, d)$ is the language $\{b, bd\}$. Undefined values at the end of the alphabet are omitted. A similar definition applies to DFAs. Our definition of dialect is more general than that of [7, 11], where only the case $\Sigma' = \Sigma$ was allowed.

Most Complex Streams It was proved that there exists a stream $(L_3, L_4, \dots)$ of regular languages which together with some dialects meets all the complexity bounds for reversal, (Kleene) star, product (concatenation), and all binary boolean operations [4, 5]. Moreover, this stream meets two additional complexity

bounds: the size of the syntactic semigroup, and the complexities of atoms (discussed later). A stream of deterministic finite automata (DFAs) corresponding to a most complex language stream is a *most complex DFA stream*. In defining a most complex stream we try to minimize the size of the union of the alphabets of the dialects required to meet all the bounds.

Most complex streams are useful in the designs of systems dealing with regular languages and finite automata. To know the maximal sizes of automata that can be handled by the system it suffices to use the most complex stream to test all the operations.

It is known that there is a most complex stream of left ideals that meets all the bounds in both the restricted [7, 11] and unrestricted [11] cases, but a most complex suffix-free stream does not exist [14].

Our Contributions

1. We derive a new left-ideal stream from the most complex left-ideal stream and show that it meets all the complexity bounds except that for product.
2. We prove that the complement of the new left-ideal stream is a most complex suffix-closed stream.
3. We find a new suffix-free stream that meets the bounds for star, product and boolean operations; it has simpler transformations than the known stream.
4. Our witnesses for left-ideal, suffix-closed, and suffix-free streams are all derived from one most complex regular stream.

2 Background

Finite Automata 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*, $\delta\colon Q \times \Sigma \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial* state, and $F \subseteq Q$ is the set of *final* states. We extend δ to a function $\delta\colon Q \times \Sigma^* \to Q$ as usual. A DFA $\mathcal{D}$ *accepts* a word $w \in \Sigma^*$ if $\delta(q_0, w) \in F$. The language accepted by $\mathcal{D}$ is denoted by $L(\mathcal{D})$. If q is a state of $\mathcal{D}$, then the language L^q of q is the language accepted by the DFA $(Q, \Sigma, \delta, q, F)$. A state is *empty* if its language is empty. Two states p and q of $\mathcal{D}$ are *equivalent* if $L^p = L^q$. A state q is *reachable* if there exists $w \in \Sigma^*$ such that $\delta(q_0, w) = q$. A DFA is *minimal* if all of its states are reachable and no two states are equivalent. Usually DFAs are used to establish upper bounds on the complexity of operations and also as witnesses that meet these bounds.

A *nondeterministic finite automaton (NFA)* is a quintuple $\mathcal{D} = (Q, \Sigma, \delta, I, F)$ where Q, Σ and F are defined as in a DFA, $\delta\colon Q \times \Sigma \to 2^Q$ is the *transition function*, and $I \subseteq Q$ is the *set of initial states*. An *ε-NFA* is an NFA in which transitions under the empty word ε are also permitted.

Transformations We use $Q_n = \{0, \ldots, n-1\}$ as our basic set with n elements. A *transformation* of Q_n is a mapping $t\colon Q_n \to Q_n$. The *image* of $q \in Q_n$ under t is denoted by qt. If s and t are transformations of Q_n, their composition is denoted $(qs)t$ when applied to $q \in Q_n$. Let $\mathcal{T}_{Q_n}$ be the set of all n^n transformations of Q_n; then $\mathcal{T}_{Q_n}$ is a monoid under composition.

For $k \geqslant 2$, a transformation (permutation) t of a set $P = \{q_0, q_1, \ldots, q_{k-1}\} \subseteq Q$ is a *k-cycle* if $q_0t = q_1, q_1t = q_2, \ldots, q_{k-2}t = q_{k-1}, q_{k-1}t = q_0$. This k-cycle is denoted by $(q_0, q_1, \ldots, q_{k-1})$. A 2-cycle (q_0, q_1) is called a *transposition*. A transformation that sends all the states of P to q and acts as the identity on the remaining states is denoted by $(P \to q)$ the transformation $(Q_n \to p)$ is called *constant*. If $P = \{p\}$ we write $(p \to q)$ for $(\{p\} \to q)$. The identity transformation is denoted by $\mathbb{1}$. The notation $\binom{j}{i}\ q \to q+1)$ denotes a transformation that sends q to $q+1$ for $i \leqslant q \leqslant j$ and is the identity for the remaining states. the notation $\binom{j}{i}\ q \to q-1)$ is defined similarly.

Semigroups The *Myhill congruence* $\approx_L$ [25] (also known as the *syntactic congruence*) of a language $L \subseteq \Sigma^*$ is defined on Σ^+ as follows: For $x, y \in \Sigma^+, x \approx_L y$ if and only if $wxz \in L \Leftrightarrow wyz \in L$ for all $w, z \in \Sigma^*$. The quotient set $\Sigma^+/\approx_L$ of equivalence classes of $\approx_L$ is a semigroup, the *syntactic semigroup* T_L of L.

Let $\mathcal{D} = (Q_n, \Sigma, \delta, 0, F)$ be a DFA. For each word $w \in \Sigma^*$, the transition function induces a transformation δ_w of Q_n by w: for all $q \in Q_n$, $q\delta_w = \delta(q, w)$. The set $T_{\mathcal{D}}$ of all such transformations by non-empty words is the *transition semigroup* of $\mathcal{D}$ under composition [26]. Sometimes we use the word w to denote the transformation it induces; thus we write qw instead of $q\delta_w$. We extend the notation to sets: if $P \subseteq Q_n$, then $Pw = \{pw \mid p \in P\}$. We also find write $P \xrightarrow{w} Pw$ to indicate that the image of P under w is Pw.

If $\mathcal{D}$ is a minimal DFA of L, then $T_{\mathcal{D}}$ is isomorphic to the syntactic semigroup T_L of L [26], and we represent elements of T_L by transformations in $T_{\mathcal{D}}$. The size of this semigroup has been used as a measure of complexity [4, 18, 21, 24].

Atoms Atoms are defined by a left congruence, where two words x and y are equivalent if $ux \in L$ if and only if $uy \in L$ for all $u \in \Sigma^*$. Thus x and y are equivalent if $x \in u^{-1}L$ if and only if $y \in u^{-1}L$. An equivalence class of this relation is an *atom* of L [17]. Thus an atom is a non-empty intersection of complemented and uncomplemented quotients of L. The number of atoms and their complexities were suggested as possible measures of complexity of regular languages [4], because all the quotients of a language, and also the quotients of atoms, are always unions of atoms [16, 17, 22].

Our Key Witness The stream $(\mathcal{D}_n(a, b, c) \mid n \geqslant 3)$ of Definition 1 and Figure 1 was introduced in [4]. It will be used as a component in all the classes of languages examined in this paper. It was shown in [4] that this stream together with some dialects is most complex for restricted operations, and also for unrestricted operations if an input d that induces the identity transformation is added [5, 11].

Definition 1. *For $n \geqslant 3$, let $\mathcal{D}_n = \mathcal{D}_n(a, b, c) = (Q_n, \Sigma, \delta_n, 0, \{n-1\})$, where $\Sigma = \{a, b, c\}$, and δ_n is defined by a: $(0, \ldots, n-1)$, b: $(0, 1)$, c: $(n-1 \to 0)$.*

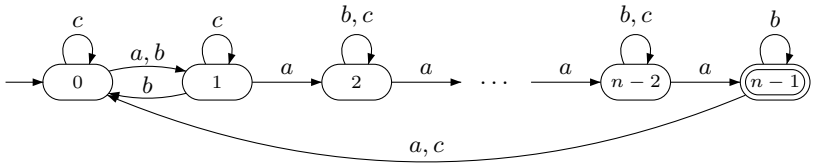

Fig. 1. Minimal DFA of a most complex regular language.

3 Left Ideals

The following stream was studied in [18] and also in [6, 7, 13]. This stream is most complex when the two alphabets are the same in binary operations [7]. It is also most complex for unrestricted operations [11].

Definition 2. *For $n \geqslant 4$, let $\mathcal{D}_n = \mathcal{D}_n(a,b,c,d,e) = (Q_n, \Sigma, \delta_n, 0, \{n-1\})$, where $\Sigma = \{a,b,c,d,e\}$, and δ_n is defined by transformations $a\colon (1,\ldots,n-1)$, $b\colon (1,2)$, $c\colon (n-1 \to 1)$, $d\colon (n-1 \to 0)$, and $e\colon (Q_n \to 1)$. See Figure 2. Let $L_n = L_n(a,b,c,d,e)$ be the language accepted by $\mathcal{D}_n$.*

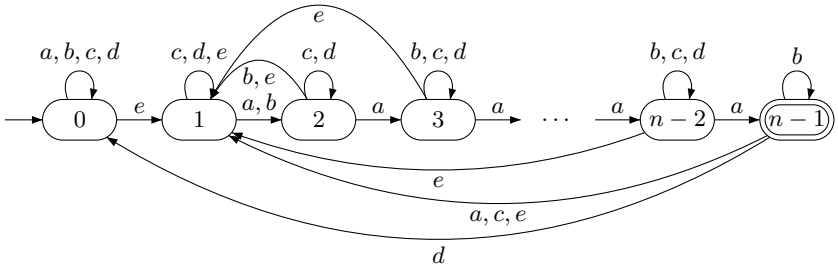

Fig. 2. Minimal DFA of a most complex left ideal $L_n(a,b,c,d,e)$.

Theorem 3 (Most Complex Left Ideals [7, 11]). *For each $n \geqslant 4$, the DFA of Definition 2 is minimal. The stream $(L_n(a,b,c,d,e) \mid n \geqslant 4)$ with some dialect streams is most complex in the class of regular left ideals.*

1. *The syntactic semigroup of $L_n(a,b,c,d,e)$ has cardinality $n^{n-1} + n - 1$.*
2. *Each quotient of $L_n(a,-,-,d,e)$ has complexity n.*
3. *The reverse of $L_n(a,-,c,d,e)$ has complexity $2^{n-1}+1$, and $L_n(a,-,c,d,e)$ has $2^{n-1}+1$ atoms.*
4. *For each atom A_S of $L_n(a,b,c,d,e)$, the complexity $\kappa(A_S)$ satisfies:*

$$\kappa(A_S) = \begin{cases} n, & \text{if } S = Q_n; \\ 2^{n-1}, & \text{if } S = \emptyset; \\ 1 + \sum_{x=1}^{|S|} \sum_{y=1}^{n-|S|} \binom{n-1}{x}\binom{n-x-1}{y-1}, & \text{otherwise.} \end{cases}$$

5. *The star of* $L_n(a,-,-,-,e)$ *has complexity* $n+1$.
6. (a) *Restricted product:* $\kappa(L'_m(a,-,-,-,e)L_n(a,-,-,-,e)) = m+n-1$.
   (b) *Unrestricted product:* $\kappa(L'_m(a,b,-,d,e)L_n(a,d,-,c,e)) = mn+m+n$.
7. (a) *Restricted complexity:* $\kappa(L'_m(a,-,c,-,e) \circ L_n(a,-,e,-,c)) = mn$.
   (b) *Unrestricted complexity:* $\kappa(L'_m(a,b,-,d,e) \circ L_n(a,d,-,c,e)) = (m+1)(n+1)$ *if* $\circ \in \{\cup, \oplus\}$), $mn+m$ *if* $\circ = \setminus$, *and* mn *if* $\circ = \cap$.
   In both cases the bounds for boolean operations are the same as those for regular languages.

We now define a new left-ideal witness similar to the witness in Definition 2.

Definition 4. *For* $n \geqslant 4$, *let* $\mathcal{E}_n = \mathcal{E}_n(a,b,c,d,e) = (Q_n, \Sigma, \delta_n, 0, \{1,\ldots,n-1\})$, *where* Σ *and the transformations induced by its letters are as in* $\mathcal{D}_n$ *of Definition 2. Let* $M_n = M_n(a,b,c,d,e)$ *be the language accepted by* $\mathcal{E}_n$.

Theorem 5 (Nearly Most Complex Left Ideals). *For each* $n \geqslant 4$, *the DFA of Definition 4 is minimal and its language* $M_n(a,b,c,d,e)$ *is a left ideal of complexity* n. *The stream* $(M_n(a,b,c,d,e) \mid n \geqslant 4)$ *with some dialect streams meets all the complexity bounds for left ideals, except those for product.*

Proof. It is easily verified that $\mathcal{E}_n(a,-,-,d,e)$ is minimal; hence $M_n(a,b,c,d,e)$ has complexity n. M_n is a left ideal because, for each letter ℓ of Σ, and each word $w \in \Sigma^*$, $w \in M_n$ implies $\ell w \in M_n$. We prove all the claims of Theorem 3 except the claims in Item 6.

1. **Semigroup** The transition semigroup is independent of the set of final states; hence it has the size of the DFA of the most complex left ideal.
2. **Quotients** Obvious.
3. **Reversal** The upper bound of $2^{n-1}+1$ was proved in [8], and it was shown in [17] that the number of atoms is the same as the complexity of the reverse. Applying the standard NFA construction for reversal, we reverse every transition in DFA $\mathcal{E}_n$ and interchange the final and initial states, yielding the NFA in Figure 3, where the initial states (unmarked) are $Q_n \setminus \{0\}$.
   We perform the subset construction. Set $Q_n \setminus \{0\}$ is initial. From $\{q_1,\ldots,q_k\}$, $1 \leqslant q_1 \leqslant q_k$, we delete q_i, $q_1 \leqslant q_i \leqslant q_k \leqslant n-1$, by applying $a^{q_i}da^{n-1-q_i}$. Thus all 2^{n-1} subsets of $Q_n \setminus \{0\}$ can be reached, and Q_n is reached from the initial state $\{1\}$ by e. For any distinct $S, T \subseteq Q_n$ with $q \in S \setminus T$, either $q = 0$, in which case S is final and T is non-final, or $Sa^{q-1}e = Q_n$ and $Ta^{q-1}e = \emptyset$. Hence all $2^{n-1}+1$ states are pairwise distinguishable.
4. **Atoms** The upper bounds in Theorem 3 for left ideals were derived in [6]. The proof of [6] that these bounds are met applies also to our witness M_n.
5. **Star** The upper bound $n+1$ was proved in [8]. To construct an NFA recognizing $(M_n(a,-,-,d,e))^*$ we add a new initial state $0'$ which is also final and has the same transitions as the former initial state 0. We then add an ε-transition from each final state of $\mathcal{E}(a,-,-,d,e)$ to the initial state $0'$. The language recognized by the new NFA $\mathcal{N}$ is $(M_n(a,-,-,d,e))^*$. The final state $\{0'\}$ in the subset construction for $\mathcal{N}$ is distinguishable from every other final state, since it rejects a, whereas other final states accept it.

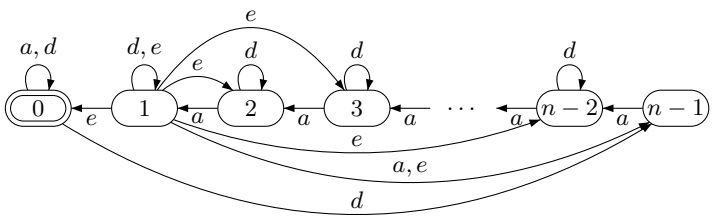

Fig. 3. NFA for reversal of $M_n(a,-,-,d,e)$.

6. **Product** Not applicable.
7. **Boolean Operations** *Proof sketch; omitted proofs can be found in [12].*
   (a) Restricted complexity: The upper bound of mn is the same as for regular languages. We show that $M'_m(a,b,-,d,e) \circ M_n(a,e,-,d,b)$ has complexity mn. Using the standard construction for boolean operations, we consider the direct product of $\mathcal{E}'_m(a,b,-,d,e)$ and $\mathcal{E}_n(a,e,-,d,b)$. The set of final states of the direct product varies depending on $\circ \in \{\cup, \oplus, \setminus, \cap\}$. It is easy to see that all mn states are reachable. We then verify for each operation that every two states in the direct product are distinguishable.
   (b) Unrestricted complexity: To produce a DFA recognizing $M'_m(a,b,c,d,e) \circ M_n(a,e,f,d,b)$, we add an empty state $\emptyset'$ to $\mathcal{E}'_m(a,b,c,d,e)$, and send all the transitions from any state of Q'_m under f to $\emptyset'$. Similarly add an empty state $\emptyset$ to $\mathcal{E}_n(a,e,f,d,b)$ and send all the transitions from any state of Q_n under c to $\emptyset$. Now the DFAs are over $\{a,b,c,d,e,f\}$ and we take their direct product. By the restricted case all the states of $Q'_m \times Q_n$ are reachable and distinguishable using words in $\{a,b,d,e\}^*$. Let $R_{\emptyset'} = \{(\emptyset', q) \mid q \in Q_n\}$ and $C_\emptyset = \{(p', \emptyset) \mid p' \in Q'_m\}$. States of $R_{\emptyset'} \cup C_\emptyset \cup \{(\emptyset', \emptyset)\}$ are easily seen to be reachable.
   Union Here all $(m+1)(n+1)$ states turn out to pairwise distinguishable.
   Symmetric Difference Same as union.
   Difference States of $R_{\emptyset'} \cup \{(\emptyset', \emptyset)\}$ are empty and therefore equivalent. However, since the alphabet of $M'_m(a,b,c,d,e) \setminus M_n(a,e,f,d,b)$ is $\{a,b,c,d,e\}$ we omit f, delete the states of $R_{\emptyset'} \cup \{(\emptyset', \emptyset)\}$, and have a DFA over $\{a,b,c,d,e\}$ accepting $M'_m(a,b,c,d,e) \setminus M_n(a,e,f,d,b)$. The $mn+m$ remaining states are pairwise distinguishable.
   Intersection States of $R_{\emptyset'} \cup C_\emptyset \cup \{(\emptyset', \emptyset)\}$ are empty and therefore equivalent. Since the alphabet of $M'_m(a,b,c,d,e) \cap M_n(a,e,f,d,b)$ is $\{a,b,d,e\}$, we omit c and f, delete the states of $R_{\emptyset'} \cup C_\emptyset \cup \{(\emptyset', \emptyset)\}$, and have a DFA over $\{a,b,d,e\}$ accepting $M'_m(a,b,c,d,e) \cap M_n(a,e,f,d,b)$. By the restricted case, all mn states are pairwise distinguishable. □

4 Suffix-Closed Languages

The complexity of suffix-closed languages was studied in [9] in the restricted case, and the syntactic semigroup of these languages, in [13, 15, 18]; however, most complex suffix-closed streams have not been examined.

Definition 6. *For $n \geqslant 4$, let $\mathcal{D}_n = \mathcal{D}_n(a,b,c,d,e) = (Q_n, \Sigma, \delta_n, 0, \{0\})$, where $\Sigma = \{a,b,c,d,e\}$, and δ_n is defined by transformations $a\colon (1,\ldots,n-1)$, $b\colon (1,2)$, $c\colon (n-1 \to 1)$, $d\colon (n-1 \to 0)$, $e\colon (Q_n \to 1)$. Let $L_n = L_n(a,b,c,d,e)$ be the language accepted by $\mathcal{D}_n$; this language is the complement of the left ideal of Definition 4. The structure of $\mathcal{D}_n(a,b,c,d,e)$ is shown in Figure 4.*

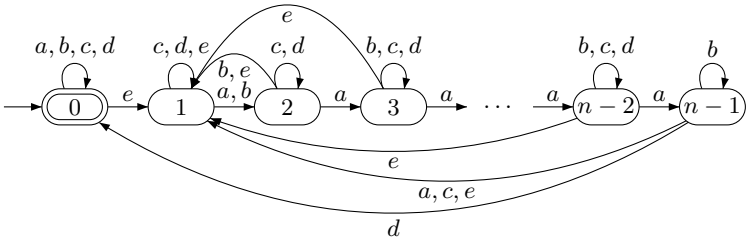

Fig. 4. Minimal DFA $\mathcal{D}_n(a,b,c,d,e)$ of Definition 6.

Theorem 7 (Most Complex Suffix-Closed Languages). *For each $n \geqslant 4$, the DFA of Definition 6 is minimal and its language $L_n(a,b,c,d,e)$ is suffix-closed and has complexity n. The stream $(L_n(a,b,c,d,e) \mid n \geqslant 4)$ with some dialect streams is most complex in the class of suffix-closed languages.*

1. *The syntactic semigroup of $L_n(a,b,c,d,e)$ has cardinality $n^{n-1}+n-1$. Moreover, fewer than five inputs do not suffice to meet this bound.*
2. *All quotients of $L_n(a,-,-,d,e)$ have complexity n.*
3. *The reverse of $L_n(a,-,-,d,e)$ has complexity $2^{n-1}+1$, and $L_n(a,-,-,d,e)$ has $2^{n-1}+1$ atoms.*
4. *For each atom A_S of $L_n(a,b,c,d,e)$, the complexity $\kappa(A_S)$ satisfies:*

$$\kappa(A_S) = \begin{cases} n, & \text{if } S = \emptyset; \\ 2^{n-1}, & \text{if } S = Q_n; \\ 1 + \sum_{x=1}^{|S|} \sum_{y=1}^{n-|S|} \binom{n-1}{y}\binom{n-y-1}{x-1}, & \text{if } \{0\} \subseteq S \subsetneq Q_n. \end{cases}$$

5. *The star of $L_n(a,-,-,d,e)$ has complexity n.*
6. *(a) Restricted complexity: $\kappa(L'_m(a,b,-,d,e)L_n(a,e,-,d,b)) = mn - n + 1$.*
   (b) Unrestricted complexity: $\kappa(L'_m(a,b,c,d,e)L_n(a,e,f,d,b)) = mn+m+1$.
7. *(a) Restricted complexity: $\kappa(L'_m(a,b,-,d,e) \circ L_n(a,e,-,d,b)) = mn$ for $\circ \in \{\cup, \oplus, \cap, \setminus\}$.*
   (b) Unrestricted complexity: $\kappa(L'_m(a,b,c,d,e) \circ L_n(a,e,f,d,b)) = (m+1)(n+1)$ if $\circ \in \{\cup, \oplus\}$, it is $mn+m$ if $\circ = \setminus$, and mn if $\circ = \cap$.

The above results for syntactic complexity, reversal, complexity of atoms, and restricted boolean operations follow easily from Theorem 5.

5 Suffix-Free Languages

The complexity of suffix-free languages was studied in detail in [10, 14, 19, 20, 23]. For completeness we present a short summary of some of those results. The main result of [14] is a proof that *a most complex suffix-free language does not exist.* Since every suffix-free language has an empty quotient, the restricted and unrestricted cases for binary operations coincide.

For $n \geqslant 6$, the transition semigroup of the DFA defined below is the largest transition semigroup of a minimal DFA accepting a suffix-free language.

Definition 8. *For $n \geqslant 4$, we define the DFA $\mathcal{D}_n(a,b,c,d,e) = (Q_n, \Sigma, \delta, 0, F)$, where $Q_n = \{0,\ldots,n-1\}$, $\Sigma = \{a,b,c,d,e\}$, δ is defined by the transformations $a\colon (0 \to n-1)(1,\ldots,n-2)$, $b\colon (0 \to n-1)(1,2)$, $c\colon (0 \to n-1)(n-2 \to 1)$, $d\colon (\{0,1\} \to n-1)$, $e\colon (Q \setminus \{0\} \to n-1)(0 \to 1)$, and $F = \{q \in Q_n \setminus \{0, n-1\} \mid q \text{ is odd}\}$. For $n = 4$, a and b coincide, and we can use $\Sigma = \{b,c,d,e\}$. Let the transition semigroup of $\mathcal{D}_n$ be $\mathbf{T}^{\geqslant 6}(n)$.*

The main result for this witness is the following theorem:

Theorem 9 (Semigroup, Quotients, Reversal, Atoms, Boolean Ops.). *Consider DFA $\mathcal{D}_n(a,b,c,d,e)$ of Definition 8; its language $L_n(a,b,c,d,e)$ is a suffix-free language of complexity n. Moreover, it meets the following bounds:*

1. *For $n \geqslant 6$, $L_n(a,b,c,d,e)$ meets the bound $(n-1)^{n-2}+n-2$ for syntactic complexity, and at least five letters are required to reach this bound.*
2. *The quotients of $L_n(a,-,-,-,e)$ have complexity $n-1$, except for L which has complexity n, and the empty quotient which has complexity 1.*
3. *For $n \geqslant 4$, the reverse of $L_n(a,-,c,-,e)$ has complexity $2^{n-2}+1$, and $L_n(a,-,c,-,e)$ has $2^{n-2}+1$ atoms.*
4. *Each atom A_S of $L_n(a,b,c,d,e)$ has maximal complexity:*

$$\kappa(A_S) = \begin{cases} 2^{n-2}+1, & \text{if } S = \emptyset; \\ n, & \text{if } S = \{0\}; \\ 1 + \sum_{x=1}^{|S|} \sum_{y=0}^{n-2-|S|} \binom{n-2}{x}\binom{n-2-x}{y}, & \emptyset \neq S \subseteq \{1,\ldots,n-2\}. \end{cases}$$

5. *For $n,m \geqslant 4$, the complexity of $L_m(a,b,-,d,e) \circ L_n(b,a,-,d,e)$ is $mn-(m+n-2)$ if $\circ \in \{\cup, \oplus\}$, $mn-(m+2n-4)$ if $\circ = \setminus$, and $mn-2(m+n-3)$ if $\circ = \cap$.*
6. *A language which has a subsemigroup of $\mathbf{T}^{\geqslant 6}(n)$ as its syntactic semigroup cannot meet the bounds for star and product.*

The DFA defined below has the largest transition semigroup when $n \in \{4,5\}$. The transition semigroup of this DFA is $\mathbf{T}^{\leqslant 5}(n)$, and at least n letters are required to generate it.

Definition 10. *For $n \geqslant 4$, $\mathcal{D}_n(a,b,c_1,\ldots,c_{n-2}) = (Q_n, \Sigma_n, \delta, 0, \{n-2\})$, where $Q_n = \{0,\ldots,n-1\}$, $\Sigma_n = \{a,b,c_1,\ldots,c_{n-2}\}$, δ is given by $a\colon (0 \to n-1)(1,\ldots,n-2)$, $b\colon (0 \to n-1)(1,2)$, and $c_p\colon (p \to n-1)(0 \to p)$ for $1 \leqslant p \leqslant n-2$.*

We now define a DFA based on Definition 10, but with only three inputs.

Definition 11. *For $n \geqslant 4$, define the DFA $\mathcal{D}_n = (Q_n, \Sigma, \delta, 0, \{n-2\})$, where $Q_n = \{0, \ldots, n-1\}$, $\Sigma = \{a, b, c\}$, and δ is defined by $a\colon (0 \to n-1)(1, \ldots, n-2)$, $b\colon (0 \to n-1)(1, 2)$, $c\colon (1, n-1)(0 \to 1)$. See Figure 5.*

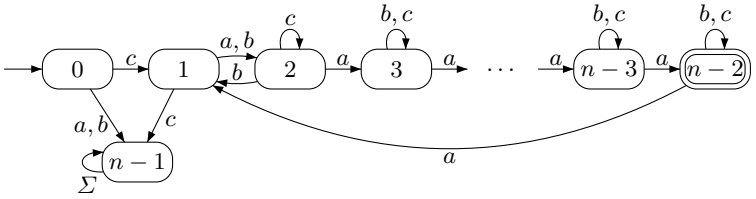

Fig. 5. Witness for star, product, and boolean operations.

Theorem 12 (Star, Product, Boolean Operations). *Let $\mathcal{D}_n(a, b, c)$ be the DFA of Definition 11, and let the language it accepts be $L_n(a, b, c)$. Then L_n and its permutational dialects meet the bounds for star, product, and boolean operations as follows:*

1. *For $n \geqslant 4$, $(L_n(a, b, c))^*$ meets the bound $2^{n-2} + 1$.*
2. *For $m, n \geqslant 4$, $L'_m(a, b, c)L_n(c, a, b)$ meets the bound $(m-1)2^{n-2} + 1$.*
3. *For $m, n \geqslant 4$, but $(m, n) \neq (4, 4)$, the complexity of $L'_m(a, b, c) \circ L_n(b, a, c)$ is $mn - (m + n - 2)$ if $\circ \in \{\cup, \oplus\}$, $mn - (m + 2n - 4)$ if $\circ = \setminus$, and $mn - 2(m + n - 3)$ if $\circ = \cap$.*

The transition semigroup of the DFA of Definition 13 below is a also a subsemigroup of $\mathbf{T}^{\leqslant 5}(n)$, and its language also meets the bounds for product, star and boolean operations. The advantage of this DFA is that its witnesses use only two letters for star and only two letters (but three transformations) for boolean operations. Its disadvantages are the rather complex transformations. For more details see [14]. The DFA of Definition 11 seems to us more natural.

Definition 13. *For $n \geqslant 6$, we define the DFA $\mathcal{D}_n = (Q_n, \Sigma, \delta, 0, \{1\})$, where $Q_n = \{0, \ldots, n-1\}$, $\Sigma = \{a, b, c\}$, and δ is defined by the transformations $a\colon (0 \to n-1)(1, 2, 3)(4, \ldots, n-2)$, $b\colon (2 \to n-1)(1 \to 2)(0 \to 1)(3, 4)$, $c\colon (0 \to n-1)(1, \ldots, n-2)$.*

6 Conclusions

We have examined the complexity properties of left-ideal, suffix-closed, and suffix-free languages together because they are all special cases of suffix-convex languages. We have used the same most complex regular language as a basic component in all three cases.

Our results are summarized in Table 1. The largest bounds are shown in boldface type. Recall that for regular languages we have the following results: semigroup: n^n; reverse: 2^n; star: $2^{n-1}+2^{n-2}$; product, restricted: $(m-1)2^n+2^{n-1}$; unrestricted: $m2^n+2^{n-1}$; $\cup$ and $\oplus$, restricted: mn; unrestricted: $(m+1)(n+1)$; $\setminus$, restricted: mn; unrestricted: $mn+m$; $\cap$, restricted: mn; unrestricted: mn.

Table 1. Complexities of special suffix-convex languages

	Left-Ideal	Suffix-Closed	Suffix-Free
Semigroup	$\mathbf{n^{n-1}+n-1}$	$\mathbf{n^{n-1}+n-1}$	$(n-1)^{n-2}+n-2$
Reverse	$\mathbf{2^{n-1}+1}$	$\mathbf{2^{n-1}+1}$	$2^{n-2}+1$
Star	$n+1$	n	$\mathbf{2^{n-2}+1}$
Product restricted	$m+n-1$	$mn-n+1$	$\mathbf{(m-1)2^{n-2}+1}$
Product unrestricted	$mn+m+n$	$mn+m+1$	$\mathbf{(m-1)2^{n-2}+1}$
$\cup$ *restricted*	$\mathbf{mn}$	$\mathbf{mn}$	$mn-(m+n-2)$
$\cup$ *unrestricted*	$\mathbf{(m+1)(n+1)}$	$\mathbf{(m+1)(n+1)}$	$mn-(m+n-2)$
$\oplus$ *restricted*	$\mathbf{mn}$	$\mathbf{mn}$	$mn-(m+n-2)$
$\oplus$ *unrestricted*	$\mathbf{(m+1)(n+1)}$	$\mathbf{(m+1)(n+1)}$	$mn-(m+n-2)$
$\setminus$ *restricted*	$\mathbf{mn}$	$\mathbf{mn}$	$mn-(m+2n-4)$
$\setminus$ *unrestricted*	$\mathbf{mn+m}$	$\mathbf{mn+m}$	$mn-(m+2n-4)$
$\cap$ *restr.* and *unrestr.*	$\mathbf{mn}$	$\mathbf{mn}$	$mn-2(m+n-3)$

The complexities for left ideals are the same as those for suffix-closed languages, except for star and product, and the complexities of boolean operations for these two classes are the same as those for arbitrary regular languages. The complexities of suffix-free languages are smaller than those of left ideals and suffix-closed languages, except for star and product.

References

1. Ang, T., Brzozowski, J.A.: Languages convex with respect to binary relations, and their closure properties. Acta Cybernet. 19(2), 445–464 (2009)
2. Berstel, J., Perrin, D., Reutenauer, C.: Codes and Automata (Encyclopedia of Mathematics and its Applications). Cambridge University Press (2010)
3. Brzozowski, J.A.: Quotient complexity of regular languages. J. Autom. Lang. Comb. 15(1/2), 71–89 (2010)
4. Brzozowski, J.A.: In search of the most complex regular languages. Int. J. Found. Comput. Sci., 24(6), 691–708 (2013)
5. Brzozowski, J.A.: Unrestricted state complexity of binary operations on regular languages. In: Câmpeanu, C., et. al (eds.) DCFS 2016. LNCS, vol. 9777, pp. 60–72. Springer (2016)

6. Brzozowski, J.A., Davies, S.: Quotient complexities of atoms in regular ideal languages. Acta Cybernet. 22(2), 293–311 (2015)
7. Brzozowski, J.A., Davies, S., Liu, B.Y.V.: Most complex regular ideal languages. Discrete Math. Theoret. Comput. Sci. 18(3) (2016), paper #15
8. Brzozowski, J.A., Jirásková, G., Li, B.: Quotient complexity of ideal languages. Theoret. Comput. Sci. 470, 36–52 (2013)
9. Brzozowski, J.A., Jirásková, G., Zou, C.: Quotient complexity of closed languages. Theory Comput. Syst. 54, 277–292 (2014)
10. Brzozowski, J.A., Li, B., Ye, Y.: Syntactic complexity of prefix-, suffix-, bifix-, and factor-free regular languages. Theoret. Comput. Sci. 449, 37–53 (2012)
11. Brzozowski, J.A., Sinnamon, C.: Unrestricted state complexity of binary operations on regular and ideal languages, to appear. See also `http://arxiv.org/abs/1609.04439`
12. Brzozowski, J.A., Sinnamon, C.: Complexity of left-ideal, suffix-closed, and suffix-free regular languages (2016), `http://arxiv.org/abs/1610.00728`
13. Brzozowski, J.A., Szykuła, M.: Upper bounds on syntactic complexity of left and two-sided ideals. In: Shur, A.M., Volkov, M.V. (eds.) DLT 2014. LNCS, vol. 8633, pp. 13–24. Springer (2014)
14. Brzozowski, J.A., Szykuła, M.: Complexity of suffix-free regular languages. In: Kosowski, A., Walukiewicz, I. (eds.) FCT 2015. LNCS, vol. 9210, pp. 146–159. Springer (2015), full paper at `http://arxiv.org/abs/1504.05159`
15. Brzozowski, J.A., Szykuła, M., Ye, Y.: Syntactic complexity of regular ideals (September 2015), `http://arxiv.org/abs/1509.06032`
16. Brzozowski, J.A., Tamm, H.: Quotient complexities of atoms of regular languages. Int. J. Found. Comput. Sci. 24(7), 1009–1027 (2013)
17. Brzozowski, J.A., Tamm, H.: Theory of átomata. Theoret. Comput. Sci. 539, 13–27 (2014)
18. Brzozowski, J.A., Ye, Y.: Syntactic complexity of ideal and closed languages. In: Mauri, G., Leporati, A. (eds.) DLT 2011. LNCS, vol. 6795, pp. 117–128. Springer Berlin / Heidelberg (2011)
19. Cmorik, R., Jirásková, G.: Basic operations on binary suffix-free languages. In: Kotásek, Z., et al. (eds.) MEMICS. pp. 94–102 (2012)
20. Han, Y.S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410(27-29), 2537–2548 (2009)
21. Holzer, M., König, B.: On deterministic finite automata and syntactic monoid size. Theoret. Comput. Sci. 327(3), 319–347 (2004)
22. Iván, S.: Complexity of atoms, combinatorially. Inform. Process. Lett. 116(5), 356–360 (2016)
23. Jirásková, G., Olejár, P.: State complexity of union and intersection of binary suffix-free languages. In: Bordihn, H., et al. (eds.) NCMA. pp. 151–166. Austrian Computer Society (2009)
24. Krawetz, B., Lawrence, J., Shallit, J.: State complexity and the monoid of transformations of a finite set. In: Domaratzki, M., Okhotin, A., Salomaa, K., Yu, S. (eds.) CIAA 2005. LNCS, vol. 3317, pp. 213–224. Springer Berlin / Heidelberg (2005)
25. Myhill, J.: Finite automata and representation of events. Wright Air Development Center Technical Report 57–624 (1957)
26. Pin, J.E.: Syntactic semigroups. In: Handbook of Formal Languages, vol. 1: Word, Language, Grammar, pp. 679–746. Springer, New York, NY, USA (1997)
27. Thierrin, G.: Convex languages. In: Nivat, M. (ed.) Automata, Languages and Programming, pp. 481–492. North-Holland (1973)

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