Information and Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Syntactic complexity of suffix-free languages

Janusz A. Brzozowski [a], Marek Szykuła [b,*]

[a] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada
[b] Institute of Computer Science, University of Wrocław, Joliot-Curie 15, PL-50-383 Wrocław, Poland

ARTICLE INFO

Article history:
Received 17 October 2015
Available online xxxx

Keywords:
Regular language
Suffix-free
Syntactic complexity
Transition semigroup
Upper bound

ABSTRACT

We solve an open problem concerning syntactic complexity: We prove that the cardinality of the syntactic semigroup of a suffix-free language with n left quotients (that is, with state complexity n) is at most $(n-1)^{n-2}+n-2$ for $n \geqslant 6$. Since this bound is known to be reachable, this settles the problem. We also reduce the alphabet of the witness languages reaching this bound to five letters instead of $n+2$, and show that it cannot be any smaller. Finally, we prove that the transition semigroup of a minimal deterministic automaton accepting a witness language is unique for each n.

1. Introduction

The *syntactic complexity* [8] of a regular language L is the size of its syntactic semigroup [14]. This semigroup is isomorphic to the transition semigroup of the quotient automaton $\mathcal{D}$, a minimal deterministic finite automaton (DFA) accepting the language. The descriptional complexity of syntactic monoids as a function of minimal DFA size for regular languages was first considered systematically in [11,13].

The number n of states of $\mathcal{D}$ is the *state complexity* of the language [16], and it is the same as the *quotient complexity* [3] (number of left quotients) of the language. The *syntactic complexity of a class* of regular languages is the maximal syntactic complexity of languages in that class expressed as a function of the quotient complexity n.

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. Prefixes and suffixes of w are also factors of w. A language L is *prefix-free* (respectively, *suffix-free, factor-free*) if $w, u \in L$ and u is a prefix (respectively, *suffix, factor*) of w, then $u = w$. A language is *bifix-free* if it is both prefix- and suffix-free. These languages play an important role in coding theory, have applications in such areas as cryptography, data compression, and information transmission, and have been studied extensively; see [2] for example. In particular, suffix-free languages (with the exception of $\{\varepsilon\}$, where ε is the empty word) are suffix codes. Moreover, suffix-free languages are special cases of suffix-convex languages, where a language is *suffix-convex* if it satisfies the condition that, if a word w and its suffix u are in the language, then so is every suffix of w that has u as a suffix [1,15]. We are interested only in regular suffix-free languages.

The syntactic complexity of prefix-free languages was proved to be n^{n-2} in [4]. The syntactic complexities of suffix-, bifix-, and factor-free languages were also studied in [4], and the following lower bounds were established $(n-1)^{n-2}+n-2$, $(n-1)^{n-3}+(n-2)^{n-3}+(n-3)2^{n-3}$, and $(n-1)^{n-3}+(n-3)2^{n-3}+1$, respectively. It was conjectured that these bounds are also upper bounds; we prove the conjecture for suffix-free languages in this paper. Moreover, we reduce the alphabet size of the witness language reaching the upper bound for suffix-free languages to five letters instead of $n+2$, and prove that

* Corresponding author.
E-mail addresses: brzozo@uwaterloo.ca (J.A. Brzozowski), msz@cs.uni.wroc.pl (M. Szykuła).

http://dx.doi.org/10.1016/j.ic.2017.08.014
0890-5401/

five is the minimal size. As well, we show that the transition semigroup of a minimal DFA accepting a witness language is unique for each n.

A much abbreviated version of these results appeared in [7].

2. Preliminaries

2.1. Languages, automata and transformations

Let Σ be a finite, non-empty alphabet and let $L \subseteq \Sigma^*$ be a language. The *left quotient* or simply *quotient* of a language L by a word $w \in \Sigma^*$ is denoted by $L.w$ and defined by $L.w = \{x \mid wx \in L\}$. A language is regular if and only if it has a finite number of quotients. We denote the set of quotients by $K = \{K_0, \dots, K_{n-1}\}$, where $K_0 = L = L.\varepsilon$ by convention. Each quotient K_i can be represented also as $L.w_i$, where $w_i \in \Sigma^*$ is such that $L.w_i = K_i$. The notation $K_i.w$ points out that each word $w \in \Sigma^*$ performs an action on the set K of quotients (states of the quotient DFA), and leads a quotient (state) K_i to quotient (state) $K_i.w$.

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.

The *quotient DFA* of a regular language L with n quotients is defined by $\mathcal{D} = (K, \Sigma, \delta_{\mathcal{D}}, K_0, F_{\mathcal{D}})$, where $\delta_{\mathcal{D}}(K_i, w) = K_j$ if and only if $K_i.w = K_j$, and $F_{\mathcal{D}} = \{K_i \mid \varepsilon \in K_i\}$. To simplify the notation, without loss of generality we use the set $Q = \{0, \dots, n-1\}$ of subscripts of quotients as the set of states of $\mathcal{D}$; then $\mathcal{D}$ is denoted by $\mathcal{D} = (Q, \Sigma, \delta, 0, F)$, where $\delta(i, w) = j$ if $\delta_{\mathcal{D}}(K_i, w) = K_j$, and F is the set of subscripts of quotients in $F_{\mathcal{D}}$. The quotient corresponding to $q \in Q$ is then $K_q = \{w \mid \delta_{\mathcal{D}}(K_q, w) \in F_{\mathcal{D}}\}$. The quotient $K_0 = L$ is the *initial* quotient. A quotient is *final* if it contains ε. A state q is *empty* (or a *sink state* or *dead state*) if its quotient K_q is empty.

The quotient DFA of L is a minimal DFA of L. The number of states in the quotient DFA of L (the quotient complexity of L) is therefore equal to the state complexity of L.

In any DFA, each letter $a \in \Sigma$ induces a transformation of the set Q of n states. Let $\mathcal{T}_Q$ be the set of all n^n transformations of Q; then $\mathcal{T}_Q$ is a monoid under composition. The *image* of $q \in Q$ under transformation t is denoted by qt. If s, t are transformations of Q, their composition is denoted $s \circ t$ and defined by $q(s \circ t) = (qs)t$; the $\circ$ is usually omitted. The *in-degree* of a state q in a transformation t is the cardinality of the set $\{p \mid pt = q\}$.

The *identity* transformation $\mathbf{1}$ maps each element to itself. For $k \geqslant 2$, a transformation (permutation) t of a set $P = \{q_0, q_1, \dots, q_{k-1}\} \subseteq Q$ is a *k-cycle* if $q_0t = q_1, q_1t = q_2, \dots, q_{k-2}t = q_{k-1}, q_{k-1}t = q_0$. A k-cycle is denoted by $(q_0, q_1, \dots, q_{k-1})$. If a transformation t of Q is a k-cycle of some $P \subseteq Q$, we say that t *has a k-cycle*. A transformation *has a cycle* if it has k-cycle for some $k \geqslant 2$. A 2-cycle (q_i, q_j) is called a *transposition*. A transformation is *unitary* if it changes only one state p to a state $q \neq p$; it is denoted by $(p \to q)$. A transformation is *constant* if it maps all states to a single state q; it is denoted by $(Q \to q)$.

The binary relation ω_t on $Q \times Q$ is defined as follows: For any $i, j \in Q$, $i \,\omega_t\, j$ if and only if $it^k = jt^\ell$ for some $k, \ell \geqslant 0$. This is an equivalence relation, and each equivalence class is called an *orbit* [9] of t. For any $i \in Q$, the orbit of t containing i is denoted by $\omega_t(i)$. An orbit contains either exactly one cycle and no fixed points or exactly one fixed point and no cycles. The set of all orbits of t is a partition of Q.

If $w \in \Sigma^*$ induces a transformation t, we denote this by $w\colon t$. A transformation mapping i to q_i for $i = 0, \dots, n-1$ is sometimes denoted by $[q_0, \dots, q_{n-1}]$. By a slight abuse of notation we sometimes represent the transformation t induced by w by w itself, and write qw instead of qt.

The *transition semigroup* of a DFA $\mathcal{D} = (Q, \Sigma, \delta, 0, F)$ is the semigroup of transformations of Q generated by the transformations induced by the letters of Σ. Since the transition semigroup of a minimal DFA of a language L is isomorphic to the syntactic semigroup of L [14], syntactic complexity is equal to the cardinality of the transition semigroup.

2.2. Suffix-free languages

For any transformation t, consider the sequence $(0, 0t, 0t^2, \dots)$; we call it the *0-path* of t. Since Q is finite, there exist i, j such that $0, 0t, \dots, 0t^i, 0t^{i+1}, \dots, 0t^{j-1}$ are distinct but $0t^j = 0t^i$. The integer $j - i$ is the *period* of t and if $j - i = 1$, t is *initially aperiodic*.

Let $Q = \{0, \dots, n-1\}$, and let $Q_M = \{1, \dots, n-2\}$ (the set of *middle* states). Let $\mathcal{D}_n = (Q, \Sigma, \delta, 0, F)$ be a minimal DFA accepting a language L, and let $T(n)$ be its transition semigroup. The following observations are well known [4,10]:

Lemma 1. *If L is a suffix-free language, then*

1. *There exists $w \in \Sigma^*$ such that $L.w = \emptyset$; hence $\mathcal{D}_n$ has an empty state, which is state $n-1$ by convention.*
2. *For $w, x \in \Sigma^+$, if $L.w \neq \emptyset$, then $L.w \neq L.xw$.*
3. *If $L.w \neq \emptyset$, then $L.w = L$ implies $w = \varepsilon$.*
4. *For any $t \in T(n)$, the 0-path of t in $\mathcal{D}_n$ is aperiodic and ends in $n-1$.*