



The Syntactic Complexity of Semi-flower Languages

Kitti Gelle, Szabolcs Iván

► To cite this version:

Kitti Gelle, Szabolcs Iván. The Syntactic Complexity of Semi-flower Languages. 21th International Conference on Descriptive Complexity of Formal Systems (DCFS), Jul 2019, Košice, Slovakia. pp.147-157, 10.1007/978-3-030-23247-4_11 . hal-02387298

HAL Id: hal-02387298

<https://inria.hal.science/hal-02387298>

Submitted on 29 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

The syntactic complexity of semi-flower languages

Kitti Gelle, Szabolcs Iván*

Department of Computer Science, University of Szeged, Hungary
{kgelle,szabivan}@inf.u-szeged.hu

Abstract. Semi-flower languages are those of the form L^* for some finite maximal prefix code L , or equivalently, those recognizable by a so-called semi-flower automaton, in which all the cycles have a common state q_0 , which happens to be the initial state and the only accepting state. We show that the syntactic complexity of these languages is exactly $n^n - n! + n$ (where n stands for the state complexity as usual) and that this bound is reachable with an alphabet of size n .

1 Introduction

The state complexity of a regular language is the number of states of its minimal automaton, or equivalently, the number of classes of its syntactic right-congruence. The syntactic complexity of a language is the number of classes of its syntactic congruence, or equivalently, the size of the transition monoid of its minimal automaton.

It is clear (and already observed by Maslov [9]) that if a language has state complexity n , then it can have a syntactic complexity of at most n^n as there are only so many transformations of an n -element set. Moreover, as three functions (an elementary swap, a circular permutation and a rank- $(n - 1)$ function) can generate all the transformations of a finite set, this maximal syntactic complexity can be reached by an automaton over a ternary alphabet. For the case of binary alphabets, Holzer and König [7] gave upper bounds for the maximal size of the transition monoid of an n -state minimal automaton while for the unary case and for the binary case for prime n , they determined a sharp bound.

When \mathcal{C} is a class of regular languages, then its syntactic complexity is a function over the single integer variable n , namely it is the maximum possible syntactic complexity of a language belonging to \mathcal{C} and having state complexity at most n . That is, for the whole class of the regular languages, this complexity is n^n . In the recent years, there is a growing interest of determining the syntactic complexity of subregular classes of languages (proper subclasses of the regular languages), e.g. for ideal and prefix- or factor closed languages [5], prefix-, suffix-,

* Ministry of Human Capacities, Hungary grant 20391-3/2018/FEKUSTRAT is acknowledged. Szabolcs Iván was supported by the János Bolyai Scholarship of the Hungarian Academy of Sciences.

bifix- and factor-free languages [4], several classes of star-free languages [3], R - and J -trivial languages [2], regular ideals [6] amongst others.

It is also an interesting question to determine the alphabet size needed to reach the maximum possible syntactic complexity: for the whole class of regular languages, an alphabet of ternary size suffices, but e.g. for factor-closed languages an alphabet of size 6 is needed [4], while for bifix-free languages $(n-2)^{n-3} + (n-3)2^{n-3} - 1$ generators are needed if $n \geq 6$ [13].

In the recent years, Singh and Krishna initiated the investigation of semi-flower automata [12, 11, 10], which are the minimal automata of valid code words over a finite maximal prefix code. In particular, in [12] they showed that if a circular semi-flower automaton over a binary alphabet has a single „branching point in”, or bpi (a state q is called a bpi if there are at least two tuples $(p, a) \in Q \times \Sigma$ with $pa = q$), then it has a linear syntactic complexity, and if it possesses exactly two bpis, then $2n(n+1)$ is a sharp bound on its syntactic complexity. Clearly, this is a serious restriction: it essentially restricts the elementary transformations to a circular permutation and some semi-flower transformation of rank at most 2 (for the definitions, see the Notation section). They also remark that over a ternary alphabet there exists a semi-flower automaton with two bpis having larger syntactic complexity than $2n(n+1)$.

In this paper we determine that the syntactic complexity of languages recognizable by semi-flower automata (without placing any restriction on the number of their “branch points going in”) is $n^n - n! + n$ and show that this bound is reachable by an alphabet of size n .

2 Notation and some facts

We assume the reader has some knowledge in automata and formal language theory (a standard resource is [8]). In this paper an *automaton* is a triple $\mathcal{A} = (Q, \Sigma, \cdot)$ with Q and Σ being the finite sets of *states* and *input symbols* or *letters*, and \cdot is an action $Q \times \Sigma \rightarrow Q$, written in infix notation: $q \cdot a$ for $q \in Q$ and $a \in \Sigma$. The action is extended to words acting on the states in the usual unique way as $q \cdot \varepsilon = q$ for the empty word ε and $q \cdot (ua) = (q \cdot u) \cdot a$ for each word $u \in \Sigma^*$ and $a \in \Sigma$. The transformation $q \mapsto q \cdot u$ for the word $u \in \Sigma^*$ is denoted as $u^{\mathcal{A}}$. The action is also extended to sets of states as $Q' \cdot u = \{q \cdot u : q \in Q'\}$.

When $f : Q \rightarrow Q$ is a transformation of some set Q , and $q \in Q$ is a member of its domain, then we often write $q \cdot f$ for $f(q)$, that is, writing the function application as a right action, moreover, for composing functions we define $f \circ g$ as $q \mapsto (q \cdot f) \cdot g$ (note the order). This way it holds that $(uv)^{\mathcal{A}} = u^{\mathcal{A}} \circ v^{\mathcal{A}}$.

We denote the transition monoid of an automaton $\mathcal{A} = (Q, \Sigma, \cdot)$ by $T(\mathcal{A})$, that is, $T(\mathcal{A})$ is the monoid over the set $\{u^{\mathcal{A}} : u \in \Sigma^*\} \subseteq Q^Q$, equipped by function composition where $u^{\mathcal{A}} \circ v^{\mathcal{A}} = (uv)^{\mathcal{A}}$. For a function $f : Q \rightarrow Q$, let $\text{rank}(f)$ stand for $|Qf|$, the size of the image of f .

The automaton $\mathcal{B} = (Q, \Delta, \bullet)$ is a *renaming* of the automaton $\mathcal{A} = (Q, \Sigma, \cdot)$ if for each $b \in \Delta$ there exists some $b' \in \Sigma$ with $b^{\mathcal{B}} = (b')^{\mathcal{A}}$. Since in that case $(b_1 \dots b_k)^{\mathcal{B}} = (b'_1 \dots b'_k)^{\mathcal{A}}$ for each k and $b_1, \dots, b_k \in \Delta$, we get that $T(\mathcal{B})$ is

a submonoid of $T(\mathcal{A})$. The automaton $\mathcal{B} = (Q', \Sigma, \bullet)$ is a *sub-automaton* of $\mathcal{A} = (Q, \Sigma, \cdot)$ if $Q' \subseteq Q$ and the action of \mathcal{B} is the restriction of the action of \mathcal{A} : $q \bullet a = q \cdot a$ for each $q \in Q'$ and $a \in \Sigma$. The automaton \mathcal{B} is a *homomorphic image* or *quotient* of \mathcal{A} if there exists some surjective mapping $h : Q \rightarrow Q'$ such that $h(q \cdot a) = h(q) \bullet a$. If the mapping h is a bijection, then the two automata are *isomorphic*. An automaton which is a homomorphic image of a subautomaton of \mathcal{A} is called a *divisor* of \mathcal{A} .

An automaton $\mathcal{A} = (Q, \Sigma, \cdot)$ is *minimal* with respect to some initial state $q_0 \in Q$ and a set $F \subseteq Q$ of final states if $\{q_0 \cdot u : u \in \Sigma^*\} = Q$ and for each pair $p \neq q$ of distinct states, there exists a word $u \in \Sigma^*$ with either $pu \in F$ and $qu \notin F$ or $pu \notin F$ and $qu \in F$.

The *language recognized by* $\mathcal{A} = (Q, \Sigma, \cdot)$ from $q_0 \in Q$ with $F \subseteq Q$ is the set $L(\mathcal{A}, q_0, F) = \{u \in \Sigma^* : q_0 \cdot u \in F\}$ of words. A language $L \subseteq \Sigma^*$ is recognizable or regular if $L = L(\mathcal{A}, q_0, F)$ for some finite automaton \mathcal{A} , initial state q_0 and set F of final states. It is well-known that for any regular language L there exists a minimal automaton \mathcal{A}_L which recognizes L from some initial state with some set of final states, moreover, \mathcal{A}_L divides every automaton in which L can be recognized.

\mathcal{A} is called a *semi-flower automaton* [10] (in short, SFA) if all cycles in \mathcal{A} have a common state. An automaton \mathcal{A} is called *circular* if some letter acts as a circular permutation on the states of \mathcal{A} , and is a *circular semi-flower automaton*, CSFA in short [11], if it is both circular and semi-flower.

Clearly, any circular automaton $\mathcal{A} = (Q, \Sigma, \cdot)$ is minimal with respect to any initial state $q_0 \in Q$ and the set $F = \{q_0\}$ of final states: if a induces a circular permutation in \mathcal{A} , then we have $Q = \{q_0 \cdot a^k : 0 \leq k < n\}$ where $n = |Q|$. On the other hand, if $p = q_0 \cdot a^k$ and $q = q_0 \cdot a^t$ are distinct states, then $p \cdot a^{n-k} = q_0$ while $q \cdot a^{n-k} \neq q_0$.

Let $[n]$ stand for the set $\{1, \dots, n\}$. We call a function $f : [n] \rightarrow [n]$ a *semi-flower transition* over $[n]$ if $i < i \cdot f$ for each $1 \leq i < n$. We denote the rotation operation by $+$: for a state $i \in [n]$ and an integer k , let $i + k$ stand for $((i + k - 1) \bmod n) + 1$. In particular, $n + 1 = 1$, and the mapping $i \mapsto i + 1$ (which happens to be a circular permutation) is a semi-flower transition.

The following proposition relates circular semi-flower automata and semi-flower transitions:

Proposition 1. *An automaton \mathcal{A} over some alphabet Σ is a semi-flower automaton if and only if it is isomorphic to some automaton of the form $([n], \Sigma, \cdot)$ such that each letter induces a semi-flower transition over $[n]$.*

This latter form is called a normal form of semi-flower automata.

Proof. If each letter induces a semi-flower transition over $[n]$ in $\mathcal{A} = ([n], \Sigma, \cdot)$ then we claim that each cycle contains the state n .

Indeed, let $p_1 \xrightarrow{a_1} p_2 \xrightarrow{a_2} \dots \xrightarrow{a_k} p_{k+1} = p_1$ be a cycle in \mathcal{A} . Then $p_i \geq p_{i+1}$ for some i , which, as a_i induces a semi-flower transition, can happen only if $p_i = n$. Thus, \mathcal{A} is semi-flower.

For the other direction, assume $\mathcal{A} = (Q, \Sigma, \cdot)$ is an n -state semi-flower automaton and let $q_n \in Q$ be a common state of all the cycles of \mathcal{A} . Then, considering the graph G with vertex set $V = Q - \{q_n\}$ and edge set $\{(p, pa) : a \in \Sigma, p, pa \in V\}$ we get that G is a directed acyclic graph, hence its vertices can be ordered as $q_1 < q_2 < \dots < q_{n-1}$ such that for each edge (p, q) of G we have $p < q$. Then, the mapping $q_i \mapsto i$ establishes an isomorphism between \mathcal{A} and a semi-flower automaton in normal form. \square

Clearly, if \mathcal{A} is a *circular* semi-flower automaton in normal form, then the letter a inducing a circular permutation of $[n]$ has to induce the function $i \mapsto i + 1$ as this is the only semi-flower permutation over $[n]$.

3 Minimal circular semi-flower automata and syntactic complexity

A language L is a *semi-flower language* if $L = L(\mathcal{A}, q_0, \{q_0\})$ for some semi-flower automaton $\mathcal{A} = (Q, \Sigma, \cdot)$ and state $q_0 \in Q$, which is a common state of all cycles of \mathcal{A} .

A language $P \subseteq \Sigma^+$ is a *prefix code* if there are no words $u, v \in P$ with u being a proper prefix of v , and is a *maximal prefix code* if additionally, for any word $w \notin P$, the set $P \cup \{w\}$ is not prefix-free anymore. If P is a maximal prefix code, $u \in \Sigma^*$ is a proper prefix of some member of P and $a \in \Sigma$ is a letter, then ua is still a (not necessarily proper) prefix of some member of P . A good reference on codes is [1].

It holds that semi-flower languages are exactly languages of the form P^* for some finite maximal prefix code $P \subseteq \Sigma^+$. Indeed, given $L = L(\mathcal{A}, q_0, \{q_0\})$ for the semi-flower automaton \mathcal{A} , we can take the language

$$P = \{u \in \Sigma^* : q_0 u = q_0, q_0 v \neq q_0 \text{ for any proper nonempty prefix } v \text{ of } u\}$$

which is a finite language since otherwise some cycle would avoid the initial-final state q_0 . For the other direction, for a finite maximal prefix-free language P one can take the state set as $Q = \{u \in \Sigma^* : u \text{ is a proper prefix of some } v \in P\}$, define the action as $u \cdot a = \varepsilon$ if $ua \in P$ and ua otherwise, and pick ε as q_0 , the resulting semi-flower automaton recognizes the language P^* .

The syntactic complexity of a class \mathcal{L} of regular languages is a unary function over the single variable n : for each n , its value is defined as

$$\max\{|T(\mathcal{A}_L)| : L \in \mathcal{L} \text{ and } \mathcal{A}_L \text{ has at most } n \text{ states}\}.$$

Since up to isomorphism and renaming there is only a finite number of automata having at most n states, this notion is well-defined for any class \mathcal{L} of regular languages.

In this paper we show that the syntactic complexity of semi-flower languages is exactly $n^n - n! + n$ by analyzing the size of $T(\mathcal{A})$ for a particular (circular) semi-flower automaton $\mathcal{A} = \mathcal{A}_n$ for each n .

4 The transition monoid of semi-flower automata

If in an n -state automaton \mathcal{A} , no letter acts as a permutation on the state set, then its transition monoid does not contain any nontrivial permutation, thus in that case, $|T(\mathcal{A})| \leq n^n - n! + 1$.

Clearly, if some letter induces a permutation in a semi-flower automaton, then it has to be a circular permutation (otherwise there would be two disjoint cycles in its graph). So in the remaining part of the section we deal with the transition monoid of an n -state *circular* semi-flower automaton.

Let us begin by handling the permutations present in $T(\mathcal{A})$ when \mathcal{A} is a circular semi-flower automaton.

Proposition 2. *If $f : Q \rightarrow Q$ is a permutation belonging to $T(\mathcal{A})$ for some circular semi-flower automaton \mathcal{A} over the state set Q , in which a induces a circular permutation, then f is induced by some word of the form a^k where $0 \leq k < n$.*

Proof. Let us assume $\mathcal{A} = ([n], \Sigma, \cdot)$ is in the normal form specified by Proposition 1. Since the only permutation which is a semi-flower transition is the transformation $i \mapsto i + 1$, $b^{\mathcal{A}}$ has to be this function for each letter b inducing a permutation.

Also, if $f = (b_1 b_2 \dots b_k)^{\mathcal{A}}$ is a permutation, then each $b_i^{\mathcal{A}}$ has to be a permutation, thus $f = (a^k)^{\mathcal{A}}$ in that case for some k and, as $(a^n)^{\mathcal{A}}$ is the identity map, we get $f = (a^k)^{\mathcal{A}}$ for some $0 \leq k < n$. \square

Note that there are n such permutations, so $T(\mathcal{A})$ contains n permutations and all its other members have rank less than n . Hence, the absolute maximum syntactic complexity an n -state (circular) semi-flower automaton can have is $n^n - n! + n$. In the rest of the paper we show that this bound is attainable.

Let us fix $n \geq 1$, the state set $Q = [n]$ and the n -state automaton $\mathcal{A} = (Q, \Sigma, \cdot)$ for the rest of the section, where

$$\Sigma = \{b_i : i \in [n]\}$$

where for each $i \in [n]$, the action of b_i is defined as

$$j \cdot b_i = \begin{cases} j + 1 & \text{if } j < n \\ i & \text{otherwise.} \end{cases}$$

Since all the functions $b_i^{\mathcal{A}}$ are semi-flower transitions over $[n]$, \mathcal{A} is a semi-flower automaton over n letters. Observe that b_1 induces the circular permutation $i \mapsto i + 1$, thus \mathcal{A} is a circular semi-flower automaton. To make a visual distinction, we also refer to b_1 as a and to ease notation, we frequently identify the mapping $b^{\mathcal{A}}$ with the letter b .

In the rest of the section we aim to show that $T(\mathcal{A})$ contains all the transformations of $[n]$ with rank less than n . Since the circular permutation induced by a is also present, we also have n permutations in $T(\mathcal{A})$ by Proposition 2, so this

yields $|T(\mathcal{A})| = n^n - n! + n \geq n^n - n! + 1$, proving that the syntactic complexity of semi-flower languages is $n^n - n! + n$ as well (and this bound is attainable by a circular semi-flower automaton over n letters.)

In the following, we show that several fundamental transformations, as “merging” two states, or “swapping” two states of some proper subset of the state set, each belong to $T(\mathcal{A})$.

Lemma 1. *For all $p, q \in Q$, the function $f_{p,q} : Q \rightarrow Q$, where $p \cdot f = q$ and for all $r \neq p$ we have $r \cdot f = r$, belongs to $T(\mathcal{A})$.*

Proof. Let us consider the function $f = a^{n-p} \circ b_{q+n-p+1} \circ a^{p-1}$. (Here $q+(n-p+1)$ is understood with the rotation operation from the Notation section, so this value is a member of $[n]$).

Clearly, for p we have $p \cdot a^{n-p} = n$, $n \cdot b_{q+n-p+1} = q+n-p+1$ (in the rotating sense) and $(q+n-p+1) \cdot a^{p-1} = q+n = q$, so that $p \cdot f = q$. On the other hand, for any $r \neq p$ we have $r \cdot a^{n-p} = r+n-p \neq n$ so $(r+n-p) \cdot b_{q+n-p+1} = r+n-p+1$ and $(r+n-p+1) \cdot a^{p-1} = r+n = r$. Thus indeed, $r \cdot f = r$ for each $r \neq p$.

For an example, see Figure 1. \square

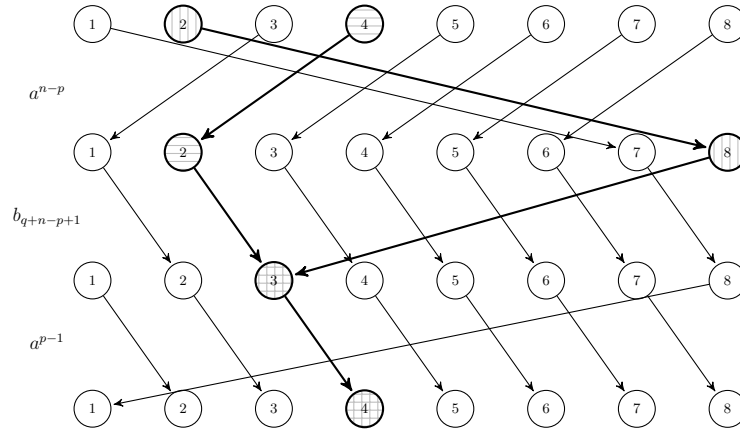


Fig. 1: The semi-flower transitions creating a function $f_{p,q}$ which merges two states (Here $p = 2, q = 4$)

As a corollary we can collapse the kernel of any transformation of Q , each kernel class to one member of the class. (Recall that the kernel of a function $f : Q_1 \rightarrow Q_2$ is the equivalence relation \sim_f over Q_1 defined as $p \sim_f q \Leftrightarrow pf = qf$.)

Corollary 1. *For any transformation $f : Q \rightarrow Q$, there exists some function $g \in T(\mathcal{A})$ such that*

- for each $p, q \in Q$, $pf = qf$ if and only if $pg = qg$;

– for each $p \in Q$, $pgf = pf$.

In other words, g associates to each state $q \in Q$ a representative state of the class of q in the kernel of f .

Proof. Lemma 1 states $f_{p,q}$ is a member of $T(\mathcal{A})$ for each $p, q \in Q$.

Let $C \subseteq Q$ be a nonempty set of states. We claim that there exists some function $g_C \in T(\mathcal{A})$ and a state $p \in C$ such that $C \cdot g_C = \{p\}$ and $r \cdot g_C = r$ for each $r \notin C$. Indeed, let $C = \{q_1, \dots, q_k\}$. If C is a singleton, then the identity function trivially satisfies the conditions, so assume $|C| > 1$.

Then the function $f_{q_1, q_k} \circ f_{q_2, q_k} \circ \dots \circ f_{q_{k-1}, q_k}$ satisfies the claim with $p = q_k$.

Hence, for any transformation $f : Q \rightarrow Q$, if $C_1, \dots, C_k \subseteq Q$ are the classes of the kernel of f (that is, the sets C_i give a partition of Q and for each $p, q \in Q$, $p \cdot f = q \cdot f$ if and only if p and q belong to the same C_i), then the function $g = g_{C_1} \circ \dots \circ g_{C_k}$ satisfies the conditions of the corollary. \square

The next lemma states that if we have some proper subset Q' of the states, then we can swap two of these states and retain the other members of Q' in their original place with some function from $T(\mathcal{A})$:

Lemma 2. *Assume $Q' \subsetneq Q$ is some subset of the states and $p, q \in Q'$. Then some function $f : Q \rightarrow Q$ satisfying $p \cdot f = q$, $q \cdot f = p$ and $r \cdot f = r$ for all $r \in Q' - \{p, q\}$ belongs to $T(\mathcal{A})$.*

Proof. If $p = q$, then the claim holds trivially as the identity belongs to $T(\mathcal{A})$ so we can assume $p \neq q$.

By $|Q'| < n$, there exists some state $\ell \notin Q'$. We claim that the function $f = f_{p, \ell} \circ f_{q, p} \circ f_{\ell, q}$ satisfies the conditions of the Lemma. Indeed, if $r \notin \{p, q, \ell\}$, then none of the functions involved moves r , so that $r \cdot f = r$ for each $r \in Q' - \{p, q\}$; and clearly, $p \cdot f_{p, \ell} = \ell$, $\ell \cdot f_{q, p} = q$ and $q \cdot f_{\ell, q} = p$ shows $p \cdot f = q$ and similarly $q \cdot f_{p, \ell} = p$, $p \cdot f_{q, p} = q$ and $q \cdot f_{\ell, q} = p$ shows $q \cdot f = p$.

So this $f \in T(\mathcal{A})$ satisfies the conditions of the lemma.

For an example, see Figure 2. \square

Now in order to show an arbitrary function $f : Q \rightarrow Q$ with rank less than n , we dissect f as $f = f_1 \circ \pi \circ g$ where f_1 collapses the kernel classes of f into their representative elements (we can do that according to Corollary 1), π is an appropriate permutation on these representatives (Lemma 3 ensures the existence of such a suitable permutation in $T(\mathcal{A})$), and g is a “monotone” mapping (Lemma 4 will ensure that $g \in T(\mathcal{A})$).

Lemma 3. *For each $Q' \subsetneq Q$ and permutation $\pi : Q' \rightarrow Q'$ there exists some $f \in T(\mathcal{A})$ extending π : $q \cdot f = q \cdot \pi$ for each $q \in Q'$.*

Proof. By Lemma 2 we get that for any transposition t swapping two states of Q' there exists some member belonging to $T(\mathcal{A})$ which extends t . Since every permutation π can be written as a composition of transpositions, and $T(\mathcal{A})$ is closed under composition, we get that each permutation of $Q' \subsetneq Q$ can be extended to some member of $T(\mathcal{A})$. \square

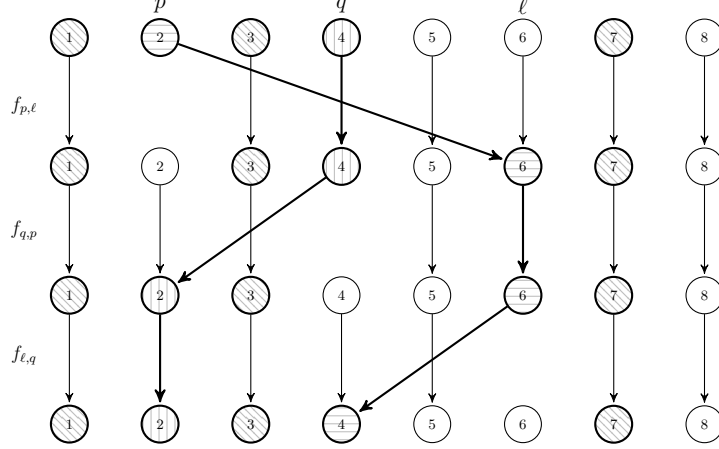


Fig. 2: Steps for building a swap function for two states (here $p = 2, q = 4$) from semi-flower transitions. The marked states belong to Q' .

For the sets $Q' \subseteq Q$, let us call a mapping $f : Q' \rightarrow Q$ *monotone* if $i < j$, $i, j \in Q'$ implies $i \cdot f < j \cdot f$. Our next lemma states that all the monotone maps belong to $T(\mathcal{A})$.

Lemma 4. *For each set $Q' \subseteq Q$ and monotone mapping $f : Q' \rightarrow Q$ there exists some $f' \in T(\mathcal{A})$ extending f .*

Proof. The statement is vacuously satisfied when $Q' = \emptyset$ so assume $Q' \neq \emptyset$.

For two functions $f, g : Q' \rightarrow Q$, let us define $d(f, g)$ as $\sum_{q \in Q'} |q \cdot f - q \cdot g|$. We show that if for two monotone functions $f, g : Q' \rightarrow Q$ we have $d(f, g) > 0$, and g has some extension $g_* \in T(\mathcal{A})$, then there also exists some monotone function $g' : Q' \rightarrow Q$ having an extension $g'_* \in T(\mathcal{A})$ with $d(f, g') < d(f, g)$. As this distance can only have nonnegative integer values, and the identity function is a monotone function belonging to $T(\mathcal{A})$ (so we can start the induction somewhere), this proves the statement.

So let us consider two monotone functions $f, g : Q' \rightarrow Q$ with $d(f, g) > 0$. This means that there exists some state $q \in Q'$ with $q \cdot f \neq q \cdot g$. Let us choose q so that $q' \cdot f = q' \cdot g$ for each $q' < q$.

There are two cases: either $q \cdot f < q \cdot g$ or $q \cdot g < q \cdot f$.

If $q \cdot f < q \cdot g$, then no state r with $q \cdot f \leq r < q \cdot g$ can be in the image of g : by monotonicity, $p \cdot g = r < q \cdot g$ would yield $p < q$ but by the choice of q , $p \cdot g = p \cdot f < q \cdot f$ holds in that case. By Lemma 1, the function $g_0 = f_{q \cdot g, q \cdot f}$ belongs to $T(\mathcal{A})$, and $q \cdot (g \circ g_0) = (q \cdot g) \cdot g_0 = q \cdot f$, and since g is injective, we have $q' \cdot g \neq q \cdot g$ if $q' \neq q$, so that $q' \cdot g \cdot g_0 = q' \cdot g$ for each $q' \in Q'$. Observe also that $g \circ g_0$ is still a monotone function. As $g \circ g_0$ belongs to $T(\mathcal{A})$ if so does g , and $d(f, g \circ g_0) < d(f, g)$, this proves the claim in this subcase.

If $q \cdot g < q \cdot f$, then consider the following sequence $q = q_0 < q_1 < \dots$ of members of Q' : first, let us set $q_0 = q$, then for each $t \geq 0$, if $(q_t \cdot g) + 1$ belongs

to the image of g , say $p \cdot g = (q_t \cdot g) + 1$ for the state $p \in Q'$, then let us set $q_{t+1} = p$, otherwise let q_t be the terminating element of the sequence.

As g is a monotone function, we get that $q_t < q_{t+1}$ holds for each t where the sequence is defined for q_{t+1} . Thus, the sequence has to be finite. Moreover, by construction, the sequence $q_0 \cdot g, q_1 \cdot g, \dots$ contains consecutive states by definition for the states $q_0 < q_1 < \dots$, thus, by $q_0 \cdot g < q_0 \cdot f$, applying induction we get $q_t \cdot g < q_t \cdot f$ for each valid index t . Hence, for the last state q_t of the sequence we have by construction that $q_t \cdot g < q_t \cdot f$ and that $(q_t \cdot g) + 1$ does not belong to the image of g . (Observe that being less than $q_t \cdot f$, the state $q_t \cdot g$ cannot be n .) By Lemma 1, the function $g_0 = f_{q_t \cdot g, (q_t \cdot g) + 1}$ belongs to $T(\mathcal{A})$ so again, for the monotone function $g \cdot g_0$ we get $q_t \cdot g \cdot g_0 = (q_t \cdot g) + 1$ and for any other member q of Q' , $q \cdot g = q \cdot g \cdot g_0$. Again, $g \circ g_0$ belongs to $T(\mathcal{A})$ if so does g , and $d(f, g \circ g_0) < d(f, g)$, proving this subcase and the lemma as well. \square

Lemma 5. For each $Q' \subsetneq Q$ and injective mapping $f : Q' \rightarrow Q$ there exists some $g \in T(\mathcal{A})$ extending f .

Proof. Let $Q' = \{q_1, \dots, q_k\}$ with $q_1 < \dots < q_k$ and let $\pi : Q' \rightarrow Q'$ be the unique permutation with $q_i \cdot \pi \cdot f < q_{i+1} \cdot \pi \cdot f$ for each $1 \leq i < k$. (That is, $q_i \cdot \pi$ is q_j if and only if $q_i \cdot f$ is the j th least element of $Q' \cdot f$).

Then $f = \pi \circ f'$ for the monotone map $f' : Q' \rightarrow Q$ with $q_i \cdot f' = (q_i \cdot \pi^{-1}) \cdot f$, hence by Lemmas 3 and 4 there is some function $g \in T(\mathcal{A})$ extending f . \square

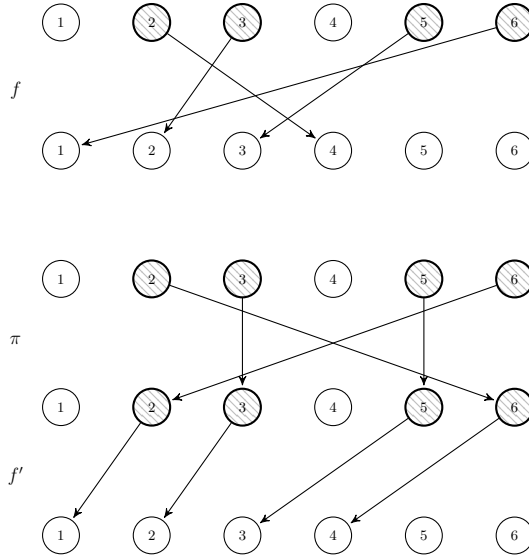


Fig. 3: Example of the decomposition $f = \pi \cdot f'$. The marked states are belonging to Q' .

Equipped with our lemmas, we can prove the main theorem of the present paper:

Theorem 1. *The monoid $T(\mathcal{A})$ contains all the transformations of $[n]$ having rank less than n , thus $|T(\mathcal{A})| = n^n - n! + n$ is the syntactic complexity of the semi-flower languages.*

This bound can be reached by an alphabet of size n for each n .

Proof. Let $f : Q \rightarrow Q$ be a function with $\text{rank}(f) < n$. By Corollary 1, there exists some function $g \in T(\mathcal{A})$ mapping each state to a representative of its kernel class, that is, with some $Q' \subsetneq Q$, $|Q'| = \text{rank}(f)$ such that $Q \cdot g = Q'$, satisfying $q \cdot g \cdot f = q \cdot f$ for each state $q \in Q$, moreover, with $p \cdot f = q \cdot f$ implying $p \cdot g = q \cdot g$. Then, let f' denote the restriction of f to Q' . Then, by construction, f' is an injective mapping from Q' to Q , thus it has an extension $g' \in T(\mathcal{A})$ by Lemma 5. Thus, for the mapping $h = g \circ g'$ we have $q \cdot h = q \cdot g \cdot g' = q \cdot g \cdot f' = q \cdot g \cdot f = q \cdot f$ for each state $q \in Q$, proving the theorem. \square

5 Conclusion

We showed that the syntactic complexity of semi-flower languages is exactly $n^n - n! + n$ and that in order to reach this complexity, an alphabet of linear size suffices. We left open the question whether n letters are needed to achieve this or maybe an even smaller alphabet suffices.

The authors wish to thank the reviewers for their careful work which improved the presentation of the paper. Reviewer 3 even sketched a proof showing that the minimal number of letters to achieve maximal syntactic complexity is $\lceil \frac{n+1}{2} \rceil$, which we include here, slightly reformatted.

Theorem 2 (Reviewer 3). *The maximal syntactic complexity $n^n - n! + n$ of an n -state semi-flower automaton can be reached by an alphabet of size $\lceil \frac{n+1}{2} \rceil$ and this bound is sharp.*

Proof. One can observe that in $\mathcal{A} = ([n], \Sigma, \cdot)$ studied in the previous section still contains redundant letters. If $1 < k < n$, then $(n+2-k) \cdot a^{k-2} \cdot b_k = k$ while $i \cdot a^{k-2} \cdot b_k = i + k - 1$ for each $i \neq n+2-k$. In particular, the image of $i = 1$ is also k , so $a^{k-2}b_k$ merges 1 and $n+2-k$ into k and permutes the $(n-1)$ -element set $\{2, \dots, n\}$ (as 1 does not belong to its image). Hence $(a^{k-2}b_k)^{(n-1)!}$ acts as the identity on $\{2, \dots, n\}$ and maps 1 to $n+2-k$. Thus, $a(a^{k-2}b_k)^{(n-1)!}$ maps n to $n+2-k$ and acts as $i \mapsto i+1$ on the other states, so it induces the same transformation as b_{n+2-k} . Thus, with an appropriate composition of b_1 and b_2 one can induce the same transformation as $b_{n+2-2} = b_n$, from b_1 and b_3 we get b_{n-1} and so on, thus the subset $\{b_i : 1 \leq i \leq \lceil \frac{n+1}{2} \rceil\}$ of Σ generates the same transition monoid $T(\mathcal{A})$, and so $\lceil \frac{n+1}{2} \rceil$ letters suffice.

On the other hand, assume $\mathcal{A} = ([n], \Delta, \cdot)$ is a semi-flower automaton in normal form. Let us define the distance of two states $p \leq q \in [n]$ as $d(p, q) = \min\{q - p, n + p - q\}$, that is, their “circular” distance and if $p > q$, then let

$d(p, q) = d(q, p)$. Clearly, $d(p, q) = d(pa, qa)$ whenever a has rank n . Assume $w = a_1 \dots a_k$ induces a transformation of rank $n - 1$. Then there is a least index $i \in [k]$ such that a_i induces a transformation of rank $n - 1$ and each a_j , $j < i$ induces the permutation $\ell \mapsto \ell + 1$. Let $p \neq q$ be the states merged by a_i and p', q' be the states with $p'a_1 \dots a_{i-1} = p$ and $q'a_1 \dots a_{i-1} = q$. Then, w merges the states p' and q' whose distance is $d(p', q') = d(p, q)$ depending only on a_i . Hence, if $T(\mathcal{A})$ contains all the transformations of rank $n - 1$, then for each possible distance $D > 0$, the alphabet Δ has to contain at least one letter of rank $n - 1$ merging two states of distance D . Since there are $\lfloor \frac{n}{2} \rfloor$ possible distances on the cycle of length n , along with the letter inducing the circular permutation an alphabet of size at least $\lfloor \frac{n}{2} \rfloor + 1 = \lceil \frac{n+1}{2} \rceil$ is needed in a circular semi-flower automaton to generate all the transformations of rank $(n - 1)$. \square

References

1. Berstel, J., Perrin, D., Reutenauer, C.: Codes and Automata (Encyclopedia of Mathematics and Its Applications). Cambridge University Press, New York, NY, USA, 1st edn. (2009)
2. Brzozowski, J., Li, B.: Syntactic complexity of R - and J -trivial regular languages. International Journal of Foundations of Computer Science **25**(07), 807–821 (2014). <https://doi.org/10.1142/S0129054114400097>, <https://doi.org/10.1142/S0129054114400097>
3. Brzozowski, J., Li, B., Liu, D.: Syntactic complexities of six classes of star-free languages. J. Autom. Lang. Comb. **17**(2), 83–105 (Mar 2012), <http://dl.acm.org/citation.cfm?id=3173440.3173443>
4. Brzozowski, J., Li, B., Ye, Y.: Syntactic complexity of prefix-, suffix-, bifix-, and factor-free regular languages. Theoretical Computer Science **449**, 37 – 53 (2012), descriptional Complexity of Formal Systems (DCFS 2011)
5. Brzozowski, J., Ye, Y.: Syntactic complexity of ideal and closed languages. In: Mauri, G., Leporati, A. (eds.) Developments in Language Theory. pp. 117–128. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
6. Brzozowski, J.A., Szykuła, M., Ye, Y.: Syntactic complexity of regular ideals. Theory of Computing Systems **62**(5), 1175–1202 (Jul 2018)
7. Holzer, M., König, B.: On deterministic finite automata and syntactic monoid size. Theoretical Computer Science **327**(3), 319 – 347 (2004)
8. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley (1979)
9. Maslov, A.N.: Estimates of the number of states of finite automata. Dokl. Akad. Nauk SSSR **194**, 1266–1268 (1970)
10. Singh, S.N.: Semi-Flower Automata. Ph.D. thesis, IIT Guwahati, India (2012)
11. Singh, S.N., Krishna, K.V.: The holonomy decomposition of some circular semi-flower automata. Acta Cybernetica **22**(4), 81–95 (Jan 2016)
12. Singh, S.N., Krishna, K.V.: On syntactic complexity of circular semi-flower automata. In: Cămpăanu, C. (ed.) Implementation and Application of Automata. pp. 312–323. Springer International Publishing, Cham (2018)
13. Szykuła, M., Wittnebel, J.: Syntactic complexity of bifix-free regular languages. Theoretical Computer Science (2019). <https://doi.org/https://doi.org/10.1016/j.tcs.2018.12.025>, <http://www.sciencedirect.com/science/article/pii/S0304397519300039>