On the complexity of a family of k-context-free sequences

Marion Le Gonidec

Laboratoire IMATH - Université du Sud Toulon Var

Abstract

We introduce a large family of k-context-free sequences for wich we give a polynomial upper bound of their complexity functions.

Key words: context-free sequences, complexity, pushdown automata

Introduction

The study of automatic sequences and more generally infinite sequences generated by simple algorithms is a classical topic in combinatorics. Automatic sequences are closely related to automata theory and language theory but these sequences also constitute a interesting tool in other mathematical areas as symbolic dynamics or number theory.

There are several characterizations of k-automatic sequences using finite automata, regular languages, k-uniform morphisms or subsequences k-kernel (for an overview, see for example [AS03a]). These characterizations have led to different generalizations of the notion of automatic sequences and fruitful results have been obtained in each case. Let us mention substitutive (or morphic) sequences, which are generated by subtitutions over finite alphabets (see for example [PF02] for an overview), k-regular sequences, which are sequences with values in a ring having finitely generated k-kernel [AS92, AS03b], or k^∞-automatic sequences, which are generated by k-uniform morphisms over countable alphabets [LG06b, LG06a]. Let also mention a generalization of the notion of k-automatic sequences for abstract numerations based upon regular languages instead of usual k-ary representaion languages [Rig00].

It appears also natural to consider an other way to extend this notion, based upon their characterization using regular languages. As suggested in [AS03a], following Chomsky's hierarchy over languages and machines [Cho56], one can asks about a similar construction of a family of infinite sequences using context-free languages instead of regular language, that is using pushdown automata instead of usual finite automata.

In this direction, D. Hamm [Ham98] formally introduces k-context-free sequences and k-quasi-context-free sequences (see section 2 for definitions). The question we partially answer in this article is the following : How can be extended complexity results obetained for k-automatic sequences in such a framework [AS03a, Open Problem 6.3] ?

The complexity function (or subword complexity) of an infinite sequence $\mathbf{a} = a_0a_1a_2\dots$ over a finite alphabet is the non decreasing sequence of integers $\mathbf{p_a} = p_\mathbf{a}(0)p_\mathbf{a}(1)p_\mathbf{a}(2)\dots$ where $p_\mathbf{a}(n)$ is the number of different factors of length n occurring in $\mathbf{a}$ with convention $p_\mathbf{a}(0) = 1$.

There are many motivations arising from different aeras for the study of this function, for example : dynamical motivations as this function can be linked to the entropy of the dynamical system associated to the sequence (see for example [Kůr03] or [PF02]), computationnal motivations as this function is also linked with the Kolmogorov complexity of prefixes of the sequence (see [Sta06]) or motivations arising from number theory as the complexity function of the k-ary expansion of a real number can be used to determine some of its algebraic properties (see [AB07]).

Email address: marion.le-gonidec@univ-tln.fr (Marion Le Gonidec)

This function can really have various behaviour depending on the sequence : from a constant function to an exponential growth function and from a simple to an irregular increasing functions. Its computation often remain awkward as there is no universal algorithm to exactly compute it for a given sequence, even if the considered sequence is generated by some simple machine. Consequently, computing complexity functions has been the aim of many works (see [All94] or [Fer99] for surveys). Methods involved in their computation are often based on the study of special factors and synchronization principles. We refer to [Cas96] and [Cas97] for formulas linking special factors and complexity. These methods allow to compute complexity functions exactly in many particular cases.

When we deal with a large class of sequences, not being that sharp, it is quite natural to ask about a generic upperbound for possible *growth orders* of the complexity functions of sequences and about their computing. By growth orders, we mean functions f satisfying $f(n) = \mathcal{O}(p_{\mathbf{a}}(n))$ and $p_{\mathbf{a}}(n) = \mathcal{O}(f(n))$ for at least one sequence $\mathbf{a}$ of the class. For example, it is well known that the possible growth orders of the complexity function of automatic sequences are 1 and n [Cob72]. For substitutive words, complexity functions are of type $\mathcal{O}(n^2)$ [ELR75]. This result had been improved by Pansiot [Pan85] for purely substitutive words, that is for fixed points of substitutions over finite alphabets : the possible growth orders of their complexity functions are 1, n, $n \log\log n$, $n \log n$ and n^2 [Pan85]. Displaying the possible growth orders of complexity functions of substitutive words, that is fixed points of substitutions over finite alphabet and their projections by letter-to-letter morphisms is still an open problem.

Regarding k-context-free sequences, Y. Moshe gives in [Mos08] a polynomial upperbound for the complexity of binary k-context-free sequences when the underlying pushdown automata is real-time deterministic recognizing by empty storage and accepting states.

The aim of this article is to extend the complexity result of [Mos08] to a larger family of k-context-free sequences, also based upon real-time deterministic pushdown automata but allowing all usual acceptance modes and arbitrary large alphabets.

In this article, we fix an integer $k \geq 2$ and we denote the set $\{0, 1, \dots, k-1\}$ by $[\![0, k-1]\!]$. For any fixed alphabet A, we use the following standard notations :

- A^* to name the free monoïd over A,
- A^+ to name A^* except the empty string,
- $A^{\leq n}$ and $A^{\geq n}$ to respectively name the set of words over A of length less than n and the set of words over A of length more than n.

Finite words and sequences will be written in contatenated form. Moreover, we will denote by $\mathrm{Pref}(w)$ the set of prefixes of a word $w \in A^*$ and by $\mathrm{Pref}_n(w)$ the prefix of length n of a word $w \in A^{\geq n}$.

1. k-pushdown automata

To construct k-context-free sequences over arbitrary large alphabets, we need to provide a way to generate what could be called k-context-free partitions of $\mathbb{N}$, that is, partitions for which the languages made of k-ary expansions of integers of a component are context-free languages. Thus, we have to simultaneously construct several (disjoint) context-free languages. To do this, the pushdown automata (PDA) point of view is more convenient than the context-free grammar point of view.

In this article, we will only consider real-time and deterministic PDA with input alphabet $[\![0, k-1]\!]$ so the set of transition rules can be considered as an action of $[\![0, k-1]\!]$ over the set of configurations. We name these machines *k-pushdown automaton* (k-PDA) for short. Moreover, we will only consider complete k-PDA as any k-PDA can be completed using a sink state so that all undefined transitions lead to this state.

According to these remarks, we represent a k-PDA as follows :

Definition 1.1. A *k-pushdown automaton* is a quadruple $\mathcal{A} = (Q, \Sigma, \phi, q_0 W^{(0)})$ where :

- Q is the set of states,
- Σ is the stack alphabet,
- $q_0 W^{(0)} \in Q\Sigma^*$ is the initial configuration,

– the set of transitions rules is a function $\phi : [\![0,k-1]\!] \times Q\,(\Sigma \cup \{\varepsilon\}) \to Q\Sigma^*$, extended to $[\![0,k-1]\!]^+ \times Q\Sigma^*$ by :
  – for all configurations $qW_0W_1\cdots W_L$ in $Q\Sigma^+$ and all integers $i \in [\![0,k-1]\!]$:

$$\phi(i, qW_0W_1\cdots W_L) = \phi(i, qW_0)W_1\cdots W_L,$$

  – for all words $w = w_0w_1\cdots w_l$ in $[\![0,k-1]\!]^*$ and all configurations qW in $Q\Sigma^*$:

$$\phi(w, qW) = \phi\left(w_l, \phi\left(\ldots \phi\left(w_1, \phi(w_0, qW)\right)\ldots\right)\right).$$

By convention, the empty word of $[\![0,k-1]\!]^*$ will be denote e while the empty word of Σ^* will be denote ε.

Notice that we will sometimes note a configuration with empty stack by $q\varepsilon \in Q\Sigma^*$ instead of q for more lisibility.

Remarks 1.2. There is a real restriction in considering only real-time deterministic PDA. Indeed, in finite automata background, allowing e-transitions or allowing non-determinism are not real relaxing hypotheses as the larger families of automata obtained this way generate exactly the same family of languages : regular languages. Dealing with PDA, this result is no longer true : the family of non-deterministic PDA generates strictly more languages than the family of deterministic PDA does and the family of real-time deterministic PDA generates strictly less languages than the family of deterministic PDA does (see for example [ABB97]).

Notice that if the stack alphabet is empty or if the transition function leave the stack uniformly bounded (i. e. there exists an integer $B \geq 0$ such that, any accessible configuration qW is in $Q\Sigma^{\leq B}$), the k-PDA exactly acts as a finite k-automaton.

Example 1.3. The graph of the 2-PDA $\mathcal{A} = (\{q_0, q_1, q_2\}, \{s, t\}, \phi, q_0\varepsilon)$ is represented on Figure 1. A transition $\phi(i, qW_0) = q'W'$ is symbolized by an arrow from q to q' labelled by $(i, W_0/W')$.

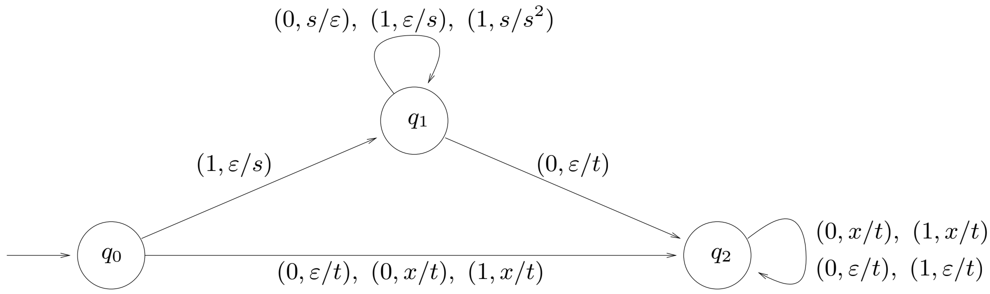

Fig. 1: Graph of the 2-PDA $\mathcal{A}$

Unlike to finite automata case, there are several way for a language to be recognizable by a PDA so a single k-PDA can recognize several context-free languages of $[\![0,k-1]\!]^*$ according to the form of the acceptance (or terminal) set $F \subset Q\Sigma^*$ (see [Aut87] for example for more details). Here follows the two usual different forms F can take :
– $F = F_Q\Sigma^*$ where $F_Q \subset Q$, in this case we talk about acceptance by accepting states,
– $F = F_Q\varepsilon$ where $F_Q \subset Q$, in this case we talk about acceptance by empty storage and accepting states.

This usual cases are in fact particular cases of the following one : $F = F_Q\Sigma'\Sigma^*$ where $F_Q \subset Q$ and $\Sigma' \subset \Sigma$ that it acceptance by topmost stack symbols and accepting states mode.

For example, the 2-PDA $\mathcal{A}$ introduce in Example 1.3 recognize different languages according to the choice of the acceptance set :
– If $F = q_1\Sigma^*$, $\mathcal{A}$ recognizes $\{w \in \{0,1\}^* \mid \forall u \in \mathrm{Pref}(w),\ |u|_0 \leq |u|_1\}$,
– If $F = q_1\varepsilon$, $\mathcal{A}$ recognizes $\{w \in \{0,1\}^* \mid \forall u \in \mathrm{Pref}(w),\ |u|_0 \leq |u|_1 \text{ et } |w|_1 = |w|_0\}$,

– If $F = \{q_0, q_1, q_2\}t$, $\mathcal{A}$ recognizes $\{w \in \{0,1\}^* \mid \exists u \in \text{Pref}(w),\ |u|_0 > |u|_1\}$.

As finite automata with ouput function generalize the notion of finite automata, we introduce a family of admissible output function for k-PDA, simulating the generic mode of acceptance by topmost stack symbols and accepting states.

Definition 1.4. Let $k \geq 2$ be an integer, A and A' two finite alphabets and L a language over A.

We call *admissible output function* a function $\Pi : L \to A'$ satisfying the following condition :

$$\text{For all words } w, v \in L^2,\ \text{if } \text{Pref}_2(w) = \text{Pref}_2(v) \text{ then } \ \Pi(w) = \Pi(v). \tag{1}$$

We call *k-PDA with output function* a sextuple $\mathcal{A} = (Q, \Sigma, \phi, q_0 W^{(0)}, A, \Pi)$ where :
– the quatruple $(Q, \Sigma, \phi, q_0 W^{(0)})$ is a k-PDA,
– A is a finite set called the *output alphabet*,
– the function $\Pi : Q\Sigma^* \to A$ is an admissible output function.

Remark 1.5. Notice that, for any k-PDA with output function $\mathcal{A} = (Q, \Sigma, \phi, q_0 W^{(0)}, A, \Pi)$ and for any letter a in A, the language $\{w \in [\![0, k-1]\!]^* \mid \Pi \circ \phi(w, q_0 W^{(0)}) = a\}$ is a context-free language, as it can be written as a (disjoint) union of context-free languages of type $\left\{w \in [\![0, k-1]\!]^* \mid qV \in \text{Pref}\left(\phi(w, q_0 W^{(0)})\right)\right\}$, for some fixed qV in $Q\Sigma$ and context-free languages of type $\{w \in [\![0, k-1]\!]^* \mid \Pi \circ \phi(w, q_0 W^{(0)}) = q\epsilon\}$, for some q in Q. Nervertheless, we cannot ensure the context-free languages obtained this way are deterministic or real-time deterministic as these two families are not closed under union.

2. A friendly family of k-context-free sequences

For an integer $n \in \mathbb{N}$, we denote by $\rho_k(n)$ its proper k-ary expansion, that is $\rho_k(n) = n_l \cdots n_1 n_0$ if $n = \sum_{i=0}^{l} n_i k^i$ with $n_i \in [\![0, k-1]\!]$ and $n_l \neq 0$. We also use concatenated notation for sequences.

Definition 2.1. A sequence $\mathbf{a} = a_0 a_1 a_2 \ldots$ over a finite alphabet A is called *k-context-free* if for all letters $a \in A$, the language $L_a(\mathbf{a}) = \{\rho_k(n) \mid a_n = a\}$ is context-free.

In [Ham98], the notion of k-quasi-context-free sequence is also introduced, as sequences $\mathbf{a} = a_0 a_1 a_2 \ldots$ for which at most one language among the languages $L_a(\mathbf{a})$ is not context-free.

Definition 2.2. A sequence $\mathbf{a} = a_0 a_1 a_2 \ldots$ over a finite alphabet A is called *k-pushdown automatic* if there exists a k-PDA with output function $\mathcal{A} = (Q, \Sigma, \phi, q_0 W^{(0)}, A, \Pi)$ such that, for all integers $n \geq 0$:

$$a_n = \Pi \circ \phi\left(\rho_k(n), q_0 W^{(0)}\right),$$

We say that the k-PDA $\mathcal{A}$ generates the sequence $\mathbf{a}$.

Example 2.3. Using the 2-PDA $\mathcal{A}$ presented in example 1.3 and the admissible ouput function Π detailed below :

$$\begin{array}{rccc} \Pi : & \{q_0, q_1, q_2\}\{s, t\}^* & \to & \{0, 1\} \\ & q_1\varepsilon & \mapsto & 1, \\ & qW \neq q_1\varepsilon & \mapsto & 0, \end{array}$$

we generate the following binary sequence :

$$\mathbf{d} = 1010^7 1010^{29} 10^5 1010^3 10^{113} 1010^5 1010^3 10^{17} 1010^5 1010^3 10 \ldots$$

whose digits can be characterized by :

$$d_n = 1 \iff \forall u \in \text{Pref}(\rho_2(n)),\ |u|_0 \leq |u|_1 \text{ and } |\rho_2(n)|_1 = |\rho_2(n)|_0.$$

Example 2.4. The following sequences $\mathbf{m}$ and $\mathbf{m}'$ defined respectively over the alphabet $\{0, 1\}$ and the alphabet $\{0, 1, 2\}$ by :

– $L_0(\mathbf{m}) = \{n \mid |\rho_2(n)|_1 = |\rho_2(n)|_0\}$,
– $L_1(\mathbf{m}) = \{n \mid |\rho_2(n)|_1 \neq |\rho_2(n)|_0\}$,

and

– $L_0(\mathbf{m}') = \{n \mid |\rho_2(n)|_1 = |\rho_2(n)|_0\}$,
– $L_1(\mathbf{m}') = \{n \mid |\rho_2(n)|_1 > |\rho_2(n)|_0\}$,
– $L_2(\mathbf{m}') = \{n \mid |\rho_2(n)|_1 < |\rho_2(n)|_0\}$,

are also 2-pushdown automatic.

$$\mathbf{m} = 010111111001011111111111111111111111010011001011110010111011111111111111111\ldots$$

$$\mathbf{m}' = 0101211120010111222121112111111222020012001011120010111011111222222212221\ldots$$

Indeed, these are generated by the 2-PDA $\mathcal{D} = (\{q\}, \{X, Y\}, \psi, q\varepsilon)$ where ψ is defined by :

$$\begin{array}{lll} \psi(0, q\varepsilon) = qY, & \psi(0, qX) = q\varepsilon, & \psi(0, qY) = qYY, \\ \psi(1, q\varepsilon) = qX, & \psi(1, qX) = qXX, & \psi(1, qY) = q\varepsilon, \end{array}$$

and using the ouput functions $\Pi_{\mathbf{m}}$ and $\Pi_{\mathbf{m}'}$ defined below :

$$\begin{array}{rccc} \Pi_{\mathbf{m}} : & \{q\}\{X, Y\}^* & \to & \{0, 1\} \\ & q\varepsilon & \mapsto & 0, \\ & qW \text{ with } W \neq \varepsilon & \mapsto & 1, \end{array}$$

$$\begin{array}{rccc} \Pi_{\mathbf{m}'} : & \{q\}\{X, Y\}^* & \to & \{0, 1, 2\} \\ & q\varepsilon & \mapsto & 0, \\ & qXW & \mapsto & 1, \\ & qYW & \mapsto & 2, \end{array}$$

From Remark 1.5, we easily get the following proposition :

Proposition 2.5. *A k-pushdown automatic sequence is a k-context-free sequence.*

A k-pushdown automatic sequence can also be constructed using some substitution and projection process. We will deal here with morphisms (or substitutions) over structured countable alphabets. The link between automata and substitution can easily be viewed considering substitutions through their graphs and considering PDA through their transitions graphs, as develloped in [MS85] and further in [Cau92, CK02, Cau03].

Substitutions involved in the following are substitution over a countable alphabet which can be structured as subsets of a free semigroup A^+, where A is a finite alphabet.

Notation 2.6. As we will use in the following both monoïds Σ^* and $(Q\Sigma^*)^*$, we denote by ϵ the empty word over $Q\Sigma^*$ while ε still denote the empty word over Σ.

Moreover, for some fixed finite alphabet A, we will have to distinguish the concatenation operator over the alphabet A and over the alphabet A^+. For two non empty words u_1 and u_2 over A the word u_1u_2 will denote as usual the concatenation of u_1 and u_2 considered as words over A, while the word $[u_1][u_2]$ will denote the concatenation of u_1 and u_2 considered as elements of A^+. The word u_1u_2 has length $|u_1| + |u_2|$ while the word $[u_1][u_2]$ has length two.

Notation 2.7. For any finite word $u = [u_0][u_1][\cdots][u_n]$ over the alphabet A^+ ($u_i \in A^+$) and any word v over a finite alphabet A', we define the word $u \star v$ over $A^+A'^*$ by $u \star v = [u_0v][u_1v][\cdots][u_nv]$.

Definition 2.8. Let A be a finite alphabet and let $S = A^+$.

We call *k-pushdown substitution* over the alphabet S a morphism σ from S to S^k defined for all $s \in S$ by $\sigma(s) = [\sigma_0(s)][\sigma_1(s)][\cdots][\sigma_k(s)]$ with $\sigma_i(s) \in S$ and satisfying the following condition :

$$\forall s = a_0a_1a_2 \cdots a_n \in A^{\geq 2},\ \sigma(s) = \sigma(a_0a_1) \star a_2 \cdots a_n. \tag{2}$$

so the substitution σ is completely given by the images of words of $A \cup A^2$.

Moreover, if there exists a word s_0 in S such that $\sigma_0(s_0) = s_0$, the k-pushdown substitution can be extended from s_0 and admits a fixed point $\mathbf{w}_\sigma = \sigma(\mathbf{w}_\sigma) = \sigma^\omega(s_0) = \lim_{n\to\infty} \sigma^n(s_0)$ which is an infinite sequence over A^+.

Remarks 2.9. These subsitutions are particular cases of subsitutions over countable alphabets [LG06a, Mau06] whose dynamical properties are studied in [Fer06]. Note that the notion of pushdown substitution can also be defined without the constant length hypothesis.

Note also that in the following, we may retrict the alphabet $S = A^+$ to a proper language over A if some words of S do not appears as letters in the fixed point.

Proposition 2.10. *A sequence is k-pushdown automatic if an only if it is the image by some admissible output function of an infinite fixed point of a k-pushdown substitution.*

Proof. Let $\mathcal{A} = (Q, \Sigma, \phi, q_0W^{(0)}, A, \Pi)$ be a k-PDA with output function generating a k-pushdown automatic sequence $\mathbf{a} = a_0a_1a_2\dots$.

We define over the alphabet $(Q\cup\{x\})\Sigma^*$, considered as a proper language of $(Q\cup\{x\}\cup\Sigma)^*$, the following k-pushdown substitution σ :

$$\forall i \in [\![0, k-1]\!], \forall qW \in Q\Sigma^*,\ \sigma_i(qW) = \phi(i, qW),$$

$$\sigma_0(x) = x \text{ and } \forall i \in [\![1, k-1]\!],\ \sigma_i(x) = \phi(i, q_0W^{(0)}),$$

$$\forall i \in [\![1, k-1]\!], \forall W \in \Sigma^+,\ \sigma_i(xW) = xW.$$

The added letter x ensures the existence of a word from which σ can be extended. This word $s_0 = x$ has almost the same behaviour under σ as the word corresponding to the initial configuration $q_0W^{(0)}$ does. Notice that the addition of such symbol x is not necessary if $\phi(0, q_0W^{(0)}) = q_0W^{(0)}$, by setting $s_0 = q_0W^{(0)}$.

As the k-pushdown substitution σ can be extended from s_0, it admits an infinite fixed point $\mathbf{w}_\sigma = \sigma(\mathbf{w}_\sigma) = [w_0][w_1][w_2][\dots$ with $w_0 = s_0$ and $w_i \in Q\Sigma^*$ for $i \neq 0$. As $\mathbf{w}$ is a fixed point of a substitution of length k, it satisfies the following property (see for example [LG06a, Mau06] for a proof) : For all positive integers n, if $\rho_k(n) = n_l \cdots n_1n_0$,

$$w_n = \sigma_{n_0} \circ \sigma_{n_1} \circ \cdots \circ \sigma_{n_l}(s_0).$$

As $\sigma_i(s_0) = \sigma_i(q_0W^{(0)})$ for all $i \neq 0$, then we have :

$$w_n = \sigma_{n_0} \circ \sigma_{n_1} \circ \cdots \circ \sigma_{n_l}(q_0W^{(0)}).$$

For the definition of σ we get finally :

$$w_0 = x \quad \text{and} \quad \forall n \in \mathbb{N}, n > 0,\ w_n = \phi(\rho_k(n), q_0W^{(0)}).$$

The output funtion Π of $\mathcal{A}$ is extended to à $(Q\cup\{x\})\Sigma^*$ by setting $\Pi(xW) = \Pi(q_0W)$ for all W of Σ^+ and $\Pi(x) = \Pi(q_0W^{(0)})$. As we keep the property that $\Pi(qW_0\dots W_L) = \Pi(qW_0)$ for all $qW \in (Q\cup\{x\})\Sigma^+$, the morphism Π is an admissible output function for $(Q\cup\{x\})\Sigma^*$ to A and we also have, for all non negative integer n, $a_n = \Pi(w_n)$.

Conversely, if $\mathbf{a} \in A^{\mathbb{N}}$ is the image by an admissible output function Π of a infinite fixed point $\mathbf{w}_\sigma = [m_0][m_1][m_2][\dots$ of a k-pushdown substitution σ over the alphabet $S = A'^+$, we construct a k-PDA with output admissible function $\mathcal{A} = (Q, \Sigma, \phi, q_0W^{(0)}, A, \Pi)$ associated to σ and Π by setting :

- $Q = \Sigma = A'$,
- $q_0W^{(0)} = w_0$,
- the transition function ϕ is defined as follows :

$$\forall qW \in Q\Sigma^*, \forall i \in [\![0, k-1]\!],\ \phi(i, qW) = \sigma_i(qW).$$

Then, for any non negative integer n such that $\rho_k(n) = n_l \cdots n_1n_0$, we have :

$$a_n = \Pi(m_n) = \Pi \circ \sigma_{n_0} \circ \sigma_{n_1} \circ \cdots \circ \sigma_{n_l}(q_0W^{(0)}),$$

and it comes :

$$a_n = \Pi(m_n) = \Pi\left(\phi(\rho_k(n), q_0W^{(0)})\right),$$

so the sequence $\mathbf{a}$ is k-pushdown automatic.

As a consequence of the proof of Proposition 2.10, we naturally define the *k-pushdown substitution associated to a k-PDA* $\mathcal{A} = (Q, \Sigma, \phi, q_0 W^{(0)})$ by considering the morphism σ from $(Q \cup \{x\})\Sigma^*$ to $((Q \cup \{x\})\Sigma^*)^*$ defined by :
- $\forall i \in [\![0, k-1]\!], \forall qW \in Q\Sigma^*,\ \sigma_i(qW) = \phi(i, qW)$,
- $\forall i \in [\![1, k-1]\!],\ \sigma_i(x) = \phi(i, q_0 W^{(0)})$ et $\sigma_0(x) = x$,
- $\forall i \in [\![1, k-1]\!], \forall W \in \Sigma^+,\ \sigma_i(xW) = xW$.

Example 2.11. The 2-pushdown substitution τ associated to the 2-PDA generating the sequences $\mathbf{m}$ and $\mathbf{m}'$ described in Example 2.4 can be defined over $\{x\} \cup q\{X, Y\}^*$ as follows :
- $\tau(x) = [x][qX]$,
- $\tau(q) = [qY][qX]$,
- and for all $W \in \{X, Y\}^*$, $\tau(qXW) = [qW][qX^2W]$ and $\tau(qYW) = [qY^2W][qW]$.

The beginning of its fixed point $\mathbf{w}_\tau$ is given by :

$$\mathbf{w}_\tau = [x][qX][q][qX^2][qY][qX][qX][qX^3][qY^2][q][q][qX^2][q][qX^2][qX^2][qX^4][qY^3][qY][qY][qX][qY][qX]\ldots$$

Using $\Pi_{\mathbf{m}}$ and $\Pi_{\mathbf{m}'}$ extended to $\{x\} \cup q\{X, Y\}^*$, we get easily $\Pi_{\mathbf{m}}(\mathbf{w}_\tau) = \mathbf{m}$ and $\Pi_{\mathbf{m}'}(\mathbf{w}_\tau) = \mathbf{m}'$.

This substitution is the same, up to a letter-to-letter correspondance, as the drunkenman substitution :

$$\begin{array}{rrcl} \tau' : & \mathbb{Z} \cup \{x\} & \rightarrow & \mathbb{Z}^2 \cup \{x1\} \\ & x & \mapsto & [x][1], \\ & n & \mapsto & [n-1][n+1], \end{array}$$

using the correspondance $qX^n \rightarrow n$ and $qY^n \rightarrow -n$. Dynamical properties of this substitution τ' are deeply studied in [Fer06].

3. Complexity of k-pushdown automatic sequences

Theorem 3.1.
Let $\mathbf{a}$ be a k-pushdown automatic sequence, generated by a k-PDA with stack alphabet Σ.
The complexity function of $\mathbf{a}$ satisfies :

$$p_{\mathbf{a}}(n) = \begin{cases} \mathcal{O}(n) & si\ \#\Sigma = 0, \\ \mathcal{O}(n \log^2 n) & if\ \#\Sigma = 1, \\ \mathcal{O}\left(n^{1+2\log_k \#\Sigma}\right) & if\ \#\Sigma \geq 2. \end{cases}$$

Remarks 3.2. This result extend the polynomial upper bound for complexity function of sequences considered in the paper of Y.Moshe [Mos08]. In some sense, it also extend the polynomial upper bound for complexity function of k^∞-automatic sequences in the sens of [LG06a]. Indeed, both k^∞-automatic sequences and k-pushdown automatic sequences can be constructed using as images by some function Π of the fixed point of some substitution σ over a countable alphabet but constraints are different :
- For k^∞-automatic sequences, the substitution $\sigma : A \rightarrow A^k$ has to be uniformly bounded : for any $a \in A$ and $i \in [\![0, k-1]\!]$, the cardinal of $\sigma_i^{-1}(\{a\})$ is bounded by some constant which is independant of a. This weak constraint enforce a strong constraint over the type of function Π : these function have to be constant except on a finite subset of A,
- For k-pushdown automatic sequences, the substitution must be a k-pushdown substitution, which is a stronger constraint as it forces a structure on the alphabet but also the form of the applications σ_i. As a counterpart, constraints over admissible output functions are weaker.

As a consequence, there exist k-pushdown automatic sequences which are not k^∞-automatic and there exist k^∞-automatic sequences which are not k-pushdown automatic. The family of k-pushdown automatic sequences which are also k^∞-automatic are k-pushdown automatic sequences gerenated by some k-PDA $(Q, \Sigma, \phi, q_0 W^{(0)}, A, \Pi)$ with admissible output function Π simulating the empty stack acceptance mode, that is Π is constant over $Q\Sigma^+$. This family contains binary sequences considered in [Mos08].

Proof. Let $\mathcal{A} = (Q, \Sigma, \phi, q_0 W^{(0)}, A, \Pi)$ be a k-PDA with output function generating $\mathbf{a} = a_0 a_1 a_2 \dots$. Let σ be the k-pushdown substitution associated to $\mathcal{A}$ and $\mathbf{w} = w_0 w_1 w_2 \dots$ its infinite fixed point.

Let also fix an integer $n \in [\![k^{i-1}, k^i - 1]\!]$ for some given integer $i \geq 1$.

By definition, for all integers n and m we have $a_m a_{m+1} \cdots a_{m+n-1} = \Pi(w_m w_{m+1} \cdots w_{m+n-1})$. Moreover, as $\mathbf{w}$ is the fixed point of σ, any word $w_m w_{m+1} \cdots w_{m+n-1}$ appears $\mathbf{w}$ as a subword of a word of type $\sigma^i(q_1 W^{(1)}) \sigma^i(q_2 W^{(2)})$.

It follows that $a_m a_{m+1} \cdots a_{m+n-1}$ is a subword of a word of type $\Pi \circ \sigma^i(q_1 W^{(1)}) \Pi \circ \sigma^i(q_2 W^{(2)})$ (which has length $2k^i$). Thus, to give an upper bound for $p_{\mathbf{a}}(n)$, we need to find an upper bound for the number of words $\Pi \circ \sigma^i(qW)$.

First, notice that for any $qW = qW_0 W_1 \cdots W_L$ in $Q\Sigma^{\geq i+1}$,

$$\sigma^i(qW) = \sigma^i(qW_0 W_1 \cdots W_{i-1}) \star W_i \cdots W_L.$$

As Π is an admissible output function, we get for any qW in $Q\Sigma^{\geq i+1}$,

$$\Pi \circ \sigma^i(qW) = \Pi \circ \sigma^i(q\mathrm{Pref}_{i+1}(W)).$$

Thus, in the worst case, we have as much words $\Pi \circ \sigma^p(qW)$ as words in $Q\Sigma^{\leq i+1}$ and a most $\left(\#(Q\Sigma^{\leq i+1})\right)^2$ words of type $\Pi \circ \sigma^i(q_1 W^{(1)}) \Pi \circ \sigma^i(q_2 W^{(2)})$. Then follows the upper bound of $p_{\mathbf{a}}(n)$:

$$p_{\mathbf{a}}(n) \leq k^i \left(\#(Q\Sigma^{\leq i+1})\right)^2,$$

as any single word $\Pi \circ \sigma^i(q_1 W^{(1)}) \Pi \circ \sigma^i(q_2 W^{(2)})$ can, at most produce k^i new words $a_m a_{m+1} \cdots a_{m+n-1}$.

According to the number of stack symbols, we get :

$$p_{\mathbf{a}}(n) \leq \begin{cases} k\left(\#Q\right)^2 n & \text{si } \#\Sigma = 0, \\ k\left(\#Q(i+2)\right)^2 n & \text{si } \#\Sigma = 1, \\ k\left(\#Q\frac{(\#\Sigma)^i - 1}{\#\Sigma - 1}\right)^2 n & \text{si } \#\Sigma \geq 2. \end{cases}$$

Using $i \leq \log_k n + 1$ and $(\#\Sigma)^i = k^{i \log_k \#\Sigma} \leq (kn)^{\log_k \#\Sigma}$ when $\#\Sigma \geq 2$, it follows :

$$p_{\mathbf{a}}(n) \leq \begin{cases} k\left(\#Q\right)^2 n & \text{si } \#\Sigma = 0, \\ k\left(\#Q\right)^2 n(\log_k n + 3)^2 & \text{si } \#\Sigma = 1, \\ k\left(\#Q\frac{(\#\Sigma)^2 (kn)^{\log_k \#\Sigma} - 1}{\#\Sigma - 1}\right)^2 n & \text{si } \#\Sigma \geq 2. \end{cases}$$

and this concludes the proof.

Remarks 3.3. The upper bounds given in Theorem 3.1 are optimal when $\#\Sigma = 0$ of course, but also when $\#\Sigma = 1$. Indeed, The complexity of the sequence $\mathbf{m}$ presented in Example 2.4 is asymptotically equivalent to $n(\log_2 n)^2$ [LG08b]. When $\#\Sigma \geq 2$, we still don't know if the given upper bound remains optimal. First examples, given in [LG08a] show up binary context-free sequences with complexity growth orders $n^{1+\log_k \#\Sigma}$.

Note that this result is non longer true for k-quasi-context-free sequences as there exist 2-quasi-context-free sequences with exponential complexity [Ham98].

4. Perspectives

Here follows a list of open problems on k-context-free sequences, related topics and perspective for further works.

The first open problem is to characterize context-free languages reached by pushdown automatic sequences and further, to compare it to context-free languages reached by context-free sequences. This topic is closely related to the study of sets of integers recognized by pushdown automata. If many works concerns sets of integers recognized by finite automata since the paper of Cobham [Cob72], only few papers [HS68, Sch68, Ber72, Ber73] deal with sets of integers recognized by pushdown automata since the paper of Minsky and Papert [MP66]. Let also mention works about the distribution in residue classes of integers with a fixed sum of digits [MS97, FM05, MPS05, Mau06], which are sets recognized by pushdown automata, providing k-context-free partitions of $\mathbb{N}$. It is also natural to consider the family of sequences generated by non real-time deterministic pushdown automata and to ask about a possible extension of Theorem 3.1 for these sequences.

The second open problem is the degeneration problem : Can we characterize context-free sequences or pushdown automatic sequences which are simply automatic, even perhaps in an other base ? This is related to a possible extension of the Cobham's theorem of base dependance [Cob72] for these classes of sequences. For this kind of problem, the substitutive point of view is probably the more convenient.

In this direction, it would also be interesting to study statistical properties of k-context-free and k-pushdown automatic sequences, as stating sufficient or necessary conditions for the existence of frequencies for letters and subwords. This is related to dynamical questions about their behaviour under the action of shift : What kind of dynamical properties (ergodicity, mixing...) for subshifts associated to k-context-free and k-pushdown automatic sequences ?

As mentionned in Remarks 2.9, pushdown substitutions can also be defined without the constant length hypothesis. Using the same process and the same notion of admissible output function, one can constructs the family of pushdown subsitutive sequences and ask about their complexity functions behaviours : Does the polynomial upperbound of Theorem 3.1 still hold ?

Références

[AB07] B. Adamczewski and Y. Bugeaud. On the complexity of algebraic numbers I. expansions in integer bases. *Annals of Math.*, 165 : 547–565, 2007.

[ABB97] J.-M. Autebert, J. Berstel, and L. Boasson. Context-free languages and pushdown automata. In *Handbook of formal languages, Vol. 1*, pages 111–174. Springer, 1997.

[All94] J.-P. Allouche. Sur la complexité des suites infinies. *Bull. Belg. Math. Soc. Simon Stevin*, 1(2) : 133–143, 1994.

[AS92] J.-P. Allouche and J. Shallit. The ring of k-regular sequences. *Theor. Comput. Sci.*, 98(2) : 163–197, 1992.

[AS03a] J.-P. Allouche and J. Shallit. *Automatic sequences. Theory, applications, generalizations.* Cambridge University Press, 2003.

[AS03b] J.-P. Allouche and J. Shallit. The ring of k-regular sequences, II. *Theor. Comput. Sci.*, 307(1), 2003.

[Aut87] J.-M. Autebert. *Langages algébriques.* Etudes et recherche en informatique. Masson, 1987.

[Ber72] J. Berstel. On sets of numbers recognized by push-down automata. In *13th IEEE Annual Symposium on Foundations of Computer Science (FOCS'72)*, pages 200–206. University of Maryland, USA, October 1972.

[Ber73] J. Berstel. Sur la densité asymptotique de langages formels. In *proceedings of International colloquium on Automata, Languages and Programming (ICALP'72)*, pages 345–358, 1973.

[Cas96] J. Cassaigne. Special factors of sequences with linear subword complexity. In *Developments in language theory (Magdeburg, 1995)*, pages 25–34. World Sci. Publishing, 1996.

[Cas97] J. Cassaigne. Complexité et facteurs spéciaux. *Bull. Belg. Math. Soc. Simon Stevin*, 4(1) : 67–88, 1997.

[Cau92] D. Caucal. On the regular structure of prefix rewriting. *Theor. Comp. Sci.*, 106 : 61–86, 1992.

[Cau03] D. Caucal. On infinite transition graphs having a decidable monadic theory. *Theor. Comp. Sci.*, 290 : 79–115, 2003.

[Cho56] N. Chomsky. Three models for the description of languages. *IRE Transactions on Information Theory*, 2 : 113–124, 1956.

[CK02] D. Caucal and T. Knapik. A chomsky-like hierarchy of infinite graphs. In *Proc. of the 27th MFCS, Warsaw*, volume 2420 of *Lecture Notes in Comput. Sci.*, pages 177–187. Springer, 2002.

[Cob72] A. Cobham. Uniform-tag sequences. *Math. Syst. Theory*, 6 : 164–192, 1972.

[ELR75] A. Ehrenfeucht, K. P. Lee, and G Rozenberg. Subword complexities of various classes of deterministic developmental languages without interactions. *Theor. Comp. Sci.*, 1 : 59–75, 1975.

[Fer99] S. Ferenczi. Complexity of sequences and dynamical systems. *Discrete Math.*, 206(1-3) : 145–154, 1999.

[Fer06] S. Ferenczi. Substitution dynamical systems on infinite alphabets. *Ann. Inst. Fourier*, 56(7) :2315–2343, 2006.

[FM05] E. Fouvry and C. Mauduit. Sur les entiers dont la somme des chiffres est moyenne. *J. Number THeory*, 114(1) : 135–152, 2005.

[Ham98] D. Hamm. Contribution to formal language theory : Fixed points, complexity, and context-free sequences. Master's thesis, University of Waterloo, 1998.

[HS68] J. Hartmanis and H. Shank. On the recognition of prime by automata. *Journal of the ACM*, 15 : 382–389, 1968.

[Kůr03] P. Kůrka. *Topological and symbolic dynamics*, volume 11 of *Cours spécialisés*. SMF, 2003.

[LG06a] M. Le Gonidec. Sur la complexité de mots infinis engendrés par des q-automates dénombrables. *Ann. Inst. Fourier*, 56(7) :2463–2491, 2006.

[LG06b] M. Le Gonidec. *Sur la complexité des mots q^∞-automatiques*. PhD thesis, Université Aix-Marseille II, 2006.

[LG08a] M. Le Gonidec. Complexity functions of infinite words associated with generalized Dyck languages. *Theoret. Comp. Sci.*, 407 : 117–133, 2008.

[LG08b] M. Le Gonidec. Drunken man infinite words complexity. *RAIRO-Theor. Inf. Appl.*, 42 : 599–613, 2008.

[Mau06] C. Mauduit. Propriétés arithmétiques des substitutions et automates infinis. *Ann. Inst. Fourier*, 56(7) : 2525–2549, 2006.

[Mos08] Y. Moshe. On some questions regarding k-regular and k-context-free sequences. *Theoret. Comp. Sci.*, 400 : 62–69, 2008.

[MP66] M. Minsky and S. Papert. Unrecognizable sets of numbers. *Journal of the ACM*, 13 : 281–286, 1966.

[MPS05] C. Mauduit, C. Pomerance, and A. Sàrközy. On the distribution in residue classes of integers with a fixed sum of digits. *Ramanujan J.*, 9(1-2) : 45–62, 2005.

[MS85] D. Muller and P. Schupp. The theory of ends, pushdown automata and second-order logic. *Theor. Comp. Sci.*, 37 : 51–75, 1985.

[MS97] C. Mauduit and A. Sárközy. On the arithmetic structure of the integers whose sum of digits is fixed. *Acta Arithmetica*, LXXXI(2) : 145–173, 1997.

[Pan85] J.-J. Pansiot. Complexité des facteurs des mots infinis engendrés par morphismes itérés. *Lecture Notes in Comp. Sci.*, 172 : 380–389, 1985. Automata, languages and programming (Antwerp, 1984).

[PF02] N. Pytheas Fogg. *Substitutions in dynamics, arithmetics and combinatorics*, volume 1794 of *Lect. Notes in Math.* Springer, 2002. Edited by V. Berthé, S. Ferenczi, C. Mauduit and A. Siegel.

[Rig00] M. Rigo. Generalization of automatic sequences for numeration systems on a regular language. *Theoret. Comp. Sci.*, 244 : 271–281, 2000.

[Sch68] M. P. Schüzenberger. A remark on acceptable sets of numbers. *Journal of the ACM*, 15 : 300–303, 1968.

[Sta06] L. Staiger. Kolmogorov complexity of infinite words. *CDMTCS Research Report Series*, 279, 2006.