



HAL
open science

Context-Free Sequences

Didier Caucal, Marion Le Gonidec

► **To cite this version:**

Didier Caucal, Marion Le Gonidec. Context-Free Sequences. Theoretical Aspects of Computing – ICTAC 2014, Gabriel Ciobanu, Dominique Méry Sep 2014, Bucharest, Romania. pp.259 - 276, 10.1007/978-3-319-10882-7_16 . hal-01450071

HAL Id: hal-01450071

<https://hal.science/hal-01450071>

Submitted on 31 Jan 2017

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.

Context-free sequences

Didier Caucal¹ and Marion Le Gonidec²

¹ CNRS, LIGM-Université Paris-Est
caucal@univ-mlv.fr,

² LIM, Université de la Réunion
marion.le-gonidec@univ-reunion.fr

Abstract. A sequence w over a finite alphabet A is generated by a uniform automaton if there exists an automaton labelled on $\{0, \dots, k-1\}$ for some $k > 1$ and recognizing for each output a in A the set of positions of a in w expressed in base k . Automatic sequences are generated by finite automata. By considering pushdown automata instead of finite ones, we generate exactly the context-free sequences. We distinguish the subfamilies of unambiguous, deterministic, real-time deterministic context-free sequences associated with the corresponding families of pushdown automata. We study the closure under shift, product, morphisms, inverse substitutions and various extractions of these four families of context-free sequences. Additionally, we show that only using multiplicatively dependent bases yields the same set of context-free sequences.

1 Introduction

Automatic sequences are well known objects appearing in many area of theoretical computer science and mathematics [Co 72, AS 03]. To generate these sequences, one can use finite k -automata. Fixing an integer k greater than 1, a k -automaton G has states labelled over an alphabet A and edges labelled on $\llbracket k \rrbracket = \{0, \dots, k-1\}$ so that the family of languages $L(G, a)$ recognized by G from an initial state to a state labelled a in A forms a partition of the set of words over $\llbracket k \rrbracket$ labelling paths in G . A k -automaton G generates the sequence w over A when $L(G, a)$ is the set of positions of a in w properly expressed in base k .

When G is finite, the sequence w is k -automatic. Finite automata are also well known devices which recognize regular languages [Kl 56].

As automatic sequences can be characterized in several other ways (using regular languages, uniform morphisms or their kernel finiteness property), there are different ways to extend this notion: context-free sequences, morphic sequences, regular sequences (see [AS 03] for example).

One can also extend this notion using a larger class of automata, the most natural one being, following Chomsky's hierarchy [Ch 56], k -pushdown automata (pushdown automata over $\llbracket k \rrbracket$). It turns out that this generalization of automatic sequences, also yields to context-free sequences defined in [Ha 98], [Mo 08] and [AS 03] (open problem 6.3, p 208): a sequence w is k -context-free if and only if

for each letter a , the set of occurrences in base k of a in w is a context-free language.

Contrary to finite automata, we obtain distinct sub-families of sequences by considering only unambiguous, deterministic, or real-time deterministic k -pushdown automata. These (proper) sub-families are respectively named unambiguous, deterministic, real-time deterministic k -context-free sequences.

The closure properties of these four families of sequences are also different. This fact underlying the different consequences of the considered transformations (synchronization product, shift, uniform morphism, inverse injective k -uniform substitution, regular extractions) over the unambiguity and the determinism of involved languages and automata. They sometimes share the same closure properties (Propositions 12 and 14) or not (Propositions 13 and 15).

Moreover, tools used to prove these properties depend on the considered family: while, for context-free sequences and unambiguous context-free sequences, closure properties follow from easy considerations over languages, for deterministic context-free sequences and real-time deterministic context-free sequences one has to use tools from infinite graph theory. Indeed, for these two last subfamilies, classes of languages associated to involved pushdown automata no longer have suitable closure properties and the usual internal representation of k -pushdown automata is not really suitable to compute automaton transformations induced by the transformations of sequences we consider. Then, we choose to favor their external representation (by their transition graphs) using the equivalent notion of regular k -automata [MS 85].

Finally, we investigate base dependence properties of context-free sequences.

2 Generating sequences with automata

We recall the general notion of automaton, and the particular cases of unambiguous and deterministic automata. The languages recognized by automata over the alphabet $\{0, \dots, k-1\}$ of $k > 1$ digits, called k -automata, can be seen as sets of natural numbers expressed in base k . This in turn allows such automata to define sequences.

2.1 Automata and languages terminology

Let A be an *alphabet* *i.e.*, a finite set of symbols called *letters*. A *word* u over A of *length* $n \geq 0$ is a mapping from the set $\llbracket n \rrbracket = \{0, \dots, n-1\}$ into A which is denoted by the n -tuple $(u(0), \dots, u(n-1)) \in A^n$ or simply by $u = u_0 \cdots u_{n-1}$ where $u_i = u(i)$ for each $i \in \llbracket n \rrbracket$; we write $|u|$ the length n of u . The word of length 0, *i.e.*, the 0-tuple $()$ is the *empty word* denoted by ε . The set A^* of words over A is the free monoid $(A^*, \cdot, \varepsilon)$ generated by A with the *concatenation* operator \cdot which can be omitted: $u_0 \cdots u_{m-1} \cdot v_0 \cdots v_{n-1} = u_0 \cdots u_{m-1} v_0 \cdots v_{n-1}$.

Any subset of A^* is a *language* over A . The family 2^{A^*} of languages over A is a semiring $(2^{A^*}, \cup, \cdot, \emptyset, \{\varepsilon\})$ for the *language concatenation* defined by $L \cdot M = \{u \cdot v \mid u \in L \wedge v \in M\}$ for any $L, M \subseteq A^*$. The closure under concatenation of a

language L is the language $L^* = \bigcup_{n \geq 0} L^n$ for $L^n = \{ u_1 \dots u_n \mid u_1, \dots, u_n \in L \}$ the concatenation n -times of L . Recall that a language is a *regular language* if it can be obtained from \emptyset and the elementary languages $\{a\}$ with $a \in A$ by finite application of the operations $\cup, \cdot, *$. The regular languages are also the languages recognized by finite automata.

We fix a symbol ι . An *automaton* over A and an alphabet C of *colours* (output alphabet) with $\iota \notin C$, is a directed graph G whose each edge is labelled by a letter in A and whose vertices can be labelled by letters in $C \cup \{\iota\}$:

$$G \subseteq V \times A \times V \cup (C \cup \{\iota\}) \times V \text{ for some (possibly infinite) set } V.$$

A triple $(s, a, t) \in G$ is an *edge* labelled by a from *source* s to *target* t ; it is identified with the transition $s \xrightarrow{a}_G t$ or directly $s \xrightarrow{a} t$ if G is understood. A pair $(c, s) \in G$ or directly $cs \in G$ means that s is labelled by c . A vertex can be uncoloured. The set of *vertices* of G is

$$V_G = \{ s \mid \exists c (cs \in G) \} \cup \{ s \mid \exists a, t (s \xrightarrow{a}_G t \vee t \xrightarrow{a}_G s) \}.$$

So G is a finite automaton if and only if its vertex set V_G is finite. An *input vertex* is a vertex s labelled by ι : $\iota s \in G$.

The *induced automaton* $G|_P$ of an automaton G to a set P is the restriction of G to its vertices in P :

$$G|_P = \{ (s, a, t) \in G \mid s, t \in P \} \cup \{ (c, s) \in G \mid s \in P \}.$$

The *inverse* G^{-1} of an automaton G is the automaton

$$G^{-1} = \{ (t, a, s) \mid (s, a, t) \in G \} \cup \{ (c, s) \mid (c, s) \in G \}.$$

The *out-degree* $d_G^+(s) = |G \cap \{s\} \times A \times V_G|$ and the *in-degree* $d_G^-(s) = d_{G^{-1}}^+(s)$ of a vertex s are the number of edges of respectively source and target s ; the *degree* of s is $d(s) = d^+(s) + d^-(s)$. An automaton is of *finite (resp. in-, out-) degree* if any vertex is of finite (resp. in-, out-) degree. A finite degree automaton with finitely many vertex degrees is of *bounded degree*.

Any tuple $(s_0, a_1, s_1, \dots, a_n, s_n) \in (VA)^*V$ for $n \geq 0$ with $s_0 \xrightarrow{a_1}_G s_1, \dots, s_{n-1} \xrightarrow{a_n}_G s_n$ is a *path* from s_0 to s_n labelled by $u = a_1 \dots a_n$; we write $s_0 \xrightarrow{u}_G s_n$ or directly $s_0 \xrightarrow{u} s_n$ if G is understood; we say that s_n is *accessible* from s_0 and also write $s_0 \xrightarrow{*} s_n$ if we do not want to specify a path label. An *accessible automaton* G means that any vertex is accessible from an input vertex: $\forall s \in V_G \exists r \in V_G (\iota r \in G \wedge r \xrightarrow{*} s)$.

We say that G is *deterministic* if it has a unique input vertex and there are no two edges with the same source and the same label: $(r \xrightarrow{a} s \wedge r \xrightarrow{a} t) \implies s = t$.

We also say that G is *unambiguous* if there is no couple of distinct paths labelled with the same word from the set of input vertices to the set of coloured vertices.

We abbreviate unambiguous by ‘una.’ and deterministic by ‘det.’.

The *language recognized* by an automaton G (from ι) to $c \in C$

$$L(G, c) = \{ u \in A^* \mid \exists s, t \in V_G (s \xrightarrow{u}_G t \wedge \iota s, ct \in G) \}$$

is the set of words labelling the paths from an input vertex to a vertex coloured by c . Kleene’s theorem [Kl 56] states that a language is regular if and only if it is recognized by a finite (resp. and det.) automaton.

A *colouring* over (A, C) is a mapping $\Lambda : C \rightarrow 2^{A^*}$. To any automaton G ,

we associate the colouring Λ_G defined for any $c \in C$ by $\Lambda_G(c) = L(G, c)$. We say that a colouring Λ is a *partition* of $P \subseteq A^*$ if $\bigcup_{c \in C} \Lambda(c) = P$ and $\Lambda(c) \cap \Lambda(d) = \emptyset$ for any $c \neq d$ in C .

2.2 Uniform automata and sequences

A *sequence* w over A is a mapping from the set \mathbb{N} of nonnegative integers into A denoted by $w = (w_n)_{n \geq 0}$. Let A^ω be the set of sequences over A .

We fix an integer $k > 1$. Recall that $\llbracket k \rrbracket = \{0, \dots, k-1\}$ and we denote

$$\llbracket k \rrbracket^\diamond = \llbracket k \rrbracket^* - 0.\llbracket k \rrbracket^*$$

the set of words over $\llbracket k \rrbracket$ not beginning with 0.

The (proper) *representation in base k* of any $n \in \mathbb{N}$ is the word

$$(n)_k = n_0 \cdots n_r \in \llbracket k \rrbracket^\diamond \quad \text{such that} \quad \sum_{i=0}^r n_i k^{r-i} = n$$

Conversely to any word $u \in \llbracket k \rrbracket^*$, we associate the integer

$$[u]_k = \sum_{i=0}^{|u|-1} u_i k^{|u|-i-1}.$$

So for any $n \geq 0$ and $u \in \llbracket k \rrbracket^*$, we have $[(n)_k]_k = n$ and $([u]_k)_k$ is the greatest suffix of u in $\llbracket k \rrbracket^\diamond$. For any $w \in A^\omega$, we define the k -colouring Λ_w^k associating with any $a \in A$ the language

$$\Lambda_w^k(a) = \{ u \in \llbracket k \rrbracket^\diamond \mid w_{[u]_k} = a \} = \{ (n)_k \mid w_n = a \}$$

of the representations in base k of the positions of a in w .

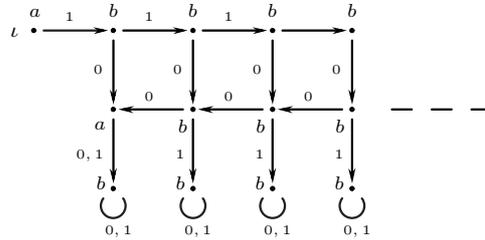
So the colouring Λ_w^k is a partition of $\llbracket k \rrbracket^\diamond$.

Definition 1. A k -automaton over A is an accessible automaton G on the set $\llbracket k \rrbracket$ of edge labels and on the set A of colours such that Λ_G is a partition of $\llbracket k \rrbracket^\diamond$. We talk about uniform automaton when k and A are not specified.

Any vertex of a uniform automaton has at most one colour. For a det. uniform automaton, each vertex has a unique colour and its input vertex has no ingoing edge. A non-det. uniform automaton can have several paths with same labels starting from distinct input vertices and leading to vertices possibly coloured but with the same colour.

Definition 2. A k -automaton G generates the sequence $\text{Seq}(G) \in A^\omega$ if for any integer n , the n -th letter $\text{Seq}(G)_n$ is the colour a such that $(n)_k \in \Lambda_G(a)$, i.e., $\Lambda_{\text{Seq}(G)}^k = \Lambda_G$.

Example 3. Let G be the following det. 2-automaton over $\{a, b\}$:



The automaton G generates the sequence $w = \text{Seq}(G)$ characterized by:

$$\Lambda_w(a) = \{ 1^n 0^n \mid n \geq 0 \} \quad \text{and} \quad \Lambda_w(b) = \{0, 1\}^\circ - \Lambda_w(a),$$

that is $w_n = a$ if and only if there exists p in \mathbb{N} such that $n = 2^p(2^{p+1} - 1)$.

3 Context-free sequences

The k -automatic sequences can be characterized using languages. Precisely, a sequence w is a k -automatic sequence if the colouring Λ_w^k only contains regular languages. In [Ha 98] and [AS 03] (open problem 6.3, p 208), authors start considering the sequences for which the colouring Λ_w^k is made of context-free languages [AU 79].

Definition 4. *A sequence w over A is a k -context-free sequence if $\Lambda_w^k(a)$ is context-free for all $a \in A$.*

Moreover, a sequence w is a k -automatic sequence if and only if it is generated by a finite k -automaton $G: w = \text{Seq}(G)$ [Co 72]. It is natural to expect that k -context-free sequences are sequences generated by pushdown automata over $\llbracket k \rrbracket$ (with no 0-edges starting from input vertices), called k -pushdown automata, as these machines recognize context-free languages. As the usual internal representation of k -pushdown automata is not really convenient to play with automata transformations involved by transformations of sequences we will consider. So we use the equivalent notion (via their transition graphs) of regular k -automata [MS 85] and we show that k -context-free sequences are sequences generated by regular k -automata (See Proposition 6).

3.1 Regular automata

Regular automata, which form a family of infinite automata (including the finite automata), is the set of finitely decomposable automata.

Let us make precise the notion of decomposition of an automaton G according to a *graduation* which is a mapping $g: V_G \rightarrow \mathbb{N}$ such that only finitely many vertices have the same value by $g: g^{-1}(n)$ is finite for every $n \geq 0$.

Let $\text{Con}(G, g, n)$ be the set of connected components of $G|_{\{s \mid g(s) \geq n\}}$ for any $n \geq 0$. In particular $\text{Con}(G, g, 0) = \{G\}$ for G connected.

We complete elements in $\text{Con}(G, g, n)$ to obtain a partition $\text{Dec}(G, g, n)$ of $G|_{\{s \mid g(s) \geq n\}} \cup \{s \xrightarrow{a}_G t \mid g(s) > n \vee g(t) > n\}$ as follows:

$$\text{Dec}(G, g, n) = \bigcup_{H \in \text{Con}(G, g, n)} H \cup \left\{ s \xrightarrow{a}_G t \mid \bigvee \begin{array}{l} (s \in V_H \wedge g(s) > n) \\ (t \in V_H \wedge g(t) > n) \end{array} \right\}$$

The information added in $\text{Dec}(G, g, n)$ consists in specifying the way(s) a graph of $\text{Con}(G, g, n)$ can be linked with others connected components in G . This additional information is useful when G contains at least a vertex of infinite degree.

The finiteness of the *decomposition* $\text{Dec}(G, g) = \bigcup_{n \geq 0} \text{Dec}(G, g, n)$ is expressed by isomorphism respecting the frontiers. The *frontier* $\text{Fr}_G(H)$ of $H \subseteq G$

is the set of vertices common to H and $G - H$ i.e.,

$$\text{Fr}_G(H) = V_H \cap V_{G-H}.$$

We say that $H, K \subseteq G$ are *strongly isomorphic* if there exists a bijection $h : V_H \rightarrow V_K$ preserving edges, colouring and frontiers:

$$h(H) = K \text{ and } h(\text{Fr}_G(H)) = \text{Fr}_G(K).$$

An automaton G is *finitely decomposable* by a graduation g if

$$\text{Dec}(G, g) \text{ has finitely many non-strongly-isomorphic automata.}$$

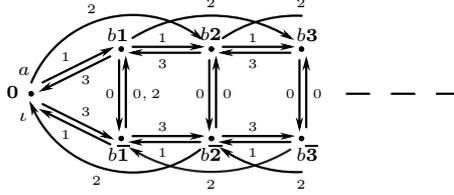
Finally G is a *regular automaton* if G is finitely decomposable by some graduation.

In particular, a regular automaton has finitely many non-isomorphic connected components but notice that there exists a non-regular automaton G with a graduation g giving a decomposition $\text{Dec}(G, g)$ with finitely many non-isomorphic components. Moreover, a regular automaton of finite degree is of bounded degree.

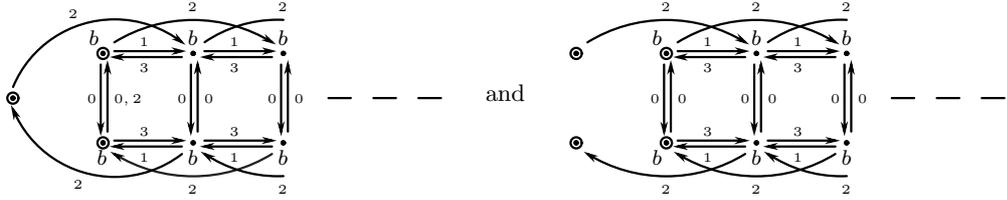
Example 5. Let us consider the following 4-automaton of vertex set \mathbb{Z} :

$$G = \{ n \xrightarrow{0} -n \mid n \in \mathbb{Z} - \{0\} \} \cup \{ n \xrightarrow{i} n+i \mid n \in \mathbb{Z}, i \in \{1, 2\} \} \\ \cup \{ n \xrightarrow{3} n-1 \mid n \in \mathbb{Z} \} \cup \{ \iota 0 \} \cup \{ a 0 \} \cup \{ bn \mid n \neq 0 \}$$

This is a 4-automaton which is represented (with vertices in bold faces, with $-n$ denoted by \bar{n}) as follows:



Such an automaton G is regular because it is finitely decomposable by the graduation $g(n) = |n|$ of the absolute value. In fact, $\text{Dec}(G, g)$ has only three non-strongly-isomorphic automata: G and the following two automata where the vertices of the frontier are circled:



For any connected automaton G of finite degree and with finitely many input vertices i.e., $\{ s \mid \iota s \in G \}$ is finite, a standard graduation is the *distance* from the input vertices: for any vertex $s \in V_G$:

$$\text{Dist}_G(s) = \min \{ |u| \mid \exists r (\iota r \in G \wedge r \xrightarrow{u} G \cup G^{-1} s) \}$$

The decomposition $\text{Dec}(G, \text{Dist}, n)$ of any automaton G at distance $n \geq 0$ is the set of connected components obtained by removing in G all the vertices at distance from ι less than n . For the automaton presented in Example 5, we have $\text{Dist}_G(n) = \left\lceil \frac{|n|}{2} \right\rceil$ for any $n \in \mathbb{Z}$.

The regular k -automata are isomorphic to the transition graphs of k -pushdown automata (ε -transitions in their transition graphs being removed by gluing vertices, and with regular set of configurations labelled by each colour) [MS 85]. So the languages recognized by the family of regular automata are the context-free languages. Moreover, det. (resp. det. and of finite degree, resp. una.) regular k -automata are isomorphic to transition graphs of det. (resp. real-time det., resp. una.) k -pushdown automata, so these classes of regular k -automata recognizing respectively the det. context-free languages, the real-time det. context-free languages, and una. context-free languages of $\llbracket k \rrbracket^\diamond$ [Ca 07]. We abbreviate real-time by ‘rt.’.

3.2 Regular automata and context-free sequences

We extend the characterization of automatic sequences by finite automata to context-free sequences using regular k -automata.

Proposition 6. *A sequence is a k -context-free sequence if and only if it is generated by a regular k -automaton $G : w = \text{Seq}(G)$.*

Proof. Let w be a k -context-free sequence. As context-free languages over $\llbracket k \rrbracket$ are recognized by pushdown automata over $\llbracket k \rrbracket$, for any letter $a \in A$, $A_w^k(a) = L(G_a, a)$ for some regular k -automaton G_a having a unique colour a . Notice that in general, the usual product of two regular automata is not regular. As automata G_a can be chosen with distinct vertex sets, w is generated by the disjoint finite union $\cup_{a \in A} G_a$ which remains a regular k -automaton.

For a sequence generated by a regular k -automaton G , the language $A_w(a)$ with $a \in A$ is recognized by the regular k -automaton obtained from G by removing colours $c \neq a$, so $A_w(a)$ is context-free. \square

It is natural to introduce the sub-families of k -context-free sequences associated with the different classes of regular automata/pushdown automata presented at the end of Section 3.1.

Definition 7. *A sequence is a una. (resp. det., rt. det.) k -context-free sequence if it is generated by a una. (resp. det., det. and of finite degree) regular k -automaton.*

Let us note $\text{RtDetCf}_k(A^\omega)$, $\text{DetCf}_k(A^\omega)$, $\text{UnaCf}_k(A^\omega)$, $\text{Cf}_k(A^\omega)$ for the respective four families of rt. det., det., una., k -context-free sequences over A .

For any $k > 1$, we have

$$\text{RtDetCf}_k(A^\omega) \subsetneq \text{DetCf}_k(A^\omega) \subsetneq \text{UnaCf}_k(A^\omega) \subsetneq \text{Cf}_k(A^\omega)$$

The strict inclusions follow from these on context-free languages and we can refine Proposition 6.

Proposition 8. For any $w \in A^\omega$ and $k > 1$,

w is a (resp. una.) k -context-free sequence

$\iff \Lambda_w^k(a)$ is (resp. una.) context-free for any $a \in A$.

Furthermore

w is a (resp. rt.) det. k -context-free sequence

$\implies \Lambda_w^k(a)$ is (resp. rt.) det. context-free for any $a \in A$.

For the unambiguous case, the proof of Proposition 6 holds, as if all automata G_a are una., the disjoint union $\cup_{a \in A} G_a$ remains una. The second implication comes from the fact that families of det. and rt. det. regular automata recognize respectively det. and rt. det. families of context-free languages [Ca 07]. The converse of the implication is true for $|A| = 2$ (because the complement of a det. context-free language is also context-free) but false in general (see Proposition 16).

Notice also that the characterization of k -automatic sequences as morphic sequences obtained with k -uniform morphism [Co 72] can be extended to the smallest subfamily $\text{RtDet}_k(A^\omega)$ of k -context-free sequences [LeG 12] using a notion of context-free morphic sequence which extend the usual notion of morphic sequence [Lo 05].

4 Closure properties

The set of k -automatic sequences is stable by various transformations: regular modifications of letters, shift, application of a uniform morphism, inverse substitution, various extractions. It underlies the robustness of this concept, and one can ask whereas these properties remain true or not for the four families of k -context-free sequences.

In this section, we present closure properties of $\text{Cf}_k(A^\omega)$ and $\text{DetCf}_k(A^\omega)$ but these properties and their proofs remain valid by replacing $\text{Cf}_k(A^\omega)$ by $\text{UnaCf}_k(A^\omega)$ and by replacing $\text{DetCf}_k(A^\omega)$ by $\text{RtDetCf}_k(A^\omega)$.

By Proposition 6, we get closure properties of $\text{Cf}_k(A^\omega)$ and $\text{UnaCf}_k(A^\omega)$ from the closure properties of context-free languages. We get closure properties of $\text{DetCf}_k(A^\omega)$ and $\text{RtDetCf}_k(A^\omega)$ from the preservation of deterministic regular automata by inverse regular path functions.

4.1 An important tool: regular path functions

Let us recall the notion of a path function.

We define the set Exp of *regular expressions* as the smallest language over

$$C \cup A \cup \{\varepsilon, (,), ^{-1}, \neg, \vee, \wedge, \cdot, ^+\}$$

such that $C \cup A \cup \{\varepsilon\} \subseteq \text{Exp}$ and for any u, v in Exp , the expressions (u^{-1}) , $(\neg u)$, $(u \vee v)$, $(u \wedge v)$, $(u \cdot v)$, (u^+) are in Exp .

We can remove parentheses using the associativity of \vee , \wedge , \cdot and by assigning priorities to operators as usual. Finally \cdot can be omitted. The expression u^* corresponds to $\varepsilon \vee u^+$ and we will use A instead of $\bigvee_{a \in A} a$.

A *finite expression* is a regular expression without the operator $+$ and we denote by FinExp the set of finite expressions.

The label $a \in A$ of an edge $s \xrightarrow{a}_G t$ from s to t of an automaton G is extended to a regular expression $u \in \text{Exp}$ by induction on the length of u : for any $c \in C$ and $u, v \in \text{Exp}$,

$$\begin{aligned} s \xrightarrow{c} t & \text{ if } s = t \wedge cs & ; & s \xrightarrow{\varepsilon} t & \text{ if } s = t \\ s \xrightarrow{u^{-1}} t & \text{ if } t \xrightarrow{u} s & ; & s \xrightarrow{\neg u} t & \text{ if } \neg (s \xrightarrow{u} t) \\ s \xrightarrow{u \vee v} t & \text{ if } s \xrightarrow{u} t \vee s \xrightarrow{v} t & ; & s \xrightarrow{u \wedge v} t & \text{ if } s \xrightarrow{u} t \wedge s \xrightarrow{v} t \\ s \xrightarrow{uv} t & \text{ if } \exists r (s \xrightarrow{u} r \wedge r \xrightarrow{v} t) & ; & s \xrightarrow{u^+} t & \text{ if } s \xrightarrow{(\neg u)^+} t. \end{aligned}$$

A regular expression formalizes a path pattern in a graph.

Example 9. For instance, if u and v are words over A and c in C ,

$s \xrightarrow{\varepsilon \wedge u} t$ means that there is a cycle on vertex s labelled by u ;

$s \xrightarrow{\varepsilon \wedge uu^{-1}} t$ means that $s = t$ and there is a path labelled by u starting from s ;

$s \xrightarrow{c} t$ or $s \xrightarrow{c} s$ means that the vertex $s = t$ is coloured by c ;

$s \xrightarrow{(\varepsilon \wedge ucu^{-1})v} t$ means that there is a path labelled by u starting from s and ending to a vertex coloured with c and a path from s to t labelled by v .

$s \xrightarrow{A^*cA^*} t$ means that there is a path labelled from s to t which goes through a vertex colored by c .

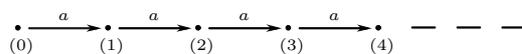
A function $h : C \cup A \rightarrow \text{Exp}$ (resp. FinExp) is called a *regular* (resp. *finite*) *path function* and is applied by inverse to any automaton G to get the automaton:

$$h^{-1}(G) = \{ s \xrightarrow{a} t \mid a \in A \wedge s \xrightarrow{h(a)}_G t \} \cup \{ cs \mid c \in C \wedge s \xrightarrow{h(c)}_G s \}.$$

That is, the graph $h^{-1}(G)$ is obtained by replacing any path of type $h(a)$ from s to t by an edge $s \xrightarrow{a} t$ and colouring of a vertex s by c if a path $h(c)$ loops on s .

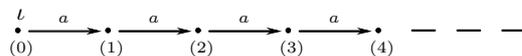
Let us give an example to illustrate the notion of inverse regular path function.

Example 10. For instance we take the following automaton $G = \{ n \xrightarrow{a} n+1 \mid n \geq 0 \}$ depicted as follows:



and the finite path function h defined by $h(a) = a$ and $h(\iota) = \neg(a^{-1}a)$.

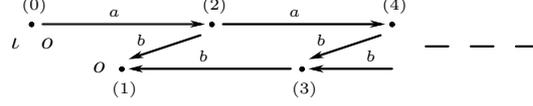
So $h^{-1}(G)$ is the following automaton:



By applying to this automaton by inverse the following regular path function g :

$$\begin{aligned}
g(\iota) &= \iota \quad ; \quad g(o) = \iota \vee a^{-1} \iota a \quad ; \\
g(a) &= (\varepsilon \wedge (a^{-1})^* \iota (aa)^*) a a \quad ; \\
g(b) &= (\varepsilon \wedge (a^{-1})^* \iota (aa)^*) a^{-1} \vee (\varepsilon \wedge (a^{-1})^* \iota a (aa)^*) a^{-1} a^{-1} .
\end{aligned}$$

we get the following automaton $g^{-1}(h^{-1}(G))$ depicted as follows:



The regularity of automata and the finiteness of its degree are preserved by inverse of regular path function under conditions [CK 01].

Proposition 11. *Let G be a regular automaton and h a regular path function.*

1. *If $h^{-1}(G)$ is deterministic or of finite degree, then $h^{-1}(G)$ is regular.*
2. *If the degree of G is finite and h is a finite path function, then $h^{-1}(G)$ is regular and of finite degree.*

Notice that $h^{-1}(G)$ is deterministic if it has a unique input vertex and for all a in A and s in V_G , $s \xrightarrow{h(a)}_G t \wedge s \xrightarrow{h(a)}_G t' \implies t = t'$.

4.2 Synchronization product and shift

Let us start with a preservation result by regularly modifying letters. Taking a mapping $*$: $A \times A \rightarrow A$, the *synchronization product* $w * w'$ of sequences $w, w' \in A^\omega$ is the sequence $w * w' = (w_n * w'_n)_{n \geq 0}$.

As the intersection of two rt. det. context-free languages can be context-sensitive but not context-free, the (resp. det.) k -context-freeness of sequences is not preserved by synchronization product. We have to restrict one sequence to be automatic and use the fact that the synchronization product of a (resp. det.) regular automaton with a finite automaton remains a (resp. det.) regular automaton [Ca 07].

Proposition 12. *The families $\text{Cf}_k(A^\omega)$ and $\text{DetCf}_k(A^\omega)$ are closed under synchronization product with any k -automatic sequence.*

In particular, the (resp. det.) k -context-freeness of a sequence is preserved by modifying finitely many letters.

The (*left*) *shift* of a sequence $w = (w_n)_{n \geq 0}$ is the sequence $S(w) = (w_n)_{n > 0}$ obtained from w by removing its first letter. The (*right*) *shift* of a sequence w by a letter a is the sequence aw , with $(aw)_0 = a$ and $(aw)_n = w_{n-1}$ for $n > 0$. The shift operations preserve k -automaticity.

Proposition 13. *The family $\text{Cf}_k(A^\omega)$ is closed under left and right shifts.*

The family $\text{DetCf}_k(A^\omega)$ is not closed under left and right shifts.

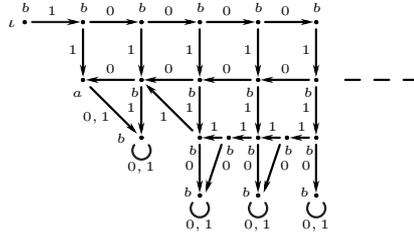
Proof. We check this proposition for the left shift operation S . Let $w \in \text{Cf}_k(A^\omega)$. For each $a \in A$, $\Lambda_w^k(a)$ is a context-free language. The transformation $R : (n)_k \rightarrow (n-1)_k$ is realized by the following finite transducer:

$$\begin{aligned} & \{ p \xrightarrow{1/\varepsilon} r \} \cup \{ p \xrightarrow{i/i} q \mid 0 \leq i < k \} \cup \{ q \xrightarrow{i/i} q \mid 0 \leq i < k \} \\ & \cup \{ q \xrightarrow{i/i-1} r \mid 0 < i < k \} \cup \{ r \xrightarrow{0/k-1} r \} \end{aligned}$$

of input state p and of final state r . Hence, the language $\Lambda_{S(w)}^k(a) = R(\Lambda_w^k(a))$ is context-free for each $a \in A$, that is $S(w) \in \text{Cf}_k(A^\omega)$.

Note that this transducer also preserves unambiguous context-freeness.

Furthermore let us consider the following deterministic regular 2-automaton G :



So $w = \text{Seq}(G) \in \text{RtDetCf}_2(\{a, b\}^\omega)$ and the language

$$\Lambda_w^2(a) = \{11\} \cup \{10^{m+n+1}10^n1^{2m}0 \mid m, n \geq 0\}$$

is det. context-free. For the shifted sequence $S(w)$, we have:

$$\begin{aligned} \Lambda_{S(w)}^2(a) &= \{10\} \cup \{10^{n+2}1^{n+1} \mid n \geq 0\} \\ &\cup \{10^{m+n+1}10^n1^{2m-1}01 \mid m > 0, n \geq 0\} \end{aligned}$$

which is not a deterministic context-free language since the language

$$\Lambda_{S(w)}^2(a) \cap (10^*1^+ + 10^*1^+01) = \{10^{n+1}1^n \mid n > 0\} \cup \{10^{n+1}1^{2n}01 \mid n > 0\}$$

is not deterministic context-free [Yu 89]. So $S(w)$ is not in $\text{DetCf}_k(A^\omega)$. \square

4.3 Morphisms and inverse substitutions

A p -uniform morphism over an alphabet A is a function $\sigma : A \rightarrow A^p$, extended to A^* and A^ω by concatenation of images.

Proposition 14. *The families $\text{Cf}_k(A^\omega)$ and $\text{DetCf}_k(A^\omega)$ are closed under any finite p -uniform morphism with $p > 0$.*

Proof. Let $w \in \text{Cf}_k(A^\omega)$ and $\sigma : A \rightarrow A^p$. Let us show that $\sigma(w) \in \text{Cf}_k(A^\omega)$. For each $a \in A$, $\Lambda_w^k(a)$ is a context-free language and we have to check that $\Lambda_{\sigma(w)}^k$ remains a context-free colouring. By denoting \tilde{u} the mirror of a (finite) word u , the relation $R = \{(u, v) \in \llbracket k \rrbracket^* \times \llbracket k \rrbracket^* \mid [\tilde{v}]_k = p \times [\tilde{u}]_k\}$ is recognized by the following finite transducer [Be 72]:

$$\begin{aligned} & \{ r \xrightarrow{q/t} s \mid r, s \in \llbracket p \rrbracket, q, t \in \llbracket k \rrbracket, pq + r = ks + t \} \\ & \cup \{ r \xrightarrow{\varepsilon/[r]_k} p \mid r \in \llbracket p \rrbracket \} \cup \{ p \xrightarrow{\varepsilon/0} p \} \end{aligned}$$

of input state 0 and of final state p . So for any context-free language $L \subseteq \llbracket k \rrbracket^*$, $p \times L = \{ (p \times [u]_k)_k \mid u \in L \} = \widetilde{R(\widetilde{L})} \cap \llbracket k \rrbracket^\diamond$ is a context-free language.

By applying $r \geq 0$ right shifts, $p \times L + r = \{ (p \times [u]_k + r)_k \mid u \in L \}$ remains context-free (see proof of Proposition 13). Also, note that this transformation $L \mapsto p \times L + r$ preserves unambiguous context-freeness.

To conclude the proof for $\text{Cf}_k(A^\omega)$, it remains to see that for any $a \in A$,

$$\Lambda_{\sigma(w)}^k(a) = \bigcup_{b \in A, \sigma(b)_r = a} p \times \Lambda_w^k(b) + r.$$

For the family $\text{DetCf}_k(A^\omega)$, we cannot use the same argument as the family of det. context-free languages is not stable by mirror image. However, this transformation of context-free sequences is done by inverse of a finite path function on associated det. regular automata.

Let $w \in A^\omega$ generated by a det. regular k -automaton G . Let $\sigma : A \rightarrow A^p$ be a p -uniform morphism.

Let r the initial vertex of G and $G_0 = G \cup \{r \xrightarrow{0} r\}$ (which is also a regular automaton). For any $s \in V_{G_0}$ and $i \in \llbracket k \rrbracket$, we denote si the i -th successor of s : $s \xrightarrow{i}_{G_0} si$.

We start by copying p times each vertex of the automaton G_0 : we take new symbols $0', \dots, (p-1)'$ and to any vertex $s \in V_{G_0}$ coloured by $a \in A$, we associate new vertices $s_{|0'}, s_{|1'}, \dots, s_{|(p-1)'}$ respectively colored by letters $\sigma(a)_0, \sigma(a)_1, \dots, \sigma(a)_{p-1}$ and linked to s by edges $s \xrightarrow{i'} s_{|i'}$ in order to complete G_0 into the following automaton:

$G' = G_0 \cup \{ s \xrightarrow{i'} s_{|i'} \mid s \in V_{G_0} \wedge i \in \llbracket p \rrbracket \} \cup \{ \sigma(a)_i s_{|i'} \mid a s \in G_0 \wedge i \in \llbracket p \rrbracket \}$. So G' remains a deterministic regular automaton.

By linking the vertices $s_{|i'}$ for $s \in V_G$ and $i \in \llbracket p \rrbracket$ with a finite path function h , we will construct a new k -automaton $h^{-1}(G')$ generating $\sigma(w)$.

Let r be the input vertex of G .

In $h^{-1}(G')$, for any $s \in V_G$, the kp successors of $s_{|0'}, \dots, s_{|(p-1)'}$ are

$$s0_{|0'}, \dots, s0_{|(p-1)'}, s1_{|0'}, \dots, s1_{|(p-1)'}, \dots, s(k-1)_{|0'}, \dots, s(k-1)_{|(p-1)'},$$

where the k first are successors of $s_{|0'}$ ordered by increasing index of edges $i \in \llbracket k \rrbracket$, the following k are successors of $s_{|1'}$ ordered by increasing index of edges $i \in \llbracket k \rrbracket$ and so on.

Formally, we can define the finite path function h as follows. We denote

$$[i, j] = ki + j \text{ for any } i \in \llbracket p \rrbracket \text{ and } j \in \llbracket k \rrbracket$$

and the Euclidian division of any integer $n \geq 0$ by p is denoted by

$$n = pq_n + r_n \text{ with } 0 \leq r_n < p.$$

We define h by

$$\text{for any } a \in A, h(a)=a \text{ and } h(\iota) = (0')^{-1}\iota 0',$$

$$\text{for any } j \in \llbracket k \rrbracket, j \neq 0, h(j) = \bigvee_{i \in \llbracket p \rrbracket} (i')^{-1} \cdot q_{[i,j]} \cdot (r_{[i,j]})'.$$

$$\text{and } h(0) = (0')^{-1} \cdot (-\iota)0 \cdot (0')' \vee \bigvee_{i \in \llbracket p \rrbracket, i \neq 0} (i')^{-1} \cdot q_{[i,0]} \cdot (r_{[i,0]})'.$$

For instance for $p = 3$ and $k = 2$, the table of $q_{[i,j]}, r_{[i,j]}$ for $i \in \llbracket 3 \rrbracket$ and $j \in \llbracket 2 \rrbracket$ is given by:

$i \setminus j$	0	1
0	0, 0	0, 1
1	0, 2	1, 0
2	1, 1	1, 2

hence

$$\begin{aligned} h(0) &\equiv (0')^{-1} \cdot (\neg \iota) \cdot 0 \cdot 0' \vee (1')^{-1} \cdot 0 \cdot 2' \vee (2')^{-1} \cdot 1 \cdot 1' \\ h(1) &\equiv (0')^{-1} \cdot 0 \cdot 1' \quad \vee (1')^{-1} \cdot 1 \cdot 0' \vee (2')^{-1} \cdot 1 \cdot 2' \end{aligned}$$

By Propositions 11, $h^{-1}(G')$ is a det. regular k -automaton. The det. regular k -automaton $h^{-1}(G')$ generates $\sigma(w)$. \square

Let us apply p -uniform morphism by inverse on a sequence: starting from position 0 each sequence of length p is replaced by a letter. A *substitution* σ on a set V is a mapping $V \rightarrow 2^{V^*}$ and it is a p -uniform substitution for $p \geq 0$ if $\sigma(s) \subseteq V^p$ for every $s \in V$. A p -uniform substitution σ is *total* if $\bigcup_{s \in V} \sigma(s) = A^p$. A substitution σ is *injective* if $\sigma(s) \cap \sigma(t) = \emptyset$ for any $s, t \in V$ with $s \neq t$. We apply by inverse an injective and total p -uniform substitution σ on any sequence $w \in A^\omega$ to get the sequence $\sigma^{-1}(w)$ defined by:

$$\sigma^{-1}(w) = (\sigma^{-1}(w_{np} \dots w_{(n+1)p-1}))_{n \geq 0}.$$

Proposition 15. *The family $\text{DetCf}_k(A^\omega)$ is closed under inverse of any injective and total k^p -uniform substitution with $p \geq 0$.*

The family $\text{Cf}_k(A^\omega)$ is not closed under inverse of injective total k -uniform substitutions.

Proof. Let $w = \text{Seq}(G)$ for some det. k -automaton G .

Let $p \geq 0$ and σ be an injective and total k^p -uniform substitution.

Let $G' = G \cup \{r \xrightarrow{0} r\}$ for $\iota r \in G$. So G' remains deterministic and regular. Let h be the finite path function on G' which renames colours as follows: the initial vertex does not change $h(\iota) = \iota$ and for any $a \in A$, a vertex s is relabelled by a if there exists a word u in $\sigma(a)$ such that for any $v \in \llbracket k \rrbracket^p$, the v -path from s ends to a vertex coloured by the $[v]_k$ -th letter of u , that is, formally:

$$h(a) = \bigvee_{u \in \sigma(a)} \bigwedge_{v \in \llbracket k \rrbracket^p} (\varepsilon \wedge v u_{[v]_k} v^{-1}) \text{ where } (v)^{-1} \text{ stands for } v_p^{-1} \dots v_2^{-1} v_1^{-1}$$

Finally, the 0-loop on the initial vertex is removed and labels on edges are unchanged: $h(0) = (\neg \iota)0$ and $h(i) = i$ for any $0 < i < k$.

Since h is a regular path function, $h^{-1}(G')$ is a det. regular k -automaton and we have $\text{Seq}(h^{-1}(G')) = \sigma^{-1}(w)$.

Let us check the non-closure of $\text{Cf}_k(A^\omega)$ under the inverse of an injective k -substitution.

We consider the following real-time deterministic context-free languages:

$$L = 1^+ \cdot \{ 0^n 1^n \mid n > 0 \} \quad \text{and} \quad M = \{ 1^n 0^n \mid n > 0 \} \cdot 1^+$$

The language $L \cdot 0 \cup M \cdot 1$ is context-free but not deterministic [Yu 89].

Let $a, b \in A$. We define the 2-context-free sequence $w \in \{a, b\}^\omega$ by

$$\Lambda_w^k(a) = L \cdot 0 \cup M \cdot 1 \quad \text{and} \quad \Lambda_w^k(b) = \llbracket k \rrbracket^\diamond - \Lambda_w^k(a).$$

The letter a appears in w in positions $[1^m 0^n 1^n 0]_k$ or $[1^n 0^n 1^m]_k$ for $m, n > 0$. Moreover, the pattern aa appears in w in positions $[1^n 0^n 1^n 0]_k$ for $n > 0$.

Let σ be the k -uniform substitution

$$\sigma(a) = aab^{k-2} \quad \text{and} \quad \sigma(b) = A^k - \sigma(a)$$

which is total and injective. As $\Lambda_{\sigma^{-1}(w)}^k(a) = L \cap M = \{1^n 0^n 1^n \mid n > 0\}$ is not context-free, $\sigma^{-1}(w)$ is not a k -context-free sequence. \square

A consequence of this last proposition is that the converse of the implication of Proposition 8 is false:

Proposition 16. *There exists a sequence w such that $\Lambda_w^k(a)$ is a rt. det. context-free language for all a , and w is not a deterministic k -context-free sequence.*

Proof. We use the same languages L, M as in the proof of Proposition 15. Let $A = \{a, \bar{a}, b, \bar{b}, c\}$ be a five letters alphabet. We define $w \in A^\omega$ by

$$\Lambda_w^k(a) = L \cdot 0 \quad ; \quad \Lambda_w^k(\bar{a}) = (\llbracket k \rrbracket^\diamond - (L \cup \{\varepsilon\})) \cdot 0 \quad ;$$

$$\Lambda_w^k(b) = M \cdot 1 \quad ; \quad \Lambda_w^k(\bar{b}) = (\llbracket k \rrbracket^\diamond - M) \cdot 1 \quad ; \quad \Lambda_w^k(c) = \llbracket k \rrbracket^\diamond \{2, \dots, k-1\} \cup \{\varepsilon\}.$$

For any $a \in A$, the language $\Lambda_w^k(a)$ is a rt. det. context-free language and the pattern ab only appears in w in positions $[1^n 0^n 1^n 0]_k$ for $n > 0$.

Let σ be the k -substitution defined by $\sigma(a) = abc^{k-2}$ and $\sigma(b) = A^k - \sigma(a)$.

As $\Lambda_{\sigma^{-1}(w)}^k(a) = L \cap M$, the sequence $\sigma^{-1}(w)$ is not k -context-free.

By Proposition 15, w cannot be a deterministic k -context-free sequence. \square

4.4 1-context-free sequences and extractions of ultimately periodic subsequences

The k -automatic sequences are characterized by the finiteness of their k -kernels. The k -kernel $K_k(w)$ of a sequence $w \in A^\omega$ is the set of subsequences $k_u(w)$ for $u \in \llbracket k \rrbracket^*$ obtained by only picking up letters in positions of type $[vu]_k$, for $v \in \llbracket k \rrbracket^\diamond$, that is, $k_u(w) = (w_{k|u|n+[u]_k})_{n \geq 0}$.

Proposition 17. *The k -kernel of any (resp. det.) k -context-free sequence only contains (resp. det.) k -context-free sequence.*

Proof. Let $i \in \llbracket k \rrbracket$ and $w \in \text{Cf}_k(A^\omega)$. As we have $k_\varepsilon(w) = w$ and $k_{uv}(w) = k_v(k_u(w))$ for any $u, v \in \llbracket k \rrbracket^*$, we just need to check that $k_i(w) \in \text{Cf}_k(A^\omega)$.

As $k_i(w) = (w_{kn+i})_{n \geq 0}$, the language $\Lambda_{k_i(w)}^k(a) = \Lambda_w^k(a)i^{-1}$ is the right residual by i of $\Lambda_w^k(a)$, for any $a \in A$, thus context-free. It follows that $k_i(w)$ is a k -context-free sequence.

Now assume that $w \in \text{DetCf}_k(A^\omega)$. We define the injective and total k -substitution h by $h(a) = A^{i-1}aA^{k-i}$ for any $a \in A$. Hence by Proposition 15, $k_i(w) = h^{-1}(w)$ so $k_i(w) \in \text{DetCf}_k(A^\omega)$. \square

To generate the ultimately periodic sequences, we extend the definition of a k -automaton to the case $k = 1$.

Definition 18. A 1-automaton over A is an accessible automaton G on the set $\llbracket 1 \rrbracket = \{0\}$ of edge labels and on the set A of colours such that Λ_G is a partition of $\{0\}^*$.

A 1-automaton G generates the sequence $\text{Seq}(G) \in A^\omega$ such that for any $n \geq 0$, its n -th letter $\text{Seq}(G)_n$ is the colour a such that $0^n \in \Lambda_G(a)$.

The 1-colouring Λ_w^1 of any $w \in A^\omega$ is the mapping associating with any $a \in A$ the language $\Lambda_w^1(a) = \{0^n \mid w_n = a\}$ of the representations in base 1 of the positions of a in w .

Definition 19. A sequence w over A is 1-automatic (resp. 1-context-free) if $\Lambda_w^1(a)$ is a regular (resp. context-free) for all a in A .

Lemma 20. For any $w \in A^\omega$, the following four statements are equivalent:

- a) w is a ultimately periodic sequence,
- b) w is a 1-automatic sequence,
- c) w is a 1-context-free sequence,
- d) $\Lambda_w^1(a)$ is a regular language for any $a \in A$.

Proof. The equivalences a) \iff b) \iff d) are well known (see [AS 03] for example) and b) \implies c) because any finite automaton is regular.

To show c) \implies d), Let $w = \text{Seq}(G)$ for some regular 1-automaton G . For any $a \in A$, $\Lambda_w^1(a) = \Lambda_G(a)$ is a context-free language over $\{0\}$. By Parikh's lemma, every context-free language over a unique letter is regular. \square

We can extract ultimately periodic sequences in any k -context-free sequence by picking k -regularly letters.

Proposition 21. Let w be a k -context-free sequence and $u, v_1, \dots, v_p \in \llbracket k \rrbracket^*$ with $p \geq 1$, $uw_1 \in \llbracket k \rrbracket^\diamond$ and $v_1, \dots, v_p \neq \varepsilon$. For any $q \geq 0$ and $i \in \{1, \dots, p\}$, we denote $v^{q+\frac{i}{p}} = (v_1 \dots v_p)^q v_1 \dots v_i$.

The sequence $\left(w_{\lfloor \frac{nw}{p} \rfloor_k} \right)_{n \geq 0}$ is ultimately periodic.

Proof. We have $w = \text{Seq}(G)$ for some regular k -automaton G .

Let h be the finite path function on G defined as follows:

$$h(\iota) = u^{-1} \iota u \quad ; \quad h(a) = a \quad \text{for any } a \in A \quad ; \quad h(i) = v_i \quad \text{for any } 1 \leq i \leq p.$$

So $h^{-1}(G)$ is a prefix-recognizable automaton (got from the complete binary tree by inverse regular path functions).

We take the following finite deterministic automaton:

$$H = \{ i \xrightarrow{i} i+1 \mid 1 \leq i < p \} \cup \{ p \xrightarrow{p} 1 \} \cup \{ \iota 1 \} \cup \{ a i \mid a \in A \wedge 1 \leq i \leq p \}$$

The following synchronisation product of $h^{-1}(G)$ and H

$$h^{-1}(G) \times H = \{ (s, p) \xrightarrow{0} (t, q) \mid \exists i (s \xrightarrow{i}_{h^{-1}(G)} t \wedge p \xrightarrow{i}_H q) \} \\ \cup \{ a(s, p) \mid a s \in h^{-1}(G) \wedge a p \in H \}$$

is a deterministic prefix-recognizable automaton, hence by [CK 01] is a regular automaton.

Let K be the restriction by accessibility from ι of $h^{-1}(G) \times H$.

So K is a regular 1-automaton generating $\text{Seq}(K) = \left(w_{[uv^{\frac{n}{p}}]_k} \right)_{n \geq 0}$.

By Lemma 20, $\text{Seq}(K)$ is an ultimately periodic sequence. \square

4.5 About base dependence

A famous theorem of Cobham states that sequences which are automatic in two multiplicatively independent bases are the ultimately periodic ones [Co 69]. This section presents two results in the direction of a possible extension of this statement for context-free sequences. First, the context-freeness is preserved for any non-null power of the base (but not the deterministic context-freeness). Second, we have the same set of context-free sequences only for multiplicatively dependent bases.

Proposition 22. *For every $k, p > 1$, we have*

$$\text{Cf}_k(A^\omega) = \text{Cf}_{k^p}(A^\omega) \quad \text{and} \quad \text{DetCf}_k(A^\omega) \subsetneq \text{DetCf}_{k^p}(A^\omega).$$

Proof. Let $h : \llbracket k^p \rrbracket \rightarrow \llbracket k \rrbracket^p$ be the bijective mapping associating with any $n \in \llbracket k^p \rrbracket$ its $(n+1)$ -th word of $\llbracket k \rrbracket^p$ by (length) lexicographic order:

$$h(n) = 0^{p - \lceil \log_k(n+1) \rceil} (n)_k \quad \text{for every } n \in \llbracket k^p \rrbracket$$

which is extended by morphism on $\llbracket k^p \rrbracket^*$. For any $w \in A^\omega$ and $a \in A$, we have

$$\Lambda_w^k(a) = (0^*)^{-1} h(\Lambda_w^{k^p}(a)) \cap \llbracket k \rrbracket^\diamond \quad \text{and} \quad \Lambda_w^{k^p}(a) = h^{-1}(0^* \Lambda_w^k(a)) \cap \llbracket k^p \rrbracket^\diamond$$

We deduce that $\text{Cf}_k(A^\omega) = \text{Cf}_{k^p}(A^\omega)$.

Let $w = \text{Seq}(G)$ for some regular deterministic k -automaton G . We complete G by adding a 0-loop to its input vertex: $G' = G \cup \{ r \xrightarrow{0} r \}$ for $\iota r \in G$. The mapping h is extended to a finite path function by adding its behaviours on colours $h(\iota) = \iota$ and $h(a) = a$ for any $a \in A$. Let $H = h^{-1}(G') - \{ r \xrightarrow{0} r \}$. We have $\text{Seq}(H) = \text{Seq}(G)$ and by Proposition 11, H is a regular deterministic k^p -automaton. Thus $\text{DetCf}_k(A^\omega) \subseteq \text{DetCf}_{k^p}(A^\omega)$.

Let us check that this inclusion is strict. As $k, p > 1$, we have $k^p \geq 4$ and

$$h(0) = 0^p \quad ; \quad h(1) = 0^{p-1}1 \quad ; \quad h(k) = 0^{p-2}10 \quad ; \quad h(k^{p-1}) = 10^{p-1}$$

Let $a, b \in A$. We define the sequence $w \in \{a, b\}^\omega$ by

$$\Lambda_w^{k^p}(a) = 1\{ 0^{n+1}(k^{p-1})^{2n} \mid n \geq 0 \} \cup k\{ 0^n 1^{n+1} \mid n \geq 0 \}, \\ \Lambda_w^{k^p}(b) = \llbracket k^p \rrbracket^\diamond - \Lambda_w^{k^p}(a).$$

So $w \in \text{DetCf}_{k^p}(A^\omega)$ and

$$\Lambda_w^k(a) = 1\{ (0^p)^{n+1}(10^{p-1})^{2n} \mid n \geq 0 \} \cup 10\{ (0^p)^n(0^{p-1}1)^{n+1} \mid n \geq 0 \} \\ = \{ 10^{pn+p}(10^{p-1})^{2n} \mid n \geq 0 \} \cup \{ 10^{pn+p}(10^{p-1})^n 1 \mid n \geq 0 \}$$

which is not a deterministic context-free language, so $w \notin \text{DetCf}_k(A^\omega)$. \square

Proposition 23. For any $p, q > 1$,

$$\text{Cf}_p(A^\omega) = \text{Cf}_q(A^\omega) \iff \exists i, j > 0, p^i = q^j.$$

Proof. \Leftarrow : This implication is straightforward from Proposition 22.

\Rightarrow : We just transpose the construction given in [Be 72] for the context-free integer sets.

Let $a, b \in A$. We define the sequence $w \in \{a, b\}^\omega$ by

$$\Lambda_w^p(a) = 10^* \text{ and } \Lambda_w^p(b) = \llbracket p \rrbracket^\diamond - 10^*.$$

So $w \in \text{Cf}_p(A^\omega) = \text{Cf}_q(A^\omega)$ thus $\Lambda_w^q(a)$ is an infinite context-free language.

As a corollary of the pumping lemma on context-free languages, there exists u, v, x, y, z in $\llbracket q \rrbracket^*$ such that $u \neq \varepsilon$ and for every $n \geq 0$, $xu^nyv^nz \in \Lambda_w^q(a)$.

The integer mapping f defined for every $n \geq 0$ by $[xu^nyv^nz]_q = p^{f(n)}$ is increasing.

Note that for any $s, t \in \llbracket q \rrbracket^*$, $[st]_q = [s]_q q^{|t|} + [t]_q$, hence for any $n \geq 0$,

$$\begin{aligned} [s^nt]_q &= [s]_q q^{|s^{n-1}t|} + \dots + [s]_q q^{|st|} + [s]_q q^{|t|} + [t]_q \\ &= [s]_q q^{|t|} (1 + q^{|s|} + [t]_q + \dots + (q^{|s|})^{n-1}) \\ &= [s]_q q^{|t|} \frac{q^{n|s|} - 1}{q - 1} + [t]_q \end{aligned}$$

Thus for any $n \geq 0$, $p^{f(n)} = [xu^nyv^nz]_q = Aq^{n|uv|} + Bq^{n|v|} + C$ with

$$A = q^{|yz|} \left([x]_q + \frac{[u]_q}{q-1} \right) ; B = q^{|z|} \left([y]_q + \frac{[v]_q - q^{|v|} [u]_q}{q-1} \right) ; C = [z]_q - \frac{q^{|z|} [v]_q}{q-1}$$

hence

$$\frac{p^{f(n+1)}}{p^{f(n)}} \sim \frac{Aq^{(n+1)|uv|}}{Aq^{n|uv|}} = q^{|uv|} \text{ i.e., } \lim_{n \rightarrow \infty} p^{f(n+1)-f(n)} = q^{|uv|}.$$

This last equality on integers implies that there exists n_0 such that

$$p^{f(n_0+1)-f(n_0)} = q^{|uv|},$$

meaning that p and q are multiplicatively dependent. \square

5 Conclusion and open problems

Let us mention again that the results of Section 4 remain valid when substituting UnaCf_k for Cf_k , and RtDetCf_k for DetCf_k .

The difference of behaviours of these families of context-free sequences under transformations and the difference of involved tools (from languages or from graph theory) also allows to deeper understand from where come the strong robustness of automatic sequences, for which concepts of unambiguity of languages and determinism of automata are totally erased.

Some properties of k -context-free sequences have to be further studied, for instance the structure of their k -kernels, properties of symbolic dynamical systems associated with these sequences, their degenerated cases (how to decide whenever a k -context-free sequence is k -automatic, periodic, etc.). Moreover, results of Section 4.5 are encouraging for a possible extension of the Cobham's theorem on base dependence.

We expect that the most of presented closure properties extend to similar constructions of sequences using indexed languages [Ah 68] and higher order indexed languages following Maslov's hierarchy of languages [Ma 74]. As regular and context-free languages are the first two level of this hierarchy, this paper is a second step towards a theory of the infinite hierarchy of higher order indexed automatic sequences following Maslov's hierarchy, the automatic sequences and context-free automatic sequences being the first two levels.

References

- [Ah 68] A. AHO, *Indexed grammars - an extension of context-free grammars*, Journal of the Association for Computing Machinery 15(4), 647–671 (1968).
- [AU 79] A. AHO and J. ULLMAN, *Introduction to automata theory, languages and computation*, Addison-Wesley, 418 pages (1979).
- [AS 03] J.-P. ALLOUCHE and J. SHALLIT, *Automatic sequences: theory, applications, generalizations*, Cambridge University Press, 588 pages (2003).
- [Be 72] J. BERSTEL, *On sets of numbers recognized by push-down automata*, 13th Symposium on Switching and Automata Theory (SWAT - FOCS), IEEE, 200–206 (1972).
- [Ca 90] D. CAUCAL, *On the regular structure of prefix rewriting*, 15th CAAP, LNCS 431, A. Arnold (Ed.), 87–102 (1990) or in Theoretical Computer Science 106, 61–86 (1992).
- [Ca 07] D. CAUCAL, *Deterministic graph grammars*, Texts in Logic and Games 2, Amsterdam University Press, J. Flum, E. Grädel, T. Wilke (Eds.), 169–250 (2007).
- [CK 01] D. CAUCAL and T. KNAPIK, *On internal presentation of regular graphs by prefix-recognizable graphs*, Theory of Computing Systems 34-4, 299–336 (2001).
- [Ch 56] N. CHOMSKY, *Three models for the description of language*, IRE Transactions on Information Theory 2, 113-124 (1956).
- [Co 72] A. COBHAM, *Uniform-tag sequences*, Mathematical Systems Theory 6, 164–192 (1972).
- [Co 69] A. COBHAM, *On the base-dependence of sets of numbers recognizable by finite automata*, Mathematical Systems Theory 3, 186–192 (1969).
- [Ha 98] D. HAMM, *Contributions to formal language theory: Fixed points, complexity, and context-free sequences*, M.Sc. Thesis, University of Waterloo (1998).
- [Kl 56] S. KLEENE, *Representation of events in nerve nets and finite automata. Automata studies*, Annals of Mathematics Studies, Princeton University Press, 3–41 (1956).
- [LeG 08] M. LE GONIDEC, *Drunken man infinite words complexity*, RAIRO - Theoretical Informatics and Applications 42, 599–613 (2008).
- [LeG 12] M. LE GONIDEC, *On the complexity of a family of k -context-free sequences*, Theoretical Computer Science 44-1, 47–54 (2012).
- [Lo 05] M. LOTHAIRE, *Applied Combinatorics on Words*, Cambridge University Press (2005).
- [Ma 74] A. MASLOV, *The hierarchy of indexed languages of an arbitrary level*, Doklady Akademii Nauk SSSR 217, 1013–1016 (1974).
- [Mo 08] Y. MOSHE, *On some questions regarding k -regular and k -context-free sequences*, Theoretical Computer Science 400, 62–69 (2008).
- [MS 85] D. MULLER and P. SCHUPP, *The theory of ends, pushdown automata, and second-order logic*, Theoretical Computer Science 37, 51–75 (1985).
- [Yu 89] S. YU, *A pumping lemma for deterministic context-free languages*, Information Processing Letters 31-1, 47–51 (1989).