Theory of Computing Systems (2024) 68:1622–1639
https://doi.org/10.1007/s00224-024-10197-5

Automatic Abelian Complexities of Parikh-Collinear Fixed Points

Michel Rigo[1] · **Manon Stipulanti**[1] · **Markus A. Whiteland**[2]

Accepted: 16 September 2024 / Published online: 10 October 2024

Abstract
Parikh-collinear morphisms have the property that all the Parikh vectors of the images of letters are collinear, i.e., the associated adjacency matrix has rank 1. In the conference DLT–WORDS 2023 we showed that fixed points of Parikh-collinear morphisms are automatic. We also showed that the abelian complexity function of a binary fixed point of such a morphism is automatic under some assumptions. In this note, we fully generalize the latter result. Namely, we show that the abelian complexity function of a fixed point of an arbitrary, possibly erasing, Parikh-collinear morphism is automatic. Furthermore, a deterministic finite automaton with output generating this abelian complexity function is provided by an effective procedure. To that end, we discuss the constant of recognizability of a morphism and the related cutting set.

Keywords Parikh-collinear morphism · Recognizable morphism · Automatic sequence · Abelian complexity · Substitution shift · Automated theorem proving

1 Introduction

This paper is an extension of the results in our previous work [1] that was presented during the joint DLT–WORDS 2023 conference. The main objects of interest are fixed

✉ Markus A. Whiteland
m.a.whiteland@lboro.ac.uk

Michel Rigo
m.rigo@uliege.be

Manon Stipulanti
m.stipulanti@uliege.be

[1] Department of Mathematics, ULiège, Allée de la Découverte 12, Liège 4000, Belgium

[2] Department of Computer Science, Loughborough University, Epinal Way, Loughborough, Leicestershire LE11 3TU, United Kingdom

points of Parikh-collinear morphisms which are defined as follows. It is assumed that the alphabet $A = \{a_1 < \cdots < a_k\}$ is ordered and $\Psi(w)$ denotes the *abelianization* or *Parikh vector* $(|w|_{a_1}, \ldots, |w|_{a_k})$ counting the number of different letters constituting the word $w \in A^*$. A morphism $f\colon A^* \to B^*$ is *Parikh-collinear* if the Parikh vectors $\Psi(f(b))$, $b \in A$, are collinear (or pairwise $\mathbb{Z}$-linearly dependent).

Parikh-collinear morphisms have received some attention in recent years. The authors of [2, Sec. 4] list a dozen of fixed points of Parikh-collinear morphisms appearing in the OEIS [3], e.g., A285249. Cassaigne et al. characterized Parikh-collinear morphisms as those morphisms that map all words to words with bounded abelian complexity [4]. These morphisms also provide infinite words with interesting properties with respect to the so-called k-binomial equivalence $\sim_k$. Two words $u, v \in A^*$ are *k-binomially equivalent* if $\binom{u}{x} = \binom{v}{x}$, for all $x \in A^*$ with $|x| \leq k$. Recall that a binomial coefficient $\binom{u}{x}$ counts the number of times x occurs as a subword of u. The *k-binomial complexity function* of an infinite word $\mathbf{x}$ introduced in [5] is defined as $\mathsf{b}_{\mathbf{x}}^{(k)}\colon \mathbb{N} \to \mathbb{N}$, $n \mapsto \#(\mathcal{L}_n(\mathbf{x})/\sim_k)$, i.e., length-$n$ factors in $\mathbf{x}$ are counted up to k-binomial equivalence. (Here $\mathsf{b}_{\mathbf{x}}^{(1)}$ is the usual abelian complexity function [6].) For a survey on abelian properties of words, see [7]. In a recent work, we showed that a morphism is Parikh-collinear if and only if it maps all words with bounded k-binomial complexity to words with bounded $(k+1)$-binomial complexity (for all k) [8]. Thus each fixed point of a Parikh-collinear morphism has a bounded k-binomial complexity for all k (and in particular a bounded abelian complexity).

Let us summarize the contributions from [1] connecting Parikh-collinear fixed points to the notions of *automaticity*. A *k-automatic sequence* is the letter-to-letter coding of an iterated fixed point of a k-uniform morphism (images of letters have length k). Equivalently, a sequence $\mathbf{x} = a_0a_1a_2\cdots \in A^{\mathbb{N}}$, with $a_n \in A$, is k-automatic if there is a deterministic finite automaton with output that, on input n represented in base k, reaches a state with output a_n. For more on automatic sequences, see [9, 10]. For an arbitrary morphism $\sigma\colon A^* \to A^*$, we let $M_\sigma \in \mathbb{N}^{A\times A}$ denote its *adjacency matrix*, where $[M_\sigma]_{b,c} = |\sigma(c)|_b$ for all $b, c \in A$. A letter $a \in A$ is called *mortal* if $\sigma^n(a) = \varepsilon$ for some $n \geq 1$. If a is not mortal, we call it *immortal*.

Lemma 1 *Let $f\colon A^* \to A^*$ be Parikh-collinear and $a \in A$ be immortal. Then $\Psi(f(a))$ is a (right) eigenvector of M_f associated with the eigenvalue $\sum_{b\in A} |f(b)|_b$.*

For a Parikh-collinear morphism f, we let $\mathrm{eig}(f) := \sum_{b\in A} |f(b)|_b$ and we call $\mathrm{eig}(f)$ *the eigenvalue* f. This is justified as, the matrix M_f having rank 1, the only other eigenvalue is 0 with multiplicity $\#A - 1$.

Even though Parikh-collinear morphisms are generally non-uniform (images of some letters have distinct lengths) we proved the following result.

Theorem 2 [1, Thm. 5] *Let $f\colon A^* \to A^*$ be a Parikh-collinear morphism prolongable on a letter $a \in A$. Then the fixed point $f^\omega(a)$ is $\mathrm{eig}(f)$-automatic. Furthermore, a coding together with an $\mathrm{eig}(f)$-uniform morphism generating $f^\omega(a)$ can be effectively computed.*

The above theorem can be considered folklore: it can be seen as a consequence of [2, Thm. 2.2 or 4.2], the former of which is itself a reformulation of a result of Dekking

[11] (we note however, that the statements speak of non-erasing morphisms). For some perspective, it is well known that there exist infinite sequences that are the fixed points of non-uniform morphisms, but not k-automatic for any k, and that every k-automatic sequence is the image of a fixed point of a non-uniform morphism [12]. A recent preprint [13] completely characterizes those uniformly recurrent (i.e., every factor occurs infinitely often and with bounded gaps) morphic words that are automatic.

Next we proved in [1, Thm. 10] that, under some mild assumptions (on the automaticity of the cutting set that we will discuss in Section 3), the abelian complexity of a binary fixed point of a Parikh-collinear morphism is automatic. We can therefore use an automatic procedure to test whether or not this function is ultimately periodic, for example. Answering a question raised by V. Salo and A. Sportiello independently (personal communication), considering the abelian complexity of the fixed point $\mathbf{w} = 0100111001\cdots$ of the morphism $f\colon \{0,1\}^* \to \{0,1\}^*$ given by $0 \mapsto 010011$, $1 \mapsto 1001$, we showed that its abelian complexity is aperiodic. To conclude, we gave a proof sketch showing that the abelian complexity function of a fixed point of a non-erasing Parikh-collinear morphism is automatic.

1.1 Our Contributions

For this special issue dedicated to DLT–WORDS 2023 we did not want to replicate the results of the proceedings [1]. Therefore our main contribution is to generalize [1, Thm. 10] to an arbitrary Parikh-collinear morphism: meaning on an alphabet of arbitrary size and the morphism may be erasing.

Theorem 3 *Let $f\colon A^* \to A^*$ be a Parikh-collinear morphism prolongable on the letter a. The the abelian complexity function $\mathsf{a}_{\mathbf{x}}$ of $\mathbf{x} := f^\omega(a)$ is $\mathrm{eig}(f)$-automatic. Moreover, the automaton generating $\mathsf{a}_{\mathbf{x}}$ can be effectively computed given f and a.*

Before proving this result in Section 4, we first need a computable bound on the so-called recognizability constant. In Section 2, we have extracted from [14–16] and, in particular [17], the relevant definitions and important results showing that, for our study, such a constant exists. Expressed roughly, when we look at a sufficiently long factor, there is a unique pre-image by the morphism f and there is only one way to factorize this factor using blocks of the form $f(b)$, where b is a letter.

On this basis, we define in Section 3 the notion of a cutting set. Since the infinite word $\mathbf{x} = x_0x_1\cdots$ can be factorized as $f(x_0)f(x_1)\cdots$, this set consists of the integers $|f(x_0\cdots x_j)|$ for all $j \geq 0$. Our main observation is that, for a Parikh-collinear morphism f, this set is $\mathrm{eig}(f)$-definable. We insist that this is a major element which then enables us to apply a decision procedure about the abelian complexity of $\mathbf{x} = f^\omega(a)$. Such a procedure is described in Section 5. We consider the Parikh-collinear morphism $0 \mapsto 012$, $1 \mapsto 112002$, $2 \mapsto \varepsilon$ and prove with the help of `Walnut` that the fixed point starting with 0 has an ultimately periodic abelian complexity $135(377)^\omega$.

2 On the Recognizability

For an arbitrary morphism $\sigma : A^* \to A^*$, we define

$$|\sigma| := \max\{|\sigma(b)| : b \in A\} \text{ and } \langle\sigma\rangle := \min\{|\sigma(b)| : b \in A\},$$

where, for a word $w \in A^*$, we let $|w|$ denote its length.

A morphism g is prolongable on a letter a if $g(a) = ax$, where x is a word for which $g^n(x) \neq \varepsilon$ for all $n \in \mathbb{N}$. In particular, the infinite word $g^\omega(a) := \lim_{n\to\infty} g^n(a)$ exists and is a fixed point of g. In what follows, $f : A^* \to A^*$ is a Parikh-collinear morphism prolongable on the letter $a \in A$. For an arbitrary morphic word $\mathbf{x}$, thanks to [18, 19], one can decide whether $\mathbf{x}$ is ultimately periodic. In the case that $\mathbf{x}$ is generated by a Parikh-collinear morphism, by Theorem 2, $\mathbf{x}$ is also $\mathrm{eig}(f)$-automatic, and we can therefore make use of the logical characterization of automatic sequences; in particular, ultimate periodicity can be readily decided with `Walnut` [20] using a formula such as

$$\neg(\exists p > 0)(\exists i \geq 0)(\forall n \geq i)(\mathbf{x}(n) = \mathbf{x}(n+p)). \tag{1}$$

We will therefore assume in what follows that $\mathbf{x}$ is not ultimately periodic. Also, we restrict the alphabet A to the letters appearing in $f^n(a)$ for some n. As an example, for the Parikh-collinear morphism $f : 1 \mapsto 12,\ 2 \mapsto 21,\ 3 \mapsto 12$ prolongable on 1, we consider the restriction to the alphabet $\{1, 2\}$.

In what follows, an arbitrary morphism σ is called *primitive* if M_σ^n only contains positive entries for some $n \in \mathbb{N}$.

Lemma 4 *A non-erasing Parikh-collinear morphism is primitive.*

Proof Observe that all entries in the adjacency matrix are positive. □

Remark 5 Note that for a non-erasing Parikh-collinear morphism g, we may apply [15, Thm. 4] which directly provides a computable upper bound on the constant of recognizability (see Definition 16) for the aperiodic word $g^\omega(a)$.

Since f is Parikh-collinear and possibly erasing, there is a strong dichotomy among the letters of the alphabet. Either they are immortal and their image by f contains all letters, or their image by f is empty. Formally, for all $b \in A$, either $\Psi(f(b)) = 0$ or $\Psi(f(b))$ is a non-zero rational multiple of $\Psi(f(a))$. In the latter case, for all $n \geq 0$, $\Psi(f^n(b))$ is therefore non-zero. So, the alphabet is partitioned as $A = B \cup C$ where

$$B := \{b \in A \mid f^n(b) \neq \varepsilon,\ \forall n \geq 0\} \quad \text{and} \quad C := \{b \in A \mid f(b) = \varepsilon\}. \tag{2}$$

Definition 6 We use notation from (2). Let $\kappa : A^* \to B^*$ be a morphism such that $\kappa(b) = b$ if $b \in B$ and $\kappa(c) = \varepsilon$ for all $c \in C$. Now we define a morphism $g : B^* \to B^*$ such that $g(b) = \kappa(f(b))$ for all $b \in B$.

Roughly, the image by g of an immortal letter b of f is obtained by deleting the mortal letters appearing in $f(b)$.

The next statement is obvious.

Lemma 7 *With the above notation, $g = \kappa \circ f$ is a non-erasing Parikh-collinear morphism prolongable on a and satisfies $f(g^\omega(a)) = f^\omega(a)$.*

As an example, consider the Parikh-collinear morphism $f : 0 \mapsto 012, 1 \mapsto 112002$, $2 \mapsto \varepsilon$. We get $g : \{0, 1\}^* \to \{0, 1\}^*$ such that $g(0) = 01$ and $g(1) = 1100$.

An infinite word is called *recurrent* if each of its factors appears infinitely often.

Definition 8 Let $\mathbf{z}$ be a recurrent infinite word and u be a factor of $\mathbf{z}$. A *return word* to u is a non-empty factor w of $\mathbf{z}$ such that wu contains exactly two occurrences of u as a prefix and as a suffix of wu. The infinite word $\mathbf{z}$ is *K-linearly recurrent* if, for all factors u, any return word w to u is such that $|w| \leq K|u|$.

We recall a result from [14] and [15, Prop. 12]. It is important to note that the given upper bound is computable.

Proposition 9 *Let $\sigma : A^* \to A^*$ be a primitive morphism prolongable on a. The infinite word $\sigma^\omega(a)$ is K_σ-linearly recurrent and the constant K_σ is bounded by $|\sigma|^{4(\#A)^2}$.*

By the above result and Lemmas 4 and 7, there exists a constant K_g such that $\mathbf{y} = g^\omega(a)$ is K_g-linearly recurrent.

Corollary 10 *The infinite word $\mathbf{x} = f^\omega(a) = f(\mathbf{y})$ is K_f-linearly recurrent and the constant K_f is bounded by $K_g|f|/\langle f|_B\rangle$.*

Proof Let u be a factor of $\mathbf{x}$. There exists a factor v of $\mathbf{y}$ such that $f(v) = pus$ for some words p, s of minimal length and $|v| \leq |u|/\langle f|_B\rangle$ (recall that the letters of B do not vanish under f). Since $\mathbf{y}$ is linearly recurrent, any return word r to v has length at most $K_g|v|$. Observe that $f(r)$ contains a return word to u and has length bounded above by $K_g|v|\,|f| \leq K_g \frac{|f|}{\langle f|_B\rangle}|u|$. Now Lemma 7 allows us to conclude. □

The constant of recognizability is usually presented in the framework of shift spaces whose elements are biinfinite words, i.e., sequences indexed by $\mathbb{Z}$. We recap some of the main definitions and results.

Definition 11 The *shift operator* $S : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ is defined by $\mathbf{z} = (z_n)_{n\in\mathbb{Z}} \mapsto S(\mathbf{z}) = (z_{n+1})_{n\in\mathbb{Z}}$. A *shift space* is a subset $X \subseteq A^{\mathbb{Z}}$ that is shift-invariant, i.e., $S(X) = X$, and topologically closed. The *language* of X is the set denoted by $\mathcal{L}(X)$ of factors of the words in X. A shift space is *aperiodic* if all its elements are aperiodic. Recall that $\mathbf{z} \in A^{\mathbb{Z}}$ is *periodic* if $\mathbf{z} = S^n(\mathbf{z})$ for some $n \geq 1$.

Let $\sigma : A^* \to A^*$ be a morphism. We let

$$\mathcal{L}(\sigma) = \bigcup_{n\geq 0} \bigcup_{a\in A} \mathrm{Fac}(\sigma^n(a))$$

and the so-called *substitution shift* associated with σ is

$$\mathsf{X}(\sigma) = \{\mathbf{x} \in A^{\mathbb{Z}} : \mathcal{L}(x) \subseteq \mathcal{L}(\sigma)\}.$$

From the definition, it is clear that $\mathcal{L}(\mathsf{X}(\sigma)) \subseteq \mathcal{L}(\sigma)$. A morphism σ is *aperiodic* if the shift space $\mathsf{X}(\sigma)$ is aperiodic.

The notion of return words and linear recurrence naturally extends to shift spaces.

Definition 12 Let X be a shift space and $u \in \mathcal{L}(X)$. A non-empty word $w \in \mathcal{L}(X)$ is a *return word to u in X* if $wu \in \mathcal{L}(X)$ contains exactly two occurrences of u as a prefix and as a suffix of wu. The shift space X is *K-linearly recurrent* if it is minimal (for every closed stable subset Y of X, i.e., $S(Y) \subseteq Y$, one has $Y = \emptyset$ or $Y = X$) and for all non-empty words $u \in \mathcal{L}(X)$, the length of every return word to u in X is bounded by $K|u|$.

Proposition 13 *Let f be a Parikh-collinear morphism prolongable on a letter $a \in A$ such that $\mathbf{x} = f^\omega(a)$ is aperiodic. Then the shift space $\mathsf{X}(f)$ is K_f-linearly recurrent. Consequently, it is also aperiodic.*

Proof Let $X = \mathsf{X}(f)$. We first show that X is K_f-linearly recurrent. To that aim, let $u \in \mathcal{L}(X)$ and w be a return word to u in X. Observe that $\mathcal{L}(X) \subseteq \mathcal{L}(f) = \mathcal{L}(f^\omega(a))$. Hence u, wu are factors of $\mathbf{x}$ which is K_f-linearly recurrent by Corollary 10. Since $\mathbf{x}$ is aperiodic by assumption and X is minimal, X cannot contain a periodic point, hence the conclusion. □

We are now ready to first define the notion of recognizable morphism on X, then to introduce recognizable morphism on X with some constant of recognizability.

Definition 14 Let $X \subseteq A^{\mathbb{Z}}$ be a shift space. A morphism $\sigma : A^* \to B^*$ is *recognizable on X* if, for all $\mathbf{y} \in \sigma(X)$, there exists exactly one pair $(\mathbf{x}, \ell) \in X \times \mathbb{N}$ such that $0 \leq \ell < |\sigma(x_0)|$ and $\mathbf{y} = S^\ell(\sigma(\mathbf{x}))$, where x_0 is the first letter of $\mathbf{x}$.

Béal, Perrin, and Restivo generalized Mossé's theorem [17, Thm. 5.4].

Theorem 15 *Every morphism $\sigma : A^* \to A^*$ is recognizable on the set of aperiodic elements of $\mathsf{X}(\sigma)$. In particular, if σ is aperiodic, then it is recognizable on $\mathsf{X}(\sigma)$.*

Let X be a shift space and u, v be two finite words such that $uv \in \mathcal{L}(X)$. The *cylinder* with basis (u, v) is defined as

$$[u \cdot v]_X = \{\mathbf{z} \in X : \mathbf{z}_{[-|u|,|v|-1]} = uv\}.$$

In particular, if $u = \varepsilon$, we simply write $[v]_X = \{\mathbf{z} \in X : \mathbf{z}_{[0,|v|-1]} = v\}$.

Definition 16 Let $\sigma : A^* \to B^*$ be a morphism. Let X be a shift space on A and let Y be the closure of $\sigma(X)$ under the shift. A pair (u, v) of words such that $uv \in \mathcal{L}(Y)$ is *synchronizing* if there is at most one pair (b, ℓ) with $b \in A$ and $0 \leq \ell < |\sigma(b)|$ such that $[u \cdot v]_Y \cap S^\ell\sigma([b]_X) \neq \emptyset$. The morphism σ is recognizable on X *with constant n* if and only if every pair $(u, v) \in \mathcal{L}_n(Y) \times \mathcal{L}_{n+1}(Y)$ such that $uv \in \mathcal{L}(Y)$ is synchronizing.

Let X be an aperiodic shift space. The *repetition index* of X (also called *critical exponent* in the case of an infinite word) denoted by $\mathrm{rep}(X)$ is the supremum of the set of rational numbers e such that $\mathcal{L}(X)$ contains words of exponent e. Finally, we invoke the following result from [21].

Theorem 17 *The constant of recognizability on* $\mathsf{X}(\sigma)$ *of an aperiodic morphism* σ *is bounded by* $4\,\mathrm{rep}(\mathsf{X}(\sigma))\,\ell^2\,|\sigma|^{(2\ell+1)(2+|\sigma|^{(2^\ell+1)\ell})}$ *where* $\ell = \#A$.

In [16, Thm. 24], it is shown that a k-linearly recurrent aperiodic word is $(k+1)$-power-free (recall that an infinite word is *k-power-free* if it does not contain a factor of the form u^k with u non-empty). This result extends to shift spaces: Let X be an aperiodic shift space and suppose that it is K-linearly recurrent, then the repetition index is bounded by $\mathrm{rep}(X) < K+1$; see [14] and also [16]. Now an immediate application of Propositions 9 and 13 together with Theorem 17 leads to the following result.

Corollary 18 *Let* f *be a Parikh-collinear morphism prolongable on a letter* $a \in A$ *such that* $f^\omega(a)$ *is aperiodic. The constant of recognizability on* $\mathsf{X}(f)$ *of the aperiodic morphism* f *is bounded by* $4(|f|^{4\ell^2}+1)\,\ell^2\,|f|^{(2\ell+1)(2+|f|^{(2^\ell+1)\ell})}$ *where* $\ell = \#A$.

Now consider the right-infinite word $\mathbf{x} = f^\omega(a)$. It appears as a factor of a element in $\mathsf{X}(f)$. Indeed, since f is Parikh-collinear, there exists some $j \geq 1$ such that $f^j(a) = auav$ (one can take $j = 2$). Take the sequence $(f^n(au)\cdot f^n(a)\,f^n(v))_{n\geq 0}$. By compactness, we can extract a subsequence converging to some biinfinite word $\mathbf{z}\cdot f^\omega(a)$ belonging to $\mathsf{X}(f)$.

We have done all this to ensure that there is a computable bound C guaranteeing that the word $\mathbf{x}$ is recognizable: there is a window size bounded by C such that any factor within such a window is uniquely "desubstituted". More precisely, this will permit us to uniquely detect elements of the cutting set, which we discuss next.

3 The Cutting Set

Let $k \geq 2$ be an integer and consider the structure $\langle \mathbb{N}, +, V_k\rangle$, where $V_k(0) := 1$ and, for all $n \geq 1$, $V_k(n)$ is the largest power of k dividing n. A set $X \subseteq \mathbb{N}^d$ is *k-definable* if it can be defined by a first-order formula with d free variables within $\langle \mathbb{N}, +, V_k\rangle$. As a consequence of a theorem of Büchi [22], an infinite word $\mathbf{x}$ is k-automatic if and only if for every letter a, the set of positions where a occurs in $\mathbf{x}$ is k-definable. A sequence $F = (F_0, F_1, \ldots) \in (\mathbb{N}^d)^{\mathbb{N}}$ of integer tuples is called *k-synchronized* if the set $\{(n, F_n) : n \in \mathbb{N}\}$ is k-definable. Equivalently, a sequence $F \in (\mathbb{N}^d)^{\mathbb{N}}$ is k-synchronized if there exists a finite automaton having as input a $(d+1)$-tuple of integers written in base-k (padded to have equal length) and accepting precisely the tuples $([n]_k, [F_n]_k)$. For a reference on the logical approach to automatic sequences, including synchronized sequences, see [9, 10].

Let σ be a morphism prolongable on a and write $\mathbf{x} = \sigma^\omega(a)$. For all $n \geq 0$, we let $\mathrm{pref}_n(\mathbf{x})$ be the length-n prefix of $\mathbf{x}$. The corresponding *cutting set* is defined by

$$\mathsf{CS}_{\sigma,a} := \left\{ |\sigma(\mathrm{pref}_n(\mathbf{x}))| : \ n \geq 0 \right\}. \tag{3}$$

This set simply provides the indices where blocks $\sigma(b)$, with $b \in A$, start in a factorization of $\mathbf{x}$ of the form $\sigma(x_0)\sigma(x_1)\sigma(x_2)\cdots$. For example, applied to the

Parikh-collinear morphism $f : 0 \mapsto 012, 1 \mapsto 112002, 2 \mapsto \varepsilon$ considered before, we get

$$\mathbf{x} = |012|112002|112002|112002|012|012|\cdots \text{ and } \mathsf{CS}_{f,0} = \{0, 3, 9, 15, 21, 24, 27, \ldots\}.$$

The unary predicate $\mathsf{CS}_{\sigma,a}(n)$ holds true whenever $n \in \mathsf{CS}_{\sigma,a}$. Making use of the theory of recognizability presented in Section 2, we show that, in our usual setting, the cutting set is definable.

Proposition 19 *Let f be a Parikh-collinear morphism prolongable on a letter $a \in A$ such that $\mathbf{x} = f^\omega(a)$ is aperiodic. The cutting set $\mathsf{CS}_{f,a}$ is a $\mathrm{eig}(f)$-definable unary predicate.*

Proof By Corollary 18, there exists a constant of recognizability C on $\mathsf{X}(f)$ with the following property. By Definition 16, each factor $w = ucv$ of $\mathbf{x} = f^\omega(a)$ of length $2C + 1$ (here $|u| = |v| = C$ and $c \in A$) gives rise to a synchronizing pair, i.e., there exists a unique pair (b, ℓ) where $b \in A$, $0 \le \ell < |f(b)|$ such that $[u \cdot cv]_{\mathsf{X}(f)} \cap S^\ell f([b]_{\mathsf{X}(f)}) \neq \emptyset$. If $\ell = 0$, we have detected an element of the cutting set starting at the "center" c of the factor w. So with each factor w of length $2C + 1$, we associate a Boolean $T(w)$ stating whether or not the center of w belongs to the cutting set.

Since $\mathbf{x}$ is $\mathrm{eig}(f)$-automatic (see Theorem 2), for every factor w of length $2C + 1$ and all $n \ge C$, the unary formula $\varphi_w(n) \equiv \mathbf{x}[n - C, n + C] = w$ tells whether or not w occurs in $\mathbf{x}$ as a factor centered at position n (in other words, whether the position n is the center of w in $\mathbf{x}$). If $n \ge C$, the formula

$$\bigvee_{w \in \mathcal{L}_{2C+1}(\mathbf{x})} \varphi_w(n) \wedge T(w)$$

holds true whenever n belongs to the cutting set. For $n < C$, this can be defined by direct inspection: there is a finite number of elements in $\mathsf{CS}_{f,a} \cap \{0, \ldots, C - 1\}$ to encode manually into the final formula.

Now we have to effectively list all factors of length $2C + 1$ occurring in $\mathbf{x}$. Again by Theorem 2 we can effectively get a $\mathrm{eig}(f)$-uniform morphism g and a coding τ such that $\mathbf{x}$ is of the form $\tau(g^\omega(e))$ for some letter e. We can first list all length-2 factors occurring in $g^\omega(e) = \mathbf{y}$. For instance, we can use a formula such as $(\exists n)(\mathbf{y}(n) = b \wedge \mathbf{y}(n + 1) = c)$ to test whether or not bc occurs in $\mathbf{y}$. Second, every factor of length $2C + 1$ of $\mathbf{y}$ appears in $g^j(bc)$ for some letters b, c and $j = \lceil \log_{\mathrm{eig}(f)}(2C + 1) \rceil$. So scanning these words $g^j(bc)$ with a window of size $2C + 1$, we get all desired factors and we apply τ to them to get all factors of length $2C + 1$ occurring in $\mathbf{x}$. □

Let us observe that the above proposition can be given in a different framework where we focus on the recognizability of a single infinite word with respect to the considered morphism. We take the following definition from [15] adapted to (right) infinite words.

Definition 20 Let $\mathbf{x} = x_0 x_1 \cdots$ be a fixed point of a prolongable morphism σ. We say that σ is *recognizable* on $\mathbf{x}$ if there exists a constant $D > 0$ such that, for all

$n \geq 0$ and all i such that $|\sigma(x_0 \cdots x_{i-1})| \geq D$, if the factors $\mathbf{x}[n - D, n + D]$ and $\mathbf{x}[|\sigma(x_0 \cdots x_{i-1})| - D, |\sigma(x_0 \cdots x_{i-1})| + D]$ are equal, then there exists an index j such that $n = |\sigma(x_0 \cdots x_{j-1})|$ and $x_i = x_j$.

The least D with the above property is then called the *constant of recognizability of* σ on $\mathbf{x}$.

Remark 21 If σ is an aperiodic morphism, then σ is recognizable on any of its fixed points $\mathbf{x}$. The constant of recognizability of σ on $\mathbf{x}$ is bounded above by the constant of recognizability of σ on $\mathsf{X}(\sigma)$ given in Theorem 17. There are some intricacies regarding the two notions of constant of recognizability given in Definitions 20 and 16; we refer the reader to [23] for more on the topic.

The reader may readily adapt the proof of Proposition 19 to the following situation. Note the weaker version (speaking of Parikh-collinear morphisms) presented in Proposition 19 will be used in the next section, but Theorem 22 gives an interesting result on its own.

Theorem 22 *Let* $\mathbf{x} = \sigma^\omega(a)$ *be a fixed point of a prolongable morphism* σ. *If* σ *is recognizable on* $\mathbf{x}$ *with computable recognizability constant* C *and if* $\mathbf{x}$ *is* k*-automatic for some* $k \geq 1$, *then the cutting set* $\mathsf{CS}_{\sigma,a}$ *is a* k*-definable unary predicate.*

4 Proof of Theorem 3

We shall now proceed with the proof of the main result. Let again $f\colon A^* \to A^*$ be a Parikh-collinear morphism and let $\mathbf{x} = f^\omega(a)$ be its fixed point.

The following theorem of Shallit outlines our strategy to complete the proof. In the following, a numeration system is *addable* if addition is recognizable by a finite automaton. All integer-based numeration systems are addable (thus covering our setting), but others exist too (see, e.g., [10]).

Theorem 23 [24] *Let* $\mathbf{x}$ *be an automatic sequence in some addable numeration system* $\mathcal{S}$, *and assume that*

1. *the sequence* $(\Psi(\mathrm{pref}_n(\mathbf{x}))_{n\geq 0}$ *is synchronized (w.r.t.* $\mathcal{S}$*); and*
2. *the abelian complexity function* $\mathsf{a}_\mathbf{x}\colon \mathbb{N} \to \mathbb{N}$ *is bounded above by a constant.*

Then $(\mathsf{a}_\mathbf{x}(n))_{n\geq 0}$ *is an automatic sequence (w.r.t.* $\mathcal{S}$*) and the deterministic finite automaton with output computing it is effectively computable.*

Furthermore, if Condition 1 *holds, then* Condition 2 *is decidable.*

Our aim is to show that the sequence $(|\mathrm{pref}_n(\mathbf{x})|_b)_{n\geq 0}$ is synchronized for each $b \in A$ (which straightforwardly implies that $(\Psi(\mathrm{pref}_n(\mathbf{x}))_{n\geq 0}$ is also synchronized). We will utilize the Parikh-collinearity of the morphism f; in particular, we use the property that for any word $w \in A^*$, we have that $|f(w)|_b = \frac{|f(a)|_b}{|f(a)|}|f(w)|$. In order to do so, given an index $i \in \mathbb{N}$, we look for two integers around i: the next and previous elements found in the cutting set $C_{f,\mathbf{x}}$. The next obvious lemma says we can do this in an automatic way (a proof can be found in [1]).

Lemma 24 *Let $C = \{0 = c_0 < c_1 < c_2 < \cdots\}$ be an infinite k-definable subset of $\mathbb{N}$ for some $k \geq 1$. The functions* $\mathrm{ne} : \mathbb{N} \to \mathbb{N}$ *mapping i to the least element in C greater than or equal to i and* $\mathrm{pr} : \mathbb{N} \to \mathbb{N}$ *mapping i to the greatest element in C less than i, are k-definable. (We set* $\mathrm{pr}(0) = 0$.*)*

Lemma 25 *Let $f : A^* \to A^*$ be a Parikh-collinear morphism prolongable on a and write $\mathbf{x} = f^\omega(a)$. For all $b \in A$, the sequence $(|\operatorname{pref}_n(\mathbf{x})|_b)_{n\geq 0}$ is $\mathrm{eig}(f)$-synchronized.*

Proof Let $b \in A$. Since f is Parikh-collinear, for each immortal letter c, the ratio $|f(c)|_b/|f(c)|$ is constant and depends only on b. Thus write $|f(a)|_b/|f(a)| = r/q$.

Consider the length-n prefix of $\mathbf{x}$ and write $\operatorname{pref}_n(\mathbf{x}) = f(p_n)t_n$, where p_n is a prefix of $\mathbf{x}$ such that $\mathrm{pr}(n) = |f(p_n)|$, and t_n is a prefix of the image of a letter. Since $|f(p_n)|_b = \frac{r}{q}|f(p_n)|$, we get $q|\operatorname{pref}_n(\mathbf{x})|_b = r|f(p_n)| + q|t_n|_b$. Define the function $F(n) = |\operatorname{pref}_n(\mathbf{x})|_b$ for all $n \geq 0$. Then the following binary predicate defines the pair $(n, F(n))$:

$$P(n, y) = \exists m, z\colon (\mathrm{pr}(n) = m) \wedge (q \cdot (y - z) = r \cdot m) \wedge (|\mathbf{x}[m \ldots n-1]|_b = z).$$

The formula for P has two free variables (y and n), so $\{(n, F(n))\colon n \in \mathbb{N}\}$ is $\mathrm{eig}(f)$-definable: since $\mathbf{x}[\mathrm{pr}(n)\ldots n-1|$ attains finitely many values (as a prefix of the image of a letter), the last check $(|\mathbf{x}[m\ldots n-1]|_b = z)$ can be expressed by a first-order logical formula with indexing into $\mathbf{x}$; the word $\mathbf{x}$ is $\mathrm{eig}(f)$-automatic by Theorem 2, hence the positions of b in $\mathbf{x}$ are defined by a unary predicate φ_b; and the cutting set $C_{f,a}$ is $\mathrm{eig}(f)$-definable by Proposition 19, so that pr is as well by Lemma 24. Hence $(|\operatorname{pref}_n(\mathbf{x})|_b)_{n\geq 0}$ is $\mathrm{eig}(f)$-synchronized. □

Proof of Theorem 3 A fixed point of a Parikh-collinear morphism is e_f-automatic by Theorem 2. As a corollary of [4, Thm. 11], its abelian complexity function is bounded by a constant, so Condition 2 in Theorem 23 is satisfied. Since Condition 1 is equivalent to the property that for each $b \in A$, the sequence $(|\operatorname{pref}_n(\mathbf{x})|_b)_{n\geq 0}$ is $\mathrm{eig}(f)$-synchronized, the above lemma allows to use Theorem 23 to conclude. □

5 A Detailed Discussion of the Procedure

Throughout this section, we let f be defined by $0 \mapsto 012, 1 \mapsto 1120022 \mapsto \varepsilon$, and $f^\omega(0) = \mathbf{x} = x_0x_1\cdots$. Our aim is to prove the following.

Proposition 26 *The fixed point $\mathbf{x} = 012112002112002\cdots$ of the Parikh-collinear morphism $f\colon 0 \mapsto 012, 1 \mapsto 112002, 2 \mapsto \varepsilon$ has abelian complexity equal to $135(377)^\omega$.*

Computing $\mathrm{eig}(f) = \sum_{a=0}^{2} |f(a)|_a = 3$, we know that $\mathbf{x}$ is 3-automatic. In [1] with Theorem 2, we give an effective procedure to compute an equivalent morphic representation; the procedure produces the coding τ defined by

$$\widehat{0}_1, \widehat{1}_4, \widehat{1}_5 \mapsto 0; \quad \widehat{0}_2, \widehat{1}_1, \widehat{1}_2 \mapsto 1; \quad \widehat{0}_3, \widehat{1}_3, \widehat{1}_6 \mapsto 2$$

and the 3-uniform morphism g defined by

$$\widehat{0}_1, \widehat{1}_5, \widehat{1}_6 \mapsto \widehat{0}_1\widehat{0}_2\widehat{0}_3; \quad \widehat{0}_2, \widehat{1}_1, \widehat{1}_3 \mapsto \widehat{1}_1\widehat{1}_2\widehat{1}_3; \quad \widehat{0}_3, \widehat{1}_2, \widehat{1}_4 \mapsto \widehat{1}_4\widehat{1}_5\widehat{1}_6,$$

so that $\tau(g^\omega(\widehat{0}_1)) = \mathbf{x}$.

One notes that there are redundant letters (i.e., they have equal images under both τ and $g \circ \tau$). We thus find a simpler morphism h by identifying them:

$$0 \mapsto 012; \quad 1 \mapsto 134; \quad 2 \mapsto 506; \quad 3 \mapsto 506; \quad 4 \mapsto 134; \quad 5 \mapsto 506; \quad 6 \mapsto 012,$$

with which $\tau'(h^\omega(0)) = \mathbf{x}$, where τ' is defined by $0, 5 \mapsto 0$; $1, 3 \mapsto 1$; $2, 4, 6 \mapsto 2$.

We may introduce the 3-automatic word $\mathbf{x}$ to `Walnut` as follows:

```
morphism h "0->012 1->134 2->506 3->506 4->134 5->506
 6->012";
morphism tau "0->0 1->1 2->2 3->1 4->2 5 ->0 6->2";
promote H h;
image X tau H;
```

`Walnut` now knows the infinite word as `X`, and it is now easy to verify that $\mathbf{x}$ is aperiodic. Indeed, (1) translates to:

```
eval isaperiodic "?msd_3 ~(Ep,i p>0
& (An n>i => (X[n]=X[n+p])))";
```

and `Walnut` produces `True`.

Following the procedure, we next wish to compute (or bound) the constant of recognizability. The bound given in Corollary 18 is $(6^{36} + 1) \cdot 6^{7\cdot(2+6^{27})+2}$, which is unmanageable in practice. At this point, we compute the actual constant of recognizability (with the help of `Walnut`) to proceed with the illustration.

Lemma 27 *Given $f : 0 \mapsto 012, 1 \mapsto 1120022 \mapsto \varepsilon$, its constant of recognizability is* 2.

Proof We observe that the factor 120 appears both in $f(00) = 012012$ and $f(1) = 112002$, so the pair (1, 20) is not synchronizing. This observation bounds the constant of recognizability from below by 2.

We observe that each factor of length 5 contains at least one occurrence of 2; this is because 2 appears at the position n if and only if $n \equiv 2 \bmod 3$, a fact that can be verified using `Walnut`:

```
eval appearance2 "?msd_3 An X[n]=@2 <=> Em n=3*m+2";
```

Let $w = uv$ be a factor of $\mathbf{x}$ with $|u| = 2$ and $|v| = 3$. From the above we deduce that v contains an occurrence of 2.

Since any cutting point is either 0 or appears just after an occurrence of 2 (both $f(0)$ and $f(1)$ end with 2), it suffices to inspect the two letters appearing just before a 2 in v. Indeed, the return words to 2 in $\mathbf{x}$ are 201, 200, and 211; if the two preceding letters are 00 or 01, then the position after 2 is a cutting point. Otherwise it is not. Thus the constant of recognizability is bounded above by 2. □

We next proceed to define the cutting sequence of $\mathbf{x}$ in `Walnut` as follows; the index n is a cutting point if $n = 0$ or $n \geq 3$ and $x_{n-1} = 2$ and $x_{n-3} \neq 1$ (this can be deduced from the proof of Lemma 27).

```
def cut "?msd_3 n=0 | (n>=3 & X[n-1]=@2 & ~(X[n-3]=@1))";
```

Using the `cut` set, we define the pairs (n, x) such that x is the largest cut point that is at most n. The following predicate `prev` recognizes exactly these pairs (see Lemma 24).

```
def prev "?msd_3 x<=n & $cut(x)
     & (Ay (y>x & y<=n)=>~$cut(y))";
```

Next, we define the synchronized sequence of the number of 0's (resp., 1's, 2's) in the prefix of length $n + 1$, $n \geq 0$.

```
def prefn0 "?msd_3 (n<=2 & y=1)
      | (3<=n & Em,z ($prev(n,m) & 3*y=m+3*z
      & ((X[m]=@0 & z=1) |
      (X[m]=@1 & ((n<m+3 & z=0)
      | (n=m+3 & z=1) | (n>=m+4 & z=2))))))";
```

Here `prefn0(n,y)` is true if y is the number of 0's appearing in the prefix of length $n + 1$ of $\mathbf{x}$, with $n \geq 0$. We note that if $|f(w)| = \ell$, then $|f(w)|_0 = \ell/3$. Hence for a prefix $f(w)z$, where z is a prefix of the image of a letter, we have $|f(w)z|_0 = \ell/3 + |z|_0$. If z begins with 0, then we know that z is a prefix of $f(0)$, so that $|z|_0 = 1$ as long as $z \neq \varepsilon$. Otherwise z is a prefix of $f(1)$, and $|z|_0 = 0$ if $|z| \leq 3$; $|z|_0 = 1$ if $|z| = 4$; and $|z|_0 = 2$ otherwise.

With similar arguments one can see that the following predicates define the pairs $(n, |\operatorname{pref}_{n+1}(\mathbf{x})|_a)$, for $a \in \{1, 2\}$.

```
def prefn1 "?msd_3 Em,z $prev(n,m) & 3*y=m+3*z &
      ((X[m]=@0 & ((m=n & z=0) | (n>=m+1 & z=1))) |
      (X[m]=@1 & ((m=n & z=1) | (n>=m+1 & z=2))))";

def prefn2 "?msd_3 Em,z $prev(n,m) & 3*y=m+3*z &
      ((X[m]=@0 & ((n<m+2 & z=0)
      | (m+2=n & z=1))) | (X[m]=@1 &
      ((n<m+2 & z=0) | (n>=m+2 & n<m+5 & z=1)
      | (n=m+5 & z=2))))";
```

From this point onward we may proceed as outlined by Shallit in [24] to find the abelian complexity function of $\mathbf{x}$ as a 3-automatic sequence.

Remark 28 We could now find the sequence $(\Psi(\operatorname{pref}_n(\mathbf{x})))_{n\geq 0}$ as a synchronized sequence with the following command:

```
def PrefParikhSync "?msd_3 (n=0 & x=0 & y=0 & z=0) |
      (n>=1 & $prefn0(n-1,x) & $prefn1(n-1,y)
      & $prefn2(n-1,z))";
```

However, the automaton seems to be too complex to work with in a practical way.

We shall opt to proceed with the approach of [24] presented for the Tribonacci word. A different, perhaps more efficient, approach is outlined in [10, §10.13.3]. In particular, we work with the synchronized sequences $(|\operatorname{pref}_n(\mathbf{x})|_a)_{n\geq 0}$, $a \in \{0, 1, 2\}$, separately instead:

```
def pref0 "?msd_3 (n=0 & y=0) | (n>=1 & $prefn0(n-1,y))";
def pref1 "?msd_3 (n=0 & y=0) | (n>=1 & $prefn1(n-1,y))";
def pref2 "?msd_3 (n=0 & y=0) | (n>=1 & $prefn2(n-1,y))";
```

Next we define the automata accepting the triples $(i, n, |x_i \cdots x_{i+n-1}|_a)$, for $a \in \{0, 1, 2\}$:

```
def sncin0 "?msd_3 Ax,y ($pref0(i,x) & $pref0(i+n,y)) =>
     (z + x = y)";
def sncin1 "?msd_3 Ax,y ($pref1(i,x) & $pref1(i+n,y)) =>
     (z + x = y)";
def sncin2 "?msd_3 Ax,y ($pref2(i,x) & $pref2(i+n,y)) =>
     (z + x = y)";
```

For each $a \in \{0, 1, 2\}$, we inspect the possible differences $|v| - |\operatorname{pref}_n(\mathbf{x})|_a$, where v ranges over the factors of $\mathbf{x}$ of length n. Since $\mathbf{x}$ is guaranteed to have bounded abelian complexity, there are only finitely many such possible values; here we may inspect the possible values of the differences as follows. The first predicate accepts the non-negative values of k such that $k = |v|_0 - |\operatorname{pref}_n(\mathbf{x})|$; the second accepts the non-negative values of k such that $-k = |v|_0 - |\operatorname{pref}_n(\mathbf{x})|_0$:

```
def diffs0pos "?msd_3 En,i,x,z $sncin0(i,n,x) & $pref0(n,z)
     & x=k+z";
def diffs0neg "?msd_3 En,i,x,z $sncin0(i,n,x) & $pref0(n,z)
     & x+k=z";
```

Inspecting the automata obtained, the possible values in both cases are $k = 0, 1, 2$. This implies that $\big||v|_0 - |\operatorname{pref}_n(\mathbf{x})|_0\big| \leq 2$, and all possible values are attained. With similar inspections for the other letters, we find that

$$-3 \leq |v|_1 - |\operatorname{pref}_n(\mathbf{x})|_1 \leq 2 \quad \text{and} \quad 0 \leq |v|_2 - |\operatorname{pref}_n(\mathbf{x})|_2 \leq 1,$$

and all possible values are attained. We thus have that

$$\Psi(v) - \Psi(\operatorname{pref}_n(\mathbf{x})) \in \{-2, \ldots, 2\} \times \{-3, \ldots, 2\} \times \{0, 1\} \tag{4}$$

for any factor v of length n. (Note that this in particular shows that $\mathbf{x}$ has bounded abelian complexity.)

If S is the set of possible triples in (4), we note that the triples $S + (2, 3, 0)$ will be non-negative. To get all the possible triples (s, t, u), we inspect the automaton constructed by `Walnut` with the command

```
def validtriples "?msd_3 Ei,n,a,b,c,d,e,f
     $sncin0(i,n,a) & $pref0(n,b) & s+b = a+2 &
     $sncin1(i,n,c) & $pref1(n,d) & t+d = c+3 &
     $sncin2(i,n,e) & $pref2(n,f) & u+f = e";
```

We obtain the automaton in Fig. 1. Inspecting it, we see that from the set appearing in (4), all ten vectors with the property that the entries sum to 0 are attainable (we are comparing Parikh vectors of two words of the same length n); they are

$$\{(-2, 2, 0), (-2, 1, 1), (-1, 1, 0), (-1, 0, 1), (0, -1, 1),$$
$$(0, 0, 0), (1, -2, 1), (1, -1, 0), (2, -3, 1), (2, -2, 0)\}.$$

Since $(0, 0, 0)$ is attained for any length n, we have 2^9 possible sets of difference vectors to consider. For each such difference vector, we may provide a predicate recognizing those i and n for which the vector is attained. For example, the vector $(-2, 2, 0)$ is defined with the command

```
def vecn220 "?msd_3 Ea,b,c,d,e,f
     $sncin0(i,n,a) & $pref0(n,b) & a+2=b &
     $sncin1(i,n,c) & $pref1(n,d) & c=d+2 &
     $sncin2(i,n,e) & $pref2(n,f) & e=f";
```

In principle, one could then consider all 2^9 possible combinations of possible difference vectors for a given length n. However, computations suggest that the possible sets of vectors are the following:

$S_1 = \{(-1, 0, 1), (-1, 1, 0), (0, 0, 0)\},$
$S_2 = \{(-1, 0, 1), (-1, 1, 0), (0, -1, 1), (0, 0, 0), (1, -1, 0)\},$
$S_3 = \{(-1, 1, 0), (0, 0, 0), (1, -1, 0)\},$
$S_4 = \{(-1, 0, 1), (-1, 1, 0), (0, -1, 1), (0, 0, 0), (1, -2, 1), (1, -1, 0), (2, -2, 0)\},$
$S_5 = \{(-1, 0, 1), (0, -1, 1), (0, 0, 0), (1, -2, 1), (1, -1, 0), (2, -3, 1), (2, -2, 0)\},$
$S_6 = \{(0, 0, 0), (1, -1, 0), (2, -2, 0)\},$
$S_7 = \{(-2, 1, 1), (-1, 0, 1), (-1, 1, 0), (0, -1, 1), (0, 0, 0), (1, -2, 1), (1, -1, 0)\},$
$S_8 = \{(-2, 1, 1), (-2, 2, 0), (-1, 0, 1), (-1, 1, 0), (0, -1, 1), (0, 0, 0), (1, -1, 0)\}.$

Let us define, for each i, the lengths n for which the set S_i is attained. For example, the lengths corresponding to S_1, S_2, S_3, and S_6 would be defined as

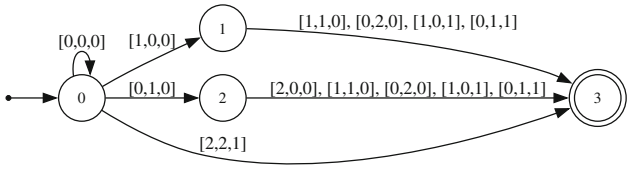

Fig. 1 The automaton accepting base-3 representations of triples of the form $\Psi(v) - \Psi(\mathrm{pref}_n(\mathbf{x})) + (2, 3, 0)$, where v ranges through the factors of length n, and n ranges through the natural numbers

```
def S1 "?msd_3 Ai ($vecn101(i,n) | $vecn110(i,n)
  | $vec000(i,n))& (Ej,k $vecn101(j,n)& $vecn110(k,n))";
def S2 "?msd_3 Ai ($vecn101(i,n) | $vecn110(i,n)
                                 | $vec0n11(i,n)
                                 | $vec000(i,n)
                                 | $vec1n10(i,n)) &
         Ej,k,l,m ($vecn101(j,n) & $vecn110(k,n)
            & $vec0n11(l,n) & $vec1n10(m,n))";
def S3 "?msd_3 (Ai $vecn110(i,n) | $vec000(i,n)
                                 | $vec1n10(i,n)) &
           (Ej,k $vecn110(j,n) & $vec1n10(k,n))";
def S6 "?msd_3 (Ai $vec000(i,n) | $vec1n10(i,n)
                                | $vec2n20(i,n)) &
          (Ej,k $vec1n10(j,n) & $vec2n20(k,n))";
```

To avoid memory issues, we split the definitions of S_4, S_5, S_7, and S_8 into several parts:

```
def S4a "?msd_3 Ai $vecn101(i,n) | $vecn110(i,n)
  | $vec0n11(i,n) | $vec000(i,n) | $vec1n21(i,n)
  | $vec1n10(i,n) | $vec2n20(i,n)";
def S4b "?msd_3 $S4a(n) & Ei $vecn101(i,n)";
def S4c "?msd_3 $S4b(n) & Ei $vecn110(i,n)";
def S4d "?msd_3 $S4c(n) & Ei $vec0n11(i,n)";
def S4e "?msd_3 $S4d(n) & Ei $vec1n21(i,n)";
def S4f "?msd_3 $S4e(n) & Ei $vec1n10(i,n)";
def S4 "?msd_3 $S4f(n) & Ei $vec2n20(i,n)";

def S5a "?msd_3 Ai ($vecn101(i,n) | $vec0n11(i,n)
  | $vec000(i,n) | $vec1n21(i,n) | $vec1n10(i,n)
  | $vec2n31(i,n) | $vec2n20(i,n))";
def S5b "?msd_3 $S5a(n) & Ei $vecn101(i,n)";
def S5c "?msd_3 $S5b(n) & Ei $vec0n11(i,n)";
def S5d "?msd_3 $S5c(n) & Ei $vec1n21(i,n)";
def S5e "?msd_3 $S5d(n) & Ei $vec1n10(i,n)";
def S5f "?msd_3 $S5e(n) & Ei $vec2n31(i,n)";
def S5 "?msd_3 $S5f(n) & Ei $vec2n20(i,n)";

def S7a "?msd_3 Ai $vecn211(i,n) | $vecn101(i,n)
  | $vecn110(i,n) | $vec0n11(i,n) | $vec000(i,n)
  | $vec1n21(i,n) | $vec1n10(i,n)";
def S7b "?msd_3 $S7a(n) & Ei $vecn211(i,n)";
def S7c "?msd_3 $S7b(n) & Ei $vecn101(i,n)";
def S7d "?msd_3 $S7c(n) & Ei $vecn110(i,n)";
def S7e "?msd_3 $S7d(n) & Ei $vec0n11(i,n)";
def S7f "?msd_3 $S7e(n) & Ei $vec1n21(i,n)";
```

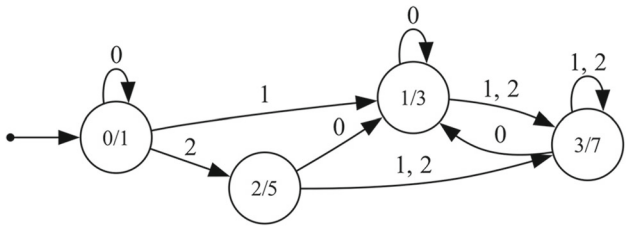

Fig. 2 The abelian complexity function of the fixed point $\mathbf{x} = 012112002112002\cdots$ of the Parikh-collinear morphism $f: 0 \mapsto 012,\ 1 \mapsto 112002,\ 2 \mapsto \varepsilon$ as a 3-automatic sequence

```
def S7 "?msd_3 $S7f(n) & Ei $vec1n10(i,n)";

def S8a "?msd_3 Ai $vecn211(i,n) | $vecn220(i,n)
  | $vecn101(i,n) | $vecn110(i,n) | $vec0n11(i,n)
  | $vec000(i,n) | $vec1n10(i,n)";
def S8b "?msd_3 $S8a(n) & Ei $vecn211(i,n)";
def S8c "?msd_3 $S8b(n) & Ei $vecn220(i,n)";
def S8d "?msd_3 $S8c(n) & Ei $vecn101(i,n)";
def S8e "?msd_3 $S8d(n) & Ei $vecn110(i,n)";
def S8f "?msd_3 $S8e(n) & Ei $vec0n11(i,n)";
def S8 "?msd_3 $S8f(n) & Ei $vec1n10(i,n)";
```

To obtain the abelian complexity function as an automatic sequence, we finally perform the following commands:

```
def abcomp1 "?msd_3 n=0";
def abcomp3 "?msd_3 $S1(n) | $S3(n) | $S6(n)";
def abcomp5 "?msd_3 $S2(n)";
def abcomp7 "?msd_3 $S4(n) | $S5(n) | $S7(n) | $S8(n)";
combine abcomp abcomp1=1 abcomp5=5 abcomp3=3 abcomp7=7;
```

The first four automata recognize those lengths n for which the abelian complexity equals 1, 3, 5, and 7, respectively. The last combines these automata to form an automatic sequence over the alphabet $\{1, 3, 5, 7\}$. The automaton obtained is depicted in Fig. 2. Inspecting the automaton, we see that the abelian complexity function of $\mathbf{x}$ equals $135(377)^\omega$, as desired.

6 Concluding Remarks

We may address similar questions. In the same vein as Sportiello and Salo's question, we may ask: Is the abelian complexity of the fixed point of any *Parikh-constant morphism*, i.e., all images of letters have the same Parikh vector [5], always ultimately periodic? We know with [1] that this is not the case for an arbitrary Parikh-collinear morphism, but Parikh-constant morphisms are more restrictive: all columns of M_f are the same.

In Theorem 22, there is an assumption about recognizability. Nevertheless, there are situations where recognizability does not hold and still, the cutting set is definable.

As an example, consider the non-uniform morphism $f : a \mapsto ab, b \mapsto b'c', b' \mapsto b, c' \mapsto ccc$ and $c \mapsto cc$. Its fixed point starting with a is also 2-automatic and generated by the morphism $a \mapsto ab, b \mapsto b'c', b' \mapsto bc, c' \mapsto cc, c \mapsto cc$ (a slight modification of the morphism used to generate the characteristic sequence of powers of 2). Because of the arbitrarily long blocks of c's appearing in $f^\omega(a)$, f is not recognizable on this infinite word. Nevertheless, the cutting set $\mathsf{CS}_{f,a} = \{0, 2, 4, 5, 8, 10, \ldots\}$ is 2-definable because it is easy to see that it is of the form $(2\mathbb{N} \setminus \{4^n + 2 \mid n > 0\}) \cup \{4^n + 1 \mid n > 0\}$. So the conclusion of Theorem 22 may hold for a larger class of morphic words (being simultaneously k-automatic for some k).

Acknowledgements We thank J. Leroy for fruitful discussions on morphic words and pointing out useful references. We also thank P. Popoli for pointing out the alternative method of computing the abelian complexity function in [10, §10.13.3]. Our appreciation is extended to A. Sportiello and V. Salo for asking the question leading to this paper. We warmly thank M.-P. Béal, F. Durand, and D. Perrin for sharing a draft of their book [21]. We also thank the reviewers of [1] for their suggestions. We thank the anonymous reviewers of the current paper for their suggestions which greatly improved the readability and accessibility of the text.

Author Contributions All authors equally contributed to the main content.

Funding M. Rigo is supported by the FNRS Research grant T.0196.23 (PDR). M. Stipulanti is an FNRS Research Associate supported by the Research grant 1.C.104.24F. Part of the work was performed while M. Whiteland was affiliated with University of Liège and supported by the FNRS Research grant 1.B.466.21F

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing Interests The authors declare no competing interests.

Ethics Approval and Consent to Participate Not applicable

Consent for Publication Not applicable

References

1. Rigo, M., Stipulanti, M., Whiteland, M.A.: Automaticity and Parikh-collinear morphisms. In: Combinatorics on Words. Lecture Notes in Comput. Sci., vol. 13899, pp. 247–260. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-33180-0_19
2. Allouche, J.-P., Dekking, M., Queffélec, M.: Hidden automatic sequences. Comb. Theory **1**(20) (2021) https://doi.org/10.5070/C61055386
3. Sloane, N.J.A., al.: The On-Line Encyclopedia of Integer Sequences. https://oeis.org

4. Cassaigne, J., Richomme, G., Saari, K., Zamboni, L.Q.: Avoiding Abelian powers in binary words with bounded Abelian complexity. Int. J. Found. Comput. S. **22**(4), 905–920 (2011). https://doi.org/10.1142/S0129054111008489
5. Rigo, M., Salimov, P.: Another generalization of abelian equivalence: binomial complexity of infinite words. Theor. Comput. Sci. **601**, 47–57 (2015). https://doi.org/10.1016/j.tcs.2015.07.025
6. Erdős, P.: Some unsolved problems. Michigan Math. J. **4**, 291–300 (1958)
7. Fici, G., Puzynina, S.: Abelian combinatorics on words: a survey. Comput. Sci. Rev. **47**, 100532 (2023). https://doi.org/10.1016/j.cosrev.2022.100532
8. Rigo, M., Stipulanti, M., Whiteland, M.A.: Characterizations of families of morphisms and words via binomial complexities. European J. Combin. **118**, 103932 (2024). https://doi.org/10.1016/j.ejc.2024.103932
9. Allouche, J.-P., Shallit, J.: Automatic Sequences: Theory, Applications, Generalizations. Cambridge University Press, Cambridge (2003). https://doi.org/10.1017/CBO9780511546563
10. Shallit, J.: The Logical Approach to Automatic Sequences: Exploring Combinatorics on Words with Walnut. London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge (2022). https://doi.org/10.1017/9781108775267
11. Dekking, F.M.: The spectrum of dynamical systems arising from substitutions of constant length. Z. Wahrscheinlichkeitstheor. Verw. Geb. **41**, 221–239 (1978). https://doi.org/10.1007/BF00534241
12. Allouche, J.-P., Shallit, J.: Automatic sequences are also non-uniformly morphic. In: Discrete Mathematics and Applications. Springer Optim. Appl., vol. 165, pp. 1–6. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-55857-4_1
13. Krawczyk, E., Müllner, C.: Automaticity of uniformly recurrent substitutive sequences (2023). https://doi.org/10.48550/arXiv.2111.13134
14. Durand, F.: A characterization of substitutive sequences using return words. Discrete Math. **179**(1–3), 89–101 (1998). https://doi.org/10.1016/S0012-365X(97)00029-0
15. Durand, F., Leroy, J.: The constant of recognizability is computable for primitive morphisms. J. Integer Seq. **20**(4), 17–4515 (2017)
16. Durand, F., Host, B., Skau, C.: Substitutional dynamical systems, Bratteli diagrams and dimension groups. Ergodic Theory Dynam. Systems **19**(4), 953–993 (1999). https://doi.org/10.1017/S0143385799133947
17. Béal, M.-P., Perrin, D., Restivo, A.: Recognizability of morphisms. Ergodic Theory Dyn. Syst. 1–25 (2023). https://doi.org/10.1017/etds.2022.109
18. Durand, F.: Decidability of the HD0L ultimate periodicity problem. RAIRO Theor. Inform. Appl. **47**(2), 201–214 (2013). https://doi.org/10.1051/ita/2013035
19. Mitrofanov, I.: A proof for the decidability of HD0L ultimate periodicity. arXiv (2011). https://doi.org/10.48550/arXiv.1110.4780
20. Mousavi, H.: Automatic Theorem Proving in Walnut. arXiv (2016). https://doi.org/10.48550/arXiv.1603.06017
21. Béal, M.-P., Durand, F., Perrin, D.: Substitution shifts. manuscript (2024)
22. Büchi, J.R.: Weak second-order arithmetic and finite automata. Z. Math. Logik Grundlagen Math. **6**, 66–92 (1960). https://doi.org/10.1002/malq.19600060105
23. Béal, M., Berthé, V., Perrin, D., Restivo, A.: A note on one-sided recognizable morphisms (2022). https://doi.org/10.48550/arXiv.2204.03892
24. Shallit, J.: Abelian complexity and synchronization. INTEGERS: Electron. J. Comb. Number Theory **21**(A.36) (2021)