algorithms

MDPI

Article

Branching Densities of Cube-Free and Square-Free Words

Elena A. Petrova and Arseny M. Shur *

Department of Algebra and Fundamental Informatics, Ural Federal University, 620075 Yekaterinburg, Russia; elena.petrova@urfu.ru
* Correspondence: arseny.shur@urfu.ru

Abstract: Binary cube-free language and ternary square-free language are two “canonical” representatives of a wide class of languages defined by avoidance properties. Each of these two languages can be viewed as an infinite binary tree reflecting the prefix order of its elements. We study how “homogenious” these trees are, analysing the following parameter: the density of branching nodes along infinite paths. We present combinatorial results and an efficient search algorithm, which together allowed us to get the following numerical results for the cube-free language: the minimal density of branching points is between $3509/9120 \approx 0.38476$ and $13/29 \approx 0.44828$, and the maximal density is between 0.72 and $67/93 \approx 0.72043$. We also prove the lower bound $223/868 \approx 0.25691$ on the density of branching points in the tree of the ternary square-free language.

Keywords: cube-free word; power-free word; prefix tree

MSC: 68R15; 68Q45

check for updates

Citation: Petrova, E.A.; Shur, A.M. Branching Densities of Cube-Free and Square-Free Words. *Algorithms* **2021**, *14*, 126. https://doi.org/10.3390/a14040126

Academic Editor: Sebastian Maneth

Received: 25 March 2021
Accepted: 18 April 2021
Published: 20 April 2021

1. Introduction

A formal language, which is a subset of the set of all finite words over some (usually finite) alphabets, is one of the most common objects in discrete mathematics and computer science. Languages are often defined by properties of their elements, and many “good” properties are hereditary—all factors (=contiguous subwords) of a word with such a property also possesses this property. Typical hereditary properties are “to be a factor of a certain infinite word” or “to contain no factors from a given set”. A factorial language forms posets under some natural order relations; the relation “to be a prefix of” is probably the simplest relation of this sort. The diagram of this relation is called a *prefix tree;* the structure of this tree reflects the properties of the language. For example, the prefix tree of a language L can be viewed as a deterministic (finite or infinite) automaton accepting L: each edge has the form (w, wa) and is labeled by the letter a, the root is the initial state, all nodes are final states.

An important class of factorial languages is constituted by *power-free languages*. Any language of this class contains no factors from the set of α-powers for a certain integer or rational α; an α-power of a nonempty word u is the prefix of an infinite word $uuu\cdots$ of length $\lceil\alpha|u|\rceil$, where $|u|$ stands for the length of u. Power-free languages are studied in hundreds of papers starting with the seminal work by Thue [1], but the topic still contains a number of challenging open problems. One group of problems concerns the structure of prefix trees of infinite power-free languages. Let us briefly recall the related known results. In the following text, the *subtree* of a prefix tree means a tree consisting of some node w and all its descendants.

For infinite power-free languages, there is a natural partition into “small” and “big” [2–8]: in binary languages avoiding small powers (up to $7/3$), the number of words grows only polynomially with length, while all other infinite power-free languages are conjectured to have exponential growth. This conjecture has been proved [4–8] for almost all power-free languages (up to a finite number of cases). Polynomial-size binary power-free languages possess several distinctive properties (see, e.g., [9] Section 2.2); all properties stem from a close relation of all words from these languages to a single infinite word, called the *Thue–Morse word* [10]. Among these languages, the *overlap-free language*, avoiding all α-powers with $\alpha > 2$, attracted the most attention; as a result, it is studied very well. For example, the asymptotic order of growth for this language is computed exactly [11,12]. Further, it is decidable whether a subtree of the prefix tree, rooted at a given word w, is finite or infinite [13]. Moreover, the results of [14] imply that the depth of a finite subtree can be computed in time linear in $|w|$, and the isomorphism of two given subtrees can be decided in linear time also. Most of the results about the overlap-free language can be extended, with additional and sometimes tedious technicalities, to all small binary power-free languages (see, e.g., [4]).

The knowledge about "big" power-free languages is rather limited. For all these languages, any subtree has at least one leaf [15]. Further, for any fixed alphabet and fixed integer α, it is decidable whether a given word generates finite or infinite subtrees, and every infinite subtree often branches out infinitely [16,17]. All other results concern two particular languages: the binary 3-free (*cube-free*) language CF and the ternary 2-free (*square-free*) language SF. These two languages are the most interesting "test cases", the analysis of which was initiated by Thue [1]. Note that the prefix tree of SF is binary in spite of the ternary alphabet, because a square-free word has no factors of the form aa. For the prefix tree of SF, it is known that (a) finite subtrees of arbitrary depth exist and can be built efficiently [18], (b) in any infinite path, the fraction of nodes with two children is at least $2/9$ [19], and (c) if a node of depth n has a single descendant of depth $n+m$, then $m = O(\log n)$ [19]. If we take the tree consisting of all infinite branches of the original prefix tree, then the analog of (c) with the bound $m = O(n^{2/3})$ is known [20,21]. For the prefix tree of CF, the property (a) was proved in [22]. The properties (b) (with the constant $23/78$) and (c) were proved in [23].

In this paper, we study the branching of prefix trees, continuing the line of research related to the property (b). Most of our results are about the language CF. By *branching point*, we mean a node of the prefix tree with two children. *Branching density* of an infinite path $\mathbf{w}$ is the limit of the ratios between the numbers of branching points in prefixes of $\mathbf{w}$ and lengths of these prefixes; if no limit exists, we consider lower/upper density as the corresponding lim inf / lim sup. Speaking about lower/upper bounds for density, we mean lower bounds for lower density and upper bounds for upper density. Our contribution is as follows:

- We establish the lower bound $3509/9120 \approx 0.38476$ on the branching density in the prefix tree of CF (Theorem 3) and the lower bound $223/868 \approx 0.25691$ on the branching density in the prefix tree of SF (Theorem 4), significantly improving the bounds from [19,23];
- We construct infinite paths in the prefix tree of CF with the branching density as small as $13/29 \approx 0.44828$ (Theorem 5);
- We establish the upper bound $67/93 \approx 0.72043$ on the branching density in the prefix tree of CF and construct infinite paths, with the branching density as big as $18/25 = 0.72$ (Theorem 6).

Let us comment on the results. The proof of each of the lower bounds consists of two parts: one is purely combinatorial, while the other requires a computer search. For the cube-free language, we significantly improve the combinatorial part (Theorem 1) over the paper [23], correcting, on the way, an error in the technical statement [23] (Theorem 7); as to the search part, we present an efficient (quadratic) algorithm replacing an exponential algorithm of [23]. There is a chance that the new bound can be slightly improved if more computational resources will be used. We also use the same search algorithm to improve the bound for the square-free language; the combinatorial part, presented in [19], is much simpler than for the cube-free case, and we see no way to improve it.

As a byproduct of the search algorithm, we find "building blocks" to construct an infinite path with small branching density. We call it small because it is smaller than the fraction of branching points at the nth level of the tree for each n that is big enough.

(See Section 4.2 for the details.) Finally, a separate combinatorial argument allows us to obtain an upper bound on the branching density for the cube-free case and present an example which is very close to this bound.

After preliminaries, we state and prove the results in Sections 3–5. In Section 3, we prove Theorem 1, which constitutes the combinatorial part of Theorem 3. The tools for the search part are described in Section 4.1. Section 4.2 presents the results of the search, Theorems 3 and 4, and a short discussion. Section 4.3 is devoted to Theorem 5. Finally, Section 5 contains Theorem 6 and its proof.

2. Preliminaries

We study words and languages over the binary alphabet $\{a, b\}$ (apart from Section 4.2, where the result over a ternary alphabet is also presented), writing λ for the empty word and $|w|$ for the length of a finite word w. If $w = xyz$ for some words x, y, and z (any of which can be empty), then x, y, z are, respectively, a *prefix*, a *factor*, and a *suffix* of w. We write $y \subseteq w$ to indicate that y is a factor of w. The set of all finite (nonempty finite, infinite) words over an alphabet Σ is denoted by Σ^* (resp., Σ^+, Σ^∞). Elements of Σ^+ (Σ^∞) are treated as functions $w : \{1, \ldots, n\} \to \Sigma$ (resp., $\mathbf{w} : \mathbb{N} \to \Sigma$). We write $[i..j]$ for the range $i, i{+}1, \ldots, j$ of positive integers; the notation $w[i..j]$ stands for the factor of the word w occupying this range, as well as for the particular occurrence of this factor in w; $w[i..i] = w[i]$ is just the ith letter of w. A factor $w[i..j]$ is *internal* if $i > 1$ and $j < |w|$. By *position* of a factor, we mean its starting (=leftmost) position. The distance between factors of a word is the difference of their positions; for example, the distance between occurrences of *aa* in *aabaa* is 3. A *cyclic shift* of a finite word w is any word $w[i..|w|]w[1..i{-}1]$. The *complement* of a finite or infinite word w is the image of w under the map which replaces all a's by b's and all b's by a's.

A word w has *period* $p < |w|$ if $w[1..|w|{-}p] = w[p{+}1..|w|]$. We use two basic properties of periodic words (see, e.g., [24]).

Lemma 1 (Lyndon, Schützenberger; [25])**.** *If $uv = vw \neq \lambda$, then there are words $x \neq \lambda$, y and an integer n such that $u = xy, w = yx$ and $v = (xy)^n x$.*

Lemma 2 (Fine, Wilf; [26])**.** *If a word u has periods p and q and $|u| \geq p + q - \gcd(p, q)$, then u has period $\gcd(p, q)$.*

The *prefix tree* of a language L is a directed tree, whose set of nodes is the set of all prefixes of words from L, and the set of edges consists of all pairs (u, uc) such that c is a letter. Edges are labelled by the last letter of the destination node: $u \xrightarrow{c} uc$. The only node having no incoming edges, and thus the root of the tree, is λ. A prefix tree is (in)finite whenever L is (in)finite. A finite prefix tree is often considered as a finite automaton and called *trie*.

A *cube* is a nonempty word of the form uuu. A word is *cube-free* if it has no cubes as factors; a cube is *minimal* if it contains no *other* cubes as factors. A *p-cube* is a minimal cube with the minimal period p (i.e., $|u| = p$). Other important repetitions include squares (words of the form uu) and overlaps (words having a period strictly smaller than half of their length).

The language CF of binary cube-free words is infinite and can be represented by its prefix tree $\mathcal{T}$, in which the nodes are precisely all cube-free words. The label of every path from the root coincides, as a word, with the terminal node of this path. A node in $\mathcal{T}$ is either a leaf (infinite paths contain no leaves), or has a single child (*fixed* node; its outgoing edge, the letter labeling this edge, and the position of this letter in the label of the path are also called *fixed*), or has two children (*branching point*; the outgoing edges, and their labels and positions are called *free*). A fragment of $\mathcal{T}$ is shown in Figure 1.

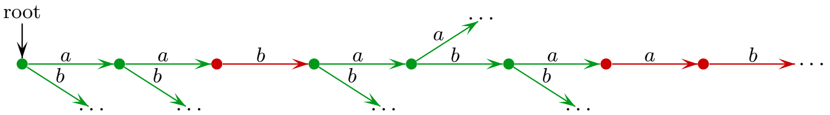

Figure 1. A fragment of the prefix tree of the binary cube-free language CF. Branching nodes and free edges are green, while fixed nodes and fixed edges are red. Nodes can be restored from the labels of paths.

To estimate the number of branching nodes in a path, we obtain bounds on the number of fixed positions/letters in the corresponding word. Assume that some position i in a cube-free word w is fixed; w.l.o.g., $w[i] = a$. Then the word $w[1..i{-}1]b$ ends with a (unique) p-cube; in this case, we say that i (or $w[i]$) is *fixed by a p-cube*. We assume that some constant h is chosen (we will choose it later) and we partition fixed positions in words into two groups: those fixed by "small" p-cubes with $p < h$ and those fixed by "big" p-cubes with $p \geq h$. To get the lower bound on the branching density, we establish separate upper bounds on the numbers of positions fixed by small and big cubes. All other results involve small cubes only.

3. Positions Fixed by Big Cubes

The aim of this section is to prove the following upper bound on the density of positions fixed by big cubes in a cube-free word.

Theorem 1. *For any integer $h \geq 2$ and any infinite cube-free word* $\mathbf{w}$*, the density of positions fixed by cubes with periods $\geq h$ in* $\mathbf{w}$ *is at most $\frac{6}{5h}$.*

Theorem 1 is based on the following result, describing the restrictions on the cubes of similar length fixing closely located letters.

Theorem 2. *Suppose that $t, l \geq 1$, $p, q \geq 2$ are integers, w is a word of length $t{+}l$ such that $w[1..t{+}l{-}1]$ is cube-free, the position t is fixed by a p-cube, and w ends with a q-cube. Then q is outside the red zone in Figure 2.*

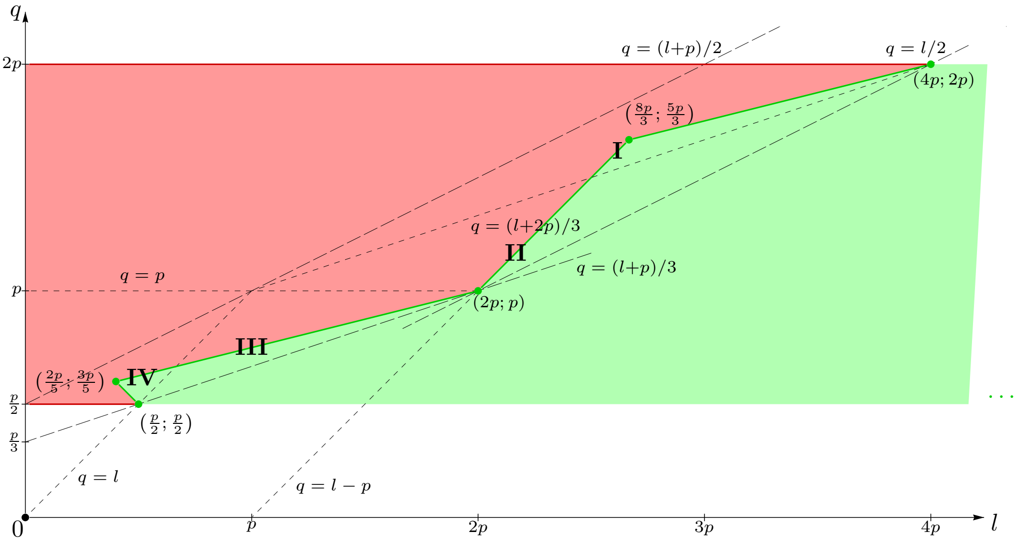

Figure 2. The restrictions on fixed positions in a cube-free word. If t is fixed by a p-cube and $(t{+}l)$ is fixed by a q-cube, then q (as a function of l with parameter p) must be outside the red polygon, including red border lines. The cases $q > 2p$ and $q < p/2$ are not considered.

Remark 1. *Theorem 2 and Figure 2 improve and correct their earlier analogs, Theorem 7 and Figure 1 of [23]. The improvement can be seen as a few additional red patches in Figure 2 w.r.t.*

to ([23] Figure 1), and the correction is that the triangle with vertices $(2p, p)$, $(4p, 2p)$, and $(5p, 2p)$ in ([23] Figure 1) should have been painted in green. This error does not affect the proofs of the main results of [23]: in those proofs, only a part of the red area is used. This part is drawn in ([23] Figure 8) and is strictly inside the red area in our Figure 2. Thus, in [23], only Theorem 7 and Remark 8 are (partially) incorrect.

Remark 2. *We believe that the boundary of the red area in Figure 2 is exact for $p/2 \le q \le 2p$, and so the result of Theorem 2 is optimal. We do not prove this claim, because it is not important for the aims of this paper. To substantiate the claim, we provide Table 1 with the examples of the words w corresponding to green points in the corners of the boundary in Figure 2.*

Table 1. Example words w with the pair (l, q) in the green corner of the red zone boundary in Figure 2. One can take $X = abbaab$ or a longer overlap-free word of similar structure.

Point	**Word w (PPP' is bold, QQQ is underlined)**
$l = p/2$ $q = p/2$	$\cdots b\,\boldsymbol{Xb\,Xa\,Xb\,Xa}\,\underline{\boldsymbol{Xb\,Xb}\,Xb}$
$l = 2p/5$ $q = 3p/5$	$\cdots b\,\boldsymbol{Xb\,Xb\,Xa\,Xb\,Xa\,Xb\,Xb\,Xa}\,\underline{\boldsymbol{Xb\,Xa\,Xb\,Xb\,Xa\,Xb\,Xb}\,Xa\,Xb}$
$l = 2p$ $q = p$	$\cdots b\,\boldsymbol{Xa\,Xa}\,\underline{\boldsymbol{Xb}\,Xb\,Xb}$
$l = 8p/3$ $q = 5p/3$	$\cdots b\,\boldsymbol{Xb\,Xa}\,\underline{\boldsymbol{Xa\,Xb\,Xa\,Xa\,Xb\,Xa\,Xb}\,Xa\,Xa\,Xb\,Xa\,Xb\,Xa\,Xa\,Xb}$
$l = 4p$ $q = 2p$	$\cdots b\,\boldsymbol{Xa}\,\underline{\boldsymbol{Xa\,Xb}\,Xa\,Xb\,Xa\,Xb}$

Proof of Theorem 2. Let $P = w[t{-}3p{+}1..t{-}2p]$, $P' = w[t{-}p{+}1..t]$. Then, $w[t{-}3p{+}1..t] = PPP'$. W.l.o.g. P ends with a; so $w[t-2p] = w[t-p] = a$, $w[t] = P'[p] = b$. We write QQQ for the q-cube, which is a suffix of w. We begin with a few observations.

O1. If $q = p/2$, the condition $w[t-p] \ne w[t]$ implies that the suffix QQQ of w does not contain $w[t-p]$. Hence, $l \ge 3q - p = p/2$, giving us the red segment of the line $q = p/2$ in Figure 2. To get the red parts of the lines $q = p$ and $q = 2p$, note that for $q = p$, the same argument gives $l \ge 2p$, and for $q = 2p$, the condition $w[t-2p] \ne w[t]$ implies $l \ge 4p$. From now on, we assume $p/2 < q < 2p$ and $q \ne p$.

O2. Let i be the bigger of the positions of PPP' and QQQ in w and consider the factor $w[i..t{-}1]$, having both periods p and q. If its length $t - i$ is big enough to apply Lemma 2, the words P, Q are integral powers of shorter words, contradicting the condition that $w[1..t{+}l{-}1]$ is cube-free. Thus, Lemma 2 must be inapplicable, giving us $t - i < p + q - \gcd(p, q) \le 3p - 2$ (recall that $q < 2p$). Hence, the position of QQQ in w is bigger than the position of PPP', implying $QQQ = w[i..t{+}l]$, so that $3q = t + l - i + 1 < l + 1 + p + q - \gcd(p, q)$. From this, $l > 2q - p$, meaning that all green points with $q < 2p$ are strictly below the line $q = \frac{l+p}{2}$ shown in Figure 2.

O3. If the factor $w[i..t{-}1]$ considered in O2 is shorter than $\max\{p, q\}$, then we are unable to restrict q: the p-periodic factor PPP' and the q-periodic suffix QQQ have too short an overlap to "interact" inside w. Recall that $t - i = 3q - l - 1$, so all are strictly above the lines $q = \frac{l+p}{3}$ and $q = \frac{l}{2}$ in Figure 2.

Thus, to prove the theorem, it remains to justify the colouring of the stripe between the line $q = \frac{l+p}{2}$ and the broken line $\{q = \frac{l+p}{3}; q = \frac{l}{2}\}$ in Figure 2. We split this stripe into zones I–IV by the lines $q = l$, $q = p$, and $q = \frac{l+2p}{3}$. The arguments for all zones are very similar, so we provide maximum details for zone I and more concise proofs for zones II–IV.

Zone I: $q > \frac{l+2p}{3}$. Together with $q < 2p$ (O1) and $2q < l + p$ (O2), this gives the mutual location of the suffix QQQ and factor PPP' of w, as depicted in Figure 3. Equal letters denote equal factors; note that $x \ne \lambda$ since $2q < l + p$ and $z \ne \lambda$, since $q < 2p$.

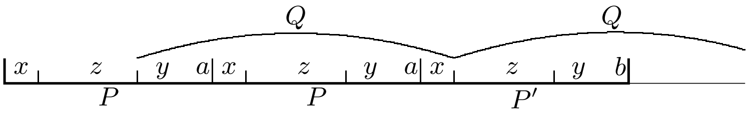

Figure 3. Location of factors of w for Zone I: $q > \frac{l+2p}{3}$, $q < 2p$, $2q < l + p$. The leftmost Q consists of a suffix of P, followed by P and a prefix of P; $P = xzya$ is partitioned accordingly.

The words y, zy, and $yaxz$ are prefixes of Q (Figure 3). By the length argument, y is a prefix of zy, which is a prefix of $yaxz$. Then $zyaxz \subseteq PP$ (Figure 3) implies $zzy \subseteq PP$. Since PP is cube-free, $zzz \not\subseteq PP$, and thus z is not a prefix of y. Since z and y are both prefixes of Q, we have $z = yz'$, $z' \neq \lambda$. Further, $yaxyaxz \subseteq QQ$ (Q begins with $yaxz$ and ends with yax) but $(yax)^3 \not\subseteq QQ$, because QQ is cube-free. Then, the fact that z and yax are both prefixes of Q implies that $z = yz'$ is a proper prefix of yax, so $ax = z'x'$, $x' \neq \lambda$. Now compare $zy = yz'y$ against $yaxz = yz'x'yz'$. We see that y is a proper prefix of $x'y$ by the length argument. By Lemma 1 we can write $x' = fg, y = (fg)^n f$ for some words f, g; note that $n \leq 1$ since $x'y$ is cube-free. If $n = 1$, we have

$$Q = yaxzyax = yz'x'yz'yz'x' = fgfz'fgfgfz'fgfz'fg, \text{ and then}$$
$$(fgfz'fg)^3 \subseteq QQ = fgfz'fgfgfz'\boldsymbol{fgfz'fg\ fgfz'fgfgfz'fg}fz'fg.$$

However, QQ is cube-free, so $n = 0$, implying $y = f$, $x' = yg$. Finally, we can write $Q = yz'ygyz'yz'yg$. Note that $g \neq \lambda$, otherwise $(yz'y)^3 \subseteq QQ$. From this representation of Q we can express q, p, and l in terms of $|y|$, $|z'|$, and $|g|$; from Figure 3 we know that $l = 2q - |yz'yb|$. Thus, we have

$$\begin{array}{rclllr} p & = & 3|y|+ & 2|z'|+ & |g| & , \\ q & = & 5|y|+ & 3|z'|+ & 2|g| & , \\ l & = & 8|y|+ & 5|z'|+ & 4|g| & -1. \end{array} \tag{1}$$

Recall that $|y| \geq 0$, $|z'|, |g| \geq 1$. From (1) we get $q = l - p - |g| + 1 \leq l - p$, $q = p + \frac{l-|z'|+1}{4} \leq p + \frac{l}{4}$, and also $q > \frac{l+2p}{3}$ (the border of Zone I). This gives us exactly the green triangle inside Zone I with the vertices $(\frac{5}{2}p, \frac{3}{2}p)$, $(\frac{8}{3}p, \frac{5}{3}p)$, $(4p, 2p)$.

Zone II: $q \leq \frac{l+2p}{3}$ and $q > p$. Together with $2q > l$ (O3), this gives the mutual location of the suffix QQQ and factor PPP' of w, as depicted in Figure 4 ($y \neq \lambda$ since $q > p$; z or x can be empty).

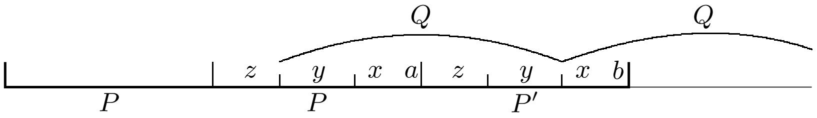

Figure 4. Location of factors of w for Zone II: $q \leq \frac{l+2p}{3}$, $q > p$, $2q > l$. The leftmost Q consists of a suffix of P, followed by a longer prefix of P; $P = zyxa$ is partitioned accordingly.

Since xb and yx are prefixes of Q and y is a suffix of Q (Figure 4), one has $yyx \subseteq QQ$. As QQ is cube-free, y is not a prefix of x. Comparing the prefixes xb and yx of Q, we have $y = xby'$ for some (possibly empty) y'. Then, $Q = xby'xazxby'$, $P = zxby'xa$. We express p, q, and $l = 2q - |xb|$ in terms of $|x|, |y'|, |z|$:

$$\begin{array}{rclll} p & = & 2(|x|+1)|+ & |y'|+ & |z|, \\ q & = & 3(|x|+1)|+ & 2|y'|+ & |z|, \\ l & = & 5(|x|+1)|+ & 4|y'|+ & 2|z|. \end{array} \tag{2}$$

From (2), we get $q = l - p - |y'| \leq l - p$; together with the boundaries of Zone II, the line $q = l - p$ forms the green triangle inside Zone II with the vertices $(2p, p)$, $(\frac{5}{2}p, \frac{3}{2}p)$, $(4p, 2p)$ (Figure 2).

Zone III: $q \le l, q < p$. Together with $q > \frac{l+p}{3}$ (O3), this gives the mutual location of the suffix QQQ and factor PPP' of w, as depicted in Figure 5 ($v, z \neq \lambda$ since $q < p$; x, y can be empty).

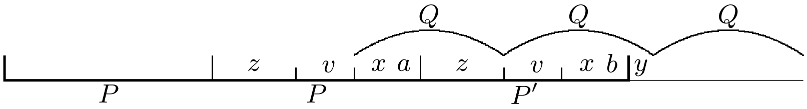

Figure 5. Location of factors of w for Zone III: $q < p, q \le l, 3q > l + p$. The leftmost Q consists of a suffix of P, followed by a shorter prefix of P; the middle Q ends with some suffix y outside P', possibly empty; $P = zvxa$ is partitioned accordingly.

Since xa and vx are prefixes of Q and $vQ \subseteq PP$ (Figure 5), one has $vvx \subseteq PP$, so v is not a prefix of x, and thus $v = xav'$ for some (possibly empty) v'. Then $z = v'xby$, $Q = xav'xby$, $P = v'xbyxav'xa$. We express p, q, and $l = q + |y|$ in terms of $|x|, |v'|, |y|$:

$$\begin{array}{rclll} p & = & 3(|x|+1)+ & 2|v'|+ & |y|, \\ q & = & 2(|x|+1)+ & |v'|+ & |y|, \\ l & = & 2(|x|+1)+ & |v'|+ & 2|y|. \end{array} \tag{3}$$

From (3), we get $q = \frac{l+2p-|v'|}{4} \le \frac{l+2p}{4}$; together with the boundaries of Zone III, this line forms the green triangle in Zone III with the vertices $(\frac{p}{2}, \frac{p}{2}), (\frac{2}{3}p, \frac{2}{3}p), (2p, p)$ (Figure 2).

Zone IV: $q > l$. One has $q > p/2$ (O1) and $2q < l + p$ (O2), and so, $q < p$. Then the mutual location of the suffix QQQ and factor PPP' of w is as in Figure 6 ($x \neq \lambda$ since $2q > p$; $v \neq \lambda$ since $q < p$; $z \neq \lambda$; since $2q < l + p$; y can be empty).

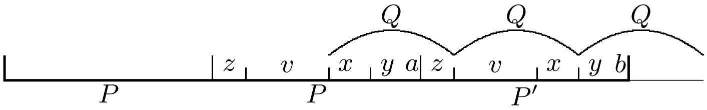

Figure 6. Location of factors of w for Zone IV: $q > p/2, q > l, 2q < l + p$. The leftmost Q consists of a suffix of P, followed by a shorter prefix of P; the middle Q is a proper factor of P; $P = zvxa$ is partitioned accordingly.

Since x, vx are prefixes of Q and $vQ \subseteq PP$ (Figure 6), one has $vvx \subseteq PP$, so $vvx \subseteq PP$, v is not a prefix of x, and thus $v = xv'$ for some $v' \neq \lambda$. Taking y and xy, which are another pair of prefixes of Q, we get $x = ybx'$ (x' is possibly empty) because $xxy \subseteq QQ$. Note that if v' is a prefix of x, then $(xv')^3 \subseteq xv'xv'xx \subseteq vQQ \subseteq PP$, which is impossible. Thus, v' is not a prefix of x and then of y. Since xv' and xya are prefixes of Q, we get $v' = yag$ for some (possibly empty) g. Thus, $Q = ybx'yagybx'$, $P = gybx'ybx'yagybx'ya$. We express p, q, and $l = q - |yb|$ in terms of $|x'|, |g|, |y|$:

$$\begin{array}{rclll} p & = & 5(|y|+1)+ & 3|x'|+ & 2|g|, \\ q & = & 3(|y|+1)+ & 2|x'|+ & |g|, \\ l & = & 2(|y|+1)+ & 2|x'|+ & |g|. \end{array} \tag{4}$$

From (4), we get $q = \frac{l+2p-|g|}{4} \le \frac{l+2p}{4}$ and $q = p - l + |x'| \ge p - l$; together with the boundary $q = l$, the obtained two lines form the green triangle in Zone IV with the vertices $(\frac{p}{2}, \frac{p}{2}), (\frac{2}{5}p, \frac{3}{5}p), (\frac{2}{3}p, \frac{2}{3}p)$ (Figure 2).

Thus, we identified all "red" and "green" parts of the areas I–IV, getting the full picture from Figure 2. Theorem 2 is proved. □

The second crucial step in the proof of Theorem 1 is the following lemma on the density of positions fixed by cubes with similar periods.

Lemma 3. *Suppose that $l \geq 1$, $p \geq 2$ are integers, and w is a cube-free word such that $|w| > l$. Among any l consecutive letters of w, those less than $\frac{8}{5} + \frac{3l}{5p}$ are fixed by cubes with periods in the range $[p..2p{-}1]$.*

Proof. Let us consider an inverse problem:

($\star$) Let $l_0 < l_1 < \cdots < l_s$ $(s \geq 1)$ be positions in w containing letters fixed by cubes with periods $q_0, \ldots, q_s$, respectively, where $q_i \in [p..2p{-}1]$ for all i; find a lower bound for $l_s - l_0$ (as a function of s and p) which applies for every sequence $q_0, \ldots, q_s$.

The distance between each two consecutive position l_i and l_{i+1} is lower-bounded by Theorem 2. More precisely, we use Theorem 2 to make conclusions of the form

- $l_{i+1} - l_i \geq A$, where the point (A, q_{i+1}) is on the border of the red polygon in Figure 2, in which $p = q_i$, $l = l_i$, and $q = q_{i+1}$.

For example, since the point $(25, 15)$ is on the segment $q = l - p$ of the border of such a polygon built for $p = 10$, we conclude that the condition $q_i = 10, q_{i+1} = 15$ implies $l_{i+1} - l_i \geq 25$. Let $\beta = q_i$, $\alpha = q_{i+1}$, $l = l_{i+1} - l_i$. Then Theorem 2 implies the following inequalities related to the boundaries of the polygon in Figure 2 (β and α play the roles of p and q, respectively):

$$l \geq 4\alpha - 4\beta \qquad \text{if} \quad \alpha \geq \tfrac{5}{3}\beta \tag{5a}$$

$$l \geq \alpha + \beta \qquad \text{if} \quad \beta \leq \alpha \leq \tfrac{5}{3}\beta \tag{5b}$$

$$l \geq 4\alpha - 2\beta \qquad \text{if} \quad \tfrac{3}{5}\beta \leq \alpha \leq \beta \tag{5c}$$

$$l \geq \beta - \alpha \qquad \text{if} \quad \alpha \leq \tfrac{3}{5}\beta. \tag{5d}$$

Assume that $\vec{q} = (q_0, \ldots, q_s)$ is a sequence of positive rational numbers such that

$$\max_{i \in [0..s]} q_i < 2 \min_{i \in [0..s]} q_i.$$

We define its *span* $\mathsf{span}(\vec{q})$ as the lower bound for the difference $l_s - l_0$ for the sequence $\vec{q}$ of periods. Precisely, $\mathsf{span}(\vec{q})$ is the minimum number such that there exists a sequence $0 = l_0 < l_1 < \cdots < l_s = \mathsf{span}(\vec{q})$ satisfying, for each i, the property "the point $(l_{i+1}{-}l_i, q_{i+1})$ is on the border of the red polygon in Figure 2, in which q_i is substituted for p". Thus, $\min \mathsf{span}(\vec{q})$, where the minimum is taken over all sequences of length $s{+}1$ in the given range $[p..2p{-}1]$, is the lower bound sought in ($\star$).

We write $\mathsf{span}(q_i, \ldots, q_j)$ for the span of the corresponding subsequence of $\vec{q}$. Note that spans are additive: $\mathsf{span}(q_i, \ldots, q_j) + \mathsf{span}(q_j, \ldots, q_m) = \mathsf{span}(q_i, \ldots, q_m)$. For the simplest case of a two-element sequence, (5a)–(5d) imply

$$\mathsf{span}(\beta, \alpha) = \begin{cases} 4\alpha - 4\beta, & \text{if } \alpha \geq \frac{5}{3}\beta \\ \alpha + \beta, & \text{if } \beta \leq \alpha \leq \frac{5}{3}\beta \\ 4\alpha - 2\beta, & \text{if } \frac{3}{5}\beta \leq \alpha \leq \beta \\ \beta - \alpha, & \text{if } \alpha \leq \frac{3}{5}\beta \end{cases}. \tag{6}$$

From (6), we immediately have

($*$) for any fixed β, the function $\mathsf{span}(\beta, \alpha)$ monotonically increases for $\alpha \in [\frac{3}{5}\beta, 2\beta)$.

Since all borders in Figure 2 are line segments, the equality $\mathsf{span}(C\vec{q}) = C \cdot \mathsf{span}(\vec{q})$ holds for any $C > 0$ (if a sequence $(l_0, \ldots, l_s)$ works for $\vec{q}$, then $(Cl_0, \ldots, Cl_s)$ works for $C\vec{q}$). Thus, we simplify the subsequent argument by considering a particular range for the sequence $\vec{q}$: q_i belongs to the semiclosed interval $[1, 2)$ for all i.

Given $\vec{q}$, we iteratively modify it from right to left. Each modification results in a sequence of the same length, in the same range, and with the same or a smaller span; the result of the last modification is one of "good" sequences, the span of which can be easily computed. The smallest span of a "good" sequence is the lower bound for the span of any sequence $\vec{q}$ in the given range. Precise definitions are as follows. We call a sequence $(r_0, \dots, r_t)$ *canonical* if it contains only numbers 1 and $\frac{5}{3}$ in a way that no two $(\frac{5}{3})$'s are consecutive and, in addition, $r_0 = r_t = 1$. A sequence $\vec{q} = (q_0, \dots, q_s)$ is *good* if it has a nonempty canonical suffix beginning at q_0, q_1, or q_2.

We transform an arbitrary sequence into a good one with local transformations changing either one element or two consecutive elements of a sequence. Note that if we change, say, q_i and q_{i+1}, this affects $\mathsf{span}(q_{i-1}, q_i, q_{i+1}, q_{i+2})$ but preserves $\mathsf{span}(q_1, \dots, q_{i-1})$ and $\mathsf{span}(q_{i+2}, \dots, q_s)$. By $(*)$ and (6) one has

$$\text{for } \beta \le \tfrac{5}{3} : \mathsf{span}(\beta, \alpha) \ge \mathsf{span}(\beta, 1) = 4 - 2\beta;$$
$$\text{for } \beta \ge \tfrac{5}{3} : \mathsf{span}(\beta, \alpha) \ge \mathsf{span}(\beta, \tfrac{3}{5}\beta) = \tfrac{2}{5}\beta.$$

These inequalities justify the first transformation rule:

T1: given a sequence $(q_0, \dots, \beta, \alpha)$, replace α by 1 if $\beta \le \frac{5}{3}$ and by $\frac{3}{5}\beta$ otherwise.

Next, we consider the span of a triple $(\gamma, \beta, \frac{3}{5}\beta)$ as a function of β. Here, $\beta \ge \frac{5}{3}$, so $\mathsf{span}(\gamma, \beta) \ge \mathsf{span}(\gamma, \frac{5}{3})$ by $(*)$. Since $\mathsf{span}(\beta, \frac{3}{5}\beta) = \frac{2}{5}\beta \ge \frac{2}{3} = \mathsf{span}(\frac{5}{3}, 1)$, we have

$$\mathsf{span}(\gamma, \beta, \frac{3}{5}\beta) = \mathsf{span}(\gamma, \beta) + \mathsf{span}(\beta, \frac{3}{5}\beta) \ge \mathsf{span}(\gamma, \frac{5}{3}, 1).$$

Further, compare $\mathsf{span}(\gamma, \frac{5}{3}, 1)$ to $\mathsf{span}(\gamma, 1, 1)$. For $\gamma \ge \frac{5}{3}$ we obtain, by (6),

$$\mathsf{span}(\gamma, \tfrac{5}{3}, 1) = \mathsf{span}(\gamma, \tfrac{5}{3}) + \mathsf{span}(\tfrac{5}{3}, 1) = (4 \cdot \tfrac{5}{3} - 2\gamma) + \tfrac{2}{3} > 3$$
$$> (\gamma - 1) + 2 = \mathsf{span}(\gamma, 1, 1).$$

For $\gamma \le \frac{5}{3}$, (6) gives us $\mathsf{span}(\gamma, \frac{5}{3}, 1) = \gamma + \frac{7}{3}$ and $\mathsf{span}(\gamma, 1, 1) = 6 - 2\gamma$. The first number is bigger (resp., smaller) if γ is bigger (resp., smaller) than $\frac{11}{9}$. Therefore, we justified the second transformation rule:

T2: given a sequence $(q_0, \dots, \gamma, \beta, \frac{3}{5}\beta)$, replace $(\beta, \frac{3}{5}\beta)$ by $(1, 1)$ if $\gamma \ge \frac{11}{9}$ and by $(\frac{5}{3}, 1)$ otherwise.

Rules T1 and/or T2 replace the last number in the processed sequence $\vec{q}$ by 1 and serve as the base case in transforming $\vec{q}$ into a good sequence. Now we describe the general case, assuming that $\vec{q}$ has a nonempty canonical suffix $(q_i, \dots, q_s)$. The subsequent transformations preserve the numbers $q_i, \dots, q_s$ and aim at extending the canonical suffix.

Consider the span of a triple $(\gamma, \beta, 1)$ as a function of β. By (6), for $\gamma \ge \frac{5}{3}$, we have

$$\mathsf{span}(\gamma, \beta) = \begin{cases} \gamma + \beta, \text{ if } \beta \ge \gamma; \\ 4\beta - 2\gamma, \text{ if } \frac{3}{5}\gamma \le \beta \le \gamma; \\ \gamma - \beta, \text{ if } \beta \le \frac{3}{5}\gamma; \end{cases} \qquad \mathsf{span}(\beta, 1) = \begin{cases} \beta - 1, \text{ if } \beta \ge \frac{5}{3}; \\ 4 - 2\beta, \text{ if } \beta \le \frac{5}{3}; \end{cases}$$

$$\mathsf{span}(\gamma, \beta, 1) = \mathsf{span}(\gamma, \beta) + \mathsf{span}(\beta, 1) = \begin{cases} 2\beta + \gamma - 1, \text{ if } \beta \ge \gamma; \\ 5\beta - 2\gamma - 1, \text{ if } \frac{5}{3} \le \beta \le \gamma; \\ 2\beta - 2\gamma + 4, \text{ if } \frac{3}{5}\gamma \le \beta \le \frac{5}{3}; \\ -3\beta + \gamma + 4, \text{ if } \beta \le \frac{3}{5}\gamma. \end{cases}$$

Thus, $\mathsf{span}(\gamma, \beta, 1)$ has a unique minimum at $\beta = \frac{3}{5}\gamma$. Similarly, for $\gamma \le \frac{5}{3}$ we have

$$\mathsf{span}(\gamma, \beta) = \begin{cases} 4\beta - 4\gamma, \text{ if } \beta \ge \frac{5}{3}\gamma; \\ \gamma + \beta, \text{ if } \gamma \le \beta \le \frac{5}{3}\gamma; \\ 4\beta - 2\gamma, \text{ if } \beta \le \gamma; \end{cases} \qquad \mathsf{span}(\gamma, \beta, 1) = \begin{cases} 5\beta - 4\gamma - 1, \text{ if } \beta \ge \frac{5}{3}\gamma; \\ 2\beta + \gamma - 1, \text{ if } \frac{5}{3} \le \beta \le \frac{5}{3}\gamma; \\ -\beta + \gamma + 4, \text{ if } \gamma \le \beta \le \frac{5}{3}; \\ 2\beta - 2\gamma + 4, \text{ if } \beta \le \gamma. \end{cases}$$

Here, $\mathsf{span}(\gamma, \beta, 1)$ has two local minima, $\gamma + \frac{7}{3}$ at $\beta = \frac{5}{3}$ and $6 - 2\gamma$ at $\beta = 1$. As we learned above, the first number is bigger (smaller) if γ is bigger (resp., smaller) than $\frac{11}{9}$. Now, the third transformation rule replaces β in a triple $(\gamma, \beta, 1)$ with a value minimizing $\mathsf{span}(\gamma, \beta, 1)$ (recall that canonical sequences begin with 1):

T3: given a sequence $(q_0, \ldots, \gamma, \beta, q_i, \ldots, q_s)$ with a canonical suffix $(q_i, \ldots, q_s)$, replace β by $\frac{3}{5}\gamma$ if $\gamma \geq \frac{5}{3}$; by $\frac{5}{3}$ if $\gamma \leq \frac{11}{9}$; and by 1 otherwise.

If the rule T3 replaces β by 1, the canonical suffix is extended. In the two remaining cases, we need additional rules. Consider the span of a triple $(\delta, \gamma, \frac{5}{3})$ as a function of γ, where $\gamma \leq \frac{11}{9}$. By (6), we have $\mathsf{span}(\gamma, \frac{5}{3}) = \gamma + \frac{5}{3}$ and

$$\mathsf{span}(\delta, \gamma) = \begin{cases} \delta + \gamma, \text{ if } \gamma \geq \delta; \\ 4\gamma - 2\delta, \text{ if } \frac{3}{5}\delta \leq \gamma \leq \delta; \\ \delta - \gamma, \text{ if } \gamma \leq \frac{3}{5}\delta. \end{cases} \qquad \mathsf{span}(\delta, \gamma, \tfrac{5}{3}) = \begin{cases} 2\gamma + \delta + \frac{5}{3}, \text{ if } \gamma \geq \delta; \\ 5\gamma - 2\delta + \frac{5}{3}, \text{ if } \frac{3}{5}\delta \leq \gamma \leq \delta; \\ \delta + \frac{5}{3}, \text{ if } \gamma \leq \frac{3}{5}\delta. \end{cases}$$

Hence, at $\gamma = 1$ the minimum is attained. Thus, the next transformation is correct:

T4: given a sequence $(q_0, \ldots, \delta, \gamma, \frac{5}{3}, q_i, \ldots, q_s)$ with a canonical suffix $(q_i, \ldots, q_s)$ and $\gamma \leq \frac{11}{9}$, replace γ by 1.

The application of T4 extends the canonical suffix of a sequence by two elements. Finally, consider a quadruple $(\delta, \gamma, \frac{3}{5}\gamma, 1)$. By (6), $\mathsf{span}(\delta, 1, \frac{5}{3}, 1) = \begin{cases} \delta + \frac{7}{3}, & \text{if } \delta \geq \frac{5}{3}; \\ \frac{22}{3} - 2\delta, & \text{if } \delta \leq \frac{5}{3}; \end{cases}$ and $\mathsf{span}(\gamma, \frac{3}{5}\gamma, 1) = 4 - \frac{4}{5}\gamma$. Then we study $\mathsf{span}(\delta, \gamma, \frac{3}{5}\gamma, 1)$ depending on δ:

$$\begin{aligned} \delta \geq \gamma : \ & \mathsf{span}(\delta, \gamma, \tfrac{3}{5}\gamma, 1) = \tfrac{16}{5}\gamma - 2\delta + 4 > \tfrac{16}{3} > \delta + {}_{7}3 \\ \tfrac{5}{3} \leq \delta \leq \gamma : \ & \mathsf{span}(\delta, \gamma, \tfrac{3}{5}\gamma, 1) = \tfrac{1}{5}\gamma + \delta + 4 > \delta + {}_{7}3 \\ \tfrac{3}{5}\gamma \leq \delta \leq \tfrac{5}{3} : \ & \mathsf{span}(\delta, \gamma, \tfrac{3}{5}\gamma, 1) = \tfrac{1}{5}\gamma + \delta + 4 \geq 2\gamma - 2\delta + 4 \geq \tfrac{22}{3} - 2\delta \\ \delta \leq \tfrac{3}{5}\gamma : \ & \mathsf{span}(\delta, \gamma, \tfrac{3}{5}\gamma, 1) = \tfrac{16}{5}\gamma - 4\delta + 4 \geq 2\gamma - 2\delta + 4 \geq \tfrac{22}{3} - 2\delta. \end{aligned}$$

In all cases, $\mathsf{span}(\delta, \gamma, \frac{3}{5}\gamma, 1) \geq \mathsf{span}(\delta, 1, \frac{5}{3}, 1)$, Thus, we have one more correct transformation rule which extends the canonical suffix:

T5: given a sequence $(q_0, \ldots, \delta, \gamma, \frac{3}{5}\gamma, q_i, \ldots, q_s)$ with a canonical suffix $(q_i, \ldots, q_s)$, replace $(\gamma, \frac{3}{5}\gamma)$ by $(1, \frac{5}{3})$.

Starting with an arbitrary sequence $\vec{q}$, we apply T1 and/or T2 to get a sequence with a nonempty canonical suffix. For any sequence with such a suffix preceded by three or more numbers, at least one of the transformations T3–T5 is applicable. Note that T3 either increases the length of the canonical suffix or makes one of T4, T5 applicable, while each of T4 and T5 increases this length. Thus, we eventually arrive at the situation where the canonical suffix of the current sequence $\vec{r} = (r_0, \ldots, r_s)$ is preceded by 0, 1, or 2 numbers, so that no other transformations are possible. If this suffix begins with r_2, then T3 and/or T4 was already applied, and then either $r_1 = \frac{5}{3}$ or $r_1 = \frac{3}{5}r_0$. In particular, $\vec{r}$ is a good sequence.

To find a good sequence of minimum span, we note that $\mathsf{span}(1,1,1) = 4 > \frac{10}{3} = \mathsf{span}(1, \frac{5}{3}, 1)$. Hence, a unique canonical sequence of odd length and minimum span is $(1, \frac{5}{3}, 1, \frac{5}{3}, \ldots, 1)$ and one of the canonical sequences of even length and minimum span is $(1, 1, \frac{5}{3}, 1, \frac{5}{3}, \ldots, 1)$. Now it is easy to find, using (6), good sequences of minimum span. Namely, for even (resp., odd) s, we have $\vec{r} = (\beta, \frac{3}{5}\beta, 1, \frac{5}{3}, 1, \frac{5}{3}, \ldots, 1)$ (resp., $\vec{r} = (\beta, \frac{3}{5}\beta, 1, 1, \frac{5}{3}, 1, \frac{5}{3}, \ldots, 1)$), where $\beta = 2 - \varepsilon$ for ε as small as possible. Thus, in the case of even (resp., odd) s, one has $\mathsf{span}(\vec{r}) = \frac{5}{3}s - \frac{14}{15} + \frac{4}{5}\varepsilon$ (resp., $\mathsf{span}(\vec{r}) = \frac{5}{3}s - \frac{3}{5} + \frac{4}{5}\varepsilon$). Thus, $\mathsf{span}(\vec{q}) > \frac{5}{3}s - 1$ for any sequence $\vec{q} = (q_0, \ldots, q_s)$ such that $q_i \in [1, 2)$ for all i.

Returning to the problem $(\star)$ we are solving, recall that $\mathsf{span}(p\vec{q}) = p \cdot \mathsf{span}(\vec{q})$, Thus, we have the lower bound $l_s - l_0 > (\frac{5}{3}s - 1)p$. This means, at most, $s{+}1$ letters fixed by cubes with periods in $[p..2p{-}1]$ among each $l = (\frac{5}{3}sp - p + 1)$ consecutive positions of w. Now, easy arithmetic gives $s{+}1 < \frac{8}{5} + \frac{3l}{5p}$, as required. □

Proof of Theorem 1. We split the range from h to infinity into disjoint finite ranges such that the kth range is $[2^{k-1}h..2^k h{-}1]$. Consider the density of positions in a cube-free word $\mathbf{w}$, fixed by p-cubes with p in the kth range. By Lemma 3 and the definition of density, it is upper-bounded by $\frac{3}{5\cdot 2^{k-1}h}$. Summing up the geometric series of all these upper bounds, we get the required bound $\frac{6}{5h}$. □

4. Positions Fixed by Small Cubes

4.1. Regular Approximations and Aho–Corasick Automata

To estimate the number of letters in a cube-free word that are fixed by small cubes, we analyze finite automata recognizing some approximations of the language CF. Let L_i be the language of all binary words containing no cubes of period $\leq i$. Then, L_i contains CF and is regular (as a language defined by a finite set of forbidden factors); L_i is referred to as ith *regular approximation* of CF. The study of regular approximations is a standard approach to power-free languages (see, e.g., [9] (Section 3)).

A regular language given by a finite set of forbidden factors can be represented by a *partial* deterministic finite automaton (dfa) built by a variation of the classical Aho-Corasick algorithm, as described in [27]. Let us provide some necessary details for regular approximations of CF, following a more general scheme from [28].

To get the dfa $\mathcal{A}_i$ accepting the language L_i, one proceeds in three steps.

1. List all p-cubes with periods $p \leq i$ and build the prefix tree $\mathcal{P}_i$ of these words; then, the leaves of $\mathcal{P}_i$ are exactly the p-cubes, and all internal nodes are cube-free words:
2. Consider $\mathcal{P}_i$ as a partial dfa with the initial state λ and complete this dfa, adding transitions by the Aho–Corasick rule: *if there is no transition from a state u by a letter c, add the transition $u \xrightarrow{c} v$, where v is the longest suffix of uc, present in $\mathcal{P}_i$;*
3. Delete all leaves of $\mathcal{P}_i$ from the obtained automaton.

The resulting partial dfa is $\mathcal{A}_i$; it accepts by any state and rejects if it cannot read the word. For details see, for example, [27]. Let us fix some $i \geq 1$ and analyze the properties of $\mathcal{A}_i$.

We write $u.v$ for the state of $\mathcal{A}_i$ reached from the state u by the path labelled by v. The following lemma connects $\mathcal{A}_i$ and fixed letters in cube-free words.

Lemma 4. *A letter $w[j]$ of a cube-free word w is fixed by a p-cube with $p \leq i$ if the state $\lambda.(w[1..j{-}1])$ of the dfa $\mathcal{A}_i$ has a single outgoing transition.*

Proof. W.l.o.g., $w[j] = a$. Since w is cube-free, the states $\lambda.(w[1..j{-}1])$ and $\lambda.(w[1..j])$ exist and are connected by an edge labelled by a. Let $w[j]$ be fixed by a p-cube; this means that $w[1..j{-}1]b$ ends with some p-cube uuu; since $p \leq i$, $\mathcal{P}_i$ has the leaf uuu. By the Aho–Corasick rule, the edge $\lambda.(w[1..j{-}1]) \xrightarrow{b} uuu$ was added to $\mathcal{P}_i$ (step 2 above) and then deleted together with the leaf uuu (step 3). Thus, the state $\lambda.(w[1..j{-}1])$ has a single outgoing transition. For the other direction, note that if $\lambda.(w[1..j{-}1])$ has the only transition to $\lambda.(w[1..j])$, then the state $\lambda.(w[1..j{-}1]b)$ was deleted at step 3. Hence, this state is some p-cube uuu with $p \leq i$. Since the Aho–Corasick rule implies that the state $\lambda.v$ is always a suffix of v, $w[1..j{-}1]b$ has the suffix uuu, whence the result. □

In accordance with the other notation, we call *fixed* the states of $\mathcal{A}_i$ with a single outgoing transition and the edges representing these transitions. The next lemma shows how to get an upper bound on the number of letters in a word, fixed by short cubes.

Lemma 5. *Let d_i and c_i be minimal numbers such that in the automaton $\mathcal{A}_i$ (a) for any m, every simple cycle of length m contains, at most, $d_i m$ fixed states, and (b) for any n, every simple path* *of length n contains at most $d_i n + c_i$ fixed states. Then, every cube-free word w contains, at most, $d_i|w| + c_i$ positions fixed by p-cubes with $p \leq i$.*

Proof. In $\mathcal{A}_i$, consider the walk W from λ to $\lambda.w$ labelled by w. By Lemma 4, the number of fixed positions we need to estimate equals the number of occurrences of fixed states in W, excluding the terminal occurrence of $\lambda.w$. Note that W, as well as any walk in $\mathcal{A}_i$, can be obtained as follows: take a simple path between the initial and the terminal states of the walk and insert repeatedly simple cycles into the walk built so far. The simple path (say, of length n) contains, at most, $d_i n + c_i$ fixed states, and the rest contains, at most, $d_i(|w| - n)$ fixed states, whence the result. □

Corollary 1. *In an infinite cube-free word, the density of the set of letters fixed by cubes with periods of at most i is upper-bounded by d_i.*

The numbers d_i and c_i can be computed from $\mathcal{A}_i$ in polynomial time using dynamic programming (due to Corollary 1, we need only d_i). A straightforward way to do this is to compute for each u, v in the order of increasing k the maximum fraction $d[u, v, k]$ of fixed states in a (u, v)-walk of length at most k; then $d_i = \max_u d[u, u, N_i]$, where $\mathcal{A}_i$ has N_i states. This algorithm has cubic complexity, but we can do significantly better. We note that any automaton $\mathcal{A}_i$ has a unique nontrivial strongly connected component; this quite nontrivial fact follows from the main result of [29].

Proposition 1. *Let N_i and n_i be the numbers of nodes in $\mathcal{A}_i$ and its nontrivial strongly connected component, respectively. Then there exists an algorithm computing d_i from $\mathcal{A}_i$ in time $O(n_i^2 + N_i)$.*

Proof. Recall that the *mean cost* of a walk in a weighted digraph is the ratio between its cost and its length. We reduce the problem of computing d_i to the problem of finding a cycle of the minimum mean cost. Considering $\mathcal{A}_i$ as a digraph, we assign cost 0 to fixed edges and cost 1 to free edges. Then we replace $\mathcal{A}_i$ with its unique nontrivial strongly connected component $\mathcal{A}'_i$ preserving the edge costs. This component contains all cycles of $\mathcal{A}_i$. Now $d_i = 1 - \mu$, where μ is the minimum mean cost of a cycle in the weighted digraph $\mathcal{A}'_i$.

The mean cost problem can be solved for an arbitrary strongly connected digraph with n nodes and m edges in $O(nm)$ time and space using Karp's algorithm [30]. Noting that, in our case, $m = O(n)$, and that the strongly connected component can be found in linear time by a textbook algorithm, we obtain the required time bound.

For the sake of completeness, let us describe Karp's algorithm for our case. Fix an arbitrary state q and define $C(j, v)$ to be the minimum cost of a length-j walk from q to v or ∞ if no such walk exists. The $(n_i + 1) \times n_i$ table with the values of $C(j, v)$ for $j = 0, \ldots, n_i$ and all states of $\mathcal{A}'_i$ is filled using the following dynamic programming rule:

$$C(j+1, v) = \min_{z : z \xrightarrow{c} v} (C(j, z) + cost(z, v)), \tag{7}$$

$$C(0, v) = \begin{cases} 0, & \text{if } v = q, \\ \infty, & \text{otherwise.} \end{cases} \tag{8}$$

According to [30] (Theorem 1), $\mu = \min_{v \in \mathcal{A}'_i} \max_{0 \leq j < n_i} \left(\frac{C(n_i, v) - C(j, v)}{n_i - j} \right)$. □

Remark 3. *Karp's algorithm also allows one to retrieve a cycle of minimum mean cost. To do this, one stores the node $z = P(j, v)$, which gives the minimum in the computation (7) of $C(j, v)$ (here, $j = 1, \ldots, n_i$, and $P(j, v)$ is undefined if $C(j, v) = \infty$). The $n_i \times n_i$ table $P(j, v)$ is then used as follows. If u is a node for which the value of μ is reached, then we built the length-n_i walk $q = u_0 \to u_1 \to \cdots \to u_{n_i} = u$ such that $P(j{+}1, u_{j+1}) = u_j$ for all j and output any simple cycle from this walk. We will need the cycles of minimum mean cost in Section 4.3.*

4.2. Lower Bounds on Branching Density

We implemented the above algorithm and ran it for all $i \leq 18$; for $i = 18$, the memory required for the table $C(j, v)$ is over 1 GB. The results are as follows.

Lemma 6. *One has $d_1 = d_2 = 1/2$, $d_3 = \cdots = d_{11} = 7/13$, $d_{12} = d_{13} = d_{14} = 13/24$, $d_{15} = d_{16} = d_{17} = d_{18} = 53/96$.*

Now we are ready to prove our first main result.

Theorem 3. *The branching density of an infinite binary cube-free word is at least $3509/9120 \approx 0.38476$.*

Proof of Theorem 3. Let us fix some integer $h \geq 2$. Theorem 1 and Corollary 1 together imply that the density of fixed positions in an infinite cube-free word is upper-bounded by $d_{h-1} + \frac{6}{5h}$. Trying all values of d_i from Lemma 6, we get the maximum at $h = 19$:

$$d_{18} + \frac{6}{5 \cdot 19} = \frac{53}{96} + \frac{6}{95} = \frac{5611}{9120}.$$

The statement of the theorem now follows. □

In the same way, we can get the lower bound for the ternary square-free language SF. From [19] (Lemma 5), we have the upper bound $\frac{2}{h}$ for the density of positions fixed by squares of periods $\geq h$. Lemmas 4 and 5, and Corollary 1 have direct analogs for ternary square-free words; Proposition 1 and the algorithm inside remain valid for any automaton having, at most, two outgoing edges for each state. Running the algorithm for the regular approximations of SF up to $i = 30$, we obtained the correspondent numbers d'_i. Taking $h = 31$ and adding $\frac{2}{h}$ to $d'_{30} = 19/28$, we arrive at the following bound.

Theorem 4. *The branching density of an infinite ternary square-free word is at least $223/868 \approx 0.25691$.*

Recall that the *growth rate* of a factorial language L over the alphabet Σ is the limit $\lim_{n\to\infty} |L \cap \Sigma^n|^{1/n}$. The growth rate β of CF is known with quite high precision [9]: $1.4575732 \leq \beta \leq 1.4575773$. In terms of the prefix tree, this means that for big n, the number of nodes at the $(n+1)$th level is approximately β times bigger than the number of nodes at the nth level. This fact makes $\beta - 1$ a lower bound on the fraction of branching nodes at the nth level (because this level also contains nodes having no children). In Theorem 5 below, we use Proposition 1 and Remark 3 to prove that there exist infinite cube-free words with the branching density strictly smaller than $\beta - 1$.

The above considerations can also be applied to the growth rate γ of SF, $1.3017597 \leq \gamma \leq 1.3017619$ [9]. However, it is still open whether an infinite square-free word can have the branching density smaller than $\gamma - 1$. The method of Theorem 5 would not work for SF because the obtained values of d'_i are too small.

4.3. Cube-Free Words with Small Branching Density

Theorem 5. *The minimum branching density of an infinite cube-free word is less than or equal to $13/29 \approx 0.44828$.*

Proof. The result of Lemma 6 gives us an idea of constructing an infinite cube-free word with branching density less than $\beta - 1$. We see that $1 - d_{15} \approx 0.44792 < \beta - 1$ (while $1 - d_{14} \approx 0.45833 > \beta - 1$). Our aim is to construct an infinite cube-free word which has the density of fixed positions very close to d_{15}.

Using the table $P(j, v)$ of Karp's algorithm (see Remark 3), we find that the automaton $\mathcal{A}_{15}$ contains exactly four cycles C_1, C_2, $\bar{C}_1$, and $\bar{C}_2$, each of length 96, reaching the minimum mean cost $(1 - d_{15})$. All four cycles are disjoint; the labels of cycles $\bar{C}_1, \bar{C}_2$ are complements of the labels of C_1 and C_2, respectively. We note that C_1 and C_2 are connected to each other by many edges. Let us consider a subgraph of $\mathcal{A}_{15}$ consisting of C_1, C_2, and two edges connecting them as in Figure 7.

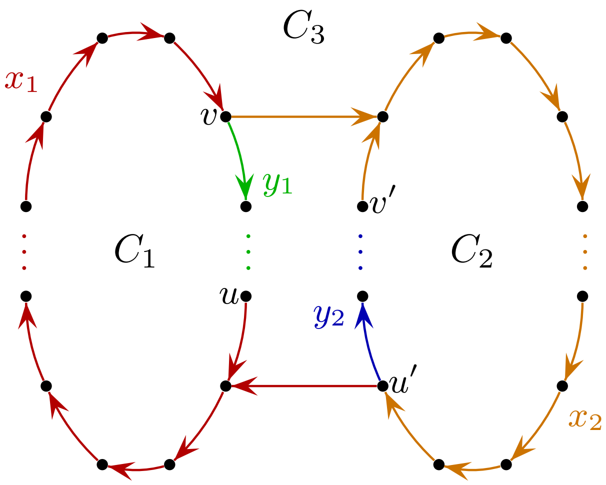

Figure 7. Building the infinite cube-free word of small branching density using the cycles of low mean cost in $\mathcal{A}_{15}$.

Since in the Aho-Corasick automaton, all edges with a common endpoint have the same label (if this endpoint is distinct from λ), the paths from u to v and from u' to v in Figure 7 are labeled by the same word x_1, while the paths from both v and v' to u' are labeled by the same word x_2. Denote the labels of the paths from v to u and from u' to v' by y_1 and y_2, respectively. Then the label of C_1 is x_1y_1 (starting from u), the label of C_2 is x_2y_2 (starting from v'), and there is an "outer" cycle C_3 with the label x_1x_2 (starting from u'). We also note that x_1 and y_2 (resp., x_2 and y_1) begin with different letters.

Analyzing the subgraph of $\mathcal{A}_{15}$ generated by C_1 and C_2, we find the cycle C_3 with the smallest mean cost: it has a length of 156 and 86 fixed states. The corresponding values of x_1, x_2, y_1, y_2 are as follows:

$$\begin{aligned}
x_1 &= \texttt{aabaabbaabaababaabaabbaabaababaabbaabaababaabaabbaabaababaabaabbaa} \\
&\quad \texttt{baababbabbaab}, \quad |x_1| = 79 \\
x_2 &= \texttt{babbababbabbaabbabbababbabbaabbababbabbaabbabbababbabbaabbabbababb} \\
&\quad \texttt{abbaabaabab}, \quad |x_2| = 77 \\
y_1 &= \texttt{aababbabbaabaabab}, \quad |y_1| = 17 \\
y_2 &= \texttt{babbaabaababbabbaab}, \quad |y_2| = 19.
\end{aligned}$$

Recall that the Thue–Morse word $\mathbf{t}$ is the fixed point of the morphism $a \to ab, b \to ba$:

$$\mathbf{t} = \mathbf{t}[1..\infty] = abba\ baab\ baab\ abba\ baab\ abba\ abba\ baab\ baab\ abba\ abba\ baab\ abba\ baab \ldots.$$

This word is *overlap-free* [10], that is, it has no factors of the form *cucuc* where c is a letter and u is a word. We map $\mathbf{t}$ to an infinite binary word by the mapping ϕ defined by two rules:

1. $\phi(\mathbf{t}[1]) = x_1$;
2. $$\phi(\mathbf{t}[i]) = \begin{cases} x_1 & \text{if} \quad \mathbf{t}[i] = a \neq \mathbf{t}[i-1], \\ y_1x_1 & \text{if} \quad \mathbf{t}[i] = a = \mathbf{t}[i-1], \\ x_2 & \text{if} \quad \mathbf{t}[i] = b \neq \mathbf{t}[i-1], \\ y_2x_2 & \text{if} \quad \mathbf{t}[i] = b = \mathbf{t}[i-1]. \end{cases}$$

In other terms, to get $\phi(\mathbf{t})$, we replace each a (resp., b) in $\mathbf{t}$ with x_1 (resp., x_2), and then insert y_i in the middle of each factor $x_i x_i$ of the obtained word:

$$\begin{array}{rcccccccccc} \mathbf{t} = & a & bb & ab & aa & bb & aa & ba & bb & a\ldots \\ \phi(\mathbf{t}) = & x_1 & x_2y_2x_2 & x_1x_2 & x_1y_1x_1 & x_2y_2x_2 & x_1y_1x_1 & x_2x_1 & x_2y_2x_2 & x_1\ldots. \end{array}$$

Thus, we naturally have a partition of $\phi(\mathbf{t})$ into factors which we call *blocks*, distinguishing between x-blocks x_1, x_2 and y-blocks y_1, y_2. Note that

($\diamond$) x_1 and x_2 have no occurrences in $\phi(\mathbf{t})$ other than x-blocks.

Indeed, the factor $x_1[4..9] = aabbaa$ does not occur in x_2, y_1, y_2, or on the border of any two blocks; the symmetric property holds for $x_2[3..9] = bbababb$.

Let us prove that $\phi(\mathbf{t})$ is cube-free. Assume to the contrary that $\phi(\mathbf{t})$ contains a cube; let XXX be the leftmost of the minimal cubes in $\phi(\mathbf{t})$. A direct check shows that all words $x_iy_ix_i$ and $x_ix_jx_i$ are cube-free. Therefore, XXX contains a whole x-block inside; below, x_i denotes the leftmost such block, and x_j denotes the x-block distinct from x_i. The period $|X|$ of the cube cannot be smaller than the minimum period of x_i. It is easy to check that the only period of x_i is $|x_i| - 3$. Thus, $|X| \geq |x_i| - 3 \geq 74$.

First, assume that x_i is the only x-block in XXX. Since for the word $x_jx_ix_j$ is cube-free, XXX is an internal factor of either $x_iy_ix_ix_j$ or $x_jx_iy_ix_i$. Since $|x_iy_ix_i|, |y_ix_ix_j| \leq 175$ and $|XXX| \geq 74 \cdot 3 = 222$, the length argument shows that y_ix_i in the first case, and x_iy_i in the second case is a factor of XXX. Thus, $|X|$ is not smaller than the minimum of the periods of words x_iy_i, y_ix_i. This minimum equals 84, which is a period of both y_1x_1 and y_2x_2. Hence, $|XXX| \geq 252 = 2|x_i| + |x_j| + |y_i|$. However, an internal factor of a word must be shorter by at least two symbols than the word itself; this contradiction shows that XXX contains more than one x-block. Therefore, XXX contains either $x_iy_ix_i$ or x_ix_j.

Let $XXX = v_1x_iy_ix_iv_2$. By the choice of x_i, v_1 is a proper suffix of x_j and v_2, if nonempty, begins with the first letter of x_j, which differs from the first letter of y_i. Thus, if $|X|$ equals the minimum period of $x_iy_ix_i$, which is $|x_iy_i|$, then $v_2 = \lambda$ and then $|XXX| < 3|X|$, which is impossible. Hence, $|X| > |x_iy_i|$. Therefore, for each of the two blocks, x_i, we see that XXX contains another factor x_i at the distance of $|X|$. By ($\diamond$), these factors are x-blocks; both are inside v_2 because v_1 is a suffix of an x-block. Then, X contains at least one occurrence of the block x_i. As a result, XXX contains a factor $x_iwx_iwx_i$ for some word w which is a product of blocks; here, x_iw is a cyclic shift of X. Taking the ϕ-pre-image of $x_iwx_iwx_i$, we obtain a factor of the form $auaua$ or $bubub$ in $\mathbf{t}$, in contradiction with the overlap-freeness property.

Finally, let $XXX = v_1x_ix_jv_2$. The word x_ix_j has no periods smaller than $|x_ix_j| - 1$. Hence, $|X| > |x_i| + |x_j| - 1$ and then X contains at least one of x_i, x_j. Since v_1 contains no x-blocks by the choice of x_i, v_1x_i has no factor x_j by ($\diamond$). Then, X must contain x_i, and we arrive at a contradiction as in the previous paragraph. Thus, finally, we have proved that $\phi(\mathbf{t})$ is cube-free.

The word $\phi(\mathbf{t})$ corresponds to an infinite walk from the node u in the subgraph of $\mathcal{A}_{15}$ depicted in Figure 7. The walk reads x_1 (rule 1 in the definition of ϕ) and then respects rule 2; the details are as follows. Let x_i be just read; if the current letter of $\mathbf{t}$ coincides with the previous one, the walk returns to the "start" node of the same cycle C_i by reading y_i and reads x_i again; otherwise, the walk reads x_j, where $i \neq j$. We know the fractions of fixed states in the cycles C_1, C_2, and C_3; to calculate the density of positions fixed by short squares in $\phi(\mathbf{t})$ we use the folklore fact that the density of the sets of positions i such that $\mathbf{t}[i..i{+}1] = aa$ (resp., ab, ba, bb) equals $1/6$ (resp., $1/3, 1/3, 1/6$). Then in the partition of $\phi(\mathbf{t})$ into blocks, the densities of the blocks x_1, x_2 are equal and twice bigger than the densities of the blocks y_1 and y_2. We group the blocks into labels of cycles C_1, C_2, C_3:

$$\phi(\mathbf{t}) = [x_1x_2][y_2x_2][x_1x_2][x_1y_1][x_1x_2][y_2x_2][x_1y_1][x_1x_2][x_1x_2][y_2x_2][x_1\ldots.$$

Since x-blocks appear in the labels of two cycles while y-blocks appear in the label of one cycle, all cycles appear with the same density. Thus, to get the density of fixed letters, we take the total number of fixed states in C_1, C_2, and C_3 and divide it by the sum of lengths of cycles to get $(86+53+53)/(156+96+96) = 16/29$. Then the branching density of $\phi(\mathbf{t})$ is, at most, $13/29$. The theorem is proved. □

Remark 4. *In fact, the branching density of $\phi(\mathbf{t})$ is exactly $13/29$: refining the analysis of cube-freeness of $\phi(\mathbf{t})$, it is possible to show that this word does not contain letters fixed by long cubes.*

5. The Bounds on Maximum Branching Density

The branching density of infinite cube-free words can be much bigger than $\beta - 1 \approx 0.45758$. The aim of this section is to prove the following theorem.

Theorem 6. *(1) The maximum branching density of an infinite binary cube-free word is less than $67/93 \approx 0.72043$.*
(2) There exists an infinite binary cube-free word with branching density $18/25 = 0.72$.

Example 1. *The branching density of the Thue–Morse word $\mathbf{t}$ is 2/3. Indeed, $\mathbf{t}$ is overlap-free, and thus all fixed letters in it are fixed by 1-cubes. Hence, the fixed letters are exactly the letters a (resp. b) preceded by the* 1-square *bb (resp. by aa); in each case, the density of such positions is 1/6, as mentioned in Section 4.3. Thus, the density of fixed positions is 1/3.*

The proof of Theorem 6 is based on the analysis of positions fixed by 1 cube. The distance between two successive occurrences of the square of a letter in a cube-free word is 2 ($aabb/bbaa$) or 3 ($aabaa/bbabb$) or 4 ($aababb/bbabaa$) or 5 ($aababaa/bbababb$); it cannot be 1 (aaa/bbb) or ≥ 6 ($aababab\cdots/bbababa\cdots$) because of cube-freeness. Hence, if we know a prefix $w[1..i]$ of a cube-free word, this prefix ends with a 1-square, and the distance d to the next 1-square is known; then we can uniquely reconstruct $w[1..i+d]$. We consider an auxiliary alphabet $\Delta = \{2,3,4,5\}$ and refer to its elements as *digits* and to the words over it as *codes*. For every cube-free word $\mathbf{w} \in \Sigma^\infty$ we define its *distance code* $\mathsf{dist}(\mathbf{w}) \in \Delta^\infty$ as follows: $\mathsf{dist}(\mathbf{w})[i]$ is the distance between the ith and $(i{+}1)$th 1-squares in $\mathbf{w}$ (counting from the left). For example, one has

$$\begin{array}{rcccccccccccccccccccccccccccccccccc}
\mathbf{t} = & a & b & b & a & b & a & a & b & b & a & a & b & a & b & b & a & b & a & a & b & a & b & b & a & a & b & b & a & b & a & a & b & \cdots \\
\mathsf{dist}(\mathbf{t}) = & & 4 & & & & 2 & & 2 & & 4 & & & & 4 & & & & 4 & & & & 2 & & 2 & & 4 & & & & & & & \cdots .
\end{array}$$

Note that $\mathsf{dist}(\mathbf{w})$ determines $\mathbf{w}$ up to the complement and the few letters preceding the first 1-square; in particular, it determines the branching density of $\mathbf{w}$ if $\mathbf{w}$ is cube-free. Thus, instead of infinite cube-free words, here we study their distance codes. We extend the definition of a distance code to finite words in the obvious way; for example, $\mathsf{dist}(bbabaabb) = 42$. Here, $\mathsf{dist}(w)$ determines w up to the complement, the letters preceding the first 1-square, and the letters following the last 1-square. We define the inverse of the map dist: for a code $X \in \Delta^+$, $w = \mathsf{word}(X)$ is the unique word which begins with aa, ends with a 1-square, and satisfies $\mathsf{dist}(w) = X$. Clearly, $\mathsf{word}(X)$ has length $[X]+2$, where $[X]$ denotes the sum of digits in X, and has $|X|$ letters fixed by 1-cubes. For example, the cube-free word $aa\underline{b}b\underline{a}bb\underline{a}baa\underline{b}abaa = \mathsf{word}(2345)$ has a length of 16 and 4 letters fixed by 1-cubes (underlined). The same definition of word, with the condition on the end of the word omitted, applies for infinite codes.

Remark 5. *The word* $\mathsf{word}(33) = aabaabaa$ *is not a proper factor of a cube-free word;* $\mathsf{word}(434) = aababbabbabaa$ *contains* $(bab)^3$*, as well as* $\mathsf{word}(435), \mathsf{word}(534)$*, and* $\mathsf{word}(535)$*. In the following list of cube-free words, letters fixed by p-cubes are underlined by p lines:*

$$\begin{array}{ll} \mathsf{word}(2) = aa\underline{b}b & \mathsf{word}(234) = aa\underline{b}b\underline{a}bb\underline{a}b\underline{\underline{\underline{a}}}a \\ \mathsf{word}(3) = aa\underline{b}aa & \mathsf{word}(235) = aa\underline{b}b\underline{a}bb\underline{a}b\underline{\underline{\underline{a}}}b\underline{\underline{b}} \\ \mathsf{word}(4) = aa\underline{b}abb & \mathsf{word}(432) = aa\underline{b}abb\underline{a}bb\underline{a}\underline{\underline{\underline{a}}} \\ \mathsf{word}(5) = aa\underline{b}aba\underline{\underline{a}} & \mathsf{word}(532) = aa\underline{b}aba\underline{\underline{a}}\underline{b}aa\underline{b}\underline{\underline{\underline{b}}} \end{array}.$$

Proof of Theorem 6. Let $A = (4^42)^44^22$, $B = (4^42)^44^45$. Let $\mathbf{X}$ be the image of the Thue-Morse word $\mathbf{t}$ under the substitution $a \to A$, $b \to B$. We prove Statement 2 by showing that $\mathbf{u} = \mathsf{word}(\mathbf{X})$ is cube-free and has the branching density $18/25 = 0.72$. We first count the positions in $\mathbf{u}$ fixed by 1-cubes and 2-cubes. We have one position fixed by a 1-cube per digit of $\mathbf{X}$ and add one position fixed by a 2-cube for each digit 5 in $\mathbf{X}$ (see Remark 5). Each block A adds $[A] = 82$ letters to $\mathbf{u}$, from which $|A| = 23$ are fixed by 1-cubes; each block B adds $[B] = 93$ letters from which $|B| = 25$ are fixed by 1-cubes and one is fixed by a 2-cube. Since A and B appear in $\mathbf{X}$ with the same density, the density of positions fixed by 1- and 2-cubes in $\mathbf{u}$ equals $(|A| + |B| + 1)/([A] + [B]) = 49/175 = 7/25$. Thus, to prove Statement 2 of the theorem, it is necessary and sufficient to show that no position in $\mathbf{u}$ is fixed by a k-cube with $k > 2$; that is, $\mathbf{u}$ contains no *almost-cubes* of the form $xxx[1..|x|-1]$ with $|x| > 2$. Aiming at a contradiction, consider an almost-cube of x in $\mathbf{u}$.

It is easy to check that $\mathsf{word}(24), \mathsf{word}(42), \mathsf{word}(44)$, and $\mathsf{word}(454)$ contain no almost-cubes and have at least six periods. Thus, $|x| \geq 6$, and hence, x contains a 1-square. Then, $|x|$ is the distance between two squares of the same letter. Each of $\mathsf{word}(2)$ and $\mathsf{word}(4)$ begins and ends with different 1-squares, while $\mathsf{word}(5)$ begins and ends with the same square. Hence, $|x| = [Y]$ for some factor Y of $\mathbf{X}$ such that (i) the total number of 2's and 4's in Y is even; (ii) YY is a factor of $\mathbf{X}$. If Y contains no 5's, there are just two cases to check. Case 1: $Y = 44$, $|x| = 8$. Long factors with period 8 are located in $\mathbf{u}$, up to a complement, within the factors

$$\begin{aligned} \mathsf{word}(244442) &= a(abbabaab)^2(abba)a \text{ (length 20);} \\ \mathsf{word}(244445) &= a(abbabaab)^2(abbaba)bb \text{ (length 22);} \\ \mathsf{word}(544442) &= aa(babaabab)^2(babaab)b \text{ (length 22);} \end{aligned}$$

their lengths are less than $8 \cdot 3 - 1 = 23$ required for an almost-cube.

Case 2: $Y = (4^42)^2$, $|x| = 36$. The code YY occurs in $\mathbf{X}$ only as a prefix of A or B and thus is preceded and followed by one of the factors 4^22 and 4^45. The longest factor of $\mathbf{u}$ with period 36, up to a complement, can be found in

$$\begin{aligned} &\mathsf{word}(24^22(4^42)^44^45) = \\ &\quad a(abbabaababbaababbabaababbabaabbabaab)^2(abbabaababbaababbabaababbabaab)abaa, \end{aligned}$$

which is again too short (102 letters) for an almost-cube $(36 \cdot 3 - 1 = 107)$.

Therefore, 5 must occur in Y. Since $YY \subseteq \mathbf{X}$, $|Y|$ is the distance between two occurrences of 5 in $\mathbf{X}$. Since 5 occurs in $\mathbf{X}$ only as a suffix of the block B, Y is a cyclic shift of some product of blocks. As above, we will show that the longest factor of $\mathbf{u}$ with period $|x| = [Y]$ is shorter than $3|x| - 1$. Let $C = (4^42)^44^2$, then $A = C2, B = C445$. Since $\mathbf{t}$ is overlap-free, the maximal factor of $\mathbf{X}$ with period $|Y|$ looks like $Y'Y'C$, where Y' is a product of blocks and a cyclic shift of Y. Then, the longest factor of $\mathbf{u}$ with period $|x|$ looks like $v_1\mathsf{word}(Y'Y'C)v_2$. Here, $|\mathsf{word}(Y'Y'C)| = 2[Y'] + [C] + 2 = 2[Y'] + 82$; note that $[Y] \geq [B] = 93$. Further, v_1 is the common suffix obtained when decoding different digits (2 and 5) and v_2 is the common prefix obtained when decoding different digits (2 and 4). Hence, $|v_1| = |v_2| = 1$. In total, the length of the $|x|$-periodic factor is strictly smaller than $3[Y'] - 1$. Therefore, no almost-cubes are present in $\mathbf{u}$. This proves Statement 2 of the theorem.

For Statement 1, we take a cube-free word $\mathbf{w}$ of maximum branching density and consider its code $\mathsf{dist}(\mathbf{w})$. By Remark 5, each digit in $\mathsf{dist}(\mathbf{w})$ corresponds to a letter fixed by a 1-cube, and a digit 5 also corresponds to a letter fixed by a 2-cube. The density of fixed positions in $\mathbf{w}$ is at its minimum, and thus is upper-bounded by $7/25$, which is such a density for $\mathbf{u}$. Since $7/25$ is closer to $1/4$ than to $1/3$, the majority of digits in $\mathbf{w}$ are 4's. Since $\mathbf{w}$ has the same branching density as each of its suffixes, we assume w.l.o.g. that $\mathsf{dist}(\mathbf{w})$ begins with 4, and represent it as a sequence of blocks: each block consists of one or more 4's in the beginning and one or more other digits in the end. Note that the words $\mathsf{word}(54^45)$ and $\mathsf{word}(c4^5d)$, for any digits c, d, contain an 8-cube (cf. Case 1 above). Then, a short search reveals all blocks providing the density of fixed positions not greater than 0.3:

$$\begin{aligned} 44442 &: 5/18 \approx 0.27778; & 4445 &: 5/17 \approx 0.29412; \\ 44445 &: 6/21 \approx 0.28571; & 442 &: 3/10; \\ 4442 &: 4/14 \approx 0.28571. \end{aligned}$$

(We recall that blocks containing 3's are restricted, as shown in Remark 5.) We note that $\mathsf{word}((4^42)^54^4)$ and $\mathsf{word}((4^42)^54^35)$ contain 36-cubes, while $\mathsf{word}(4^324^324^3)$ contains a 14-cube. As a result, the density of fixed letters in $\mathbf{w}$ cannot be smaller than such density in $\mathsf{word}(((4^42)^44^45)^\infty)$, which is $26/93$. This gives us the upper bound $67/93$ on the branching density of $\mathbf{w}$, as required. □

6. Discussion and Future Work

As we have seen in this paper, the branching density of particular infinite words in a typical power-free language of exponential growth can vary significantly. Thus, a natural question is to determine the average density. The first problem is to define what is "average"; we suggest that this should be the *expected* density of a word randomly chosen from all infinite binary cube-free words according to the distribution which is "uniform" in some sense. One possible way to choose a random infinite word is a random walk down the prefix tree (with all finite subtrees trimmed).

Another possible next step is to check whether the ternary square-free language SF, which is another "typical" power-free language of exponential growth, demonstrates the same patterns as CF. Currently we do not know whether some infinite square-free words have branching density strictly less than its growth rate minus one. We also know no reasonable bound for the maximum branching density in SF.

Author Contributions: Methods and algorithms, A.M.S.; software and experiments, E.A.P.; theorems and proofs, A.M.S. and E.A.P.; writing—original draft preparation, A.M.S.; writing—review and editing, A.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and Higher Education of the Russian Federation (Ural Mathematical Center project No. 075-02-2020-1537/1).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Thue, A. Über unendliche Zeichenreihen. *Nor. Vid. Selsk. Skr. Mat. Nat. Kl.* **1906**, *7*, 1–22.
2. Restivo, A.; Salemi, S. Overlap free words on two symbols. In *Automata on Infinite Words, Proceedings of the Ecole de Printemps d'Informatique Theorique, Le Mont Dore, France, 14–18 May 1984*; Nivat, M., Perrin, D., Eds.; Springer: Berlin/Heidelberg, Germany, 1985; Volume 192, pp. 198–206.
3. Shur, A.M.; Gorbunova, I.A. On the growth rates of complexity of threshold languages. *RAIRO Inform. Théor. App.* **2010**, *44*, 175–192. [CrossRef]
4. Karhumäki, J.; Shallit, J. Polynomial versus exponential growth in repetition-free binary words. *J. Combin. Theory. Ser. A* **2004**, *104*, 335–347. [CrossRef]
5. Ochem, P. A generator of morphisms for infinite words. *RAIRO Inform. Théor. App.* **2006**, *40*, 427–441. [CrossRef]

6. Kolpakov, R.; Rao, M. On the number of Dejean words over alphabets of 5, 6, 7, 8, 9 and 10 letters. *Theoret. Comput. Sci.* **2011**, *412*, 6507–6516. [CrossRef]
7. Tunev, I.N.; Shur, A.M. On two stronger versions of Dejean's conjecture. In Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science (MFCS 2012), Bratislava, Slovakia, 27–31 August 2012; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7464, pp. 801–813.
8. Currie, J.D.; Mol, L.; Rampersad, N. The Number of Threshold Words on n Letters Grows Exponentially for Every n $\geq$ 27. *J. Integer Seq.* **2020**, *23*, 1–12.
9. Shur, A.M. Growth properties of power-free languages. *Comput. Sci. Rev.* **2012**, *6*, 187–208. [CrossRef]
10. Thue, A. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. *Nor. Vid. Selsk. Skr. Mat. Nat. Kl.* **1912**, *1*, 1–67.
11. Jungers, R.M.; Protasov, V.Y.; Blondel, V.D. Overlap-free words and spectra of matrices. *Theoret. Comput. Sci.* **2009**, *410*, 3670–3684. [CrossRef]
12. Guglielmi, N.; Protasov, V. Exact Computation of Joint Spectral Characteristics of Linear Operators. *Found. Comput. Math.* **2013**, *13*, 37–97. [CrossRef]
13. Carpi, A. On the centers of the set of weakly square-free words on a two-letter alphabet. *Inform. Process. Lett.* **1984**, *19*, 187–190. [CrossRef]
14. Shur, A.M. Deciding context equivalence of binary overlap-free words in linear time. *Semigroup Forum* **2012**, *84*, 447–471. [CrossRef]
15. Bean, D.A.; Ehrenfeucht, A.; McNulty, G. Avoidable patterns in strings of symbols. *Pac. J. Math.* **1979**, *85*, 261–294. [CrossRef]
16. Currie, J.D. On the structure and extendibility of *k*-power free words. *Eur. J. Comb.* **1995**, *16*, 111–124. [CrossRef]
17. Currie, J.D.; Shelton, R.O. The set of *k*-power free words over Σ is empty or perfect. *Eur. J. Comb.* **2003**, *24*, 573–580. [CrossRef]
18. Petrova, E.A.; Shur, A.M. Constructing premaximal ternary square-free words of any level. In Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science (MFCS 2012), Bratislava, Slovakia, 27–31 August 2012; Volume 7464, pp. 752–763.
19. Petrova, E.A.; Shur, A.M. On the tree of ternary square-free words. In *Combinatorics on Words, Proceedings of the 10th International Conference (WORDS 2015)*, Kiel, Germany, 14–17 September 2015; Springer: Cham, Switzerland, 2015; Volume 9304, pp. 223–236.
20. Shelton, R. Aperiodic words on three symbols. II. *J. Reine Angew. Math.* **1981**, *327*, 1–11.
21. Shelton, R.O.; Soni, R.P. Aperiodic words on three symbols. III. *J. Reine Angew. Math.* **1982**, *330*, 44–52.
22. Petrova, E.A.; Shur, A.M. Constructing premaximal binary cube-free words of any level. *Internat. J. Found. Comp. Sci.* **2012**, *23*, 1595–1609. [CrossRef]
23. Petrova, E.A.; Shur, A.M. On the tree of binary cube-free words. In *Developments in Language Theory , Proceedings of the 21st International Conference (DLT 2017)*, Liège, Belgium, 7–11 August 2017; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10396, pp. 296–307.
24. Lothaire, M. *Combinatorics on Words; Volume 17, Encyclopedia of Mathematics and Its Applications*; Addison-Wesley: Reading, MA, USA, 1983.
25. Lyndon, R.C.; Schützenberger, M.P. The equation $a^M = b^N c^P$ in a free group. *Mich. Math. J.* **1962**, *9*, 289–298. [CrossRef]
26. Fine, N.J.; Wilf, H.S. Uniqueness theorems for periodic functions. *Proc. Am. Math. Soc.* **1965**, *16*, 109–114. [CrossRef]
27. Crochemore, M.; Mignosi, F.; Restivo, A. Automata and forbidden words. *Inform. Process. Lett.* **1998**, *67*, 111–117. [CrossRef]
28. Shur, A.M. Growth rates of complexity of power-free languages. *Theoret. Comput. Sci.* **2010**, *411*, 3209–3223. [CrossRef]
29. Petrova, E.A.; Shur, A.M. Transition property for cube-free words. In *Computer Science—Theory and Applications, Proceedings of the 14th International Computer Science Symposium in Russia (CSR 2019)*, Novosibirsk, Russia, 1–5 July 2019; Springer: Berlin, Germany, 2019; Lecture Notes in Computer Science; Volume 11532, pp. 311–324.
30. Karp, R.M. A characterization of the minimum cycle mean in a digraph. *Discret. Math.* **1978**, *23*, 309–311. [CrossRef]