The minimal density of a letter in an infinite ternary square-free word is $\frac{883}{3215}$.

Andrey Khalyavin

Mech. & Math. Department
Moscow State University
119992 Moscow, Russia
emails: halyavin@land.ru

Abstract

The problem of the minimal density of a letter in an infinite ternary square-free word was investigated in some articles. In this paper we solve this problem. Precisely, we prove that this value is equal to $\frac{883}{3215}$.

1 Introduction

A word is called *square-free* if it cannot be written in the form $axxb$ for two words a, b and a nonempty word x. In this paper we study the minimal density of a letter a in an infinite square-free word in alphabet $\{a, b, c\}$. *The density* of a letter a in a finite word w is $\frac{n(w)}{|w|}$ where $n(w)$ is the number of letters a in this word and $|w|$ is the length of the word. *The density* of a letter a in an infinite word is $\liminf\limits_{l\to\infty} \frac{n(w_l)}{|w_l|}$ where w_l is the prefix of the word w of length l. In the article [1] the minimal density of a letter a was found with up to 4 tight digits after the decimal point. The article [2] improved this result proving that the minimal density of a letter a belongs to the interval $[\frac{1000}{3641}, \frac{883}{3215}]$. In this article we solve completely this problem and prove that the minimal density of a letter a is equal to $\frac{883}{3215}$. We use huge computer calculations in our proof. You can find sources of the programs at

The author is grateful to his scientific supervisor Prof. Yuriy Tarannikov and Dr. Roman Kolpakov for the attention to this work and helpful remarks each.

2 Main result

Our aim is to prove that the minimal density is at least $\rho = \frac{883}{3215}$. Let us construct 4 special words $b_1, \ldots, b_4$ of length 3215 which contain 883 letters a

(they are probably the same words which were used in [2] in order to construct the infinite word with the density ρ). We call these words *the basic words*. The files with these basic words can be found at url address above. We proved the next theorem using backtrack technique on computer.

Theorem 1 *In any infinite square-free word there exist either a prefix with density not less that ρ or two adjacent basic words (a subword $b_i b_j$). .*

So, either an infinite square-free word can be splitted to words with the density not less than ρ or there exist two basic words one after another. In the first case the density of an infinite word is not less than ρ because lengths of the words in the splitting are bounded (computer shows that these lengths are not greater than 90000). So we can assume that an infinite square-free word begins with a word $b_i b_j$.

When our program that proves Theorem 1 finds two adjacent basic words in the current word w then it saves the pair $(n(w), |w|)$ and backtracks. Let us denote by W the set of all such current words. Then we delete equal pairs $(n(w), |w|)$ from out list and obtain 80 distinct pairs $(n(w), |w|)$. Let us denote by L the set of these pairs. Then for all $l < 200000$ we calculate the maximal number $p(l)$ such that after the concatination of any words $w \in W$ and v provided $n(v) \geq p(l), |v| = l$, the word vw has the density not less than ρ. Our second program proves the next theorem.

Theorem 2 *Let us consider an infinite square-free word u which begins with a word $b_i b_j$. Then this word contains one of the following prefices:*

1) *the prefix w provided $2 \cdot 3215 < |w| < 200000$ and $n(w) \geq p(|w|)$,*

2) *the prefix w provided $4 \cdot 3215 \leq |w| < 200000, n(w) \geq \rho|w|$, which ends with two adjacent basic words.*

Theorem 3 *Let us consider an infinite square-free word u which begins with a word $b_i b_j$. If the density of the word u is less than ρ then this word contains one of the following prefices:*

1) *the prefix $ww_1w_2 \dots w_n v$ provided $|w| < 200000, |w_i| < 200000, 2 \cdot 3215 < |v| < 200000, n(w_i) \geq \rho|w_i|, n(wv) \geq \rho|wv|$ which ends with two adjacent basic words.*

2) *the prefix w provided $4 \cdots 3215 \leq |w| < 200000, n(w) \geq \rho|w|$ which ends with two adjacent basic words.*

Let us apply Theorem 2 to our infinite square-free word. If the second case takes place, we have proved our theorem. Otherwise we apply Theorem 1 to the word u without the prefix w. We obtain either a subword v which ends by two adjacent basic words or a subword w_1 with the density not less than ρ. In the first case we have proved our theorem since $n(w) \geq p(|w|)$. In the second case we apply Theorem 1 to the word u without the prefix ww_1, and so on. If the second case of Theorem 1 never occurs then $u = ww_1w_2\dots$ where lengths of words w and w_i are bounded. So the density of u is not less than ρ that contradicts to the condition of the Theorem.

The next Corollary follows straightforwardly from Theorem 3.

Corollary 1 *An infinite square-free word which begins with a word b_ib_j has the density not less than ρ.*

From Theorem 1 and Corollary 1 we obtain immediately our main Theorem:

Theorem 4 *The density of a letter a in an infinite square-free word is not less than ρ.*

Using the result from the article [2] and above Theorem we obtain that the minimal density of an infinite square-free word is equal to ρ.

3 Optimization of calculations

We have spent about 12 hours of calculations (at 2GHz CPU) in order to prove Theorems 1 and 2. In order to speed up calculations we used the special algorithm for checking the square-free property.

The program proving above theorems uses the backtrack algorithm. So we must check on every step does the current word S end by word of type ww. Although the algorithm described below uses $O(n^2)$ time in the worst case for checking the square-free property, in average it uses sublinear time. First, we choose m. In our experiments the optimal m was about 100. Then we calculate exact and inexact hash for all positions from mth character to the end of current string. Let us denote by S_i the number corresponding to ith symbol ($S_i = 0$ for symbol a, $S_i = 1$ for symbol b, $S_i = 2$ for symbol c). Let us define *inexact hash* for jth symbol as $h_j = S_j + S_{j-1}p + S_{j-2}p^2 + \cdots + S_{j-m+1}p^{m-1} (\bmod P)$ where $p = 5$ and P is the size of the hash table (we use $P = 100003$ for Theorem 1 and $P = 200003$ for Theorem 2). If there does not exist positions before jth with the exact hash h_j then we define exact hash of jth position equal to inexact hash. If such positions there exist we check last m symbols before jth position. If last m symbols before jth position coinside with last m symbols before kth position with exact hash h_j, we define exact hash for position j equal to h_j. In opposite case we check all positions which have exact hash is equal to $h_j + 1$ and repeat the previous procedure. Then we check positions with exact hash equal to $h_j + 2$ and so on. We continue to increase hash value until we have found the position with the same last m characters or hash value which does not correspond to any position in the string. We define exact hash of the jth position equal to the last checked hash value. In order to check hash values quickly we use the table where for each hash value it is written the last position with this exact hash. Also for each position in string we store the nearest position before it with the same exact hash. So, all positions with the same exact hash are linked to the list.

So, exact hash uniquely defines last m symbols and we can quickly update it after deleting or inserting symbol to the end of the current string.

In order to check does current word S end by word like ww, we check possible lengths (of word w) less than m directly. Then we search all positions with exact hash equal to exact hash of last symbol. Then we compare strings for these positions. We speed up this procedure by m times using exact hash values.

4 Literature

References

[1] Y. Tarannikov. The minimal density of a letter in an infinite ternary square-free word is 0.2746... J. Integer Sequences 5(2): Article 02.2.2 (2002)

[2] P. Ochem. Letter frequency in infinite repetition-free words. Proceedings of Words 2005, Montreal, Canada, September 13-17 (2005)