On the maximal number of cubic subwords in a string[⋆]

Marcin Kubica[1], Jakub Radoszewski[1], Wojciech Rytter[1,2], and Tomasz Waleń[1]

[1] Department of Mathematics, Computer Science and Mechanics,
University of Warsaw, Warsaw, Poland
{kubica,jrad,rytter,walen}@mimuw.edu.pl
[2] Faculty of Mathematics and Informatics,
Copernicus University, Toruń, Poland

Abstract. We investigate the problem of the maximum number of cubic subwords (of the form www) in a given word. We also consider square subwords (of the form ww). The problem of the maximum number of squares in a word is not well understood. Several new results related to this problem are produced in the paper. We consider two simple problems related to the maximum number of subwords which are squares or which are highly repetitive; then we provide a nontrivial estimation for the number of cubes. We show that the maximum number of squares xx such that x is not a primitive word (nonprimitive squares) in a word of length n is exactly $\lfloor \frac{n}{2} \rfloor - 1$, and the maximum number of subwords of the form x^k, for $k \geq 3$, is exactly $n-2$. In particular, the maximum number of cubes in a word is not greater than $n-2$ either. Using very technical properties of occurrences of cubes, we improve this bound significantly. We show that the maximum number of cubes in a word of length n is between $\frac{1}{2}n$ and $\frac{4}{5}n$ [3].

1 Introduction

A repetition is a word composed (as a concatenation) of several copies of another word. The exponent is the number of copies. We are interested in natural exponents higher than 2. In [4] the authors considered also exponents which are not integer.

In this paper we investigate the bounds for the maximum number of highly repetitive subwords in a word of length n. A word is highly repetitive iff it is of the form x^k for some integer k greater than 2. In particular, cubes w^3 and squares x^2 with nonprimitive x are highly repetitive.

The subject of computing maximum number of squares and repetitions in words is one of the fundamental topics in combinatorics on words [16, 20] initiated

[⋆] Supported by grant N206 004 32/0806 of the Polish Ministry of Science and Higher Education.

[3] In particular, we improve the lower bound from the conference version of the paper [19].

by A. Thue [27], as well as it is important in other areas: lossless compression, word representation, computational biology etc.

The behaviour of the function $\mathsf{squares}(n)$ of maximum number of squares in a word of length n is not well understood, though the subject of squares was studied by many authors, see [7, 8, 15, 23]. The best known results related to the value of $\mathsf{squares}(n)$ are, see [11, 13, 14]:

$$n - o(n) \leq \mathsf{squares}(n) \leq 2n - O(\log n) \ .$$

In this paper we concentrate on larger powers of words and show that in this case we can have much better estimations. Let $\mathsf{cubes}(n)$ denote the maximum number of cubes in a word of length n. We show that:

$$\frac{1}{2}n \leq \mathsf{cubes}(n) \ \leq \frac{4}{5}n \ .$$

There are known efficient algorithms for the computation of integer powers in words, see [1, 3, 9, 21, 22].

The powers in words are related to maximal repetitions, also called *runs*. It is surprising that the bounds for the number of runs are much tighter than for squares, this is due to the work of many people [2, 5, 6, 12, 17, 18, 24–26].

Our main result is a new estimation of the number of cubic subwords. We use a new interesting technique in the analysis: the proof of the upper bound is reduced to the proof of an invariant of some abstract algorithm (in our invariant lemma). There is still some gap between upper and lower bound but it is much smaller than the corresponding gap for the number of squares.

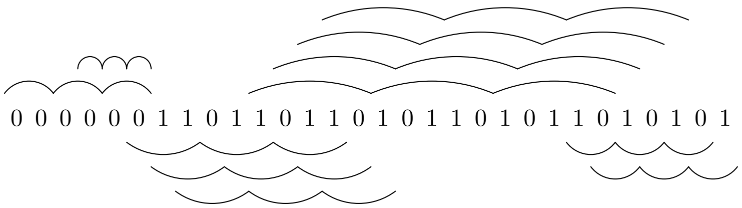

Fig. 1. Example of a word with 11 distinct cubes. This is a word of length 30 with maximal number of cubes among binary words of the same length.

2 Periodicities in strings

We consider *words* over a finite alphabet A, $u \in A^*$; by ε we denote an empty word. The positions in a word u are numbered from 1 to $|u|$. For $u = u_1 \ldots u_k$, by $u[i\,.\,.\,j]$ we denote a *subword* of u equal to $u_i \ldots u_j$; in particular, $u[i] = u[i\,.\,.\,i]$.

We say that a positive integer p is a *period* of a word $u = u_1 \ldots u_k$ if $u_i = u_{i+p}$ holds for $1 \leq i \leq k - p$. If $w^k = u$ (k is a nonnegative integer) then we say that u is the k^{th} power of the word w.

The *primitive root* of a word u, denoted $\mathsf{root}(u)$, is the shortest word w, such that $w^k = u$ for some positive k. We call a word u *primitive* if $\mathsf{root}(u) = u$, otherwise it is called *nonprimitive*. It can be proved that the primitive root of a word u is the only primitive word w, such that $w^k = u$ for some positive k.

A *square* is the 2^{nd} power of some word, and an *np-square* (a nonprimitive square) is a square of a word that is **not** primitive. A *cube* is a 3^{rd} power of some word.

In this paper we focus on the last occurrences of subwords. Hence, whenever we say that word u *occurs at position* i of the word v we mean its **last** occurrence, that is $v[i\,..\,i + |u| - 1] = u$ and $v[j\,..\,j + |u| - 1] \neq u$ for $j > i$. The following lemma is used extensively throughout the article.

Lemma 1 (Periodicity lemma [10, 20]). *If a word of length n has two periods p and q, such that $p + q \leq n + \gcd(p, q)$, then $\gcd(p, q)$ is also a period of the word.*

In this paper we often use, so called, weak version of this lemma, where we only assume that $p + q \leq n$.

3 Basic properties of highly repetitive subwords

A word is said to be *highly repetitive* (hr-word) if it is a k^{th} power of a nonempty word, for $k \geq 3$.

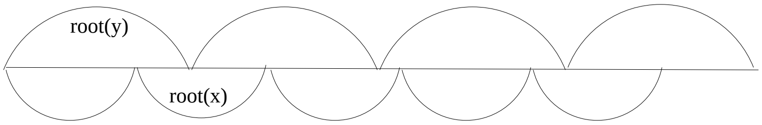

Fig. 2. The situation when one hr-word is a (long) prefix of another hr-word implies that $\mathsf{root}(x) = \mathsf{root}(y)$, consequently x is a suffix of y.

Lemma 2. *If a hr-word x is a prefix of a hr-word y and $|x| \geq |y| - |\mathsf{root}(y)|$, then x is also a suffix of y.*

Proof. Due to the periodicity lemma, both words have the same smallest period and it is a common divisor of the lengths of their primitive roots, see Figure 2. Consequently, we have $\mathsf{root}(x) = \mathsf{root}(y)$ and x is a suffix of y. □

Lemma 3. *Assume that x and y are two hr-words, where $y = z^3$ and x is a subword of y starting at position i and ending at position j such that*

$$i \;\leq\; \left\lceil \frac{|\mathsf{root}(z)|}{2} \right\rceil + 1 \quad \text{and} \quad j > |z^2| \;.$$

Then, $|\mathsf{root}(x)| = |\mathsf{root}(y)|$.

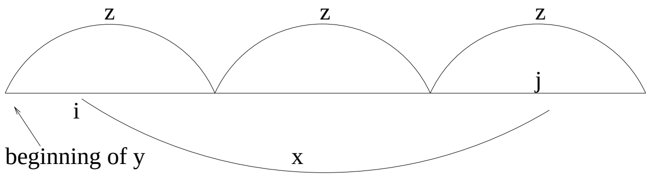

Fig. 3. The situation from Lemma 3.

Proof. Let $x = w^k$, for some $k \geq 3$. Using the inequalities on i and j from the lemma, we obtain:

$$|x| \; = \; j - i + 1 \; \geq \; |z^2| + 1 - \left\lceil \frac{|\mathsf{root}(z)|}{2} \right\rceil - 1 + 1 \; \geq$$

$$\geq \; 2 \cdot |z| - \left\lceil \frac{|z|}{2} \right\rceil + 1 \; \geq \; 2 \cdot |z| - \frac{|z|}{2} \; = \; \frac{3}{2} \cdot |z| \; .$$

Let us also observe that $|\mathsf{root}(x)|$ and $|\mathsf{root}(y)|$ are both periods of x. Moreover:

$$|x| \; = \; |w^k| \; = \; |w| + \frac{k-1}{k} \cdot |x| \; \geq \; |w| + \frac{2}{3} \cdot |x| \; \geq$$

$$\geq \; |w| + |z| \; \geq \; |\mathsf{root}(x)| + |\mathsf{root}(y)| \; .$$

From this, by the periodicity lemma, we obtain that $\gcd(|\mathsf{root}(x)|, |\mathsf{root}(y)|)$ is also a period of x. However, $\mathsf{root}(x)$ and $\mathsf{root}(y)$ are subwords of x, so $|\mathsf{root}(x)| = |\mathsf{root}(y)|$, since in the opposite case one of the words $\mathsf{root}(x), \mathsf{root}(y)$ would not be primitive. □

4 Simple bounds for highly repetitive subwords

In this section we give some simple estimations of the number of square subwords with nonprimitive roots and cubic subwords.

Lemma 4. *Let u be a word. Let us consider highly repetitive subwords of u of the form v^k, for $k \geq 3$ and v primitive. For each such subword we consider its (last) occurrence in u. For each position i in u, at most one such subword can have its (last) occurrence at position i.*

Proof. Let us assume that we have two different hr-words x and y with their last occurrences starting at position i, and let us assume that x is shorter. Then, we have $|x| \geq |y| - |\mathsf{root}(y)|$, otherwise the considered occurrence of x would not be the last one.

Now we can apply Lemma 2 — x is not only a prefix of y, but also its suffix. Hence, x appears later in the text and the last occurrence of x in u does not start at position i. This contradiction proves that the assumption that the last occurrences of x and y start at position i is false. □

The following fact is a consequence of Lemma 4.

Theorem 1. *The maximum number of highly repetitive subwords of a word of length $n \geq 2$ is exactly $n - 2$.*

Proof. From Lemma 4 we know that at each position there can be at most one last occurrence of a nonempty hr-word. Moreover, the minimum possible length of such a word is 3. Therefore, there can be no such occurrences at positions n and $n-1$. On the other hand, this upper bound is reached by the word a^n. $\square$

As a corollary, we obtain a simple upper bound for the number of cubes, since cubes are hr-words.

Corollary 1. *Let us consider a word u of length n. The number of nonempty cubes appearing in u is not greater than $n - 2$.*

We improve this upper bound substantially in the next sections. However, it requires a lot of technicalities. Another implication of Theorem 1 is a tight bound for the number of np-squares.

Theorem 2. *Let u be a word of length n. The maximum number of nonempty np-squares appearing in u is exactly $\left\lfloor \frac{n}{2} \right\rfloor - 1$.*

Proof. Each nonempty np-square can be viewed as v^{2i} for some nonempty primitive v and $i \geq 2$. However, each such np-square contains a subword v^{2i-1}, which is not an np-square (due to the periodicity lemma), but still a hr-word. Hence, the number of nonempty subwords of the form v^{2i-1} (for primitive v and $i \geq 2$), appearing in the given word, is not smaller than the number of nonempty np-squares.

Observe that Theorem 1 limits the total number of both subwords of the form v^{2i} and v^{2i-1} by $n - 2$.

Hence, the total number of nonempty np-squares appearing in the given word is not greater than $\frac{n}{2} - 1$, and since it is integer, it is not greater than $\left\lfloor \frac{n}{2} \right\rfloor - 1$. On the other hand, this upper bound is reached by the word a^n. $\square$

5 The structure of occurrences of cubic subwords

In this section we introduce some combinatorial facts about words that are necessary in the proof of the $\frac{4}{5}n$ upper bound on the number of cubes in a word of length n.

Lemma 5. *Let v^3 and w^3 be two nonempty cubes occurring in a word u at positions i and j respectively, such that:*

$$i \;<\; j \;\leq\; i + \left\lceil \frac{|\mathsf{root}(v)|}{2} \right\rceil \;.$$

Then:

$$|\mathsf{root}(w)| = |\mathsf{root}(v)| \quad \textit{or} \quad |\mathsf{root}(w)| \geq 2 \cdot |\mathsf{root}(v)| - (j - i - 1) \;.$$

Proof. Let us denote $p = |\mathsf{root}(v)|$, $q = |\mathsf{root}(w)|$, and let k be the position of the last letter of w^3.

Case 1.
Let us first consider the case, when the (last) occurrence of w^3 is totally inside v^3. Observe that k must then be within the last of the three v's, since otherwise w^3 would occur in u at position $j + p$ or further (see also Fig. 3). Hence, due to Lemma 3, we obtain $q = p$.

Case 2.
In the opposite case, let x be the maximal prefix of w^3 that lays inside v^3. If $p \neq q$ then $p+q$ must be greater than $|x|$. Indeed, if $p+q \leq |x|$ then both $\mathsf{root}(v)$ and $\mathsf{root}(w)$ would be subwords of x, so if $p \neq q$, then one of them would not be primitive due to the periodicity lemma. Therefore:

$$p + q > |x| > |v^3| - (j - i) \geq 3p - (j - i) \ .$$

Consequently $q \geq 2p - (j - i) + 1$. □

Let us introduce a useful notion of *p-occurrence.*

Definition 1. *A* p-occurrence *is the (last) occurrence of a cube with primitive root of length* p*.*

It turns out that the primitive roots of cubes appearing close to each other cannot be arbitrary. It is formally expressed by the following lemma.

Lemma 6. *Let* $a_1, a_2, \ldots, a_{p+1}$ *be an increasing sequence of positions in a word* u*, such that* $a_{j+1} \leq a_j + p$ *for* $j = 1, 2, \ldots, p$*. It is not possible for all these positions to contain* p*-occurrences.*

Proof. Let us assume, to the contrary, that at each of the positions $a_1, a_2, \ldots, a_{p+1}$ there is a p-occurrence. Observe that the inequalities from the hypothesis of the lemma imply that the primitive roots of cubes occurring at these positions are all cyclic rotations of each other. There are only p different rotations of such primitive roots; therefore, due to the pigeonhole principle, some two of them must be equal.

It suffices to show that all these cubes have the same length, because then some two of them are equal, and consequently one of them is not the last occurrence of the cube.

Assume to the contrary that some of the considered cubes have different lengths. Let a_j and a_{j+1} be two considered positions, such that cubes (v^3 and w^3 respectively) occurring at these positions have different lengths ($3kp$ and $3lp$ respectively, for $k \neq l$). Let us consider two cases.

Case 1. If $l < k$, then $3kp - 3lp \geq 3p$, and w^3 occurs in u at position $a_{j+1} + p$ or further (see Fig. 4).

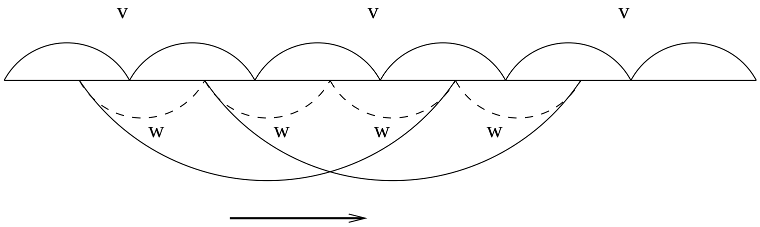

Fig. 4. The positions of cubes v^3 and w^3 in the case $l < k$: a_{j+1} is not the last occurrence of w^3.

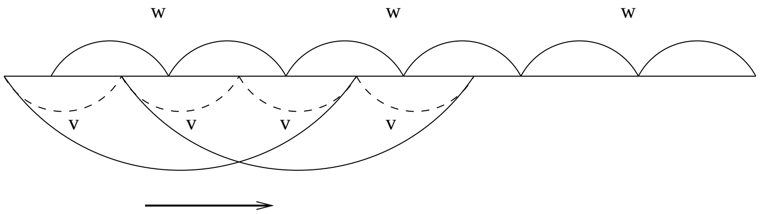

Fig. 5. The positions of cubes v^3 and w^3 in the case $k < l$: a_j is not the last occurrence of v^3.

Case 2. If $k < l$, then $3lp - 3kp \geq 3p$ and v^3 appears in u at position $a_j + p$ or further (see Fig. 5).

In both cases we obtain a contradiction. Hence, it is not possible that the lengths of the cubes differ. □

Let us introduce a notion of independent prefixes.

Definition 2. *We say that v is the* independent prefix *of u if it is the shortest prefix of u that is:*

1. *a single letter word, if there is no occurrence of a cube at the first position of u, or otherwise*
2. *a prefix that ends with a q-occurrence (for some $q \geq 1$) followed by exactly $\left\lceil \frac{q}{2} \right\rceil$ positions without any occurrences (here all occurrences are considered within u).*

It is not obvious that the above definition is valid. Therefore, we prove the following lemma:

Lemma 7. *For every word u, there exists an independent prefix v of u.*

Proof. If there is no occurrence of a cube at the first position of u, then obviously $v = u[1]$.

In the opposite case, let us assume — to the contrary — that the independent prefix does not exist. Let q be the maximum such value, that there exists a q-occurrence in u, and let i be the rightmost position in u that contains a q-occurrence. From Lemma 5, $\lceil \frac{q}{2} \rceil$ positions following i do not contain any occurrences of cubes. Thus, the prefix $u[1..i + \lceil \frac{q}{2} \rceil]$ satisfies the definition of an independent prefix — a contradiction. □

6 Algorithm Abstract-Simulation

Let v be the independent prefix of a word u and let $|v| > 1$. Let $(c_i)_{i=1}^{|v|}$ be a sequence describing the occurrences starting within v: $c_i = 0$ iff there are no occurrences in position $u[i]$, and $c_i = q$ iff there is a q-occurrence in position $u[i]$. We start with the following observations.

a) If c_i and c_j is a pair of consecutive nonzero elements of c (i.e. $i < j$, $c_i, c_j > 0$ and $c_{i+1} = \ldots = c_{j-1} = 0$) then $j - i \le \lceil \frac{c_i}{2} \rceil$. Indeed, if $j - i > \lceil \frac{c_i}{2} \rceil$, then the prefix of u of length $i + \lceil \frac{c_i}{2} \rceil$ or shorter would be an independent prefix of u.

b) For c_i and c_j as in a), $c_j \ge 2c_i - (j - i - 1)$. This observation is due to Lemma 5.

c) From Lemma 6 and due to a) we have that no $q + 1$ consecutive positive elements of c are equal to q.

From now on, we abstract from the actual word u, and focus only on the properties of sequence c. We will analyze the ratio R of nonzero elements of c to the length of c.

Let us observe that if c contains such a pair of equal elements $c_i = c_j > 0$, that all the elements between them are equal zero, then all the elements between c_i and c_j can be removed from c without decreasing R. Also, if c contains a subsequence of consecutive elements equal to q ($q > 0$) of length less than q then this subsequence can be extended to length q without decreasing R. Let c' be the sequence obtained from c by performing the described modification steps (as many times as possible). Observe that none of these steps violates properties a)–c).

Every possible sequence c' can be generated by the (nondeterministic) pseudocode shown below. The following variables are used in the pseudocode:

- p — the value of the last positive element of c'
- len — the length of the sequence c' without $\lceil p/2 \rceil$ trailing zeros
- occ — the number of positive elements in c'
- l — the gap between consecutive different positive elements of c'
- α — the difference between the actual value of a positive element of c' and the lower bound from Lemma 5.

Each step of the **repeat** loop corresponds to extending sequence c', i.e. adding l zeros and p elements of value p.

$$3\ 3\ 3\ 0\ \underbrace{5\ldots5}_{5\text{ times}}\ 0\ 0\ \underbrace{20\ldots20}_{20\text{ times}}\ \underbrace{0\ldots0}_{6\text{ times}}\ \underbrace{34\ldots34}_{34\text{ times}}\ \underbrace{0\ldots0}_{17\text{ times}}$$

Fig. 6. An example of sequence c'. The length of the sequence is 88 and it contains 62 positive elements. The ratio is $62/88 \approx 0.70 < 4/5$.

Note that the algorithm specified by the pseudocode is nondeterministic in several different aspects — the initial value of p, the number of steps of the **repeat** loop and values of l and α.

```
Algorithm Abstract-Simulation
$p :=$ some positive integer;
$occ := p$;    $len := p$;
output: $\underbrace{p\ldots p}_{p\text{ times}}$

repeat an arbitrary number of times
    Invariant $I(p, occ, len) : \frac{occ}{len+\frac{p}{2}} \leq \frac{4}{5}$.
    $l :=$ some integer from interval $[0, \lceil\frac{p}{2}\rceil)$;
    $\alpha :=$ some nonnegative integer;
    $p := 2p - l + \alpha$;
    $occ := occ + p$;
    $len := len + l + p$;
    output: $\underbrace{0\ldots0}_{l\text{ times}}\ \underbrace{p\ldots p}_{p\text{ times}}$
```

7 Upper bound on the number of cubic subwords

Lemma 8 (Invariant lemma). *The following condition $I(p, occ, len)$:*

$$\frac{occ}{len + \frac{p}{2}} \leq \frac{4}{5}$$

is an invariant of the Abstract-Simulation Algorithm.

Proof. Before the first execution of the **repeat** loop, $occ = len = p$, and consequently $I(p, occ, len)$ holds:

$$\frac{p}{p + \frac{p}{2}} = \frac{1}{\frac{3}{2}} = \frac{2}{3} \leq \frac{4}{5}\ .$$

Therefore, we only need to prove that if $I(p, occ, len)$ holds then $I(p', occ', len')$ also holds, where p', occ' and len' are the values obtained as a result of a single step of the **repeat** loop, i.e.:

$$\boxed{\begin{array}{l} p' = 2p - l + \alpha, \\ occ' = occ + 2p - l + \alpha, \\ len' = len + 2p + \alpha. \end{array}}$$

Let us restate $I(p', occ', len')$ equivalently in the following way:

$$5 \cdot occ + 10p - 5l + 5\alpha \;\leq\; 4 \cdot len + 8p + 4\alpha + 4 \cdot \frac{2p - l + \alpha}{2} \;. \tag{1}$$

On the other hand, $I(p, occ, len)$ can be expressed as

$$5 \cdot occ \;\leq\; 4 \cdot len + 4 \cdot \frac{p}{2} \;.$$

Hence, in order to show (1), it is sufficient to prove that:

$$10p - 5l + 5\alpha \;\leq\; 8p + 4\alpha + 2 \cdot (2p - l + \alpha) - 2p \;. \tag{2}$$

As a result of some rearrangement, (2) can be expressed as

$$0 \leq 3l + \alpha$$

and this inequality trivially holds. □

We can now show the upper bound for the number of cubes in independent prefixes.

Lemma 9. *Let v be the independent prefix of u. The number of different nonempty cubes that occur in u and start within v is not greater than $\frac{4}{5} \cdot |v|$.*

Proof. Observe that if v satisfies the first condition of Definition 2, then the conclusion trivially holds. Therefore, from now on we assume that $|v| > 1$.

As described in the previous section, instead of computing the ratio of cubes that occur in u and start within v, we can deal with the ratio R of nonzero elements of the corresponding sequence c to the length of c and show that $R \leq \frac{4}{5}$. For this it suffices to prove that for any valid sequence c' the ratio of nonzero elements does not exceed $\frac{4}{5}$.

The Abstract-Simulation Algorithm generates every possible sequence c'. Hence, in order to prove the $\frac{4}{5}$ bound, we need to show that inequality

$$\frac{occ}{len + \lceil \frac{p}{2} \rceil} \;\leq\; \frac{4}{5}$$

holds for every possible execution of the Algorithm. But this inequality is a consequence of the fact that $I(p, occ, len)$ is an invariant of the Algorithm (Lemma 8). □

Theorem 3. *The number of different nonempty cubes that occur in a word of length n is not greater than $\frac{4}{5}n$.*

Proof. We prove the theorem by induction on n. The basis ($n = 0$) is trivial. Now assume that the conclusion holds for all words of length not exceeding n and consider a word u of length $n+1$. Due to Lemma 7, there exists the independent prefix v of u, $v \neq \varepsilon$, $u = vw$. The cubes occurring within u can be divided into two groups: the ones that start within v and the ones that occur totally inside w. By Lemma 9, the number of cubes in the first group does not exceed $\frac{4}{5}|v|$, and by the inductive hypothesis, $\mathsf{cubes}(w) \leq \frac{4}{5} \cdot |w|$. In total, there are at most

$$\frac{4}{5} \cdot |v| + \frac{4}{5} \cdot |w| \leq \frac{4}{5} \cdot |u|$$

cubes within u — this ends the inductive proof. □

8 Lower bound on the number of cubic subwords

A trivial lower bound on the number of different cubic subwords is the word a^n with $\lfloor \frac{n}{3} \rfloor$ cubic occurrences. The table presented in Figure 7 contains examples of some words with higher number of cubic subwords. These words have been computed using extensive computer experiments.

n	word	#cubes	ratio
20	01110101011011011000	7	0.35
30	000000110110110101101011010101	11	0.36
40	1101101101110111011100010001000100100100	16	0.40
50	11111111110010010010100101001010100101010010101000	20	0.40
60	101001010010100101010010100101010010100101010010101001010100	25	0.41
70	0000001101101101011010110101011010110101011010110101011010101101010111	30	0.42
80	11011011010110110101101101011010110101011010110101011010110101011010101101010111	34	0.42
90	111011011011101101101110110110111011011101101101110110111011011011101101110110111011101110	40	0.44
100	1000101010010101001010100101001010100101001010100101001010010101001010010100101010010100101001010111	44	0.44

Fig. 7. Examples of words with high number of distinct cubic subwords.

Let us proceed to the construction of the $\frac{1}{2}n$ lower bound. For $i \geq 1$, let p_i be the word $0^i 1 0^{i+1} 1$. Let q_n be the concatenation $p_1 p_2 \ldots p_n$. Thus, for instance, $q_4 =$ 01001001000100010000100001000001.

Lemma 10. *The length of q_n is $n^2 + 4n$.*

Proof. Clearly p_i contains $2i+3$ bits, so

$$|q_n| \;=\; \sum_{i=1}^{n} 2i+3 \;=\; n^2+4n \ .$$

□

Lemma 11. *The word q_n contains exactly*

$$\frac{n^2}{2}+\frac{n}{2}-1+\left\lfloor\frac{n+1}{3}\right\rfloor$$

distinct cubes.

Proof. Note that the concatenation $p_i p_{i+1} = 0^i 1 0^{i+1} 1 0^{i+1} 1 0^{i+2} 1$ contains the following $i+1$ cubes:

$$\left(0^i 10\right)^3,\ \left(0^{i-1}10^2\right)^3,\ \ldots,\ \left(010^i\right)^3,\ \left(10^{i+1}\right)^3 \ .$$

Apart from that, in q_n there are $\lfloor\frac{n+1}{3}\rfloor$ cubes of the form $0^3, 0^6, 0^9, \ldots$ Thus far we obtained

$$\sum_{i=1}^{n-1}(i+1)+\left\lfloor\frac{n+1}{3}\right\rfloor \;=\; \frac{n^2}{2}+\frac{n}{2}-1+\left\lfloor\frac{n+1}{3}\right\rfloor$$

cubes.

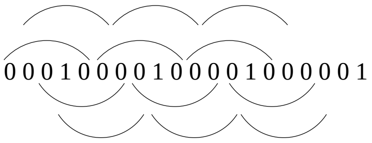

Fig. 8. For $i=3$ the word $p_i p_{i+1}$ contains 4 cubes of length $3i+6=15$.

It remains to show that there are no more cubes in q_n. Notice that we have considered all cubes u^3 for which the number of 1's in u equals 0 or 1. On the other hand, if this number exceeds 1 then u would contain the factor 10^i1 for some $i \geq 1$ and this is impossible, since for a given i such a factor appears within q_n at most twice. □

Theorem 4. *For infinitely many positive integers m there exists a word of length m for which the number of cubes is $\frac{m}{2}-o(m)$.*

Proof. Due to Lemmas 10 and 11, for any word q_n we have:

$$\begin{aligned}\frac{|q_n|}{2}-\mathsf{cubes}(q_n) \;&=\; \frac{n^2}{2}+2n-\frac{n^2}{2}-\frac{n}{2}+1-\left\lfloor\frac{n+1}{3}\right\rfloor = \\ &\frac{3}{2}n-\left\lfloor\frac{n+1}{3}\right\rfloor+1 \;=\; O(n)=o(|q_n|) \ .\end{aligned}$$

Thus, $\mathsf{cubes}(q_n) = \frac{|q_n|}{2} - o(|q_n|)$. □

Interestingly, the example from the paper [11] of a family of words that contain $m - o(m)$ squares is quite similar to our example, but instead of p_i it utilizes words of the form $p'_i = 0^{i+1}10^i10^{i+1}1$.

9 Conclusions

In this paper we prove a tight bound for the number of nonprimitive squares in a word of length n. Unfortunately, this does not improve the overall bound of the number of squares — the main open problem is improving the bound for primitive squares.

We also give some estimations of the number of cubes in a string of length n. These bounds are much better than the best known estimations for squares in general. We believe that at least the upper bound established in our paper is not tight.

References

1. Alberto Apostolico and Franco P. Preparata. Optimal off-line detection of repetitions in a string. *Theor. Comput. Sci.*, 22:297–315, 1983.
2. Pawel Baturo, Marcin Piatkowski, and Wojciech Rytter. The number of runs in sturmian words. In *CIAA 2008*, pages 252–261, 2008.
3. Maxime Crochemore. An optimal algorithm for computing the repetitions in a word. *Inf. Process. Lett.*, 12(5):244–250, 1981.
4. Maxime Crochemore, Szilard Zsolt Fazekas, Costas S. Iliopoulos, and Inuka Jayasekera. Bounds on powers in strings. In *DLT*, pages 206–215, 2008.
5. Maxime Crochemore and Lucian Ilie. Maximal repetitions in strings. *J. Comput. Syst. Sci.*, 74(5):796–807, 2008.
6. Maxime Crochemore, Lucian Ilie, and Liviu Tinta. Towards a solution to the "runs" conjecture. In Paolo Ferragina and Gad M. Landau, editors, *CPM*, volume 5029 of *Lecture Notes in Computer Science*, pages 290–302. Springer, 2008.
7. Maxime Crochemore and Wojciech Rytter. Squares, cubes, and time-space efficient string searching. *Algorithmica*, 13(5):405–425, 1995.
8. Maxime Crochemore and Wojciech Rytter. *Jewels of Stringology.* World Scientific, 2003.
9. David Damanik and Daniel Lenz. Powers in sturmian sequences. *Eur. J. Comb.*, 24(4):377–390, 2003.
10. N. J. Fine and H. S. Wilf. Uniqueness theorems for periodic functions. *Proceedings of the American Mathematical Society*, 16:109–114, 1965.
11. A. S. Fraenkel and J. Simpson. How many squares can a string contain? *J. of Combinatorial Theory Series A*, 82:112–120, 1998.
12. Mathieu Giraud. Not so many runs in strings. In Carlos Martín-Vide, Friedrich Otto, and Henning Fernau, editors, *LATA*, volume 5196 of *Lecture Notes in Computer Science*, pages 232–239. Springer, 2008.
13. L. Ilie. A simple proof that a word of length n has at most $2n$ distinct squares. *J. of Combinatorial Theory Series A*, 112:163–164, 2005.

14. L. Ilie. A note on the number of squares in a word. *Theoretical Computer Science*, 380:373–376, 2007.
15. Costas S. Iliopoulos, Dennis Moore, and William F. Smyth. A characterization of the squares in a fibonacci string. *Theor. Comput. Sci.*, 172(1-2):281–291, 1997.
16. Juhani Karhumaki. Combinatorics on words. *Notes in pdf.*
17. Roman M. Kolpakov and Gregory Kucherov. Finding maximal repetitions in a word in linear time. In *FOCS*, pages 596–604, 1999.
18. Roman M. Kolpakov and Gregory Kucherov. On maximal repetitions in words. In Gabriel Ciobanu and Gheorghe Paun, editors, *FCT*, volume 1684 of *Lecture Notes in Computer Science*, pages 374–385. Springer, 1999.
19. Marcin Kubica, Jakub Radoszewski, Wojciech Rytter, and Tomasz Walen. On the maximal number of cubic subwords in a string. In *Proceedings of the 20th International Workshop on Combinatorial Algorithms (to appear)*, 2009.
20. M. Lothaire. *Applied Combinatorics on Words.* Cambridge University Press, Cambridge, UK, 2005.
21. Michael G. Main. Detecting leftmost maximal periodicities. *Discrete Applied Mathematics*, 25(1–2):145–153, 1989.
22. Michael G. Main and Richard J. Lorentz. An o(n log n) algorithm for finding all repetitions in a string. *J. Algorithms*, 5(3):422–432, 1984.
23. Marcin Piatkowski and Wojciech Rytter. Asymptotic behaviour of the maximal number of squares in standard sturmian words. In *Prague Stringology Conference*, pages 237–248, 2009.
24. Simon J. Puglisi, Jamie Simpson, and William F. Smyth. How many runs can a string contain? *Theor. Comput. Sci.*, 401(1-3):165–171, 2008.
25. Wojciech Rytter. The number of runs in a string: Improved analysis of the linear upper bound. In Bruno Durand and Wolfgang Thomas, editors, *STACS*, volume 3884 of *Lecture Notes in Computer Science*, pages 184–195. Springer, 2006.
26. Wojciech Rytter. The number of runs in a string. *Inf. Comput.*, 205(9):1459–1469, 2007.
27. A. Thue. Uber unendliche zeichenreihen. *Norske Vid. Selsk. Skr. I Math-Nat.*, 7:1–22, 1906.