On Prefix Normal Words and Prefix Normal Forms

Péter Burcsi[a], Gabriele Fici[b], Zsuzsanna Lipták[c,*], Frank Ruskey[d], Joe Sawada[e]

[a]*Dept. of Computer Algebra, Eötvös Loránd Univ., Budapest, Hungary*
[b]*Dip. di Matematica e Informatica, University of Palermo, Italy*
[c]*Dip. di Informatica, University of Verona, Italy*
[d]*Dept. of Computer Science, University of Victoria, Canada*
[e]*School of Computer Science, University of Guelph, Canada*

Abstract

A 1-prefix normal word is a binary word with the property that no factor has more 1s than the prefix of the same length; a 0-prefix normal word is defined analogously. These words arise in the context of indexed binary jumbled pattern matching, where the aim is to decide whether a word has a factor with a given number of 1s and 0s (a given Parikh vector). Each binary word has an associated set of Parikh vectors of the factors of the word. Using prefix normal words, we provide a characterization of the equivalence class of binary words having the same set of Parikh vectors of their factors.

We prove that the language of prefix normal words is not context-free and is strictly contained in the language of pre-necklaces, which are prefixes of powers of Lyndon words. We give enumeration results on $pnw(n)$, the number of prefix normal words of length n, showing that, for sufficiently large n,

$$2^{n-4\sqrt{n \lg n}} \leq pnw(n) \leq 2^{n-\lg n+1}.$$

For fixed density (number of 1s), we show that the ordinary generating function of the number of prefix normal words of length n and density d is a rational function. Finally, we give experimental results on $pnw(n)$, discuss further properties, and state open problems.

Keywords: prefix normal words, prefix normal forms, binary languages,

*Corresponding author

Email addresses: bupe@compalg.inf.elte.hu (Péter Burcsi), gabriele.fici@unipa.it (Gabriele Fici), zsuzsanna.liptak@univr.it (Zsuzsanna Lipták), ruskey@cs.uvic.ca (Frank Ruskey), jsawada@uoguelph.ca (Joe Sawada)

Preliminary versions of parts of this article have appeared in [21], [10] and [9].

binary jumbled pattern matching, pre-necklaces, Lyndon words, enumeration.

1. Introduction

A binary word is called 1-*prefix normal* if no factor (substring) has more 1s than the prefix of the same length. For example, 11010 is 1-prefix normal, but 10110 is not. Similarly, a binary word is called 0-*prefix normal* if no factor has more 0s than the prefix of the same length. When not further specified, by *prefix normal* we mean 1-prefix normal. In [10], we gave an algorithm for generating all prefix normal words of fixed length n. As we will see later, to each binary word, a 1-prefix normal word and a 0-prefix normal word can be associated in a unique way, which we will call its *prefix normal forms.*

The *Parikh vector* of a binary word u is the pair (x, y), where x is the number of 1s in u, and y is the number of 0s in u. The set of Parikh vectors of factors of a word w is called the *Parikh set* of w. For binary words, the problem of deciding whether a particular pair (x, y) lies in the Parikh set of a word w is known as *Binary Jumbled Pattern Matching* (BJPM). There has been much interest recently in the indexed version of this problem (IBJPM), where an index for the Parikh set is created in a preprocessing step, which can then be used to answer queries fast. The Parikh set can be represented in linear space due to the following *interval property* of binary strings: If w has k-length substrings with x_1 resp. x_2 occurrences of 1, where $x_1 < x_2$, then it also has a k-length substring with y occurrences of 1, for every $x_1 \leq y \leq x_2$. Thus the Parikh set can be represented by storing, for every $1 \leq k \leq |w|$, the minimum and maximum number of 1s in a substring of length k. Much recent research has focused on how to compute these numbers efficiently [14, 29, 30, 16, 2, 23, 22]. The problem has also been extended to graphs and trees [22, 15], to the streaming model [27], and to approximate indexes [16]. There is also interest in the non-binary variant [20, 17, 11, 14, 7, 8, 26], as well as in reconstruction from the Parikh multi-set of a string [1]. Applications in computational biology include SNP discovery, alignment, gene clusters, pattern discovery, and mass spectrometry data interpretation [4, 3, 5, 19, 33].

The current best construction algorithm for the linear size index for IB-JPM runs in $O(n^{1.864})$ time [13], for a word of length n. As we will see later, computing the prefix normal forms of a word w is equivalent to creating an index for the Parikh set of w. Currently, we know no faster computation algorithms for the prefix normal forms than already exist for the linear-size index. However, should better algorithms be discovered, these would immediately carry over to the problem of IBJPM.

It is worthwhile noting that some relevant sequences have made it into the On-Line Encyclopedia of Integer Sequences (OEIS [35]): A194850 is the number of prefix normal words of length n, A238109 is a list of prefix normal words (over the alphabet $\{1, 2\}$), and A238110 is the maximum size of a class of binary words of length n having the same prefix normal form.

The paper is organized as follows: Section 2 contains basic definitions and results about prefix normal words; in particular that there are unique 0-prefix normal and 1-prefix normal words associated with every word, and thus the set of words can be partitioned according to this association. In Section 3 we consider the set of prefix normal words, giving several properties and characterizations and showing that their language is not context free. One of these properties is then used in Section 4, which is concerned with counting the number of prefix normal words of a given length. Finally, the paper concludes with some open problems in Section 5.

2. Basics

A *binary word* (or *string*) $w = w_1 \cdots w_n$ over $\Sigma = \{0, 1\}$ is a finite sequence of elements $w_i \in \Sigma$, for $i = 1, \ldots, n$. Its length n is denoted by $|w|$. We denote by Σ^n the set of words over Σ of length n, by $\Sigma^* = \cup_{n \geq 0} \Sigma^n$ the set of finite words over Σ, and the empty word by ε. Let $w \in \Sigma^*$. If $w = uv$ for some $u, v \in \Sigma^*$, we say that u is a *prefix* of w and v is a *suffix* of w. A *factor* or *substring* of w is a prefix of a suffix of w. We denote the set of factors of w by $\mathit{Fact}(w)$. Let $w = w_1 \cdots w_n \in \Sigma^*$, then the word $\tilde{w} = w_n w_{n-1} \cdots w_1$ is called the *reversal* of w. A word w s.t. $w = \tilde{w}$ is called a *palindrome*. A *binary language* is any subset $\mathcal{L}$ of Σ^*.

We denote by $|w|_1$ the number of 1s in the word w; similarly, $|w|_0$ is the number of 0s in w. The *Parikh vector* of a word w over Σ is defined as $p(w) = (|w|_0, |w|_1)$. The *Parikh set* of w is $\Pi(w) = \{p(v) \mid v \in \mathit{Fact}(w)\}$, the set of Parikh vectors of the factors of w. For example $p(011) = p(101) = (1, 2)$ and $\Pi(011) = \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2)\} = \Pi(101) \cup \{(0, 2)\}$.

Given a binary word w, we denote by $P_1(w, i)$ the number of 1s in the prefix of length i and by $\mathit{pos}_1(w, i)$ the position of the ith 1 in the word w, i.e. $P_1(w, i) = |w_1 \cdots w_i|_1$ and $\mathit{pos}_1(w, i) = \min\{k : |w_1 \cdots w_k|_1 = i\}$. The functions P_0 and pos_0 are defined similarly. Note that in the context of

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
F_1	0	1	2	3	3	4	4	4	5	6	6	7	7	7	8	8	9	10	10	10	11	11	12
F_0	0	1	2	3	3	3	4	4	5	5	6	6	7	7	7	8	8	9	9	10	10	10	10

Table 1: The sequences F_1 and F_0 for the word $w = 1010011011000111001011$.

succint indexing, these functions are frequently called *rank* and *select*, cf. [32]: We have, for $x = 0, 1$, $P_x(w, i) = \mathit{rank}_x(w, i)$ and $\mathit{pos}_x(w, i) = \mathit{select}_x(w, i)$.

2.1. Prefix normal words

Definition 1 (Maximum-ones and maximum-zeros functions). Let $w \in \Sigma^*$. We define, for each $0 \leq k \leq |w|$:

$$F_1(w, k) = \max\{|v|_1 \mid v \in \mathit{Fact}(w) \cap \Sigma^k\},$$

the maximum number of 1s in a factor of w of length k. When no confusion can arise, we also write $F_1(k)$ for $F_1(w, k)$. The function $F_0(w, k)$ is defined analogously by taking 0 in place of 1.

For a word w, we denote by $F_1(w)$ the function $k \mapsto F_1(w, k)$ (and similarly with other functions taking arguments w and k).

Example 1. Take $w = 1010011011000111001011$. In Table 1, we give the values of F_1 and F_0 for w.

Definition 2 (Prefix normal words). A word $w \in \{0, 1\}^*$ is called 1-*prefix normal* if $P_1(w) = F_1(w)$. It is called 0-*prefix normal* if $P_0(w) = F_0(w)$. In other words, w is 1-prefix normal (0-prefix normal) if and only if it does not have any factors with more 1s (more 0s) than the prefix of the same length. When not specified, then by *prefix normal* we mean 1-*prefix normal*.

Example 2. The word $w = 1100110$ is 1-prefix normal, but the word $w1 = 11001101$ is not 1-prefix normal because the factor 1101 has three 1s, while the prefix of length 4 has only two. Also, w is not 0-prefix normal since every 0-prefix normal word, except those of the form 1^*, must start with a 0.

We will soon see that it is possible to find, for every word w, a 1-prefix normal word which has the same maximum-ones function F_1 as w; and analogously for 0. These will be called the prefix normal forms of w. To this end, we define the following equivalence; we will then see that equivalent words have the same prefix normal form.

Definition 3 (Prefix equivalence). Two words $v, w \in \Sigma^*$ are called 1-*prefix equivalent* if $F_1(v) = F_1(w)$. They are called 0-*prefix equivalent* if $F_0(v) = F_0(w)$.

Example 3. The words $11010, 10110, 01101, 01011$ are all 1-prefix equivalent, but not 0-prefix equivalent. When considering 0, we have that $\{01011,$ $11010,$ $10101\}$ constitute one equivalence class, and $\{01101, 10110\}$ another one (note that in the first class, there is an additional word not present in the 1-prefix equivalence class).

Next we will show that every equivalence class contains exactly one prefix normal word (Theorem 2), which can thus be used as its representative. This will allow us to associate two prefix normal words to every word w (Definition 4). First we need the following lemma.

Lemma 1. *Let $w \in \Sigma^*$. Then, for all $0 \leq i \leq j \leq |w|$: $F_1(j) - F_1(i) \leq F_1(j-i)$.*

Proof. Observe that if $v = yz$, then $|v|_1 \leq F_1(|y|) + F_1(|z|)$. Thus if v is a length j word such that $|v|_1 = F_1(j)$ and $|y| = i$, then $F_1(j) \leq F_1(i) + F_1(j-i)$. □

Theorem 2. *For every $w \in \Sigma^*$ there is a unique 1-prefix normal word w' such that $F_1(w') = F_1(w)$; similarly, there is a unique 0-prefix normal word w'' such that $F_0(w'') = F_0(w)$.*

Proof. We only give the proof for w'. The construction of w'' is analogous.

First note that if the 1-prefix normal words u and v are 1-prefix equivalent, then necessarily $u = v$. This holds because the prefix function P_1 determines the word, i.e. $P_1(u) = P_1(v)$ implies $u = v$ for *any* u, v. But since u and v are 1-prefix normal words, their prefix and maximum-ones functions coincide, and since they are 1-prefix equivalent, we have $P_1(u) = F_1(u) = F_1(v) = P_1(v)$. This proves uniqueness.

Next, we will construct w', given w. It is easy to see that for $1 \leq k \leq |w|$, one has either $F_1(w,k) = F_1(w,k-1)$ or $F_1(w,k) = 1 + F_1(w,k-1)$. Now define the word w' by

$$w'_k = \begin{cases} 1 & \text{if } F_1(w,k) = 1 + F_1(w,k-1) \\ 0 & \text{if } F_1(w,k) = F_1(w,k-1) \end{cases}$$

for every $1 \leq k \leq |w|$.

By construction, we have $P_1(w',k) = F_1(w,k)$ for every $1 \le k \le |w|$. We still need to show that $P_1(w',k) = F_1(w',k)$ for all k. This will prove that w' is 1-prefix normal, as well as that it is 1-prefix equivalent to w.

By definition, $P_1(w',k) \le F_1(w',k)$ for all k. Now let $v \in Fact(w')$, $|v| = k$, and $v = w_{i+1} \cdots w_j$. Then $|v|_1 = P_1(w',j) - P_1(w',i) = F_1(w,j) - F_1(w,i) \le F_1(w,j-i) = P_1(w',j-i) = P_1(w',k)$, where the inequality holds by Lemma 1. We have thus proved that $F_1(w',k) \le P_1(w',k)$, and hence w' is 1-prefix normal. $\square$

2.2. Normal forms and Parikh sets

Definition 4 ((Prefix) normal forms). Let $w \in \Sigma^*$. Then we denote by $\mathrm{PNF}_1(w)$ the unique 1-prefix normal word which is 1-prefix equivalent to w, and by $\mathrm{PNF}_0(w)$ the unique 0-prefix normal word which is 0-prefix equivalent to w. We refer to $\mathrm{PNF}_1(w)$ and $\mathrm{PNF}_0(w)$ as the *prefix normal form w.r.t.* 1 *(resp. w.r.t.* 0*)* or just *normal form w.r.t.* 1 *(resp. w.r.t.* 0*)* of w.

Example 4. Let $w = 1010011011000111001011$. The normal forms of w are the words

$$\mathrm{PNF}_1(w) = 1110100110100101100101,$$

$$\mathrm{PNF}_0(w) = 0001101010101101010111.$$

Refer to Example 1 for the values of the two functions $F_1(w)$ and $F_0(w)$.

The operators PNF_1 and PNF_0 are idempotent operators; i.e., if $u = \mathrm{PNF}_x(w)$ then $\mathrm{PNF}_x(u) = u$, for $x = 0, 1$. This gives us an equivalent definition of prefix normality: a word w is x-prefix normal if $\mathrm{PNF}_x(w) = w$. Also, for any $w \in \Sigma^*$ and $x \in \Sigma$, it holds that $\mathrm{PNF}_x(w) = \mathrm{PNF}_x(\tilde{w})$. Note further that if the equivalence class of w contains only one element, then w is necessarily prefix normal and a palindrome. In Table 2 we list all eight 1-prefix equivalence classes for words of length 4.

The normal forms of a word allow us to determine the Parikh vectors of the factors of the word, as we will show in Theorem 4. We first recall the following lemma from [14] (which also appears to be folklore). We say that a Parikh vector q *occurs* in a word w if w has a factor v with $p(v) = q$.

Lemma 3 (Interval Lemma [14]). *Let $w \in \Sigma^*$. Fix $1 \le k \le |w|$. If the Parikh vectors $(x_1, k - x_1)$ and $(x_2, k - x_2)$ both occur in w, then so does $(y, k - y)$ for any $x_1 \le y \le x_2$.*

PNF_1	class	cardinality
1111	{1111}	1
1110	{1110, 0111}	2
1101	{1101, 1011}	2
1100	{1100, 0110, 0011}	3
1010	{1010, 0101}	2
1001	{1001}	1
1000	{1000, 0100, 0010, 0001}	4
0000	{0000}	1

Table 2: The sets of 1-prefix equivalent words of length 4.

The lemma can be proved with a simple sliding window argument, exploiting the fact that when a fixed size window is shifted by one, then the number of 1s in the window changes by at most one.

Theorem 4. *Let w, w' be words over Σ. Then $\Pi(w) = \Pi(w')$ if and only if $\mathrm{PNF}_1(w) = \mathrm{PNF}_1(w')$ and $\mathrm{PNF}_0(w) = \mathrm{PNF}_0(w')$.*

Proof. Let $f_1(w, k)$ denote the minimum number of 1s in a factor of w of length k. As a direct consequence of Lemma 3, we have that for a Parikh vector $q = (x, y)$, $q \in \Pi(w)$ if and only if $f_1(w, x+y) \leq x \leq F_1(w, x+y)$. Thus for two words w, w', we have $\Pi(w) = \Pi(w')$ if and only if $F_1(w) = F_1(w')$ and $f_1(w) = f_1(w')$. It is easy to see that for all k, $f_1(w, k) = k - F_0(w, k)$, thus the last statement is equivalent to $F_1(w) = F_1(w')$ and $F_0(w) = F_0(w')$. This holds if and only if $\mathrm{PNF}_1(w) = \mathrm{PNF}_1(w')$ and $\mathrm{PNF}_0(w) = \mathrm{PNF}_0(w')$, and the claim is proved. $\square$

Define $I(w) = \{(P_0(w, k), P_1(w, k)) \mid 0 \leq k \leq |w|\}$, the set of Parikh vectors of all prefixes of w. The following lemma is immediate.

Lemma 5. *For all $w \in \Sigma^*$,*

$$\Pi(w) = \bigcup_{i=1}^{n} I(w_i \cdots w_n).$$

There is an interesting geometrical way to view Lemma 5 which we describe now. Imagine each Parikh pair as the coordinates of a point in the

Euclidean plane that has been rotated clockwise $\pi/4$ radians. Each word w can be interpreted as a polygonal path in this plane going up and to the right for each 1 ($\nearrow$) or down and to the right for each 0 ($\searrow$), for each successive bit of w. To obtain $\Pi(w)$ imagine grabbing the polygonal path for w and pulling it one step at a time through the origin, keeping track of the integer lattice points that are hit after each pull (and ignoring the stuff to the left of the origin). The normal forms $\mathrm{PNF}_1(w)$ and $\mathrm{PNF}_0(w)$ are obtained by forming polygonal paths starting at the origin, and connecting the uppermost and the lowermost points of the region, respectively.

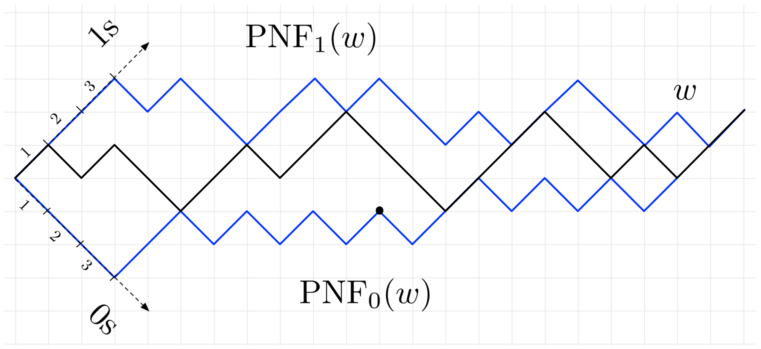

Figure 1: The word w = 10100110111000111001011 (dark line), its normal forms $\mathrm{PNF}_1(w)$ = 11101001101001011001 01 and $\mathrm{PNF}_0(w)$ = 0001101010101101010111 (lighter lines); the region between the two is the Parikh set of w; e.g. w has a substring containing 5 ones and 6 zeros (black dot). Note that the axes giving the number of 0s and 1s are rotated by 45 degrees clockwise.

2.3. Indexing for binary jumbled pattern matching

Theorem 4 is relevant for the problem known as Indexed Binary Jumbled Pattern Matching, which has attracted much interest recently. Recall that a Parikh vector over $\{0, 1\}$ is a multiplicity vector of a string, i.e. it has non-negative integer entries.

> INDEXED BINARY JUMBLED PATTERN MATCHING (IBJPM)
> Given a string w of length n over $\{0, 1\}$, create an index which allows fast answers to queries of the following form:
> **Input:** a Parikh vector q,
> **Output:** return **yes** if q occurs in $\Pi(w)$, and **no** otherwise.

For $1 \le k \le n$, let $f_1(w, k)$ be the minimum number of 1s in a factor of length k, and $F_1(w, k)$, as before, the maximum number of 1s in a factor

k	0	1	2	3	4	5	6	7
$F_1(w,k)$	0	1	2	2	3	3	3	4
$f_1(w,k)$	0	0	0	1	2	2	3	4

Table 3: The maximum and minimum number of 1s for the the word $w = 1001101$.

k	0	1	2	3	4	5	6	7
$F_1(w,k)$	0	1	2	2	3	3	3	4
$F_0(w,k)$	0	1	2	2	2	3	3	3

Table 4: The maximum number of 1s and 0s for the the word $w = 1001101$. The normal forms of w are $\mathrm{PNF}_1(w) = 1101001$ and $\mathrm{PNF}_0(w) = 0011011$.

of length k. It follows from Lemma 3 that the answer for query $q = (x, y)$ is **yes** if and only if $F_1(w, x+y) \geq x \geq f_1(w, x+y)$. Therefore, it suffices to store, for every $1 \leq k \leq n$, the two numbers $F_1(w,k)$ and $f_1(w,k)$, and queries can be answered in constant time. The size of this data structure is $O(n)$.

All current solutions for IBJPM are based on this observation. The crux is how to construct this linear size data structure. The construction time of the index has steadily decreased since its first introduction: from $O(n^2)$ [14] to $O(n^2/\log n)$ [6, 29], to $O(n^2/\log^2 n)$ in the word RAM-model [30], to $n^2/2^{\Omega(\log n/\log\log n)^{1/2}}$ [24]. The fastest solution at present is due to Chan and Lewenstein and has running time $O(n^{1.859})$ [13].

Normal forms are in effect an encoding of this linear size index. We have already seen that the F-function can be viewed as a binary string, namely $\mathrm{PNF}_1(w)$. We have observed in the proof of Theorem 4 how the function $f_1(w)$ is determined by $F_0(w)$ and thus also by $\mathrm{PNF}_0(w)$, thus we have shown the following lemma.

Lemma 6. *The answer for an IBJPM query $q = (x, y)$ is* **yes** *if and only if $P_1(\mathrm{PNF}_1(w), x+y) \geq x \geq P_1(\mathrm{PNF}_0(w), x+y)$.*

Note that P_1 can be computed in constant time with constant time rank-queries on bit vectors, using only $o(n)$ bits of extra space [31, 18].

Example 5. Let $w = 1001101$. Then the linear size data structure is given in the Table 3, and the F_1 and F_0 functions in Table 4.

At present, no faster computation of the normal forms is known than the algorithms cited above for the IBJPM problem. But the connection shown here implies that, should a fast normal form computation be found, it would immediately translate into a new solution for IBJPM.

3. The language of prefix normal words

In this section, we take a closer look at prefix normal words. We give several equivalent characterizations of prefix normality, explore some properties of prefix normal words, and then look at the language of prefix normal words. We denote by $\mathcal{L}_{\mathrm{PN}1} \subset \Sigma^*$ the language of 1-prefix normal words, and by $\mathcal{L}_{\mathrm{PN}0} \subset \Sigma^*$ the language of 0-prefix normal words. Note that these are exactly complemented, i.e. replacing every 1 by a 0 and vice versa, in each word of $\mathcal{L}_{\mathrm{PN}1}$, yields $\mathcal{L}_{\mathrm{PN}0}$. Therefore, every result about $\mathcal{L}_{\mathrm{PN}1}$ has an equivalent formulation for $\mathcal{L}_{\mathrm{PN}0}$, as well. Recall that whenever not further specified, we refer to 1-prefix normality. In Section 3.2 only, we will talk about 0-prefix normal words, and we will show that $\mathcal{L}_{\mathrm{PN}0}$ is strictly contained in the language of pre-necklaces, when adopting the usual order $0 < 1$ on the alphabet.

3.1. General observations about prefix normal words

We start with several characterizations of prefix normal words.

Proposition 7. *Let $w \in \Sigma^*$. The following properties are equivalent:*

1. *w is a prefix normal word;*
2. *$\forall i, j$ where $0 \le i \le j \le |w|$, we have $P_1(j) - P_1(i) \le P_1(j-i)$;*
3. *$\forall v \in \mathit{Fact}(w)$ such that $|v|_1 = i$, we have $|v| \ge \mathit{pos}_1(i)$;*
4. *$\forall i, j$ such that $i + j - 1 \le |w|_1$, we have $\mathit{pos}_1(i) + \mathit{pos}_1(j) - 1 \le \mathit{pos}_1(i+j-1)$.*

Proof. (1) $\Rightarrow$ (2). Follows from Lemma 1, since $P_1(w) = F_1(w)$.

(2) $\Rightarrow$ (3). Assume otherwise. Then there exists $v \in \mathit{Fact}(w)$ s.t. $|v| < \mathit{pos}_1(k)$, where $k = |v|_1$. Let $v = w_{i+1} \cdots w_j$, thus $j - i = k$. Then $P_1(j) - P_1(i) = k$. But $P_1(j-i) = P_1(|v|) \le k - 1 < k = P_1(j) - P_1(i)$, a contradiction.

(3) $\Rightarrow$ (4). Again assume that the claim does not hold. Then there are i, j s.t. $\mathit{pos}_1(i+j-1) < \mathit{pos}_1(i) + \mathit{pos}_1(j) - 1$. Let $k = \mathit{pos}_1(j)$ and $l = \mathit{pos}_1(i+j-1)$ and define $v = w_k \cdots w_l$. Then v has i many 1s. But $|v| = \mathit{pos}_1(i+j-1) - \mathit{pos}_1(j) + 1 < \mathit{pos}_1(i) + \mathit{pos}_1(j) - 1 - \mathit{pos}_1(j) + 1 = \mathit{pos}_1(i)$, in contradiction to (3).

(4) $\Rightarrow$ (1). Let $v \in \mathit{Fact}(w)$, $|v|_1 = i$. We have to show that $P_1(|v|) \geq i$. This is equivalent to showing that $\mathit{pos}_1(i) \leq |v|$. Let $v = w_{l+1} \cdots w_r$, thus $P_1(r) - P_1(l) = i$. Let $j = P_1(l) + 1$, thus the first 1 in v is the j'th 1 of w. Note that we have $l < \mathit{pos}_1(j)$ and $r \geq \mathit{pos}_1(i + j - 1)$. By the assumption, we have $\mathit{pos}_1(i) \leq \mathit{pos}_1(i + j - 1) - \mathit{pos}_1(j) + 1 \leq r - l = |v|$. □

Next we formulate a characterization of the prefix normal property that will be useful in the enumeration of fixed-length prefix normal words (Section 4).

Lemma 8. *Let $w \in 1\Sigma^*$. For some sequence of positive integers r_1, r_2, ..., r_{d-1} we can write $w = 10^{r_1-1}10^{r_2-1} \cdots 10^{r_d-1}$. The word w is prefix normal if and only if the following inequalities hold.*

$$\begin{array}{rcll} r_1 & \leq & r_j & j = 2, 3, \ldots, d-1 \\ r_1 + r_2 & \leq & r_j + r_{j+1} & j = 2, 3, \ldots, d-2 \\ & \vdots & & \vdots \\ r_1 + r_2 + \cdots + r_{d-2} & \leq & r_j + r_{j+1} + \cdots + r_{d-1} & j = 2 \end{array}$$

Proof. Note that for $k = 1, 2, \ldots d-1$, we have $\mathit{pos}_1(k) = 1 + \sum_{j=1}^{k-1} r_j$. The statement of the lemma then follows by property (4) of Proposition 7. □

We now give some simple facts about the language $\mathcal{L}_{\text{PN1}}$.

Proposition 9. *Let $\mathcal{L}_{PN1}$ be the language of prefix normal words.*

1. *$\mathcal{L}_{PN1}$ is prefix-closed, that is, any prefix of a word in $\mathcal{L}_{PN1}$ is a word in $\mathcal{L}_{PN1}$.*
2. *If $w \in \mathcal{L}_{PN1}$, then any word of the form $1^k w$ or $w0^k$, $k \geq 0$, also belongs to $\mathcal{L}_{PN1}$.*
3. *Let $|w|_1 < 3$. Then $w \in \mathcal{L}_{PN1}$ iff either $w = 0^n$ for some $n \geq 0$ or the first letter of w is 1.*
4. *Let $w \in \Sigma^*$. Then there exist infinitely many $v \in \Sigma^*$ such that $vw \in \mathcal{L}_{PN1}$.*

Proof. The claims *1., 2., 3.* follow easily from the definition. For *4.*, note that for any $n \geq |w|$, the word $1^n w$ belongs to $\mathcal{L}_{\text{PN1}}$. □

We now deal with the question of how a prefix normal word can be extended to the right into another prefix normal word.

Lemma 10. *Let $w \in \mathcal{L}_{PN1}$. Then $w1 \in \mathcal{L}_{PN1}$ if and only if for every $0 \leq k < |w|$ the suffix of w of length k has less 1s than the prefix of w of length $k+1$.*

Proof. Note that for all $1 \leq k \leq |w|$, $P_1(w1, k) = P_1(w, k)$. Now if $w1 \in \mathcal{L}_{\text{PN1}}$, then for the k-length suffix u of w: $|u|_1 < |u1|_1 \leq P_1(w1, k+1) = P_1(w, k+1)$. Conversely, let u be a factor of $w1$. If u is a factor of w, then $|u|_1 \leq P_1(w, |u|) = P_1(w1, |u|)$. Else $u = u'1$, with u' a suffix of w, and $|u|_1 = |u'|_1 + 1 < P_1(w, |u'|+1) + 1 = P_1(w1, |u|) + 1 = P_1(w1, |u|) + 1$, and thus $|u|_1 \leq P_1(w1, |u|)$. Therefore, $w1 \in \mathcal{L}_{\text{PN1}}$. □

We close this section by proving that $\mathcal{L}_{\text{PN1}}$ is not context-free.

Theorem 11. *$\mathcal{L}_{PN1}$ is not context-free.*

Proof. Recall that the intersection of a CFL with a regular language is a CFL. We will show that $L' = \mathcal{L}_{\text{PN1}} \cap 1^*01^*01^*$ is not a CFL by using the pumping lemma. Let n be the constant of the pumping lemma and let $z = 1^n01^n01^n \in L'$. Let $z = uvwxy$ be the usual factorization of the pumping lemma, where we may assume that $|vx| \geq 1$, $|vwx| \leq n$, and for all $i \geq 0$ we have $uv^iwx^iy \in L'$. Clearly vx can not contain 0s. If vx contains some 1s from the first block of 1s in z, then taking $i = 0$ give a contradiction since the third block of 1s is too long. If vx contains no 1s from the first block of 1s then taking $i = 2$ makes the second or third block of 1s too long. □

3.2. Connection with Lyndon words and pre-necklaces

In this section we explore the relationship between the language $\mathcal{L}_{\text{PN0}}$ of prefix normal words w.r.t. 0 and some known classes of words defined by means of lexicographic properties. Note that in this section, when referring to prefix normality, we mean with respect to 0. We assume the usual order $0 < 1$ on the alphabet.

A *Lyndon word* is a word which is lexicographically strictly smaller than any of its proper non-empty suffixes. Equivalently, w is a Lyndon word if it is the strictly smallest, in the lexicographic order, among its conjugates, i.e., for any factorization $w = uv$, with u, v non-empty words, one has that the word vu is lexicographically greater than w [28]. A word w is a *power* if it can be obtained by concatenating two or more copies of another word, i.e. if there exists a non-empty v and a $k > 1$ such that $w = v^k$. A word that is not a power is called *primitive.* Note that, by definition, a Lyndon word is primitive. Let us denote by *Lyn* the set of Lyndon words over Σ. One has that $Lyn \not\subseteq \mathcal{L}_{\mathrm{PN0}}$ and $\mathcal{L}_{\mathrm{PN0}} \not\subseteq Lyn$. For example, the word $w = 0101$ belongs to $\mathcal{L}_{\mathrm{PN0}}$ but is not a Lyndon word since it is not primitive. An example of a Lyndon word which is not in normal form is $w = 00110100111$.

A necklace is a Lyndon word or a power of a Lyndon word. A *pre-necklace* is a prefix of a necklace [34] (also called *preprime word* [25], or *sesquipower* or *fractional power* of a Lyndon word [12]). Let us denote by *PL* the language of pre-necklaces. The next proposition shows that every prefix normal word different from a power of the letter 1 is a prefix of a Lyndon word.

Proposition 12. *Let $w \in \mathcal{L}_{PN0}$ with $|w|_0 > 0$. Then the word $w1^{|w|}$ is a Lyndon word.*

Proof. We have to prove that every rotation of $w' = w1^{|w|}$ is strictly greater than w'. If the rotation starts at a position within the second half of w', then this is clearly true, since then its first character is 1, while w' starts with a 0, w being a prefix normal word containing at least one 0. So let v be a suffix of w' of length at least $|w| + 1$, and let u be the longest common prefix of v and w'. If $u = v$, then v is a border (both a prefix and suffix) of w', of length more than half its length, and thus w' has a period of length $i = |w'| - |v| < |w|$, i.e., every character is the same as the one which follows i positions later. Since the second half of w' consists of 1s only, this implies that so does the first half, contrary to our assumption. So v is not a prefix of w', and therefore u is followed by two different characters in v and in w'. Let us write $v = v'1^{|w|}$. If $|u| \geq |v'|$, then $u1$ is a prefix of v, implying that $u0$ is a prefix of w', and thus w' is smaller than v. If $|u| < |v'|$, assume that $u0$ is a prefix of v and $u1$ of w'. Then w has a substring ($u0$) which has more 0s than the prefix of the same length ($u1$), a contradiction to w being prefix normal. Therefore, again we have that w' is smaller than v. $\square$

We can now state the following result:

Theorem 13. *Every prefix normal word is a pre-necklace.*

Proof. If w is of the form 1^n, $n \geq 1$, then w is a power of the Lyndon word 1, hence it is a pre-necklace. Otherwise, w contains at least one 0, thus by by Proposition 12, it is the prefix of a Lyndon word. $\square$

The languages $\mathcal{L}_{\mathrm{PN0}}$ and *PL*, however, do not coincide. A shortest word in *PL* that does not belong to $\mathcal{L}_{\mathrm{PN0}}$ is $w = 00110100$. Below we give the table of the number of words in $\mathcal{L}_{\mathrm{PN0}}$ of each length $n \leq 16$, compared with that of pre-necklaces. Both sequences are listed in the On-Line Encyclopedia of Integer Sequences [35] (sequences A062692 and A194850), where the reader can find further terms.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\mathcal{L}_{\mathrm{PN0}} \cap \Sigma^n$	2	3	5	8	14	23	41	70	125	218	395	697	1273	2279	4185	7568
$PL \cap \Sigma^n$	2	3	5	8	14	23	41	71	127	226	412	747	1377	2538	4720	8800

Table 5: The number of words in $\mathcal{L}_{\mathrm{PN0}}$ and in PL for each length up to 16.

4. Enumeration results about prefix normal words

Let $pnw(n)$ denote the number of prefix normal words of length n. It is an easy consequence both of Lemma 8 and of Proposition 9 that $pnw(n)$ grows exponentially. To see this, note that the conditions of Lemma 8 are always satisfied if $r_1 \leq r_2 \leq \ldots \leq r_k$, and thus the number of partitions of n is a lower bound for $pnw(n)$. On the other hand, Proposition 9 states that for all w, $1^{|w|}w$ is prefix normal, so $pnw(2n) \geq 2^n$.

In Table 5, we give $pnw(n)$ for n up to 16, the sequence for n up to 50 can be found in the On-Line Encyclopedia of Integer Sequences [35], sequence A194850. In Fig. 2 we show the growth ratio for small values of n. Two interesting phenomena can be observed: the values seem to approach 2 slowly, i.e., the number of prefix normal words almost doubles as we increase the length by 1. Second, the values show on oscillation pattern between even and odd values.

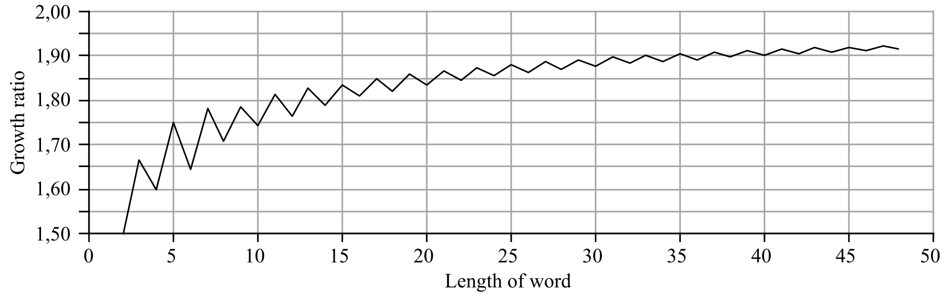

Figure 2: The value of $pnw(n)/pnw(n-1)$ for prefix normal words w of length n, for $n \leq 50$ (loglinear scale).

4.1. Asymptotic bounds on the number of prefix normal words

We give lower and upper bounds on the number $pnw(n)$ of prefix normal words of length n.

Theorem 14. *For n sufficiently large*

$$pnw(n) \geq 2^{n-4\sqrt{n\log n}}. \tag{1}$$

Proof. Let $k = k(n)$ be a positive integer to be fixed later. First we only consider words whose length, n, is a multiple of $2k$, whose first $4k$ letters are 1s, and in each of the following blocks of length $2k$, there are exactly k 1s and k 0s. The number of such words is $\binom{2k}{k}^{(n-4k)/2k}$ and by construction, they are all prefix normal.

We use the inequality $\binom{2k}{k} \geq 2^{2k}/(2\sqrt{k})$ and substitute $k = \sqrt{n\log n}$ in the third step.

$$\begin{aligned}
\binom{2k}{k}^{(n-4k)/2k} &\geq \left(\frac{2^{2k}}{2\sqrt{k}}\right)^{n/(2k)-2} \\
&= \frac{2^n}{(2\sqrt{k})^{n/(2k)}}\frac{4k}{2^{4k}} \\
&= \frac{2^n}{2^{4\sqrt{n\log n}}}(2\sqrt{k})^{1-n/(2k)} \\
&\geq \frac{2^n}{2^{4\sqrt{n\log n}}} \text{ for sufficiently large } n.
\end{aligned}$$

The last inequality follows from the fact that $\lim_{n\to\infty}(2\sqrt{k})^{1-n/(2k)} = 0$ if $k = \sqrt{n\log n}$. □

Next we show how to obtain an upper bound on $pnw(n)$, considering the length of the first 1-run.

Theorem 15. *For n sufficiently large, we have $pnw(n) \leq 2^{n-\lg n+1}$.*

Proof. This will follow from enumeration results about pre-necklaces since every 0-prefix normal word is a pre-necklace. Let $PL(n)$ be the number of pre-necklaces of length n. In [34] it is shown (top of page 424) that

$$PL(n) \leq \sum_{i=1}^{n}\frac{2^i}{i} + \sum_{i=1}^{n}\sqrt{2^i}.$$

They also show that (Lemma 5 of [34])

$$\lim_{n\to\infty} \frac{n}{2^n} \sum_{i=1}^{n} \frac{2^i}{i} = 2.$$

Thus, for large enough n, and fixed $\varepsilon > 0$,

$$PL(n) \leq (1+\varepsilon) \sum_{i=1}^{n} \frac{2^i}{i} \leq (1+2\varepsilon)2^n/n \leq 2^{n-\lg n+1}.$$

□

4.2. Exact formulas for words with fixed density.

For a binary word w, its *density* is defined as the number of 1s in w, i.e. as $|w|_1$. If we count the number of prefix normal words of length n with a given fixed number of 1s, we get exact results in a few cases. Let us denote by $pnw(n,d)$ the cardinality of the set $\{w \in \mathcal{L}_{\text{PN1}} \cap \Sigma^n \mid |w|_1 = d\}$.

Proposition 16. *For $d = 0, 1, \ldots, 6$, we have the generating functions $f_d(x) = \sum_{n=0}^{\infty} pnw(n,d)x^n$:*

$$\begin{aligned} f_0(x) &= \frac{1}{1-x} \\ f_1(x) &= \frac{x}{1-x} \\ f_2(x) &= \frac{x^2}{(1-x)^2} \\ f_3(x) &= \frac{x^3}{(1-x^2)(1-x)^2} \end{aligned}$$

$$\begin{aligned} f_4(x) &= \frac{x^4}{(1-x^3)(1-x)^3} \\ f_5(x) &= \frac{x^5(1+x+x^2)}{(1-x^4)(1-x^2)^2(1-x)^2} \\ f_6(x) &= \frac{x^6(1+x+x^2+x^3)}{(1-x^5)(1-x^3)(1-x^2)(1-x)^3} \end{aligned}$$

PROOF. For $d \leq 3$, one easily checks $pnw(n,0) = pnw(n,1) = 1$, $pnw(n,2) = n-1$ and $pnw(n,3) = \lfloor (n+1)^2/4 \rfloor$, giving the desired functions.

For $d = 4$, we calculate the number of positive solutions r_1, r_2, r_3, r_4 to the inequalities in Lemma 8. Let $q_1 = r_1 - 1$, $q_4 = r_4 - 1$, $d_2 = r_2 - r_1$ and $d_3 = r_3 - r_1$. We are counting the nonnegative solutions of

$$3q_1 + d_2 + d_3 + q_4 + 4 = n,$$

which give generating function $f_4(x)$ by equating the coefficients of x^n in the expansion of the following product:

$$\begin{aligned} &(1 + x^3 + x^6 + \cdots)(1 + x + x^2 + \cdots)^3 \cdot x^4 && (2) \\ &= \frac{x^4}{(1 - x^3)(1 - x)^3}. && (3) \end{aligned}$$

More complicated but manageable case analysis leads to the results for $d = 5$ and 6. □

Similar formulas can be derived for $pnw(n, n - d)$ for small values of d. Unfortunately, no clear pattern is visible for $f_d(x)$ that we could use for calculating $pnw(n)$.

The inequalities in Lemma 8 define linear diophantine equations. The general theory for enumerating solutions of such equations [36] guarantees that there is a closed rational function form for the generating functions with the observed denominators, in [37] there are algorithms for calculating these functions (which, however are not efficient enough to get results for much larger values of d). Above, we only discussed the first few simple cases. We did not succeed in extending our list of concrete formulas for the rational functions f_d for $d > 6$ using automated computation.

4.3. Exact formulas for words with a fixed prefix.

We now fix a prefix w and give enumeration results on prefix normal words with prefix w. Our first result indicates that we have to consider each w separately.

Definition 5. If w is a binary word, let $\mathcal{L}_{\text{ext}}(w) = \{w' : ww' \text{ is prefix normal}\}$, and $\mathcal{L}_{\text{ext}}(w, m) = \mathcal{L}_{\text{ext}}(w) \cap \Sigma^m$. Let $ext(w, m, d) = |\{w' : ww' \text{ is prefix normal of length } |w| + m \text{ and density } d\}|$, and $ext(w, m) = |\mathcal{L}_{\text{ext}}(w, m)|$.

Lemma 17. *Let $v, w \in 1\{0, 1\}^*$ be both prefix normal. If $v \neq w$ then $\mathcal{L}_{ext}(v) \neq \mathcal{L}_{ext}(w)$.*

PROOF. We may assume $|v| \leq |w|$.
First case. v is not a prefix of w. Let i denote the first position where they differ. If $v_i = 1$ and $w_i = 0$, then for $u = 0^{|w|}v$ we have that vu is prefix normal while wu is not. If $v_i = 0$ and $w_i = 1$, then let $u = 0^{|w|}w$. We have that vu is not prefix normal but wu is.
Second case. v is a prefix of w. If w has a 1 in any position after $|v|$, then we can proceed as in the first case. The remaining case is when $w = v0^m$ for some $m > 0$. If vv is prefix normal, then so must be vvv, but $v0^m vv$ cannot be. Otherwise, let $k \geq 1$ be the smallest integer (which is sure to exist) such that $v0^k v$ is prefix normal. Then $v0^{k-1}v$ is not prefix normal while $w0^{k-1}v$ is. This completes the proof. □

We were unable to prove that the growth of these two extension languages also differ.

Conjecture 18. *Let $v, w \in 1\{0,1\}^*$ be both prefix normal. If $v \neq w$ then the infinite sequences $(ext(v,m))_{m\geq 1}$ and $(ext(w,m))_{m\geq 1}$ are different.*

The values $ext(w,m,d)$ seem hard to analyze. We give exact formulas for a few special cases of interest. Using Lemma 8, it is possible to give formulas similar to those in Proposition 16 for $ext(w,m,d)$ for fixed w and d. We only mention one such result.

Lemma 19. *For $1 \leq d \leq n$ we have $ext(10, n+d-3, d) = pnw(n,d)$.*

PROOF. Consider the following map: let w be an arbitrary word of length n and density $d > 1$, starting with 1. Except for the starting 1, insert a 0 right before each subsequent occurrence of 1. This gives a word w' of length $n + d - 1$, starting with 10 that does not contain the factor 11. Clearly, the map is injective and all words of length $n + d - 1$ starting with 10 and containing no factor 11 are obtained this way. In order to prove the lemma, we only need to show that prefix normality is preserved by the map and its inverse. For this, observe that there exists a prefix (resp. factor) of w of length k containing r 1s if and only if there exists a prefix (resp. factor) of w' of length $k + r - 1$ containing r 1s. □

The following lemma lists exact values for $ext(w, |w|)$ for some infinite families of words w. Here $F(n)$ denotes the nth Fibonacci number, i.e. $F(1) = F(2) = 1$ and $F(n+2) = F(n+1) + F(n)$.

Lemma 20. *For all values of n where the exponents are nonnegative, we have the following formulas:*

$$\begin{aligned}
ext(0^n, n) &= 1\\
ext(1^n, n) &= 2^n\\
ext(1^{n-1}0, n) &= 2^n - 1\\
ext(1^{n-2}01, n) &= 2^n - 5\\
ext(1^{n-2}00, n) &= 2^n - (n+1)
\end{aligned}$$

$$\begin{aligned}
&ext((10)^{\frac{n}{2}}, n) = F(n+2) \textit{ if } n \textit{ is even}\\
&ext((10)^{\frac{n-1}{2}}1, n) = F(n+1) \textit{ if } n \textit{ is odd}\\
&ext(10^{n-2}1, n) = 3\\
&ext(10^{n-1}, n) = n+1
\end{aligned}$$

Proof. For $w = 1^n$, $w = 1^{n-1}0$, $w = 1^{n-2}01$ and $w = 1^{n-2}00$, it is easy to count those extensions that fail to give prefix normal words: None for $w = 1^n$; only one for $w = 1^{n-1}0$, namely $1^{n-1}01^n$; for $w = 1^{n-2}01$, those extensions which contain a 1-run of length $n-1$, namely 1^{n-2} followed by any two characters, or 01^{n-1}; and for $w = 1^{n-2}00$, those that contain at least $n-1$ many 1s in the second half, i.e. with second half $1^n, 1^{n-1}0, 1^{n-2}01, \ldots, 01^{n-1}$.

Similarly, for $w = 10^{n-2}1$, $w = 10^{n-1}$ and $w = 0^n$, counting the extensions that yield prefix normal words gives the result in a straightforward way.

Let n be even. For $w = (10)^{\frac{n}{2}}$, note that ww' is prefix normal if and only if w' avoids 11. The number of such words is known to equal $F(n+2)$. For n odd, the argument is similar, with the prefix of interest, $w1$, being of length $n+1$, hence the previous Fibonacci number. □

4.4. *Some experimental results about enumeration of prefix normal words*

We consider extensions of prefix normal words by a single symbol to the right. It turns out that this question has implications for the enumeration of prefix normal words.

Definition 6 (Extension-critical words). We call a prefix normal word w *extension-critical* if $w1$ is not prefix normal. Let $ecrit(n)$ denote the number of extension-critical words in $\mathcal{L}_{\mathrm{PN1}} \cap \Sigma^n$.

The lemma below applies to any family of words B for which $\varepsilon \in B$ and such that $x \in B$ implies $x0 \in B$.

Lemma 21. *For $n \geq 1$ we have*

$$pnw(n) = 2pnw(n-1) - ecrit(n-1) = pnw(n-1)\left(2 - \frac{ecrit(n-1)}{pnw(n-1)}\right). \quad (4)$$

From this it follows that

$$pnw(n) = 2\prod_{i=1}^{n-1}\left(2 - \frac{ecrit(i)}{pnw(i)}\right). \quad (5)$$

Proof. The number of prefix normal words of length n ending in 0 is $pnw(n-1)$, that of prefix normal words of length n ending in 1 is $pnw(n-1) - ecrit(n-1)$, hence we have (4). The product form follows if we use $pnw(n) = pnw(1)\prod_{i=1}^{n-1}\frac{pnw(i+1)}{pnw(i)}$. □

Lemma 22. *For n going to infinity,* $\liminf ecrit(n)/pnw(n) = 0$.

Proof. Assume that there exist an integer N_0 and a real number $\varepsilon > 0$ such that for $n \geq N_0$ we have $ecrit(n)/pnw(n) > \varepsilon$. Then by (5) we would have $pnw(n) = O((2-\varepsilon)^n)$, contradicting Theorem 14. □

We conjecture that in fact the ratio of extension-critical words converges to 0. We study the behavior of $ecrit(n)/pnw(n)$ for $n \leq 49$. The left plot in Fig. 3 shows the ratio of extension-critical words for $n \leq 49$. These data support the conjecture that the ratio tends to 0. Interestingly, the values decrease monotonically for both odd and even values, but we have $ecrit(n+1)/pnw(n+1) > ecrit(n)/pnw(n)$ for even n. We were unable to find an explanation for this.

The right plot in Fig. 3 shows the ratio of extension-critical words multiplied by $n/\log n$. Apart from a few initial data points, the values for even n increase monotonically and the values for odd n decrease monotonically, and the values for odd n stay above those for even n.

Conjecture 23. *Based on empirical evidence, we conjecture the following:*

$$\begin{aligned} ecrit(n) &= pnw(n)\Theta(\log n/n), & (6)\\ pnw(n) &= 2^{n-\Theta((\log n)^2)}. & (7) \end{aligned}$$

Note that the second estimate follows from the first one by (5).

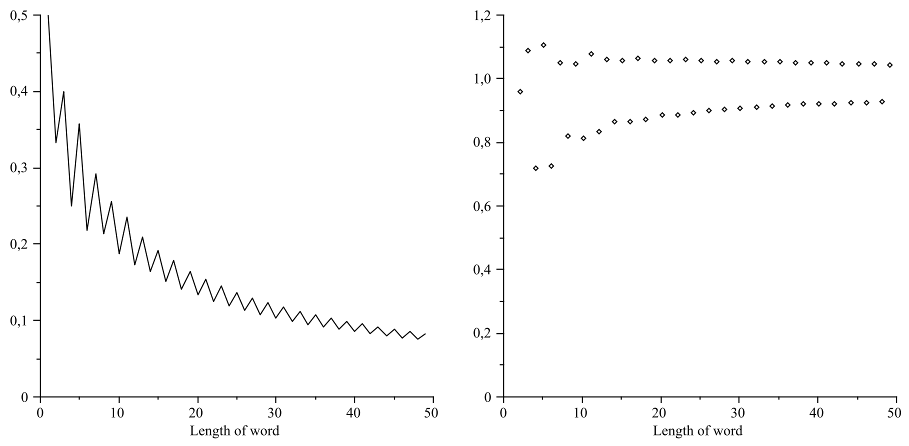

Figure 3: The ratio $\frac{ecrit(n)}{pnw(n)}$ (left), and the value $\frac{ecrit(n)}{pnw(n)} \cdot \frac{n}{\ln n}$ (right).

5. Conclusion and open problems

We introduced two new normal forms of binary words, the prefix normal forms with respect to 1 and 0, and showed how they arise naturally in the investigation of Parikh sets of binary words and jumbled pattern matching. We introduced prefix normal words (w.r.t. 1 or 0), words which equal their own normal form, and discussed several properties of these words. We showed results about the language of prefix normal words, among these that 0-prefix normal are strictly contained in the language of pre-necklaces. We also discussed extensively the growth behaviour of the number of fixed-length prefix normal words.

Many open problems remain. It would be nice to have exact, or at least more precise asymptotic formulas for the enumeration of prefix normal words. Related to the enumeration, the strange oscillating behavior in Figures 2 and 3 between odd and even values calls for an explanation.

Another question is testing binary words for prefix normality. Currently, no faster method is known (in worst-case running time), then calculating the normal form.

It would be an interesting direction to explore the connection between the normal forms w.r.t. 1 and 0, for example how many different values can $\mathrm{PNF}_0(w)$ take (and what can we say about them) if we fix $\mathrm{PNF}_1(w)$.

Finally, prefix normality could also be defined over non-binary alphabets. In this case however, we do not obtain an index directly applicable to jumbled pattern matching. Combinatorial or formal language theoretic investigation and enumeration of prefix normal words for general alphabets is subject of future work.

Acknowledgements

Gabriele Fici was partially supported by the PRIN 2010/2011 project "Automi e Linguaggi Formali: Aspetti Matematici e Applicativi" of the Italian Ministry of Education (MIUR). The research of Joe Sawada and Frank Ruskey was partially funded by grants from the National Engineering Research Council of Canada. We thank an anonymous referee for a very careful reading and helpful suggestions.

References

[1] J. Acharya, H. Das, O. Milenkovic, A. Orlitsky, and S. Pan. Reconstructing a string from its substring compositions. In *Proc. of IEEE International Symposium on Information Theory (ISIT 2010)*, pages 1238–1242, 2010.

[2] G. Badkobeh, G. Fici, S. Kroon, and Zs. Lipták. Binary jumbled string matching for highly run-length compressible texts. *Inf. Process. Lett.*, 113(17):604–608, 2013.

[3] G. Benson. Composition alignment. In *Proc. of the 3rd International Workshop on Algorithms in Bioinformatics (WABI 2003)*, pages 447–461, 2003.

[4] S. Böcker. Simulating multiplexed SNP discovery rates using base-specific cleavage and mass spectrometry. *Bioinformatics*, 23(2):5–12, 2007.

[5] S. Böcker, K. Jahn, J. Mixtacki, and J. Stoye. Computation of median gene clusters. In *Proc. of the Twelfth Annual International Conference on Computational Molecular Biology (RECOMB 2008)*, pages 331–345, 2008. LNBI 4955.

[6] P. Burcsi, F. Cicalese, G. Fici, and Zs. Lipták. On Table Arrangements, Scrabble Freaks, and Jumbled Pattern Matching. In *Proc. of the 5th International Conference on Fun with Algorithms (FUN 2010)*, volume 6099 of *LNCS*, pages 89–101, 2010.

[7] P. Burcsi, F. Cicalese, G. Fici, and Zs. Lipták. Algorithms for jumbled pattern matching in strings. *Int. J. Found. Comput. Sci.*, 23(2):357–374, 2012.

[8] P. Burcsi, F. Cicalese, G. Fici, and Zs. Lipták. On approximate jumbled pattern matching in strings. *Theory Comput. Syst.*, 50(1):35–51, 2012.

[9] P. Burcsi, G. Fici, Zs. Lipták, F. Ruskey, and J. Sawada. Normal, abby normal, prefix normal. In *Proc. of the 7th International Conference on Fun with Algorithms (FUN 2014)*, volume 8496 of *LNCS*, pages 74–88, 2014.

[10] P. Burcsi, G. Fici, Zs. Lipták, F. Ruskey, and J. Sawada. On combinatorial generation of prefix normal words. In *Proc. 25th Ann. Symp. on Comb. Pattern Matching (CPM 2014)*, volume 8486 of *LNCS*, pages 60–69, 2014.

[11] A. Butman, R. Eres, and G. M. Landau. Scaled and permuted string matching. *Inf. Process. Lett.*, 92(6):293–297, 2004.

[12] J. Champarnaud, G. Hansel, and D. Perrin. Unavoidable sets of constant length. *Internat. J. Algebra Comput.*, 14:241–251, 2004.

[13] T. M. Chan and M. Lewenstein. Clustered integer 3SUM via additive combinatorics. In *Proc. of the 47th Annual ACM on Symposium on Theory of Computing (STOC 2015)*, pages 31–40, 2015.

[14] F. Cicalese, G. Fici, and Zs. Lipták. Searching for jumbled patterns in strings. In *Proc. of the Prague Stringology Conference (PSC 2009)*, pages 105–117. Czech Technical University in Prague, 2009.

[15] F. Cicalese, T. Gagie, E. Giaquinta, E. S. Laber, Zs. Lipták, R. Rizzi, and A. I. Tomescu. Indexes for jumbled pattern matching in strings, trees and graphs. In *Proc. of the 20th String Processing and Information Retrieval Symposium (SPIRE 2013)*, volume 8214 of *LNCS*, pages 56–63, 2013.

[16] F. Cicalese, E. S. Laber, O. Weimann, and R. Yuster. Near linear time construction of an approximate index for all maximum consecutive sub-sums of a sequence. In *Proc. 23rd Annual Symposium on Combinatorial Pattern Matching (CPM 2012)*, volume 7354 of *LNCS*, pages 149–158, 2012.

[17] M. Cieliebak, T. Erlebach, Zs. Lipták, J. Stoye, and E. Welzl. Algorithmic complexity of protein identification: combinatorics of weighted strings. *Discrete Appl. Math.*, 137(1):27–46, 2004.

[18] D. Clark. *Compact PAT trees.* PhD thesis, University of Waterloo, Canada, 1996.

[19] K. Dührkop, M. Ludwig, M. Meusel, and S. Böcker. Faster mass decomposition. In *Proc. of the 13th International Workshop on Algorithms in Bioinformatics, (WABI 2013)*, pages 45–58, 2013.

[20] R. Eres, G. M. Landau, and L. Parida. Permutation pattern discovery in biosequences. *Journal of Computational Biology*, 11(6):1050–1060, 2004.

[21] G. Fici and Zs. Lipták. On prefix normal words. In *Proc. of the 15th Intern. Conf. on Developments in Language Theory (DLT 2011)*, volume 6795 of *LNCS*, pages 228–238. Springer, 2011.

[22] T. Gagie, D. Hermelin, G. M. Landau, and O. Weimann. Binary jumbled pattern matching on trees and tree-like structures. In *Proc. of the 21st Annual European Symposium on Algorithm (ESA 2013)*, pages 517–528, 2013.

[23] E. Giaquinta and Sz. Grabowski. New algorithms for binary jumbled pattern matching. *Inf. Process. Lett.*, 113(14-16):538–542, 2013.

[24] D. Hermelin, G. M. Landau, Y. Rabinovich, and O. Weimann. Binary jumbled pattern matching via all-pairs shortest paths. *CoRR*, abs/1401.2065, 2014.

[25] D. E. Knuth. *Generating All Tuples and Permutations. The Art of Computer Programming, Vol. 4, Fascicle 2.* Addison-Wesley, 2005.

[26] T. Kociumaka, J. Radoszewski, and W. Rytter. Efficient indexes for jumbled pattern matching with constant-sized alphabet. In *Proc. of the 21st Annual European Symposium on Algorithm (ESA 2013)*, pages 625–636, 2013.

[27] L.-K. Lee, M. Lewenstein, and Q. Zhang. Parikh matching in the streaming model. In *Proc. of 19th International Symposium on String Processing and Information Retrieval, (SPIRE 2012)*, volume 7608 of *Lecture Notes in Computer Science*, pages 336–341. Springer, 2012.

[28] M. Lothaire. *Algebraic Combinatorics on Words.* Encyclopedia of Mathematics and its Applications. Cambridge Univ. Press, 2002.

[29] T. M. Moosa and M. S. Rahman. Indexing permutations for binary strings. *Inf. Process. Lett.*, 110:795–798, 2010.

[30] T. M. Moosa and M. S. Rahman. Sub-quadratic time and linear space data structures for permutation matching in binary strings. *J. Discrete Algorithms*, 10:5–9, 2012.

[31] J. I. Munro. Tables. In *Proc. of Foundations of Software Technology and Theoretical Computer Science (FSTTCS 1996)*, pages 37–42, 1996.

[32] G. Navarro and V. Mäkinen. Compressed full-text indexes. *ACM Comput. Surv.*, 39(1), 2007.

[33] L. Parida. Gapped permutation patterns for comparative genomics. In *Proc. of the 6th International Workshop on Algorithms in Bioinformatics, (WABI 2006)*, pages 376–387, 2006.

[34] F. Ruskey, C. Savage, and T. M. Y. Wang. Generating necklaces. *J. Algorithms*, 13(3):414 – 430, 1992.

[35] N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. Available electronically at `http://oeis.org`.

[36] R. P. Stanley. *Enumerative Combinatorics.* Wadsworth Publ. Co., Belmont, CA, USA, 1986.

[37] D. Zeilberger. Lindiophantus: A maple package that finds generating functions representating solutions of systems of linear diophantine equations. `http://http://www.math.rutgers.edu/~zeilberg/tokhniot/LinDiophantus`. Accessed: 2015-05-30.