Theoretical Computer Science 659 (2017) 1–13

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

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

ARTICLE INFO

Article history:
Received 10 March 2016
Received in revised form 15 October 2016
Accepted 17 October 2016
Available online 14 November 2016
Communicated by D. Perrin

Keywords:
Prefix normal words
Prefix normal forms
Binary languages
Binary jumbled pattern matching
Pre-necklaces
Lyndon words
Enumeration

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.

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*.

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

* Corresponding author.
E-mail addresses: bupe@compalg.inf.elte.hu (P. Burcsi), gabriele.fici@unipa.it (G. Fici), zsuzsanna.liptak@univr.it (Zs. Lipták), ruskey@cs.uvic.ca (F. Ruskey), jsawada@uoguelph.ca (J. Sawada).

http://dx.doi.org/10.1016/j.tcs.2016.10.015
0304-3975/

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 IBJPM 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 $Fact(w)$. 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 $\Pi(w) = \{p(v) \mid v \in Fact(w)\}$, 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 $pos_1(w, i)$ the position of the ith 1 in the word w, i.e. $P_1(w, i) = |w_1 \cdots w_i|_1$ and $pos_1(w, i) = \min\{k : |w_1 \cdots w_k|_1 = i\}$. The functions P_0 and pos_0 are defined similarly. Note that in the context of succint indexing, these functions are frequently called *rank* and *select*, cf. [32]: We have, for $x = 0, 1$, $P_x(w, i) = rank_x(w, i)$ and $pos_x(w, i) = select_x(w, i)$.

2.1. Prefix normal words

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

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

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 $w = 10100110111000111001011$. In Table 1, we give the values of F_1 and F_0 for w.