ARTICLE IN PRESS

Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

A note on easy and efficient computation of full abelian periods of a word[☆]

Gabriele Fici[a], Thierry Lecroq[b], Arnaud Lefebvre[b,*], Élise Prieur-Gaston[b], William F. Smyth[c,d]

[a] Dipartimento di Matematica e Informatica, Università di Palermo, Italy
[b] Normandie Université, Université de Rouen, Normastic FR CNRS 3638, LITIS EA 4108, 76821 Mont-Saint-Aignan Cedex, France
[c] Department of Computing and Software, McMaster University, Hamilton ON L8S 4K1, Canada
[d] Faculty of Engineering & Information Technology, Murdoch University, Murdoch WA 6150, Australia

ARTICLE INFO

Article history:
Received 1 October 2014
Received in revised form 10 July 2015
Accepted 25 September 2015
Available online xxxx

Keywords:
Abelian period
Abelian power
Weak repetition
Design of algorithms
Text algorithms
Combinatorics on words

ABSTRACT

Constantinescu and Ilie (2006) introduced the idea of an Abelian period with head and tail of a finite word. An Abelian period is called full if both the head and the tail are empty. We present a simple and easy-to-implement $O(n \log \log n)$-time algorithm for computing all the full Abelian periods of a word of length n over a constant-size alphabet. Experiments show that our algorithm significantly outperforms the $O(n)$ algorithm proposed by Kociumaka et al. (2013) for the same problem.

1. Introduction

The study of repetitions in words is a classical topic in Stringology. A word is called an (integer) power if it can be written as the concatenation of two or more copies of another word, like barbar. However, any word can be written as a *fractional* power; that is, given a word $\boldsymbol{w}$, one can always find a word $\boldsymbol{u}$ such that $\boldsymbol{w} = \boldsymbol{u}^n\boldsymbol{u}'$, where $\boldsymbol{u}'$ is a (possible empty) prefix of $\boldsymbol{u}$ and n is an integer greater than or equal to one. In this case, the length of $\boldsymbol{u}$ is called *a period* of the word $\boldsymbol{w}$. A word $\boldsymbol{w}$ can have different periods, the least of which is usually called *the period* of $\boldsymbol{w}$.

Recently, a natural extension of this setting has been considered involving the notion of commutative equivalence. Two words are called commutatively equivalent if they have the same number of occurrences of each letter; that is, if one is an anagram of the other. An Abelian power (also called a weak repetition [5]) is a word that can be written as the concatenation of two or more words that are commutatively equivalent, like iceddice.

Recall that the Parikh vector $\mathcal{P}_{\boldsymbol{w}}$ of a word $\boldsymbol{w}$ is the vector whose ith entry is the number of occurrences of the ith letter of the alphabet in $\boldsymbol{w}$. For example, given the (ordered) alphabet $\Sigma = \{a, b, c\}$, the Parikh vector of the word $\boldsymbol{w} = aaba$ is $\mathcal{P}_{\boldsymbol{w}} = (3, 1, 0)$. Two words are therefore commutatively equivalent if and only if they have the same Parikh vector.

☆ The results in this note have been presented in preliminary form in Fici et al. (2012).

* Corresponding author.
E-mail addresses: Gabriele.Fici@unipa.it (G. Fici), Thierry.Lecroq@univ-rouen.fr (T. Lecroq), Arnaud.Lefebvre@univ-rouen.fr (A. Lefebvre), Elise.Prieur@univ-rouen.fr (É. Prieur-Gaston), smyth@mcmaster.ca (W.F. Smyth).

http://dx.doi.org/10.1016/j.dam.2015.09.024
0166-218X/© 2015 Published by Elsevier B.V.

Constantinescu and Ilie [3] introduced the definition of an Abelian period with head and tail of a word $\boldsymbol{w}$ over a finite ordered alphabet $\Sigma = \{a_1, a_2, \ldots, a_\sigma\}$: An integer $p > 0$ is an Abelian period of $\boldsymbol{w}$ if one can write $\boldsymbol{w} = \boldsymbol{u}_0\boldsymbol{u}_1 \cdots \boldsymbol{u}_{k-1}\boldsymbol{u}_k$ where for $0 < i < k$ all the factors $\boldsymbol{u}_i$'s have the same Parikh vector $\mathcal{P}$ such that $\sum_{j=1}^{\sigma} \mathcal{P}[j] = p$ and the Parikh vectors of $\boldsymbol{u}_0$ and $\boldsymbol{u}_k$ are "contained" in $\mathcal{P}$, in the sense that they are proper sub-Parikh vectors of $\mathcal{P}$ (see next section for the formal definition of "contained"). In this case, $\boldsymbol{u}_0$ and $\boldsymbol{u}_k$ are called the head and the tail of the Abelian period p, respectively. This definition of an Abelian period matches that of an Abelian power when $\boldsymbol{u}_0$ and $\boldsymbol{u}_k$ are both empty and $k > 2$.

As an example, the word $\boldsymbol{w} = abaababa$ over the alphabet $\Sigma = \{a, b\}$ can be written as $\boldsymbol{w} = \boldsymbol{u}_0\boldsymbol{u}_1\boldsymbol{u}_2\boldsymbol{u}_3$, where $\boldsymbol{u}_0 = ab$, $\boldsymbol{u}_1 = aab$, $\boldsymbol{u}_2 = aba$, $\boldsymbol{u}_3 = \varepsilon$, with ε the empty word, so that 3 is an Abelian period of $\boldsymbol{w}$ with Parikh vector $(2, 1)$ (the Parikh vector of $\boldsymbol{u}_0$ is $(1, 1)$ and that of $\boldsymbol{u}_3$ is $(0, 0)$ which are both "contained" in $(2, 1)$). Notice that $\boldsymbol{w}$ has also Abelian period 2, since it can be written as $\boldsymbol{w} = \boldsymbol{u}_0\boldsymbol{u}_1\boldsymbol{u}_2\boldsymbol{u}_3\boldsymbol{u}_4$, with $\boldsymbol{u}_0 = a$, $\boldsymbol{u}_1 = ba$, $\boldsymbol{u}_2 = ab$, $\boldsymbol{u}_3 = ab$, $\boldsymbol{u}_4 = a$.

This example shows that a word can have different Abelian periods. Moreover, a word can have the same Abelian period p corresponding to different factorizations; that is, with different heads. Actually, a word of length n can have $\Theta(n^2)$ many different Abelian periods [7], if these are represented in the form (h, p), where h is the length of the head—the length of the tail is uniquely determined by h and p.

Recently [6,7] we described algorithms for computing all the Abelian periods of a word of length n in time $O(n^2 \times \sigma)$. This was improved to time $O(n^2)$ in [2]. In [4] the authors derived an efficient algorithm for computing the Abelian periods based on prior computation of the Abelian squares.

An Abelian period is called *full* if both the head and the tail are empty. Clearly, a full Abelian period is a divisor of the length of the word.

A preliminary version of the present paper appeared in [8] where we presented brute force algorithms to compute full Abelian periods and Abelian periods without head and with tail in $O(n^2)$ time and a quasi-linear time algorithm QLFAP for computing all the full Abelian periods of a word. In [10] Kociumaka et al. gave a linear time algorithm LFAP for the same problem. Here we first briefly outline LFAP, followed by a description of QLFAP. Then, extending the presentation in [8], we add an experimental section to demonstrate that our algorithm significantly outperforms LFAP in practice, both on pseudo-randomly generated and genomic data. Our method has the additional advantage of being conceptually simple and easy to implement.

2. Notation

Let $\Sigma = \{a_1, a_2, \ldots, a_\sigma\}$ be a finite ordered alphabet of cardinality σ and Σ^* the set of finite words over Σ. We let $|\boldsymbol{w}|$ denote the length of the word $\boldsymbol{w}$. Given a word $\boldsymbol{w} = \boldsymbol{w}[0..n-1]$ of length $n > 0$, we write $\boldsymbol{w}[i]$ for the $(i+1)$th symbol of $\boldsymbol{w}$ and, for $0 \leqslant i \leqslant j < n$, we write $\boldsymbol{w}[i..j]$ for the factor of $\boldsymbol{w}$ from the $(i+1)$th symbol to the $(j+1)$th symbol, both included. We let $|\boldsymbol{w}|_a$ denote the number of occurrences of the symbol $a \in \Sigma$ in the word $\boldsymbol{w}$.

The *Parikh vector* of $\boldsymbol{w}$, denoted by $\mathcal{P}_{\boldsymbol{w}}$, counts the occurrences of each letter of Σ in $\boldsymbol{w}$, that is, $\mathcal{P}_{\boldsymbol{w}} = (|\boldsymbol{w}|_{a_1}, \ldots, |\boldsymbol{w}|_{a_\sigma})$. Notice that two words have the same Parikh vector if and only if one word is a permutation of the other (in other words, an anagram).

Given the Parikh vector $\mathcal{P}_{\boldsymbol{w}}$ of a word $\boldsymbol{w}$, we let $\mathcal{P}_{\boldsymbol{w}}[i]$ denote its ith component and $|\mathcal{P}_{\boldsymbol{w}}|$ its norm, defined as the sum of its components. Thus, for $\boldsymbol{w} \in \Sigma^*$ and $1 \leqslant i \leqslant \sigma$, we have $\mathcal{P}_{\boldsymbol{w}}[i] = |\boldsymbol{w}|_{a_i}$ and $|\mathcal{P}_{\boldsymbol{w}}| = \sum_{i=1}^{\sigma} \mathcal{P}_{\boldsymbol{w}}[i] = |\boldsymbol{w}|$.

Finally, given two Parikh vectors $\mathcal{P}, \mathcal{Q}$, we write $\mathcal{P} \subset \mathcal{Q}$ if $\mathcal{P}[i] \leqslant \mathcal{Q}[i]$ for every $1 \leqslant i \leqslant \sigma$ and $|\mathcal{P}| < |\mathcal{Q}|$. This makes precise the notion of "contained" used in the Introduction.

Definition 1 (*Abelian Period [3]*)**.** A word $\boldsymbol{w}$ has an Abelian period (h, p) if $\boldsymbol{w} = \boldsymbol{u}_0\boldsymbol{u}_1 \cdots \boldsymbol{u}_{k-1}\boldsymbol{u}_k$ such that:

- $\mathcal{P}_{\boldsymbol{u}_0} \subset \mathcal{P}_{\boldsymbol{u}_1} = \cdots = \mathcal{P}_{\boldsymbol{u}_{k-1}} \supset \mathcal{P}_{\boldsymbol{u}_k}$,
- $|\boldsymbol{u}_0| = h$, $|\boldsymbol{u}_1| = p$.

We call $\boldsymbol{u}_0$ and $\boldsymbol{u}_k$ respectively the *head* and the *tail* of the Abelian period. Notice that the length $t = |\boldsymbol{u}_k|$ of the tail is uniquely determined by h, p and $|\boldsymbol{w}|$, namely $t = (|\boldsymbol{w}| - h) \bmod p$.

The following lemma gives a bound on the maximum number of Abelian periods of a word.

Lemma 1 ([6])**.** *The maximum number of different Abelian periods (h, p) for a word of length n over an alphabet of size σ is $\Theta(n^2)$.*

Proof. The word $(a_1a_2 \cdots a_\sigma)^{n/\sigma}$ has Abelian period (h, p) for any $p \equiv 0 \mod \sigma$ and every h such that $0 \leqslant h \leqslant \min(p-1, n-p)$. □

An Abelian period is called *full* if it has head and tail both empty. We are interested in computing all the full Abelian periods of a word. Notice that a full Abelian period of a word of length n is a divisor of n. In the remainder of this note, we will therefore write that a word $\boldsymbol{w}$ has an Abelian period p if and only if it has full Abelian period $(0, p)$.

3. Previous work

We now outline the linear algorithm LFAP given in [10].

Let w be a word of length n. Let $\mathcal{P}_{w_i} = \mathcal{P}_{w[0..i]}$. Two positions $i, j \in \{1, \ldots, n\}$ are called proportional, which is denoted by $i \sim j$, if $\mathcal{P}_{w_i}[k] = c \times \mathcal{P}_{w_j}[k]$ for each k, where c is a real number independent of k.

Please cite this article in press as: G. Fici, et al., A note on easy and efficient computation of full abelian periods of a word, Discrete Applied Mathematics (2015), http://dx.doi.org/10.1016/j.dam.2015.09.024