Discrete Applied Mathematics 179 (2014) 174–192

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

An algorithmic toolbox for periodic partial words☆

Florin Manea [a,*], Robert Mercaş [a,*], Cătălin Tiseanu [b]

[a] Christian-Albrechts-Universität zu Kiel, Institut für Informatik, Christian-Albrechts-Platz 4, D-24098 Kiel, Germany

[b] Faculty of Mathematics and Computer Science, University of Bucharest, Str. Academiei 14, RO-010014 Bucharest, Romania

ARTICLE INFO

Article history:
Received 15 January 2014
Received in revised form 19 June 2014
Accepted 20 July 2014
Available online 10 September 2014

Keywords:
Combinatorics on words
Periodicity
Partial words
Algorithms

ABSTRACT

This work presents efficient solutions to several basic algorithmic problems regarding periodicity of partial words. In the first part of the paper, we show that all periods of a partial word of length n are determined in $\mathcal{O}(n \log n)$ time. Moreover, we define algorithms and data structures that help us answer in constant time queries regarding the periodicity of a word's factors. For these we need an $\mathcal{O}(n^2)$ preprocessing time and an $\mathcal{O}(n)$ updating time, whenever the words are extended by adding a letter. In the second part of the paper, we show that identifying a way to construct a periodic partial word by substituting the letters on some positions of a full word w with holes, where the distance between two consecutive holes must be greater than a given number, can be done in optimal time $\mathcal{O}(|w|)$. We also show that identifying a substitution which replaces the minimum number of positions by holes can be done as fast as in the previous case.

1. Introduction

Periodicity is one of the most fundamental properties of words. Problems correlated to periodicity computation have applications in formal languages and automata theory, algorithmic combinatorics on words, data compression, string searching and pattern matching algorithms (see [2,7,20] and the references therein). The first idea of a fast algorithm identifying all periods of a word was given in [15]. Almost a decade later, in [11], Crochemore provides the first correct time–space optimal algorithm taking into account also the fact that the alphabet is ordered. This solution, as many other subsequent efficient solutions of the problem, relies heavily on the possibility of solving in linear time and space the string matching problem: that is, finding all the occurrences of a shorter word (called pattern) in a larger one (called text) in linear time and space. Solutions that fulfil these efficiency requirements were given, for instance, in [12,17].

For partial words, sequences that beside regular symbols contain some "holes" or "don't care symbols" that can be substituted with any letter of the word's alphabet, the concept of periodicity was also deeply analysed ([2] surveys most of the work in this area and discusses the obtained results in comparison with the ones existing for words). To start with, the problem of testing the primitivity of a partial word (i.e., checking whether a given partial word has no period that divides its length) was discussed in [3,4], where partial solutions were proposed; similarly to the classical case, these solutions were based on (partial) words matching algorithms. To this end, we recall that fast partial words matching algorithms were

☆ This work represents an extended version of a paper presented at the *36th International Symposium on Mathematical Foundations of Computer Science, MFCS 2011*, Manea et al. (2011).

* Corresponding authors. Tel.: +49 431 8807547; fax: +49 431 8805745.

E-mail addresses: flm@informatik.uni-kiel.de (F. Manea), robynete@gmail.com, rgm@informatik.uni-kiel.de, robertmercas@gmail.com (R. Mercaş), ctiseanu@gmail.com (C. Tiseanu).

http://dx.doi.org/10.1016/j.dam.2014.07.017
0166-218X/

provided in [9], starting from the ideas initiated in [14]. The fastest deterministic partial matching algorithms, known so far, solve this problem in $\mathcal{O}(n \log n)$ time (see also [8]).

The study of repetitions in partial words was developed in [21]. A string is said to contain a repetition if it has consecutive factors compatible with the same full word. In [21], it is proved that over a binary alphabet there exist infinite partial words that are cube-free, which, in other words, means that for all factors the periods are greater than one third of their length. In [6] the authors solve a conjecture regarding the minimum size alphabet needed in order to construct an infinite partial word that remains overlap-free even after arbitrarily many insertions of holes. That is to say, here, all factors of the infinite word have periods greater than half their length. The authors use, in order to prove this, an $\mathcal{O}(nd)$ algorithm that determines if, after hole insertion such that between each two holes there are at least d non-hole symbols, a word has a certain period.

Algorithms regarding freeness of factors of partial words, factors free of some property, were first discussed in [21,13]. In [13] the authors construct some data structures which enable them to answer, after a preprocessing phase done in $\mathcal{O}(n^2)$, queries regarding the freeness of their factors in constant time. Moreover, the authors provide a method for updating the data structures in $\mathcal{O}(n \log n)$ time, whenever a symbol is concatenated to the right end of the existing string, and still being able to answer the queries in constant time.

This paper proposes a series of algorithms, for some of the most basic problems related to periodicity in partial words, more efficient than the already existing ones, and discusses possible generalisations of these problems. After presenting some basic concepts regarding partial words and periodicity, in the next section, our paper continues with two main parts.

First, in Section 3 results from [3,4,13] are extended and improved. The main result of that section is an algorithm that computes all periods of a partial word of length n in $\mathcal{O}(n \log n)$ time. Further, motivated by natural questions like finding efficiently all the factors of a word that have a given period, we improve on the results of [13] dealing with algorithms and data structures that help us answer in constant time queries regarding the periodicity of the factors of a word. Whenever the words are extended by the simple operation of adding one symbol (that is, a regular letter or a hole) at the right end of the word, we can update the data structures we constructed in linear time such that the query-answering time remains unchanged. Note that, the idea of updating a word by adding a letter at its end while managing information regarding its combinatorial properties (initiated in [13] and developed in [22]) opens the discussions on streaming and on-line algorithms testing combinatorial properties of partial words (and their factors).

Second, in Section 4, we give optimal algorithms that identify ways of making a word periodic by substituting the letters on some positions of the input word by holes in a restricted manner and improve the already mentioned results from [6]. While the results of Section 3 regard some natural algorithmic questions on the periodicity of words, note that the results presented in Section 4 may become useful in the area of combinatorics on words. For instance, we might be interested in constructing words in which we can randomly substitute letters with holes, such that no two holes are too close, while they remain nonperiodic [21,6]. Theorem 4, from the Section 4, enables us to test efficiently whether a given word satisfies this property or not. This strategy is not new, as in [18,19] the idea of producing from given full words, by substitution of some positions of the words with holes, partial words that satisfy some combinatorial properties is discussed, and possible connections with bioinformatics are established. Furthermore, Theorem 4 seems important as it shows the existence of an optimal algorithm used in a meaningful context (to prove [6, Conjecture 1]).

2. Basic definitions

We continue with several basic definitions.

Let V be a non-empty finite set of symbols called an *alphabet*, and denote its cardinality by $|V|$. Each element $a \in V$ is called a *letter*. A *full word* over V is a finite sequence of letters from V. A *partial word* over V is a finite sequence of letters from $V_\diamond = V \cup \{\diamond\}$, the alphabet V extended with the hole symbol $\diamond$. A full word is a partial word that does not contain the $\diamond$ symbol. The set containing all finite full words over the alphabet V is denoted by V^*, while the set of all finite partial words over the alphabet V is denoted by $V_\diamond^*$.

The *length* of a partial word u is denoted by $|u|$ and represents the total number of symbols in u. The *empty word*, denoted by ε, is the sequence of length zero. Thus, alternatively, a length n partial word $u \in V_\diamond$ can be viewed as a function $u : \{1, \ldots, n\} \to V_\diamond$ or as a partial function $u : \{1, \ldots, n\} \overset{\circ}{\to} V$.

A partial word u is a *factor* of a partial word v if $v = xuy$ for some x, y. We say that u is a *prefix* of v if $x = \varepsilon$ and a *suffix* of v if $y = \varepsilon$. We denote by $u[i]$ the symbol at position i in u and by $u[i..j]$ the factor of u starting at position i and ending at position j, consisting of the concatenation of the symbols $u[i], \ldots, u[j]$, where $1 \le i \le j \le n$.

If u and v are partial words of equal length, then u is *contained* in v, denoted by $u \subset v$, if $u[i] = v[i]$, for all $u[i] \in A$. Moreover, partial words u and v are *compatible*, denoted by $u \uparrow v$, if exists w such that $u \subset w$ and $v \subset w$.

The powers of a partial word u are defined recursively by $u^0 = \varepsilon$ and for $n \ge 1$, $u^n = uu^{n-1}$. A *period* of a partial word u over V is a positive integer p such that $u[i] = u[j]$ whenever $u[i], u[j] \in V$ and $i \equiv j \pmod p$. In such a case, we say u is *p-periodic*. If $p < |u|$, then u is periodic. A partial word u is said to be a k-repetition if it has a period p such that $p = \frac{|u|}{k}$; if the partial word u is not a k-repetition for any $k > 1$ we say that u is primitive.

As an example for the above, we see that the length 6 partial word $w = ab\diamond ba\diamond$ is 2-periodic and, thus it is a 3-repetition, since $a\diamond, \diamond b \subset ab$, and, therefore, $ab \uparrow \diamond b$, $ab \uparrow a\diamond$, and $\diamond b \uparrow a\diamond$.

A partial word u is said to be d-valid for some positive integer d if $u[i..i + d - 1]$ contains at most one $\diamond$-symbol, for all i with $1 \le i \le |u| - d + 1$. For the above example, we see that w is d-valid for any $d \in \{1, 2, 3\}$.