Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Two fast constructions of compact representations of binary words with given set of periods

Wojciech Rytter [1]

Department of Mathematics, Computer Science and Mechanics, Warsaw University, Warsaw, Poland

ARTICLE INFO

Article history:
Received 7 August 2015
Received in revised form 12 April 2016
Accepted 25 April 2016
Available online xxxx

Keywords:
Periodicity
Collage system
Algorithm
Borders

ABSTRACT

Assume we are given a sorted set $\mathcal{P}$ of size n of all periods of an unknown string of size N. Our main result is an algorithm generating in $O(n)$ time and $O(1)$ space a compressed representation of size $O(n)$ of the lexicographically-first binary string having $\mathcal{P}$ as the set of its periods. The input is read-only and the output is write-only. We also present a very simple preliminary algorithm generating some binary string (not necessarily lexicographically-first one) with the same set of periods. The *explicit* size N of the produced string can be exponential with respect to n. We assume that a given set of periods is valid: there *exists* some unknown string realizing this set.

1. Introduction

Periodicity plays central role in combinatorics and algorithms of words. A period of a string w is an integer $0 < p \leq |w|$ such that $w[i+p] = w[i]$, whenever both sides are defined. Periodicities are extensively used as crucial combinatorial tools in construction of many efficient algorithms on strings, see [3] and [5].

One of natural questions about periodicities is how to construct strings with a given periodicity structure and small alphabet. This helps to understand better the periodicities in string, see [2,6]. Many important properties of the family of all sets of periods for strings of a given length are investigated in [2].

We say that a word w realizes a given set $\mathcal{P}$ of periods iff $\mathcal{P}$ is the set of all periods of w. A set $\mathcal{P}$ is a *valid* set of periods iff there is a string realizing it as the set of its periods. In this paper for each considered set of periods we assume that it is valid: there *exists* some unknown string realizing this set.

It is known (and rather surprising and quite nontrivial) that binary alphabet is sufficient in the following sense: any valid set of periods (no restriction on the alphabet) is realized by a *binary* string.

The first algorithm to construct a binary string w realizing a given set of periods was given in [7]. Then, using a different and simpler approach, it has been shown in [4] that w can be constructed in $O(|w|)$ time.

The interesting point here was that the constructed string is binary, though an unknown string can be over arbitrary alphabet. However there is another interesting point: the construction given in [4] implicitly shows that the produced string $|w|$ is highly compressible.

In this paper we give two fast and simple algorithms which produce compact representations of highly compressible binary strings w realizing a given set of periods in time linear with respect to the size of the representation, which is $O(n)$. The explicit size N of w can be exponential with respect to n.

E-mail address: rytter@mimuw.edu.pl.

[1] The author is supported by grant no. NCN2014/13/B/ST6/00770 of the National Science Centre.

http://dx.doi.org/10.1016/j.tcs.2016.04.027
0304-3975/

Please cite this article in press as: W. Rytter, Two fast constructions of compact representations of binary words with given set of periods, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.04.027

A string-period of a word w is any string $z = uv$ such that $w = (uv)^k u$, where u is a proper prefix of z (possibly empty) and k is a positive integer. Hence an integer p is a period of w iff it is the size of some string-period of w. For example the word *ababa* has 3 string-periods $\{ab, abab, ababa\}$. The periodic structure of a string is given by the set $\mathcal{P}(w)$ of integers which are its periods.

A border is an integer $1 \le q \le |w|$ such that $|w| - q$ is a period of w or $q = |w|$. A border-string of w is a string which is simultaneously a prefix and a suffix of w. We distinguish between borders (integers) and string-borders. Consequently borders are sizes of string-borders for a given string. In this paper we consider equivalent structure: integer sequences $\mathcal{B}$ of borders, since they are more convenient in the description of algorithms. Instead of periodic structure of a given string w we use the border structure given by the increasing sequence $\mathcal{B}(w) = (q_1, q_2, \ldots q_n)$ of sizes of all string-borders of w, including $q_n = |w| = N$.

Example 1.1. The sequence of periods of the string $w = abaaba$ is $\mathcal{P} = (3, 5, 6)$ and its sorted sequence of borders is $\mathcal{B} = (1, 3, 6)$. In this case the length of an unknown string is $N = 6$, and the length of the border sequence in $n = 3$.

A string v is said *to realize* the border sequence $\mathcal{R}$ iff $\mathcal{R} = \mathcal{B}(v)$ is the border sequence of v. In the whole paper we assume the input border $\mathcal{R}$ is *valid*, which means that there exists some (possibly unknown) string w realizing $\mathcal{R}$, in other words $\mathcal{B}(w) = \mathcal{R}$.

There are many types of compressed representation of words, see [9]. One such representation is in terms of *collage systems*, see [1]. A collage system defines larger strings in terms of concatenations of parts of smaller, already defined, strings. Such system is an extension of *grammar-based compression*, see [9]. The collage system defines the sequence of string variables, where we can use prefixes/suffixes of previously defined values of smaller variables to define the values of larger ones.

For a string x denote by $Pref(X, j)$ the prefix of X of size j. A collage system is formally defined as a sequence of assignments:

$$X_1 = expr_1;\ X_2 = expr_2;\ \ldots;\ X_n = expr_n,$$

where each X_k is a variable and $expr_k$ is of one of the forms:

- a single letter,
- $X_i \cdot X_j$ for $i; j < k$,
- $Pref(X_i, j)$ for $i < k$ and an integer j,
- $(X_i)^j$ for $i < k$ and an integer j.

Assume we know $\mathcal{B}$ of some unknown string, but we don't know the string w. In the reconstruction process we construct two binary strings realizing $\mathcal{B}$.

1. A binary string, denoted by $Alternating(\mathcal{B})$, highly compressible but without particular structure otherwise. It is interesting since it gives an especially simple reconstruction algorithm.
2. The string $LexFirst(\mathcal{B})$ which is the lexicographically-first binary string realizing $\mathcal{B}$. This string has also a short compressed representation which will be computed in $O(n)$ time.

Example 1.2. Assume the "unknown" string is the Fibonacci word *abaababaabaab*, its border sequence is $\mathcal{B} = (2, 5, 13)$. The first algorithm gives as the output the string 0110100001101 and the second one will give the lexicographically-first string 0100100001001. Both of them realize the given border sequence $(2, 5, 13)$.

Both our algorithms have similar structure.

X_1 = a border-free string of size q_1;
for $i = 1$ **to** $n - 1$ **do**
  if $q_{i+1} - q_i \le q_i$ **then** *write*("$X_{i+1} = Pref(X_i, q_{i+1} - q_i) \cdot X_i$");
  else
    $\sigma_i := q_{i+1} - 2 \cdot q_i$;
    write("$X_{i+1} = X_i \cdot Y_i \cdot X_i$") where Y_i is a binary string of length σ_i
    such that $X_i \cdot Y_i \cdot X_i$ has no proper string-border longer than $|X_i|$;

Please cite this article in press as: W. Rytter, Two fast constructions of compact representations of binary words with given set of periods, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.04.027