Optimality of some algorithms to detect maximal quasiperiodicities

R. Groult, G. Richomme
UPJV, Laboratoire MIS
33, Rue Saint Leu
80039 Amiens cedex 01
{richard.groult, gwenael.richomme}@u-picardie.fr

1 Maximal quasiperiodicities

The study of repetitions in words constitutes an important stream of research both in combinatorics on words and in text algorithmics. In 1993, Apostolico and Ehrenfeucht introduced the notion of *quasiperiodic* words, a kind of approximated repetition [1]: a word w is quasiperiodic if there exists a second word $q \neq w$, called *a quasiperiod* of w, such that occurrences of q in w cover w entirely, *i.e.* every position of w falls within some occurrence of q in w. For instance, the smallest quasiperiod of the word *abaababaabaab* is *abaab*.

Apostolico and Ehrenfeucht designed an algorithm detecting all quasiperiodic factors of a word w in time $O(|w|(log(|w|))^2)$. Actually, the output of their algorithm consists only on some particular quasiperiodicities, called *maximal quasiperiodicities*. A *quasiperiodicity* of a word w is any quasiperiodic factor of w, and a *maximal* quasiperiodicity is a quasiperiodicity that cannot be extended to a longer quasiperiodicity. More precisely, let $w = w_1 \dots w_n$ be a word. We denote by $w[i..j]$ the word $w_i \dots w_j$, for $1 \leq i \leq j \leq n$. A quasiperiodicity $w[i..j]$ is a maximal quasiperiodicity of w if, q being its smallest quasiperiod:

1. $w[i..j]$ cannot be extended to a longer quasiperiodicity with same quasiperiodicity q, and
2. when considering the $j+1^{\text{th}}$ letter of w (with $j < |w|$), $q.w_{j+1}$ does not cover $w[i..j+1]$.

For instance, the word *abaababaabaab* contains 9 maximal quasiperiodicities.

Two different $O(|w|log(|w|))$ time complexity algorithms detecting maximal quasiperiodicities were designed by Iliopoulos and Mouchard in 1999 [4], and by Brodal and Petersen in 2000 [2]. The natural question we consider and solve here is: are these later algorithms optimal?

2 Maximal quasiperiodicities of the Fibonacci words

As said in [3], "*Fibonacci strings turn out to constitute worst cases for a number of computer algorithms which find generic patterns in strings*". Hence, we naturally consider Fibonacci words and Theorem 2.1 below provides exact formulas for the number of their quasiperiodicities.

Let recall that Fibonacci words $(f_n)_{n\geq 0}$ are defined inductively by $f_0 = a$, $f_1 = ab$ and for all $n \geq 2$, $f_n = f_{n-1}f_{n-2}$. One of the numerous properties of Fibonacci words is the fact that the lengths of these words are the Fibonacci numbers. Indeed denoting $F_n = |f_n|$, we have $F_0 = 1$, $F_1 = 2$ and for all $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$.

Theorem 2.1 *The number of maximal quasiperiodicities of* f_n *is in* $\Theta(|f_n|)$. *More precisely, this number is* 0 *for* $n = 0, 1, 2$, *is* 1 *for* $n = 3$ *and, for* $n \geq 4$, *it values:*

$$\begin{cases} F_{n-1} + F_{n-4} - 3 & \textit{if } n \textit{ is even,} \\ F_{n-1} + 2F_{n-5} - 1 & \textit{if } n \textit{ is odd.} \end{cases}$$

The main property that we use to prove this result is the following well-known relation between Fibonacci words and the Fibonacci morphism φ defined by $\varphi(a) = ab$, $\varphi(b) = a$: for all $n \geq 1$, $f_n = \varphi^n(a)$ (where for any morphism h, h^n is defined inductively by h^0 is the identity morphism, and $h^n = h^{n-1} \circ h$ for $n \geq 1$). In other words, $f_0 = a$ and for all $n \geq 1$, $f_n = \varphi(f_{n-1})$. This morphic vision allows to relate quasiperiodicities of two successive Fibonacci words.

3 About the number of maximal quasiperiodicities

Theorem 2.1 does not answer the question raised in Section 1. For this purpose, we consider the following words. Let consider the morphism f defined over $\{a, b\}$ by

$$\begin{cases} f(a) = ababa, \\ f(b) = bbb, \end{cases}$$

and the sequence of words $(u_n)_{n\geq 0}$, such that for all $n \geq 0$,

$$u_n = f^n(bababab).$$

Theorem 3.1 *The number of occurrences of maximal quasiperiodicities of* u_n, *for all* $n \geq 0$, *is in* $\Theta(|u_n| log(|u_n|))$.

In conclusion, an immediate corollary of Theorem 3.1 is:

Corollary 3.2 *The algorithms provided in* [4] *and* [2] *to find all maximal quasiperiodicities of a word* w *have an optimal* $O(|w| log(|w|))$ *time complexity.*

References

[1] A. Apostolico and A. Ehrenfeucht. Efficient detection of quasiperiodicities in strings. *Theoretical Computer Science*, 119:247–265, 1993.

[2] Gerth Stølting Brodal and Christian N. S. Pedersen. Finding maximal quasiperiodicities in strings. In *Combinatorial Pattern Matching (CPM'2000), 11th Annual Symposium, CPM 2000, Montreal, Canada, June 21-23, 2000*, volume 1848 of *Lecture Notes in Computer Science*, pages 397–411, 2000.

[3] C. S. Iliopolos, D. Moore, and W. F. Smyth. The covers of a circular Fibonacci string. *J. Combinatorial Math. and Combinatorial Computing*, 26:227–236, 1998.

[4] C. S. Iliopoulos and L. Mouchard. Quasiperiodicity: from detection to normal forms. *Journal of Automata, Languages and Combinatorics*, 4(3):213–228, 1999.