



Theoretical Computer Science 218 (1999) 205–216

---

---

**Theoretical  
Computer Science**

---

---

# Quasiperiodicity and string covering

Costas S. Iliopoulos<sup>a,b</sup>, Laurent Mouchard<sup>a,c,\*</sup>,<sup>2</sup><sup>a</sup> *Department of Computer Science, King's College London, Strand, London, WC2R 2LS, UK*<sup>b</sup> *School of Computing, Curtin University of Technology, GPO Box U1987, Perth 6845, Western Australia*<sup>c</sup> *LIR – ABlSS, Université de Rouen, 76821 Mont Saint Aignan Cedex, France*

---

## Abstract

In this paper, we study word regularities and in particular extensions of the notion of the word period: quasiperiodicity, covers and seeds. We present overviews of algorithms for computing the quasiperiodicity, the covers and the seeds of a given word. We also present an overview of an algorithm that finds maximal word factors with the above regularities. Finally, we show how Fine and Wilf's Theorem fails if we try to extend it directly to quasiperiodicity, as well as a new property on concatenation of periodic words. © 1999 Elsevier Science B.V. All rights reserved.

*Keywords:* Regularities; Periodicity; Quasiperiodicity; Covers; Seeds

---

## 1. Introduction

This paper is focussed on the study and the identification of various kinds of periodicities and other regularities in words; much of it turns out to bear directly on problems that arise in DNA sequence analysis, image validation/decomposition and melody recognition. The ability to locate repeats is useful in a wide area of applications which involve word manipulations. Pattern recognition, computer vision, speech recognition, data compression, data communication, combinatorics, coding and automata theory, formal language theory, system theory, are classic examples. Finding repeats also has applications in database work and general text editing, such as finding duplicate entries in a database.

The study of word repetitions and word periodicity was pioneered by Axel Thue at the beginning of this century [20], and since then it has been intensively studied and it has been one of the building blocks in Automata and Formal Language Theory,

---

\* Corresponding author. E-mail: [lm@dir.univ-rouen.fr](mailto:lm@dir.univ-rouen.fr).

<sup>1</sup> Partially supported by the EPSRC grant GR/J 17844.

<sup>2</sup> Supported by the Conseil Régional de Haute Normandie.

Algebraic Coding, Systems Theory and Combinatorics. Thue in [20] has constructed an infinite word over an alphabet of size 3 which contains no square proving that squares are avoidable regularities in words.

A typical regularity, the period  $u$  of a given word  $x$ , grasps the repetitiveness of  $x$  since  $x$  is a prefix of a word constructed by concatenations of  $u$ . For example,  $u^2 = abaaba$  is a repetition of  $u = aba$  and  $abc$  is a period of  $abcabcabca$ . Apostolico et al. [2] introduced the notion of quasiperiodicity, which extends periodicity by allowing not only the repetition of similar factors, but also superpositions. For example, a superposition of  $u = aba$  is  $v = ababa$ .

A factor  $w$  of  $x$  is called a *cover* of  $x$  if  $x$  can be constructed by concatenations and superpositions of  $w$ . The smallest such cover is called *the quasiperiod* of the word.

A factor  $w$  of  $x$  is called a *seed* of  $x$  if there exists an extension of  $x$  which is constructed by concatenations and superpositions of  $w$ .

For example,  $w = abaabababababababa$  can be obtained by a series of six concatenations or superpositions of the factor  $u = aba$ . Furthermore,  $abca$  is a seed of  $abcabcaabc$ .

The notions “cover” and “seed” are generalizations of periods in the sense that superpositions as well as concatenations are considered to define them, whereas only concatenations are considered for periods.

In the next section we present the basic definitions. In Section 3 we present results on periodicity and in Section 4 we survey results on quasiperiodicity (from [1, 13]). In Section 5 we present a “well-known” result (due to Fine and Wilf [12]) which fails if we try to extend it to quasiperiodicity. And finally in Section 6 we present a new result on the concatenation of two periodic words.

## 2. Preliminaries

An *alphabet*  $\mathcal{A}$  is a set of elements, that we will call *letters*, characters or symbols. A *word*  $w$  over the alphabet  $\mathcal{A}$  is a sequence of zero or more letters of  $\mathcal{A}$ , that is  $(w_1, w_2, \dots, w_n)$  with  $w_i \in \mathcal{A}^*$ ,  $i = 1, 2, \dots, n$ . The *empty word* is the empty sequence (of zero letters) and it will be denoted by  $\varepsilon$ . The set of all words over  $\mathcal{A}$  is denoted by  $\mathcal{A}^*$ , and can be equipped with a binary operation, *concatenation*:

$$\begin{aligned} \therefore \mathcal{A}^* \times \mathcal{A}^* & \longrightarrow \mathcal{A}^* \\ ((a_1, \dots, a_m), (b_1, \dots, b_n)) & \longmapsto (a_1, \dots, a_m, b_1, \dots, b_n) \end{aligned}$$

The concatenation of  $k$  copies of  $w$  is denoted by  $w^k$  and is called the *k-th power* of  $w$ . As concatenation is obviously associative and we can write  $(a_1) = a_1$ , this allows us to write  $(w_1, w_2, \dots, w_n)$  as  $w_1 w_2 \dots w_n$  and  $n$  is called the *length* of  $w$  and is denoted by  $|w|$ .

Let  $w$  be a word over  $\mathcal{A}$ . A word  $u$  over  $\mathcal{A}$  is a *factor* of  $w$  if and only if there exist two words  $t$  and  $v$  over  $\mathcal{A}$  such that  $w = twv$ . A word  $u$  over  $\mathcal{A}$  is a *left (respectively*

right) factor of  $w$  or a prefix (respectively suffix) of  $w$  if and only if there exists a word  $v$  over  $\mathcal{A}$  such that  $w = uv$  (resp.  $w = vu$ ).

For example,  $bb$  is a factor  $ababbc$ ,  $ab$  is a left factor of  $ababbc$  and  $bbc$  is a right factor of  $ababbc$ . A word  $u$  over  $\mathcal{A}$  is an extension of  $w$  if and only if  $w$  is a factor of  $u$ . A word  $u$  over  $\mathcal{A}$  is a left (resp. right) extension of  $w$  if and only if  $w$  is a right (resp. left) factor of  $u$ .

For example  $ababbc$  is a right extension of  $aba$  and  $ababbc$  is a left extension of  $bbc$ . A word  $u$  over  $\mathcal{A}$  is a border of  $w$  if and only if there exist two words  $t$  and  $v$  over  $\mathcal{A}$  such that  $w = tu$  and  $w = uv$ , i.e.  $u$  is both prefix and suffix of  $w$ .

For example,  $a$ ,  $aba$  and  $ababa$  are borders of  $ababaababa$ .

### 3. Periodicity and primitivity

Let  $w$  be a word over  $\mathcal{A}$ . The word  $w$  is said to be primitive if and only if  $w$  is not a  $k$ th power of another word. Thus, for any word  $u$  over  $\mathcal{A}$ , setting  $w = u^k$  implies  $u = w$  and  $k = 1$ .

For example  $w = abaababa$  is primitive but  $w' = abaabaaba = (aba)^3$  is not. A word  $w$  over  $\mathcal{A}$  is said to be strongly primitive<sup>3</sup> if and only if every factor of  $w$  is primitive. For example the word  $w = aba$  is strongly primitive. One can easily prove that any word over a binary alphabet whose length is greater or equal to 4 contains a square (2nd power of a word) and therefore it is not strongly primitive.

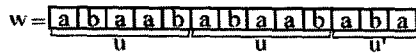


Fig. 1. Periodicity.

An integer  $p$  is a period of  $w$  if and only if  $w_i = w_{i+p}$  for  $0 < i \leq |w| - p$ . The smallest period is called the period. For example,  $ab$ ,  $aba$ ,  $abab$  are periods of  $abababab$  and  $ab$  is the period of  $abababab$ .

By misnomer, we have the following definition: a word  $u$  over  $\mathcal{A}$  is a period of  $w$  if and only if there exists  $k$  such that  $w$  is a prefix of  $u^k$ . A word  $w$  is periodic if and only if  $w$  has a period  $p$  such that  $p \leq |w|/2$ .

#### 3.1. A “well-known” property on periodicity

We consider a standard property on periodicity which is due to Fine and Wilf [12] and in Section 5, we show how it fails as soon as we consider quasiperiodicity instead of periodicity.

**Theorem 1** (Fine and Wilf [12]). *Let  $w$  be a word over the alphabet  $\mathcal{A}$  and  $m$  and  $n$  be two periods of  $w$ . The condition  $|w| \geq m + n - \gcd(m, n)$  implies  $\gcd(m, n)$  is a period of  $w$ .*

<sup>3</sup> Also said square-free.



It is not difficult to note that a periodic string is always quasiperiodic, but the converse is not true. Also, clearly a superprimitive string is always primitive, however the converse is not true. For example, *aba* is superprimitive and primitive, but *abaabaab* is primitive but not superprimitive, since the superprimitive string *abaab* covers it.

The algorithm in [2] is based on the observation that the cover of a word  $x$  is also a border of  $x$ : the cover of  $x$  must cover positions 1 and  $n$  of  $x$ . The algorithm in [2] exploits this fact by using the failure function of [16] for computing the borders and then testing whether they cover the word or not. The linear time is dominated by the computation of the failure function and it is achieved by reducing periodical cases to primitive ones.

In the computation of covers, two problems have been considered in the literature: the quasiperiodicity problem (also known as the superprimitivity test) is that of computing the shortest cover of a given string of length  $n$ , and the *all-covers* problem is that of computing all the covers of a given string. Breslauer [6] presented a linear-time on-line algorithm for the quasiperiodicity problem. Moore and Smyth [19] presented a linear-time algorithm for the all-covers problem. Lin and Smyth in [18] presented an on-line computation of the all-covers problem.

In parallel computation, Breslauer [6] gave two algorithms for the shortest-cover problem. The first one is an optimal  $O(\alpha(n) \log \log n)$ -time algorithm, where  $\alpha(n)$  is the inverse Ackermann function, and the second one is a non-optimal algorithm that requires  $O(\log \log n)$  time and  $O(n \log n)$  work. Breslauer [7] also obtained an  $\Omega(\log \log n)$  lower bound on the time complexity of the shortest-cover problem from the lower bound of string matching [5]. Iliopoulos and Park in [14] gave a work-time optimal  $O(\log \log n)$  algorithm for the shortest-cover problem and in [15] a work-time optimal  $O(\log \log n)$  algorithm for the all-covers problem.

#### 4.1. Efficient detection of quasiperiodicities in word

Apostolico and Ehrenfeucht presented in [1] an algorithm to find all the maximal quasiperiodic factors of a given word, that is find all the longest  $a$ -covered factors of a word. A quasiperiodic factor  $z$  is maximal, if no extension of  $z$  could be covered by either the same word  $w$  covering  $z$  or by an extension  $wa$  of  $w$ . All maximal  $a$ -covered factors of a word  $w$  over  $\mathcal{A}$  can be detected in time  $O(|w| \log^2 |w|)$ .

The algorithm in [1] shadows the Apostolico and Preparata [3] algorithm for detection of all the squares in a string. It is not difficult to see the link between the two problems: the starting position of every quasiperiodic factor is also the starting position of a square. The main steps of the Apostolico and Ehrenfeucht algorithm for a given word  $w$  over  $\mathcal{A}$  is as follows:

1. Build the compact suffix tree for the word  $w$ .
2. For each node  $n_i$  of this tree, maintain the list of all leaves in the subtree whose root is  $n_i$ .
3. For each list  $l_i$ , compute its span, that is, the longest uninterrupted cover.
4. Determine the longest span over the tree and the appropriate factor.

**Example.** Given  $w = abaabababa$ . Find all maximal quasiperiodic factors of  $w$ . Build the compact suffix tree for the word  $w = abaabababa$  (Fig. 5).

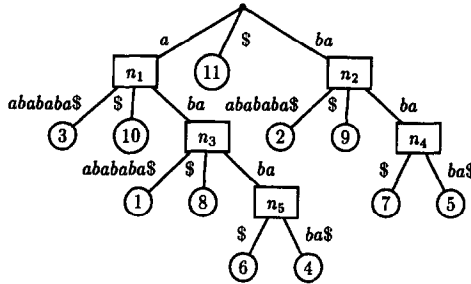


Fig. 5. Suffix-tree of  $w = abaabababa$ .

For each node  $n_i$  of this tree, maintain the list of all leaves in the subtree whose root is  $n_i$  (Fig. 6).

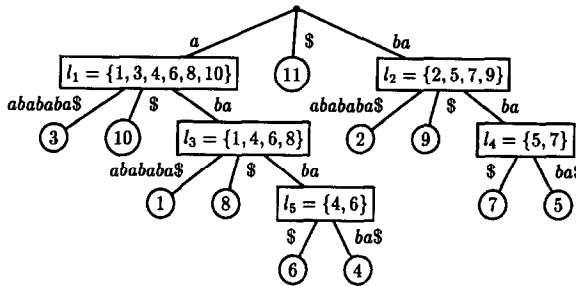


Fig. 6. Suffix-tree of  $w = abaabababa$  with lists

For each list  $l_i$ , compute its span (longest uninterrupted cover)

| Node  | Factor(s)     | List                          | Span                     |
|-------|---------------|-------------------------------|--------------------------|
| $n_1$ | $a$           | $l_1 = \{1, 3, 4, 6, 8, 10\}$ | 2 (3-4)                  |
| $n_2$ | $ba$          | $l_2 = \{2, 5, 7, 9\}$        | 6 (5-10)                 |
| $n_3$ | $ab, aba$     | $l_3 = \{1, 4, 6, 8\}$        | 10 (1-10) for $u = aba$  |
| $n_4$ | $bab, baba$   | $l_4 = \{5, 7\}$              | 6 (5-10) for $u = baba$  |
| $n_5$ | $abab, ababa$ | $l_5 = \{4, 6\}$              | 7 (4-10) for $u = ababa$ |

Determine the longest span over the tree and the appropriate factor. The longest span is 10, for node  $n_3$  and factor  $aba$ . In this example, the word  $w = abaabababa$  is  $a$ -covered by  $u = aba$  as  $\text{span}(u) = |w|$ .

One observation that may lead to faster than the Apostolico and Ehrenfeucht algorithm is that the number of different spans than one can have at any given position is bounded by log  $n$ :

Let  $b_1, \dots, b_k$  be the sequence  $B$  of all nontrivial borders of  $x$  from smallest to largest. Let  $b_0$  denote the empty string and  $b_{k+1}$  the given string  $x$ . A subsequence  $b_l, \dots, b_{l+m}$  of  $B$  is said to be a *chain of covers* of  $x$  if every  $b_{l+i}$ ,  $0 \leq i \leq m$ , covers  $x$ . If additionally each of  $b_{l-1}$  and  $b_{l+m+1}$  is a trivial border or it does not cover  $x$ , then  $b_l, \dots, b_{l+m}$  is said to be a *maximal chain of covers*.

**Theorem 2.** *There are at most  $\lceil \log n \rceil$  maximal chains of covers.*

**Proof.** See [15].  $\square$

#### 4.2. Covering a string

Iliopoulos et al. [13] have proposed a new notion of string regularity and an extension of the notions of period and cover, called *seed*. The focus of [13] was on the *General String Covering* problem. We say that a word  $u$  covers a word  $w$  if there exists an extension of  $w$  which is constructed by concatenations and superpositions of  $y$ . For example,  $abca$  covers  $abcabcaabc$ . A factor  $u$  of a word  $w$  is called a *seed* of  $x$  if  $u$  covers  $w$ . The GSC problem is as follows: given a word  $w$  of length  $n$ , compute all the seeds of  $w$ . Note that there may be more than one shortest seed (e.g. for  $abababa$ , both  $ab$  and  $ba$  are the shortest seeds). In [13] a method for finding all the seeds of a given word  $w$  over  $\mathcal{A}$  in time  $O(|w| \log |w|)$  is presented. A parallel PRAM algorithm and a lower bound for the GSC can be found in [4].

The seeds of  $w$  are classified into two kinds: A seed  $u$  is an *easy seed* if there is a factor of  $u$  which covers  $w$  by concatenations only;  $u$  is a *hard seed* otherwise. For example, for  $x = (abab)^3abb$ , the words  $abbab, babab$  cover  $w$  by concatenations and thus are easy seeds. The words  $babbab, bababba$  are also easy seeds of  $w$ , having  $abbab$  and  $babab$  as factors respectively which cover  $w$  by concatenations. But the word  $bab$  is a hard seed of  $w$ . Let  $u = w_1 \cdots w_p$  be the period of  $w$ . It is easy to see that  $u$  covers  $w$  by concatenations. The following lemmas characterize easy and hard seeds of  $w$ .

**Easy seeds:** A seed  $u$  is an *easy seed* if there is a factor of  $u$  which covers  $w$  only by concatenations. Otherwise  $u$  is a *hard seed* of  $w$ .

**Theorem 3.** *A factor  $u$  of a word  $w$  over  $\mathcal{A}$  is an easy seed if and only if  $u$  is a right extension in  $w$  of a conjugate of the period.*

**Proof.** See [13], page 4.  $\square$

Easy seeds can be found by the preprocessing of Knuth [16]. They can be found in time  $O(|w|)$ .

**Hard seeds:** A factor  $u$  of the word  $w$  over  $\mathcal{A}$  is a *candidate for a hard seed* if there exist  $(t, w', v)$ , words over  $\mathcal{A}$  such that

- $w = tw'v$ .





4. Determine the candidate sets.
5. If it is a hard seed, determine l-size, r-size and type-A or type-B.

**Example.** Consider  $w = babaabaaba$ . Compute hard seeds of  $w$ .

Compute the period of  $w$  from KMP [16] The period is 8. The word  $babaabaaba$  is not periodic.

Compute  $E_i$  for  $1 \leq i \leq p$  or all classes are singletons

$$\begin{aligned}
 E_1 &= \{\{1, 3, 6, 9\}, \{2, 4, 5, 7, 8, 10\}\}, & E_2 &= \{\{1, 3, 6, 9\}, \{2, 5, 8\}, \{4, 7\}\}, \\
 E_3 &= \{\{1\}, \{3, 6\}, \{2, 5, 8\}, \{4, 7\}\}, & E_4 &= \{\{1\}, \{3, 6\}, \{2, 5\}, \{4, 7\}\}, \\
 E_5 &= \{\{1\}, \{3, 6\}, \{2, 5\}, \{4\}\}, & E_6 &= \{\{1\}, \{3\}, \{2, 5\}, \{4\}\}, \\
 E_7 &= \{\{1\}, \{3\}, \{2\}, \{4\}\}.
 \end{aligned}$$

Table 1  
Compute start-sets and equi-sets

| Equi-set                       | Start-set        | Equi-set               | Start-set               |
|--------------------------------|------------------|------------------------|-------------------------|
| $\{b, ba\}$                    | $\{1, 3, 6, 9\}$ | $\{a\}$                | $\{2, 4, 5, 7, 8, 10\}$ |
| $\{ab, aba\}$                  | $\{2, 5, 8\}$    | $\{aa, aab, aaba\}$    | $\{4, 7\}$              |
| $\{bab, baba, babaa, babaab\}$ | $\{1\}$          | $\{baa, baab, baaba\}$ | $\{3, 6\}$              |
| $\{abaa, abaab, abaaba\}$      | $\{2, 5\}$       | $\{aabaa, aabaab\}$    | $\{4\}$                 |
| $\{baabaa\}$                   | $\{3\}$          | $\{abaabaa\}$          | $\{2\}[1pt]$            |

Table 2  
Determine candidate sets

| Candidate   | Decomposition              | Candidate   | Decomposition              |
|-------------|----------------------------|-------------|----------------------------|
| $\{aba\}$   | $(b)(abaabaaba)(\epsilon)$ | $\{baa\}$   | $(ba)(baabaa)(ba)$         |
| $\{aaba\}$  | $(bab)(aabaaba)(\epsilon)$ | $\{abaa\}$  | $(b)(abaabaa)(ba)$         |
| $\{baab\}$  | $(ba)(baabaab)(a)$         | $\{aabaa\}$ | $(bab)(aabaa)(ba)$         |
| $\{abaab\}$ | $(b)(abaabaab)(a)$         | $\{baaba\}$ | $(ba)(baabaaba)(\epsilon)$ |

Table 3  
If hard seed, determine l-size, r-size and type-A or type-B

| Candidate   | Decomposition              | Seed ? | l-size | r-size | type |
|-------------|----------------------------|--------|--------|--------|------|
| $\{aba\}$   | $(b)(abaabaaba)(\epsilon)$ | Yes    | 1      | 0      | A    |
| $\{baa\}$   | $(ba)(baabaa)(ba)$         | No     |        |        |      |
| $\{aaba\}$  | $(bab)(aabaaba)(\epsilon)$ | No     |        |        |      |
| $\{abaa\}$  | $(b)(abaabaa)(ba)$         | No     |        |        |      |
| $\{baab\}$  | $(ba)(baabaab)(a)$         | No     |        |        |      |
| $\{aabaa\}$ | $(bab)(aabaa)(ba)$         | No     |        |        |      |
| $\{abaab\}$ | $(b)(abaabaab)(a)$         | Yes    | 0      | 2      | B    |
| $\{baaba\}$ | $(ba)(baabaaba)(\epsilon)$ | Yes    | 0      | 0      | A    |

The details of this procedure can be found in [13].

### 5. Extension of Fine and Wilf’s theorem

Fine and Wilf’s Theorem does not hold, if we consider quasiperiods instead of periods. For example we consider the word  $u = (ab)^n a$ . This word is obviously periodic (for  $n > 1$ ) and 2,4,... are periods. This word has two trivial quasiperiods:  $m = 2$  and  $n = 3$  (for words  $ab$  and  $aba$ ).

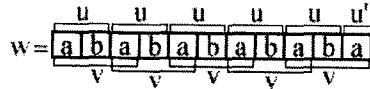


Fig. 7.  $\gcd(2,3)=1$  is not a cover of  $w$ .

Extending Fine and Wilf’s Theorem to quasiperiods will incite us to imagine that  $\gcd(m, n) = 1$  is a period of  $u$ , which is obviously false. We have presented quasiperiodicity as an extension of periodicity, but some of the known properties on periodicity cannot be extended directly to quasiperiodicity. On the other hand we will show in the following section that quasiperiodicity may be useful to solve problems related to periods.

### 6. Concatenation of two periodic – quasiperiodic words

A point of a word  $w$  over  $\mathcal{A}$  is a pair of words  $(w', w'')$  over  $\mathcal{A}$  such that  $w = w'w''$ . We will always assume  $w'$  and  $w''$  are not empty.

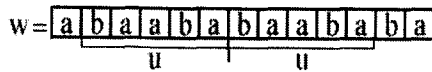


Fig. 8.  $u = abaaba$  is a repetition at point  $(abaaba, baababa)$

A nonempty word  $u$  over  $\mathcal{A}$  is a repetition<sup>4</sup> at point  $(w', w'')$  of  $w$  if and only if  $u$  is a suffix of a left extension of  $w'$  and a prefix of a right extension of  $w''$  ( $\mathcal{A}^* u \cap \mathcal{A}^* w' \neq \emptyset$  and  $u \mathcal{A}^* \cap w'' \mathcal{A}^* \neq \emptyset$ ). The minimum of the length of the repetitions at point  $(w', w'')$  is the local period<sup>5</sup> at point  $(w', w'')$ . A critical point is a point  $(w', w'')$  whose local period is maximal.

#### 6.1. Concatenation of two periodic words

Given two periodic words  $u$  of period  $m$  and  $v$  of period  $n$ , we can consider the concatenation of  $u$  and  $v$  as a point  $(u, v)$  and we can try to figure out if the observation of local periods at point  $(u, v)$  can be of any help in finding a global period of word  $uv$ . The general answer has been pointed out by Duval [11], but is very restrictive

<sup>4</sup> Also called cross factor.

<sup>5</sup> Also called virtual period.

since the words have to be  $k$ th-power of one of their proper factor whose length is a local period at point  $(u, v)$ .

Here we present a different result based on quasiperiodicity instead of periodicity.

**Theorem 4.** *Given two periodic words  $u$  of period  $m$  and  $v$  of period  $n$ , if there exists a repetition of length  $p$  at point  $(u, v)$  such that  $\max(m, n) \leq p \leq (|uv|)/2$  then  $uv$  is quasiperiodic of cover at most  $p$ .*

**Proof.** A right extension of  $v$  is  $a$ -covered by  $v_1 \dots v_p$ . A left extension of  $u$  can be constructed using only superpositions of  $v_1 \dots v_p$  with  $p - m$  overlaps. So an extension of  $uv$  is  $a$ -covered by  $v_1 \dots v_p$  and  $p \leq |uv|/2$ . Then  $uv$  is quasiperiodic of cover  $q \leq p$ .  $\square$

**Example:** The words  $u = abaababaababaaba$  and  $v = abaababaababa$  are periodic (the period is  $w = abaab$ ). Repetitions at point  $(abaababaababaaba, abaababaababa)$  include  $abaaba$  whose length is 6, which is greater or equal to 5. Then  $uv = abaababaababaabaabaababaababa$  is quasiperiodic and the quasi-period at most equal to 6. The words  $aba$  and  $abaaba$  cover  $uv$ . Furthermore the word  $uv$  is quasiperiodic of cover 3.

## 6.2. Concatenation of two quasiperiodic words

Once again, we can imagine that extending the previous theorem to quasiperiodic words can lead us to an easy result, but an easy counterexample can be built.

The word  $u = aabababab$  and  $v = babababaa$  are quasiperiodic and  $aba$  covers both of them. The local period at point  $(u, v)$  includes  $babab$  whose length is greater or equal to the length of  $aba$ . But  $uv = aababababbabababaa$  is not quasiperiodic.

## 7. Conclusion

In this paper, we have presented a survey of results on quasiperiodicity, and an overview of algorithms that find maximal quasiperiodic factors and seeds. Locating such a regularity can be useful in a wide area of applications, for example in molecular biology (study of the *dosDNA* microsatellites). We have shown that properties of periodic words can not be directly extended to quasiperiodicities. We think it could be possible to extend the notion of local period to local quasiperiod to extend the Critical Factorisation Theorem given in [9] or [10], in order to find a broader context for this theorem.

## References

- [1] A. Apostolico, A. Ehrenfeucht, Efficient detection of quasiperiodicities in strings, *Theoret. Comput. Sci.* 119(2) (1993) 247–265.
- [2] A. Apostolico, M. Farach, C.S. Iliopoulos, Optimal superprimitivity testing for strings, *Inform. Process. Lett.* 39(1) (1991) 17–20.

- [3] A. Apostolico, F.P. Preparata, Optimal off-line detection of repetitions in a string, *Theoret. Comput. Sci.* 22(3) (1983) 297–315.
- [4] A. Ben-Amram, O. Berkman, C.S. Iliopoulos, K. Park, The subtree max gap problem with application to parallel string covering, *Proc. 5th ACM-SIAM Ann. Symp. on Discrete Algorithms*, Arlington, VA, 1994, pp. 501–510.
- [5] D. Breslauer, Z. Galil, A lower bound for parallel string matching *SIAM J. Comput.* 21(5) (1992) 856–862.
- [6] D. Breslauer, An on-line string superprimitivity test, *Inform. Process. Lett.* 44(6) (1992) 345–347.
- [7] D. Breslauer, Testing string superprimitivity in parallel, *Inform. Process. Lett.* 49(5) (1994) 235–241.
- [8] M. Crochemore, An optimal algorithm for computing the repetitions in a word, *Inform. Process. Lett.* 12(5) (1981) 244–250.
- [9] Y. Césari, M. Vincent, Une caractérisation des mots périodiques. *Comptes Rendus Acad. Sci. Paris Sér. I Math.* (1978) 1175–1177.
- [10] J.-P. Duval, Contribution à la combinatoire du monoïde libre, Thèse, Université de Rouen, 1980.
- [11] J.-P. Duval, Périodes locales et propagation de périodes dans un mot. Rapport LIR 97.05, Université de Rouen, France, 1997.
- [12] N.J. Fine, H.S. Wilf, Uniqueness theorems for periodic functions, *Proc. Amer. Math. Soc.* 16 (1965) 109–114.
- [13] C.S. Iliopoulos, D.W.G. Moore, K. Park, Covering a string, in A. Apostolico, M. Crochemore, Z. Galil, U. Manber (Eds.), *Proc. 4th Ann. Symp. on Combinatorial Pattern Matching*, Lecture Notes in Computer Science, vol. 684, Padova, Italy, Springer, Berlin, 1993, pp. 54–62.
- [14] C.S. Iliopoulos, K. Park, An optimal  $o(\log \log n)$ -time algorithm for parallel superprimitivity testing, *J. Korea Inform. Sci. Soc.* 21(8) (1994) 1400–1404.
- [15] C.S. Iliopoulos, K. Park, An  $o(\log \log n)$  pram algorithm for computing all the seeds of a string *Theoret. Comput. Sci.* 2(164) (1996) 299–310.
- [16] D.E. Knuth, J.H. Morris, Jr, V.R. Pratt, Fast pattern matching in strings, *SIAM J. Comput.* 6(1) (1977) 323–350.
- [17] M. Lothaire, *Combinatorics on Words*, Addison-Wesley, Reading, MA, 1983.
- [18] Yin Li, W.F. Smyth, An optimal on-line algorithm to compute all the covers of a string, 1998, submitted
- [19] A. Mateescu A. Salomaa, Nondeterminism in patterns, in: P. Enjalbert, E.W. Mayr, K.W. Wagner (Eds.), *Proc. 11th Ann. Symp. on Theoretical Aspects of Computer Science*, Caen, France, Lecture Notes in Computer Science, vol. 775 Springer, Berlin, 1994, pp. 661–668.
- [20] A. Thue, Über die gegenseitige lage gleicher teile gewisser zeichenreichen, *Nor. Vidensk. Selsk. Skr. Mat. Nat. Kl. 1* (1912) 1–67.