ELSEVIER

Theoretical Computer Science 249 (2000) 313–324

Theoretical Computer Science

www.elsevier.com/locate/tcs



# [illegible]

Laurent Mouchard [1]

*Faculte des Sciences, ABISS-ESA 6037, Université de Rouen, Pl. E Blondel, F-76821 Mont Saint Aignan Cedex, France*

**Abstract**

Here we consider the problem of computing normal forms of quasiperiodic strings. A string $x$ is *quasiperiodic* if it can be constructed by concatenation and superpositions of one of its proper factor (cover). The notion of quasiperiodicity is a generalization of *periodicity* in the sense that superpositions as well as concatenations are allowed to define it. It is shown here that given a quasiperiodic string $x$, there exists a unique factorization of $x$ into roots of its shortest cover and how we can efficiently build such a factorization in linear time. These forms can be used, for example, to test whether or not a string $v$ covers $x^k$ for some integer $k$, where $v$ covers $x$.


*Keywords:* Regularity; Quasiperiodicity; Overlap; Seed; Cover; Normal form

## 1. Introduction

In recent study of repetitive structures of strings, generalized notions of periods have been introduced. A typical regularity, the root $u$ of a given string $x$, grasps the repetitiveness of $x$ since $x$ is a prefix of a string constructed by concatenations of $u$. A substring $w$ of $x$ a-covers $x$, if $x$ can be constructed by concatenations and superpositions of $w$. We say that $w$ is a cover of $x$ and $x$ is *a-covered*. A substring $w$ of $x$ is called a *seed* of $x$, if there exists a superstring of $x$ which is constructed by concatenations and superpositions of $w$. In this case we say that $x$ is *covered*. For example, ACG is a period of ACGACGACGA, ACGA is a cover of ACGACGAACGA, and ACGA is a seed of GACGACGAACG. The notions "cover" and "seed" are generalizations

*E-mail address:* laurent.mouchard@univ-rouen.fr (L. Mouchard).

[1] Supported by Conseil Régional de Haute-Normandie and C.N.R.S. program "Informatique et Génome".


PII: S0304-3975(00)00065-7

of periods in the sense that superpositions as well as concatenations are considered to define them, whereas only concatenations are considered for periods. A variant of the covering problem [10] studied here, was shown to have applications to DNA sequencing by hybridization using oligonucleotide probes.

In computation of covers, two problems have been considered in the literature. The *shortest-cover* 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. Apostolico et al. [2] introduced the notion of covers and gave a linear-time algorithm for the shortest-cover problem. Breslauer [7] presented a linear-time on-line algorithm for the same problem. Moore and Smyth [15] presented a linear-time algorithm for the all-covers problem. In parallel computation, Breslauer [7, 8] 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, 8] 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. Iliopoulos and Park [13] gave an optimal $O(\log\log n)$-time (thus work-time optimal) algorithm for the shortest-cover problem.

Iliopoulos et al. [11] introduced the notion of seeds and gave an $O(n\log n)$-time algorithm for computing all the seeds of a given string of length $n$. For the same problem Ben-Amram et al. [4] presented a parallel algorithm that requires $O(\log n)$ time and $O(n\log n)$ work.

Apostolico and Ehrenfeucht [1] considered another variant of the covering problem; in [1] they presented an $O(n\log^2 n)$ algorithm for finding all the maximal quasiperiodic substrings (local covers) of a given string, i.e. find all the longest coverable substrings of a string. Informally, quasiperiodic substring $z$ is maximal, if no extension $z$ could be covered by either the same string $w$ covering $z$ or by an extension $wa$ of $w$. 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 association between the two problems: the starting position of every quasiperiodic substring is also the starting position of a square. Iliopoulos and Mouchard [12] presented an $O(n\log n)$-time algorithm solving the same problem. Here, we will rather focus on finding normal forms of quasiperiodic (and covered) strings after their covers or seeds have been found. This approach will enable us to study the spread of quasiperiodicity under concatenation, and additionally determine whether or not $v$ is a seed of $x^k$, when $v$ is a seed of $x$.

For example, $v=\mathrm{ATA}$ is a seed of $x=\mathrm{T}\underline{\mathrm{ATAATATATATAATA}}\mathrm{A}$ and the string $x$ can be written in the form $x=\mathrm{T.ATA.AT.AT.AT.ATA.ATA.A}$. From this form, we can easily derive that $x^k$ is covered by $v$ for $k\geqslant 1$.

In Section 2, we present basic definitions as well as a few remarks on the new definitions which will be useful for the results we will present. In Section 3, we present left and right normal forms of a-covered (and covered) strings and an efficient algorithm to compute these forms. In Section 4, we present the concatenation of covered strings. In Section 5, we conclude.

## 2. Preliminaries

A *string* is a sequence of zero or more symbols from an alphabet $\mathscr{A}$. The set of all strings over the alphabet $\mathscr{A}$ is denoted by $\mathscr{A}^\star$. The set of all non-empty strings over the alphabet $\mathscr{A}$ is denoted by $\mathscr{A}^+$.

A string of length $n$ is represented by $x_1 \dots x_n$, where $x_i \in \mathscr{A}$ for $1 \leqslant i \leqslant n$. A string $v$ is a *factor* of $x$ if $x = uvw$ for $u, w \in \mathscr{A}^\star$; we equivalently say that the string $v$ occurs at position $|u| + 1$ of the string $x$. A string $x$ is an extension of $v$ if $v$ is a factor of $x$, that is $x = uvw$ for $u, v \in \mathscr{A}^\star$. A string $u$ is a *prefix* (or a *left factor*) of $x$ if $x = uv$ for $v \in \mathscr{A}^\star$. We also write $u = x.v^{-1}$, for example, $\mathrm{ATGCATG} = \mathrm{ATGC.ATG}$ and therefore $\mathrm{ATGC} = \mathrm{ATGCATG.(ATG)}^{-1}$i.

Similarly, $v$ is a *suffix* (or a *right factor*) of $x$ if $x = uv$ for $u \in \mathscr{A}^\star$. We also write $v = u^{-1}x$, for example, $\mathrm{ATGCATG} = \mathrm{AT.GCATG}$ and therefore $\mathrm{GCATG} = (\mathrm{AT})^{-1}.\mathrm{ATGCATG}$

The reversal of $x = x_1 x_2 \dots x_{n-1} x_n$ is $\tilde{x} = x_n x_{n-1} \dots x_2 x_1$.

The string $xy$ is the *concatenation* of the two strings $x$ and $y$.

Let $X$ and $Y$ denote two sets of words. The set $X.Y$ is the set of all words obtained by concatening a word of $X$ and a word of $Y$. For example, if $X = \{\mathrm{ATA}, \mathrm{TA}\}$ and $Y = \{\mathrm{AT}, \mathrm{TT}\}$, $X.Y = \{\mathrm{ATAAT}, \mathrm{ATATT}, \mathrm{TAAT}, \mathrm{TATT}\}$.

$X^k$ is the set of all words obtained from $(k-1)$ concatenations of elements of $X$ ($X^k$ is the set of all words of length $k$ over the alphabet $X$).

For example, if $X = \{\mathrm{AT}, \mathrm{TA}\}$, $X^2 = \{\mathrm{ATAT}, \mathrm{ATTA}, \mathrm{TAAT}, \mathrm{TATA}\}$.

The concatenations of $k$ copies of $x$ is denoted by $x^k$ and is called the *kth power of* $x$. A string $u$ is *primitive* if the condition $u = v^k$ implies $v = u$ and $k = 1$. For example, ATAATAATA is the 3rd power of ATA and therefore is not primitive, but ATATAATA is primitive.

Any decomposition $w = v_1.v_2 \dots v_k$ of a word $w$ is called a factorization of $w$. A factorization is sometimes the finite sequence $(v_1, v_2, \dots, v_k)$ itself.

A subset $\mathscr{X}$ of $\mathscr{A}^\star$ is a code if any string in $\mathscr{X}^+$ can be written *uniquely* as a concatenation of strings in $\mathscr{X}$, that is, has a unique factorization into strings in $\mathscr{X}$. For example, $\mathscr{X}_1 = \{\mathrm{AT}, \mathrm{TA}\}$ is a code, since any string in $\mathscr{X}_1^+$ has a unique factorization into $\{\mathrm{AT}, \mathrm{TA}\}$. $\mathscr{X}_2 = \{\mathrm{AT}, \mathrm{ATA}, \mathrm{TA}\}$ is not a code, since $w = \mathrm{ATATA} \in \mathscr{X}_2^+$ can be written $w = \mathrm{AT.ATA} = \mathrm{ATA.TA}$.

Therefore, the factorization of a string in $\mathscr{X}_2^+$ into strings in $\mathscr{X}_2$ is not unique.

For two strings $x = x_1 \dots x_n$ and $y = y_1 \dots y_m$ such that $x_{n-i+1} \dots x_n = y_1 \dots y_i$ for some $i \geqslant 1$, the string $s = x_1 \dots x_{n-i} y_1 \dots y_m = x_1 \dots x_n y_{i+1} \dots y_m$ is a *superposition of* $x$ *and* $y$, in fact *the superposition of* $x$ *and* $y$ *with* $i$ *overlaps*. For example, if $s_1 = \mathrm{ATA}$ and $s_2 = \mathrm{ATT}$ then $\underline{\mathrm{ATATT}}$ is the superposition of $s_1$ and $s_2$ with 1 overlap.

Despite this definition, the concatenation of $x$ and $y$ can be regarded in a certain way as the superposition of $x$ and $y$ with 0 overlaps.

Let $x$ be a string of length $n$.

An integer $p > 0$ is *a period* of $x$ iff $\forall i \in [1, n-p] \quad x_i = x_{i+p}$.


Fig. 1. Periodicity.


Fig. 2. ATA a-covers ATAATATATA.

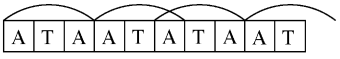



Fig. 3. ATA covers ATAATATAAT (since ATA a-covers ATAATATAATA).

If we consider $p \geqslant n$, $[1, n-p]$ is empty, and according to the above definition, $p$ is a period of $x$, therefore $p$ is a period of all the strings whose lengths are smaller than $p$. So, a non-empty string has at least one period, its length.

The shortest period is *the* period of $x$. Let $p$ be the period of a non-empty string $x$. The string $x$ is *periodic* if and only if $p \leqslant \lfloor |x|/2 \rfloor$. It means that $x = (uv)^k u$ with $|uv| = p$ and $k > 1$. For example, $s = \mathrm{ATAATAATA}$ has periods $3, 6, 8, 9$ and $p > 9$. The period of $s$ is 3 (see Fig. 1).

We say that a prefix $x_1 \ldots x_p, 1 \leqslant p \leqslant n$ of $x$ is a *root* of $x$, if $x_{i+p} = x_i$ for all $1 \leqslant i \leqslant n-p$.

*The* root of a string $x$ is the shortest root of $x$.

For example, $s = \mathrm{ATAATAATAA}$ has roots ATA, ATAATA, ATAATAATA, ATAATAATAA.

A string $b$ is a *border* of $x$ if $b$ is a prefix and a suffix of $x$. The empty string $\varepsilon$ and $x$ itself are trivial borders of $x$. For example, $s = \mathrm{ATAATATAATATA}$ has non-trivial borders (or *proper borders*) A, ATA and ATAATATA.

**Fact 1.** *A string $u \neq \varepsilon$ is a root of $x = ub$ iff $b$ is a border of $x$.*

For example, ATA is a root of $s = \mathrm{ATAATAAT}$ and therefore ATAAT is a border of $s = \underline{\mathrm{ATAATAAT}}$ (see Fig. 2).

Given a string $x$ and $v$ a proper factor of $x$, $v$ *a-covers* $x$ iff $x$ can be constructed by concatenations and superpositions of $v$ (*a-cover* stands for aligned-cover: *ASC — aligned string covering* [11]). We say that such a $v$ is a *cover* of $x$. The shortest cover is *the* cover. Note that all covers of $x$ are borders and therefore the cover is unique. We say that $u$ is quasiperiodic. A string is *superprimitive* if it is not quasiperiodic. For example, ATAAATA is superprimitive (see Fig. 3).

Given a string $x$ and $v$ a proper factor of $x$, $v$ *covers* $x$ if and only if an extension of $x$ can be constructed by concatenations and superpositions of $v$ (*GSC — general string covering* [11]). We say that $v$ is a *seed* of $x$.

## 3. Normal forms

In this section we present the underlying idea which leads to the normal forms. Consider the string $S = \mathrm{ATATA}$, which is quasiperiodic since it is a-covered by $s = \mathrm{ATA}$. We can decompose $S$ in the following two different forms: $S = \mathrm{ATA.TA}$ or $S = \mathrm{AT.ATA}$

The first of these two forms can be viewed as the concatenation of $s$ and one of its suffixes and the second one can be viewed as the concatenation of a prefix of $s$ and $s$. But considering any suffix of a cover for decomposing $S$ is meaningless, for example $S' = \mathrm{ATA.A}$ is not a-covered by $s = \mathrm{ATA}$ although A is a suffix of $s$.

Informally, consider a string $w$ which has at least one proper border $u$, i.e. there exist two non-empty strings $t$ and $v$ such that $w = tu$ and $w = uv$. Now consider the superposition of $w$ with itself with $|u|$ overlaps ($tuv$). This superposition can be written $s = w.u^{-1}.w$, and it can be viewed in two different ways (see the above definition of superposition) $s = wv = tw$ where $v = u^{-1}.w$ and $t = w.u^{-1}$. Formally, we have the following definitions.

**Definition 2.** Consider the string $u$.

The longest proper border of $u$ will be denoted by $\mathrm{Border}(u)$. The set of all borders of $\mathrm{Border}(u)$ (including the trivial borders of $\mathrm{Border}(u)$) will be denoted by $\mathscr{B}(u)$.

Furthermore, let

$$\mathscr{L}(u) = \{u.(\mathscr{B}(u))^{-1}\} = \{v \mid vw = u \text{ for } w \in \mathscr{B}(u)\},$$

i.e. the set of prefixes of $u$ obtained by cutting off an element of $\mathscr{B}(u)$ from the rightmost end of $u$ ($\mathscr{L}(u)$ is the set of all roots of $u$).

Similarly, let

$$\mathscr{R}(u) = \{(\mathscr{B}(u))^{-1}.u\} = \{v \mid wv = u \text{ for } w \in \mathscr{B}(u)\},$$

i.e. the set of suffixes of $u$ obtained by cutting off an element of $\mathscr{B}(u)$ from the leftmost end of $u$.

For example, for $s = \mathrm{AATAA}$, we have $\mathrm{Border}(s) = \mathrm{AA}$, $\mathscr{B}(s) = \{\varepsilon, \mathrm{A}, \mathrm{AA}\}$, $\mathscr{L}(s) = \{\mathrm{AAT}, \mathrm{AATA}, \mathrm{AATAA}\}$ and $\mathscr{R}(s) = \{\mathrm{TAA}, \mathrm{ATAA}, \mathrm{AATAA}\}$.

Note that $\varepsilon$ is not an element of $\mathscr{L}(s) \cup \mathscr{R}(s)$, since $s$ does not appear in $\mathscr{B}(s)$.

Once we have defined these sets, we can easily represent any possible superposition of $s$ with itself (eventually with no overlap, that is concatenation) as the concatenation of an element of $\mathscr{L}(s)$ and $s$, or $s$ and an element of $\mathscr{R}(s)$. In the above example, we can build the set of all possible superpositions of $s$ with itself as the set $\mathscr{S}_1(s) = \mathscr{L}(s).s = s.\mathscr{R}(s)$. For example

$$\begin{aligned}\mathscr{S}_1(\mathrm{AATAA}) &= \{\mathrm{AAT.AATAA}, \mathrm{AATA.AATAA}, \mathrm{AATAA.AATAA}\}\\ &= \{\mathrm{AATAA.TAA}, \mathrm{AATAA.ATAA}, \mathrm{AATAA.AATAA}\}\\ &= \{\mathrm{AATAATAA}, \mathrm{AATAAATAA}, \mathrm{AATAAAATAA}\}.\end{aligned}$$

Next we focus on the maximal length of the overlap of the superposition of a string $u$ with itself. Given a string $u$, the maximal length of the overlap can be as long as $(|u|-1)$ symbols for $u=a^n$ for example. But in the case of superprimitive strings (strings which are not a-covered by one of their proper factors), we obtain a different result:

**Proposition 3.** *If $u$ is superprimitive then* $|\mathrm{Border}(u)| \leqslant \lfloor |u|/2 \rfloor$.

**Proof.** Suppose that $|\mathrm{Border}(u)| > \lfloor |u|/2 \rfloor$. Since $\mathrm{Border}(u)$ is both prefix and suffix of $u$, $\mathrm{Border}(u)$ a-covers $u$ (contradiction). □

**Remark 4.** If $u$ is superprimitive then

$$\min_{w \in \mathscr{L}(u)} \{|w|\} = \min_{w \in \mathscr{R}(u)} \{|w|\} > \lfloor |u|/2 \rfloor.$$

A straightforward proof follows the definition of $\mathscr{L}(u)$, $\mathscr{R}(u)$ and Proposition 3.

For example, $s=\mathrm{AATAA}$, $\mathscr{B}(s)=\{\varepsilon, \mathrm{A}, \mathrm{AA}\}$, $\mathscr{L}(s)=\{\mathrm{AAT}, \mathrm{AATA}, \mathrm{AATAA}\}$ and $\mathscr{R}(s)=\{\mathrm{TAA}, \mathrm{ATAA}, \mathrm{AATAA}\}$:

$$\min_{w\in\mathscr{L}(s)}\{|w|\} = \min_{w\in\mathscr{R}(s)}\{|w|\} = |\mathrm{AAT}| = |\mathrm{TAA}| = 3 > 2 = \lfloor |s|/2 \rfloor.$$

**Remark 5.** If $u$ is superprimitive then $\mathscr{L}(u) \cap \mathscr{R}(u) = \{u\}$.

The proof follows by contradiction:

Assume that there exists a string $v \neq u$ in $\mathscr{L}(u) \cap \mathscr{R}(u)$.

Then $v$ is both prefix of $u$ ($v \in \mathscr{L}(u)$) and suffix of $u$ ($v \in \mathscr{R}(u)$). From Remark 4, it follows $|v| > \lfloor |u|/2 \rfloor$, therefore $u$ is a-covered by $v$. It contradicts $u$ is superprimitive.

### *3.1. Normal forms of a quasiperiodic string*

The definition of the set $\mathscr{L}(u)$ for a given string $u$ will enable us to build all possible superpositions of $u$ with itself. Step by step, we can consider that all strings obtained by two successive superpositions of $u$ can be written in the form $u'.u''.u$ with $u'$, $u'' \in \mathscr{L}(u)$. We will denote by $\mathscr{S}_2(s)$ the set of all strings obtained by two successive superpositions of $u$ (that is the language $\mathscr{L}(u).\mathscr{L}(u).u$ itself).

For example, $s=\mathrm{ATA}$, $\mathscr{B}(s)=\{\varepsilon, \mathrm{A}\}$, $\mathscr{L}(s)=\{\mathrm{AT}, \mathrm{ATA}\}$,

$$\begin{aligned}\mathscr{S}_2(s) &= \{\mathrm{AT.AT.ATA}, \mathrm{AT.ATA.ATA}, \mathrm{ATA.AT.ATA}, \mathrm{ATA.ATA.ATA}\} \\ &= \{\mathrm{ATATATA}, \mathrm{ATATAATA}, \mathrm{ATAATATA}, \mathrm{ATAATAATA}\}.\end{aligned}$$

Given a quasiperiodic string $x$ and its cover $u$, it can be interesting to obtain a factorization of $x$ into words of $\mathscr{L}(u)$, but if we want a unique factorization, we have to answer the following question: When is $\mathscr{L}(u)$ a code?

**Proposition 6.** *If $u$ is periodic then $\mathscr{L}(u)$ and $\mathscr{R}(u)$ are not codes.*

Let $s = \mathrm{ATAATAA}$, then $\mathscr{B}(s) = \{\varepsilon, \mathrm{A}, \mathrm{ATAA}\}$, $\mathscr{L}(s) = \{\mathrm{ATA}, \mathrm{ATAATA}, s\}$ and $\mathscr{R}(s) = \{\mathrm{TAA}, \mathrm{TAATAA}, s\}$ which are obviously not codes since $\mathrm{ATAATA} = (\mathrm{ATA})^2$ and $\mathrm{TAATAA} = (\mathrm{TAA})^2$.

**Proof.** Given a periodic string $w = (uv)^k u$ for $k > 1$, we have $\mathrm{Border}(w) = (uv)^{k-1}u$.

Therefore, $\mathscr{B}(w) \supseteq \{(uv)^i u \mid 0 \leqslant i < k\} = \{u(vu)^i \mid 0 \leqslant i < k\}$.

We have $\mathscr{L}(w) \supseteq \{(uv)^j \mid 1 \leqslant j \leqslant k\}$ and since $k > 1$, $uv$ and $(uv)^2$ are elements of $\mathscr{L}(w)$, therefore $\mathscr{L}(w)$ is not a code. □

If $\mathscr{L}(u)$ or $\mathscr{R}(u)$ is a code then $u$ is not periodic.

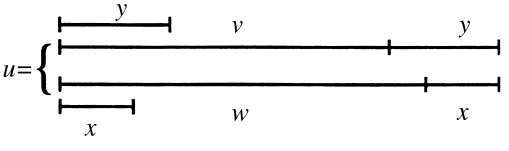
**Proposition 7.** *If $u$ is superprimitive then $\mathscr{L}(u)$ and $\mathscr{R}(u)$ are codes.*

**Proof.** A prefix (resp. suffix) set $S$ is a set of words such that given $v, w$ in $S$, if $v$ is a prefix (resp. suffix) of $w$ then $v = w$. Prefix and suffix sets are well-known to be codes [5].

We will prove that if $u$ is superprimitive then $\mathscr{L}(u)$ (resp. $\mathscr{R}(u)$) is a suffix (resp. prefix) set and therefore prove that if $u$ is superprimitive then $\mathscr{L}(u)$ (resp. $\mathscr{R}(u)$) is a code. □

Given two words $v$ and $w$ in $\mathscr{L}(u)$ such that $v$ is a suffix of $w$. We will prove that if we assume that $v \neq w$ it contradicts $u$ is superprimitive.

Since $v$ and $w$ are elements of $\mathscr{L}(u)$, there exist two words $x$ and $y$ such that $vy = u$ and $wx = u$ and moreover $|w| - |v| = (|u| - |x|) - (|u| - |y|) = |y| - |x|$.




Since $v$ is an element of $\mathscr{L}(u)$, $v$ is a prefix of $u$ and a suffix of $w$ and moreover a border of $w$. From Fact 1, it follows that $|w| - |v|$ is a period of $w$.

So, there exists an occurence of $y$ ending in every interval $]k, k + (|w| - |v|)]$ for $|y| \leqslant k \leqslant |w| - (|w| - |v|)$.

It follows that $y$ a-covers $u_1 \ldots u_j$ with $|v| < j \leqslant |w|$ $(k = |v|)$.

Therefore, $y$ a-covers $u_1 \ldots u_j$ with $|v| \leqslant j \leqslant |w|$ and $u_{|v|+1} \ldots u_{|u|} = y$, that is $y$ a-covers $u$ which contradicts the hypothesis $u$ is superprimitive.

If $u$ is superprimitive then $\mathscr{L}(u)$ is a suffix set and moreover a code.

Similarly, if $u$ is superprimitive then $\mathscr{R}(u)$ is a prefix set and moreover a code.

The following proposition gives a broader approach to quasiperiodic strings.

**Proposition 8** (Left and right normal form of a quasiperiodic string). *Let $v$ be a string that a-covers $x$. If $v$ is superprimitive then there exists a unique factorization into words of $\mathscr{L}(v)$ $v = u_1.u_2 \dots u_k$ with $u_i \in \mathscr{L}(v)$ for $1 \leqslant i \leqslant k$. This factorization will be named the left normal form of $v$ and will be denoted by* $\mathrm{LNF}_v(x)$.

*If $v$ is superprimitive then there exists a unique factorization into words of $\mathscr{R}(v)$ $v = w_1.w_2 \dots w_k$ with $w_i \in \mathscr{R}(v)$ for $1 \leqslant i \leqslant k$. This factorization will be named the right normal form of $v$ and will be denoted by* $\mathrm{RNF}_v(x)$.

**Proof.** Definition 2 insures the existence, and Proposition 7 proves the unicity. □

For example, $S = \mathrm{AATAATAAATAATAAAATAATAATAA}$ is a-covered by $s = \mathrm{AATAA}$ which is superprimitive.

Therefore it can be written in the unique forms:

$$\mathrm{LNF}_{\mathrm{AATAA}}(S) = \mathrm{AAT.AATA.AAT.AATAA.AAT.AAT.AATAA},$$

$$\mathrm{RNF}_{\mathrm{AATAA}}(S) = \mathrm{AATAA.TAA.ATAA.TAA.AATAA.TAA.TAA}.$$

**Fact 9.** *There exist strings $x$ and $v$ such that $x$ is a-covered by $v$, $v$ is quasiperiodic and there exists a unique factorization into words of $\mathscr{L}(v)$ (resp. words of $\mathscr{R}(v)$).*

For example, consider the Fibonacci words over the binary alphabet $\{\mathrm{A}, \mathrm{T}\}$ (we recall that $F_1 = \mathrm{T}$, $F_2 = \mathrm{A}$ and $F_{i+2} = F_{i+1}F_i$ for $i \geqslant 1$).
$F_8 = \mathrm{ATAATATAATAATATAATATA}$ is a-covered by $F_6 = \mathrm{ATAATATA}$,
$\mathscr{B}(F_6) = \{\varepsilon, \mathrm{A}, \mathrm{ATA}\}$,
$\mathscr{L}(F_6) = \{\mathrm{ATAAT}, \mathrm{ATAATAT}, F_6\}$ and
$\mathscr{R}(F_6) = \{\mathrm{ATATA}, \mathrm{TAATATA}, F_6\}$.
$F_6$ is quasiperiodic (a-covered by $F_4 = \mathrm{ATA}$) but there exists a unique form
$F_8 = \mathrm{ATAATATA.ATAAT.ATAATATA} \in (\mathscr{L}(F_6))^{\star}$;
and a unique form $F_8 = \mathrm{ATAATATA.ATAATATA.ATATA} \in (\mathscr{R}(F_6))^{\star}$.

These propositions and facts lead to an important result:

**Theorem 10.** *Given a quasiperiodic string $w$ over an alphabet $\mathscr{A}$.*
*There exists a unique factorization of $w$ into roots of its shortest cover.*

The proof is the direct consequence of Proposition 8.

### *3.2. Determining normal forms*

We present an algorithm to determine the right normal form, which operates from left to right. The left normal form operates on $\tilde{x}$ and $\widetilde{L(u)}$ using the same methodology (we will use $x = \mathrm{AATAATAAATAATAAAATAATAATAA}$ to illustrate the main steps of the algorithm).

The main steps of the algorithm are as follows:

(1) Compute the shortest cover $u$ of $x$. This can be done in $\mathrm{O}(|x|)$ time using [2, 7, 15].
$u = \mathrm{AATAA}$

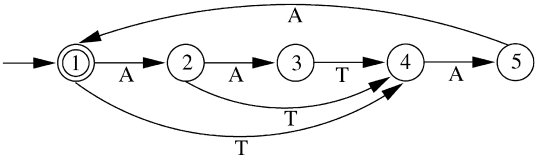



Fig. 4. The minimal automaton recognizing $(\mathscr{R}(\mathrm{AATAA}))^*$.

(2) Construction of the set $\mathscr{R}(u)$. This can done in $O(|u|)$ time by computing the borders of $u$ using the failure function of Knuth-Morris-Pratt's algorithm [14]. $\mathscr{R}(u) = \{\mathrm{TAA}, \mathrm{ATAA}, \mathrm{AATAA}\}$.
(3) Construct a Deterministic Finite Automaton $M_u$ that accepts the language $(\mathscr{R}(u))^*$. The construction of $M_u$ requires $O(|u| \log |\mathscr{A}|)$ time adapting [6, 9] (see Fig. 4).
(4) Decompose the string $x$ using the automaton $M_u$. This requires $|x|$ steps, for following the transitions of the automaton; a factor of the normal form is obtained each time that a final state is reached:

$$\mathrm{RNF}_u(x) = \mathrm{AATAA.TAA.ATAA.TAA.AATAA.TAA.TAA.}$$

**Proposition 11.** *The normal form of a quasiperiodic string of length n can be computed in* $O(n)$ *time.*

### 3.3. Normal forms of a covered string

We present now the left and right normal forms of covered strings, which is less intuitive than a-covered strings, since we are considering seeds, instead of covers. Therefore, the left (right) normal form of quasiperiodic strings can be extended by adding a left and right context to adapt to covered string. This idea leads to the following proposition:

**Proposition 12** (Left and right normal form of a covered string). *Let v is a seed of the string x. If v is superprimitive then there exists a unique form*

$$\mathrm{LNF}_v(x) = \mathrm{PREF}_v(x).Q_v(x).\mathrm{SUFF}_v(x),$$

*where* $Q_v(x)$ *is the largest* (*quasiperiodic*) *factor of x a-covered by v,* $\mathrm{PREF}_v(x)$ *is a suffix of an element of* $\mathscr{L}(v)$ *and* $\mathrm{SUFF}_v(x)$ *is a prefix of an element of* $\mathscr{R}(v)$.

*If v is superprimitive then there exists a unique form*

$$\mathrm{RNF}_v(x) = \mathrm{PREF}'_v(x).Q'_v(x).\mathrm{SUFF}'_v(x),$$

*where* $Q'_v(x)$ *is the largest* (*quasiperiodic*) *factor of x a-covered by v,* $\mathrm{PREF}'_v(x)$ *is a suffix of an element of* $\mathscr{L}(v)$ *and* $\mathrm{SUFF}'_v(x)$ *is a prefix of an element of* $\mathscr{R}(v)$.

**Proof.** Since an extension of the covered string $x$ is quasiperiodic (and a-covered by $v$), the form exists. The unicity is due to the maximality of the quasiperiodic factor a-covered by $v$. □

**Fact 13.** *From the above definitions, we have the following equalities:*

$$\text{PREF}_v(x) = \text{PREF}'_v(x),$$

$$Q_v(x) = Q'_v(x),$$

$$\text{SUFF}_v(x) = \text{SUFF}'_v(x).$$

*Note that $Q_v(x) = Q'_v(x)$ but their factorizations may differ.*

For example, $s = \text{AATAA}$ is a seed of $S = \text{ATAAATAAATAATAAAATA}$ and $s$ is superprimitive therefore there exist a unique left normal form and a unique right normal form:

$$\text{LNF}_s(S) = \text{ATA.AATA.AAT.AATAA.AATA},$$

$$\text{RNF}_s(S) = \text{ATA.AATAA.ATAA.TAA.AATA},$$

where

$$\text{PREF}_v(x) = \text{PREF}'_v(x) = \text{ATA},$$

$$Q_v(x) = \text{AATA.AAT.AATAA} = \text{AATAA.ATAA.TAA} = Q'_v(x),$$

$$\text{SUFF}_v(x) = \text{SUFF}'_v(x) = \text{AATA},$$

## 4. Concatenation of covered strings

**Proposition 14.** *Let $v$ be superprimitive and a seed of both strings $x$ and $y$. The string $v$ is a seed of $xy$ if and only if there exist $z_1, z_2 \in \mathscr{B}(v)$ such that $z_1.\text{SUFF}_v(x).\text{PREF}_v(y).z_2$ is a-covered by $v$ or $\text{SUFF}_v(x).\text{PREF}_v(y) = \varepsilon$.*

**Proof.** A straighforward proof is based the facts that $Q_v(x)$ and $Q_v(y)$ are both a-covered by $v$ and the definition of $\text{SUFF}_v(x)$ and $\text{PREF}_v(y)$. □

For example, consider $S = \text{ATAAATAAATAATAAAATA}$, $s = \text{AATAA}$ is superprimitive and is a seed of $S$. We have $\mathscr{B}(s) = \{\varepsilon, \text{A}, \text{AA}\}$,

$$\mathscr{L}(s) = \{\text{AAT}, \text{AATA}, s\}, \quad \text{LNF}_s(S) = \text{ATA.AATA.AAT.AATAA.AATA},$$

$$\mathscr{R}(s) = \{\text{TAA}, \text{ATAA}, s\}, \quad \text{RNF}_s(S) = \text{ATA.AATAA.ATAA.TAA.AATA},$$

$$\text{PREF}_s(S) = \text{ATA} \quad \text{and} \quad \text{SUFF}_s(S) = \text{AATA}.$$

Thus, $\text{SUFF}_s(S).\text{PREF}_s(S) = \text{AATAAATA}$ and $\exists \varepsilon, \text{A} \in \mathscr{B}(s) \mid \varepsilon.\text{SUFF}_s(S).\text{PREF}_s(S)$. $\text{A} = \text{AATAAATAA}$ is a-covered by $s$. Therefore we can conclude that $S^k$ is covered by $s$ for $k \geqslant 1$.

On the other hand, consider $S = \mathrm{TAAATAAATAATAAAATA}$. We have

$$\mathrm{LNF}_s(S) = \mathrm{TA.AATA.AAT.AATAA.AATA} \quad \text{with } \mathrm{PREF}_s(S) = \mathrm{TA},$$

$$\mathrm{RNF}_s(S) = \mathrm{TA.AATAA.ATAA.TAA.AATA} \quad \text{and} \quad \mathrm{SUFF}_s(S) = \mathrm{AATA}.$$

Now we have $\mathrm{SUFF}_s(S).\mathrm{PREF}_s(S) = \mathrm{AATATA}$ and $\forall z_1, z_2 \in \mathscr{B}(s), z_1.\mathrm{SUFF}_s(S).\mathrm{PREF}_s(S).z_2 = z_1.\mathrm{AATATA}.z_2$ is not a-covered by $s = \mathrm{AATAA}$. Therefore $S^k$ is not covered by $s$ for $k \geqslant 2$.

**Fact 15.** *Let $u$, $x$ et $y$ be words of $\mathscr{A}^*$ and $k$ be a non-negative integer.*

(1) *If $u$ is a cover of $x$ then $u$ is a cover of $x^k$.*

$$(\mathrm{PREF}_u(x) = \mathrm{SUFF}_u(x) = \varepsilon).$$

(2) *If $u$ is a seed of $x$ and $u$ is a cover of $b_1.\mathrm{SUFF}_u(x).\mathrm{PREF}_u(x).b_2$ for some $b_1, b_2 \in \mathscr{B}(u)$ then $u$ is a seed of $x^k$.*

(3) *If $u$ is a cover of $x$ and $y$ then $u$ is a cover of $xy$.*

$$(\mathrm{PREF}_u(x) = \mathrm{SUFF}_u(x) = \mathrm{PREF}_u(y) = \mathrm{SUFF}_u(y) = \varepsilon).$$

(4) *If $u$ is a cover of a left extension of $x$ and a cover of $y$ then $u$ is a cover of a left extension of $xy$ :*

$$(\mathrm{SUFF}_u(x) = \mathrm{PREF}_u(y) = \mathrm{SUFF}_u(y) = \varepsilon).$$

(5) *If $u$ is a cover of $x$ and $u$ is a cover of a right extension of $y$ then $u$ is a cover of a right extension of $xy$.*

$$(\mathrm{PREF}_u(x) = \mathrm{SUFF}_u(x) = \mathrm{PREF}_u(y) = \varepsilon).$$

(6) *If $u$ is a cover of a left extension of $x$ and a cover of a right extension of $y$ then $u$ is a seed of $xy$.*

## 5. Conclusion

Here we presented, right and left normal forms of quasiperiodic strings together with a linear algorithm for computing such forms. Furthermore, normal forms of covered strings were presented. Additionally, we showed the criteria required for preserving the seeds of covered strings under concatenation. The key open question is whether these criteria lead to a linear algorithm for computing all seeds of a given string.

## 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, 5th ACM-SIAM Ann. Symp. on Discrete Algorithms, Arlington, VA, 1994, pp. 501–510.
[5] J. Berstel, D. Perrin, Theory of Codes, Academic Press, Orlando, FL, 1985.
[6] A. Blumer, J. Blumer, A. Ehrenfeucht, D. Haussler, M.T. Chen, J. Seiferas, The smallest automaton recognizing the subwords of a text, Theoret. Comput. Sci. 40 (1) (1985) 31–55.
[7] D. Breslauer, An on-line string superprimitivity test, Inform. Process. Lett. 44 (6) (1992) 345–347.
[8] D. Breslauer, Testing string superprimitivity in parallel, Report CUCS-053-92, Computer Science Department, Columbia University, NY, 1992.
[9] M. Crochemore, E. Rytter, Text Algorithms, Oxford University Press, Oxford, 1994.
[10] A.M. Duval, W.M. Smyth, Covering a circular string with substrings of fixed length, Internat. J. Found. Comput. Sci. (1998), (to appear).
[11] 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.
[12] C.S. Iliopoulos, L. Mouchard, An o($n$log$n$) algorithm for computing all maximal quasiperiodicities in strings, in: C.S. Calude, M.J. Dinneen (Eds.), Combinatorics, Computations and Logic. Proc. DMTCS'99 and CATS'99, (vol. 21) Lecture Notes in Computer Science, Auckland, New-Zealand, Springer-Verlag, Singapore, 1999, pp. 262–272.
[13] C.S. Iliopoulos, K. Park, An optimal o(loglog$n$)-time algorithm fir parallel superprimitivity testing, J. Korea Inform. Sci. Soc. 21 (8) (1994) 1400–1404.
[14] D.E. Knuth, J.H. Morris Jr., V.R. Pratt, Fast pattern matching in strings, SIAM J. Comput. 6 (1) (1977) 323–350.
[15] D. Moore, W.F. Smyth, An optimal algorithm to compute all the covers of a string, Inform. Process. Lett. 50 (5) (1994) 239–246.