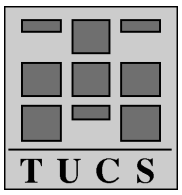
On Morphisms Preserving Primitive Words

Victor Mitrana

Faculty of Mathematics, University of Bucharest
Str. Academiei 14, 70109, Bucharest, ROMANIA



Turku Centre for Computer Science
TUCS Technical Report No 69
November 1996

ISBN 951-650-895-2
ISSN 1239-1891

Abstract

A word is called primitive if it cannot be expressed as the power of another word. Morphisms preserving primitive words are investigated. Similarly to the word case, each square-free morphism is a primitive morphism but the converse does not hold. A precise characterization of primitive morphisms is provided in terms of pure codes. An easily testable characterization is given for uniform morphisms over the binary alphabet.

TUCS Research Group
Mathematical Structures of Computer Science

1 Introduction

Primitive words play an important role in algebraic coding [11] as well as in combinatorial theory of words [8].

There has been conjectured [4] that the set of all primitive words over a given alphabet is not context-free. However, this language satisfies different necessary conditions for context-free languages as iteration teorems, Interchange Lemma, etc. Hopfully, this conjecture requires new methods based on the structure of context-free languages and perhaps will lead to sharper necessary conditions for languages to be context-free. To study objects with a given property, it is usually helpful to analyse the operators on these objects that preserve the given property.

The present paper is dealing with the primitive morphisms, namely those morphisms that preserve the primitive words. Some results on primitive words have been reported in [12], where $M0L$ schemes are introduced and characterized by their preserving properties. In the aforementioned paper, one proves that whenever the set of letter images by a given morphism is a pure code, the morphism preserves the primitive words. More recently [6], primitive $D0L$ systems ($D0L$ systems generating only primitive words) have been characterized by means of cylindrical languages.

Since each primitive morphism $h : A^+ \longrightarrow B^+$ will turn out to be injective, hence $h(A)$ is a code, the theory of codes furnishes a good framework to investigate the primitive morphisms.

We shall characterize the primitive morphisms in terms of pure codes. This characterization has a series of important consequences concerning the decidability of the primitivity of a morphism. Another characterization of primitive morphisms is provided but only in a very particular case; it leads to efficient algorithms for testing the primitivity of morphisms.

Our investigation on primitive morphisms will be also related to the study of square-free morphisms, i.e. morphisms preserving the square-free words, which was initiated by Thue [14]. These morphisms are powerful tools to prove properties on square-free words and are used in particular to generate infinite sequences without repetition. Obviously, each square-free word is a primitive word but the converse does not hold. Similarly, we shall provide a straightforward proof of the fact that each square-free morphism is a primitive morphism but not conversely. Furthermore, weaker conditions than those sufficient for square-free morphisms are sufficient for primitive morphisms.

Finally, some open problems are formulated.

2 Definitions and basic properties

An alphabet is always a finite, nonempty set. Given an alphabet A, the free semigroup generated by A is denoted by A^+. By adding of the empty word ε we get the monoid A^*. The length of a word w in A^* is denoted by $|w|$. If a word w is equal to uxv, for some words $u, v \in A^*$, $x \in A^+$, x is said to be subword of w; it is a prefix of w if $u = \varepsilon$ and a suffix of w if $v = \varepsilon$. Denote by $Pref(w), Sub(w), Suf(w)$ the sets of all prefixes, subwords, suffixes of the word w, respectively. For a finite set A, we denote by $card(A)$ the cardinality of A.

We say that the words x and y are *conjugates*, if they are obtainable from each other by a cyclic permutation, i. e. there are u, v such that $x = uv$ and $y = vu$.

A word x is *primitive* if $x = u^k$ implies $k = 1$. A morphism $h : A^* \longrightarrow B^*$ is called *k-primitive* (for an integer $k \geq 1$) if for every primitive word $x \in A^*$, with $|x| \leq k$, $h(x)$ is primitive. A morphism is *primitive* if it is k-primitive, for all $k \geq 1$.

Note the following remarks quite useful in the sequel.

1. If a word x is primitive, then all its conjugates are primitive as well.
2. If x, y are conjugates and $x = u^k$, for some u and $k \geq 1$, then there exists v with $|u| = |v|$ and $y = v^k$. Moreover, if u is primitive, then v is primitive.

A morphism $h : A^* \longrightarrow B^*$ is *uniform* if $|h(a)| = |h(b)|$, for any $a, b \in A$.

The remaining part of this section presents some simple and mostly well-known results that will be used throughtout the paper. For more details, the reader may consult [8], [13].

Lemma 1 . *Two words u and v commute (i. e. $uv = vu$) if and only if they are powers of a common word. Consequently, for any two nonempty commuting words u, v the word uv is not primitive.*

Lemma 2 . *Each word w can be uniquely represented in the form $w = u^k$, with $k \geq 1$ and u primitive.*

The word u is called the *primitive root* of w.

Lemma 3 . *The equation $uv = vx$ has the solutions $v = (\alpha\beta)^k\alpha$, $k \geq 0$, $u = \alpha\beta$, $v = \beta\alpha$.*

Theorem 1 . [5] *(Periodicity Theorem) The words u and v are powers of a same word if, and only if, exist n and m such that u^n and v^m have a common prefix of length $|u| + |v| - gcd(|u|, |v|)$. (gcd means the greatest common divisor)*

3 Characterizations of primitive morphisms

A language $X \subseteq A^+$ is a *code* if X^* is a free submonoid of A^*, i.e. all words of X^+ have an unique factorization in terms of elements of X. A code $X \subseteq A^*$ is called *pure* if for any $x \in X^*$ the primitive root of x belongs to X^*.

Theorem 2 . *Let h be a morphism from A^+ to B^+. Then, h is primitive if and only if $h(A)$ is a pure code.*

Proof. The "if" part is proved in [12]. For the sake of completeness we recall it below. Assume that $h(A)$ is a pure code and let x be a primitive word in A^+ and $h(x) = u^k$, for some primitive word u. Since $h(A)$ is pure by assumption, it follows that $u \in (h(A))^*$, that is $u = h(y)$, for some $y \in A^+$. Therefore, $h(x) = h(y^k)$ that implies $x = y^k$ because h, being a code, is injective. Consequently, $k = 1$ which concludes the "if" part.

Let us assume now that h is primitive. We are going to prove that h is injective, hence $h(A)$ is a code. Take two different words, say x and y from A^+ and assume that $h(x) = h(y)$. Without loss of generality we may assume that $|x| \leq |y|$. We claim that at least one of the words xy and xyy is primitive. Assume the contrary, i. e.

$$\begin{aligned} xy &= u^m,\ m \geq 2 \\ xyy &= v^n,\ n \geq 2 \end{aligned}$$

Observe that $m, n \geq 3$. Indeed, if $m = 2$ or $n = 2$, then it follows that either $x = y$ or x is a proper subword of y, hence $h(x) \neq h(y)$. Consequently, we may assume that $n, m \geq 3$. The following inequalities hold.

$$\begin{aligned} |u| &\leq \frac{|x| + |y|}{3} \\ |v| &\leq \frac{2|y| + |x|}{3} \end{aligned}$$

It follows that

$$|u| + |v| \leq |y| + \frac{2|x|}{3} \leq |x| + |y|$$

For u^m and v^n have a common prefix of length $|u| + |v|$, by Periodicity Theorem, it follows that u and v are powers of a common word, say w. Furthermore, x and y are powers of w that contradicts the equality $h(x) = h(y)$. In conclusion, at least one of xy and xyy is primitive.

But $h(xy) = (h(x))^2$ and $h(xyy) = (h(x))^3$, which implies that h does not preserve primitive words. Consequently, h is injective.

Let us assume that $h(x) = u^k \in (h(A))^*$, $k \geq 2$, u primitive. Since h preserves primitive words we have $x = v^q$, for some primitive word v and $q \geq 2$, hence $u^k = (h(v))^q$.

From the primitivity of h and the uniqueness of the representation of u^k as a power of a primitive word we infer that u is equal to $h(v)$, hence in $(h(A))^*$ which concludes the proof. □

This theorem has a series of important consequences.

Corollary 1 . *The following problems are decidable:*

1. Is a uniform morphism primitive?

2. Is a morphism, over a binary alphabet, primitive?

Proof. 1. Let $h : A^+ \longrightarrow B^+$ be a uniform morphism. Obviously, one can check whether a uniform morphism is a code (it suffices to be injective on the letters in his domain). Of course, if it is not injective, then it is not primitive. Now, $(h(A))^*$ is a recognizable language. By Theorem 3.1 in [10], the language X^* is a star free language if and only if X is a pure code. Furthermore, by Schützenberger's theorem, a recognizable language is star free if and only if its syntactical monoid is aperiodic, and the proof is complete (see [9]).

2. Let $h : \{a, b\}^+ \longrightarrow B^+$ be a given morphism. In order to be primitive, h has to satisfy $h(ab)$ primitive. By Theorem 1.26 in [13], it follows that $\{h(a), h(b)\}$ is a code. Now, the proof goes on as in the previous case. □

However, the next result provides a very easily testable characterization of the uniform morphisms over a binary alphabet. To this end, we need the following lemma.

Lemma 4 . *If the following conditions are satisfied*

$$(i) \quad y'z = zy \text{ and } yz = z'y',$$
$$(ii) \quad |y| = |y'| \text{ and } |z| = |z'|,$$

then $y = y'$.

Proof. Clearly, if $|y| \leq |z|$, then $y = y'$ holds, since both y and y' are prefixes of z, of the same length. Assume that $|y| > |z|$. From $yz = z'y'$ we have that $y' = x'z$, for some x'. Analogously, $y = xz$ follows from $y'z = zy$. Consequently, $x'z = zx$ and $xz = z'x'$. Moreover, by the second hypothesis, we have $|x| = |x'|$ and $|x| < |y|$. If $|x| \leq |z|$, then $x = x'$, hence $y = y'$. Otherwise, resume the reasoning for the new equalities and so on. □

Theorem 3 . *Let h be a uniform morphism from $\{a, b\}$ into an alphabet V. Then, h is primitive if, and only if, it is 2-primitive.*

Proof. The "only if" part is trivial. Assume that h preserves all words of length at most two and take $x = x_1x_2 \dots x_n$, $x_i \in \{a,b\}$, $1 \leq i \leq n$, a primitive word of length at least three. Furthermore, suppose that $h(x)$ is not primitive, that is $h(x) = u^k$ for some primitive word $u \in \{a,b\}^+$ and $k > 1$. Without loss of generality, we may assume that $x_1 = x_n = a$ and $x_{n-1} = b$.

Note that h must be injective, hence $|u| \neq |h(a)| = p$. We distinguish two cases.

• *Case 1.* $|u| < p$

Consider that $h(a) = h(x_1) = u^iy$, $|y| < |u|$. Note that y cannot be the empty word ε, otherwise $h(a)$ is not primitive. Therefore, y is a proper prefix of u. On the other hand, $h(x_n) = h(a) = y'u^i$, for some y'. Since $|y| = |y'|$ and $u^iy = y'u^i$ it follows that $y = y'$, hence u^i and y are power of a common word. This implies that $h(a)$ is not primitive, contradiction.

• *Case 2.* $|u| > p$

Consider $u = h(x_1x_2 \dots x_i)y$, for some $i > 1$ and $y \in V^*$, $|y| < p$. Obviously, y cannot be ε. Indeed, if $y = \varepsilon$, then $x = (x_1x_2 \dots x_i)^k$, contradiction. We conclude further that $h(a) = zy$, for some z. Since $h(x_{i+1}) = yz$, we deduce that $x_{i+1} = b$.

Let q be the biggest index such that $x_q = a$ (such an index always exists). Note that q cannot be smaller than $i-1$. Indeed, $q < i-1$ contradicts our hypothesis $x_{n-1} = b$. Therefore, two cases should be considered.

• *Case 2. 1.* $q = i$. We infer that $h(x_i) = h(a) = y'z$ which implies $h(x_{n-1}) = h(b) = z'y'$. We get that $zy = y'z$ and $yz = z'y'$ with $|y| = |y'|$ and $|z| = |z'|$. By Lemma 4 it follows that $y = y'$, hence $h(a)$ is not primitive, contradiction.

• *Case 2. 2.* $q = i-1$. We infer that $h(x_{n-1}) = h(b) = z'y$ which implies $h(x_{i-1}) = h(a) = y'z'$. We get that $zy = y'z'$ and $yz = z'y$ with $|y| = |y'|$ and $|z| = |z'|$. By Lemma 4 it follows that $z = z'$, which contradicts the primitivity of $h(b)$.

For all possibilities are covered by the cases considered above we conclude that the morphism h is primitive. □

Note the optimality of the bound in the above theorem as shown by the morphism $h(a) = aba$, $h(b) = bab$.

Corollary 2 . *The primitivity of a given uniform morphism over a two letter alphabet is decidable in linear time.*

We do not know whether the primitivity of an arbitrary morphism can be decided by examining all words of length smaller than a given bound. Anyway, if such an algorithm exists, the bound depends esentially on the letter images by the morphism.

Theorem 4 . *For any integer n there is a binary morphism which is n-primitive but not primitive.*

Proof. Let n be a given integer. Define the morphism $h : \{a,b\}^+ \longrightarrow \{a,b\}^+$ by:

$$h(a) = aba, \qquad h(b) = (baa)^{n-1}b$$

The word a^nb is a primitive word of length $n+1$ but

$$h(a^nb) = (aba)^{n-1}ab(aba)^{n-1}ab$$

is not primitive.

On the other hand, h is n-primitive. In order to prove this claim, we consider a primitive word of minimal length, say $x = a^{i_1}b^{j_1}a^{i_2}b^{j_2}\ldots a^{i_r}b^{j_r}$, such that $h(x) = u^k$, for some primitive word u and $k \geq 2$. Assume that

$$\sum_{s=1}^{r} i_s + \sum_{s=1}^{r} j_s \leq n.$$

We cannot have $u = h(v)$ for any prefix v of x. Indeed, if $k = 2$ the aforementioned equality contradicts the primitivity of x; if $k > 2$, then the choice of x is contradicted. Let v be the minimal prefix of x such that $u \in Pref(h(v))$. We infer that v cannot end by a and be followed by a in x. Indeed, if it were the case, then either both ab and aa, or both ab and ba, would be prefixes of u.

Assume that v ends by a and it is followed by b in x. Then, u can be written as $u = h(a^{i_1}b^{j_1}a^{i_2}b^{j_2}\ldots a^{i_s-1})ab$ and $u^{k-1} = ah(b^{j_s}a^{j_{s+1}}\ldots b^{j_r})$. Note that $ah(b) = (aba)^{i_1-1}aba(aba)^{n-i_1-1}ab$ and $i_1 < n$. The case $s = 1$ is impossible because it implies that u starts by aa. The case $s > 1$ implies that u is not a prefix of u^{k-1}, contradiction.

Assume now that v ends by b. If $u = (aba)^{i_1}(baa)^q ba$, for some $0 \leq q \leq n-2$, then $h(v) = u(aba)^{n-q-2}ab$. It follows that $n-q-2 \leq i_1 - 1$ and v has to be followed by at least $q+1$ a's in x. Consequently, $|x| > n$, contradiction.

Let us finally suppose that $(aba)^{i_1}(baa)^{n-2}ba$ is a prefix of u. We have that $h(v) = u(aba)^t ab$, for some t. Therefore, $t \leq i_1 - 1$ and v has to be followed by at least $n-1$ a's in x. Thus, we get again $|x| > n$. Now the proof is complete. □

4 Square-freeness and primitivity

A word is said *square-free* if all its subwords are primitive words. Clearly, every square-free word is primitive but vice versa does not hold. Accordingly,

a k-square-free morphism is a morphism preserving all square-free words of length at most k. A square-free morphism is a k-square-free morphism for any $k \geq 1$. They are also known as Thue morphisms. Of course, there are primitive morphisms that are not square-free. For instance, consider the morphisms

$$h : \{a,b,c\}^* \longrightarrow \{a,b,c\}^* \text{ defined by } \begin{array}{rcl} h(a) & = & ac^2 \\ h(b) & = & bc^2 \\ h(c) & = & abc^2 \end{array}$$

Obviously, the above morphism is not square-free but one can easily show that it is primitive. Let us suppose that $h(x_1x_2\ldots x_n) = u^k$, for some $k \geq 2$, $x_i \in \{a,b,c\}$, $1 \leq i \leq n$. For u has to finish by c^2 it follows that $u = h(x_1x_2\ldots x_i) = h(x_{i+1}x_{i+2}\ldots x_t)$, for some $1 \leq i < t \leq n$. Consequently, $t = 2i$ and $x_j = x_{i+j}$, for all $1 \leq j \leq i$. In conclusion, $x_1x_2\ldots x_n$ is not primitive.

As a matter of fact, there are primitive morphisms which are neither overlap-free. The binary morphism defined by $h(a) = abb$, $h(b) = baa$ is primitive (see the next section) but not overlap-free (see [14], cf. also [6]).

A natural question arises: Is any square-free morphism a primitive morphism ? In [7] it is asserted that each square-free code is a pure code. By combining this assertion with Theorem 2, we have that each square-free morphism is a primitive morphism. However, we shall give a straightforward proof that has some other consequences.

Theorem 5 . *Each square-free morphism is primitive.*

Proof. Let $h : A^+ \longrightarrow B^+$ be a square-free morphism.

Fact 1. h is injective. Clearly, $h(a) \neq h(b)$ for all $a \neq b$, $a,b \in A$. Since h is square-free, it follows that for two different letters $a,b \in A$, $Pref(h(a)) \cap Suf(h(b)) = \emptyset$. Assume that $h(a_1a_2\ldots a_n) = h(b_1b_2\ldots b_m)$, $a_i, b_j \in A$, $1 \leq i \leq n$, $1 \leq j \leq m$. If $|h(a_1)| < |h(b_1)|$, then $h(a_1b_1)$ is not square-free, contradiction. Analogously for $|h(a_1)| > |h(b_1)|$. In conclusion, $h(a_1) = h(b_1)$ that is $a_1 = b_1$, and inductively $n = m$ and $a_i = b_i, 1 \leq i \leq n$, which concludes the proof of the fact.

Assume that h is not primitive and take $w = a_1a_2\ldots a_n$, $a_i \in A, 1 \leq i \leq n$, a primitive word of minimal length such that $h(w) = u^k$, for some primitive word u and $k \geq 2$.

Fact 2. There are no $p < k$ and $1 \leq i < n$ such that $h(a_1a_2\ldots a_i) = u^p$. Otherwise, the choice of w would imply that $a_1a_2\ldots a_i = v^t$, with v a primitive word, hence $h(a_1a_2\ldots a_i) = (h(v))^t$. Consequently, $u = h(v)$ and $t = p$ which implies $w = v^k$ (h is injective), contradiction. The proof of the second fact is complete.

We distinguish three main cases.

• *Case 1.* $|u| \leq min\{|h(a_i)|,\ 1 \leq i \leq n\}$. We may suppose that $|h(a_1)| = min\{|h(a_i)|,\ 1 \leq i \leq n\}$ and $a_n \neq a_1$. It follows that $u \in Pref(h(a_1)) \cap Suf(h(a_n))$, contradiction.

• *Case 2.* $min\{|h(a_i)|,\ 1 \leq i \leq n\} < |u| \leq max\{|h(a_i)|,\ 1 \leq i \leq n\}$. Without loss of generality, we may assume that $|h(a_n)| = max\{|h(a_i)|,\ 1 \leq i \leq n\}$ and $a_1 \neq a_n$. Note that it is not assumed any more that $|h(a_1)| = min\{|h(a_i)|, 1 \leq i \leq n\}$.

If $|u| \leq |h(a_1)|$, then $u \in Pref(h(a_1)) \cap Suf(h(a_n))$, contradiction.

Let $u = h(a_1a_2 \ldots a_i)\alpha$, $h(a_{i+1}) = \alpha\rho$. Since $\alpha \in Pref(h(a_{i+1})) \cap Suf(h(a_n))$, we infer that $a_{i+1} = a_n$. Note that u has to be square-free, otherwise $h(a_n)$ would not be square-free. It follows that $h(a_1a_2 \ldots a_i)$ is square-free. On the other hand, all letters $a_1, a_2, \ldots, a_i$ are different than a_n that results in $h(a_1a_2 \ldots a_{i+1})$ is square-free. Therefore, ρ is a prefix of u, so that we may write $u = \rho\eta$, for some η. Moreover, $i + 1 \neq n$.

But, from $h(a_n) = \alpha\rho = \beta\alpha$ we have $\beta = \alpha\theta$ and $\rho = \theta\alpha$. (See Lemma 3 and take into account that $h(a_n)$ is square-free).

The word $a_na_1a_2 \ldots a_ia_n$ is square-free, but

$$h(a_na_1a_2 \ldots a_ia_n) = \alpha\theta\alpha u\rho = \alpha\theta\alpha\rho\eta\rho = \alpha\theta\alpha\theta\alpha\eta\rho,$$

contradiction.

• *Case 3.* $|u| > max\{|h(a_i)|,\ 1 \leq i \leq n\}$. We work under the same hypothesis as in the Case 2.

Let

$$\begin{aligned} u &= h(a_1a_2 \ldots a_i)\alpha, \\ h(a_{i+1}) &= \alpha\beta, \\ u^{k-1} &= \beta h(a_{i+2} \ldots a_n) \end{aligned}$$

By our hypothesis, $a_{i+1} = a_n$. It follows that the following hold:

$$\begin{aligned} \beta &= h(a_1a_2 \ldots a_j)\gamma, \text{ for some } j \leq i \\ h(a_n) &= \delta\alpha, \text{ for some } \delta \end{aligned}$$

But $\gamma \in Suf(h(a_n)) \cap Pref(h(a_{j+1}))$ that results in $a_n = a_{j+1}$ and $h(a_n) = \gamma\eta$, for some η. We deduce that $h(a_1a_2 \ldots a_j)\gamma = \rho\alpha$, hence $h(a_n) = \alpha\rho\alpha$. Furthermore, $h(a_1a_2 \ldots a_j)$ is square-free, hence $h(a_na_1a_2 \ldots a_ja_n)$ is square-free as well (none of $a_i, 1 \leq i \leq j$, is equal to a_n). But, $h(a_na_1a_2 \ldots a_ja_n) = \alpha\rho\alpha h(a_1a_2 \ldots a_j)\gamma\eta = \alpha\rho\alpha\rho\alpha\eta$, contradiction.

Since the cases mentioned above exhaust all possible cases, we conclude that h is primitive. □

An analysis of the previous proof shows that:

Corollary 3 .

1. Each k-square-free morphism is k-primitive.

2. Each 3-square-free morphism over a binary alphabet is primitive.

There have been provided sufficient conditions as well as exact characterizations for square-free morphisms [14], [3]. For a morphism h from A^+ into B^+ let us set $sqc(h)$ the least integer bigger than or equal to $\frac{max\{|h(a)|,a\in A\}-3}{min\{|h(a)|,a\in A\}}$. As consequences of the previous theorem, and following [3], we have:

Corollary 4 .

1. A morphism h is primitive if it is k-square-free with $k = max(3, sqc(h))$.

2. A morphism over a three letter alphabet is primitive if it is 5-square-free.

Note that in [3] one proves the optimality of the above bounds for square-free morphisms. We do not know whether they are optimal for primitive morphisms. In the same paper, the optimal bound for uniform square-free morphisms is 3. With a simple reasoning (some hints are given by the proof of Theorem 3), one can prove that

Theorem 6 . *A uniform morphism is primitive if it is 2-square-free.*

However, we shall prove that weaker conditions than those which imply the square-freeness are sufficient for a morphism to be primitive.

Theorem 7 . *Let h be a morphism from* A^+ *to* B^+ *satisfying the following two conditions:*

1. h preserves all square-free words in A^+ *of length 2;*

2. For any $a, b \in A$ *if* $h(a) \in Sub(h(b))$, *then* $a = b$.

Then, h is primitive.

Proof. The first part of the proof follows the proof in [1]. We claim that if $a, b_1, b_2, \ldots, b_n \in A$, $n \geq 2$, and $h(b_1 b_2 \ldots b_n) = xh(a)y$, $x, y \in B^*$, then there exists $1 \leq j \leq n$ such that

$$\begin{aligned} a &= b_j \\ x &= \begin{cases} \varepsilon, \text{ if } j = 1 \\ h(b_1 b_2 \ldots b_{j-1}, \text{ if } j > 1 \end{cases} \\ y &= \begin{cases} \varepsilon, \text{ if } j = n \\ h(b_{j+1} b_{j+2} \ldots b_n), \text{ if } j < n \end{cases} \end{aligned}$$

Indeed, if $h(a) \in Sub(h(b_1 b_2 \ldots b_n))$, then one of the following cases holds.

1. $h(a) \in Sub(h(b_j))$, for some j, that results in $a = b_j$.

2. $h(b_j) \in Sub(h(a))$, for some j, that results in $a = b_j$.

3. $h(a) \in Sub(h(b_j b_{j+1})) \setminus Sub(h(b_j))$, for some j. It follows that $h(b_j) = uv$, $h(a) = vz$, $h(b_{j+1}) = zw$, $u, v, z, w \in B^*$, $v, z \neq \varepsilon$. But $h(b_j a) = uvvz$ and $h(ab_{j+1}) = vzzw$, which are not square-free words, hence $b_j = a = b_{j+1}$. Therefore, $h(a)$ is a proper subword of $h(a)h(a)$ which implies that $h(a)$ is not primitive, hence $h(ab)$ is not square-free, contradiction.

We will prove now that h is primitive. Assume that $h(x) = h(x_1 x_2 \dots x_n) = u^k$, for some $x_i \in A$, $1 \leq i \leq n$, and $k \geq 2$. Moreover, without loss of generality we may suppose that $|h(x_1)| = min\{|h(x_i)| \ | 1 \leq i \leq n\}$ and $x_1 \neq x_n$.

If $|u| < |h(x_1)|$, then $uu \in Pref(h(x_n x_1))$ that contradicts the first hypothesis.

If $u = h(x_1 x_2 \dots x_i)y$ for some $|y| < |h(x_{i+1})|$, then there exists $i + 1 \leq t \leq n$ such that $h(x_{i+1} x_{x+2} \dots x_t) = yuz = yh(x_1)z'$, for some $z, z' \in B^*$. Consequently, $y = \varepsilon$, hence $u = h(x_1 x_2 \dots x_i)$. Analogously, one proves that the next subword u of $h(x)$ is of the form $h(x_{i+1} x_{i+2} \dots x_q)$, for some q, and so on. It follows that $q = 2i$ and $x_1 x_2 \dots x_i = x_{i+1} x_{i+2} \dots x_q$ and, finally $x = (x_1 x_2 \dots x_i)^k$. □

The following problem appears to be of interest: Can one decide whether or not a primitive morphism is square-free? The answer is affirmative.

Theorem 8 . *It is decidable whether or not a primitive morphism is a square-free morphism.*

Proof. Let $h : A^+ \longrightarrow B^+$ be a primitive morphism. Since h is primitive, it follows that $h(A)$ is a code. The algorithm consists of two steps. First, one verifies whether h preserves all words of length 2. If this is not the case, then h is not square-free. Otherwise, $h(A)$ is an almost circular code [7]. A code X is an almost circular code if, for any non-empty words x, y such that xy is a word of X, $yX^*x \cap X^*$ is either a finitely generated semigroup or the empty set.

At the second step, it suffices to check whether h is q-square-free with

$$q = 4(\sum_{a \in A} |h(a)| - card(A)) + 1.$$

The correctness follows from [7]. □

We finish by pointing out some open problems and further directions of research.

1. Theorem 3 gives a bound for checking the primitivity of a uniform morphism, over a binary alphabet, that does not depend on the morphism. Are there such bounds for uniform morphisms over arbitrary alphabets?

2. Is it true that for any morphism h over a binary alphabet there is a constant C_h such that h is primitive iff it is C_h-primitive? What about the constant given by the maximal length of the letter images by h?

3. Is the monoid of (uniform) primitive endomorphisms of A^+ finitely generated?

We hope to return in a forthcoming paper on the counting of primitive words and morphisms.

References

[1] D. Bean, A. Ehrenfeucht and G. McNulty, Avoidable patterns in strings of symbols, *Pacific J. Math.* 85 (1979), 261–294.

[2] J. Berstel and D. Perrin, *The Theory of Codes*, Academic Press, New York, 1984.

[3] M. Crochemore, Sharp characterizations of squarefree morphisms, *Theoret. Comput. Sci.* 18 (1982), 221–226.

[4] P. Dömösi, S. Horvath and M. Ito, Formal languages and primitive words, *Publ. Math. Debrecen* 42, (3-4) (1993), 315–321.

[5] N. J. Fine and H. S. Wilf, Uniqueness theorem for periodic functions, *Proc. Am. Math. Soc.* 16 (1965), 109–114.

[6] T. Harju, On cyclically overlap-free words in binary alphabets. in *The Book of L* (G. Rozenberg, A. Salomaa eds.) 1986, 123–130.

[7] M. Leconte, A characterization of power-free morphisms, *Technical Rep. LITP*, 84–65.

[8] M. Lothaire, *Combinatoric on Words*, Addisson-Wesley, Reading MA, 1983.

[9] J. E. Pin, *Varieties of Formal Languages* (English translation) North Oxford Academic Publ., 1986.

[10] A. Restivo, Codes and aperiodic languages, *Fachtagunguber Automatentheoric und Formale Sprachen*, LNCS 2, Springer, Berlin, 1973, 175–181.

[11] H. J. Shyr and G. Thierrin, Disjunctive languages and codes, *Proc. FCT 77*, LNCS 56, Springer, Berlin, 1977, 171–176.

[12] H. J. Shyr and G. Thierrin, Codes, languages and M0L schemes, *RAIRO/Theoretical Computer Science*, 11, 4 (1977), 293–301.

[13] H. J. Shyr, *Free Monoids and Languages*, Lect. Notes. Dept. Math., Soochow Univ., Taipei, 1991.

[14] A. Thue, Über unendliche Zeichenreichen, *Kra. Vidensk. Selsk. Skrifter. I. Math.-Nat. Kl.*, Christiana, 7 (1906), 1–22.

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.abo.fi

University of Turku
- **Department of Mathematical Sciences**

Åbo Akademi University
- **Department of Computer Science**
- **Institute for Advanced Management Systems Research**

Turku School of Economics and Business Administration
- **Institute of Information Systems Science**