POLYNOMIAL ALGORITHM FOR FIXED POINTS OF NONTRIVIAL MORPHISMS

ŠTĚPÁN HOLUB

ABSTRACT. A word w is a fixed point of a nontrival morphism h if $w = h(w)$ and h is not the identity on the alphabet of w. The paper presents the first polynomial algorithm deciding whether a given word is such a fixed point. The algorithm also constructs the corresponding morphism, which has the smallest possible number of non-erased letters.

1. Introduction

Fixed points of morphisms are interesting objects in all mathematical structures. In this paper we consider morphisms of free monoids, that is, of finite words with the operation of concatenation. Given a morphism $h : \Sigma^* \to \Sigma^*$, its fixed points were characterized first in [2] and later in a more algebraic manner in [1], where also one-sided infinite fixed points are characterized. Two sided fixed points are studied in [4].

The inverse problem, that is, to determine whether a given word is a fixed point of a morphism, seems to be more difficult. Obviously, each word is a fixed point of the identity, therefore the question asks about morphisms that are not the identity on the input word, in literature often called nontrivial. The importance of the property of being a fixed point of a nontrivial morphism is highlighted in [3] by pointing out a couple of equivalent formulations relevant to the theory of formal languages and the combinatorics on words.

Here we present the first polynomial algorithm deciding whether a given word has the discussed property. Moreover, the algorithm also allows to construct a corresponding morphism which is in a good sense minimal possible.

2. Basic concepts

Let Σ be an alphabet. A word w is a fixed point of a morphism $h : \Sigma^* \to \Sigma^*$ if $h(w) = w$. Following [3], we shall say that w is *morphically imprimitive* if $h(x) \neq x$ for at least one letter occurring in w. Accordingly, a word w is called *morphically primitive* if $h(w) = w$ implies that h is the identity on $\mathrm{alph}(w)$, where $\mathrm{alph}(w)$ denotes the set of letters occurring in w.

By $|w|_x$ we shall denote the number of occurrences of the letter x in w. If L is a (sub)set of letters, we define

$$|w|_L = \sum_{x \in L} |w|_x .$$

We shall now briefly summarize properties of finite fixed points of a morphism h. For proofs and more details consult [1].

Given a morphism $h : \Sigma^* \to \Sigma^*$ we shall say that a letter $a \in \Sigma$ is *mortal* if $h^j(a) = \varepsilon$ for some $j \in \mathbb{N}$. Here ε denotes the empty word, and h^j means j-times repeated application of h. The set of mortal letters of morphism h is denoted by $\mathcal{M}_h$. Second important set of letters, denoted by $\mathcal{E}_h$, is the set of *expanding* letters. A letter b is expanding if $h(b) = xby$ with $xy \in \mathcal{M}_h^*$.

It can be shown that if w is a fixed point of h, then $w \in (\mathcal{M}_h \cup \mathcal{E}_h)^*$. Obviously, $h^i(w) = w$ for each $i \in \mathbb{N}$. A number $t \in \mathbb{N}$ can be chosen such that $h^t(a) = \varepsilon$ for each mortal letter $a \in \mathrm{alph}(w)$. If t is the smallest possible, then it is called the *mortality exponent* of h, and is denoted by $\exp(h)$. One can also see, under the assumption $h(w) = w$, that $h^{\exp(h)}$ is the identity on $\mathrm{alph}(w)$ if and only if h is. Since it is usually convenient to consider the morphism $h^{\exp(h)}$ instead of h, we shall say that h is a *stable* morphism if $\exp(h) = 1$, and limit ourselves to stable morphisms.

If h is a stable morphism, then the equality $h(w) = w$ defines a k-tuple

$$(w_1, w_2, \ldots, w_k)$$

such that $w = w_1 w_2 \ldots w_k$, and for each $i = 1, \ldots, k$ there is a letter $a_i \in \Sigma$ with

$$w_i = h(a_i).$$

In particular, we have $|w_i|_{a_i} = 1$. We shall say that the k-tuple $(w_1, w_2, \ldots, w_k)$ is *the morphic factorization of* w induced by h.

Example 1. Define

$$h(a) = \varepsilon, \qquad h(b) = a, \qquad h(c) = bcd, \qquad h(d) = \varepsilon, \qquad h(e) = de.$$

Then $abcdde$ is a fixed point of h, $\exp(h) = 2$, and the morphic factorization of w induced by h is $(abcd, de)$.

Morphically imprimitive words can now be characterized in terms of factorizations as follows.

Theorem 2. *A word $w \in \Sigma^*$ is morphically imprimitive if and only if there is a stable morphism h defined on* $\mathrm{alph}(w)$ *such that $h(w) = w$ and at least one factor in the morphic factorization of w induced by h has length at least two.*

This is equivalent to the following conditions: there is a factorization $F = (w_1, w_2, \ldots, w_k)$ of w, and a subset $\mathcal{E}_F$ of $\mathrm{alph}(w)$ *such that*

1. $|w_i|_{\mathcal{E}_F} = 1$ *for each $i = 1, 2, \ldots, k$; and*
2. $\mathrm{alph}(w_i) \cap \mathcal{E}_F = \mathrm{alph}(w_j) \cap \mathcal{E}_F$ *implies $w_i = w_j$.*

The correspondence between the two descriptions is straightforward: If a stable morphism h is given, then we can choose $\mathcal{E}_F = \mathcal{E}_h$, and to define the factorization so that for each w_i we have $w_i = h(a_i)$ for some $a_i \in \mathcal{E}_h$. Then

$$w_i = \ell(a_i)\, a_i\, r(a_i),$$

for some words $\ell(a_i), r(a_i) \in \mathcal{M}_h^*$.

Conversely, if a factorization described in the theorem is given, we can define h on $a \in \mathcal{E}_F$ by $h(a) = w_i$ where $\{a\} = \mathrm{alph}(w_i) \cap \mathcal{E}_F$, and $h(b) = \varepsilon$ for $b \notin \mathcal{E}_F$.

This correspondence allows to use the term *expanding letters* also for elements of $\mathcal{E}_F$.

We say that the morphic factorization $(w_1, w_2, \ldots, w_k)$ is *trivial*, if $k = |w|$, that is, each word w_i has length one. Trivial factorizations are induced by morphisms which are the identity on $\mathrm{alph}(w)$.

The set of expanding letters is not given uniquely by the factorization, as the following example shows.

Example 3. Consider the word $w = caddbccaddbc$. It has a morphic factorization $F_1 = (caddbc, caddbc)$, which is induced by two different stable morphisms h and h' defined by:

$$h_1(a) = caddbc, \qquad \mathcal{M}_{h_1} = \{b, c, d\};$$
$$h'_1(b) = caddbc, \qquad \mathcal{M}_{h'_1} = \{a, c, d\}.$$

The only expanding letter in F_1 is therefore either a or b.

Another morphic factorization of w is $F_2 = (cad, dbc, cad, dbc)$, with the unique set of expanding letters $\{a, b\}$. It is induced by just one morphism h_2 given by

$$h_2(a) = cad, \qquad h_2(b) = dbc, \qquad \mathcal{M}_{h_2} = \{c, d\}.$$

The remaining morphic factorization of w is the trivial one.

We shall return to the ambiguity of expanding letters later.

Let a be a letter in w. We denote by $\mathbf{r}_a$ the word such that $a\mathbf{r}_a$ is the longest common prefix of all suffixes of w starting with a. Similarly, we denote by $\mathbf{l}_a$ the word such that $\mathbf{l}_a a$ is the longest common suffix of all prefixes of w ending with a.

We also denote the word $\mathbf{l}_a a\, \mathbf{r}_a$ by $\mathbf{n}_a$. It is easy to see, considering the last and the first occurrence of a in w, that $a \notin \operatorname{alph}(\mathbf{r}_a)$ as well as $a \notin \operatorname{alph}(\mathbf{l}_a)$.

Informally, we can say that $\mathbf{n}_a$ is the largest neighborhood common to all occurrences of a in w. Such a neighborhood constitutes an "upper bound" for the image of a. More precisely, if $h(w) = w$ then $h(a)$ is a factor of $\mathbf{n}_a$. The following lemma is an easy, but important observation.

Lemma 4. *For each letter* $b \in \operatorname{alph}(\mathbf{n}_a)$ *we have* $|w|_b \geq |w|_a$.

We say that letters a and b are *twins* if $b \in \operatorname{alph}(\mathbf{n}_a)$ and $a \in \operatorname{alph}(\mathbf{n}_b)$. We shall write $a \| b$ in such a case. It is easy to verify the following properties.

Lemma 5. (1) $a \| a$.
(2) *If* $a \| b$*, then* $|w|_a = |w|_b$.
(3) *If* $|w|_a = |w|_b$ *and* $a \in \operatorname{alph}(\mathbf{n}_b)$*, then* $a \| b$.
(4) *If* $a \in \operatorname{alph}(\mathbf{n}_b)$ *and* $b \in \operatorname{alph}(\mathbf{n}_a)$ *then* $a \| b$.
(5) *If* $a \| b$*, then* $\mathbf{n}_a = \mathbf{n}_b$.

Our algorithm, deciding whether a given word w is morphically primitive, works with *positions* within the word. The i-th position in w is defined by the prefix of w of length i, which means that the set of positions in w is simply the set

$$\mathcal{C}_w = \{0, 1, \dots, |w|\}.$$

The factor of w between positions i and j, with $i \leq j$, will be denoted by $w[i, j]$. In other words, $w[0, j]$ is the prefix of w of length j, and

$$w[i, j] = w[0, i]^{-1} w[0, j].$$

Clearly, $w[i, i] = \varepsilon$.

An important role in our considerations play letters with minimal frequency in w occurring in a given factor. Therefore we define

$$\mu(u) = \{a \mid a \in \operatorname{alph}(u) \text{ and } |w|_a \leq |w|_b \text{ for each } b \in \operatorname{alph}(u)\}.$$

Let F be a morphic factorization of w and $\mathcal{E}_F$ be its set of expanding letters. The basic idea behind the algorithm is to classify the positions as *left* or *right*, according to whether they are situated left or right from the expanding letter in the corresponding factor. Formally, let h be a stable morphism inducing F with $\mathcal{E}_h = \mathcal{E}_F$. We define

$$\mathcal{L} = \mathcal{L}(F, \mathcal{E}_F) = \{i \mid i \geq |h(w[0,i])|\},$$
$$\mathcal{R} = \mathcal{R}(F, \mathcal{E}_F) = \{i \mid i \leq |h(w[0,i])|\},$$

where $\mathcal{L}$ denotes the set of left positions and $\mathcal{R}$ the set of right ones. Since the inequalities in the definition are not strict, it can be observed that $\mathcal{L}$ and $\mathcal{R}$ are not disjoint; $\mathcal{L} \cap \mathcal{R}$ is the set of positions that constitute the morphic factorization F.

The following lemma will become the crucial tool for our algorithm.

Lemma 6. *Let F be a morphic factorization of w, induced by a stable morphism h, and $\mathcal{E}_F$ be its set of expanding letters. Let $i, j \in \mathcal{C}_w$ satisfy $i < j$, $i \in \mathcal{L}$ and $j \in \mathcal{R}$. If $c \in \mu(w[i,j])$, then either $c \in \mathcal{E}_F$, or there is a letter $c' \in \operatorname{alph}(w[i,j])$ such that $c' \in \mathcal{E}_F$, $c \| c'$ and $c \in \operatorname{alph}(h(c'))$.*

Proof. The assumptions on i and j imply that $\operatorname{alph}(w[i,j]) \cap \mathcal{E}_h \neq \emptyset$, and for each letter $b \in \operatorname{alph}(w[i,j]) \cap \mathcal{M}_h$ there is a letter $a \in \operatorname{alph}(w[i,j]) \cap \mathcal{E}_h$ such that $b \in \operatorname{alph}(h(a))$.

Suppose that $c \notin \mathcal{E}_h$ and let $c' \in \operatorname{alph}(w[i,j]) \cap \mathcal{E}_h$ be such that $c \in \operatorname{alph}(h(c'))$. Then $|w|_c \geq |w|_{c'}$, by Lemma 4, and the assumption $c \in \mu([i,j])$ yields $|w|_c = |w|_{c'}$. Therefore c and c' are twins, by Lemma 5(3). $\square$

Let us explain informally the content of the previous lemma. We can call a factor $w[i,j]$ satisfying $i \in \mathcal{L}$ and $j \in \mathcal{R}$ a *stretch* factor since it expands on both sides when mapped by h. The claim of the lemma can be then shortly expressed as follows: the least frequent letter occurring in a stretch factor, or one of its twins, has to be expanding.

Example 7. A trivial example illustrating the use of Lemma 6 are words w such that $|w|_a = 1$ for some letter a. Such words are always morphically imprimitive as soon as $w \neq a$. It is enough to define $\mathcal{E}_h = a$ and $h(a) = w$.

Connection to Lemma 6 is the following. Note that w itself is always its stretch factor, which means that the positions $0, |w|$ lie in $\mathcal{L} \cap \mathcal{R}$ for each morphic factorization of w. Lemma 6 now implies $a \in \mathcal{E}_h$.

Note that the sets $\mathcal{L}$ and $\mathcal{R}$ depend on the choice of $\mathcal{E}_F$. The ambiguity of $\mathcal{E}_F$ therefore becomes very inconvenient. Lemma 6, however, yields a partial remedy for this difficulty, which is formulated in the following classification of sets of expanding letters.

Lemma 8. *Let $F = (w_1, w_2, \ldots, w_k)$ be a morphic factorization of w. Then $\mathcal{E}_F$ can be chosen as an arbitrary set of representatives of the collection of sets $\mu(w_1), \mu(w_2), \ldots, \mu(w_k)$. Moreover, for each $i \in \{1, \ldots, k\}$ the elements of $\mu(w_i)$ are pairwise twins.*

Proof. Follows immediately from Lemma 6, since each w_i is obviously a stretch factor independently of the choice of $\mathcal{E}_F$. $\square$

In order to fix the choice of the set of expanding letters for a factorization $F = (w_1, w_2, \ldots, w_k)$, we say that $\mathcal{E}_F$ is the *standard* set of expanding letters, if

$\mathrm{alph}(w_i) \cap \mathcal{E}_F$ is the letter from $\mu(w_i)$ that occurs leftmost in w_i. Since we are interested only in the morphic factorization itself, not in the particular choice of expanding letters, we can always work, without loss of generality, with the standard set of expanding letters.

3. The algorithm

Note that the problem to find out whether there is a nontrivial morphic factorization of a word w is in **NP** since it is easy to verify whether a suggested factorization is correct. Suppose that we have somehow obtained the set $\mathcal{E}$ of expanding letters for a morphic factorization. Then we have a unique factorization

$$w = z_0 a_1 z_1 a_2 z_2 \cdots z_{k-1} a_k z_k$$

such that $a_1 a_2 \cdots a_k \in \mathcal{E}^*$ and $\mathrm{alph}(z_0 z_1 \cdots z_k) \cap \mathcal{E}$ is empty. It remains to split the words $z_1, \ldots, z_{k-1}$ in a suitable way to create a correct morphic factorization of w. It should not be surprising that the latter task is not very difficult. The core of the problem is therefore to find the set of expanding letters, and the main tool to do it is Lemma 6.

In order to exploit Lemma 6 for less trivial cases than the one described in Example 7 we have to determine other positions in w that are forced to be either in $\mathcal{L}$ or in $\mathcal{R}$, in addition to 0 and $|w|$. The following lemmas yield three more rules that can be applied.

Lemma 9. *Let F be a morphic factorization of w with the set of expanding letters $\mathcal{E}_F$. If $w[i, i+1] = a$ and $a \in \mathcal{E}_F$, then $i \in \mathcal{L}$ and $i+1 \in \mathcal{R}$.*

Proof. Follows directly from the definition of $\mathcal{L}$ and $\mathcal{R}$, since a is an expanding letter. □

Lemma 10. *Let $w[i, j] = \mathbf{n}_a$ for some letter a. Then for each morphic factorization F of w such that $a \in \mathcal{E}_F$ we have $i \in \mathcal{R}$ and $j \in \mathcal{L}$.*

Proof. Suppose, for a contradiction, that there is a morphic factorization F of w such that $i \notin \mathcal{R}$. This means that there is a letter b in $\mathbf{l}_a$ such that $\mathbf{l}_a = u_1 b u_2$, and there is a word $t u_1 b v$ in the factorization F, where t is nonempty. This implies that the neighborhood $\mathbf{n}_a$ should be extended into $t\,\mathbf{n}_a$, a contradiction with its maximality. The proof of $j \in \mathcal{L}$ is analogical. □

Finally, the next lemma says that two different neighborhoods of the same letter have the same structure of $\mathcal{L}$ and $\mathcal{R}$ positions.

Lemma 11. *Let F be a morphic factorization of w. Let $a \in \mathcal{E}_F$ and let $w[i, j] = w[i', j'] = \mathbf{n}_a$. If $i+k \in \mathcal{L}$ ($i+k \in \mathcal{R}$ resp.), with $i \le i+k \le j$, then also $i'+k \in \mathcal{L}$ ($i'+k \in \mathcal{R}$ resp.).*

Proof. The claim is very intuitive. Formally, it can be proved as follows. Let h be the stable morphism inducing F with $\mathcal{E}_h = \mathcal{E}_F$, and let

$$w[m, m+1] = w[m', m'+1] = a$$

with $i \le m < j$ and $i' \le m' < j'$. Note that m and m' are unique, and

$$m - (i+k) = m' - (i'+k). \tag{1}$$

Since $a \in \mathcal{E}_h$ we have

$$|h(w[0,m])| - m = |h(w[0,m'])| - m'. \tag{2}$$

From $w[i+k,m] = w[i'+k,m']$, if $i+k \leq m$, or $w[m,i+k] = w[m,i'+k]$, otherwise, we deduce

$$|h(w[0,m])| - |h(w[0,i+k])| = |h(w[0,m'])| - |h(w[0,i'+k])|. \tag{3}$$

Combining the equalities (1)–(3) we obtain

$$|h(w[0,i+k])| - (i+k) = |h(w[0,i'+k])| - (i'+k),$$

which concludes the proof. □

We now have all necessary ingredients to formulate our algorithm. We have listed several conditions forcing some positions to be left or right. On the other hand, if we already know some left and right positions, then Lemma 6 may yield new expanding letters.

We define subsets $\mathtt{L(E)}$ and $\mathtt{R(E)}$ of $\mathcal{C}_w$ as the smallest sets satisfying the following conditions

(a) $0, |w| \in \mathtt{L(E)}$ and $0, |w| \in \mathtt{R(E)}$.
(b) If $w[i,i+1] \in \mathtt{E}$, then $i \in \mathtt{L(E)}$ and $i+1 \in \mathtt{R(E)}$.
(c) If $w[i,j] = \mathbf{n}_a$ for some $a \in \mathtt{E}$, then $i \in \mathtt{R(E)}$ and $j \in \mathtt{L(E)}$.
(d) If $w[i,j] = w[i',j'] = \mathbf{n}_a$ for some $a \in \mathtt{E}$, then
  - $i+k \in \mathtt{L(E)}$ with $i \leq i+k \leq j$ implies $i'+k \in \mathtt{L(E)}$; and
  - $i+k \in \mathtt{R(E)}$ with $i \leq i+k \leq j$ implies $i'+k \in \mathtt{R(E)}$.

On the other hand, for two subsets of $\mathtt{L}$ and $\mathtt{R}$ of $\mathcal{C}_w$ we define $\mathtt{E(L,R)}$ as the smallest subset of $\mathrm{alph}(w)$ satisfying the following condition:

(f) If $i < j$, $i \in \mathtt{L}$ and $j \in \mathtt{R}$ then $\mathtt{E(L,R)}$ contains the letter from $\mu(w[i,j])$ that has the leftmost occurrence in $w[i,j]$.

The crucial part of our algorithm can now be described by the following simple pseudocode.

FixedPoint(w)

$\mathtt{E} \leftarrow \emptyset$;
repeat
    $\mathtt{L} \leftarrow \mathtt{L(E)}$; $\mathtt{R} \leftarrow \mathtt{R(E)}$;
    $\mathtt{E} \leftarrow \mathtt{E(L,R)}$;
  until $\mathtt{L} = \mathtt{L(E)} \wedge \mathtt{R} = \mathtt{R(E)}$;
return $\mathtt{E}, \mathtt{L}, \mathtt{R}$;

To find the corresponding morphic factorization is no more difficult. As noted above, we have a uniquely given factorization

$$w = z_0 a_1 z_1 a_2 z_2 \cdots z_{k-1} a_k z_k, \tag{4}$$

where $k = |w|_{\mathtt{E}}$, and we search for words $u_1, v_1, u_2, v_2, \ldots, u_{k-1}, v_{k-1}$ such that

- $u_i v_i = z_i$ for $i = 1, \ldots, k-1$
- $a_i = a_j$ implies $u_{i-1} a_i v_i = u_{j-1} a_j v_j$ (where we define $u_0 = z_0$ and $v_k = z_k$).

This task may seem nontrivial, but it turns out that it is enough to split z_i in such a way that u_i contains all positions of z_i lying in $\mathtt{R}$, and v_i contains all positions lying in $\mathtt{L}$. In particular, it is possible to define u_i as the maximal prefix of z_i satisfying $|z_0 a_1 z_1 a_2 \cdots z_{i-1} a_i u_i| \in \mathtt{R}$. Such a prefix exists since the empty word satisfies the condition.

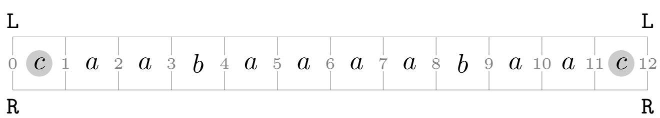
The whole algorithm looks as follows.

MorphicFactorization(w)
$\mathtt{E}, \mathtt{L}, \mathtt{R} \leftarrow$ FixedPoint(w);
$k \leftarrow |w|_{\mathtt{E}}$;
if $\mathtt{E} = \mathrm{alph}(w)$
  then return Primitive;
  else return Imprimitive;
$(z_0, a_1, z_1, a_2, \ldots, z_{k-1}, a_k, z_k) \leftarrow$ the words from (4);
for $i = 1, \ldots, k-1$
  do $u_i \leftarrow$ maximal prefix of z_i such that $|z_0 a_1 z_1 a_2 \cdots z_{i-1} a_i u_i| \in \mathtt{R}$;
    $v_i \leftarrow u_i^{-1} z_i$;
return $(z_0 a_1 u_1, v_1 a_2 u_2, \ldots, u_{k-1} a_k z_k)$;

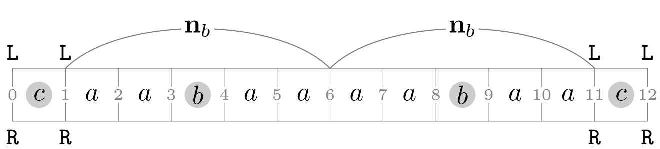
Let us see some examples of how the algorithm works.

Example 12. Let $w = caabaaaabaac$. We have $\mathbf{n}_a = a$, $\mathbf{n}_b = aabaa$ and $\mathbf{n}_c = c$. Let us follow the run of the algorithm FixedPoint. At the beginning we set $\mathtt{E} = \emptyset$. Rounds of the **repeat** loop yield the following:

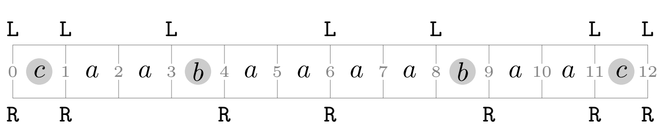
Round 1.
- (a) implies $\mathtt{L}(\emptyset) = \{0, 12\}$; $\mathtt{R}(\emptyset) = \{0, 12\}$;
- since $\mu(w[0, 12]) = \{b, c\}$, (f) implies $\mathtt{E}(\{0, 12\}, \{0, 12\}) = \{c\}$;



Round 2.
- since $c \in \mathtt{E}$, (b) implies $0, 11 \in \mathtt{L}$, $1, 12 \in \mathtt{R}$, and (c) implies $1, 12 \in \mathtt{L}$, $0, 11 \in \mathtt{R}$;
- $\mu(w[1, 11]) = \{b\}$, and (f) implies $b \in \mathtt{E}$;



Round 3.
- since $b \in \mathtt{E}$, (b) implies $3, 8 \in \mathtt{L}$, $4, 9 \in \mathtt{R}$, and (c) implies $6, 11 \in \mathtt{L}$, $1, 6 \in \mathtt{R}$; the condition (d) is satisfied.
- the condition (f) does not yield any new elements for $\mathtt{E}$.



The remaining part of the algorithm MorphicFactorization yields the factorization (c,$aabaa$, $aabaa$,c) with $\mathcal{E} = \{b, c\}$.

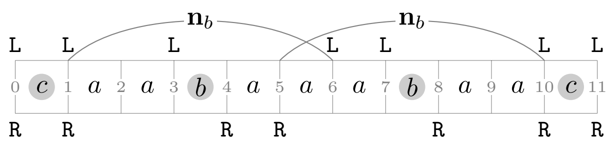
A slightly modified word from the following example shows the importance of the condition (d), which was not needed in Example 12.

Example 13. Consider $w = caabaaabaac$. We have again $\mathbf{n}_a = a$, $\mathbf{n}_b = aabaa$ and $\mathbf{n}_c = c$. The rounds of the **repeat** loop now yield the following (the first two rounds are very similar as in Example 12):

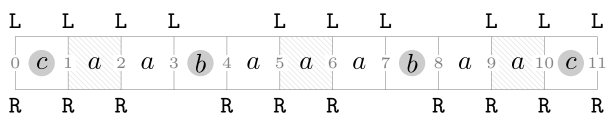
Round 1.
- (a) implies $\mathtt{L}(\emptyset) = \{0, 11\}$; $\mathtt{R}(\emptyset) = \{0, 11\}$;
- since $\mu(w[0, 11]) = \{b, c\}$, (f) implies $\mathtt{E}(\{0, 11\}, \{0, 11\}) = \{c\}$;

Round 2.
- since $c \in \mathtt{E}$, (b) implies $0, 10 \in \mathtt{L}$, $1, 11 \in \mathtt{R}$, and (c) implies $1, 11 \in \mathtt{L}$, $0, 10 \in \mathtt{R}$;
- $\mu(w[1, 10]) = \{b\}$, and (f) implies $b \in \mathtt{E}$;

Round 3.
- since $b \in \mathtt{E}$, (b) implies $3, 7 \in \mathtt{L}$, $4, 8 \in \mathtt{R}$, and (c) implies $6, 10 \in \mathtt{L}$, $1, 5 \in \mathtt{R}$;



- since $w[1, 6] = w[5, 10] = \mathbf{n}_b$, we use (d) to obtain that
  - $1 \in \mathtt{L} \cap \mathtt{R}$ implies $5 \in \mathtt{L} \cap \mathtt{R}$;
  - $5 \in \mathtt{L} \cap \mathtt{R}$ implies $9 \in \mathtt{L} \cap \mathtt{R}$;
  - $10 \in \mathtt{L} \cap \mathtt{R}$ implies $6 \in \mathtt{L} \cap \mathtt{R}$;
  - $6 \in \mathtt{L} \cap \mathtt{R}$ implies $2 \in \mathtt{L} \cap \mathtt{R}$;



- $w[1, 2]$, $w[5, 6]$ and $w[9, 10]$ are stretch factors, whence (f) implies $a \in \mathtt{E}$;

Round 4.
- we already have $\mathtt{E} = \mathrm{alph}(w)$, and we obtain $\mathtt{L} = \mathtt{R} = \{0, 1, \dots, 11\}$.

Therefore w is morphically primitive.

We shall now declare and proof the correctness of the algorithms.

Theorem 14. *Let w be a word, let $\mathtt{E}$, $\mathtt{L}$, $\mathtt{R}$ be the output of* FixedPoint(w)*, and let F be the output of* MorphicFactorization(w)*.*

1. *F is a morphic factorization of w such that $\mathcal{E}_F = \mathtt{E}$.*
2. *Let F' be a morphic factorization of w, and let $\mathcal{E}_{F'}$ be its standard set of expanding letters. Then $\mathtt{E} \subseteq \mathcal{E}_{F'}$, $\mathtt{L} \subseteq \mathcal{L}(F', \mathcal{E}_{F'})$ and $\mathtt{R} \subseteq \mathcal{R}(F', \mathcal{E}_{F'})$.*

In particular, w is morphically primitive if and only if $\mathtt{E} = \mathrm{alph}(w)$.

Proof. (1) Denote $v_0 = z_0$ and $u_k = z_k$. It is obvious from the construction that $|v_{i-1} a_i u_i|_{\mathcal{E}_F} = 1$ holds for each $i = 1, \dots, k$. It remains to show that $a_j = a_m$ implies $v_{j-1} = v_{m-1}$ and $u_j = u_m$. For each $i \in \{1, 2, \dots, k\}$ denote

$$d_i = |z_0 a_1 z_1 a_2 \cdots z_{i-1} a_i|.$$

We first claim that both u_j and u_m are prefixes of $\mathbf{r}_a$. To see this, suppose that one of them, say u_j, is not; then $\mathbf{r}_a$ is a prefix of u_j. This implies, by condition (c), that $d_j + |\mathbf{r}_a| \in \mathtt{L}$. Also $d_j + |u_j| \in \mathtt{R}$ by the definition of u_j. Therefore $w[d_j + |\mathbf{r}_a|, d_j + |u_j|]$ is a stretch factor that does not contain any expanding letter, a contradiction with $\mathtt{E} = \mathtt{E}(\mathtt{L}, \mathtt{R})$ and the condition (f).

Since both u_j and u_m are prefixes of $\mathbf{r}_a$, they are both prefixes of the word $z_i a_{i+1} z_{i+1} \cdots a_k z_k$ for both $i = j$ and $i = m$. Therefore both u_j and u_m are prefixes of both z_j and z_m since neither u_j nor u_m contains an expanding letter. Condition (d) and the definition of u_j and u_m now implies $u_j = u_m$.

The proof that $v_{j-1} = v_{m-1}$ is analogical.

(2) Note that the algorithm FixedPoint adds elements into sets `E`, `L` or `R` following the rules given by conditions (a)–(f). It is transparent that the condition (a) corresponds to the trivial fact that 0 and $|w|$ are in $\mathcal{L} \cap \mathcal{R}$ for all morphic factorizations of w; conditions (b)–(d) correspond to Lemma 9–Lemma 11 respectively; and the condition (f) corresponds to Lemma 6. An inspection of just mentioned lemmas implies that all additions into `E`, `L` or `R` made by the algorithm are forced, in the sense expressed by the claim (2) of the present theorem.

Finally, $\mathrm{alph}(w) = \mathcal{E}$ is equivalent to the fact that the corresponding factorization is trivial, which concludes the proof. □

It is not difficult to see that the complexity of the algorithm is polynomial in $n = |w|$. We give just some basic hints.

- The cardinality of `E` is bound by n, whence the **repeat** loop runs at most $\mathrm{alph}(w) + 1$ times, since the cardinality of `E` grows in each round except the last one.
- Consider the construction of sets `L(E)`, `R(E)`. To check each of the conditions (a)–(d) takes polynomial amount of time. After each check either one of the sets grows, or the construction is over. Therefore the construction is polynomial.
- Similarly, the construction of `E`, which is ruled just by the condition (f), is polynomial.
- Therefore FixedPoint runs in polynomial time. Polynomiality of the rest of MorphicFactorization is obvious.

4. Conclusion

The algorithm we have given yields a morphic factorization that is minimal. This means that each morphic factorization is a refinement of the found one, possibly up to residual ambiguity given by the choice of twins, and by the position of cuts between two expanding letters. The minimality is given by the fact that the algorithm yields a list of "obligatory" left and right positions. The factorization itself then easily follows, including information about the above mentioned ambiguity.

We hope that the information yielded by the algorithm can help to answer other questions about morphical primitivity and imprimivity of words.

References

[1] David Hamm and Jeffrey Shallit. Characterization of finite and one-sided infinite fixed points of morphisms on free monoids. Technical Report CS-99-17, University of Waterloo, July 1999.

[2] Tom Head. Fixed languages and the adult languages of OL schemes. *Internat. J. Comput. Math.*, 10(2):103–107, 1981/82.

[3] Daniel Reidenbach and Johannes C. Schneider. Morphically primitive words. In *Proceedings of the 6th International Conference on Words, WORDS 2007*, pages 262–272.

[4] Jeffrey Shallit and Ming wei Wang. On two-sided infinite fixed points of morphisms. *Theor. Comput. Sci.*, 270(1-2):659–675, January 2002.