

EVERY ITERATED MORPHISM YIELDS A co-CFL

Jean BERSTEL

L.I.T.P., Université Pierre et Marie Curie, 4 Place Jussieu, 75230 Paris, France

Communicated by L. Boasson

Received March 1985

The set of left factors of any infinite word generated by an iterated morphism is shown to be the complement of a one-counter context-free language. This is used to solve a problem stated by Main, Bucher and Haussler (1985).

Keywords: Iterated morphism, context-free language, permutation-free word

1. Introduction

Recently, Main [3] and Main, Bucher and Haussler [4] disproved quite a lot of conjectures about various classes of context-free languages by showing that the complement of the set of left factors of the Thue–Morse sequence and of other related infinite words is context-free. The aim of this article is to extend the latter result by showing, with almost the same proof technique, that, for any iterated morphism, the complement of the set of left factors of an infinite word it generates is context-free. It appears in the construction that the language is even one-counter (in the sense of Greibach [2]).

2. Notation

Let A be an alphabet. A morphism $h: A^* \rightarrow A^*$ is *prolongeable* in $a \in A$ if $h(a) = au$ for some nonempty word u . In this case, each $h^n(a)$ is a left factor of $h^{n+1}(a)$. If the lengths of these words are unbounded, then the morphism h defines an infinite word

$$x = a_0 a_1 \dots a_n \dots \quad (a_i \in A),$$

completely characterized by the condition that each

$h^n(a)$ is a left factor of x . The word x is also denoted by $h^\omega(a)$. It is called the infinite word *generated by h at a* . The morphism h itself is called an *iterated morphism* if there is a letter a for which it generates an infinite word.

Observe that the word $x = h^\omega(a)$ is completely determined by the following conditions:

$$a_0 = a, \quad h(x) = x.$$

Also, observe that the words $h^n(a)$ have in fact strictly increasing length, and, moreover, that for each left factor u of x ,

$$|h(u)| > |u|.$$

Indeed, consider the integer p such that

$$|h^p(a)| \leq |u| < |h^{p+1}(a)|.$$

Then,

$$|h(u)| \geq |h^{p+1}(a)| > |u|.$$

3. Result

Theorem. *Let $h: A^* \rightarrow A^*$ be a morphism that generates an infinite word at a letter $a \in A$, and let F be the set of left factors of this infinite word. Then $A^* - F$ is one-counter context-free.*

Proof. Set

$$x = h^\omega(a) = a_0 a_1 \dots a_n \dots \quad (a_n \in A).$$

Since $h(x) = x$, each $h(a_0 a_1 \dots a_n)$ is a left factor of x . More precisely, x is defined inductively as follows:

- (1) $a_0 = a$,
- (2) $a_{k+1} \dots a_{k+r} = h(a_n)$
with $r = |h(a_n)|$,
 $k = |h(a_0 a_1 \dots a_{n-1})|$.

We shall describe a PDA that accepts a word $w = b_0 b_1 \dots b_m$ iff

- (i) $b_0 \neq a_0$,
- (ii) there exists an integer $i < m$ such that

$$|h(b_0 b_1 \dots b_{i-1})| = k < m,$$

and

$$b_{k+1} \dots b_{k+r} \text{ is a left factor of } h(b_i) \quad (\star)$$

with $r = \min\{|h(b_i)|, m - k\}$.

The check of condition (i) is easy. Also, statement (\star) is easily verified by a finite state memory. The only interesting point is the computation of the length of

$$h(b_0 b_1 \dots b_{i-1}).$$

The PDA guesses the integer i , and then, for each of the letters b_k ($0 \leq k \leq i-1$), it pushes $|h(b_k)| - 1$ marks on the stack. (If $h(b_i)$ is the empty word, then the PDA pops. If the stack is empty, then the PDA stops in an unsuccessful state.) Thus, after reading $b_0 b_1 \dots b_{i-1}$, there are

$$|h(b_0 b_1 \dots b_{i-1})| - i$$

marks on the stack. (Observe that if the guessed integer is $i = 0$, then the PDA immediately passes to the next phase.)

In the second phase, the PDA remembers b_i and then reads input letters, one for each mark in the stack. If there are not enough letters to empty the stack, the computation fails. Otherwise, the process stops when the bottom symbol is encountered. It is easily seen that, at this stage, the total number of input letters read is

$$|h(b_0 b_1 \dots b_{i-1})|.$$

Now the final test starts. Remembering that the guessed letter is b_i , the PDA reads available input, at most $|h(b_i)|$ letters, and checks that the word read is not a left factor of $h(b_i)$. In this case, the word is accepted and the computation halts.

It is clear that for a word in F there is no accepting computation. Conversely, if a word w is not in F , then either it does not start with a_0 , or there is a longest word u such that u and $h(u)$ are left factors of both w and x . But then, setting $w = a_0 a_1 \dots a_i w'$ and $u = a_0 a_1 \dots a_s$, the PDA will accept w when guessing the letter a_{s+1} . This completes the proof of the Theorem. \square

4. An application to permutation-free words

An *abelian square* (or *permuted square*) is a nonempty word of the form xy where x and y are equal up to the order of the letters. A word is *permutation-free* if none of its factors is an abelian square. An infinite permutation-free word over a five-letter alphabet is given by Pleasants [5], no such word exists for three letters, and the question for four letters remains open. Main, Bucher and Haussler ask whether there is any context-free language with an infinite, permutation-free complement. We give a positive answer to this question.

Corollary. *There exists a context-free language over a five-letter alphabet with an infinite, permutation-free complement.*

Proof. A slight modification of a construction of Pleasants shows that there is an iterated morphism generating an infinite, permutation-free word. Let $A = \{a, b, c, d, e\}$ and let $s: A^* \rightarrow A^*$ be the isomorphism defined by the cyclic permutation $s(a) = b$, $s(b) = c$, \dots , $s(e) = a$. The morphism

$$h: A^* \rightarrow A^*$$

defined by

$$h(a) = aebedebecedecae,$$

$$h \circ s = s \circ h,$$

has the property that $h^\omega(a)$ is permutation-free [5]. Thus, the Corollary follows from the Theorem. \square

Remark. A counting argument easily shows that there exist infinite words x such that the language

$$L(x) = \{w \mid w \text{ is not a left factor of } x\}$$

is not context-free. The following explicit example is from Grazon [1]. For the infinite word

$$x = a^2ba^{2^2}b \dots a^{2^{(2^n)}}b \dots,$$

the language $L(x)$ is not context-free. The proof is difficult because of the quite remarkable observation, also due to Grazon [1], stating that any language of the form $L(x)$ satisfies Ogden's iteration lemma for context-free languages.

References

- [1] A. Grazon, Thèse de troisième cycle, June 1985.
- [2] S. Greibach, An infinite hierarchy of context-free languages, *J. Assoc. Comput. Mach.* 16 (1969) 91–106.
- [3] M.G. Main, An infinite square-free co-CFL, *Inform. Process. Lett.* 20 (2) (1985) 105–107.
- [4] M.G. Main, W. Bucher and D. Haussler, Applications of an infinite square-free co-CFL, 12th ICALP, Nafplion, July 1985.
- [5] P.A.B. Pleasants, Nonrepetitive sequences, *Proc. Camb. Phil. Soc.* 68 (1970) 267–274.