Constructing Premaximal Binary Cube-free Words of Any Level

Elena Petrova and Arseny Shur

Ural Federal University, Ekaterinburg, Russia

8th International Conference on Combinatorics on Words
Prague, 2011

Repetition-free languages as trees

All repetition-free languages are factorial.

Repetition-free languages as trees

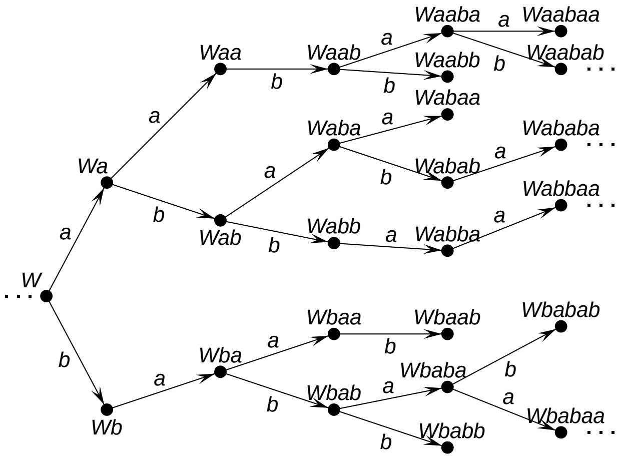
All repetition-free languages are factorial. Any factorial language is a poset with respect to prefix, suffix, or factor order.

Repetition-free languages as trees

All repetition-free languages are factorial. Any factorial language is a poset with respect to prefix, suffix, or factor order. In case of prefix [suffix] order, the diagram of such a poset is a tree:

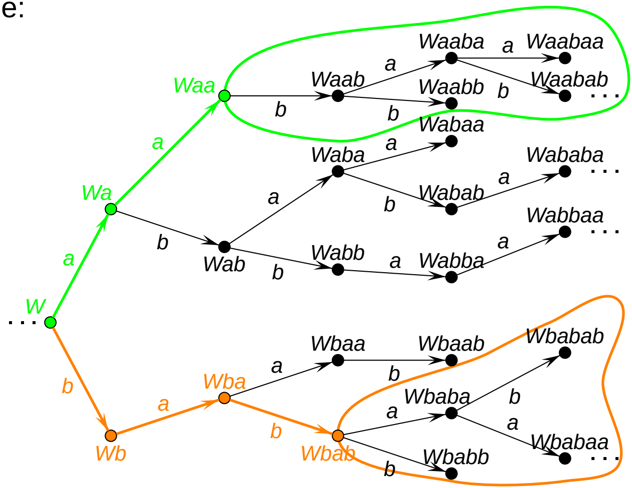
Repetition-free languages as trees

All repetition-free languages are factorial. Any factorial language is a poset with respect to prefix, suffix, or factor order. In case of prefix [suffix] order, the diagram of such a poset is a tree:



Repetition-free languages as trees

All repetition-free languages are factorial. Any factorial language is a poset with respect to prefix, suffix, or factor order. In case of prefix [suffix] order, the diagram of such a poset is a tree:



Questions and known results

1. Does a given word generate finite or infinite subtree?
2. Are the subtrees generated by two given words isomorphic?
3. Can words generate arbitrarily large finite subtrees?

Questions and known results

1. Does a given word generate finite or infinite subtree?
2. Are the subtrees generated by two given words isomorphic?
3. Can words generate arbitrarily large finite subtrees?

- Question 1 for some power-free languages is decidable [Currie, 1995]
- For all kth power-free languages, the subtree generated by any word has at least one leaf [Bean, Ehrenfeucht, McNulty, 1979]
- For the overlap-free binary language, all these questions were answered [Shur, 1998, 2011]

Questions and known results

1. Does a given word generate finite or infinite subtree?
2. Are the subtrees generated by two given words isomorphic?
3. Can words generate arbitrarily large finite subtrees?

- Question 1 for some power-free languages is decidable [Currie, 1995]
- For all kth power-free languages, the subtree generated by any word has at least one leaf [Bean, Ehrenfeucht, McNulty, 1979]
- For the overlap-free binary language, all these questions were answered [Shur, 1998, 2011]

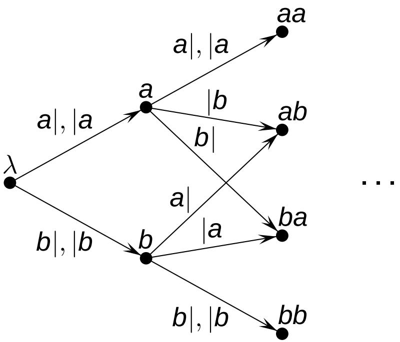
We are going to answer question 3 for the binary cube-free language CF.

Two-sided case

If we consider a repetition-free language as a poset with respect to factor order, its diagram is a DAG. Edges have one of two forms: $w \xrightarrow{c|} cw$ or $w \xrightarrow{|c} wc$, where c is a letter.

Two-sided case

If we consider a repetition-free language as a poset with respect to factor order, its diagram is a DAG. Edges have one of two forms: $w \xrightarrow{c|} cw$ or $w \xrightarrow{|c} wc$, where c is a letter.



Definitions

Let $L \subset \Sigma^*$ and $W \in L$. Any word $U \in \Sigma^*$ such that $UW \in L$ is called a left context of W in L.

Definitions

Let $L \subset \Sigma^*$ and $W \in L$. Any word $U \in \Sigma^*$ such that $UW \in L$ is called a left context of W in L.

- The word W is left maximal [left premaximal] if it has no nonempty left contexts [respectively, finitely many left contexts]. The level of the left premaximal word W is the length of its longest left context.

Definitions

Let $L \subset \Sigma^*$ and $W \in L$. Any word $U \in \Sigma^*$ such that $UW \in L$ is called a left context of W in L.

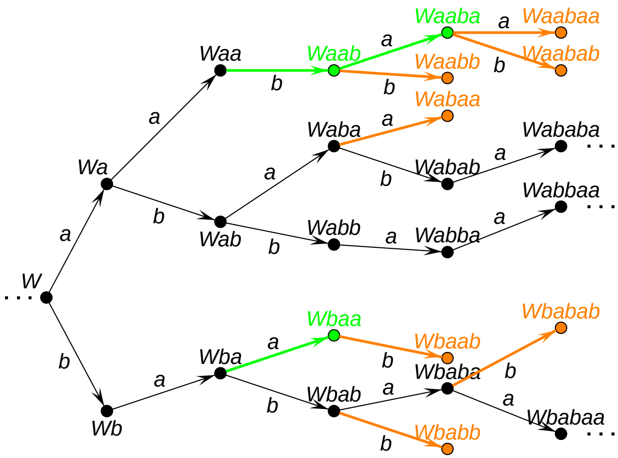
- The word W is left maximal [left premaximal] if it has no nonempty left contexts [respectively, finitely many left contexts]. The level of the left premaximal word W is the length of its longest left context.
- The right counterparts of the above notions are defined in a symmetric way.

Definitions

Let $L \subset \Sigma^*$ and $W \in L$. Any word $U \in \Sigma^*$ such that $UW \in L$ is called a left context of W in L.

- The word W is left maximal [left premaximal] if it has no nonempty left contexts [respectively, finitely many left contexts]. The level of the left premaximal word W is the length of its longest left context.
- The right counterparts of the above notions are defined in a symmetric way.
- We say that a word is maximal [premaximal] if it is both left and right maximal [respectively, premaximal]. The level of a premaximal word W is the pair $(n, k) \in \mathbb{N}$ such that n and k are the length of the longest left context of W and the length of its longest right context, respectively.

Premaximal words



Theorems

Theorem

In CF, there exist left premaximal words of any level $n \in \mathbb{N}_0$*.*

Theorem

In CF, there exist premaximal words of any level $(n, k) \in \mathbb{N}_0^2$*.*

- Moreover, the words $W_{n,k}$ of level (n, k) we have found are such that one can add a n-letter left context and a k-letter right context to $W_{n,k}$ simultaneously.

Constructing premaximal words

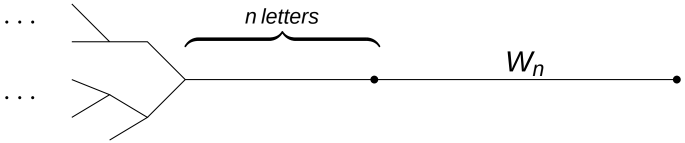
Theorem 1 is proved by exhibiting a series of left premaximal words, containing words of any level. The series is constructed in two steps.

Constructing premaximal words

Theorem 1 is proved by exhibiting a series of left premaximal words, containing words of any level. The series is constructed in two steps.

First step.

We construct an auxiliary series $\{W_n\}_0^\infty$ such that each word W_n has, up to one easily handled exception, a unique left context of any length $\geq n$.



Constructing premaximal words

$$W_0 = aabaaba$$

Constructing premaximal words

$$W_0 = aabaaba$$

We construct the series so that all the fixed left contexts are suffixes of Thue-Morse ω^*-word ${}_\infty U$:

$${}_\infty U = \ldots abba\, baab\, baab\, abba$$

Constructing premaximal words

$$W_0 = aabaaba$$

We construct the series so that all the fixed left contexts are suffixes of Thue-Morse ω^*-word ${}_\infty U$:

$${}_\infty U = \ldots abba\,baab\,baab\,abba$$

The basic idea for obtaining W_{n+1} with the fixed left context $\overline{x}X_n$ from the word W_n with the fixed left context X_n is to let

$$W_{n+1} = \underbrace{\quad W_n}\underbrace{xX_nW_n}\underbrace{xX_nW_n}.$$

Constructing premaximal words

$$W_0 = aabaaba$$

We construct the series so that all the fixed left contexts are suffixes of Thue-Morse ω^*-word ${}_\infty U$:

$${}_\infty U = \ldots abba\, baab\, baab\, abba$$

The basic idea for obtaining W_{n+1} with the fixed left context $\overline{x}X_n$ from the word W_n with the fixed left context X_n is to let

$$W_{n+1} = \underbrace{\quad W_n}\underbrace{xX_nW_n}\underbrace{xX_nW_n}.$$

An attempt to build the series $\{W_n\}_0^\infty$ directly by this scheme fails because cubes will occur at the border of some words W_n and xX_n. So we insert a special "buffer" word after each of three occurrences of W_n in W_{n+1}:

$$W_{n+1} = \underbrace{\quad W_nS_n}\underbrace{xX_nW_nS_n}\underbrace{xX_nW_nS_n}$$

The series S_n

- S_n is the word inserted after W_n at the $(n+1)$th iteration;
- $S'_n = S_0 S_1 \cdots S_n$ is the factor of W_{n+1} between W_0 and the nearest occurrence of xX_n;
- S'_∞ is an image of Thue-Morse ω-word *baab abba abba baab* ... under some 108-uniform cube-free morphism
- S'_∞ consists of Thue-Morse 2-blocks *baab* and *abba*

First members of the series W_n

$$W_0 = W_1 = W_2 = W_3 = aabaaba$$
$$W_4 = \quad aabaaba\,S'_3\,aabbaabaaba\,S'_3\,aabbaabaaba\,S'_3,$$

where $S'_3 = baab\,abba\,abba\,baab$

First members of the series W_n

$$W_0 = W_1 = W_2 = W_3 = aabaaba$$
$$W_4 = \quad aaabaaba\, S_3'\, aabbaabaaba\, S_3'\, aabbaabaaba\, S_3',$$

where $S_3' = baab\, abba\, abba\, baab$

First members of the series W_n

$$W_0 = W_1 = W_2 = W_3 = aabaaba$$
$$W_4 = \quad baabaaba\,S'_3\,aabbaabaaba\,S'_3\,aabbaabaaba\,S'_3,$$

where $S'_3 = baab\,abba\,abba\,baab$

First members of the series W_n

$$W_0 = W_1 = W_2 = W_3 = aabaaba$$
$$W_4 = \underbrace{aba}\,\underbrace{aba}\,\underbrace{aba}\,S'_3\,aabbaabaaba\,S'_3\,aabbaabaaba\,S'_3,$$

where $S'_3 = baab\,abba\,abba\,baab$

First members of the series W_n

$W_0 = W_1 = W_2 = W_3 = aabaaba$
$W_4 = \quad bbaabaaba\,S'_3\,aabbaabaaba\,S'_3\,aabbaabaaba\,S'_3,$

where $S'_3 = baab\,abba\,abba\,baab$

First members of the series W_n

$$W_0 = W_1 = W_2 = W_3 = aabaaba$$
$$W_4 = \ bbbaabaaba\, S_3'\, aabbaabaaba\, S_3'\, aabbaabaaba\, S_3',$$

where $S_3' = baab\, abba\, abba\, baab$

First members of the series W_n

$$W_0 = W_1 = W_2 = W_3 = aabaaba$$
$$W_4 = \ abbaabaaba\,S_3'\,aabbaabaaba\,S_3'\,aabbaabaaba\,S_3',$$

where $S_3' = baab\,abba\,abba\,baab$

First members of the series W_n

$$W_0 = W_1 = W_2 = W_3 = aabaaba$$
$$W_4 = \underbrace{aabbaabaaba\,S_3'}\ \underbrace{aabbaabaaba\,S_3'}\ \underbrace{aabbaabaaba\,S_3'},$$
where $S_3' = baab\,abba\,abba\,baab$

First members of the series W_n

$$W_0 = W_1 = W_2 = W_3 = aabaaba$$
$$W_4 = babbaabaaba\,S_3'\,aabbaabaaba\,S_3'\,aabbaabaaba\,S_3',$$

where $S_3' = baab\,abba\,abba\,baab$

Second step

Second step is the completion of the word W_n to a left premaximal word $\overline{W}_n$.

Second step

Second step is the completion of the word W_n to a left premaximal word $\overline{W}_n$. In order to obtain a premaximal word of level n, we build the word W_{n+1} in $n+1$ iterations and then prohibit the extension of W_{n+1} by the first letter of the word xX_n.

Second step

Second step is the completion of the word W_n to a left premaximal word $\overline{W}_n$. In order to obtain a premaximal word of level n, we build the word W_{n+1} in $n+1$ iterations and then prohibit the extension of W_{n+1} by the first letter of the word xX_n.

$$\overline{W}_n = \underbrace{\qquad W_{n+1}\overline{S}_n}\underbrace{\overline{x}X_nW_{n+1}\overline{S}_n}\underbrace{\overline{x}X_nW_{n+1}\overline{S}_n},$$

where $\overline{S}_n$ is a “buffer” inserted similarly to S_n in order to avoid cubes at the border of the occurrences of W_{n+1} and $\overline{x}X_n$.

Second step

Second step is the completion of the word W_n to a left premaximal word $\overline{W}_n$. In order to obtain a premaximal word of level n, we build the word W_{n+1} in $n+1$ iterations and then prohibit the extension of W_{n+1} by the first letter of the word xX_n.

$$\overline{W}_n = \underbrace{\qquad W_{n+1}\overline{S}_n}\underbrace{\overline{x}X_nW_{n+1}\overline{S}_n}\underbrace{\overline{x}X_nW_{n+1}\overline{S}_n},$$

where $\overline{S}_n$ is a "buffer" inserted similarly to S_n in order to avoid cubes at the border of the occurrences of W_{n+1} and $\overline{x}X_n$.

- We do not need to build a cube-free right-infinite word similar to S'_∞, because this construction is used only once.

Děkuji mnohokrát!

Thank you!