Factor Complexity and Abelian Complexity for Infinite Arrays

Huldah Samuel\(^a\)\(^*\) and V. Rajkumar Dare\(^b\)

\(^a,b\)Department of Mathematics, Madras Christian College, Tambaram, Chennai - 600 059

Abstract

Combinatorics on words is an interesting area of research. For a given word, various complexity measures like factor complexity, abelian complexity, arithmetical complexity, palindromic complexity, permutation complexity etc have been defined in the literature. An infinite word can be defined as the limit of an increasing sequence of finite words. The problem of computing the complexity function \(p\) where \(p(n)\) is the number of distinct factors of length \(n\) of an infinite word was introduced in 1975 by Ehrenfeucht et al [3]. Abelian complexity of infinite words have been examined by Ethan M. Covan and Gustav A. Hedlund [1] and they have revealed that it could serve as an alternative way of characterisation of periodic words and Sturmian words. An infinite array over an alphabet is a rectangular arrangement of symbols in infinite rows and infinite columns. In this paper we define Thue-Morse array, and Variant Thue-Morse array. We also define the abelian complexity and factor complexity for infinite arrays such as Fibonacci array, Thue-Morse array, and Variant Thue-Morse array.

Keywords: Mizoguchi-Takahashi contraction, fixed point theorem.

2010 MSC: 34G20.

1 Preliminaries

In this section we recall some of the definitions and notations which are useful for this paper.

**Definition 1.1.** An alphabet \(\Sigma\) is a finite non-empty set \(\Sigma = \{a_1, a_2, \ldots, a_n\}\). The elements of \(\Sigma\) are called letters. A finite word \(w\) over \(\Sigma\) is given by \(w = a_1a_2\ldots a_n\) where \(a_i \in \Sigma\), for \(1 \leq i \leq n\).

The set of all finite words over \(\Sigma\) is denoted by \(\Sigma^*\), we write \(\Sigma^+ = \Sigma^* - \{\lambda\}\) where \(\lambda\) is the empty word.

**Definition 1.2.** An infinite word \(w\) over a finite alphabet \(\Sigma\) is a mapping from positive integers into \(\Sigma\). We write \(w = a_1a_2\ldots a_i\ldots\) where \(a_i \in \Sigma\), for all \(i \geq 1\).

**Notation:** The set of all finite words of length \(n\) over \(\Sigma\) is denoted by \(\Sigma^n\) and the set of all infinite words over \(\Sigma\) is denoted by \(\Sigma^\omega\).

**Definition 1.3.** An infinite word \(w\) is ultimately periodic if \(w = uv^\omega\), where \(u \in \Sigma^*\) and \(v \in \Sigma^+\).

**Definition 1.4.** An infinite word \(w\) over a binary alphabet \(\Sigma = \{a, b\}\) which is not ultimately periodic such that for any positive integer \(n\), the number \(g_x(n)\) of its factors of length \(n\) is minimal. i.e., \(g_x(n) = n + 1\), is called a sturmian word.

**Definition 1.5.** The Fibonacci word \(f = abaaababaabababaabababa\ldots\) which is an important example of a sturmian word is the fixed point of a morphism \(\phi : B^* \rightarrow B^*\) where \(B = \{a, b\}\) and \(\phi(a) = ab, \phi(b) = a\).

**Definition 1.6.** The Thue-Morse word is a fixed point of the morphism \(\phi : T^* \rightarrow T^*\) where \(T = \{a, b\}\) and \(\phi(a) = ab, \phi(b) = ba\) and starts with \(a\). i.e., \(t = aababbabaababababababa\ldots\)

\(^*\)Corresponding author.
E-mail address: huldahs@yahoo.com (Huldah Samuel), rajkumardare@yahoo.com (V. Rajkumar Dare).
Definition 1.7. An array \( A = (a_{ij})_{m \times n} \) over an alphabet \( \Sigma \) is a rectangular arrangement of symbols of \( \Sigma \) in \( m \) rows and in \( n \) columns. The size of the array \( A \) is the ordered pair \((m, n)\). The set of all arrays over \( \Sigma \) together with the empty array \( \lambda \) is denoted by \( \lambda^{\ast\ast} \).

Definition 1.8. An infinite array \( U \) has an infinite number of rows and infinite number of columns. The collection of all infinite arrays over \( \Sigma \) is denoted by \( \Sigma^{\omega \omega} \). The collection of all arrays with finite number of rows and an infinite number of columns is denoted by \( \Sigma^{\ast \omega} \) and the collection of all arrays with an infinite number of rows and a finite number of columns is denoted by \( \Sigma^{\omega \ast} \).

Definition 1.9. The set of all factors of an infinite word \( u \) is called language and we denote it by \( L(u) \). Its subset containing all factors of \( u \) of length \( n \) is denoted by \( L_n(u) \).

Definition 1.10. Let \( U \) be an infinite array over \( \Sigma \). Any array over the same \( \Sigma \), having \( m \) rows and \( n \) columns where \( m \) and \( n \) are finite, is a sub array of \( U \).

2 Complexity Functions

For a given infinite word, there are defined various complexity functions which measure different kinds of complexities such as factor complexity, Abelian complexity, arithmetical complexity, permutation complexity, palindromic complexity etc. In this section, we recall factor complexity for infinite words and define the factor complexity and abelian complexity for infinite arrays.

Definition 2.11. Let \( u \) be an infinite word. The map \( C_u : N^+ \rightarrow N^+ \) defined as \( C_u(n) = \#L_n(u) \) is called factor complexity of \( u \).

Example 2.1. Let us consider the Fibonacci word \( f \).
\[
L_1(f) = \{0, 1\}
\]
\[
L_2(f) = \{00, 01, 10\}
\]
\[
L_3(f) = \{001, 010, 100, 101\}
\]
\[
\vdots
\]
Indeed, the factor complexity of the Fibonacci word \( C_f(n) = n + 1 \).

Definition 2.12. Let \( U \) be an infinite array over \( \Sigma = \{a, b\} \). Let \( L_{m \times n}(U) \) be the set of all subarrays of \( U \) of order \( m \times n \). Then the factor complexity of \( U \) is defined as \( \#L_{m \times n}(U) \). i.e., the number of distinct elements in the set \( L_{m \times n}(U) \).

Example 2.2. Let \( U = a a a a a \ldots 
\]
\[
\vdots
\]
Then \( L_{1 \times 2} = \{aa, ba, bb\} \)
\[
L_{3 \times 2} = \{a a a a a b a b b \}
\]
The factor complexity of \( L_{3 \times 2}(U) = 5 \) and that of \( L_{1 \times 2}(U) = 3 \).

Let us consider the Fibonacci array \( F = a b a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a b a a
We now discuss abelian complexity for infinite arrays. The Abelian complexity of an infinite word is defined in a similar way as factor complexity of an infinite word. Instead of different factors of a given length, the distinct Parikh vectors are counted.

**Definition 2.13.** Let \( u \) be an infinite word over the alphabet \( \Sigma = \{0, 1, \ldots, d - 1\} \). Then the Abelian complexity is the map \( AC_u : N^+ \to N^+ \) defined as \( AC_u(n) = \# \{ \psi(w) | w \in L_n(u) \} \), where \( \psi(w) = (|w_0|, \ldots, |w_{d-1}|) \) is Parikh vector of a word \( w \).

**Example 2.3.** The abelian complexity of the Fibonacci word \( f \) is calculated as follows:

- Since \( \psi(0) = (1,0) \), \( \psi(1) = (0,1) \), \( AC_f(1) = 2 \)
- Since \( \psi(00) = (2,0) \), \( \psi(01) = (1,1) \), \( \psi(10) = (1,1) \), \( AC_f(2) = 2 \)
- \( \psi(001) = (2,1) \), \( \psi(010) = (2,1) \), \( \psi(100) = (2,1) \), \( \psi(101) = (1,2) \)

The distinct Parikh vectors for each factor complexity gives the Abelian complexity. The Abelian complexity of the Fibonacci word is constantly equal to 2.

**Definition 2.14.** Let \( U \) be an infinite array over the alphabet \( \Sigma = \{a, b\} \). Then the Abelian complexity of \( U \) is the map \( AC^\Sigma_{m \times n}(U) : N^+ \times N^+ \to N^+ \) defined as \( AC^\Sigma_{m \times n}(U) = \# \{ L_{m \times n}(U) \text{ having distinct rowwise Parikh vectors} \} \).

For example, if we consider the Fibonacci array \( f \), the abelian complexity of \( f \) is calculated as follows:

\[
L_{2 \times 3}(f) = \begin{pmatrix}
    a & b & a \\
    b & a & b \\
    a & b & a \\
    b & a & a \\
    a & a & b \\
    b & a & b
\end{pmatrix}
\]

Therefore \( AC(f)_{2 \times 3} = 3 \) because as there are only three distinct subarrays which have rowwise distinct Parikh vectors.

**Proposition 2.1.** Let \( U \) be an infinite array on \( \Sigma = \{a, b\} \).

Then for all \( n \in N^+ \), for subarrays of order \( m \times n \), it is satisfied that

\[
1 \leq AC^\Sigma_{m \times n}(U) \leq \binom{n+1}{1}.
\]

3 Thue-Morse Array and its Complexities

**Definition 3.15.** The Thue-Morse word is defined as the fixed point morphism \( \phi : T^* \to T^* \) with \( \phi(a) = ab \), \( \phi(b) = ba \) where \( T = \{a, b\} \).

The factor complexity and Abelian complexity of the Thue-Morse word is defined and discussed in [1].

In Thue Morse word, both the letter images are of the same length 2 and have the same Parikh vectors. Hence there can be only two Parikh vectors for an odd length and three for an even length. This was given by a theorem in [1].

**Theorem 3.1.** Let \( TM \) be the Thue Morse word.

Then \( AC_n(TM) = \begin{cases} 
2 & \text{for } n \text{ odd} \\
3 & \text{for } n \text{ even} 
\end{cases} \)

The Thue-Morse word denoted by \( TM \) is ababaabbaababab... starting with \( a \).

The factor complexity counts the number of distinct factors of the word of length \( n \) denoted by \( C_w(n) \). The factor complexity of the Thue-Morse word is as follows: \( C_{TM}(0) = 1 \), \( C_{TM}(1) = 2 \), \( C_{TM}(2) = 4 \), \( C_{TM}(3) = 6 \) and for \( n \geq 2 \) we can write \( C_{TM}(2n + 1) = 2C_{TM}(n + 1) \), \( C_{TM}(2n) = C_{TM}(n) + C_{TM}(n) \).

We now construct the Thue-Morse Array with the Thue-Morse word.

Let \( TM = a_1a_2a_3... \) be the Thue Morse word over \( \Sigma = \{a, b\} \). Then we construct the Thue Morse Array \( T \) from \( TM \) as follows: \( T = (t_{jk}) \) where \( t_{jk} = a_{j+k-1} \).
Note: A Thue-Morse Array is not a Sturmian array.

The Thue-Morse Array is as follows:

\[ T = \ldots b a a b b a a b a b b a a b a a b b a a b a b b a a b b a a b a a b b a a b a b \ldots \]

The Factor complexities of the Thue-Morse array of order $1 \times m$ with $m \geq 2$ is

\[ \mathcal{L}(1 \times (2n+1))(T) = 2\mathcal{L}(n+1)(T), \quad n \geq 2 \text{ and} \]

\[ \mathcal{L}(1 \times 2n)(T) = \mathcal{L}(n+1)(T) + \mathcal{L}_n(T), \quad n \geq 2. \]

**Theorem 3.2.** Let $T$ be the Thue-Morse Array. Then all subarrays of $T$ of order $m \times n$ with equal $m+n$, have the same factor complexity.

**Theorem 3.3.** Let $T$ be the Thue-Morse Array. The number of subarrays of $T$ of order $m \times n$, $m, n \in N^+$ with the same factor complexity is $m+n-1$.

**Properties of the Thue-Morse Array**

1. The Thue-Morse Array is cube-free.
2. A subword of the Thue Morse word of length greater than or equal to 16 will be a palindrome.
3. The diagonal elements of any subarray of $T$ are the same.
4. For any subarray of $T$ of order $m \times n$, the horizontal and vertical words starting with $a_{ij}$ are the same.

For Example,

\[ \begin{array}{ccc}
  b & a & b \\
  b & b & a \\
  a & b & b \\
\end{array} \quad \begin{array}{ccc}
  b & a & a \\
  b & a & a \\
  b & a & b \\
\end{array} \]

**Theorem 3.4.** The Abelian complexity of the Thue-Morse array of order

\[ 1 \times m \text{ is } \begin{cases} 3 & \text{ when } m \text{ even} \\ 2 & \text{ when } m \text{ odd} \end{cases}, \quad m \geq 2 \]

and for subarrays of order

\[ 2 \times m \text{ it is } \begin{cases} 5 & \text{ when } m \text{ even} \\ 6 & \text{ when } m \text{ odd} \end{cases}, \quad \text{ for } m \geq 2. \]

4 Acknowledgement

The authors would like to acknowledge the computing facility due to DST-FIST to the institution.

**References**


Received: July 10, 2015; Accepted: August 23, 2015

UNIVERSITY PRESS

Website: http://www.malayajournal.org/