Enumerating and Locating the Subwords of the Two-dimensional Infinite Fibonacci Word

Sivasankar M∗ Rama R

1 Department of Mathematics, Indian Institute of Technology Madras, Chennai, India

Given an infinite word, enumerating its subwords is an important exercise for understanding the structure of the word. The process of finding all the subwords is quite tricky for two-dimensional words. In this paper we enumerate the subwords of the two-dimensional infinite Fibonacci word, \( f_{\infty, \infty} \), in a few possible ways. In addition, we extend a method for locating the subwords of the one-dimensional infinite Fibonacci word \( f_{\infty} \) to locate the positions of the subwords of \( f_{\infty, \infty} \).

Keywords: Two-dimensional Fibonacci words, Subwords, Conjugates of two-dimensional words, Directed acyclic word graph.

1 Introduction

Let \( w \) be finite/infinite word over an alphabet \( \Sigma \). The details about the subwords (otherwise called, the factors) of \( w \) would be of considerable use for a better understanding of the structure and characteristics of \( w \). The number of factors and the periodic/primitive nature of the word \( w \) are closely related and are in general analysed simultaneously. Any additional information about the factors of \( w \) can help in the factorization/decomposition of \( w \). In turn, factorizations like Lyndon, Ziv–Lempel and Crochemore are used in text compression algorithms [6, 16].

Fibonacci words (more generally Sturmian words) are simple morphic words. By simple, we mean that the morphisms defining these words are short and are easily conceivable. Also, it is known that, for infinite words \( w \), which are not ultimately periodic, \( p_n(w) \geq n + 1 \), where \( p_n(w) \) is the number of factors of length \( n \) of \( w \) [4]. It is interesting to note that Sturmian words are a class of aperiodic infinite words that achieve the least possible \( p_n \) value, namely \( n + 1 \) [21].

Generating the Fibonacci words over \{0, 1\} can be systematically achieved either by the famous Fibonacci morphism, \( \phi(1) = 10, \phi(0) = 1 \) [21] or by recursive constructions like \( f_0 = 0, f_1 = 1, f_n = f_{n-1}f_{n-2}, n \geq 2 \) [8, 9, 10]. Infinite iterations of the Fibonacci morphism or the recursion, generates the infinite Fibonacci word \( f_\infty = 10110101\ldots \) Some remarkable properties of \( f_n \) and \( f_\infty \) are: (i) \( f_\infty \) contains no fourth power, (ii) if a word \( u^2 \) is a factor of \( f_\infty \), then \( u \) is a conjugate of some finite Fibonacci word, (iii) The finite Fibonacci words are primitive [4, 1].

With a minimum number of subwords of any particular length, it is no wonder that the subwords repeat more often in \( f_\infty \) [11, 31, 24]. There are a few interesting systematic ways to list these subwords. In [4],

∗Corresponding Author

ISSN subm. to DMTCS © by the author(s) Distributed under a Creative Commons Attribution 4.0 International License
subwords of length \( k \) are used to list the subwords of length \( k + 1 \). In [26], a directed acyclic word graph (DAWG) is used to analyse the subwords. In [11], the suffixes of the conjugates of a specific conjugate of a finite Fibonacci word is used to find all the factors of a given length.

As a natural extension to the one-dimensional words, two-dimensional words are studied [14, 25, 27]. (We will interchangeably use 1D for one-dimensional and 2D for two-dimensional, hereafter in this article.) Two-dimensional words finding some useful applications in image processing, data compression and crystallography is another push for exploring two-dimensional words. In [3], 2D Fibonacci words, \( f_{m,n}, m, n \geq 0 \), are introduced to show that they attain the general upper bound for the number of occurrences of a particular type of tandem. In [19, 22], a few combinatorial and palindromic properties of \( f_{m,n} \) are studied. In [28], the authors obtain \( f_{\infty, \infty} \), the 2D infinite Fibonacci word, using a 2D morphism. Further, they count the number of tandems occurring in it.

In this paper, we list all the subwords of a given size \((k, l)\), \( k, l \geq 1 \) of the 2D infinite Fibonacci word \( f_{\infty, \infty} \). We systematically extend the methods used in [26, 4, 11] for finding the subwords of \( f_{\infty, \infty} \), to \( f_{\infty, \infty} \). The remaining of the paper is organized as follows. In section 2, all the required definitions and notions are elaborated. In section 3, a DAWG for \( f_{\infty, \infty} \) is constructed and the subwords of \( f_{\infty, \infty} \) are enumerated. The method explained in section 4, uses the subwords of size \((k, l)\) to find the subwords of size \((k + 1, l + 1)\). In section 5, given a \( k \geq 2 \) and a \( l \geq 2 \), to list all the subwords of size \((k, l)\), conjugates of a special conjugate of \( f_{m,n} \) \((m, n \text{ depend on } k, l)\) are used. Section 7 has concluding remarks and a few future directions.

2 Preliminaries

2.1 One-dimensional Words

In formal language theory, \( \Sigma \), an alphabet is a finite set of symbols and \( \Sigma^* \) is the free monoid generated by \( \Sigma \). The elements of \( \Sigma^* \) are called words and are obtained by concatenating symbols from \( \Sigma \). The neutral element of \( \Sigma^* \) is the empty word (denoted by \( \lambda \)) and we have \( \Sigma^+ = \Sigma^* - \{\lambda\} \). For a word \( u \in \Sigma^* \), \( \|u\| \) called the length of the word is the number of letters occurring in \( u \). By definition, \( \|\lambda\| = 0 \). Given a word \( w \in \Sigma^* \), \( u \in \Sigma^* \) is a prefix (suffix, respectively) of \( w \), if \( w = uv \) \((w = vu, \text{respectively})\) for some \( v \in \Sigma^* \). The reversal of a word \( u = a_1a_2\cdots a_n, a_i \in \Sigma \), for \( 1 \leq i \leq n \), is the word \( u^R = a_n\cdots a_2a_1 \). A word \( u \) is said to be a palindrome (or a one-dimensional palindrome) if \( u = u^R \).

A word \( w \) is said to be primitive if \( w = u^n \) implies \( n = 1 \) and \( w = u \). Note that a power of a word is nothing but repeated concatenation of the word with itself. That is \( u^n \) is obtained by concatenating \( u \) with itself \( n \) number of times. For a detailed study of formal language theory and combinatorics on words, the reader is referred to [20].

2.2 Two-dimensional Words

The concepts of formal language theory can be obviously extended to two dimensions [14]. A two-dimensional word is called a picture or array and is a rectangular array of symbols taken from \( \Sigma \).

Definition 1. [19] A 2D word \( u = [u_{i,j}]_{1 \leq i \leq m, 1 \leq j \leq n} \) of size \((m, n)\) over \( \Sigma \) is a two-dimensional
Subwords of 2D Fibonacci words

A rectangular finite arrangement of letters:

\[
\begin{array}{cccc}
  u_{1,1} & u_{1,2} & \cdots & u_{1,n-1} & u_{1,n} \\
  u_{2,1} & u_{2,2} & \cdots & u_{2,n-1} & u_{2,n} \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  u_{m-1,1} & u_{m-1,2} & \cdots & u_{m-1,n-1} & u_{m-1,n} \\
  u_{m,1} & u_{m,2} & \cdots & u_{m,n-1} & u_{m,n}
\end{array}
\]

We denote the number of rows and columns of \( u \) by \(|u|_{\text{row}}\) and \(|u|_{\text{col}}\), respectively. An empty array, denoted by \( \Lambda \) is an array of size \((0,0)\). Note that the arrays of size \((m,0)\) and \((0,m)\) for \( m > 0 \) are not defined. The set of all arrays over \( \Sigma \) including \( \Lambda \) is denoted by \( \Sigma^{**} \) and \( \Sigma^{**} \) will denote the set of all non-empty arrays over \( \Sigma \). Any subset of \( \Sigma^{**} \) is called a picture language.

To locate any position or region in an array, we require a reference system [2]. Given an array \( u \), the set of coordinates \( \{1,2,\ldots,|u|_{\text{row}}\} \times \{1,2,\ldots,|u|_{\text{col}}\} \) is referred to as the domain of \( u \). A subdomain or subarray of an array \( u \) (that is, a factor of the 2D word \( u \)), denoted by \( u[[i,j],[i',j']] \), is the portion of \( u \) located in the region \( \{i,i+1,\ldots,i'\} \times \{j,j+1,\ldots,j'\} \), where \( 1 \leq i \leq |u|_{\text{row}}, 1 \leq j \leq |u|_{\text{col}} \).

Similar to the concatenation operation in one dimension, the column concatenation and the row concatenation operations between two arrays are as follows.

**Definition 2.** [14] Let \( u,v \) be arrays of sizes \((m_1,n_1)\) and \((m_2,n_2)\), respectively with \( m_1,n_1,m_2,n_2 > 0 \), over \( \Sigma \). Then, the column concatenation of \( u \) and \( v \), denoted by \( \oplus \), is a partial operation, defined if \( m_1 = m_2 = m \), and is given by

\[
\begin{array}{cccc}
  u_{1,1} & \cdots & u_{1,n_1} & v_{1,1} & \cdots & v_{1,n_2} \\
  \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
  u_{m,1} & \cdots & u_{m,n_1} & v_{m,1} & \cdots & v_{m,n_2}
\end{array}
\]

Similarly, the row concatenation of \( u \) and \( v \), denoted by \( \ominus \), is another partial operation, defined if \( n_1 = n_2 = n \), and is given by

\[
\begin{array}{cccc}
  u_{1,1} & \cdots & u_{1,n} \\
  \vdots & \ddots & \vdots \\
  u_{m_1,1} & \cdots & u_{m_1,n_1} \\
  v_{1,1} & \cdots & v_{1,n} \\
  \vdots & \ddots & \vdots \\
  v_{m_2,1} & \cdots & v_{m_2,n_2}
\end{array}
\]

The column and row concatenation of \( u \) and the empty array \( \Lambda \) are always defined and \( \Lambda \) is a neutral element for both the operations.

For a \( u \in \Sigma^{**} \), an array \( v \in \Sigma^{**} \) is said to be a prefix of \( u \) (suffix of \( u \), respectively), if \( u = (v \ominus x) \oplus y \) \((u = y \ominus (x \ominus v)) \), respectively) for some \( x,y \in \Sigma^{**} \). If \( x \in \Sigma^{++} \), then by \((x^{k_1 \ominus})^{k_2 \ominus}\) we mean that the array is constructed by repeating \( x \), \( k_1 \) times column-wise and \( x^{k_1 \ominus}, k_2 \) times row-wise. An array \( w \in \Sigma^{++} \) is said to be 2D primitive if \( w = (x^{k_1 \ominus})^{k_2 \ominus} \) implies that \( k_1 k_2 = 1 \) and \( w = x \) [13].
2.3 Fibonacci Words

Fibonacci words are closely related with the Fibonacci numbers. Recall the recursive definition of the Fibonacci numerical sequence: $F(0) = 1, F(1) = 1, F(n) = F(n-1) + F(n-2)$ for $n \geq 2$. Likewise, for $\Sigma = \{a, b\}$, the sequence $\{f_n\}_{n \geq 0}$ of Fibonacci words, is defined recursively by $f_0 = a, f_1 = b, f_n = f_{n-1}f_{n-2}$ for $n \geq 2$. First few words of this sequence are: $f_0 = a, f_1 = b, f_2 = ba, f_3 = bab, f_4 = babba$. Note that $|f_n| = F(n)$ for $n \geq 0$. The sequence of Fibonacci words can be obtained by iterating the Fibonacci morphism $\phi : \Sigma^* \rightarrow \Sigma^*$ defined by $\phi(a) = b, \phi(b) = ba$. Infinite number of iterations of $\phi$ produces the $1D$ infinite Fibonacci word $f_\infty$. That is,

$$\lim_{n \rightarrow \infty} \phi^n(b) = f_\infty = \text{babbabab...}$$

Throughout this paper, we use $f_{\infty}^{s_1,s_2}$ to denote the $1D$ infinite Fibonacci word $s_1s_2s_1s_2s_1s_2 \ldots$ over the alphabet $\{s_1, s_2\}$. Similarly, $f_{n}^{s_1,s_2}$ denotes the $1D$ finite Fibonacci word $s_1s_2s_1s_2 \ldots s_1s_2$ or $s_1s_2s_1s_2 \ldots s_2s_1$ accordingly $n$ is even or odd.

The extension of $1D$ Fibonacci words to $2D$ Fibonacci words is presented in [3].

Definition 3. [3] Let $\Sigma = \{a, b, c, d\}$. The sequence of Fibonacci arrays, $\{f_{m,n}\}$ where $m, n \geq 0$, is defined as:

1. $f_{0,0} = \beta, f_{0,1} = \gamma, f_{1,0} = \delta, f_{1,1} = \alpha$ where $\alpha, \beta, \gamma$ and $\delta$ are symbols from $\Sigma$ with some but not all, among $\alpha, \beta, \gamma$ and $\delta$ might be identical.

2. For $k \geq 0$ and $m, n \geq 1$,

$$f_{k,n+1} = f_{k,n} \oplus f_{k,n-1}, \quad f_{m+1,k} = f_{m,k} \oplus f_{m-1,k}.$$ 

For convenience, let us fix $f_{0,0} = a, f_{0,1} = b, f_{1,0} = c, f_{1,1} = d$, where some but not all of $a, b, c$ and $d$ might be identical. For example, let us derive the $2D$ Fibonacci word $f_{2,2}$ [19].

$$f_{2,2} = f_{1,2} \oplus f_{0,2} = (f_{1,1} \oplus f_{1,0}) \oplus (f_{0,1} \oplus f_{0,0}).$$

It can also be obtained by column-wise expansion,

$$f_{2,2} = f_{2,1} \oplus f_{2,0} = (f_{1,1} \oplus f_{0,1}) \oplus (f_{1,0} \oplus f_{0,0}).$$

Using the $f_{0,0} = a, f_{0,1} = b, f_{1,0} = c, f_{1,1} = d$, $f_{2,2}$ is given by

$$f_{2,2} = \begin{bmatrix} f_{1,1} \\ f_{0,1} \end{bmatrix} = \begin{bmatrix} d & c \\ b & a \end{bmatrix}.$$ 

We state here some properties of $f_{m,n}$ which we would use later in our proofs.

Lemma 1. [19] Let $f_{m,n}, (m, n = 0, 1, 2, \ldots)$ be the sequence of $2D$ Fibonacci arrays over $\Sigma = \{a, b, c, d\}$, with $f_{0,0} = a, f_{0,1} = b, f_{1,0} = c, f_{1,1} = d$. Also let $\Sigma_1 = \{a, b\}, \Sigma_2 = \{c, d\}, \Sigma_1' = \{a, c\}$ and $\Sigma_2' = \{b, d\}$ such that $\Sigma = \Sigma_1 \cup \Sigma_2 = \Sigma_1' \cup \Sigma_2'$. Then,

a. any row of $f_{m,n}$ is a $1D$ Fibonacci word over either $\Sigma_1$ or $\Sigma_2$. 

Subwords of 2D Fibonacci words

b. if $\Sigma_1 \neq \Sigma_2$ then all the rows of $f_{m,n}$ over $\Sigma_1$ are identical and all the rows of $f_{m,n}$ over $\Sigma_2$ are identical.

c. any column of $f_{m,n}$ is a 1D Fibonacci word over either $\Sigma'_1$ or $\Sigma'_2$.

d. if $\Sigma'_1 \neq \Sigma'_2$ then all the columns of $f_{m,n}$ over $\Sigma'_1$ are identical and all the columns of $f_{m,n}$ over $\Sigma'_2$ are identical.

e. if $\Sigma_1 = \Sigma_2(\Sigma'_1 = \Sigma'_2)$, then either all the rows(columns) of $f_{m,n}$ are identical or a set of rows are identical and are complementary to the set of remaining rows (columns, respectively) which are identical.

2.4 The 2D Infinite Fibonacci Word, $f_{\infty,\infty}$

The sequence of 2D finite Fibonacci words, $\{f_{m,n}\}_{m,n \geq 0}$, in a sense, has the 2D infinite Fibonacci word, $f_{\infty,\infty}$, as its limit. This can be perceived by extending each row, column of any $f_{m,n}$, $m, n \geq 2$, to the 1D infinite Fibonacci word over the alphabet of the word present in that row, column. But this outlook is informal. Formally, in [28], the authors have defined the 2D infinite Fibonacci word through the 2D morphism,

$$\mu : d \rightarrow d \ c \ b \ a, \ c \rightarrow d \ b, \ b \rightarrow d \ c, \ a \rightarrow d. \ (1)$$

For a detailed study of multidimensional morphisms, [7] can be referred.

Observe that the morphism defined by (1) is prolongable on $d$ and infinite number of iterations of $\mu$ on $d$ produces $f_{\infty,\infty}$ [28]. That is to say, $f_{\infty,\infty}$ is the fixed point of the morphism $\mu$. That is,

$$f_{\infty,\infty} = \lim_{n \rightarrow +\infty} \mu^n(d) = \mu^{\omega}(d).$$

First few iterations of $\mu$ on $d$ are shown below.

$$d \rightarrow d \ c \ b \ a \rightarrow d \ c \ d \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b \ a \ b.
The directed acyclic word graph of a word \( w \), \( \text{DAWG}(w) \), is the smallest finite state automaton that recognizes all the suffixes of the word [5]. \( C\text{DAWG}(w) \), a space efficient variant of \( \text{DAWG}(w) \), is obtained by compacting \( \text{DAWG}(w) \) [12].

In [26], the subwords of \( f_\infty \) are analysed through the graph \( G_\infty \), which is, in a certain sense, a \( \text{DAWG} \) of \( f_\infty = abaaababb \ldots = f_\infty(1, 2, 3, \ldots) \). The \( \text{DAWG} \) is constructed as below:

Let \( F(0) = 1, F(1) = 2, F(n) = F(n - 1) + F(n - 2), \) for \( n \geq 2 \), be the Fibonacci sequence. The nodes of \( G_\infty \) are all non-negative integers. For \( i > 0 \), with \( F(i) \) being the \( i^{th} \) Fibonacci number, the labelled edges of \( G_\infty \) are

\[
(i - 1) \xrightarrow{f_\infty(i)} i, \quad F(i) - 2 \xrightarrow{s} F(i + 1) - 1
\]

where \( s = a \) whenever \( i \) is even and \( s = b \) whenever \( i \) is odd (Refer Fig. [1]).

Fig. 1: DAWG, \( G_\infty \) of \( f_\infty^{a,b} \)

3.1 Cross Product of DAWGs

As the \( 2D \) finite Fibonacci words, \( f_{m,n} \), can be obtained by the Cartesian product of Fibonacci reduced representation of the integers \( m, n \) [12], a natural extension of \( G_\infty \) for the \( 2D \) infinite Fibonacci word will be the Cartesian product of \( G_\infty \) with itself.

**Definition 4.** [32] The Cartesian product of \( G \) and \( H \), written \( G \square H \), is the graph with vertex set \( V(G) \times V(H) \) specified by putting \( (u, v) \) adjacent to \( (u', v') \) if and only if \((1)u = u' \text{ and } vv' \in E(H), \) or \((2)v = v' \text{ and } uu' \in E(G). \)

Since \( f_\infty \) has two distinct rows (one over \( \{d, c\} \) and one over \( \{b, a\} \) [22], to obtain a \( \text{DAWG} \) of \( f_\infty \), we slightly modify the labels of \( G_\infty \). Note that, all the rows of \( f_\infty \) are \( f_\infty \) only. In fact the rows over \( \{d, c\} \) would be \( dcdcdc \ldots \) and the rows over \( \{b, a\} \) would be \( babababb \ldots \). In order to simultaneously control these two categories of rows/words, we will use a single \( \text{DAWG} \), the \( \text{DAWG} \) of the Fibonacci word \( DCDDCDDC \ldots \), with \( D = \{d, b\} \) and \( C = \{c, a\} \). With this adaptation, \( D \) is allowed to assume either \( d \) or \( b \) and \( C \) is allowed to assume either \( c \) or \( a \). As the rows of \( f_\infty \) are words over a binary alphabet, we also impose an additional condition that, if \( D \) assumes \( d \) then \( C \) would assume \( c \) and if \( D \) assumes \( b \) then \( C \) would assume \( a \). This \( \text{DAWG} \), say \( "G_\infty \text{ for rows}" \), is depicted at the top, in Fig. [2]. In the graph, for convenience, we have written \( D = \{d, b\} \) and \( C = \{c, a\} \) as ‘\( d, b \)’ and ‘\( c, a \)’, respectively.

Similarly, since \( f_\infty \) has two distinct columns (one over \( \{d, b\} \) and one over \( \{c, a\} \) [22], to manage both the type of columns through a single \( \text{DAWG} \), we consider the \( \text{DAWG} \) of the Fibonacci word \( D'BD'D'BD'BD' \ldots \), where \( D' = \{d, c\} \) and \( B = \{b, a\} \), implying \( D' \) can be either \( d \) or \( c \), and \( B \) can be either \( b \) or \( a \). With an additional condition that, if \( D' \) is \( d \) then \( B \) would be \( b \) and if \( D' \) is \( c \) then \( B \) would
be $a$. Again, in the graph, for convenience we write only ‘$d, c$’ and ‘$b, a$’ (without the curly braces). This DAWG, say "$G_{\infty}$ for columns", is depicted at the left, in Fig. 2.

Now we obtain the Cartesian product of "$G_{\infty}$ for columns" and "$G_{\infty}$ for rows". Note that, when $G$ and $H$ are labelled, the labels are carried over to the edges of the Cartesian product appropriately. The resulting graph is given in Fig. 2.

Since 1D words have only one direction, one can get all the letters of a subword by traversing along a directed path (starting at the root) of their DAWGs. But in the DAWGs of 2D words (if they exist), to get all the letters in a subword, all the edges that lie between the root and any node that lie in a different column/row may have to be traversed. Clearly, this is not possible as the intended DAWG (that is, the Cartesian product) is acyclic and also prevents any back-and-forth traversals.

But the structure of 2D Fibonacci words is such that, for a subword $u$ of $f_{\infty, \infty}$, the knowledge of any one row and any one column of $u$ is enough to write down the entire $u$. Due to this, the Cartesian product will serve as the DAWG of $f_{\infty, \infty}$. Further, since it is enough to know just a row and a column of $u$, even the Cartesian product is redundant and we need only the "rooted product" of "$G_{\infty}$ for rows" and "$G_{\infty}$ for...
columns”.

3.2 Rooted Product of DAWGs

Definition 5. [18] The rooted product of a graph $G$ and a rooted graph $H$, denoted by $G \circ H$, is defined as follows: take $|V(G)|$ copies of $H$, and for every vertex $v_i$ of $G$, identify $v_i$ with the root vertex of the $i^{th}$ copy of $H$.

In other words if the vertex set of $G$ is $\{g_1, \ldots, g_n\}$ and the vertex set of $H$ is $\{h_1, \ldots, h_m\}$ with $h_1$ as its root, then the vertex set, $V$ and the edge set, $E$ of $G \circ H$ will be as below.

$$V = \{(g_i, h_j) : 1 \leq i \leq n, 1 \leq j \leq m\}$$

$$E = E_1 \cup E_2$$

where,

$$E_1 = \{((g_i, h_1), (g_k, h_1)) : (g_i, g_k) \in E(G)\},$$

$$E_2 = \bigcup_{i=1}^{n} \{((g_i, h_j), (g_i, h_k)) : (h_j, h_k) \in E(H)\}.$$

In fact, it is easy to see that, $G \circ H$ is a subgraph of $G \Box H$.

Now, we take the "rooted product" of "$G_\infty$ for rows" and "$G_\infty$ for columns" (Refer Fig. 3) to get the DAWG of $f_{\infty, \infty}$ and denote it by $G_{\infty, \infty}$. From $G_{\infty, \infty}$, we can obtain the first row and the last column of any subword of $f_{\infty, \infty}$. We designate the node $(0,0)$ as the root node of $G_{\infty, \infty}$.

3.3 Enumerating the subwords: The DAWG way

In this subsection, we prove that the number of finite paths in $G_{\infty, \infty}$, starting at its root node, equals the number of subwords of $f_{\infty, \infty}$. In particular, we prove that, for $k, l \geq 1$, a path of length $k + l$, comprising of a horizontal path of length $k$ and a vertical path of length $l$, will lead to subword of $f_{\infty, \infty}$ of size $(k, l)$. Note that by a horizontal path (a vertical path, respectively), we mean a path whose adjacent vertices are in $G_{\infty}$ for rows ($G_{\infty}$ for columns, respectively).

Theorem 1. Let $k, l \in \mathbb{N}$ be given. Then, from a path of length $k + l$ (starting at the root) in $G_{\infty, \infty}$ comprising of a horizontal path of length $l$ and a vertical path of length $k$, we can construct a subword of $f_{\infty, \infty}$ of size $(k, l)$.

Proof: Due to the construction of $G_{\infty, \infty}$, when we start at the root and traverse a horizontal path of length $l \geq 1$, we get a subword of the 1D Fibonacci infinite word $DCDDCDDC \ldots$. In fact, we can obtain two horizontal subwords of length $l$, one over $\{d, c\}$ (obtained by taking $d$ for $D$ and $c$ for $C$) and one over $\{b, a\}$ (obtained by taking $b$ for $D$ and $a$ for $C$). The former subword occurs in any row of $f_{\infty, \infty}$ which is over $\{d, c\}$, and the later occurs in any row of $f_{\infty, \infty}$ which is over $\{b, a\}$.

Now, starting from the last node of this horizontal path, we traverse a vertical path of length $k$. Note that, the rooted product guarantees such a path. Similar to the earlier argument, here we obtain a vertical path of length $k \geq 1$, which corresponds to a subword of length $k$ of the 1D Fibonacci infinite word $D'BD'D'BD'BD' \ldots$. Here also we can obtain two vertical subwords of length $k$, one over $\{d, b\}$ (obtained by taking $d$ for $D'$ and $b$ for $B$) and one over $\{c, a\}$ (obtained by taking $c$ for $D'$ and $a$ for $B$). The former subword occurs in any column of $f_{\infty, \infty}$ which is over $\{d, b\}$, and the later occurs in any column of $f_{\infty, \infty}$ which is over $\{c, a\}$.
To prove that these two paths can produce a unique subword of size \((k, l)\) of \(f_{\infty, \infty}\), we use the fact that ‘the last letter in the first row and the first letter in the last column of a 2D word are the same’. Hence, while constructing the subword, the last letter (say "s\_joint") in the horizontal path has to be the first letter in the vertical path. For example, out of the two available subwords of length \(l\), suppose we select the subword over \(\{d, c\}\), say \(H\), and if \(s\_joint = d\) (\(s\_joint = c\), respectively), then we will (have to) select the vertical subword, say \(V\), over \(\{d, b\}\) (\(\{c, a\}\), respectively). Now, by taking \(H\) and \(V\) as the first row and the last column, respectively, in a 2D word of size \((k, l)\), we will obtain the entire subword. Again note that, this is not possible for all 2D words, but for \(f_{\infty, \infty}\) due to its structure.

As any row of \(f_{\infty, \infty}\) is either over \(\Sigma_1 = \{a, b\}\) or \(\Sigma_2 = \{c, d\}\), \(s\_joint\) has to be either in \(\Sigma_1\) or in \(\Sigma_2\). As any column of \(f_{\infty, \infty}\) is either over \(\Sigma'_1 = \{a, c\}\) or \(\Sigma'_2 = \{b, d\}\), \(V\) has to be either in \(\Sigma'_1\) or in \(\Sigma'_2\). Hence the following four cases only arise.

**Case (i)** : \(s\_joint = a\) (then, \(V\) will be over \(\{a, c\}\))
**Case (ii)** : \(s\_joint = b\) (then, \(V\) will be over \(\{b, d\}\))
**Case (iii)** : \(s\_joint = c\) (then, \(V\) will be over \(\{a, c\}\))
**Case (iv)** : \(s\_joint = d\) (then, \(V\) will be over \(\{b, d\}\))

To find the letters occurring at the other positions of \(u\) we define two substitutions. If \(H\) is over \(\{a, b\}\), we create a 1D word \(H'\) from \(H\) using the substitution \(\theta_1 : \theta_1(a) = c, \theta_1(b) = d\). If \(H\) is over \(\{c, d\}\), we create a 1D word \(H''\) from \(H\) using the substitution \(\theta_2 : \theta_2(c) = a, \theta_2(d) = b\). These words \(H'\) and
$H''$ will be used to fill up/find the other rows of the subword we are constructing. These substitutions are motivated by the fact that, a row of $f_{\infty,\infty}$ over $\{a, b\}$ can be obtained from a row of $f_{\infty,\infty}$ over $\{c, d\}$ and vice-versa through simple substitutions.

Let $R_1, R_2, R_3, \ldots, R_k$ be the $k$ rows of the subword being constructed. Note that $R_1 = H$. Now, for $2 \leq j \leq k$,

**Case(i):** $s_{\text{joint}} = a$ (and hence $H$ is over $\{a, b\}$)
If the letter in the $j^{\text{th}}$ row of $V$ is $s_{\text{joint}}$, then $R_j = H$ else $R_j = H'$.

**Case(ii):** $s_{\text{joint}} = b$ (and hence $H$ is over $\{a, b\}$)
If the letter in the $j^{\text{th}}$ row of $V$ is $s_{\text{joint}}$, then $R_j = H$ else $R_j = H'$.

**Case(iii):** $s_{\text{joint}} = c$ (and hence $H$ is over $\{c, d\}$)
If the letter in the $j^{\text{th}}$ row of $V$ is $s_{\text{joint}}$, then $R_j = H$ else $R_j = H''$.

**Case(iv):** $s_{\text{joint}} = d$ (and hence $H$ is over $\{c, d\}$)
If the letter in the $j^{\text{th}}$ row of $V$ is $s_{\text{joint}}$, then $R_j = H$ else $R_j = H''$.

Note that while constructing the subword, the alphabet of each row and the order in which the two distinct rows ($H$ and $H'$ (or $H$ and $H''$)) of the subword are getting arranged are decided/guided by $V$. Since $V$ is a subword of length $l$ of some column of $f_{\infty,\infty}$, the obtained 2D word is a subword of $f_{\infty,\infty}$ of size $(k, l)$. \qed

**Remark 1.** Theorem 1 can be proved by taking "rooted product" of "$G_{\infty,\infty}$ for columns" and "$G_{\infty,\infty}$ for rows". In that case, first we have to traverse a vertical path of length $k$, then a horizontal path of length $l$ to obtain the first column and the last row of the subword in that order. Finding the other rows can be done similar to the process explained in the proof.

**Remark 2.** Since we constructed $G_{\infty,\infty}$ as the "rooted product" of "$G_{\infty,\infty}$ for rows" by "$G_{\infty,\infty}$ for columns", we will always use a horizontal edge (an edge of $G_{\infty,\infty}$ for rows) at first. Also, as $l \geq 1$, we will never use the copy of $G_{\infty,\infty}$ for columns rooted at $(0, 0)$. Hence we can remove this redundant copy from $G_{\infty,\infty}$ and can still entitle the new graph $G_{\infty,\infty}$.

**Remark 3.** The DAWG also can be constructed by a similar methodology as given in [26]. Let

$$f_{\text{row},\infty} = DCDDCD\ldots = f_{\text{row},\infty}(1, 2, 3, \ldots),$$

$$f_{\text{col},\infty} = D'B'D'B'D'B'\ldots = f_{\text{col},\infty}(1, 2, 3, \ldots)$$

where $D, C, D'$ and $B$ are as defined earlier. The nodes of $G_{\infty,\infty}$ are all non-negative integer pairs, $(i, j), i, j \geq 0$.

For $j > 0$, with $F(j)$ being the $j^{\text{th}}$ Fibonacci number, the labelled edges of $G_{\infty,\infty}$ are

$$(0, j - 1) f_{\text{row},\infty}(j) \rightarrow (0, j), \quad (0, F(j) - 2) \xrightarrow{s} (0, F(j + 1) - 1)$$

where $s = D$ whenever $j$ is even and $s = C$ whenever $j$ is odd, and for each $j \geq 0$ ($j \geq 1$ is suffice; refer Remark 2) and $i > 0$,

$$(i - 1, j) f_{\text{col},\infty}(i) \rightarrow (i, j), \quad (F(i) - 2, j) \xrightarrow{s} (F(i + 1) - 1, j)$$

where $s = D'$ whenever $i$ is even and $s = B$ whenever $i$ is odd.
Corollary 1. For \(k, l \geq 1\), there are \((k + 1)(l + 1)\) subwords of size \((k, l)\) in \(f_{\infty, \infty}\).

**Proof:** As the graph "\(G_{\infty} for rows\)" is the DAWG of the 1D Fibonacci word \(DCDCC\ldots\), there are \((l + 1)\) number of horizontal paths in "\(G_{\infty, \infty} for rows\)" [26]. Since the graph "\(G_{\infty} for columns\)" is the DAWG of the 1D Fibonacci word \(D'B'D'B'B\ldots\) from the last node of every horizontal path, there are \((k + 1)\) number of vertical paths available for traversing. Note that, though paths with labels from \(\{D, C\}\) and \(\{D', B\}\) have two possibilities, due to the condition on \(s_{\text{joint}}\) (as explained in the proof of Theorem 1), only one path with labels \(\{a, b, c, d\}\) will materialize. Thus, there are \((k + 1)(l + 1)\) number of paths of length \(k + l\), comprising of a horizontal path of length \(l\) and a vertical path of length \(k\). Now, by Theorem 1, a path of length \(k + l\) in \(G_{\infty, \infty}\), comprising of a horizontal path of length \(l\) and a vertical path of length \(k\), uniquely corresponds to a subword of size \((k, l)\) of \(f_{\infty, \infty}\). Hence the corollary.

The following example will explain the construction used in the proof of Theorem 1.

**Example 1.** Let \(k = 2\) and \(l = 2\) so that all the subwords of size \((2, 2)\) will be obtained. By corollary, there will be 9 subwords of this size. Construction of one of these 9 subwords is explained here.

The horizontal paths of length 2 in \(DCDCC\ldots\) are \(DC, DD\) and \(CD\). Suppose we select \(D\) for \(D\). Then \(H\) can be any one of \(\{dc, dd, cd\}\). Let us choose \(H\) as \(cd\).

Now the vertical paths of length 2 in \(D'B'D'B'\ldots\) are \(D'B, D'D'\) and \(B\). Since \(s_{\text{joint}} = d\), to have a subword of \(f_{\infty, \infty}\), the vertical path of length 2 should start with \(d\). By selecting \(d\) for \(D'\) we have the three vertical paths \(\{d, d' b\}\).

Let us take \(V = d' b\). Then the first column and the last row of the subword are fixed. The incomplete subword is, \(\begin{array}{c} c \star b \\ a \end{array}\) where the symbol ‘*’ denotes the entry therein is unknown yet.

Since \(s_{\text{joint}} = d\) and the letter in the second row of \(V\) is not \(a\), we fill the second row with \(H'' = ab\).

Hence the subword corresponding to this path is \(\begin{array}{c} c \end{array} \begin{array}{c} d \\ a \end{array} \begin{array}{c} b \\ b \end{array}\).

All the possible 9 cases of \(H, V\) and their corresponding subwords are listed in Tab. 1.

| \(H\) | dc | dc | dd | dd | cd | cd | ba | bb | ab |
|------|----|----|----|----|----|----|----|----|----|
| \(V\) | c  | c  | d  | d  | d  | d  | a  | b  | b  |
| Incomplete Subword | d  | c  | d  | d  | d  | d  | c  | d  | d  |
| Complete Subword    | d  | c  | d  | d  | d  | d  | c  | d  | d  |

Tab. 1: All the factors of size \((2, 2)\) of \(f_{\infty, \infty}\)
4 Enumeration by Extension

In this section the basic methodology discussed in [4] for enumerating the subwords of 1D infinite Fibonacci word is extended to 2D infinite Fibonacci word. In [4], all the subwords of length \( k \) of \( f_\infty \) were obtained from the subwords of length \( (k - 1) \) by properly extending them to the right by one symbol (Refer Fig. 4). In 1D, each subword of length \( (k - 1) \), except one, will extend to one subword of length \( k \). The exceptional subword of length \( (k - 1) \) which produces two subwords of length \( k \) is called "special". In 2D also we will encounter a similar word.

**Definition 6.** Let \( f_\infty = abab\ldots \) be the 1D Fibonacci word over \( \{a,b\} \). A factor \( w \) of \( f_\infty \) is called special if both \( wa \) and \( wb \) are factors of \( f_\infty \).

![Fig. 4: Extending subwords to the right (Special subwords are underlined)](image)

Before constructing the subwords of \( f_{a,b,\infty} \), let us define a structure called "FRAME".

**Definition 7.** Let \( w \) be a 2D word. The structure obtained by considering only the first row, the first column, the last row and the last column of \( w \) is called the FRAME of \( w \). In particular, the first row (first column, last row, and last column, respectively) is called FRAME\(_T\) (FRAME\(_L\), FRAME\(_B\), and FRAME\(_R\), respectively).

Extending Definition 7 we can have the substructures FRAME\(_{TL}\) (which consists of FRAME\(_T\) and FRAME\(_L\)), FRAME\(_{TR}\), FRAME\(_{LB}\) and FRAME\(_{RB}\). We will be predominantly using FRAME\(_{TL}\) only. Note that FRAME\(_T\) and FRAME\(_L\) share a common prefix of length one. We call this common symbol \( s_{joint,TL} \). Similarly \( s_{joint,TR} \) is defined (Refer Fig. 5).

The following Lemma is inspired by the properties listed in Lemma 1. Note that there are only two distinct rows in \( f_{a,b,\infty} \). These distinct rows also are one and the same words except that their respective alphabets are different. Hence, given the entire first row and any one letter of another row, say \( R \), \( R \) can be written down with ease using a substitution rule.

**Lemma 2.** Given FRAME\(_{TL}\) of a subword of \( f_{a,b,\infty} \) with its FRAME\(_T\) being a subword of length \( l \) of \( f_{a,b}^\infty \) or \( f_{c,d}^\infty \) and FRAME\(_L\) being a subword of length \( k \) of \( f_{a,c}^\infty \) or \( f_{b,d}^\infty \), we can construct a subword of size \((k,l)\) of \( f_{a,b,\infty} \).

**Proof:** Let \( u \) be the 2D word whose FRAME\(_{TL}\) is given. We will make use of the two substitution rules defined in the proof of Theorem 1 to get the corresponding subword of \( f_{a,b,\infty} \).

If FRAME\(_T\) is over \( \{a,b\} \), then define \( \theta_1 : \theta_1(a) = c, \theta_1(b) = d \). Now, for any row \( i, 2 \leq i \leq k \), of FRAME\(_L\), if the letter present therein is \( s_{joint,TL} \), the \( i^{th} \) row of \( u \) is FRAME\(_T\) itself; else, the \( i^{th} \) row of \( u \) is \( \theta_1(FRAME_T) \).
Subwords of 2D Fibonacci words

Fig. 5: Two of the four substructures of FRAME

If $FRAME_T$ is over \{c, d\}, then define $\theta_2 : \theta_2(c) = a, \theta_2(d) = b$. Now, for any row $i$, $2 \leq i \leq k$, of $FRAME_L$, if the letter present therein is $s_{\text{joint}, TL}$, the $i^{th}$ row of $u$ is $FRAME_T$ itself; else, the $i^{th}$ row of $u$ is $\theta_2(FRAME_T)$.

As mentioned in the proof of Theorem 1, the rows other than $FRAME_T$ are constructed using $FRAME_L$. That is the alphabet of a particular row and the order in which the two distinct rows of the subword are arranged are decided by $FRAME_L$. Since $FRAME_L$ is a subword of length $l$ of some column of $f_{\infty, \infty}$, the obtained 2D word is a subword of $f_{\infty, \infty}$ of size $(k, l)$.

\[ \square \]

Remark 4. In Lemma 2 we have constructed the entire subword from $FRAME_{TL}$. Similarly, with appropriate conditions on $FRAME_L$, $FRAME_T$, $FRAME_R$, $FRAME_B$, one can construct the entire subword from any of $FRAME_{LB}, FRAME_{TR}$ and $FRAME_{RB}$ also.

Remark 5. Given a $FRAME_{TL}$ such that its $FRAME_L$ and $FRAME_T$ are vertical and horizontal subwords of length $k$ and $l$ respectively of $f_{\infty, \infty}$ and if $s_{\text{joint}, TL}$ (and hence $FRAME_{TL}$) is positioned at the position $(1, 1)$ of $f_{\infty, \infty}$, then the subword constructed from $FRAME_{TL}$ will be a prefix of size $(k, l)$ of $f_{\infty, \infty}$.

We know that there are $(k + 1)(l + 1)$ subwords of size $(k, l)$ in $f_{\infty, \infty}$ (Refer Preprint or previous section). In the following theorem, we prove that given the $(k + 1)(l + 1)$ subwords of size $(k, l)$ of $f_{\infty, \infty}$, we can extend them to get the $(k + 2)(l + 2)$ subwords of size $(k + 1, l + 1)$.

Theorem 2. Given the $(k + 1)(l + 1)$ subwords of size $(k, l)$ of $f_{\infty, \infty}$, by appropriately extending each subword’s $FRAME_T$ to the right and $FRAME_L$ to the bottom, we can obtain all the $(k + 2)(l + 2)$ subwords of size $(k + 1, l + 1)$ of $f_{\infty, \infty}$.

Proof: Let $u$ be a subword of size $(k, l)$ of $f_{\infty, \infty}$.

Since $f_{\infty, \infty}$ has only two distinct rows (refer Lemma 1), $FRAME_T$ of $u$ will be either over \{a, b\} or over \{c, d\}. Also, a row of $f_{\infty, \infty}$ which is over \{a, b\} (\{c, d\}, respectively), is nothing but the 1D Fibonacci word generated by the morphism $b \rightarrow ba, a \rightarrow b (d \rightarrow dc, c \rightarrow d$, respectively). Thus, if $FRAME_T$ of $u$ is over \{a, b\} (\{c, d\}, respectively), the first row of $u$ is nothing but a factor of length $l$ of the infinite Fibonacci word, $f_{\infty, a}^l = babba \ldots (f_{\infty, c}^l = dcddc \ldots$, respectively). So, $FRAME_T \in H$. 
where $H$ is the union of the $l + 1$ subwords of length $l$ of $f_{\infty}^{b,a}(with one out of them being special) and the $l + 1$ subwords of length $l$ of $f_{\infty}^{a,c}(with one out of them being special). Note that, $|H| = 2(l + 1)$.

Similarly, a column of $f_{\infty,\infty}$ which is over \{b, d\} (\{a, c\}, respectively), can be perceived as a 1D Fibonacci word generated by the morphism $d \to db, b \to d (c \to ca, a \to c$, respectively). Thus, if $FRAME_L$ of $u$ is over \{b, d\} (\{a, c\}, respectively), the first column of $u$ is nothing but a factor of length $k$ of the infinite Fibonacci word, $f_{\infty}^{d,b} = dbdb\ldots (f_{\infty}^{a,c} = cacca\ldots$, respectively). So, $FRAME_L \in V$, where $V$ is the union of the $k + 1$ subwords of length $k$ of $f_{\infty}^{d,b}(with one out of them being special) and the $k + 1$ subwords of length $k$ of $f_{\infty}^{c,a}(with one out of them being special). Note that, $|V| = 2(k + 1)$.

Now let us construct a subword of size $(k + 1, l + 1)$ from $u$. Consider $FRAME_{TL}$ of $u$. We encounter the following four types of $FRAME_{TL}$s.

- **Type(i)**: The word in $FRAME_T$ and the word in $FRAME_L$ both are not special
- **Type(ii)**: The word in $FRAME_T$ is not special and the word in $FRAME_L$ is special
- **Type(iii)**: The word in $FRAME_T$ is special and the word in $FRAME_L$ is not special
- **Type(iv)**: The word in $FRAME_T$ and the word in $FRAME_L$ both are special

Let $Alph_T = \text{Alphabet of } FRAME_T$ and let $Alph_L = \text{Alphabet of } FRAME_L$.

In **Type(i)**, we can extend $FRAME_T$ to the right by appending one appropriate letter from $Alph_T$ and can extend $FRAME_L$ to the bottom by appending one appropriate letter from $Alph_L$. Thus we get $FRAME_{TL}^{new}$ with a new $FRAME_L$, say $FRAME_L^{new}$, with $(k + 1)$ rows, and a new $FRAME_T$, say $FRAME_T^{new}$, with $(l + 1)$ columns. Note that, by the way $FRAME_{TL}$ is extended to $FRAME_{TL}^{new}$, $FRAME_{TL}^{new}$ will be $FRAME_{TL}$ of a subword of size $(k + 1, l + 1)$ of $f_{\infty,\infty}$. Now by Lemma 2, we can construct the subword of size $(k + 1)(l + 1)$ corresponding to this $FRAME_{TL}^{new}$.

In **Type(ii)**, we can extend $FRAME_T$ to the right by appending one appropriate letter from $Alph_T$. But, $FRAME_L$ being special, will produce two $FRAME_{TL}^{new}$s. Thus we get $FRAME_{TL}^{new1}$ consisting of $FRAME_L^{new1}$ with $FRAME_T^{new}$ and $FRAME_{TL}^{new2}$ consisting of $FRAME_L^{new2}$ with $FRAME_T^{new}$. Each one will be $FRAME_{TL}$ of a subword of size $(k + 1, l + 1)$ of $f_{\infty,\infty}$. Again by Lemma 2, we can construct two distinct subwords of size $(k + 1)(l + 1)$ from this type of $FRAME_{TL}$.

Similarly, a $FRAME_{TL}$ of **Type(iii)** will produce two distinct subwords of size $(k + 1)(l + 1)$ and a $FRAME_{TL}$ of **Type(iv)** will produce four distinct subwords of size $(k + 1)(l + 1)$.

We now prove that exactly $(k + 2)(l + 2)$ subwords only are produced by this method.

Towards that, let us analyse the $FRAME_{TL}$s of the $(k + 1)(l + 1)$ factors of size $(k, l)$ which are being extended. Let us recall how these factors of size $(k, l)$ are created (preprint). We know that, in the $(2k + 1)$ vertical factors of size $k$, available in the columns of $f_{\infty,\infty}$, $k + 1$ are over \{a, c\} and $k + 1$ are over \{b, d\}. Let us name the factors over \{a, c\} as $V_1^{a,c}, V_2^{a,c}, \ldots, V_k^{a,c}, V_{k+1}^{a,c}$, with the last one being the special factor over \{a, c\}. As the two distinct columns occurring in $f_{\infty,\infty}$ are structurally the same, the other $(k + 1)$ factors over \{b, d\}, namely, $V_1^{b,d}, V_2^{b,d}, \ldots, V_k^{b,d}, V_{k+1}^{b,d}$, are obtained by the simple substitution $s(a) = b, s(c) = d$ so that $V_1^{b,d} = s(V_1^{a,c}), V_2^{b,d} = s(V_2^{a,c}), \ldots, V_{k+1}^{b,d} = s(V_{k+1}^{a,c})$, with the last one being the special factor over \{b, d\}. Then the $(k + 1)$ pairs $V_i = \{V_i^{a,c}, V_i^{b,d}\}$, $1 \leq i \leq (k + 1)$ are created. Similarly the $(l + 1)$ pairs $H_j = \{H_j^{b,d}, H_j^{c,d}\}$, $1 \leq j \leq (l + 1)$ are created. Here $H_j^{b,d}$, $1 \leq j \leq (l + 1)$ are the $l + 1$ horizontal factors over \{a, b\} with the last one being the special factor over \{a, b\} and $H_j^{c,d}$, $1 \leq j \leq (l + 1)$ are the $l + 1$ horizontal factors over \{c, d\} with the last one being the special factor over \{c, d\}.

The nature of $FRAME_{TL}$s and hence the type of each of the $(k + 1)(l + 1)$ subwords depends on
Subwords of 2D Fibonacci words

the possible ways the factors from \( V_i \) and \( H_j \) occupy \( FRAME_L \) and \( FRAME_T \) of the subwords. These features are listed in Tab. 2. In the table, the cell at the intersection of \( V_i, 1 \leq i \leq k + 1 \) and \( H_j, 1 \leq j \leq l + 1 \) corresponds to the subword having \( FRAME_L \in V_i \) and \( FRAME_T \in H_j \). Also, \( F_L \) means \( FRAME_L \) is special and \( \neg F_L \) means \( FRAME_L \) is not special. \( F_T \) and \( \neg F_T \) have similar meanings.

Tab. 2: Distribution of the Types of \( FRAME_T \) in the subwords of size \((k, l)\)

| \( H_i \rightarrow V_i \) | \( H_1 \) | \( H_2 \) | ... | \( H_i \) | \( H_{i+1} \) |
|-------------------------|-----|-----|-----|-----|-----|
| \( V_1 \)               | \( \neg F_L \) and \( \neg F_T \) Type (i) | \( \neg F_L \) and \( \neg F_T \) Type (i) | ... | \( \neg F_L \) and \( \neg F_T \) Type (i) | \( \neg F_L \) and \( F_T \) Type (iii) |
| \( V_2 \)               | \( \neg F_L \) and \( \neg F_T \) Type (i) | \( \neg F_L \) and \( \neg F_T \) Type (i) | ... | \( \neg F_L \) and \( \neg F_T \) Type (i) | \( \neg F_L \) and \( F_T \) Type (iii) |
| ...                     | ... | ... | ... | ... | ... |
| \( V_k \)               | \( \neg F_L \) and \( \neg F_T \) Type (i) | \( \neg F_L \) and \( \neg F_T \) Type (i) | ... | \( \neg F_L \) and \( \neg F_T \) Type (i) | \( \neg F_L \) and \( F_T \) Type (iii) |
| \( V_{k+1} \)           | \( F_L \) and \( \neg F_T \) Type (ii) | \( F_L \) and \( \neg F_T \) Type (ii) | ... | \( F_L \) and \( \neg F_T \) Type (ii) | \( F_L \) and \( F_T \) Type (iv) |

From the table, it is clear that, by extending their \( FRAME_{TL} \), each of the \( kl \) number of subwords of size \((k, l)\) produce 1 subword of size \((k + 1, l + 1)\), each of the \( k + l \) number of subwords of size \((k, l)\) produce 2 subwords of size \((k + 1, l + 1)\) and one subword of size \((k, l)\) produces 4 subwords of size \((k + 1, l + 1)\). Therefore, \( kl + 2(k + 1) + 4 = (k + 2)(l + 2) \) subwords of size \((k + 1, l + 1)\) will be produced.

Example 2. Consider all the 9 subwords of size \((2, 2)\) of \( f_{∞,∞} \) listed in Table 1. We will extend each of them to get the 16 subwords of size \((3, 3)\) of \( f_{∞,∞} \). The 16 subwords are listed in Tab. 2.

5 Enumeration by Conjugation

For a given \( k \), let \( n \) be such that \( 1 \leq k < F(n) \), where \( F(n) \) is the \( n^{\text{th}} \) Fibonacci number. In this section we use the method described in [11], wherein it is proved that the prefixes of length \( k \) of the conjugates of a "special" conjugate of \( f_n \) are the subwords of length \( k \) of \( f_{∞} \). The Lemma is recalled here.

With \( \Sigma \), an alphabet, define the operator \( T \) on \( \Sigma^+ \) as follows. For a word \( w = a_1a_2 \ldots a_n \in \Sigma^+ \), \( T(a_1a_2 \ldots a_{n-1}a_n) = a_2 \ldots a_{n-1}a_1a_2 \ldots a_n \) and \( T^{-1}(a_1a_2 \ldots a_{n-1}a_n) = a_n a_1 a_2 \ldots a_{n-1} \). Higher powers of \( T \) are defined iteratively. That is, \( T^p(w) = T(T^{p-1}(w)) \) and \( T^{-p}(w) = T^{-1}(T^{-(p-1)}(w)) \).

Lemma 3. [11] Let \( f_0 = a, f_1 = b, f_n = f_{n-1}f_{n-2}, n \geq 2 \) be the sequence of Fibonacci words. Let \( F(n) = |f_n| \) and let

\[
q_n = \begin{cases} 
T^{F(n)-1}(f_n) & \text{if } n \text{ is even} \\
T^{F(n)-1}(f_{n-1}) & \text{if } n \text{ is odd}.
\end{cases}
\]
Tab. 3: Obtaining the factors by Extension

| Subword of Size (2,2) | Type | Extended FRAME$^\text{new}_{TL}$ | Subword of Size (3,3) |
|-----------------------|------|----------------------------------|----------------------|
| \(d\ c\ b\ a\)      | (i)  | \(d\ c\ d\)                       | \(d\ c\ d\)         |
| \(d\ c\ d\)          | (i)  | \(b\ *\ *\)                       | \(b\ a\ b\)         |
| \(d\ c\ d\)          | (i)  | \(d\ *\ *\)                       | \(d\ c\ d\)         |
| \(d\ c\ d\)          | (i)  | \(b\ *\ *\)                       | \(b\ a\ b\)         |
| \(d\ d\ b\ b\)       | (i)  | \(d\ d\ c\)                       | \(d\ d\ c\)         |
| \(d\ d\ d\)          | (i)  | \(b\ *\ *\)                       | \(d\ d\ c\)         |
| \(c\ d\ a\ b\)       | (iii)| \(c\ d\ c\)                       | \(c\ d\ c\)         |
| \(c\ d\ c\)          | (iii)| \(a\ *\ *\), \(a\ *\ *\)         | \(a\ b\ a\), \(a\ b\ b\) |
| \(c\ d\ c\)          | (iii)| \(c\ *\ *\), \(c\ *\ *\)         | \(c\ d\ c\), \(c\ d\ d\) |
| \(c\ d\ c\)          | (iii)| \(a\ *\ *\), \(a\ *\ *\)         | \(a\ b\ a\), \(a\ b\ b\) |
| \(b\ a\ d\ c\)       | (ii) | \(b\ a\ b\)                       | \(b\ a\ b\)         |
| \(b\ a\ b\)          | (ii) | \(d\ *\ *\), \(d\ *\ *\)         | \(d\ c\ d\), \(d\ c\ d\) |
| \(b\ a\ b\)          | (ii) | \(b\ *\ *\), \(b\ *\ *\)         | \(b\ a\ b\)         |
| \(b\ b\ d\ d\)       | (ii) | \(b\ b\ a\)                       | \(b\ b\ a\)         |
| \(b\ b\ a\)          | (iv) | \(a\ b\ a\)                       | \(a\ b\ a\)         |
| \(a\ b\ a\)          | (iv) | \(c\ *\ *\), \(c\ *\ *\)         | \(c\ d\ c\), \(c\ d\ c\) |
| \(a\ b\ a\)          | (iv) | \(a\ *\ *\), \(a\ *\ *\)         | \(a\ b\ a\), \(a\ b\ c\) |
| \(a\ b\ b\)          | (iv) | \(a\ b\ a\)                       | \(a\ b\ b\)         |
| \(c\ *\ *\), \(c\ *\ *\) | (iv) | \(c\ d\ d\), \(c\ d\ d\)         | \(a\ b\ b\)         |
| \(a\ *\ *\), \(a\ *\ *\) | (iv) | \(a\ b\ b\)                       | \(a\ b\ c\)         |
Then for each $k$ with $1 \leq k < F(n)$, the $k+1$ prefixes of $T^0(q_n), T^{-1}(q_n), \ldots, T^{-k}(q_n)$ having length $k$ are the $k+1$ distinct factors of $f_\infty$ of length $k$.

**Example 3.** Let $f_\infty = abaab$. For $k = 4$, $n$ will be 4, as $4 < F(4)$. So, $f_n = f_4 = abaab$. With $F(4) = 5, F(3) = 3$, we have $q_4 = T^4(abaab) = babaa$, is the special conjugate of $f_4$. Now, $T^0(q_4), T^{-1}(q_4), T^{-2}(q_4), T^{-3}(q_4), T^{-4}(q_4)$ are babaa, babaa, babaa, ababa, ababa respectively and the subwords of $f_\infty$ of length 4 are baba, abab, abaa, babaa.

Similar to the operators $T$ and $T^{-1}$, we define four operators on 2D words.

**Definition 8.** Let $r_1, r_2, \ldots, r_m$ and $c_1, c_2, \ldots, c_n$ be the $m$ rows and the $n$ columns of a 2D word $w$ of size $(m, n)$. Then the operators $T_{\text{col}}(w)$, $T_{\text{col}}^{-1}(w)$, $T_{\text{row}}(w)$ and $T_{\text{row}}^{-1}(w)$ are defined as below:

\[
T_{\text{col}}(w) = c_2 \odot c_3 \odot \cdots \odot c_n \odot c_1 \\
T_{\text{col}}^{-1}(w) = c_n \odot c_1 \odot c_2 \odot \cdots \odot c_{n-2} \odot c_{n-1} \\
T_{\text{row}}(w) = r_2 \odot r_3 \odot \cdots \odot r_m \odot r_1 \\
T_{\text{row}}^{-1}(w) = r_n \odot r_1 \odot r_2 \odot \cdots \odot r_{n-2} \odot r_{n-1}.
\]

Higher powers of $T_{\text{col}}(w)$, $T_{\text{col}}^{-1}(w)$, $T_{\text{row}}(w)$ and $T_{\text{row}}^{-1}(w)$ are defined iteratively. For example, with $s \geq 1, T_{\text{col}}^s(w) = T_{\text{col}}(T_{\text{col}}^{s-1}(w))$.

Through these operators we define the conjugacy class of a 2D word $w$.

**Definition 9.** Let $w$ be a 2D word of size $(m, n)$. Then

\[
\text{Conj}(w) = \left\{ T_{\text{row}}^i T_{\text{col}}^j(w), 0 \leq i \leq m-1, 0 \leq j \leq n-1 \right\} \\
= \left\{ T_{\text{col}}^i T_{\text{row}}^j(w), 0 \leq j \leq n-1, 0 \leq i \leq m-1 \right\}
\]

is called the Conjugacy Class of $w$.

Since $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$, it is easy to see that the number of conjugates of $w$ can be at the maximum $mn$. Note that, if $w$ is primitive then the maximum possible value of $mn$ will be achieved by $|\text{Conj}(w)|$.

Now, we will enumerate the subwords of size $(k, l)$ of $f_{\infty, \infty}$ using the conjugates of a "special" conjugate of $f_{m,n}$ ($m, n \geq 3$ and depend on $k, l$).

**Theorem 3.** Let $F(0) = F(1) = 1, F(2) = 2, F(3) = 3, F(4) = 5, \ldots$ be the sequence of Fibonacci numbers. For a given $k, l \geq 1$, consider the 2D finite Fibonacci word $f_{m,n}$, where $m, n$ are such that $k < F(m)$ and $l < F(n)$. Let

\[
q_{m,n} = \begin{cases} 
T_{\text{row}}^{F(m)-1} T_{\text{col}}^{F(n)-1}(f_{m,n}) & \text{if } m \text{ is even and } n \text{ is even} \\
T_{\text{row}}^{F(m)-1} T_{\text{col}}^{F(n)-1}(f_{m,n})^{-1} & \text{if } m \text{ is even and } n \text{ is odd} \\
T_{\text{row}}^{F(m)-1} T_{\text{col}}^{F(n)-1}(f_{m,n}) & \text{if } m \text{ is odd and } n \text{ is even} \\
T_{\text{row}}^{F(m)-1} T_{\text{col}}^{F(n)-1}(f_{m,n})^{-1} & \text{if } m \text{ is odd and } n \text{ is odd}
\end{cases}
\]

Then for each $k$ with $1 \leq k < F(m)$ and for each $l$ with $1 \leq l < F(n)$, the $(k+1)(l+1)$ prefixes of
We can call this the prefixes of length \(k\) of \(f\), arranged in such a way that, for each \(i\), the prefixes of length \(k\) of the first \(i\) rows of \(f\) are distinct factors of \(f_{\infty, \infty}\) of size \((k, l)\).

**Proof:** Suppose that we want to find all the subwords of \(f_{\infty, \infty}\) of size \((k, l)\). Let \(F(0) = F(1) = 1, F(2) = 2, F(3) = 3, F(4) = 5, \ldots\) be the sequence of Fibonacci numbers. Consider the 2D finite Fibonacci word \(f_{m, n}\) where \(m\) and \(n\) are such that \(k < F(m)\) and \(l < F(n)\). Note that \(f_{m, n}\) will be of size \((F(m), F(n))\).

We prove the theorem for the case where both \(m\) and \(n\) are even. The proofs of other cases are similar.

Denote the columns of \(f_{m, n}\) by \(C_1, C_2, \ldots, C_{F(n)}\). Since there are only two distinct columns (refer Lemma 1), let us symbolize the columns over \([b, d]\) by \(D\) and the columns over \([a, c]\) by \(C\). As every row of \(f_{m, n}\) is a Fibonacci word of size \(F(n)\), the two distinct columns are indeed arranged in a Fibonacci pattern in \(f_{m, n}\). That is, the symbolized word for \(f_{m, n}\) is \(C_1 \oplus C_2 \oplus \ldots \oplus C_{F(n)} = DCDDC \ldots DC = H_n\), say, is a Fibonacci word of size \(F(n)\). Since \(n\) is even, the suffix of length 2 of \(H_n\) will be \(DC\). Now by Lemma 3, the prefixes of length \(l\) of the conjugates of \(T^{F(n)-1}(H_n) = q_m\) (say), are the subwords of length \(l\) of \(H_n\). We now replace the symbols \(D\) and \(C\) occurring in \(q_m\) by the original columns to get the 2D word \(q_m\). What we have proved is that, we can arrange the columns of \(f_{m, n}\) in a way that we can obtain all the subwords of length \(l\) of the infinite Fibonacci words occupying the rows of \(f_{m, n}\) through the conjugates of \(q_m\).

Now, let us denote the rows of \(f\) by \(R_1, R_2, \ldots, R_{F(m)}\). By symbolizing the rows over \([d, c]\) as \(D'\) and \([a, b]\) as \(B\), we get \(V_m\), the symbolized word of \(q_m\) over \([D', B]\) as \(V_m = R_1 \oplus R_2 \oplus \ldots \oplus R_{F(m)} = D'B'D'B \ldots D'B\). Following a similar argument as above, we get a word \(q_m' = T^{F(m)-1}(V_m)\) over \([D', B]\). We can now replace the symbols occurring in \(q_m'\) to get the 2D word \(q_m\). What we have proved is that, we can arrange the columns of \(q_m\) in a way that we can get all the subwords of length \(k\) of the infinite Fibonacci words occupying the columns of \(f_{m, n}\) through the conjugates of \(q_m\).

Note that \(q_{m, n}\) is a conjugate of \(f_{m, n}\). In fact, by the two stage process, what we have obtained as \(q_{m, n}\) is nothing but \(T_{row}^{F(m)-1}(T_{col}^{F(n)-1}(f_{m, n}))\). As assured by Lemma 3, the rows and columns of \(q_{m, n}\) are arranged in such a way that, for each \(0 \leq i \leq k, 0 \leq j \leq l\), the prefixes of length \(k\) of the first \(i\) and the prefixes of length \(l\) of the first \(j\) of the rows of \(T_{row}^iT_{col}^j(q_{m, n})\), produces \((k + 1)(l + 1)\) distinct \(\text{FRAME}_L\)s. We can call \(q_{m, n}\) a “special” conjugate of \(f_{m, n}\), in this context. Since in each of these \(\text{FRAME}_L\)s, \(\text{FRAME}_L\) are subwords of \(f^{\infty, b}_L\) or \(f^{\infty, a}_L\), and \(\text{FRAME}_T\)s are subwords of \(f^{\infty, c}_T\) or \(f^{\infty, a}_T\), by Lemma 2 we get \((k + 1)(l + 1)\) distinct subwords of \(f_{\infty, \infty}\).

Let us understand the enumeration of the subwords through an example.

**Example 4.** Let \(k = 2, l = 2\). That is, we wish to find all the \((2, 2)\)-subwords of \(f_{\infty, \infty}\). As \(k\) and \(l\) are less than \(F(3) = 3\), \(m = n = 3\) and we consider

\[
\begin{align*}
d & c & d \\
D & a & b \\
d & c & d
\end{align*}
\]
Then, as both $m$ and $n$ are odd, $q_{3,3} = T^{-1}_{\text{row}}(T^{-1}_{\text{col}}(f_{3,3})) = f_{3,3} = \begin{bmatrix} a & b & b \\ c & d & d \\ c & d & d \end{bmatrix}$.

As mentioned earlier $q_{3,3}$ is a "special" conjugate of $f_{3,3}$. Since $f_{m,n}, m, n \geq 0$ are primitive, $q_{3,3}$ is primitive and has 9 conjugates. All the 9 conjugates and the corresponding subword of size $(2, 2)$ of $f_{\infty, \infty}$ are listed in Table 4.

| $T^i_{\text{row}}(T^j_{\text{col}}(q_{3,3}))$ | Conjugate of $q_{3,3}$ | Subword |
|---------------------------------------------|------------------------|---------|
| $T^0_{\text{row}}(T^0_{\text{col}}(q_{3,3}))$ | $\begin{bmatrix} a & b & b \\ c & d & d \\ c & d & d \end{bmatrix}$ | $a \ b \ b$ |
| $T^0_{\text{row}}(T^{-1}_{\text{col}}(q_{3,3}))$ | $\begin{bmatrix} b & a & b \\ d & c & d \\ d & c & d \end{bmatrix}$ | $b \ a \ a$ |
| $T^0_{\text{row}}(T^{-2}_{\text{col}}(q_{3,3}))$ | $\begin{bmatrix} b & b & a \\ d & d & c \\ d & d & c \end{bmatrix}$ | $b \ b \ b$ |
| $T^{-1}_{\text{row}}(T^0_{\text{col}}(q_{3,3}))$ | $\begin{bmatrix} c & d & d \\ a & b & b \\ c & d & d \end{bmatrix}$ | $c \ d \ d$ |
| $T^{-1}_{\text{row}}(T^{-1}_{\text{col}}(q_{3,3}))$ | $\begin{bmatrix} d & c & d \\ b & a & b \\ d & c & d \end{bmatrix}$ | $d \ c \ b$ |
| $T^{-1}_{\text{row}}(T^{-2}_{\text{col}}(q_{3,3}))$ | $\begin{bmatrix} d & d & c \\ b & b & a \\ d & d & c \end{bmatrix}$ | $d \ d \ b$ |
| $T^{-2}_{\text{row}}(T^0_{\text{col}}(q_{3,3}))$ | $\begin{bmatrix} c & d & d \\ c & d & d \\ a & b & b \end{bmatrix}$ | $c \ d \ d$ |
| $T^{-2}_{\text{row}}(T^{-1}_{\text{col}}(q_{3,3}))$ | $\begin{bmatrix} d & c & d \\ d & c & d \\ b & a & b \end{bmatrix}$ | $d \ c \ b$ |
| $T^{-2}_{\text{row}}(T^{-2}_{\text{col}}(q_{3,3}))$ | $\begin{bmatrix} d & d & c \\ d & d & c \\ b & b & a \end{bmatrix}$ | $d \ d \ d$ |

In [11], apart from the sophisticated way of obtaining the subwords of $f_{\infty}$, described in Lemma 3, the author provides another simple way of obtaining the subwords of length $k$.

**Proposition 1.** ([11]) Let $n \geq 2$ and $F(n) \leq k < F(n + 1)$. Then the $k + 1$ prefixes of $T^i(f_{n+1})$ of length...
Proposition 1 is extended to $f_{\infty}$ as below.

**Proposition 2.** Let $m,n \geq 2$ and $F(m) \leq k < F(m+1)$, $F(n) \leq l < F(n+1)$. Then the $(k+1)(l+1)$ prefixes of $T_{\text{row}}(T_{\text{col}}(f_{m+n+1}))$ of size $(k,l)$, where $i \in \{0,1,\ldots,F(m)-1\} \cup \{F(m+2)-k-1, F(m+2)-k, \ldots, F(m+1)-1\}$, $j \in \{0,1,\ldots,F(n)-1\} \cup \{F(n+2)-l-1, F(n+2)-l, \ldots, F(n+1)-1\}$, are the $(k+1)(l+1)$ distinct factors of $f_{\infty}$ of size $(k,l)$.

**Proof:** For $m,n \geq 2$, consider the 2D finite Fibonacci word $f_{m+n+1}$. Recall that the columns and rows of $f_{m+n+1}$ are 1D finite Fibonacci words (in fact, they are $f_{m+1}$ or $f_{n+1}$). Hence, the $\text{FRAME}_L$s and $\text{FRAME}_T$s of the $(k+1)(l+1)$ number of $\text{FRAME}_T$Ls obtained from the conjugates $T_{\text{row}}(T_{\text{col}}(f_{m+n+1}))$, $i \in \{0,1,\ldots,F(m)-1\} \cup \{F(m+2)-k-1, F(m+2)-k, \ldots, F(m+1)-1\}$, $j \in \{0,1,\ldots,F(n)-1\} \cup \{F(n+2)-l-1, F(n+2)-l, \ldots, F(n+1)-1\}$ are nothing but the $(k+1)(l+1)$ appropriate combinations of vertical factors of length $k$ and horizontal factors of length $l$ of the infinite Fibonacci words occurring in the columns and in the rows of $f_{\infty}$. Since all these $\text{FRAME}_T$Ls are taken from the prefixes of $f_{\infty}$, the subword constructed from these $\text{FRAME}_T$Ls (refer Lemma 2) will be obviously prefixes of $f_{\infty}$. Hence, the prefixes of size $(k,l)$ of $T_{\text{row}}(T_{\text{col}}(f_{m+n+1}))$ for the stated values of $i,j$ are the factors of $f_{\infty}$ of size $(k,l)$.

### 6 Locating the Factors of $f_{\infty}$

In the previous sections, we developed a few methods for listing all the $(k+1)(l+1)$ factors of size $(k,l)$ of $f_{\infty}$. In this section we will locate (find the exact positions $(i,j)$ of) these factors in the domain of $f_{\infty}$. We know that since there are only $k+1$ factors of length $k$ in $f_{\infty}$, there are many repetitions of every factor in $f_{\infty}$ [4]. As the rows and columns of $f_{\infty}$ are composed of $f_{\infty}$, the same thing happens in $f_{\infty}$ also.

For locating the factors of $f_{\infty}$, the reader may either refer [11] or [26]. We recall some terminologies from [26] for our use.

Let $f_0 = a, f_1 = ab$ and for $n \geq 1, f_{n+1} = f_n f_{n-1}$ so that $f_{\infty} = abaababaababaab\ldots = f_{\infty}(1,2,3,\ldots)$. Also, for $n \geq 0$ let $F(n) = |f_n|$ be the $n^{\text{th}}$ Fibonacci number. For $n \geq 2$, let $g_n$ be the $n^{\text{th}}$ truncated Fibonacci word, the word obtained from $f_n$ by removing its last two letters.

Let $u$ be a subword of $f_{\infty}$. By an occurrence of $u$ we mean a $i \geq 0$ such that $f_{\infty}(i+1) f_{\infty}(i+2)\ldots f_{\infty}(i+|u|) = u$. By first-occ($u$) we mean the least value of occurrence of $u$ and by occ($u$) we mean the set of all occurrences of $u$ in $f_{\infty}$. Now, for a set of integers $X$ and for a $j \geq 0$, define the operator $\ominus$ as, $X \ominus j = \{x + j : x \in X\}$.

Recall that the Fibonacci number system represents a number as a sum of Fibonacci numbers such that no two consecutive Fibonacci numbers are used. Also, the sum of zero number of integers equals zero. This representation of any nonnegative integer $n$, in the Fibonacci number system is called the Fibonacci representation of $n$. For $n \geq 1$, let $Z_n$ be the set of nonnegative integers which do not use Fibonacci numbers $F(0), F(1), F(2), \ldots, F(n-1)$ in their Fibonacci representation. For example $Z_1 = \{0,2,3,5,\ldots\}$ and $Z_2 = \{0,3,5,8,11,\ldots\}$. Then, it is proved in [26] that,

$$\text{occ}(u) = \text{occ}(g_n) \ominus \text{first-occ}(u),$$
Subwords of 2D Fibonacci words

where $n$ is such that $g_n$ is the shortest truncated Fibonacci word containing $u$. Since for $n \geq 2$, $occ(g_{n+1}) = occ(f_n) = Z_n$, we have,

$$occ(u) = Z_{n-1} \square first-occ(u).$$

We also have that $occ(f_1) = occ(f_2)$ and $occ(f_0) = Z_1$.

**Example 5.** Let us locate the positions of the factor $u = abab$ in $f_{a,b}^n$. We have first-occ($u$) = 3. Since $u$ occurs for the first time in $g_3$, we get $n = 5$ and hence $occ(abab) = Z_4 \square 3 = \{0, 8, 13, 21, 29, \ldots \} \square 3 = \{3, 11, 16, 24, 32, \ldots \}$.

We are now ready to locate any factor of $f_{\infty, \infty}$. Let $w$ be a subword of $f_{\infty, \infty}$. Let the size of $w$ be $(k, l)$. Note that, because $w$ is a 2D word, first-occ($w$) is a pair $(i, j)$ such that $w$ occurs in the domain $\{i + 1, i + 2, \ldots, i + k\} \times \{j + 1, j + 2, \ldots, j + l\}$ of $f_{\infty, \infty}$. The definition of $occ(w)$ is similar to its 1D counterpart. Since a subword of $f_{\infty, \infty}$ is uniquely determined by its FRAME$_T$, its first occurrence and hence its all other occurrences will be determined by the first occurrences of its FRAME$_T$ and FRAME$_L$. With FRAME$_T$ and FRAME$_L$ both being 1D Fibonacci words we have the following Proposition.

**Proposition 3.** Let $w$ be a subword of $f_{\infty, \infty}$. Let FRAME$_T$ and FRAME$_L$ denote its first row and first column respectively. Then,

$$first-occ(w) = \begin{cases} (f_{o_{L}^{d,b}}, f_{o_{T}^{d,c}}) & \text{if FRAME$_L$ is over } \{d, b\} \text{ and FRAME$_T$ is over } \{d, c\} \\ (f_{o_{L}^{d,b}}, f_{o_{T}^{b,a}}) & \text{if FRAME$_L$ is over } \{d, b\} \text{ and FRAME$_T$ is over } \{b, a\} \\ (f_{o_{L}^{c,a}}, f_{o_{T}^{d,c}}) & \text{if FRAME$_L$ is over } \{c, a\} \text{ and FRAME$_T$ is over } \{d, c\} \\ (f_{o_{L}^{c,a}}, f_{o_{T}^{b,a}}) & \text{if FRAME$_L$ is over } \{c, a\} \text{ and FRAME$_T$ is over } \{b, a\} \end{cases}$$

where $f_{o_{L}^{d,b}}$ is the first-occ(FRAME$_L$) in $f_{\infty, \infty}^{d,b}$, $f_{o_{L}^{c,a}}$ is the first-occ(FRAME$_L$) in $f_{\infty, \infty}^{c,a}$, $f_{o_{T}^{d,c}}$ is the first-occ(FRAME$_T$) in $f_{\infty, \infty}^{d,c}$ and $f_{o_{T}^{b,a}}$ is the first-occ(FRAME$_T$) in $f_{\infty, \infty}^{b,a}$.

**Proof:** We discuss the proof for the case in which FRAME$_L$ is over $\{d, b\}$ and FRAME$_T$ is over $\{d, c\}$. Proofs of the other cases are similar.

Since $w$ can occur in $f_{\infty, \infty}$, only when FRAME$_{TL}$ of $w$ occurs in $f_{\infty, \infty}$, it is clear that first-occ($w$) is decided by first-occ(FRAME$_L$) in $f_{\infty, \infty}^{d,b}$ and first-occ(FRAME$_T$) in $f_{\infty, \infty}^{d,c}$. Let $f_{o_{L}^{d,b}} \geq 0$, be first-occ(FRAME$_L$) in $f_{\infty, \infty}^{d,b}$. Let $f_{o_{T}^{d,c}} \geq 0$ be first-occ(FRAME$_T$) in $f_{\infty, \infty}^{d,c}$. Since all the columns of $f_{\infty, \infty}$ over $\{d, b\}$ are identical, $f_{o_{L}^{d,b}}$ value will be the same in all the columns which are over $\{d, b\}$. So in the $f_{o_{L}^{d,b}}^{th}$ column (where FRAME$_T$($w$) occurs for the first time) also, $f_{o_{L}^{d,b}}$ will be the same. Similarly, since all the rows of $f_{\infty, \infty}$ over $\{d, c\}$ are identical, $f_{o_{L}^{d,b}}$ value will be the same in all the rows which are over $\{d, c\}$. Now, first-occ(FRAME$_{TL}$($w$)) can be $(i, j)$, say, only when both first-occ(FRAME$_L$($w$)) and first-occ(FRAME$_T$($w$)) are $(i, j)$. Hence first-occ($w$) = first-occ(FRAME$_{TL}$($w$)) = $(f_{o_{L}^{d,b}}, f_{o_{T}^{d,c}})$, if FRAME$_L$ is over $\{d, b\}$ and FRAME$_T$ is over $\{d, c\}$. □

**Corollary 2.** Let $w$ be a subword of $f_{\infty, \infty}$. Let first-occ($w$) be given by Proposition 3. Let FRAME$_L$($w$) be over $\{s_1, s_2\}$ and FRAME$_T$($w$) be over $\{s_1', s_2'\}$. Then, occ($w$) = $X \times Y$, where $X = Z_{m-1} \square f_{o_{L}^{s_1, s_2}}$ and $Y = Z_{n-1} \square f_{o_{T}^{s_1', s_2'}}$. 

Proof: In all the columns of $f_{\infty,\infty}$ which are over $\{s_1, s_2\}$, $\text{occ}(\text{FRAME}_L) = Z_{m-1} \sqcup f_0^{s_1, s_2}$, where $m$ is such that $g_m$ is the shortest truncated Fibonacci word over $\{s_1, s_2\}$ containing $\text{FRAME}_L$. Similarly, in all the rows of $f_{\infty,\infty}$ which are over $\{s'_1, s'_2\}$, $\text{occ}(\text{FRAME}_T) = Z_{n-1} \sqcup f_0^{s'_1, s'_2}$, where $n$ is such that $g_n$ is the shortest truncated Fibonacci word over $\{s'_1, s'_2\}$ containing $\text{FRAME}_T$. Therefore, $\text{occ}(\text{FRAME}_{TL}(w)) = (\text{occ}(\text{FRAME}_L(w)), \text{occ}(\text{FRAME}_T(w))) = \{(x, y) : x \in Z_{m-1} \sqcup f_0^{s_1, s_2}, y \in Z_{n-1} \sqcup f_0^{s'_1, s'_2}\} = X \times Y$ where $X = Z_{m-1} \sqcup f_0^{s_1, s_2}$ and $Y = Z_{n-1} \sqcup f_0^{s'_1, s'_2}$. Since $\text{occ}(w) = \text{occ}(\text{FRAME}_{TL}(w))$, the result follows.

Example 6. Let us find the $\text{occ}(w)$ where $w = d \ d \ c$. $b \ b \ a$

Note that, $\text{FRAME}_L$ of $w$ is $d$ and first-occ($\text{FRAME}_L$) in $f_{\infty}^{d, b}$ is 2. That is $f_0^{d, b} = 2$. Also the value of $m$ such that $g_m^{d, b}$ contains $\text{FRAME}_L$ is 4. Similarly, $\text{FRAME}_T$ of $w$ is “$d \ d \ c$” and first-occ($\text{FRAME}_T$) in $f_{\infty}^{d, c}$ is $f_0^{d, c} = 2$. The value of $n$ such that $g_n^{d, c}$ contains $\text{FRAME}_T$ is 4.

Therefore, first-occ($w$) = (2, 2). And, $\text{occ}(w) = X \times Y$, where $X = Z_2 \sqcup 2$ and $Y = Z_2 \sqcup 2$. With $Z_2 = \{0, 5, 8, 13, 18, \ldots\}$, we have $X = \{2, 7, 10, 15, 20, \ldots\}$ and $Y = \{2, 7, 10, 15, 20, \ldots\}$. Hence $\text{occ}(w) = \{(2, 2), (2, 7), (2, 10), \ldots\}$.

7 Concluding Remarks

The knowledge of all the subwords of an infinite word would be very useful to analyse the characteristics of the word. Though any sort of analysis like periodicity, factor complexity is tricky in 2D words, 2D Fibonacci words with their simple and elegant structure are pliable for exploring their properties. In this paper we have enumerated the subwords of the 2D infinite Fibonacci word, $f_{\infty,\infty}$, in a few possible ways. The location of the occurrences of these subwords are also found out.

Suffix tree is an important tool used for pattern matching and dictionary searching [23][29]. Again, there are some limitations while extending this tool for 2D words [15]. But the relatively simpler structure of $f_{\infty,\infty}$ may help us to develop one for 2D words of similar type. Also, variations attempted in the generation of the Fibonacci sequence [30] lead to variants of 1D/2D Fibonacci words [17]. We might start exploring these directions.

References

[1] Aldo de Luca: A combinatorial property of the Fibonacci words. Information Processing Letters 12(4), 193–195 (1981)

[2] Anselmo, M., Giammarresi, D., Madonia, M.: Prefix picture codes: a decidable class of two-dimensional codes. International Journal of Foundations of Computer Science 25(08), 1017–1031 (2014)

[3] Apostolico, A., Brimkov, V.E.: Fibonacci arrays and their two-dimensional repetitions. Theoretical Computer Science 237(1-2), 263–273 (2000)
Subwords of 2D Fibonacci words

[4] Berstel, J.: Fibonacci words - a survey. In: Rozenberg, G., Salomaa, A. (eds.) The book of L, pp. 13–27. Springer-Verlag (1986)

[5] Blumer, A., Blumer, J., Haussler, D., Ehrenfeucht, A., Chen, M., Seiferas, J.: The smallest automation recognizing the subwords of a text. Theoretical Computer Science 40, 31–55 (1985)

[6] Burcroff, A., Winsor, E.: Generalized Lyndon factorizations of infinite words. Theoretical Computer Science 809, 30–38 (2020)

[7] Charlier, E., Kärki, T., Rigo, M.: Multidimensional generalized automatic sequences and shape-symmetric morphic words. Discrete Mathematics 310(6), 1238–1252 (2010)

[8] Chuan, W.F.: Fibonacci words. Fibonacci Quarterly 30(1), 68–76 (1992)

[9] Chuan, W.F.: Symmetric Fibonacci words. Fibonacci Quarterly 31(3), 251–255 (1993)

[10] Chuan, W.F.: Generating Fibonacci words. Fibonacci Quarterly 33(2), 104–112 (1995)

[11] Chuan, W.F., Ho, H.L.: Locating factors of the infinite Fibonacci word. Theoretical Computer Science 349(3), 429–442 (2005)

[12] Crochemore, M., Vérin, R.: Direct construction of compact directed acyclic word graphs. In: Proceedings of the 8th Annual Symposium on Combinatorial Pattern Matching. pp. 116–129. CPM ’97, Springer-Verlag (1997)

[13] Gamard, G., Richomme, G., Shallit, J., Smith, T.: Periodicity in rectangular arrays. Information Processing Letters 118, 58–63 (2017)

[14] Giammarresi, D., Restivo, A.: Two-dimensional languages. In: G. Rozenberg, A. Salomaa (eds), Handbook of Formal Languages, Vol. 3. Springer-Verlag (1997)

[15] Giancarlo, R., Guaiana, D.: On-line construction of two-dimensional suffix trees. Journal of Complexity 15(1), 72–127 (1999)

[16] Jahannia, M., Mohammad-noori, M., Rampersad, N., Stipulanti, M.: Palindromic Ziv–Lempel and Crochemore factorizations of m-bonacci infinite words. Theoretical Computer Science 790, 16–40 (2019)

[17] Jishe, F.: Some new remarks about the dying rabbit problem. Fibonacci Quarterly 49(2), 171–176 (2011)

[18] Kitaev, S., Lozin, V.: Words and Graphs. Springer Cham (2015)

[19] Kulkarni, M.S., Mahalingam, K., Sivasankar, M.: Combinatorial properties of Fibonacci arrays. In: Gopal, T., Watada, J. (eds.) Theory and Applications of Models of Computation. pp. 448–466. Springer International Publishing (2019)

[20] Lothaire, M.: Combinatorics on words. Cambridge University Press (1997)

[21] Lothaire, M.: Algebraic combinatorics on words. Cambridge University Press (2002)
[22] Mahalingam, K., Sivasankar, M., Krithivasan, K.: Palindromic properties of two dimensional Fibonacci words. The Romanian Journal of Information Science and Technology 21(3), 256 – 266 (2018)

[23] Maxime, C., Christophe, H., Thierry, L.: Algorithms on Strings. Cambridge University Press (2007)

[24] Mignosi, F., Pirillo, G.: Repetitions in the Fibonacci infinite word. RAIRO Theoretical Informatics and Applications 26(3), 199 – 204 (1992)

[25] Rosenfeld, A.: Picture languages: Formal models of picture recognition. Academic Press (1979)

[26] Rytter, W.: The structure of subword graphs and suffix trees of Fibonacci words. Theoretical Computer Science 363(2), 211–223 (2006)

[27] Siromoney, G., Siromoney, R., Krithivasan, K.: Picture languages with array rewriting rules. Information and Control 22, 447–470 (1973)

[28] Sivasankar, M., Rama, R.: Two-dimensional Fibonacci words: Tandem repeats and factor complexity. Preprint (2022)

[29] Smyth, B., Smyth, W.: Computing Patterns in Strings. Pearson Education (2003)

[30] Swain, Gordon, A.: Exploring sequences through variations on fibonacci. Ohio Journal of School Mathematics 77(1), 29–33 (2017)

[31] Walczak, B.: A simple representation of subwords of the Fibonacci word. Information Processing Letters 110(21), 956–960 (2010)

[32] West, D.B.: Introduction to Graph Theory. Pearson Education, 2nd edn. (2001)