Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 363 (2006) 211–223
Theoretical Computer Science
www.elsevier.com/locate/tcs

The structure of subword graphs and suffix trees of Fibonacci words☆

Wojciech Rytter

Institute of Informatics, Warsaw University, Warsaw, Poland

Abstract

We use automata-theoretic approach to analyze properties of Fibonacci words. The directed acyclic subword graph (dawg) is a useful deterministic automaton accepting all suffixes of the word. We show that dawg's of Fibonacci words have particularly simple structure. Our main result is a unifying framework for a large collection of relatively simple properties of Fibonacci words. The simple structure of dawgs of Fibonacci words gives in many cases simplified alternative proofs and new interpretation of several well-known properties of Fibonacci words. In particular, the structure of lengths of paths corresponds to a number-theoretic characterization of occurrences of any subword. Using the structural properties of dawg's it can be easily shown that for a string w we can check if w is a subword of a Fibonacci word in time $O(|w|)$ and $O(1)$ space. Compact dawg's of Fibonacci words show a very regular structure of their suffix trees and show how the suffix tree for the Fibonacci word grows (extending the leaves in a very simple way) into the suffix tree for the next Fibonacci word.

Keywords: Fibonacci words; Subword graphs; Suffix trees

1. Introduction

Fibonacci words form a famous family of words, due to many interesting properties related to text algorithms and combinatorics on words, see [7,18]. In particular, Fibonacci words have $\Theta(n \log n)$ positioned squares and they have linear number of *runs*: maximal periodic subsegments of the string, see [11,17]. The string x is said to be *periodic* iff $period(x) \leqslant |x|/2$. The structure of runs in general strings is rather mysterious, and the structure of runs in Fibonacci words helps to understand this structure. In this sense Fibonacci words are very representative. A very good source for properties of these words is for example the book [18]. We rediscover several known/unknown properties of Fibonacci words in a novel way: analyzing the automaton for the set of subwords. Let F_n be the nth Fibonacci word, where

$$F_0 = a, \quad F_1 = ab, \quad F_{n+1} = F_n \cdot F_{n-1}.$$

Denote by Φ_n the nth Fibonacci number, where $|F_n| = \Phi_n$. Define also the infinite Fibonacci word $\mathcal{F}_\infty = \mathcal{F}_\infty(1, 2, 3, 4, \ldots)$, such that each F_n is a prefix of $\mathcal{F}_\infty$. Hence

$$\mathcal{F}_\infty = abaababaabaababaababaabaababaababaabaababaababaabaababaababaab\ldots$$

☆ Supported by the Grant KBN 4T11C04425.
E-mail address: rytter@duch.mimuw.edu.pl.

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.07.025

The structure of lengths of paths in the dawg's of Fibonacci words is closely related to the Fibonacci number system. This system consists in representing a number as a sum of Fibonacci numbers, in such a way that no two consecutive Fibonacci numbers are used. The sum of zero number of integers equals zero. The corresponding representation of the number is called $\mathcal{Z}$-representation.

Let $\mathcal{Z}_n$ be the set of nonnegative integers which do not use Fibonacci numbers $\Phi_0, \Phi_1, \ldots, \Phi_{n-1}$ in their Fibonacci representation.

If X is a set of integers then define:

$$X \oplus j = \{x + j : x \in X\}.$$

Denote by g_i (the ith truncated Fibonacci word) the word F_i with the last two letters removed. Denote by w^R the reverse of a word w. Let $R_i = F_i^R$, and let for $|w| \geqslant 2$ $twolast(w)$ denote the word of length 2 composed of the last two letters of w. It can be shown, see [2,12], that for $n \geqslant 2$:

$$g_n = R_0 R_1 R_2 \ldots R_{n-2}, \quad g_n^R = g_n, \quad twolast(F_n)^R = twolast(F_{n+1}). \tag{1}$$

It follows directly from Eq. (1) and definition of g_n that:

$$R_{n+1} = twolast(F_n) g_{n+1} = twolast(F_{n+1})^R g_{n+1}. \tag{2}$$

By an *occurrence* of u in $\mathcal{F}_\infty$ we mean a position i such that $\mathcal{F}_\infty[i+1 \ldots i+|u|] = u$. Denote by *first-occ*$(u)$ the first occurrence of u in $\mathcal{F}_\infty$, and by $occ(u)$ the set of all occurrences. Using the dawg's we show:
for each nonempty subword u of $\mathcal{F}_\infty$ we have:

$$occ(u) = occ(g_i) \oplus \textit{first-occ}(u),$$

where g_i is the shortest truncated Fibonacci word containing u.

Similarly, it is shown that:

$$occ(g_{n+1}) = occ(F_n) = \mathcal{Z}_n \quad \text{for } n > 1,$$

$$\text{and } occ(F_1) = occ(F_2),\ occ(F_0) = \mathcal{Z}_1.$$

It follows also easily from the structure of the dawg's that every run in $\mathcal{F}_\infty$ (except aa, $(ab)^2$) is of the form $(F_i)^k g_{i-1}$, where $k \in \{2, 3\}$. In case of runs a similar analysis of the structure of runs of squares has been already done by Iliopoulos et al. [9]. However their proofs were syntactic, we present different graph-theoretic proofs, based on a natural number-theoretic interpretation of the sets of lengths of paths of the dawg of $\mathcal{F}_\infty$.

2. The structure of subword graphs

We construct the infinite labelled graph $\mathcal{G}_\infty$. The nodes of $\mathcal{G}_\infty$ are all nonnegative integers and for $i > 0$ we construct the edges:

$$(i-1) \overset{\mathcal{F}_\infty(i)}{\rightarrow} i, \quad \Phi_i - 2 \overset{s}{\rightarrow} \Phi_{i+1} - 1, \tag{3}$$

where

$$s = \begin{cases} a & \text{if even } i, \\ b & \text{otherwise.} \end{cases}$$

The graph $\mathcal{G}_\infty$ is, in a certain sense, a subword graph of the infinite Fibonacci word F_∞. The initial segments of this graph are dawg's of finite Fibonacci words.

Let $\mathcal{G}_n$ be the subgraph of $\mathcal{G}_\infty$ induced by the nodes $[0 \ldots \Phi_n]$, see Fig. 1.

The edges of the form $(i, i+1)$ of $\mathcal{G}_\infty$ are called *main edges*. Other edges are called *jump edges*.

Denote by $dawg(w)$ the acyclic directed subword graph of a word w. It is a minimal deterministic finite automaton accepting all suffixes of w, in which we ignore accepting states and transitions leading to the sink (the rejecting state in which the automaton loops), see for example [7,5,18] for more detailed definition. Each path leading from the source

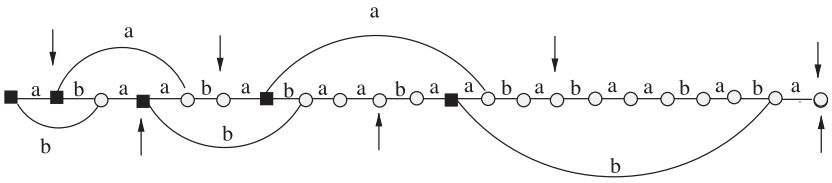

Fig. 1. The subgraph of $\mathcal{G}_\infty$ induced by the nodes $[0 \dots \Phi_6]$: the subword graph $dawg(F_6)$, the *fork nodes* (of out-degree 2) are drawn as squares. The arrows show the ends of prefixes which are Fibonacci words.

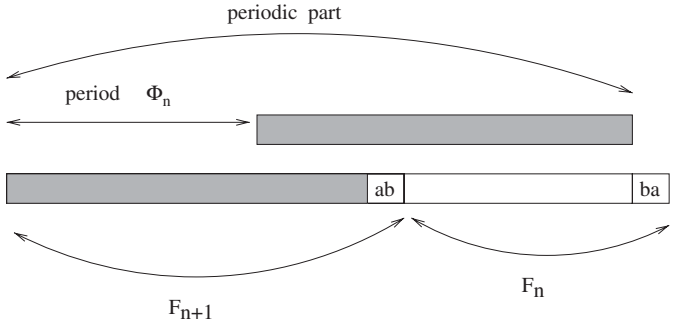

Fig. 2. The word g_{n+2} (F_{n+2} with the last two letters cut off) has a period Φ_n. The last two symbols of F_{n+2} are the swap of the last two symbols of F_{n+1}. The figure shows the case of even n.

node of this graph corresponds to a subword of w, and conversely, each subword corresponds to a unique path in $dawg(w)$. This graph has always at most $2n$ nodes, in our case the number of nodes is even smaller.

The main branch in $dawg(w)$ corresponds to the whole word w. We assume that the nodes on the *main branch* (corresponding to main edges) of such a graph are consecutive integers starting with 0.

Fact 1. *For each $n > 1$ $dawg(F_n) = \mathcal{G}_n$ and $paths(\mathcal{G}_n) = suffixes(F_n)$.*

Proof. The thesis follows by induction on n. Consider how G_{n+2} grows from G_{n+1} using the linear-time on-line algorithm, see [7]. In this construction the main branch $[0 \dots \Phi_{n+1}]$ of the dawg G_{n+1} is extended to the set of nodes $[0 \dots \Phi_{n+2}]$. It is enough to show that no extra node outside the main branch is created.

Claim 2. *g_{n+1} is a prefix and suffix of g_{n+2}.*

The claim follows from Eq. (1). We know that g_{n+1} is a prefix of g_{n+2}, since F_{n+1} is a prefix of g_{n+2}. Due to Eq. (1) the word g_{n+2}, hence the reverse of each of its prefixes is also its suffix. However, g_{n+1} is symmetric also. Hence g_{n+1} is also a suffix of g_{n+2}. This completes the proof of the claim. □

If $dawg(F_{n+1}) = \mathcal{G}_{n+1}$ then the next $|F_{n+1}| - 2$ symbols do not create new nodes or new edges outside the main branch since, due to the claim, g_{n+1} is a prefix and suffix of g_{n+2}. Consequently, g_{n+2} has the period $|F_n|$. This is illustrated in Fig. 2.

One extra edge is created from $|F_{n+1}| - 2$ to $|F_{n+2}| - 1$ because the next read symbol terminates the period of $|F_{n+1}|$.

Hence $dawg(F_{n+2})$ results from $dawg(F_{n+1})$ by adding the path labelled F_n and creating a jump edge from $|F_{n+1}|-2$ to $|F_{n+2}| - 1$ with the label different from the main edge from $|F_{n+1}| - 2$ to $|F_{n+1}| - 1$. In this way the labels of the edges are as in Eq. (2). This completes the proof.

Denote by finite-paths($\mathcal{G}_\infty$) the set of all finite words spelled by the paths of $\mathcal{G}_\infty$ originating at 0, and by finite-subwords($\mathcal{F}_\infty$) the set of all finite sub-words of $\mathcal{F}_\infty$.

The following fact follows directly from Fact 1.

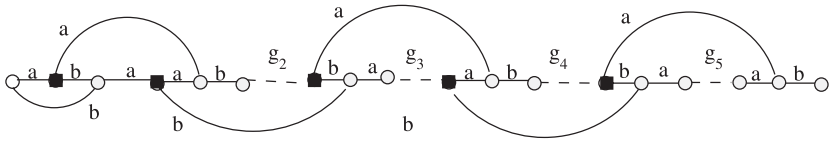

Fig. 3. The *structure* of the intermediate compressed version of $dawg(F_7)$ from Fig. 1. The dashed edges labelled g_i correspond to compressed chains.

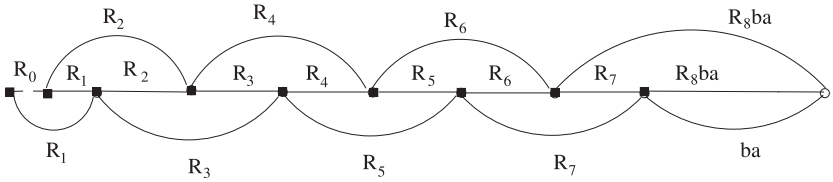

Fig. 4. The compacted subword graph $cdawg(F_{11})$ of the Fibonacci word of length 233. All labels (but one) are reverses R_i of Fibonacci words.

Fact 3. *finite-paths*$(\mathcal{G}_\infty) =$ *finite-subwords*$(\mathcal{F}_\infty)$.

The graph $dawg(F_n)$ can be compactly described in $O(n)$ space. In the first intermediate compaction each chain (a sequence of nodes of indegree and outdegree one) is represented by a single edge, see Fig. 3.

We can further compact the graph. The nodes of $\mathcal{G}_\infty$ and $\mathcal{G}_n$ of outdegree greater than one are called *fork nodes*. We say that a path starting from 0 is a *fork-path* iff it ends at a fork node in $\mathcal{G}_\infty$. In case of $\mathcal{G}_n$ the sink node is also consider as a fork node.

Only the fork nodes and compacted fork paths remain in the fully compacted version of $\mathcal{F}_\infty$. We remove all nodes except fork nodes. Then for each edge outgoing from a remaining node replace it by an edge going to the next remaining node, with label representing the word "spelled" by the compressed path, see Fig. 4. The resulting compacted subword graph is denoted by $cdawg(F_\infty)$ for the infinite graph and as $cdawg(F_n)$ for the compacted version of the finite graph $dawg(F_n)$.

We say that a path is an a-path if it is an infinite path in $\mathcal{G}_\infty$ which starts at 0, and chooses the edge labelled a whenever there is a choice. Similarly, define b-path.

Denote by a-path$(\mathcal{G}_\infty)$ the infinite word spelled by the a-path, similarly define b-path$(\mathcal{G}_\infty)$. The b-path$(\mathcal{G}_\infty)$ can be treated as the infinite lexicographically maximal *pseudo-suffix* of $\mathcal{F}_\infty$ (each prefix of b-path$(\mathcal{G}_\infty)$ is a prefix of maximal suffix of some finite Fibonacci word).

The suffixes of $\mathcal{F}_\infty$ are infinite words resulting by cutting off a finite prefix of the infinite word $\mathcal{F}_\infty$.

Denote by $R_i = F_i^R$ the reverse of the Fibonacci word F_i. Due to Eq. (1) we have:

$$\mathcal{F}_\infty = R_0R_1R_2R_3R_4R_5R_6\ldots. \tag{4}$$

Fact 4.

(a) *a-path*$(\mathcal{G}_\infty) = a \cdot \mathcal{F}_\infty$, *b-path*$(\mathcal{G}_\infty) = b \cdot \mathcal{F}_\infty$.

(b) *a-path*$(\mathcal{G}_\infty)$ *and b-path*$(\mathcal{G}_\infty)$ *are not suffixes of* $\mathcal{F}_\infty$.

(c) *The infinite string corresponding to a path* π *of* $\mathcal{G}_\infty$ *is a suffix of* $\mathcal{F}_\infty$ *iff almost all edges of* π (*all but a finite number*) *are main edges*.

Proof. Fig. 3 shows the compact version of $\mathcal{G}_\infty$, the labels on edges are words $ab \cdot g_n$ for even n and $ba \cdot g_n$ for odd n. These words are reverses of the Fibonacci words. Hence

$$\text{a-path}(\mathcal{G}_\infty) = a\ R_2R_4R_6R_8R_{10}\ldots \text{ and b-path}(\mathcal{G}_\infty) = R_1R_3R_5R_7\ldots.$$

The thesis follows from Eq. (1) and the fact that $R_{k+2} = R_kR_{k+1}$.

The points (b) and (c) follow directly from (a), since the infinite string $\mathcal{F}_\infty$ is not periodic. This completes the proof. □

The critical factorization point is a word w is a position k in w for which minimal local period at k equals the (global) minimal period of w. A positive integer p is a local period at k iff $w[k-i] = w[k+i-1]$ for all $1 \leqslant p$ for which $w[k-i]$, $w[k+i-1]$ are defined. We refer the reader to [7] for more detailed definition of the critical factorization point. The starting position of a lexicographically maximal suffix, maximized over all possible orders of the alphabet, is a critical factorization point, see [6,7]. This implies the following fact:

Fact 5. *$\Phi_n - \min\{|a\text{-path}(\mathcal{G}_n)|,\ |b\text{-path}(\mathcal{G}_n)|\}$ is the critical factorization point of the nth Fibonacci word.*

This gives alternative proof, see [8], of the following fact.

Fact 6. *$\Phi_{n-1} - 1$ is a critical factorization point of F_n.*

A word $w \in \{a,b\}^+$ is well balanced iff for each two subwords $u1, u2$ of w of the same length we have $|\#_a(u1) - \#_a(u2)| \leqslant 1$, where $\#_a(v)$ denotes the number of a's in v. It is known that each subword of a Fibonacci word is well balanced and one can test if a word is well balanced in quadratic time, see [16]. We give a linear test for subwords of Fibonacci words working in a constant space.

By O(1) space we mean constant number of nonnegative integers not greater than n.

Fact 7. *We can test if a word w is a subword of a Fibonacci word in time* $O(|w|)$ *and* $O(1)$ *space.*

Proof. It is easy to see that we can test if a specified subword of w is a Fibonacci word in linear time and O(1) space. Then we can traverse $\mathcal{G}$ without remembering it explicitly. In some places we have to test if a subword of w is a Fibonacci word. □

3. The structure of occurrences of subwords in Fibonacci words

Recall that $occ(u)$ is the set of occurrences of u in $\mathcal{F}_\infty$, we define also the set of final positions of occurrences of a word u:

$$\mathit{fin}(u) = occ(u) \oplus |u| \quad \text{and} \quad \mathit{first\text{-}fin}(u) = \min(\mathit{fin}(u)).$$

Observe that

$$\mathit{first\text{-}fin}(u) = \mathit{first\text{-}occ}(u) \oplus |u|.$$

Fact 8.

(a) *For each pair of nonempty subwords u, w of F_∞ we have*:

$$\mathit{first\text{-}fin}(u) = \mathit{first\text{-}fin}(w) \ \Leftrightarrow\ \mathit{fin}(u) = \mathit{fin}(w).$$

(b) *For each nonempty subword u of $\mathcal{F}_\infty$ we have*

$$occ(u) = occ(g_i) \oplus \mathit{first\text{-}occ}(u),$$

where g_i is the shortest truncated Fibonacci word containing u.

Proof. First, we show a fact which follows from the properties of the dawgs.

Claim 9. *For a subword w of F_∞ the node in $\mathcal{G}_\infty$ reached by the path labelled by w is first-fin(w).*

Proof of the claim. Each finite subword graph has the following property, see [7]:

If the path corresponding to a subword w terminates in a node k, then the set of words corresponding to paths terminating at k is $\{u : \mathit{fin}(u) = \mathit{fin}(w)\}$.

This property also holds for $\mathcal{G}_\infty$, as a limit of a series of finite subword graphs.

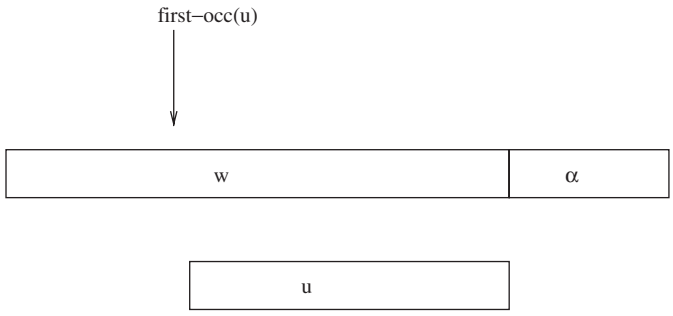

Fig. 5. w is a shortest prefix of F_∞ with the suffix equal to u, and $w\alpha$ is the shortest extension of w such that $w\alpha = g_i$ for some i. This g_i is the shortest g_i containing u as a subword. The string α can be empty. We have: $\mathit{first\text{-}occ}(u) = |w| - |u|$.

Assume the path labelled w leads to a node k. Then $\mathit{fin}(u) = \mathit{fin}(w)$ for any other word u which leads to k. In particular, the prefix u of $\mathcal{G}_\infty$ of length k leads to k. We have that $\mathit{first\text{-}fin}(u) = k$, hence also $\mathit{first\text{-}fin}(w) = k$, due to the equality $\mathit{fin}(u) = \mathit{fin}(w)$. Therefore, the path corresponding to w leads to $\mathit{first\text{-}fin}(w)$ in $\mathcal{G}_\infty$. This completes the proof of the claim. □

The point (a) follows directly from Claim 9. We show now the point (b).

Let g_i be the shortest truncated Fibonacci word containing u, and w be the shortest prefix of F_∞ containing u as a suffix, see Fig. 5. Then, according to Claim 9 $\mathit{fin}(u) = \mathit{fin}(w)$. On the other hand $occ(w) = occ(g_i)$, since there is no fork node between $|w|$ and $|g_i| - 1$. If X is a set of integers then define:

$$X \ominus j = \{x - j : x \in X\}.$$

We use the following equality:

$$(\ X = A \ominus j \text{ and } Y = A \ominus i\ ) \Rightarrow X = Y \oplus (i - j). \tag{5}$$

Observe that $occ(u) = \mathit{fin}(u) \ominus |u|$. We have now the equalities:

$$occ(u) = \mathit{fin}(u) \ominus |u| = \mathit{fin}(w) \ominus |u|, \quad occ(g_i) = occ(w) = \mathit{fin}(w) \ominus |w|.$$

Consequently $occ(u) = \mathit{fin}(w) \ominus |u|$, $occ(g_i) = \mathit{fin}(w) \ominus |w|$. Now, Eq. (5) and equality $|w| - |u| = \mathit{first\text{-}occ}(u)$ imply that

$$occ(u) = occ(g_i) \oplus (|w| - |u|) = occ(g_i) \oplus \mathit{first\text{-}occ}(u).$$

This completes the proof. □

Example. The shortest truncated Fibonacci word containing aa, as well as $F_3 = abaab$ is $g_4 = abaaba$. We have $occ(F_3) = occ(g_4) = \{0, 5, 8, 13, 18, 21, 26, 29, \ldots\}$, and $\mathit{first\text{-}occ}(aa) = 2$, hence $occ(aa) = occ(F_3) \oplus 2 = occ(g_4) \oplus 2 = \{2, 7, 10, 15, 20, 23, 28, 31, \ldots\}$.

For $k \geqslant 1$ we investigate also the structure of the set

$$FIN(k) = \{\mathit{first\text{-}fin}(u) : u \text{ is a subword of } \mathcal{F}_\infty \text{ of size } k\}.$$

Lemma 1. *The set $FIN(k)$ consists of a single interval or of two disjoint intervals. In particular, $FIN(\Phi_n - 1) = [\Phi_n - 1 \ldots 2 \cdot \Phi_n - 1]$;*

Proof. The structure of the set $FIN(k)$ easily follows from the way how paths of length $k-1$ are extended into paths of length k. Only fork nodes $i \in FIN(k-1)$ generate two elements of $FIN(k)$, each other node i in $FIN(k-1)$ generates a single element $i + 1$ in $FIN(k)$, see Fig. 6. We have:

$$FIN(k+1) = (FIN(k) \oplus 1) \cup \{\Phi_{i+1} - 1\} \quad \text{where } \Phi_i - 2 \in FIN(k).$$

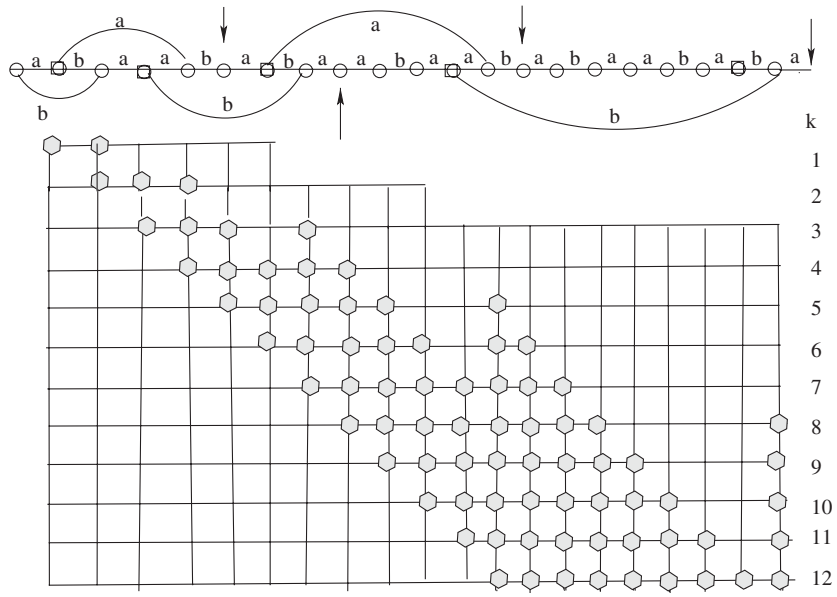

Fig. 6. The structure of the sets $FIN(k)$ of endpoints of first occurrences of all $k+1$ different strings of length k. The set $FIN(k)$ is illustrated as a set of nodes in the kth line of the table. For example $FIN(6) = \{6, 7, 8, 9, 10, 12, 13\}$. We have: $|FIN(k)| = k+1$. The end-positions of Fibonacci prefixes are indicated by vertical arrows.

Hence the set $FIN(k+1)$ results from $FIN(k)$ by shifting each position by 1 to the right and adding one extra position $\Phi_{i+1} - 1$. □

We say that a subword w of $\mathcal{F}_\infty$ is a right special subword, iff wa and wb are subwords of $\mathcal{F}_\infty$. Such subwords are responsible for the increase of the number of subwords with respect to their length. These are the words corresponding to paths to fork nodes, they are considered for example in [3]. It follows from the structure of $\mathcal{G}$ that right special subwords are exactly suffixes of g_i's. On the other hand, each suffix of g_i is a reverse of a prefix of $\mathcal{F}_\infty$. Let w^R denote the reverse of w.

In this way we give a new proof of another property of $\mathcal{F}_\infty$:

a word w is a right special factor of $\mathcal{F}_\infty$ iff w^R is a prefix of $\mathcal{F}_\infty$.

This property is already known, see for example [2], where the proof is different.

Recall that the Fibonacci number system consists in representing a number as a sum of Fibonacci numbers, in such a way that no two consecutive Fibonacci numbers are used. The sum of zero number of integers equals zero.

Lemma 2 (*Zeckendorff Lemma [19]*). *Every nonnegative integer is uniquely represented in the Fibonacci number system.*

Define the *dual Fibonacci system*: In this system each positive integer x is represented as a sum of different Fibonacci numbers, however, we require that if Φ_i is not taken then Φ_{i+1} is taken in the sum, whenever any Fibonacci number after Φ_i is taken. It follows directly from Zeckendorff's Lemma that:

Lemma 3. *Every integer $k > 0$ is uniquely represented in the dual Fibonacci number system.*

The next fact follows from the structure of the compacted infinite dawg $\mathcal{G}_\infty$.

Fact 10. *For each $k > 1$ there is exactly one fork-path of length k in $\mathcal{G}_\infty$.*

Proof. Representation of a non-negative integer x in the dual Fibonacci system is given by a sequence $\gamma = (\phi_{i_0}, \phi_{i_1},$

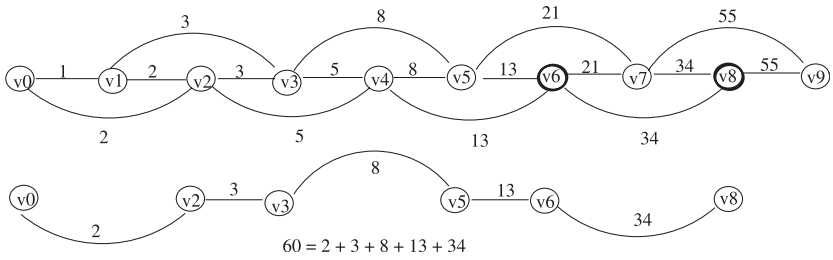

Fig. 7. We put the lengths of the edges as the lengths of compacted paths in $cdawg(\mathcal{G}_\infty)$. The representation $\gamma = (2, 3, 8, 13, 34) = (\phi_1, \phi_2, \phi_4, \phi_5, \phi_7)$ of $x = 60$ in the dual Fibonacci system corresponds to a fork-path $\pi = (v_0,\ v_2,\ v_3,\ v_5,\ v_6,\ v_8)$. We have $\Psi(\pi) = \gamma,\ length(\pi) = val(\gamma) = 60$.

$\phi_{i_2}, \ldots \phi_{i_n}$), which satisfies the conditions:
(a) ϕ_{i_k} is the i_kth Fibonacci number;
(b) $i_1 \in \{0, 1\}$;
(c) for each $k = 1, 2, \ldots n$ we have $1 \leqslant i_k - i_{k-1} \leqslant 2$.
Let $val(\gamma)$ denote the integer x represented by γ, it is the sum of the Fibonacci numbers which appear in γ. Denote by Γ the set of all such representations γ.

Consider now the set Π of finite fork paths π in $cdawg(F_\infty)$. Denote the fork nodes in $cdawg(F_\infty)$ by $v_0, v_1, v_2, \ldots$, where the number of v_i in $\mathcal{G}_\infty$ is the sum of the first i consecutive Fibonacci numbers. Each edge between fork nodes is of a type

$$v_i \to v_{i+1} \quad \text{or} \quad v_i \to v_{i+2}.$$

The length of the label of the edge $v_i \to v_j$ is Φ_{j-1}.

Each path $\pi \in \Pi$ is a sequence of nodes $\pi = (v_0,\ v_{i_1},\ v_{i_2}, \ldots v_{i_n})$, where $v_0 = 0$ and $1 \leqslant i_k - i_{k-1} \leqslant 2$ for each $k = 1, 2, \ldots n$. The length of this path is

$$length(\pi) = \sum_{k=1}^{n} \phi_{i_k - 1}.$$

We describe a function $\Psi : \Pi \to \Gamma$ as follows:

$$\pi = (v_0,\ v_{i_1},\ v_{i_2}, \ldots v_{i_n}) \ \Rightarrow\ \Psi(\pi) = (\phi_{i_1-1}, \phi_{i_2-1}, \phi_{i_3-1}, \ldots \phi_{i_n-1}).$$

Fig. 7 illustrates the correspondence between a representation γ of a $x = 60$ in the dual Fibonacci system and a fork-path π in $cdawg(\mathcal{G}_\infty)$. The function Ψ is the bijection between Π and Γ. Moreover, it satisfies:

$$\forall\ \pi \in \Pi\ \ length(\pi) = val(\Psi(\pi)).$$

Each integer $x \geqslant 0$ has a unique representation $\gamma \in \Gamma$ with $val(\gamma) = x$. Hence, due to bijection of Ψ there is a unique fork-path of a given length in $\mathcal{G}_\infty$. This completes the proof. □

Recall that $\mathcal{Z}_n$ is the set of nonnegative integers which do not use Fibonacci numbers $\Phi_0, \Phi_1, \ldots, \Phi_{n-1}$ in their Fibonacci representation. The representation of each number in $\mathcal{Z}_n$ is called a $\mathcal{Z}_n$-representation.

Example. We have the following $\mathcal{Z}_9$-representation

$$1000000 = F_9 + F_{11} + F_{23} + F_{25} + F_{29}.$$

The sorted set $\mathcal{Z}_k[0], \mathcal{Z}_k[1], \mathcal{Z}_k[2], \ldots$ is closely related to Fibonacci words, denote by $\mathcal{D}_k$ the displacement structure of $\mathcal{Z}_k$:

$$\mathcal{D}_k = (\mathcal{Z}_k[1] - \mathcal{Z}_k[0],\ \mathcal{Z}_k[2] - \mathcal{Z}_k[1],\ \mathcal{Z}_k[3] - \mathcal{Z}_k[2],\ \mathcal{Z}_k[4] - \mathcal{Z}_k[3],\ \ldots).$$

Example. For $k = 3$ we have $h_k(a) = \Phi_3 = 5,\ h_k(b) = \Phi_2 = 3$, hence in this case:

$$\mathcal{Z}_3 = \{0, 5, 8, 13, 18, 21, 26, 29, 34, 39, 42, \ldots\} = \{0, 0+5, 5+3, 8+5, 13+5, 18+3, 21+5, \ldots\}$$

$$\mathcal{D}_3 = (5, 3, 5, 5, 3, 5, 3, 5, 5, 3, 5, 5, 3, 5, 3, 5, \ldots,$$

$$\mathcal{F}_\infty = a, b, a, a, b, a, b, a, a, b, a, a, b, a, b, a, \ldots$$

The remarkable property of $\mathcal{Z}_k$ is that its displacement sequence is structurally isomorphic to the infinite Fibonacci word with large difference corresponding to a, and smaller one to the symbol b. It can be shown by induction that:

Lemma 4. *$\mathcal{D}_k = h_k(\mathcal{F}_\infty)$, where $h_k(a) = \Phi_k,\ h_k(b) = \Phi_{k-1}$.*

The structure of the graph $\mathcal{G}$ implies several number-theoretic properties of the set of occurrences of subwords in Fibonacci words. It follows from Fact 8 and the structure of the graph $\mathcal{G}$, see Fig. 7, that:

Fact 11.
(1) *$occ(g_{n+1}) = occ(F_n) = \mathcal{Z}_n$ for $n > 1$.*
(2) *$occ(F_1) = occ(F_2), occ(F_0) = \mathcal{Z}_1$.*
(3) *For each subword $u \notin \{F_0, F_1\}$ of $\mathcal{F}_\infty$ we have $occ(u) = \mathcal{Z}_i \oplus$first-occ$(u)$, where g_i is the smallest truncated Fibonacci word containing u as a subword.*

Proof. The subword u "*moves*" to the right by starting at *first-occ*(u) in $\mathcal{G}$ and making shortcuts. Each shortcut corresponds to taking a Fibonacci number, no two consecutive Fibonacci numbers are taken. □

4. The structure of runs in Fibonacci words

We say that a run w is a p-run iff $period(w) = p$. The run is short if $|w| < 3 \cdot period(w)$, otherwise the run is called long. The structure of runs has been already investigated in [13,9]. Some of our results are similar to those from [13,9] but our approach is different, it follows from the role played by words g_n with respect to occurrences of other words. This role of words g_n is expressed in Fact 8 and is a consequence of the structure of subword graphs $dawg(F_n)$.

Due to Fact 8 every repetitive occurrence of a subword in $\mathcal{F}_\infty$ implies an occurrence of an overlap of some word g_i, see Fig. 8. Hence the runs correspond to adjacent occurrences (or overlaps) of words g_i, see Fig. 9.

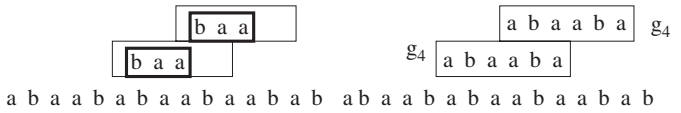

Fig. 8. An occurrence of a square $baa \cdot baa$ occurs always inside an overlap of $g_4 = abaaba$, since $occ(baa) = occ(g_4) \oplus 1$.

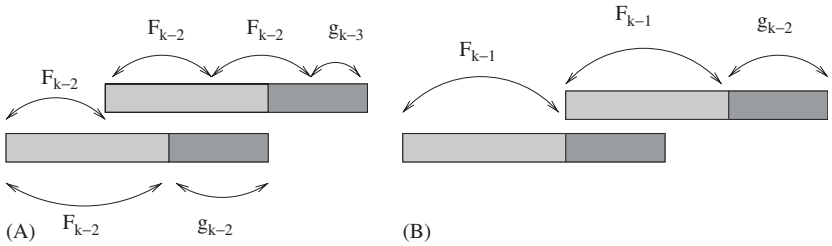

Fig. 9. There are two possibilities of the relative structure of two consecutive occurrences of g_k, for $k \geqslant 4$, they can have overlap of size Φ_{k-2} or Φ_{k-1}, which give rise to two types of runs: $\alpha_k = F_{k-2}^3 g_{k-3}$ and $\beta_k = F_{k-1}^2 g_{k-2}$.

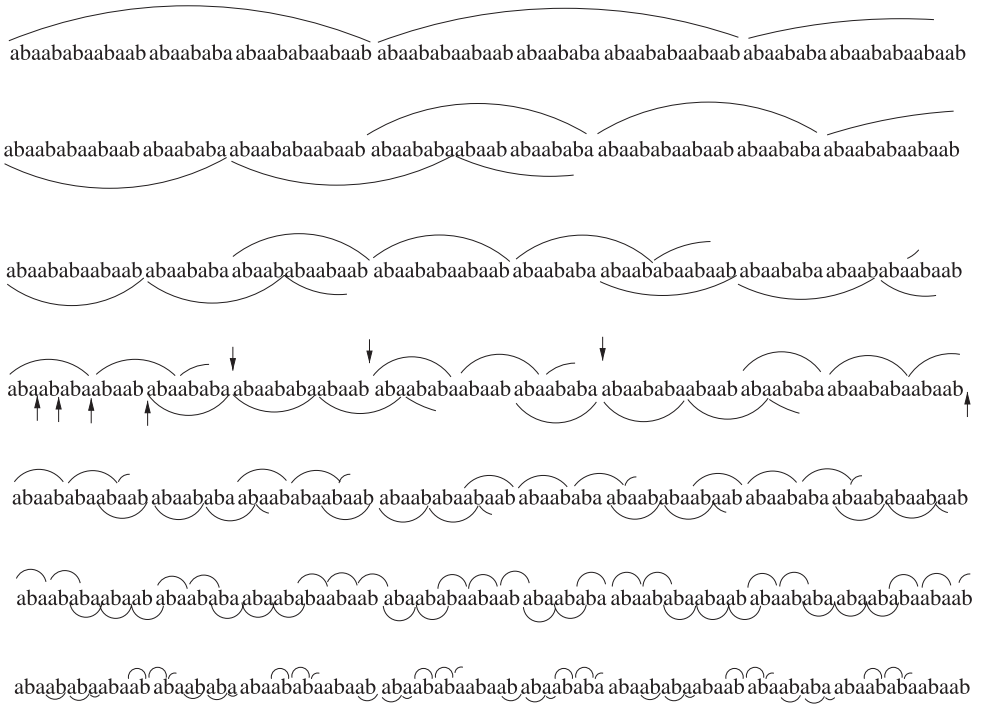

Fig. 10. The structure of *runs* in the Fibonacci word F_9. The arrows show endpoints of prefixes of F_9 which are Fibonacci words.

The displacement sequence for the occurrences of g_k consists in Φ_{k-1} or Φ_{k-2}. Hence the overlap of two g_k's can generate two types of runs, the short Φ_{k-1}-runs, and long Φ_{k-2}-runs, see Fig. 9. Consequently we have the following fact.

Lemma 5 (*Mignosi and Pirillo [13]*). *Every run of $\mathcal{F}_\infty$ with period larger than two is of one of two types a short β-run $\beta_k = F_{k-1}^2 g_{k-2}$, or a long α-run $\alpha_k = F_{k-2}^3 g_{k-3}$, for an integer k.*

We define the *repetition order*, denoted by $rep(x)$, of the string (finite or infinite) x as

$$rep(x) = \sup\{|w|/period(w) : w \in \text{finite-subwords(x)}\}.$$

The maximal repetitions correspond to long α-runs in $\mathcal{F}_\infty$. We have:

$$\frac{|\alpha_k|}{period(\alpha_k)} = \frac{3\cdot\Phi_{k-2} + \Phi_{k-3} - 2}{\Phi_{k-2}} = 2 + \frac{\Phi_{k-1} - 2}{\Phi_{k-2}}.$$

We have

$$\lim_{k\to\infty} \frac{\Phi_{k-1} - 2}{\Phi_{k-2}} = \phi.$$

This implies the following fact (already shown in [13]).

Fact 12 (*Mignosi and Pirillo [13]*). *$rep(\mathcal{F}_\infty) = 2 + \phi$, where $\phi = (1+\sqrt{5})/2$ is the golden ratio.*

All runs correspond to occurrences of g_i's. However, $\mathcal{Z}_i$ is the set of all occurrences of g_i. The Displacement Lemma (Lemma 4) plays the crucial role in understanding the structure of runs in F_n. We know that the displacement sequence is isomorphic to Fibonacci sequence, hence we can compute number of different types of runs by computing numbers of a's and b's in prefix segments of $\mathcal{F}_\infty$. Using Lemma 4 we can describe the structure of runs in F_n.

Fact 13. *The Fibonacci word F_n has*: F_{n-3} Φ_0*-runs*; $F_{n-4}-1$ Φ_1*-runs, and* F_{n-k-2} Φ_k*-runs for* $2 \leqslant k \leqslant n-2$.

This gives alternative (compared with [11]) proof for the number of all runs.

Fact 14 (*Kolpakov and Kucherov [11]*). *F_n has $2 \cdot F_{n-2} - 3$ runs.*

Example. The structure of runs in F_9 is shown in Fig. 10. The string F_9 has 65 runs. The 21 1-runs of *aa* are not shown in the figure. There are $\Phi_5 - 1$ 2-runs, Φ_5 3-runs, Φ_4 5-runs, Φ_3 8-runs, Φ_2 13-runs, Φ_1 21-runs and Φ_0 34-run.

It follows from the structure of runs α_k, β_k that they do not contain a power x^4 of a nonempty word x. This implies the following well known fact, see [2,10], in a novel way:

Fact 15. *There are no subwords in $\mathcal{F}_\infty$ of type x^4, where x is nonempty.*

Using the displacement sequence $\mathcal{D}_k$, due to its recursive Fibonacci-like structure, we can easily show the following:

Fact 16. *For $n > k$, the number of occurrences of F_k in F_n is $F_{n-k} - odd(n-k)$, where $odd(x) = 1$ if x is an odd integer, and $odd(x) = 0$ otherwise.*

5. The structure of suffix trees of Fibonacci words

The suffix tree T_n of F_n is the tree of all paths of $cdawg(F_n)$. The structure of this tree and the way how T_n evolves into T_{n+1} follows from the structure and evolution of compacted dawg's, see Figs. 11–13. A terminal edge is an edge leading to a leaf. The suffix trees of Fibonacci words grow at their leaves, by changing the terminal edges in a very simple regular way.

Fact 17. *For $n > 2$ the suffix tree T_n of F_n has Φ_{n-1} leaves and Φ_{n-2} internal nodes. Let x be the last two symbols of F_{n+1}. T_n evolves into T_{n+1} in the following way*:

Long edges: *Each terminal edge (u, v) with label $x F_{n-2}$ is transformed into the subtree isomorphic to S_n, two end symbols are cut off from the label of (u, v), and two edges originated at v are created, with labels $x^R F_{n-1}$ and x.*

Short edges: *Each terminal edge with label x changes its labels to $x \cdot F_{n-1}$.*

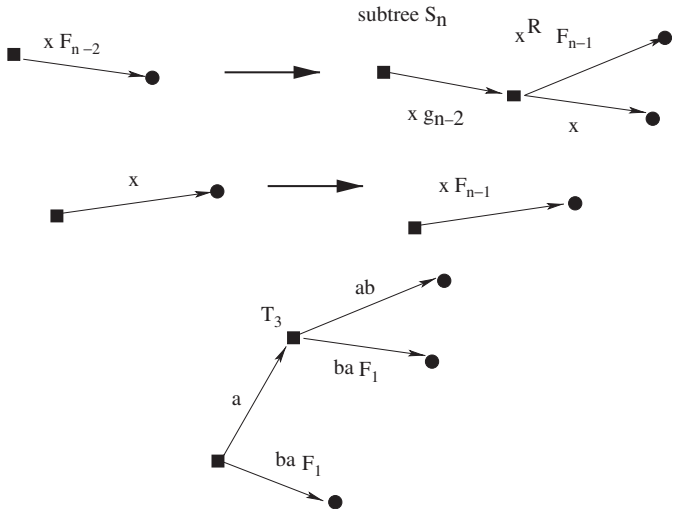

Fig. 11. The suffix tree T_3 and the general rules to generate T_{n+1} from T_n. The word $x \in \{ab,\ ba\}$.

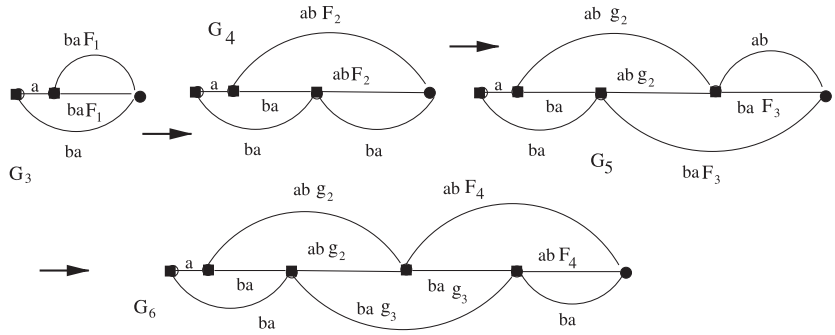

Fig. 12. The evolution of the compacted graph $cdawg(F_6)$.

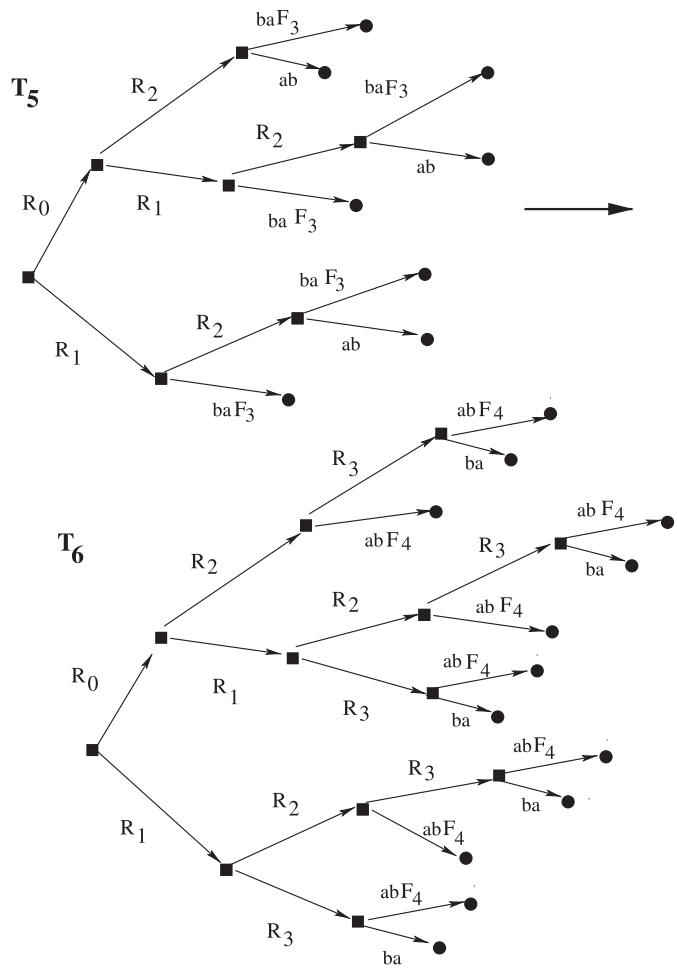

Fig. 13. The evolution of the suffix tree $T_6 = T(F_6)$. Compare with Fig. 12, observe that g_1 is the empty string and that the labels $ab\ g_i$ (for even i) and $ba\ g_j$ (for odd j) are reverses R_k of Fibonacci words. We can obtain *in the limit* an *infinite suffix tree* of $\mathcal{F}_\infty$.

We know precisely how the suffix trees grow. The sum of lengths of edges of the suffix tree is the number of different subwords. We have Φ_{n-3} short edges, each of them grows by Φ_{n-1}, and Φ_{n-2} long edges, each grows by $\Phi_{n-1}+2$. This gives easily a simple recurrence and a new suffix-tree oriented proof for the known formula of the number $Sub(n)$ of different subwords of F_n.

Corollary 1. *For $n > 2$ we have*:
$Sub(n+1) = Sub(n) + \Phi_{n-3} \cdot \Phi_{n-1} + \Phi_{n-2} \cdot (\Phi_{n-1} + 2)$,
$Sub(n) = \Phi_{n-1}\Phi_{n-2} + 2 \cdot \Phi_{n-1} - 1$.

We say that two labelled trees are structurally isomorphic iff they are isomorphic as unordered trees in graph-theoretic sense, disregarding the labels. The following fact also follows from the structure of cdawg's.

Fact 18 (*Fibonacci-like structure of suffix trees of Fibonacci words*). *For $n > 4$ the two subtrees rooted at the sons of the root of the suffix tree $T(F_n)$ are structurally isomorphic to the suffix trees $T(F_{n-1})$ and $T(F_{n-2})$, respectively.*

Acknowledgment

I thank Łukasz Mikulski [15] for helpful comments related to Lemma 1. He has written an independent and more formal proof of this lemma. We refer also to [2,4] for alternative proofs.

The subword graphs for general Sturmian words have been already considered in [14], where the emphasis was on Sturmian graphs and their counting properties. There is a huge literature on Sturmian words (each Fibonacci word is a Sturmian word).

Some applications of the structure of subword graphs of Fibonacci words to the lexicographic properties of the Fibonacci number system were presented in [1].

References

[1] P. Baturo, K. Czarkowski, M. Plichowski, M. Piatkowski, W. Rytter, Suffix arrays of Fibonacci words and lexicographic properties of the Fibonacci number system, presented at Internat. School and Conference on Combinatorics, Automata and Number Theory (CANT 2006), 2006, University of Liège.
[2] J. Berstel, Fibonacci words—a survey, in: G. Rosenberg, A. Salomaa (Eds.), The Book of L, Springer, Berlin, 1986, pp. 11–26.
[3] J. Berstel, J. Karhumäki, Combinatorics on words—a tutorial, Bull. EATCS 79 (2003) 178–228.
[4] J. Berstel, P. Seebold, Sturmian words, in: M. Lothaire (Ed.), Algebraic Combinatorics on Words, Encyclopedia of Mathematics and its Applications, Vol. 90, Cambridge University Press, Cambridge, 2002, pp. 45–110, (Chapter 2).
[5] Ch. Charras, T. Lecroq, Handbook of Exact String Matching, King's College Publications, 2004.
[6] M. Crochemore, D. Perrin, Two-way string matching, J. ACM 38 (3) (1991) 651–675.
[7] M. Crochemore, W. Rytter, Jewels of Stringology, World Scientific, Singapore, 2003.
[8] T. Harju, D. Nowotka, On the density of critical factorizations, ITA 36 (3) (2002) 315–327.
[9] C. Iliopoulos, D. Moore, W.F. Smyth, A characterization of the squares in a Fibonacci string, Theoret. Comput. Sci. 172 (1–2) (1997) 281–291.
[10] J. Karhumäki, On cube-free ω-words generated by binary morphisms, Discrete Appl. Math. 5 (1983) 279–297.
[11] R. Kolpakov, G. Kucherov, Finding maximal repetitions in a word in linear time, FOCS 1999, pp. 596–604.
[12] A. De Luca, A combinatorial property of the Fibonacci words, Inform. Process. Lett. 12 (1981) 193–195.
[13] F. Mignosi, G. Pirillo, Repetitions in the Fibonacci infinite words, RAIRO Theoret. Inform. Appl. 26 (1992) 194–204.
[14] F. Mignosi, J. Shallit, I. Venturini, Sturmian graphs and a conjecture of moser, DLT'04, Lecture Notes in Computer Science, Vol. 3340, Springer, Berlin, 2004, pp. 175–187.
[15] Ł. Mikulski, Personal communication, 2005.
[16] G. Richomme, Tests-words for Sturmian morphisms, Bull. Belgian Math. Soc. Simon Stevin 6 (1999) 481–489.
[17] W. Rytter, The maximal number of runs in a string: improved analysis of the linear upper bound, STACS 2006, pp. 195–205.
[18] B. Smyth, Computing Patterns in Strings, Addison Wesley, Reading, MA, 2003.
[19] E. Zeckendorff, Représentation des nombres naturels par une somme de nombres de Fibonacci ou des nombres de Lucas, Bulletin de la Societé Royale des Sciences de Liège 41 (1972) 179–182.