Decision algorithms for Fibonacci-automatic words, with applications to pattern avoidance

Chen Fei Du [1], Hamoon Mousavi [1], Luke Schaeffer [2], and Jeffrey Shallit [1]

Abstract

We implement a decision procedure for answering questions about a class of infinite words that we call "Fibonacci-automatic". This class includes the infinite Fibonacci word $\mathbf{f} = 01001010\cdots$ defined as the fixed point of the morphism mapping $0 \to 01$ and $1 \to 0$. We give three applications of this decision procedure, proving two new results and correcting one old result. The first is the existence of an aperiodic mirror-invariant infinite binary word avoiding the pattern xxx^R. The second is a description of the lexicographically least sequence over $\mathbb{N} \setminus \{0\}$ of a specific form avoiding additive squares. The third is a correction of an old result regarding the number of squares occurring in the finite Fibonacci words.

1 Introduction

Presburger arithmetic, the first-order logical theory $\text{Th}(\mathbb{N}, 0, 1, +)$, has long been known to be decidable [35, 36]. Augmented with the function $V_k(n) := \max_{k^i \mid n} k^i$ for some fixed integer $k \geq 2$, the resulting theory is still decidable [14]. This theory is powerful enough to define finite automata; for a survey, see [13]. In essence, we have the following theorem (see, e.g., [40]).

Theorem 1. *For each $k \geq 2$, there exists an algorithm that, given a first-order proposition using constants and relations definable in* $\text{Th}(\mathbb{N}, 0, 1, +)$ *and indexing into one or more k-automatic sequences, decides if the proposition holds.*

Many results in the literature about properties of automatic sequences, for which some had only long and involved proofs, can be proved purely mechanically using such a decision procedure. It suffices to express the property as an appropriate logical predicate, convert the predicate into an automaton accepting representations of integers for which the predicate holds, and then examine the automaton (see, e.g., the recent papers [2, 25, 27, 26, 28]). Furthermore, in many cases, we can explicitly enumerate various aspects of such sequences, such as subword complexity [16].

Beyond fixed radixes, one can define automata taking representations in more exotic numeration systems as input. For example, in the Pisot numeration systems, addition is computable [23, 24]; hence, a theorem analogous to Theorem 1 holds for these systems (see, e.g., [12]). We contend that the power of this approach has not been widely appreciated, and that many results, previously proved using long and involved ad hoc techniques, can be proved with much less effort by phrasing them as logical predicates and employing a decision procedure. Furthermore, many enumeration questions can be solved with a similar approach.

We have implemented a decision procedure for one such system: Fibonacci representation. In this extended abstract, we briefly describe the decision procedure, and then, as applications of this decision procedure, prove two new results and reprove (with a correction) an old result.

[1]School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada; `cfdu@uwaterloo.ca`, `sh2mousa@uwaterloo.ca`, `shallit@uwaterloo.ca`.

[2]Computer Science and Artificial Intelligence Laboratory, The Stata Center, MIT Building 32, 32 Vassar Street, Cambridge, MA 02139, USA; `lrschaeffer@gmail.com`.

2 Decision procedure for Fibonacci-automatic words

We define the Fibonacci numbers, as usual, by $F_0 := 0$, $F_1 := 1$, and $F_n := F_{n-1} + F_{n-2}$ for $n \geq 2$. It is well-known, due to Ostrowski [33], Lekkerkerker [31], and Zeckendorf [42], that every natural number can be represented, in an essentially unique way, as a sum of Fibonacci numbers $(F_i)_{i \geq 2}$, subject to the constraint that no two consecutive Fibonacci numbers are used. (Also see [15, 19].) For $w \in \Sigma_2^*$, we define the natural number it represents in base Fibonacci by $\langle w \rangle_F := \sum_{i=0}^{|w|-1} w[i] F_{n+2-i}$ with most significant bits first. We define the *canonical* Fibonacci representation of each $n \in \mathbb{N}$, denoted $(n)_F$, to be the one having no leading 0s or consecutive 1s. For example, $(0)_F = \epsilon$ (the empty string) and $(43)_F = \texttt{10010001}$ since $43 = F_9 + F_6 + F_2$.

We say that an infinite binary word $\mathbf{a}$ is *Fibonacci-automatic* if there exists a deterministic finite automaton with output (DFAO) of the form $(Q, \Sigma_2, \Sigma_2, q_0, \delta, \kappa)$ such that $\mathbf{a}[n] = \kappa(\delta(q_0, (n)_F))$ for all $n \in \mathbb{N}$. (Also see [41].) This is an analogue of the more familiar notion of k-automatic sequence [17, 3]. An example of a Fibonacci-automatic sequence is the infinite Fibonacci word $\mathbf{f} = \texttt{01001010}\cdots$, which is generated by the (incomplete) DFAO depicted in Figure 1.

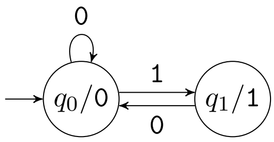

Figure 1: Canonical Fibonacci representation DFAO generating the Fibonacci word

We can extend Fibonacci representation to finite tuples of natural numbers by, for each $n \in \mathbb{N}$, viewing $z \in (\Sigma_2^n)^*$ as a Fibonacci representation of $(\pi_1(z)_F, \pi_2(z)_F, \ldots, \pi_n(z)_F) \in \mathbb{N}^n$, where $\pi_i(z)$ is the projection over the i-th coordinate. Since the canonical Fibonacci representations of different numbers may have different lengths, padding some coordinates with leading 0s will often be necessary. Hence, we define the canonical Fibonacci representation of $(k_1, k_2, \ldots, k_n) \in \mathbb{N}^n$, denoted $(k_1, k_2, \ldots, k_n)_F$, to be the one having no leading [0,0,...,0]s and where the projection into each coordinate has no consecutive 1s. For example, $(9, 16)_F = $ [0,1][1,0][0,0][0,1][0,0][1,0] because $(9)_F = \texttt{10001}$ and $(16)_F = \texttt{100100}$.

The crux of our decision procedure for Fibonacci-automatic words is that, just as with fixed radix systems, addition in Fibonacci representation can also be performed by a deterministic finite automaton (DFA). More precisely, there exists a DFA $M_{F\text{-add}}$ that accepts $z \in (\Sigma_2^3)^*$ iff $\pi_1(z)_F + \pi_2(z)_F = \pi_3(z)_F$. For example, $M_{F\text{-add}}$ accepts [0,0,1][1,0,0][0,1,0][1,0,1] because $\langle \texttt{0101} \rangle_F + \langle \texttt{0010} \rangle_F = 4 + 2 = 6 = \langle \texttt{1001} \rangle_F$. This result is apparently originally due to Berstel [4]. (Also see [5, 21, 22, 1].)

Since $M_{F\text{-add}}$ does not appear to have been given explicitly in the literature and it is essential to our implementation of the decision procedure, we present the (incomplete) minimal DFA here: $M_{F\text{-add}}$ has state set $\{1, 2, \ldots, 16\}$, input alphabet Σ_2^3, final states $\{1, 7, 11\}$, initial state 1, and its transition function $\delta_{F\text{-add}}$ is given in Table 1. Note that $M_{F\text{-add}}$ works for all Fibonacci representations; a DFA working only for canonical Fibonacci representations can be obtained by intersecting $M_{F\text{-add}}$ with a DFA that accepts only canonical Fibonacci representations.

$\delta_{F\text{-add}}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
[0,0,0], [0,1,1], [1,0,1]	1	4		5		2	8	3			6	10				
[0,1,0], [1,0,0], [1,1,1]	3	6		4		1			5	9	7	2				
[0,0,1]	2	5	8			10	11	1			4	14	15		3	
[1,1,0]		7		6	9	3			4	12	13	1		16		5

Table 1: Transition function for $M_{F\text{-add}}$ computing addition in Fibonacci representation

Using $M_{F\text{-add}}$ and other derived DFAs such as one for the $<$ relation, which can be defined in the context of $\mathrm{Th}(\mathbb{N}, 0, 1, +)$ by $a < b := \neg \exists c (b + c = a)$, our full decision procedure is as follows.

Procedure 2 (Decision procedure for Fibonacci-automatic words)**.**
Input: $m, n \in \mathbb{N}$, m DFAOs witnessing Fibonacci-automatic words $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m$, a first-order proposition with n free variables $\varphi(v_1, v_2, \ldots, v_n)$ using constants and relations definable in $\mathrm{Th}(\mathbb{N}, 0, 1, +)$ and indexing into $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m$.
Output: DFA with input alphabet Σ_2^n accepting $\{(k_1, k_2, \ldots, k_n)_F \,:\, \varphi(k_1, k_2, \ldots, k_n) \text{ holds}\}$.

3 The Rote-Fibonacci word and avoiding the pattern xxx^R

In this section, we show how to apply Procedure 2 to a problem involving infinite binary words avoiding the pattern xxx^R. Although avoiding patterns with reversal has been considered before (e.g., [37, 7, 18, 6]), it seems this particular problem has not been studied.

If our goal is just to produce some infinite binary word avoiding xxx^R, then one solution is easy: $(\mathtt{01})^\omega$ clearly avoids xxx^R. Thus, a more interesting question is whether there exists an aperiodic infinite binary word avoiding xxx^R. To answer this question, we study a special infinite word, which we call the *Rote-Fibonacci word*. (We name it as such because it is a special case of a class of words discussed in 1994 by Rote [39].)

Theorem 3. *The Rote-Fibonacci word* $\mathbf{r}$ *has the following equivalent descriptions:*

1. *As the output of the transducer depicted in Figure 2 on input* $\mathbf{f}$*.*

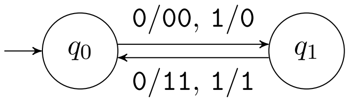

Figure 2: Transducer converting Fibonacci words to Rote-Fibonacci words

2. *As* $\tau(h^\omega(a))$*, where* h *and* τ *are morphisms defined by*

$$\begin{array}{lllllllll} h: & a \mapsto ab_1 & b \mapsto a & a_0 \mapsto a_2 b & a_1 \mapsto a_0 b_0 & a_2 \mapsto a_1 b_2 & b_0 \mapsto a_0 & b_1 \mapsto a_1 & b_2 \mapsto a_2 \\ \tau: & a \mapsto \mathtt{0} & b \mapsto \mathtt{1} & a_0 \mapsto \mathtt{0} & a_1 \mapsto \mathtt{1} & a_2 \mapsto \mathtt{1} & b_0 \mapsto \mathtt{0} & b_1 \mapsto \mathtt{0} & b_2 \mapsto \mathtt{1} \end{array}$$

3. *As the infinite binary word generated by the (incomplete) DFAO depicted in Figure 3.*

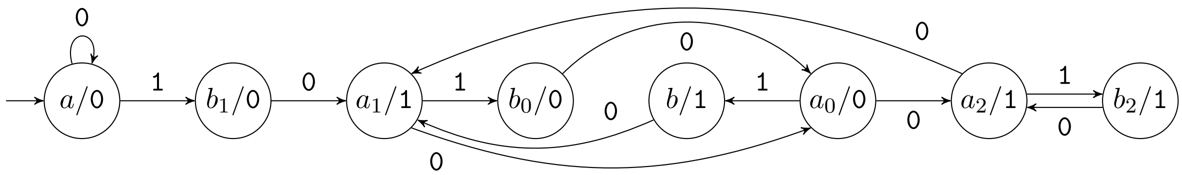

Figure 3: Canonical Fibonacci representation DFAO generating the Rote-Fibonacci word

4. *As* $\lim_{n\to\infty} R_n$*, where the finite Rote-Fibonacci words* R_n *are defined by* $R_0 := \mathtt{0}$*,* $R_1 := \mathtt{00}$*, and for* $n \geq 3$*,*

$$R_n := \begin{cases} R_{n-1}R_{n-2}, & \text{if } n \equiv 0 \pmod 3, \\ R_{n-1}\overline{R_{n-2}}, & \text{if } n \not\equiv 0 \pmod 3. \end{cases}$$

5. *As the sequence obtained from the binary complement of the Fibonacci sequence* $\overline{\mathbf{f}} = \mathtt{1011010110110}\cdots$ *as follows: change every second* $\mathtt{1}$ *that appears to* -1 *(which we write as* $\overline{1}$ *for clarity, obtaining* $\mathtt{10\overline{1}10\overline{1}01\overline{1}01\overline{1}0}\cdots$*), then take the running sum (obtaining* $\mathtt{1101100100100}\cdots$*), and finally, complement it to obtain* $\mathbf{r} = \mathtt{0010011011011}\cdots$*.*

6. *As* $\rho(g^\omega(a))$*, where* g *and* ρ *are morphisms defined by*

$$\begin{array}{lllll} g: & a \mapsto abcab & b \mapsto cda & c \mapsto cdacd & d \mapsto abc \\ \rho: & a \mapsto \mathtt{0} & b \mapsto \mathtt{0} & c \mapsto \mathtt{1} & d \mapsto \mathtt{1} \end{array}$$

Theorem 4. *The Rote-Fibonacci word* $\mathbf{r}$ *is aperiodic, avoids the pattern* xxx^R*, and is mirror-invariant (hence also avoids the pattern* xx^Rx^R*).*

Proof. We ran Procedure 2 on input the DFAO depicted in Figure 3 and the following predicates.

$$\exists n \forall i (i \geq n \rightarrow \mathbf{r}[i] = \mathbf{r}[i+p]).$$
$$\exists i \forall t (t < n \rightarrow (\mathbf{r}[i+t] = \mathbf{r}[i+t+n] \wedge \mathbf{r}[i+t] = \mathbf{r}[i+3n-1-t])).$$
$$\exists j \forall t (t \leq n \rightarrow \mathbf{r}[i+t] = \mathbf{r}[j-t]).$$

The first predicate says "$\mathbf{r}$ is eventually periodic with period p"; the output of Procedure 2 accepts only $(0)_F$, so $\mathbf{r}$ is aperiodic. The second predicate says "$\mathbf{r}$ contains the pattern xxx^R of length $3n$"; the output of Procedure 2 accepts only $(0)_F$, so $\mathbf{r}$ avoids the pattern xxx^R. The third predicate says "$\mathbf{r}[i\,..\,i+n]^R$ appears in $\mathbf{r}$"; the output of Procedure 2 accepts all $(i,n)_F$ with $i, n \in \mathbb{N}$, so $\mathbf{r}$ is mirror-invariant. ∎

The Rote-Fibonacci word has (essentially) appeared before in several places. In a 2009 preprint of Monnerot-Dumaine [32], the author studies a plane fractal called the "Fibonacci word fractal", specified by certain drawing instructions which can be coded over the alphabet $\{S, R, L\}$ by taking $g^\omega(a)$ from Theorem 3(6) and applying the coding $\gamma : a \mapsto S, b \mapsto R, c \mapsto S, d \mapsto L$. Here, S means "move straight one unit", "R" means "right turn one unit", and "L" means "left turn one unit". More recently, Blondin Massé, Brlek, Labbé, and Mendès France studied a remarkable sequence of words closely related to $\mathbf{r}$ [8, 9, 10]. For example, in their paper "Fibonacci snowflakes" [8], they defined a certain sequence q_i which has the following relationship to g: let $\xi(a) = \xi(b) := L$ and $\xi(c) = \xi(d) := R$; then $R\xi(g^n(a)) = q_{3n+2}L$.

4 Combining representations and avoiding additive squares

In this section, we demonstrate the robustness of Procedure 2 by showing how it can be modified to handle an avoidability problem where multiple different representations arise.

It is currently unknown, and a relatively famous open problem, whether there exists an infinite word over a *finite* subset of $\mathbb{N} \setminus \{0\}$ that avoids additive squares [11, 34, 30], although it is known that additive cubes can be avoided over an alphabet of size 3 [38]. However, it is easy to avoid additive squares over any *infinite* subset of $\mathbb{N} \setminus \{0\}$; for example, any sequence that grows sufficiently fast will avoid additive squares. Thus, it is reasonable to ask about the *lexicographically least* sequence over $\mathbb{N} \setminus \{0\}$ that avoids additive squares. This sequence begins $\texttt{12131214212521312134121 72}\cdots$ but even its boundedness remains an open problem.

We consider the following variation on this problem. Instead of considering arbitrary sequences, we start with a sequence $\mathbf{b} \in (\mathbb{N} \setminus \{0\})^\omega$ and from it construct the sequence $S(\mathbf{b})$ defined by $S(\mathbf{b})[n] := \mathbf{b}[\nu_2(n+1)]$ for all $n \in \mathbb{N}$, where $\nu_2(n) := \max_{2^i \mid n} i$. For example, if $\mathbf{b} := \texttt{12345}\cdots$, then $S(\mathbf{b}) = \texttt{1213121412131215}\cdots$, which is the so-called "ruler sequence" and is known to be the lexicographically least square-free sequence over $\mathbb{N} \setminus \{0\}$ [29]. Our variation of the problem is to seek a description of the lexicographically least sequence over $\mathbb{N} \setminus \{0\}$ of the form $S(\mathbf{b})$ that avoids additive squares.

Theorem 5. *The lexicographically least sequence over* $\mathbb{N} \setminus \{0\}$ *of the form* $S(\mathbf{b})$ *that avoids additive squares is defined by* $\mathbf{b}[i] := F_{i+2}$.

Proof. First, we show that $\mathbf{a} := S(\mathbf{b}) = \prod_{k=1}^{\infty} \mathbf{b}[\nu_2(k)] = \prod_{k=1}^{\infty} F_{\nu_2(k)+2}$ avoids additive squares. For $m, n, j \in \mathbb{N}$, let $A(m,n,j)$ denote the number of occurrences of j in $\nu_2(m+1), \ldots, \nu_2(m+n)$. A careful treatment of (in)equalities involving floors reveals that for all $m, m', n, j \in \mathbb{N}$, we have $|A(m',n,j) - A(m,n,j)| \leq 1$.

Note that for all $i,n \in \mathbb{N}$, we have $\sum_{k=i}^{i+n-1} \mathbf{a}[k] = \sum_{j=0}^{\lfloor \log_2(i+n) \rfloor} A(i,n,j)F_{j+2}$, so for adjacent blocks of length n, $\sum_{k=i+n}^{i+2n-1} \mathbf{a}[k] - \sum_{k=i}^{i+n-1} \mathbf{a}[k] = \sum_{j=0}^{\lfloor \log_2(i+2n) \rfloor} (A(i+n,n,j) - A(i,n,j))F_{j+2}$. Hence, $\mathbf{a}[i\,.\,.\,i+2n-1]$ is an additive square iff $\sum_{j=0}^{\lfloor \log_2(i+2n) \rfloor} (A(i+n,n,j) - A(i,n,j))F_{j+2} = 0$, and by above, each $A(i+n,n,j) - A(i,n,j) \in \{-1,0,1\}$.

The above suggests that we can take advantage of "unnormalized" Fibonacci representation in our computations. For $\Sigma \subseteq \mathbb{Z}$ and $w \in \Sigma^*$, we let the unnormalized Fibonacci representation $\langle w \rangle_{uF}$ be defined in the same way as $\langle w \rangle_F$, except over the alphabet Σ.

In order to use Procedure 2, we need two auxiliary DFAs: one that, given $i,n \in \mathbb{N}$ (in any representation; we found that base 2 works), computes $\langle A(i+n,n,\_) - A(i,n,\_) \rangle_{uF}$, and another that, given $w \in \{-1,0,1\}^*$, decides whether $\langle w \rangle_{uF} = 0$. The first task can be done by a 6-state (incomplete) DFA M_{add22F} that accepts the language $\{z \in (\Sigma_2^2 \times \{-1,0,1\})^* \ : \ \forall j(\pi_3(z)[j] = A(\langle \pi_1(z) \rangle_2 + \langle \pi_2(z) \rangle_2, \langle \pi_2(z) \rangle_2, j) - A(\langle \pi_1(z) \rangle_2, \langle \pi_2(z) \rangle_2, j))\}$. The second task can be done by a 5-state (incomplete) DFA $M_{\text{1uFisZero}}$ that accepts the language $\{w \in \{-1,0,1\}^* \ : \ \langle w \rangle_{uF} = 0\}$.

We applied a modified Procedure 2 to the predicate $n \geq 1 \wedge \exists w(\texttt{add22F}(i,n,w) \wedge \texttt{1uFisZero}(w))$ and obtained as output a DFA that accepts nothing, so $\mathbf{a}$ avoids additive squares.

Next, we show that $\mathbf{a}$ is the lexicographically least sequence over $\mathbb{N} \setminus \{0\}$ of the form $S(\mathbf{b})$ that avoids additive squares.

Note that for all $\mathbf{x},\mathbf{y} \in \mathbb{N} \setminus \{0\}$, $S(\mathbf{x}) < S(\mathbf{y})$ iff $\mathbf{x} < \mathbf{y}$ in the lexicographic ordering. Thus, we show that if any entry $\mathbf{b}[s]$ with $\mathbf{b}[s] > 1$ is changed to some $t \in [1, \mathbf{b}[s]-1]$, then $\mathbf{a} = S(\mathbf{b})$ contains an additive square using only the first occurrence of the change at $\mathbf{a}[2^s - 1]$. More precisely, we show that for all $s,t \in \mathbb{N}$ with $t \in [1, F_{s+2}-1]$, there exist $i,n \in \mathbb{N}$ with $n \geq 1$ and $i + 2n < 2^{s+1}$ such that either ($2^s - 1 \in [i, i+n-1]$ and $\sum_{k=i+n}^{i+2n-1} \mathbf{a}[k] - \sum_{k=i}^{i+n-1} \mathbf{a}[k] + t = 0$) or ($2^s - 1 \in [i+n, i+2n-1]$ and $\sum_{k=i+n}^{i+2n-1} \mathbf{a}[k] - \sum_{k=i}^{i+n-1} \mathbf{a}[k] - t = 0$).

Setting up for a modified Procedure 2, we use the following predicate, which says "r is a power of 2 and changing $\mathbf{a}[r-1]$ to any smaller number results in an additive square in the first $2r$ positions", and six auxiliary DFAs. Note that all arithmetic and comparisons are in base 2.

$$\begin{aligned}
&\texttt{powOf2}(r) \wedge \forall t((t \geq 1 \wedge t < r \wedge \texttt{canonFib}(t)) \rightarrow \exists i \exists n(n \geq 1 \wedge i + 2n < 2r \wedge \\
&\quad ((i < r \wedge r \leq i + n \wedge \forall w(\texttt{add22F}(i,n,w) \rightarrow \forall x(\texttt{bitAdd}(t,w,x) \rightarrow \texttt{2uFisZero}(x)))) \vee \\
&\quad\ (i + n < r \wedge r \leq i + 2n \wedge \forall w(\texttt{add22F}(i,n,w) \rightarrow \forall x(\texttt{bitSub}(t,w,x) \rightarrow \texttt{2uFisZero}(x))))))).
\end{aligned}$$

$$\begin{aligned}
L(M_{\text{powOf2}}) &= \{w \in \Sigma_2^* \ : \ \exists n(w = (2^n)_2)\}. \\
L(M_{\text{canonFib}}) &= \{w \in \Sigma_2^* \ : \ \exists n(w = (n)_F)\}. \\
L(M_{\text{bit(Add/Sub)}}) &= \{z \in (\Sigma_2 \times \{-1,0,1\} \times \{-1,0,1,2\})^* \ : \ \forall i(\pi_1(z)[i] \pm \pi_2(z)[i] = \pi_3(z)[i])\}. \\
L(M_{\text{2uFisZero}}) &= \{w \in \{-1,0,1,2\}^* \ : \ \langle w \rangle_{uF} = 0\}.
\end{aligned}$$

We applied a modified Procedure 2 to the above predicate and auxiliary DFAs and obtained as output M_{powOf2}, so $\mathbf{a}$ is the lexicographically least sequence over $\mathbb{N} \setminus \{0\}$ of the form $S(\mathbf{b})$ that avoids additive squares. ∎

5 Enumeration and finite Fibonacci words

In this section, we exhibit how Procedure 2 can be used to solve enumeration problems related to Fibonacci-automatic words by using it to count the total number of occurrences of squares in the finite Fibonacci words. We define the finite Fibonacci words by $X_0 := \epsilon$, $X_1 := \texttt{1}$, $X_2 := \texttt{0}$, and $X_n := X_{n-1}X_{n-2}$ for $n \geq 3$. Note that $|X_n| = F_n$ for all $n \in \mathbb{N}$ and X_n is a prefix of $\mathbf{f}$ for all $n \geq 2$. This particular enumeration problem was solved in [20, Theorem 2], but their solution contains a small error (their coefficient of F_{n-2} is 1 but should be 4; note that their F_n and X_n are indexed differently from ours), which was first pointed out to us by Kalle Saari.

Theorem 6. *For each $n \in \mathbb{N}$, let $B(n)$ denote the total number of occurrences of squares in X_n. Then for all $n \geq 3$, $B(n+1) = \frac{4}{5}nF_{n+1} - \frac{2}{5}(n+6)F_n - 4F_{n-1} + n + 1$.*

Proof. Let $L := \{(n,i,j)_F \in (\Sigma_2^3)^* \ : \ j \geq 1, i+2j \leq n, \mathbf{f}[i..i+2j-1] \text{ is a square}\}$. We ran Procedure 2 on input the DFAO depicted in Figure 1 and the predicate $j \geq 1 \wedge i + 2j \leq n \wedge \forall t(t < j \rightarrow \mathbf{f}[i+t] = \mathbf{f}[i+j+t])$, obtaining a 27-state (incomplete) DFA M accepting L.

For each $n \in \mathbb{N}$, let $b(n)$ denote the total number of occurrences of squares in $\mathbf{f}[0..n-1]$. Note that for all $n \in \mathbb{N}$, $b(n) = |\{w \in L \ : \ \pi_1(w)_F = n\}|$. Since $i + 2j \leq n$ implies $i, j \leq n$, which in turn implies $|(i)_F|, |(j)_F| \leq |(n)_F|$, it follows that the first coordinate of any $w \in L$ has no leading $\mathtt{0}$s, so $b(n) = |\{w \in L \ : \ \pi_1(w) = (n)_F\}|$. We can compute $b(n)$ using M as follows.

First, arbitrarily number the states of M as $\{1, 2, \ldots, 27\}$. Then, define $M_{\mathtt{0}}, M_{\mathtt{1}} \in \mathbb{N}^{27\times 27}$ by $M_a[k,l] := \left|\{(b,c) \in \Sigma_2^2 \ : \ \delta_M([a,b,c],k) = l\}\right|$ for $a \in \Sigma_2$, where δ_M is the transition function of M. Also define $v_{\text{init}}, v_{\text{fin}} \in \mathbb{N}^{27}$ to be the characteristic vectors of the initial and final states of M respectively. Now, we have that for all $n \in \mathbb{N}$, $b(n) = v_{\text{init}}^T \left(\prod_{i=0}^{|(n)_F|-1} M_{(n)_F[i]}\right) v_{\text{fin}}$.

For all $n \geq 2$, since $X_n = \mathbf{f}[0 \ldots F_n - 1]$, we have $B(n) = b(F_n) = b(\langle \mathtt{10}^{n-2}\rangle_F) = v_{\text{init}}^T M_{\mathtt{1}} M_{\mathtt{0}}^{n-2} v_{\text{fin}}$. We computed $M_{\mathtt{0}}$ and its minimal polynomial $\mu_{M_{\mathtt{0}}}(x) = x^4(x-1)^2(x+1)^2(x^2-x-1)^2$. For each $(i,j) \in \{1,2,\ldots,27\}^2$, the sequence $(M_{\mathtt{0}}^n[i,j])_{n\geq 4}$ satisfies the homogeneous linear recurrence relation defined by the expanded form of $\mu_{M_{\mathtt{0}}}$, whose characteristic polynomial is $\frac{\mu_{M_{\mathtt{0}}}(x)}{x^4}$. Hence, by the theory of linear recurrences, there exists $\vec{c} \in \mathbb{R}^8$ such that for all $n \geq 5$, we have $B(n+1) = (c_1 n + c_2)\phi^n + (c_3 n + c_4)(-\phi)^{-n} + (c_5 n + c_6) + (c_7 n + c_8)(-1)^n$, where $\phi := \frac{\sqrt{5}+1}{2}$. We computed $B(6), B(7), \ldots, B(13)$ to solve for $\vec{c} = (\frac{2}{5}, -\frac{2}{25}\sqrt{5} - 2, \frac{2}{5}, \frac{2}{25}\sqrt{5} - 2, 1, 1, 0, 0)$. Finally, using Binet's formula $F_n = \frac{\phi^n - (-\phi)^{-n}}{\phi + \phi^{-1}}$, simplification yields $B(n+1) = \frac{4}{5}nF_{n+1} - \frac{2}{5}(n+6)F_n - 4F_{n-1} + n + 1$, which is valid for all $n \geq 5$. By inspection, the formula is also valid for $n \in \{3, 4\}$. ∎

6 Computational feasibility and future work

There was substantial skepticism that any implementation of Procedure 2 would be practical because the best known worst case running time for such a procedure is not elementary recursive. While our implementation of Procedure 2 likely also has non-elementary recursive worst case running time, it has worked quite well in practice, and has given us many results about Fibonacci-automatic words that we present in the full version of this paper. We sometimes encounter automata with more than 10^6 states during intermediate computations, but this is usually because of mistakes in entering the input. Most computations involving predicates encoding "natural" properties of words, including the ones in our proof of Theorem 5, are finished in just a few seconds. Only in a few cases have we encountered extremely simple predicates for which our implementation of Procedure 2 seemingly runs forever. With the relative success we encountered with Fibonacci representation, in the future, we hope to be able to implement Procedure 2, or some variant of it, in more exotic numeration systems, such as α-Ostrowski representations with $\alpha \neq \phi$ and mixed radix representations.

7 Acknowledgments

Eric Rowland thought about the problem we consider in section 4 with us in 2010, and at that time was able to prove the first half of Theorem 5, namely that the word $\prod_{k=1}^{\infty} F_{\nu_2(k)+2}$ avoids additive squares. We acknowledge his prior work on this problem and thank him for allowing us to quote it here. We thank Kalle Saari for bringing our attention to the small error in [20]. We thank Narad Rampersad and Michel Rigo for useful suggestions.

The full version of this paper is available at `http://arxiv.org/abs/1406.0670`.

References

[1] C. Ahlbach, J. Usatine, C. Frougny, and N. Pippenger. Efficient algorithms for Zeckendorf arithmetic. *Fibonacci Quart.*, 51:249–256, 2013.

[2] J.-P. Allouche, N. Rampersad, and J. Shallit. Periodicity, repetitions, and orbits of an automatic sequence. *Theoret. Comput. Sci.*, 410:2795–2803, 2009.

[3] J.-P. Allouche and J. Shallit. *Automatic Sequences: Theory, Applications, Generalizations.* Cambridge University Press, 2003.

[4] J. Berstel. Fonctions rationnelles et addition. In M. Blab, editor, *Théorie des Langages, École de printemps d'informatique théorique*, pages 177–183. LITP, 1982.

[5] J. Berstel. Fibonacci words—a survey. In G. Rozenberg and A. Salomaa, editors, *The Book of L*, pages 13–27. Springer-Verlag, 1986.

[6] B. Bischoff, J. D. Currie, and D. Nowotka. Unary patterns with involution. *Internat. J. Found. Comp. Sci.*, 23:1641–1652, 2012.

[7] B. Bischoff and D. Nowotka. Pattern avoidability with involution. In *WORDS 2011*, pages 65–70, 2011. Available at `http://rvg.web.cse.unsw.edu.au/eptcs/content.cgi?WORDS2011`.

[8] A. Blondin Massé, S. Brlek, A. Garon, and S. Labbé. Two infinite families of polyominoes that tile the plane by translation in two distinct ways. *Theoret. Comput. Sci.*, 412:4778–4786, 2011.

[9] A. Blondin Massé, S. Brlek, S. Labbé, and M. Mendès France. Fibonacci snowflakes. *Ann. Sci. Math. Québec*, 35:141–152, 2011.

[10] A. Blondin Massé, S. Brlek, S. Labbé, and M. Mendès France. Complexity of the Fibonacci snowflake. *Fractals*, 20:257–260, 2012.

[11] T. C. Brown and A. R. Freedman. Arithmetic progressions in lacunary sets. *Rocky Mountain J. Math.*, 17:587–596, 1987.

[12] V. Bruyère and G. Hansel. Bertrand numeration systems and recognizability. *Theoret. Comput. Sci.*, 181:17–43, 1997.

[13] V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire. Logic and p-recognizable sets of integers. *Bull. Belgian Math. Soc.*, 1:191–238, 1994. Corrigendum, *Bull. Belg. Math. Soc.* **1** (1994), 577.

[14] J. R. Büchi. Weak secord-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960. Reprinted in S. Mac Lane and D. Siefkes, eds., *The Collected Works of J. Richard Büchi*, Springer-Verlag, 1990, pp. 398–424.

[15] L. Carlitz. Fibonacci representations. *Fibonacci Quart.*, 6:193–220, 1968.

[16] E. Charlier, N. Rampersad, and J. Shallit. Enumeration and decidable properties of automatic sequences. *Internat. J. Found. Comp. Sci.*, 23:1035–1066, 2012.

[17] A. Cobham. Uniform tag sequences. *Math. Systems Theory*, 6:164–192, 1972.

[18] J. D. Currie. Pattern avoidance with involution. Available at `http://arxiv.org/abs/1105.2849`, 2011.

[19] A. S. Fraenkel. Systems of numeration. *Amer. Math. Monthly*, 92:105–114, 1985.

[20] A. S. Fraenkel and J. Simpson. The exact number of squares in Fibonacci words. *Theoret. Comput. Sci.*, 218:95–106, 1999.

[21] C. Frougny. Linear numeration systems of order two. *Inform. Comput.*, 77:233–259, 1988.

[22] C. Frougny. Fibonacci representations and finite automata. *IEEE Trans. Inform. Theory*, 37:393–399, 1991.

[23] C. Frougny. Representations of numbers and finite automata. *Math. Systems Theory*, 25:37–60, 1992.

[24] C. Frougny and B. Solomyak. On representation of integers in linear numeration systems. In M. Pollicott and K. Schmidt, editors, *Ergodic Theory of $\mathbb{Z}^d$ Actions (Warwick, 1993–1994)*, volume 228 of *London Mathematical Society Lecture Note Series*, pages 345–368. Cambridge University Press, 1996.

[25] D. Goc, D. Henshall, and J. Shallit. Automatic theorem-proving in combinatorics on words. In N. Moreira and R. Reis, editors, *CIAA 2012*, volume 7381 of *Lecture Notes in Computer Science*, pages 180–191. Springer-Verlag, 2012.

[26] D. Goc, H. Mousavi, and J. Shallit. On the number of unbordered factors. In A.-H. Dediu, C. Martin-Vide, and B. Truthe, editors, *LATA 2013*, volume 7810 of *Lecture Notes in Computer Science*, pages 299–310. Springer-Verlag, 2013.

[27] D. Goc, K. Saari, and J. Shallit. Primitive words and Lyndon words in automatic and linearly recurrent sequences. In A.-H. Dediu, C. Martin-Vide, and B. Truthe, editors, *LATA 2013*, volume 7810 of *Lecture Notes in Computer Science*, pages 311–322. Springer-Verlag, 2013.

[28] D. Goc, L. Schaeffer, and J. Shallit. The subword complexity of k-automatic sequences is k-synchronized. In M.-P. Béal and O. Carton, editors, *DLT 2013*, volume 7907 of *Lecture Notes in Computer Science*, pages 252–263. Springer-Verlag, 2013.

[29] M. Guay-Paquet and J. Shallit. Avoiding squares and overlaps over the natural numbers. *Discrete Math.*, 309:6245–6254, 2009.

[30] L. Halbeisen and N. Hungerbühler. An application of Van der Waerden's theorem in additive number theory. *INTEGERS: Elect. J. of Combin. Number Theory*, 0:#A7, 2000. `http://www.integers-ejcnt.org/vol0.html`.

[31] C. G. Lekkerkerker. Voorstelling van natuurlijke getallen door een som van getallen van Fibonacci. *Simon Stevin*, 29:190–195, 1952.

[32] A. Monnerot-Dumaine. The Fibonacci word fractal. Published electronically at `http://hal.archives-ouvertes.fr/hal-00367972/fr/`, 2009.

[33] A. Ostrowski. Bemerkungen zur Theorie der Diophantischen Approximationen. *Abh. Math. Sem. Hamburg*, 1:77–98,250–251, 1922. Reprinted in *Collected Mathematical Papers*, Vol. 3, pp. 57–80.

[34] G. Pirillo and S. Varricchio. On uniformly repetitive semigroups. *Semigroup Forum*, 49:125–129, 1994.

[35] M. Presburger. Über die Volständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Sparawozdanie z I Kongresu matematyków krajów slowianskich*, pages 92–101, 395. Warsaw, 1929.

[36] M. Presburger. On the completeness of a certain system of arithmetic of whole numbers in which addition occurs as the only operation. *Hist. Phil. Logic*, 12:225–233, 1991.

[37] N. Rampersad and J. Shallit. Words avoiding reversed subwords. *J. Combin. Math. Combin. Comput.*, 54:157–164, 2005.

[38] M. Rao. On some generalizations of abelian power avoidability. Preprint, 2013.

[39] G. Rote. Sequences with subword complexity $2n$. *J. Number Theory*, 46:196–213, 1994.

[40] J. Shallit. Decidability and enumeration for automatic sequences: a survey. In A. A. Bulatov and A. M. Shur, editors, *CSR 2013*, volume 7913 of *Lecture Notes in Computer Science*, pages 49–63. Springer-Verlag, 2013.

[41] J. O. Shallit. A generalization of automatic sequences. *Theoret. Comput. Sci.*, 61:1–16, 1988.

[42] E. Zeckendorf. Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres Lucas. *Bull. Soc. Roy. Liège*, 41:179–182, 1972.