Decision Algorithms for Fibonacci-Automatic Words, II: Related Sequences and Avoidability

Chen Fei Du[1], Hamoon Mousavi[1], Eric Rowland[2], Luke Schaeffer[3], and Jeffrey Shallit[1]

April 10, 2015

Abstract

We use a decision procedure for the "Fibonacci-automatic" words to solve problems about a number of different sequences. In particular, we prove that there exists an aperiodic infinite binary word avoiding the pattern xxx^R. This is the first avoidability result concerning a nonuniform morphism proven purely mechanically.

1 Introduction

In the paper [18], we introduced a decision procedure for the "Fibonacci-automatic" words. These are infinite words $(a_n)_{n\geq 0}$ that are generated by finite automata as follows: the input to the automaton is the Fibonacci (or "Zeckendorf" representation) w of the number n. The output associated with the last state reached upon reading w is then a_n.

We recall this representation more precisely. The Fibonacci numbers are defined by $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$.

Every non-negative integer 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. See [19, 16, 25, 7, 12].

Such a representation can be written as a binary string $a_1a_2\cdots a_n$ representing the integer $\sum_{1\leq i\leq n} a_iF_{n+2-i}$. For $w = a_1a_2\cdots a_n \in \Sigma_2^*$, we define $[a_1a_2\cdots a_n]_F := \sum_{1\leq i\leq n} a_iF_{n+2-i}$, even if $a_1a_2\cdots a_n$ has leading zeroes or consecutive 1's. By $(n)_F$ we mean the *canonical* Fibonacci representation for the integer n, having no leading zeroes or consecutive 1's.

The canonical example of a Fibonacci-automatic word is the infinite Fibonacci word

$$\mathbf{f} = (f_i)_{i\geq 0} = 01001010\cdots$$

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

[2]Department of Mathematics, Université of Liege, Liège, Belgium; erowland@ulg.ac.be

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

where f_i is the last (least significant) digit in the Fibonacci representation of i. Properties of this word provable by our decision procedure were given in [18].

In this paper we apply our method to some other Fibonacci-automatic sequences. In particularly, we study a particular word, the Rote-Fibonacci word, and show that it avoids the pattern xxx^R. To the best of our knowledge, this is the first nontrivial avoidability property proved using a decision procedure based on Fibonacci representation.

The main technique is to write assertions about sequences in the first-order theory of the natural numbers with +, allowing indexing into the Fibonacci-automatic sequences under consideration. Once this is done, the decision procedure of [18] can be used to resolve the truth of the assertion. Our implementation of the decision procedure is called `Walnut`, and is available for free download at

`https://www.cs.uwaterloo.ca/~shallit/papers.html` .

2 Avoiding the pattern xxx^R and the Rote-Fibonacci word

In this section we show how to apply our decision method to an interesting and novel avoidance property: avoiding the pattern xxx^R . An example matching this pattern in English is a factor of the word `bepepper`, with $x =$ `ep`. Here, however, we are concerned only with the binary alphabet $\Sigma_2 = \{0, 1\}$.

Although avoiding patterns with reversal has been considered before (e.g., [21, 2, 9, 1]), it seems our particular problem has not been studied.

If our goal is just to produce some infinite word avoiding xxx^R, then a solution seems easy: namely, the infinite word $(01)^\omega$ clearly avoids xxx^R, since if $|x| = n$ is odd, then the second factor of length n cannot equal the first (since the first symbol differs), while if $|x| = n$ is even, the first symbol of the third factor of length n cannot be the last symbol of x. In a moment we will see that even this question seems more subtle than it first appears, but for the moment, we'll change our question to

Are there infinite aperiodic binary words avoiding xxx^R?

To answer this question, we'll study a special infinite word, which we call the *Rote-Fibonacci word*. (The name comes from the fact that it is a special case of a class of words discussed in 1994 by Rote [23].) Consider the following transducer T:

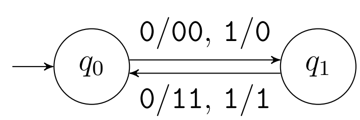

Figure 1: Transducer converting Fibonacci words to Rote-Fibonacci words

This transducer acts on words by following the transitions and outputting the concatenation of the outputs associated with each transition. Thus, for example, the input 01001 gets transduced to the output 00100110.

Theorem 1. *The Rote-Fibonacci word*

$$\mathbf{r} = 0010011011011001001101101100100100110110010010011011001001001\cdots = r_0 r_1 r_2 \cdots$$

has the following equivalent descriptions:

0. As the output of the transducer T, starting in state 0, on input $\mathbf{f}$.

1. As $\tau(h^\omega(a))$ where h and τ are defined by

$$\begin{aligned} h(a) &= ab_1 & \tau(a) &= 0 \\ h(b) &= a & \tau(b) &= 1 \\ h(a_0) &= a_2 b & \tau(a_0) &= 0 \\ h(a_1) &= a_0 b_0 & \tau(a_1) &= 1 \\ h(a_2) &= a_1 b_2 & \tau(a_2) &= 1 \\ h(b_0) &= a_0 & \tau(b_0) &= 0 \\ h(b_1) &= a_1 & \tau(b_1) &= 0 \\ h(b_2) &= a_2 & \tau(b_2) &= 1 \end{aligned}$$

2. As the binary sequence generated by the following DFAO, with outputs given in the states, and inputs in the Fibonacci representation of n.

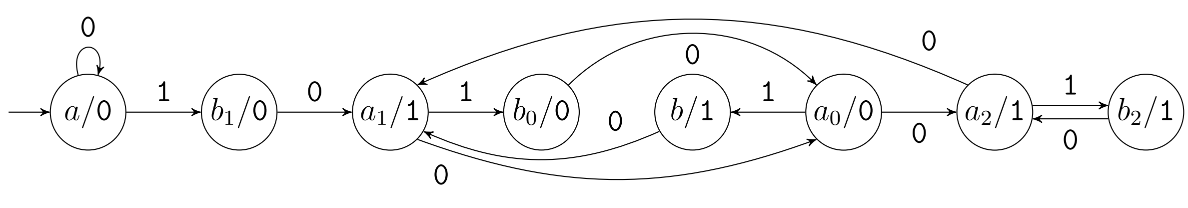

Figure 2: Canonical Fibonacci representation DFAO generating the Rote-Fibonacci word

3. As the limit, as $n \to \infty$, of the sequence of finite Rote-Fibonacci words $(R_n)_n$ defined as follows: $R_0 = 0$, $R_1 = 00$, and for $n \geq 3$

$$R_n = \begin{cases} R_{n-1} R_{n-2}, & \text{if } n \equiv 0 \pmod 3; \\ R_{n-1} \overline{R_{n-2}}, & \text{if } n \equiv 1,2 \pmod 3. \end{cases}$$

4. As the sequence obtained from the Fibonacci sequence $\mathbf{f} = f_0 f_1 f_2 \cdots = 0100101001001\cdots$ as follows: first, change every 0 to 1 and every one to 0 in $\mathbf{f}$, obtaining $\overline{\mathbf{f}} = 1011010110110\cdots$.

Next, in $\overline{\mathbf{f}}$ *change every second* 1 *that appears to* -1 *(which we write as* $\overline{1}$ *for clarity):* $10\overline{1}10\overline{1}01\overline{1}01\overline{1}0\cdots$. *Now take the running sum of this sequence, obtaining* $1101100100100\cdots$, *and finally, complement it to get* $\mathbf{r}$.

5. *As* $\rho(g^\omega(a))$, *where* g *and* ρ *are defined as follows*

$$\begin{aligned} g(a) &= abcab & \rho(a) &= 0 \\ g(b) &= cda & \rho(b) &= 0 \\ g(c) &= cdacd & \rho(c) &= 1 \\ g(d) &= abc & \rho(d) &= 1 \end{aligned}$$

Proof. (0) $\iff$ (3): Let $T_0(x)$ (resp., $T_1(x)$) denote the output of the transducer T starting in state q_0 (resp., q_1) on input x. Then a simple induction on n shows that $T_0(X_{n+1}) = R_n$ and $T_1(X_{n+1}) = \overline{R_n}$. We give only the induction step for the first claim:

$$\begin{aligned} T_0(X_{n+1}) &= T_0(X_n X_{n-1}) \\ &= \begin{cases} T_0(X_n)T_0(X_{n-1}), & \text{if } |X_n| \text{ is even;} \\ T_0(X_n)T_1(X_{n-1}), & \text{if } |X_n| \text{ is odd;} \end{cases} \\ &= \begin{cases} R_{n-1}R_{n-2}, & \text{if } n \equiv 0 \pmod 3; \\ R_{n-1}\overline{R_{n-2}}, & \text{if } n \not\equiv 0 \pmod 3; \end{cases} \\ &= R_n. \end{aligned}$$

Here we have used the easily-verified fact that $|X_n| = F_n$ is even iff $n \equiv 0 \pmod 3$.

(1) $\iff$ (3): we verify by a tedious induction on n that for $n \geq 0$ we have

$$\begin{aligned} \tau(h^n(a)) &= \tau(h^{n+1}(a)) = R_n \\ \tau(h^n(a_i)) &= \tau(h^{n+1}(b_i)) = \begin{cases} R_i, & \text{if } n \equiv i \pmod 3; \\ \overline{R_i}, & \text{if } n \not\equiv i \pmod 3. \end{cases} \end{aligned}$$

(2) $\iff$ (4): Follows from the well-known transformation from automata to morphisms and vice versa (see, e.g., [15]).

(3) $\iff$ (4): We define some transformations on sequences, as follows:

- $C(x)$ denotes $\overline{x}$, the complement of x;
- $s(x)$ denotes the sequence arising from a binary sequence x by changing every second 1 to -1;
- $a(x)$ denotes the running sum of the sequence x; that is, if $x = a_1a_2a_3\cdots$ then $a(x)$ is $a_1(a_1+a_2)(a_1+a_2+a_3)\cdots$.

Note that

$$a(s(xy)) = \begin{cases} a(s(x))\ a(s(y)), & \text{if } |x|_1 \text{ even;} \\ a(s(x))\ C(a(s(y))), & \text{if } |x|_1 \text{ odd.} \end{cases}$$

Then we claim that $C(R_n) = a(s(C(X_{n+2})))$. This can be verified by induction on n. We give only the induction step:

$$\begin{aligned} a(s(C(X_{n+2}))) &= a(s(C(X_{n+1})C(X_n))) \\ &= \begin{cases} a(s(C(X_{n+1})))\ a(s(C(X_n))), & \text{if } C(X_{n+1})_1 \text{ even;} \\ a(s(C(X_{n+1})))\ C(a(s(C(X_n)))), & \text{if } C(X_{n+1})_1 \text{ odd;} \end{cases} \\ &= \begin{cases} C(R_{n-1})\ C(R_{n-2}), & \text{if } n \equiv 0 \pmod 3; \\ C(R_{n-1})\ R_{n-2}, & \text{if } n \not\equiv 0 \pmod 3; \end{cases} \\ &= R_n. \end{aligned}$$

(3) $\iff$ (5): Define γ by

$$\begin{aligned} \gamma(a) = \gamma(a_0) &= a \\ \gamma(b_0) = \gamma(b_1) &= b \\ \gamma(a_1) = \gamma(a_2) &= c \\ \gamma(b) = \gamma(b_2) &= d. \end{aligned}$$

We verify by a tedious induction on n that for $n \geq 0$ we have

$$\begin{aligned} g^n(a) &= \gamma(h^{3n}(a)) = \gamma(h^{3n}(a_0)) \\ g^n(b) &= \gamma(h^{3n}(b_0)) = \gamma(h^{3n}(b_1)) \\ g^n(c) &= \gamma(h^{3n}(a_1)) = \gamma(h^{3n}(a_2)) \\ g^n(d) &= \gamma(h^{3n}(b)) = \gamma(h^{3n}(b_2)). \end{aligned}$$

□

Corollary 2. *The first differences* $\Delta\mathbf{r}$ *of the Rote-Fibonacci word* $\mathbf{r}$*, taken modulo* 2*, give the complement of the Fibonacci word* $\overline{f}$*, with its first symbol omitted.*

Proof. Note that if $\mathbf{x} = a_0a_1a_2\cdots$ is a binary sequence, then $\Delta(C(\mathbf{x})) = -\Delta(\mathbf{x})$. Furthermore $\Delta(a(x)) = a_1a_2\cdots$. Now from the description in part 4, above, we know that $\mathbf{r} = C(a(s(C(\mathbf{f}))))$. Hence $\Delta(\mathbf{r}) = \Delta(C(a(s(C(\mathbf{f}))))) = -\Delta(a(s(C(\mathbf{f})))) = \mathrm{dr}(-s(C(\mathbf{f})))$, where dr drops the first symbol of its argument. Taking the last result modulo 2 gives the result. □

We are now ready to prove our avoidability result.

Theorem 3. *The Rote-Fibonacci word* $\mathbf{r}$ *avoids the pattern* xxx^R*.*

Proof. We use our decision procedure to prove this. A predicate is as follows:

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

When we run this on our program, we get the following log:

```
t < n with 7 states, in 36ms
 R[i + t] = R[i + t + n] with 245 states, in 1744ms
  R[i + t] = R[i + 3 * n - 1 - t] with 1751 states, in 14461ms
   R[i + t] = R[i + t + n] & R[i + t] = R[i + 3 * n - 1 - t] with 3305 states, in 565ms
    t < n => R[i + t] = R[i + t + n] & R[i + t] = R[i + 3 * n - 1 - t] with 2015 states, in 843ms
     At t < n => R[i + t] = R[i + t + n] & R[i + t] = R[i + 3 * n - 1 - t] with 3 states, in 747ms
      Ei At t < n => R[i + t] = R[i + t + n] & R[i + t] = R[i + 3 * n - 1 - t] with 2 states, in 0ms
overall time: 18396ms
```

Then the only length n accepted is $n = 0$, so the Rote-Fibonacci word $\mathbf{r}$ contains no occurrences of the pattern xxx^R. □

We now prove some interesting properties of $\mathbf{r}$.

Theorem 4. *The minimum $q(n)$ over all periods of all length-n factors of the Rote-Fibonacci word is as follows:*

$$q(n) = \begin{cases} 1, & \text{if } 1 \leq n \leq 2; \\ 2, & \text{if } n = 3; \\ F_{3j+1}, & \text{if } j \geq 1 \text{ and } L_{3j} \leq n < L_{3j+2}; \\ L_{3j+1}, & \text{if } j \geq 1 \text{ and } L_{3j+2} \leq n < L_{3j+2} + F_{3j-2}; \\ F_{3j+2} + L_{3j}, & \text{if } j \geq 2 \text{ and } L_{3j+2} + F_{3j-2} \leq n < L_{3j+2} + F_{3j-1}; \\ 2F_{3j+2}, & \text{if } L_{3j+2} + F_{3j-1} \leq n < L_{3j+3}. \end{cases}$$

Proof. To prove this, we mimic the proof of Theorem 20 in [18]. The resulting automaton is displayed below in Figure 3.

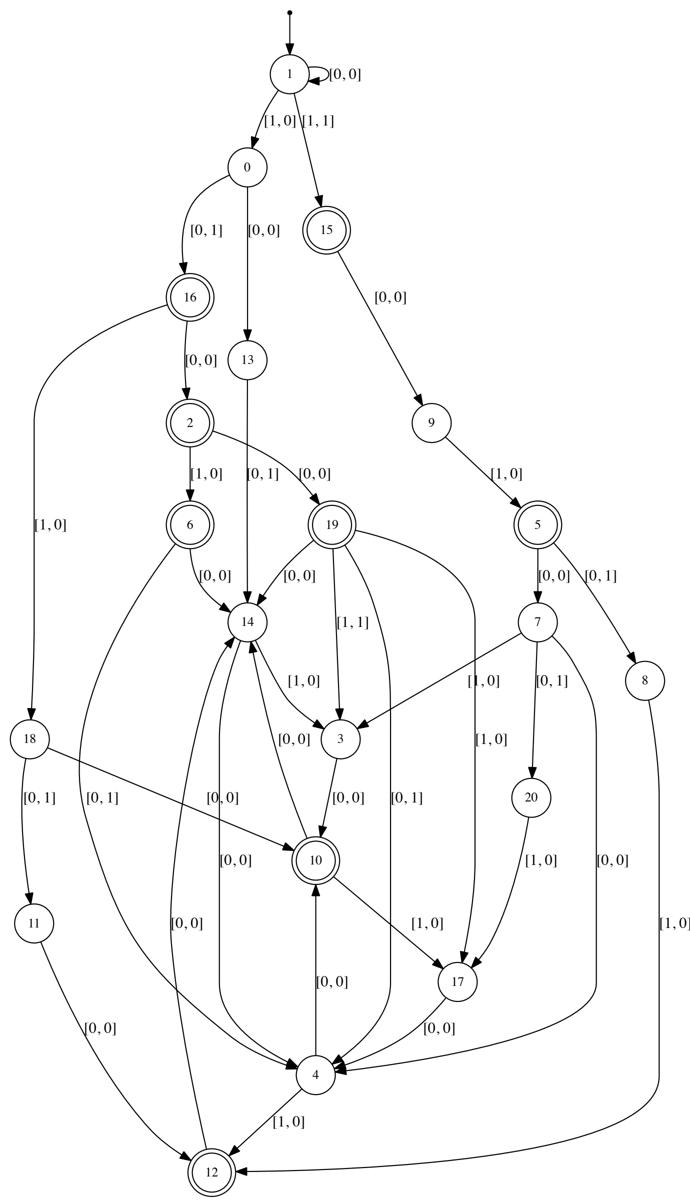

Figure 3: Automaton accepting least periods of prefixes of length n

□

Corollary 5. *The critical exponent of the Rote-Fibonacci word is* $2 + \alpha$.

Proof. An examination of the cases in Theorem 4 show that the words of maximum exponent are those corresponding to $n = L_{3j+2} - 1$, $p = F_{3j+1}$. As $j \to \infty$, the quantity n/p approaches $2 + \alpha$ from below. □

Theorem 6. *All squares in the Rote-Fibonacci word are of order F_{3n+1} for $n \geq 0$, and each such order occurs.*

Proof. We use the predicate

$$(n \geq 1) \;\wedge\; \exists i\; \forall j < n\; (\mathbf{r}[i+j] = \mathbf{r}[i+j+n]).$$

The resulting automaton is depicted in Figure 4. The accepted words correspond to F_{3n+1} for $n \geq 0$.

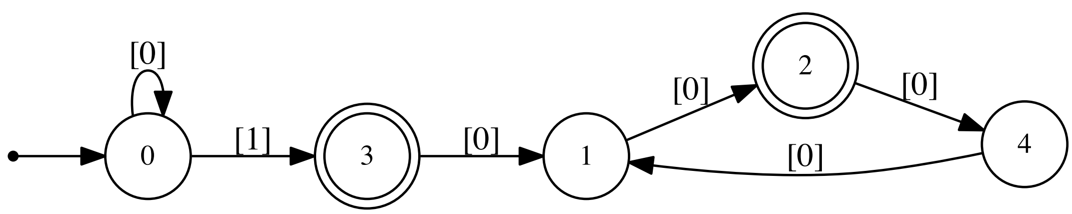

Figure 4: Automaton accepting orders of squares in the Rote-Fibonacci word

□

We now turn to problems considering prefixes of the Rote-Fibonacci word $\mathbf{r}$.

Theorem 7. *A length-n prefix of the Rote-Fibonacci word $\mathbf{r}$ is an antipalindrome iff $n = F_{3i+1} - 3$ for some $i \geq 1$.*

Proof. We use our decision method on the predicate

$$\forall j < n\; \mathbf{r}[j] \neq \mathbf{r}[n-1-j].$$

The result is depicted in Figure 5. The only accepted expansions are given by the regular expression $\epsilon + 1(010101)^*0(010 + 101000)$, which corresponds to $F_{3j+1} - 3$. We use the predicate

$$(n \geq 1) \;\wedge\; \exists i\; \forall j < n\; \mathbf{r}[i+j] = \mathbf{r}[i+j+n]).$$

The resulting automaton is depicted in Figure 5. The accepted words correspond to F_{3n+1} for $n \geq 0$.

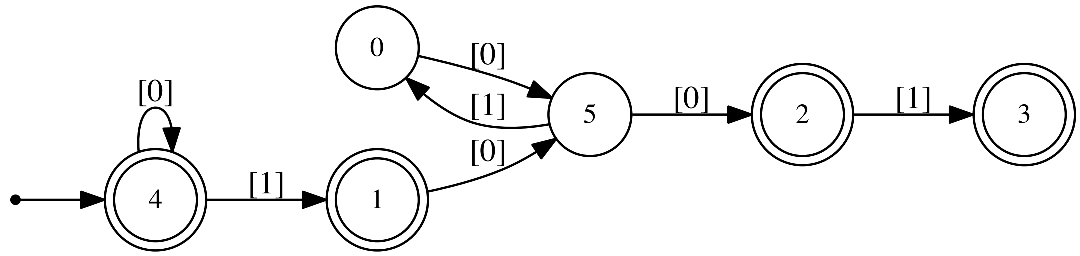

Figure 5: Automaton accepting lengths of antipalindrome prefixes in the Rote-Fibonacci word

□

Theorem 8. *A length-n prefix of the Rote-Fibonacci word is an antisquare if and only if $n = 2F_{3k+2}$ for some $k \geq 1$.*

Proof. The predicate for having an antisquare prefix of length n is

$$\forall k < n \ \mathbf{r}[i+k] \neq \mathbf{r}[i+k+n].$$

When we run this we get the automaton depicted in Figure 6.

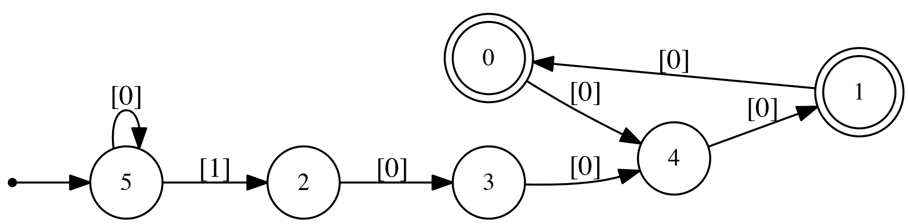

Figure 6: Automaton accepting orders of antisquares that are prefixes of $\mathbf{f}$

□

Theorem 9. *The Rote-Fibonacci word has subword complexity $2n$.*

Proof. Follows from Corollary 2 together with [23, Thm. 3]. □

Theorem 10. *The Rote-Fibonacci word is mirror invariant. That is, if z is a factor of* $\mathbf{r}$ *then so is z^R.*

Proof. We use the predicate

$$\forall i \ \exists j \ \forall t < n \ \mathbf{r}[i+t] = \mathbf{r}[j+n-1-t].$$

The resulting automaton accepts all n, so the conclusion follows. The largest intermediate automaton has 2300 states and the calculation took about 6 seconds on a laptop. □

Corollary 11. *The Rote-Fibonacci word avoids the pattern xx^Rx^R.*

Proof. Suppose xx^Rx^R occurs in $\mathbf{r}$. Then by Theorem 10 we know that $(xx^Rx^R)^R = xxx^R$ occurs in $\mathbf{f}$. But this is impossible, by Theorem 3. □

As it turns out, the Rote-Fibonacci word has (essentially) appeared before in several places. For example, in a 2009 preprint of Monnerot-Dumaine [17], 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 the fixed point $g^\omega(a)$ and applying the coding $\gamma(a) = S$, $\gamma(b) = R$, $\gamma(c) = S$, and $\gamma(d) = L$. 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 $\mathbf{r}$ [3, 4, 5]. For example, in their paper "Fibonacci snowflakes" [3] they defined a certain sequence q_i which has the following relationship to g: let $\xi(a) = \xi(b) = L$, $\xi(c) = \xi(d) = R$. Then

$$R\xi(g^n(a)) = q_{3n+2}L.$$

2.1 Conjectures and open problems about the Rote-Fibonacci word

In this section we collect some conjectures we have not yet been able to prove. We have made some progress and hope to completely resolve them in the future.

Conjecture 12. Every infinite binary word avoiding the pattern xxx^R has critical exponent $\geq 2 + \alpha$.

Conjecture 13. Let z be a finite nonempty primitive binary word. If z^ω avoids xxx^R, then $|z| = 2F_{3n+2}$ for some integer $n \geq 0$. Furthermore, z is a conjugate of the prefix $\mathbf{r}[0..2F_{3n+2} - 1]$, for some $n \geq 0$. Furthermore, for $n \geq 1$ we have that z is a conjugate of $y\overline{y}$, where $y = \tau(h^{3n}(a))$.

We can make some partial progress on this conjecture, as follows:

Theorem 14. *Let $k \geq 1$ and define $n = 2F_{3k+2}$. Let $z = \mathbf{r}[0..n-1]$. Then z^ω contains no occurrence of the pattern xxx^R.*

Proof. We have already seen this for $k = 0$, so assume $k \geq 1$.

Suppose that z^ω does indeed contain an occurrence of xxx^R for some $|x| = \ell > 0$. We consider each possibility for ℓ and eliminate them in turn.

Case I: $\ell \geq n$.

There are two subcases:

Case Ia: $n \not| \ell$: In this case, by considering the first n symbols of each of the two occurrences of x in xxx^R in z^ω, we see that there are two different cyclic shifts of z that are identical. This can only occur if $\mathbf{r}[0..n-1]$ is a power, and we know from Theorem 6 and Corollary 5 that this implies that $n = 2F_{3k+1}$ or $n = 3F_{3k+1}$ for some $k \geq 0$. But $2F_{3k+1} \neq 2F_{3k'+2}$ and $3F_{3k+1} \neq 2F_{3k'+2}$ provided $k, k' > 0$, so this case cannot occur.

Case Ib: $n \mid \ell$: Then x is a conjugate of z^e, where $e = \ell/n$. By a well-known result, a conjugate of a power is a power of a conjugate; hence there exists a conjugate y of z such that $x = y^e$. Then $x^R = y^e$, so x and hence y is a palindrome. We can now create a predicate that says that some conjugate of $\mathbf{r}[0..n-1]$ is a palindrome:

$$\exists i < n \ \wedge \ (\forall j < n \ \text{cmp}(i+j, n+i-1-j))$$

where

$$\begin{aligned} \text{cmp}(k,k') := (((k < n) \ \wedge \ (k' < n)) \implies (\mathbf{r}[k] = \mathbf{r}[k'])) \wedge \\ (((k < n) \ \wedge \ (k' \geq n)) \implies (\mathbf{r}[k] = \mathbf{r}[k'-n])) \wedge \\ (((k \geq n) \ \wedge \ (k' < n)) \implies (\mathbf{r}[k-n] = \mathbf{r}[k'])) \wedge \\ (((k \geq n) \ \wedge \ (k' \geq n)) \implies (\mathbf{r}[k-n] = \mathbf{r}[k'-n])). \end{aligned}$$

When we do this we discover the only n with Fibonacci representation of the form 10010^i accepted are those with $i \equiv 0, 2 \pmod 3$, which means that $2F_{3k+2}$ is not among them. So this case cannot occur.

Case II: $\ell < n$.

There are now four subcases to consider, depending on the number of copies of z needed to "cover" our occurrence of xxx^R. In Case II.j, for $1 \leq j \leq 4$, we consider j copies of z and the possible positions of xxx^R inside that copy.

Because of the complicated nature of comparing one copy of x to itself in the case that one or both overlaps a boundary between different copies of z, it would be very helpful to be able to encode statements like $\mathbf{r}[k \bmod n] = \mathbf{r}[\ell \bmod n]$ in our logical language. Unfortunately, we cannot do this if n is arbitrary. So instead, we use a trick: assuming that the indices k, k' satisfy $0 \leq k, k' < 2n$, we can use the $\text{cmp}(k, k')$ predicate introduced above to simulate the assertion $\mathbf{r}[k \bmod n] = \mathbf{r}[k' \bmod n]$. Of course, for this to work we must ensure that $0 \leq k, k' < 2n$ holds.

The cases are described in Figure 7. We assume that that $|x| = \ell$ and xxx^R begins at position i of z^ω. We have the inequalities $i < n$ and $\ell < n$ which apply to each case. Our predicates are designed to compare the first copy of x to the second copy of x, and the first copy of x to the x^R.

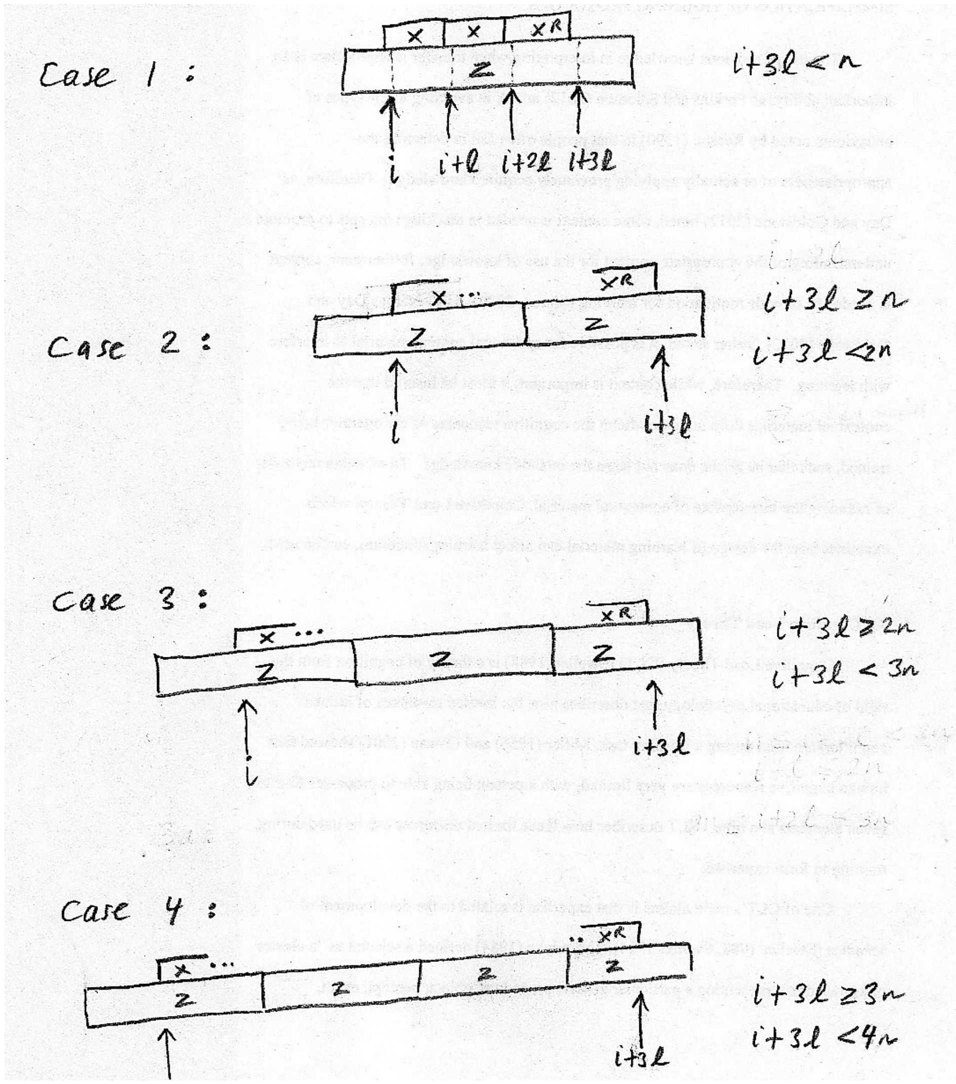

Figure 7: Cases of the argument

Case 1: If xxx^R lies entirely within one copy of z, it also lies in $\mathbf{r}$, which we have already seen cannot happen, in Theorem 3. This case therefore cannot occur.

Case 2: We use the predicate

$$\exists i\, \exists \ell\, (i{+}3\ell \geq n) \wedge (i{+}3\ell < 2n) \wedge (\forall j < \ell \text{ cmp}(i{+}j, i{+}\ell{+}j)) \wedge (\forall k < \ell \text{ cmp}(i{+}k, i{+}3\ell{-}1{-}k))$$

to assert that there is a repetition of the form xxx^R.

Case 3: We use the predicate

$$\exists i\, \exists \ell\, (i{+}3\ell \geq 2n) \wedge (i{+}3\ell < 3n) \wedge (\forall j < \ell \text{ cmp}(i{+}j, i{+}\ell{+}j{-}n)) \wedge (\forall k < \ell \text{ cmp}(i{+}k, i{+}3\ell{-}1{-}k{-}n))).$$

Case 4: We use the predicate

$$\exists i\, \exists \ell\, (i{+}3\ell \geq 3n) \wedge (i{+}3\ell < 4n) \wedge (\forall j < \ell \text{ cmp}(i{+}j, i{+}\ell{+}j{-}n)) \wedge (\forall k < \ell \text{ cmp}(i{+}k, i{+}3\ell{-}1{-}k{-}2n))).$$

When we checked each of the cases 2 through 4 with our program, we discovered that $n = 2F_{3k+2}$ is never accepted. Actually, for cases (2)–(4) we had to employ one additional trick, because the computation for the predicates as stated required more space than was available on our machine. Here is the additional trick: instead of attempting to run the predicate for all n, we ran it only for n whose Fibonacci representation was of the form 10010^*. This significantly restricted the size of the automata we created and allowed the computation to terminate. In fact, we propagated this condition throughout the predicate.

We therefore eliminated all possibilities for the occurrence of xxx^R in z^ω and so it follows that no xxx^R occurs in z^ω. □

Very recently, Currie and Rampersad [10] have solved the problem of enumerating the number of binary words of length n avoiding the pattern xxx^R. In a very surprising result, they proved that it is $n^{\Theta(\log n)}$. This is the first pattern-avoidance problem having this form of growth rate.

Open Problem 15. How many binary words of length n avoid both the pattern xxx^R and $(2+\alpha)$-powers?

Consider finite words of the form xxx^R having no proper factor of the form www^R.

Conjecture 16. For $n = F_{3k+1}$ there are 4 such words of length n. For $n = F_{3k+1} \pm F_{3k-2}$ there are 2 such words. Otherwise there are none.

For $k \geq 3$ the 4 words of length $n = F_{3k+1}$ are given by $\mathbf{r}[p_i..p_i + n - 1]$, $i = 1, 2, 3, 4$, where

$$\begin{aligned}(p_1)_F &= 1000(010)^{k-3}001\\ (p_2)_F &= 10(010)^{k-2}001\\ (p_3)_F &= 1001000(010)^{k-3}001\\ (p_4)_F &= 1010(010)^{k-2}001\end{aligned}$$

For $k \geq 3$ the 2 words of length $n = F_{3k+1} - F_{3k-2}$ are given by $\mathbf{r}[q_i..q_i + n - 1]$, $i = 1, 2$, where

$$\begin{aligned}(q_1)_F &= 10(010)^{k-3}001\\ (q_2)_F &= 10000(010)^{k-3}001\end{aligned}$$

For $k \geq 3$ the 2 words of length $n = F_{3k+1} + F_{3k-2}$ are given by $\mathbf{r}[s_i..s_i + n - 1]$, $i = 1, 2$, where

$$\begin{aligned}(s_1)_F &= 10(010)^{k-3}001\\ (s_2)_F &= 1000(01)^{k-2}001\end{aligned}$$

3 Other sequences

In this section we briefly apply our method to some other Fibonacci-automatic sequences, obtaining several new results.

Consider a Fibonacci analogue of the Thue-Morse sequence

$$\mathbf{v} = (v_n)_{n \geq 0} = 0111010010001100010111000101 \cdots$$

where v_n is the sum of the bits, taken modulo 2, of the Fibonacci representation of n. This sequence was introduced in [24, Example 2, pp. 12–13].

We recall that an *overlap* is a word of the form $axaxa$ where x may be empty; its order is defined to be $|ax|$. Similarly, a *super-overlap* is a word of the form $abxabxab$; an example of a super-overlap in English is the word `tingalingaling` with the first letter removed.

Theorem 17. *The only squares in* $\mathbf{v}$ *are of order* 4 *and* F_n *for* $n \geq 2$*, and a square of each such order occurs. The only cubes in* $\mathbf{v}$ *are the strings* 000 *and* 111*. The only overlaps in* $\mathbf{v}$ *are of order* F_{2n} *for* $n \geq 1$*, and an overlap of each such order occurs. There are no super-overlaps in* $\mathbf{v}$*.*

Proof. As before. We omit the details. □

We might also like to show that $\mathbf{v}$ is recurrent. The obvious predicate for this property holding for all words of length n is

$$\forall i \; \exists j \; ((j > i) \wedge (\forall t \; ((t < n) \implies (\mathbf{v}[i+t] = \mathbf{v}[j+t])))).$$

Unfortunately, when we attempt to run this with our prover, we get an intermediate NFA of 1159 states that we cannot determinize within the available space.

Instead, we rewrite the predicate, setting $k := j - i$ and $u := i + t$. This gives

$$\forall i \; \exists j \; (j > i) \wedge \forall k \; \forall u \; ((k \geq 1) \wedge (i = j + k) \wedge (u \geq i) \wedge (u < n + i)) \implies \mathbf{v}[u] = \mathbf{v}[u+k].$$

When we run this we discover that $\mathbf{v}$ is indeed recurrent. Here the computation takes a nontrivial 814007 ms, and the largest intermediate automaton has 625176 states. This proves

Theorem 18. *The word* $\mathbf{v}$ *is recurrent.*

Another quantity of interest for the Thue-Morse-Fibonacci word $\mathbf{v}$ is its subword complexity $\rho_{\mathbf{v}}(n)$. It is not hard to see that it is linear. To obtain a deeper understanding of it, let us compute the first difference sequence $d(n) = \rho_{\mathbf{v}}(n+1) - \rho_{\mathbf{v}}(n)$. It is easy to see that $d(n)$ is the number of words w of length n with the property that both $w0$ and $w1$ appear in $\mathbf{v}$. The natural way to count this is to count those i such that $t := \mathbf{v}[i..i+n-1]$ is the first appearance of that factor in $\mathbf{v}$, and there exists a factor $\mathbf{v}[k..k+n]$ of length $n+1$ whose length-n-prefix equals t and whose last letter $\mathbf{v}[k+n]$ differs from $\mathbf{v}[i+n]$.

$$(\forall j < i \; \exists t < n \; \mathbf{v}[i+t] \neq \mathbf{v}[j+t]) \; \wedge \; (\exists k \; (\forall u < n \; \mathbf{v}[i+u] = \mathbf{v}[k+u]) \wedge \mathbf{v}[i+n] \neq \mathbf{v}[k+n]).$$

Unfortunately the same blowup appears as in the recurrence predicate, so once again we need to substitute, resulting in the predicate

$$(\forall j < i\ \exists k \geq 1\ \exists v\ (i = j + k) \wedge (v \geq j) \wedge (v < n + j) \wedge \mathbf{v}[u] \neq \mathbf{v}[u + k]) \wedge$$
$$(\exists l > i\ \mathbf{v}[i + n] \neq \mathbf{v}[l + n]) \wedge$$
$$(\forall k'\ \forall u'\ (k' \geq 1) \wedge (l = i + k') \wedge (u' \geq i) \wedge (v' < n + i) \implies \mathbf{v}[k' + u'] = \mathbf{v}[u']).$$

From the resulting automaton we can, as in [11], obtain a linear representation of rank 46. We can now consider all vectors of the form $u\{M_0, M_1\}^*$. There are only finitely many and we can construct an automaton out of them computing $d(n)$.

Theorem 19. *The first difference sequence* $(d(n))_{n\geq 0}$ *of the subword complexity of* $\mathbf{v}$ *is Fibonacci-automatic, and is accepted by the following machine.*

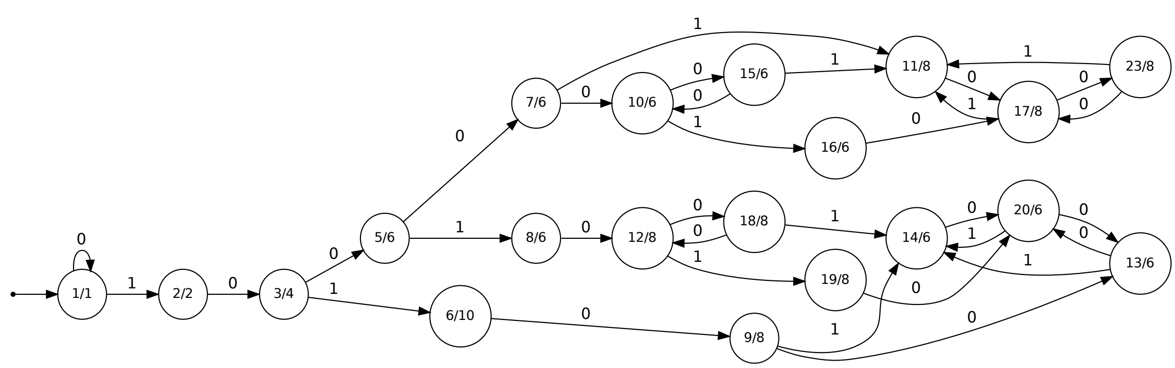

Figure 8: Automaton computing $d(n)$

4 Combining two representations and avoidability

In this section we show how our decidability method can be used to handle an avoidability question where two different representations arise.

Let x be a finite word over the alphabet $\mathbb{N}^* = \{1, 2, 3 \ldots\}$. We say that x is an *additive square* if $x = x_1 x_2$ with $|x_1| = |x_n|$ and $\sum x_1 = \sum x_2$. For example, with the usual association of `a` $= 1$, `b` $= 2$, and so forth, up to `z` $= 26$, we have that the English word `baseball` is an additive square, as `base` and `ball` both sum to 27.

An infinite word $\mathbf{x}$ over $\mathbb{N}^*$ is said to *avoid additive squares* if no factor is an additive square. It is currently unknown, and a relatively famous open problem, whether there exists an infinite word over a *finite* subset of $\mathbb{N}^*$ that avoids additive squares [6, 20, 14], although it is known that additive cubes can be avoided over an alphabet of size 4 [8]. (Recently this was improved to alphabet size 3; see [22].)

However, it is easy to avoid additive squares over an *infinite* subset of $\mathbb{N}^*$; for example, any sequence that grows sufficiently quickly will have the desired property. Hence it is reasonable to ask about the *lexicographically least* sequence over $\mathbb{N}^*$ that avoids additive squares. Such a sequence begins

$$121312142125213121341217 2\cdots,$$

but we do not currently know if this sequence is unbounded.

Here we consider the following variation on this problem. Instead of considering arbitrary sequences, we start with a sequence $\mathbf{b} = b_0 b_1 b_2 \cdots$ over $\mathbb{N}^+$ and from it construct the sequence $S(\mathbf{b}) = a_1 a_2 a_3 \cdots$ defined by

$$\mathbf{a}[i] = \mathbf{b}[\nu_2(i)]$$

for $i \geq 1$, where $\nu_2(i)$ is the exponent of the largest power of 2 dividing i. (Note that $\mathbf{a}$ and $\mathbf{b}$ are indexed differently.) For example, if $\mathbf{b} = 123\cdots$, then $\mathbf{a} = 1213121412131215\cdots$, the so-called "ruler sequence". It is known that this sequence is squarefree and is, in fact, the lexicographically least sequence over $\mathbb{N}^*$ avoiding squares [13].

We then ask: what is the lexicographically least sequence avoiding additive squares that is of the form $S(\mathbf{b})$? The following theorem gives the answer.

Theorem 20. *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): Consider two consecutive blocks of the same size say $a_{i+1} \cdots a_{i+n}$ and $a_{i+n+1} \cdots a_{i+2n}$. Our goal is to compare the sums $\sum_{i<j\leq i+n} a_j$ and $\sum_{i+n<j\leq i+2n} a_j$.

First we prove

Lemma 21. *Let $m, j \geq 0$ and $n \geq 1$ be integers. Let $A(m, n, j)$ denote the number of occurrences of j in $\nu_2(m+1), \ldots, \nu_2(m+n)$. Then for all $m, m' \geq 0$ we have $|A(m', n, j) - A(m, n, j)| \leq 1$.*

Proof. We start by observing that the number of positive integers $\leq n$ that are divisible by t is exactly $\lfloor n/t \rfloor$. It follows that the number $B(n, j)$ of positive integers $\leq n$ that are divisible by 2^j but not by 2^{j+1} is

$$B(n, j) = \lfloor \frac{n}{2^j} \rfloor - \lfloor \frac{n}{2^{j+1}} \rfloor. \tag{1}$$

Now from the well-known identity

$$\lfloor x \rfloor + \lfloor x + \frac{1}{2} \rfloor = \lfloor 2x \rfloor,$$

valid for all real numbers x, substitute $x = n/2^{j+1}$ to get

$$\lfloor \frac{n}{2^{j+1}} \rfloor + \lfloor \frac{n}{2^{j+1}} + \frac{1}{2} \rfloor = \lfloor \frac{n}{2^j} \rfloor,$$

which, combined with (1), shows that

$$B(n,j) = \lfloor \frac{n}{2^{j+1}} + \frac{1}{2} \rfloor.$$

Hence

$$\frac{n}{2^{j+1}} - \frac{1}{2} \le B(n,j) < \frac{n}{2^{j+1}} + \frac{1}{2}. \tag{2}$$

Now the number of occurrences of j in $\nu_2(m+1), \ldots, \nu_2(m+n)$ is $A(m,n,j) = B(m+n,j) - B(m,j)$. From (2) we get

$$\frac{n}{2^{j+1}} - 1 < A(m,n,j) < \frac{n}{2^{j+1}} + 1 \tag{3}$$

for all $m \ge 0$. Since $A(m,n,j)$ is an integer, the inequality (3) implies that $|A(m',n,j) - A(m,n,j)| \le 1$ for all m, m'. □

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 our decision procedure, 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 $w \in \{\texttt{-1},\texttt{0},\texttt{1}\}^*$, 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 $\{z \in (\Sigma_2^2 \times \{\texttt{-1},\texttt{0},\texttt{1}\})^* : \forall j (\pi_3(z)[j] = A(\langle \pi_1(z) \rangle_2 + \langle \pi_2(z) \rangle_2, \langle \pi_2(z) \rangle_2, j) - A(\langle \pi_1(z) \rangle_2, \langle \pi_2(z) \rangle_2, j))\}$. The second task can be done by a 5-state (incomplete) DFA $M_{\text{1uFisZero}}$ that accepts the language $\{w \in \{\texttt{-1},\texttt{0},\texttt{1}\}^* : \langle w \rangle_{uF} = 0\}$.

We applied a modified decision procedure to the predicate $n \ge 1 \wedge \exists w (\texttt{add22F}(i,n,w) \wedge \texttt{1uFisZero}(w))$ 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 $n \ge 1$ 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 decision procedure, 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.

$$\texttt{powOf2}(r) \land \forall t((t \geq 1 \land t < r \land \texttt{canonFib}(t)) \to \exists i \exists n(n \geq 1 \land i + 2n < 2r \land ((i < r \land r \leq i + n \land \forall w(\texttt{add22F}(i, n, w) \to \forall x(\texttt{bitAdd}(t, w, x) \to \texttt{2uFisZero}(x)))) \lor (i + n < r \land r \leq i + 2n \land \forall w(\texttt{add22F}(i, n, w) \to \forall x(\texttt{bitSub}(t, w, x) \to \texttt{2uFisZero}(x))))))).$$

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

We applied a modified decision procedure 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 Acknowledgments

We thank Narad Rampersad and Michel Rigo for useful suggestions.

References

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

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

[3] 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** (2011), 4778–4786.

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

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

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

[7] L. Carlitz. Fibonacci representations. *Fibonacci Quart.* **6** (1968), 193–220.

[8] J. Cassaigne, J. Currie, L. Schaeffer, and J. Shallit. Avoiding three consecutive blocks of the same size and same sum. *J. Assoc. Comput. Mach.* **61**(2) (2014), Paper 10.

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

[10] J. D. Currie and N. Rampersad. Growth rate of binary words avoiding xxx^R. Preprint, `http://arxiv.org/abs/1502.07014`, 2015.

[11] C. F. Du, H. Mousavi, L. Schaeffer, and J. Shallit. Decision algorithms for Fibonacci-automatic words, III: Enumeration and abelian properties. Submitted, 2015.

[12] A. S. Fraenkel. Systems of numeration. *Amer. Math. Monthly* **92** (1985), 105–114.

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

[14] 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** (2000), #A7. `http://www.integers-ejcnt.org/vol0.html`.

[15] C. Holton and L. Q. Zamboni. Directed graphs and substitutions. *Theory Comput. Systems* **34** (2001), 545–564.

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

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

[18] H. Mousavi, L. Schaeffer, and J. Shallit. Decision algorithms for Fibonacci-automatic words, I: Basic results. Submitted, 2015.

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

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

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

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

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

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

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