Note on a Fibonacci Parity Sequence

Jeffrey Shallit
School of Computer Science
University of Waterloo
Waterloo, ON N2L 3G1
Canada
shallit@uwaterloo.ca

March 22, 2022

Abstract

Let $ftm = 0111010010001 \cdots$ be the analogue of the Thue-Morse sequence in Fibonacci representation. In this note we show how, using the Walnut theorem-prover, to obtain a measure of its complexity, previously studied by Jamet, Popoli, and Stoll. We strengthen one of their theorems and disprove one of their conjectures.

1 Introduction

Recall that the Fibonacci numbers $(F_n)$ are defined by

$$F_0 = 0, \quad F_1 = 1, \quad \text{and} \quad F_n = F_{n-1} + F_{n-2} \quad \text{for} \quad n \geq 2.$$ 

In Fibonacci representation (aka Zeckendorf representation), we express a natural number $n$ uniquely as a sum of non-adjacent Fibonacci numbers: $n = \sum_{2 \leq i \leq t} e_i F_i$, where $e_i \in \{0, 1\}$ and $e_i e_{i+1} = 0$ for $2 \leq i < t$. See, for example, [5, 12].

The sum of the Fibonacci bits is $s_F(n) = \sum_{2 \leq i \leq t} e_i$, and the so-called Fibonacci-Thue-Morse sequence $ftm$ is then defined as $ftm[n] = s_F(n) \mod 2$. Here are the first few terms of this binary sequence, which is sequence A095076 in the On-Line Encyclopedia of Integer Sequences (OEIS) [11]:

| $n$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----|---|---|---|---|---|---|---|---|---|---|----|----|----|----|
| $ftm[n]$ | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

This sequence was previously studied by Ferrand [2] and the author [8], to name just two appearances.

Recently $ftm$ appeared in a paper of Jamet, Popoli, and Stoll [3], as follows. A factor (contiguous block) $x$ in a (finite or infinite) binary sequence $s$ is said to be special if both
$x_0$ and $x_1$ appear in $s$. Now let $s$ be an infinite sequence, and define $f_s(n)$ be the length of the longest special factor in a length-$n$ prefix of $s$.

For example, here are the first few terms of $f_s(n)$ when $s = \text{ftm}$:

| $n$ | $2$ | $3$ | $4$ | $5$ | $6$ | $7$ | $8$ | $9$ | $10$ | $11$ | $12$ | $13$ | $14$ | $15$ | $16$ | $17$ | $18$ | $19$ |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| $f_{\text{ftm}}(n)$ | $0$ | $0$ | $0$ | $2$ | $2$ | $2$ | $2$ | $2$ | $2$ | $4$ | $4$ | $4$ | $4$ | $4$ | $4$ | $4$ | $5$ |

Jamet et al. studied a certain measure of the complexity of a binary sequence $s$, called maximum order complexity $M_s(n)$, and observed that by a result of Jansen [4], we have the relationship $M_s(n) = f_s(n) + 1$.

Letting $\alpha = (1 + \sqrt{5})/2$ be the golden ratio, their Theorem 1.1 is the following:

**Theorem 1.** There exists $N_0$ such that for all $n > N_0$ we have $M_{\text{ftm}}(n) \geq \frac{n}{\alpha + \alpha^2} + 1$.

They also gave the following conjecture, Conjecture 3.1 in their paper:

**Conjecture 2.** $M_{\text{ftm}}(n) \sim \frac{n}{1+\alpha^2}$.

In this note we first obtain an exact formula for $f_{\text{ftm}}(n)$ (our Theorem 3), using the Walnut theorem-prover. (For more about Walnut, see [6] and [10].) Then, using this result, we disprove Conjecture 2. Finally, we show how to reprove Theorem 1 and also determine the explicit value of $N_0$ in that theorem.

## 2 First-order formulas

The idea behind our approach is that we can express the assertion that $m = f_s(n)$ as a formula in first-order logic. This is done as follows:

- $\text{factoreq}(i, j, n)$ asserts that $s[i..i + n - 1] = s[j..j + n - 1]$;
- $\text{spec}(i, m, n)$ asserts that there is a $j$ such that $\text{factoreq}(i, j, m)$ but $s[i + m] \neq s[j + m]$, and $\max(i + m, j + m) < n$;
- $\text{speclen}(m, n)$ asserts that there is a special factor of length $m$ in $s[0..n-1]$, the length-$n$ prefix of $s$;
- $\text{maxspec}(m, n)$ asserts that the length of the longest special factor occurring in a prefix of length $n$ is $m$.

We can translate these into first-order formulas as follows:

$$\text{factoreq}(i, j, n) = \forall u, v \ (i + v = j + u \land u \geq i \land u < i + n) \implies s[u] = s[v]$$

$$\text{spec}(i, m, n) = \exists j \ \text{factoreq}(i, j, m) \land s[i + m] \neq s[j + m] \land i + m < n \land j + m < n$$

$$\text{speclen}(m, n) = \exists i \ \text{spec}(i, m, n)$$

$$\text{maxspec}(m, n) = \text{speclen}(m, n) \land \forall j \ (j > m) \implies \neg \text{speclen}(j, n).$$
It now follows from known results [1] that if \( s \) is a (generalized) automatic sequence, then there is an effective algorithm for computing an automaton that accepts the representation of those pairs of natural numbers \( m, n \) for which \( m \) is the length of the longest special factor in \( s[0..n - 1] \). In particular, the sequence \( \text{ftm} \) is Fibonacci-automatic, so we can find this automaton for the pairs \( (m, n) \) in Fibonacci representation [7]. This is done in the next section.

3 Translation to Walnut

We now translate the first-order formulas to \textit{Walnut}. The first step is to obtain an automaton for the sequence \( \text{ftm} \) itself. We can either get this directly from the morphism and coding in the paper of Jamet et al., or compute it with \textit{Walnut}. For this latter approach we use the \textit{Walnut} commands

\begin{verbatim}
reg odd1 msd_fib "0*(10*10*)*10*":
def zecksum "?msd_fib (n>=0) & $odd1(n)":
combine FTM zecksum:
\end{verbatim}

The first line defines a regular expression for a binary string having an odd number of 1’s; the second asserts that an integer \( n \) has an odd number of 1’s in its Fibonacci representation; and the third line turns this automaton into an automaton with output (DFAO).

Next, we translate the first-order formulas from Section 2 into \textit{Walnut}:

\begin{verbatim}
def factoreq "?msd_fib Au,v (i+v=j+u & u>=i & u<i+n) => FTM[u]=FTM[v]":
def spec "?msd_fib (i+m<n) & Ej (j+m<n) & $factoreq(i,j,m) & FTM[i+m] != FTM[j+m]":
def speclen "?msd_fib Ei $spec(i,m,n)":
def maxspec "?msd_fib $speclen(m,n) & Aj (j>m) => ~$speclen(j,n)":
\end{verbatim}

By comparing these with the formulas above, we see that they are more-or-less a direct translation.

When we run all these commands through \textit{Walnut}, the last one produces an automaton of 17 states depicted in Figure 1. This shows that the map \( n \to \text{ftm}(n) \) is “Fibonacci-synchronized”; see, for example, [9].

We can use this automaton to prove the following “exact” formula for \( \text{ftm}(n) \). Define the Lucas numbers \( (L_n) \) by \( L_0 = 2, \ L_1 = 1, \) and \( L_n = L_{n-1} + L_{n-2} \) for \( n \geq 2 \). Then we have the following result:

\textbf{Theorem 3.} \textit{Suppose \( i \geq 4 \) and \( n \geq 8 \). If \( L_i < n \leq L_{i+1} \), then}

\[ \text{ftm}(n) = \begin{cases} 
F_{i-1}, & \text{for } i \text{ even;} \\
F_{i-1} + 1, & \text{for } i \text{ odd.}
\end{cases} \]
Figure 1: Automaton accepting Fibonacci representation of \((\text{maxspec}(n), n)\).

Proof. We can use \texttt{Walnut} to verify the statement. In order to determine which case applies, we need to be able to compute \(F_{i-1}, L_i,\) and \(L_{i+1}\) simultaneously. In Fibonacci representation, this can be done with the following four simple observations:

- For \(i \geq 2,\) the number \(F_i\) has Fibonacci representation \(10^{i-2}0\cdots0.\)
- If \(F_i\) and \(F_j\) are two Fibonacci numbers with \(F_i < F_j \leq 2F_i,\) then \(j = i + 1.\)
- For \(i \geq 1\) we have \(L_i = 2F_{i-1} + F_i.\)
- For \(i \geq 1\) we have \(L_{i+1} = F_{i-1} + 3F_i.\)

We can assert that \((x, y) = (F_{2i-1}, F_{2i})\) for some \(i \geq 2\) by asserting that \((x)_{F}\) ends in an odd number of 0’s, \((y)_{F}\) ends in an even number of 0’s, and \(x < y\) and \(2x \geq y.\) Similarly, we can assert that \((x, y) = (F_{2i}, F_{2i+1})\) for some \(i \geq 1\) by asserting that \((x)_{F}\) ends in an even number of 0’s, \((y)_{F}\) ends in an odd number of 0’s, and \(x < y\) and \(2x \geq y.\) This can be done with the following \texttt{Walnut} commands:

\begin{verbatim}
reg isevenfib msd_fib "0*1(00)*":
reg isoddfib msd_fib "0*10(00)*":
def fiboddeven "$msd_fib isoddfib(x) & isevenfib(y) & x<y & (2*x)>=y":
def fibevenodd "$msd_fib isevenfib(x) & isoddfib(y) & x<y & (2*x)>=y":
\end{verbatim}
We can now check the claim of the theorem as follows:

def check_i_even "?msd_fib An,x,y ($fiboddeven(x,y) & n>=8 & 2*x+y<n & n<x+3*y) => $maxspec(x,n)"

def check_i_odd "?msd_fib An,x,y ($fibevenodd(x,y) & n>=8 & 2*x+y<n & n<x+3*y) => $maxspec(x+1,n)"

and both return TRUE. Here \(x\) plays the role of \(F_{i-1}\) and \(y\) plays the role of \(F_i\).

Because \(f_{ftm}(n)\) is constant on longer and longer intervals of \(n\), it becomes easy to compute \(\lim \inf_{n \to \infty} f_{ftm}(n)/n\) and \(\lim \sup_{n \to \infty} f_{ftm}(n)/n\).

In particular, if \(L_i < n \leq L_{i+1}\) for \(i\) even, \(i \geq 4\), Theorem 3 implies that the quotient \(f_{ftm}(n)/n\) is minimized at \(n = L_{i+1}\), where it takes the value \(F_{i-1}/L_{i+1}\), and maximized at \(n = L_i + 1\), where it takes the value \(F_{i-1}/(L_i + 1)\).

Similarly, if \(L_i < n \leq L_{i+1}\) for \(i\) odd, then the quotient \(f_{ftm}(n)/n\) is minimized at \(n = L_{i+1}\), where it takes the value \((F_{i-1} + 1)/L_{i+1}\), and is maximized at \(n = L_i + 1\), where it takes the value \((F_{i-1} + 1)/(L_i + 1)\).

From the above remarks, together with the Binet forms for \(L_i\) and \(F_i\), which are

\[
F_i = \frac{(\alpha^i - \beta^i)}{\sqrt{5}} \\
L_i = \alpha^i + \beta^i,
\]

where \(\alpha = (1 + \sqrt{5})/2\) and \(\beta = (1 - \sqrt{5})/2\), we obtain the following result.

**Corollary 4.** We have

\[
\lim \inf_{n \to \infty} f_{ftm}(n)/n = 1/(\sqrt{5} \alpha^2) = 1/(\alpha + \alpha^3) \approx 0.17082039
\]

and

\[
\lim \sup_{n \to \infty} f_{ftm}(n)/n = 1/(\sqrt{5} \alpha) = 1/(1 + \alpha^2) \approx 0.276393202.
\]

These results refute Conjecture 3.1 of [3]. The behavior of \(f_{ftm}(n)/n\) is depicted in Figure 2 below.

We can also recover Theorem 1 (that is, Theorem 1.1 of Jamet et al. [3]), with an explicit constant:

**Theorem 5.** We have \(f_{ftm}(n) \geq \frac{n}{\alpha + \alpha^3}\) for all \(n \geq 5\).

**Proof.** We can verify the inequality for \(n \geq 8\) using Theorem 3. Suppose \(L_i < n \leq L_{i+1}\).

For \(i \geq 4\) and even we have \(\beta^{i-2} + \beta^{i-4} \geq 0\), while \(\beta^{i+1} < 0\). Hence

\[
\beta^{i-2} + \beta^{i-4} \geq \sqrt{5} \cdot \beta^{i+1}. \tag{1}
\]

Now observe that \(\sqrt{5} = \alpha + 1/\alpha\), so adding \(\alpha^{i+1}(\alpha + 1/\alpha)\) to both sides of Eq. (1) gives

\[
(\alpha^{i-1} - \beta^{i-1})(\alpha + \alpha^3) = \alpha^{i+2} + \alpha^i + \beta^{i-2} + \beta^{i-4} \geq \sqrt{5}(\alpha^{i+1} + \beta^{i+1}).
\]
Figure 2: The quotient $f_{\text{fm}}(n)/n$ for $2 \leq n \leq 1000.$

By rearranging we get

$$\frac{f_{\text{fm}}(n)}{n} \geq \frac{F_{i-1}}{L_{i+1}} = \frac{(\alpha^{i-1} - \beta^{i-1})/\sqrt{5}}{\alpha^{i+1} + \beta^{i+1}} \geq \frac{1}{\alpha + \alpha^3}.$$  

On the other hand, if $i \geq 5$ is odd, we have $\sqrt{5} - \beta^{i-1} > 1$, and $\beta^{i+1} < 1$, so

$$(\sqrt{5} - \beta^{i-1})(\alpha + \alpha^3) \geq (\alpha + \alpha^3) \geq \sqrt{5}\beta^{i+1}.$$  

as before, adding $\alpha^{i+1}(\alpha + 1/\alpha)$ to both sides and simplifying gives

$$(\alpha^{i-1} - \beta^{i-1} + \sqrt{5})(\alpha + \alpha^3) \geq \sqrt{5}(\alpha^{i+1} + \beta^{i+1}),$$

so

$$\frac{f_{\text{fm}}(n)}{n} \geq \frac{F_{i-1} + 1}{L_{i+1}} = \frac{(\alpha^{i-1} - \beta^{i-1})/\sqrt{5} + 1}{\alpha^{i+1} + \beta^{i+1}} \geq \frac{1}{\alpha + \alpha^3},$$

as desired.

Finally, for $5 \leq n \leq 7$, we can easily check the inequality, while for $n \leq 4$ it fails.

References

[1] É. Charlier, N. Rampersad, and J. Shallit. Enumeration and decidable properties of automatic sequences. Internat. J. Found. Comp. Sci. 23 (2012), 1035–1066.
[2] E. Ferrand. An analogue of the Thue-Morse sequence. *Electron. J. Combin.* 14 (2007), R30 (electronic), https://doi.org/10.37236/948

[3] D. Jamet, P. Popoli, and T. Stoll. Maximum order complexity of the sum of digits function in Zeckendorf base and polynomial subsequences. *Cryptography and Communications* 13 (2021), 791–814.

[4] C. J. A. Jansen. *Investigations on nonlinear streamcipher systems: construction and evaluation methods.* PhD thesis, Technische Universiteit Delft, Netherlands, 1989.

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

[6] H. Mousavi. Automatic theorem proving in *Walnut*. Arxiv preprint arXiv:1603.06017 [cs.FL], available at http://arxiv.org/abs/1603.06017, 2016.

[7] H. Mousavi, L. Schaeffer, and J. Shallit. Decision algorithms for Fibonacci-automatic words, I: Basic results. *RAIRO Inform. Théor. App.* 50 (2016), 39–66.

[8] J. Shallit. Subword complexity of the Fibonacci-Thue-Morse sequence: the proof of Dekking’s conjecture. *Indag. Math.* 32 (2021), 729–735.

[9] J. Shallit. Synchronized sequences. In T. Lecroq and S. Puzynina, editors, *WORDS 2021*, Vol. 12847 of *Lecture Notes in Computer Science*, pp. 1–19. Springer-Verlag, 2021.

[10] J. Shallit. *The Logical Approach to Automatic Sequences: Exploring Combinatorics on Words with Walnut.* Cambridge University Press, 2022. In press.

[11] N. J. A. Sloane et al. The on-line encyclopedia of integer sequences. Electronic resource, available at https://oeis.org, 2022.

[12] E. Zeckendorf. Repr´esentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas. *Bull. Soc. Roy. Li`ege* 41 (1972), 179–182.