Automatic complexity of Fibonacci and Tribonacci words

Bjørn Kjos-Hanssen*

October 15, 2020

Abstract

For a complexity function $C$, the lower and upper $C$-complexity rates of an infinite word $x$ are

$$\underline{C}(x) = \liminf_{n \to \infty} \frac{C(x | n)}{n}, \quad \overline{C}(x) = \limsup_{n \to \infty} \frac{C(x | n)}{n},$$

respectively. Here $x | n$ is the prefix of $x$ of length $n$. We consider the case $C = A_N$, the nondeterministic automatic complexity. If these rates are strictly between 0 and $1/2$, we call them intermediate. Our main result is that words having intermediate $A_N$-rates exist, viz. the infinite Fibonacci and Tribonacci words.

1 Introduction

The automatic complexity of Shallit and Wang [10] is the minimal number of states of an automaton accepting only a given word among its equal-length peers. This paper continues a line of investigation into the automatic complexity of particular words of interest such as

- maximal length sequences for linear feedback shift registers [7],
- overlap-free and almost square-free words [3], and
- random words [8].

All these examples have high complexity: to be precise, they have maximal automatic complexity rate (Definition [5]). On the other hand, a periodic word has low complexity and a rate of 0. In the present paper we give the first examples of infinite words with intermediate automatic complexity rate: the infinite Fibonacci and Tribonacci words.

*This work was partially supported by grants from the Simons Foundation (#315188 and #704836 to Bjørn Kjos-Hanssen) and Decision Research Corporation (Automatic Complexity of Fibonacci arrays, University of Hawai’i Foundation Account #129-4779-4).
Automatic complexity is an automata-based and length-conditional analogue of Sipser’s CD complexity [11] which is in turn a computable analogue of the noncomputable Kolmogorov complexity. The nondeterministic case was taken up by Hyde and Kjos-Hanssen [3]. We recall our basic notions. Let $|x|$ denote the length of a word $x$.

**Definition 1 (10).** The nondeterministic automatic complexity $A_N(x)$ of a word $x$ is the minimal number of states of a nondeterministic finite automaton $M$ (without $\varepsilon$-transitions) such that $M$ accepts $x$ and moreover there is only one accepting path in $M$ of length $|x|$. We let $A^-$ denote the deterministic but non-total automatic complexity, defined as follows: automata are required to be deterministic, but the transition functions need not be total: there does not need to be a transition for every symbol at every state.

Shallit and Wang’s original automatic complexity $A(x)$ does have the totality requirement.

### 1.1 Fibonacci and Tribonacci words

**Definition 2 (k-bonacci numbers).** For $k \geq 2$ and $n \geq 0$, the $n$th $k$-bonacci number $w_n = w_n^{(k)}$ is defined by $w_n = 0$ if $n \leq k - 1$, $w_k = 1$, and $w_n = \sum_{i=n-k}^{n-1} w_i$ for $n \geq k + 1$.

In particular, the Fibonacci numbers are the 2-bonacci numbers.

Let $\Sigma_k = \{a_0, \ldots, a_{k-1}\}$ be an alphabet of cardinality $k \geq 0$. We shall denote specific symbols in the alphabet as $0 = a_0$, $1 = a_1$ and so on.

**Definition 3 (k-bonacci words).** We define the $k$-morphism $\varphi_k : \Sigma_k \to \Sigma_k^*$ by

\[
\varphi_k(a_i) = a_0 a_{i+1}, \quad 0 \leq i \leq k - 2,
\]
\[
\varphi_k(a_{k-1}) = a_0.
\]

We also let $\varphi$ act on words of length greater than 1, by the morphism property

\[
\varphi(uv) = \varphi(u)\varphi(v).
\]

Let $\varepsilon$ be the empty word. The $k$-bonacci word $W_n = W_n^{(k)}$ is defined by

\[
W_n = \varepsilon, \quad 0 \leq n \leq k - 2,
\]
\[
W_{k-1} = a_{k-1},
\]
\[
W_n = \varphi_k(W_{n-1}), \quad n \geq k.
\]

**Lemma 4.** The length of the $n$th $k$-bonacci word is equal to the $n$th $k$-bonacci number: $|W_n^{(k)}| = w_n^{(k)}$.

**Lemma 5.** For all $k \geq 2$ and $n \geq 2k - 1$, if $W_n$ is the $n$th $k$-bonacci word then

\[
W_n = W_{n-1} \ldots W_{n-k}.
\]
We then say Fibonacci for 2-bonacci and Tribonacci for 3-bonacci. Thus the finite Tribonacci words $T_n$ are defined by

\[
\begin{align*}
T_0 &= \varepsilon \\
T_1 &= \varepsilon \\
T_2 &= 2 \\
T_3 &= 0 \\
T_4 &= 01 \\
T_n &= T_{n-1}T_{n-2}T_{n-3}, \quad n \geq 5,
\end{align*}
\]

and the finite Fibonacci words $F_n$ by

\[
\begin{align*}
F_0 &= \varepsilon \\
F_1 &= 1 \\
F_2 &= 0 \\
F_n &= F_{n-1}F_{n-2}, \quad n \geq 3.
\end{align*}
\]

**Definition 6.** The infinite $k$-bonacci word $f^{(k)}$ is the fixed point $\varphi_k(0)$ of the morphism $\varphi_k$.

Thus, the infinite Tribonacci word

\[ T = f^{(3)} = 0102010010201 \ldots \]

is a fixed point of the morphism $0 \to 01, 1 \to 02, 2 \to 0$. It is a variant of the Fibonacci word obtained from the morphism $0 \to 01, 1 \to 0$.

## 2 Lower bounds from critical exponents

A word is square-free if it does not contain any subword of the form $xx$ (denoted $x^2$), $|x| > 0$. Shallit and Wang showed that a square-free word has high automatic complexity, and we shall show that integrality of powers is not crucial: that is, we shall use critical exponents.

**Definition 7.** Let $w$ be an infinite word over the alphabet $\Sigma$, and let $x$ be a finite word over $\Sigma$. Let $\alpha > 0$ be a rational number. The word $x$ is said to occur in $w$ with exponent $\alpha$ if there is a subword $y$ of $w$ with $y = x^ay$ where $x_0$ is a prefix of $x$, $a$ is the integer part of $\alpha$, and $|y| = \alpha|x|$. We say that $y$ is an $\alpha$-power. The word $w$ is $\alpha$-power-free if it contains no subwords which are $\alpha$-powers.

The critical exponent for $w$ is the supremum of the $\alpha$ for which $w$ has $\alpha$-powers, or equivalently the infimum of the $\alpha$ for which $w$ is $\alpha$-power-free.

**Definition 8.** Fix a finite alphabet $\Sigma$. For an infinite word $w \in \Sigma^\infty$, let $w \upharpoonright n$ denote the prefix of $w$ of length $n$. Let $C : \Sigma^* \to \mathbb{N}$. The lower $C$-complexity rate of $w$ is

\[ C(w) = \liminf_{n \to \infty} \frac{C(w \upharpoonright n)}{n}. \]
The upper $C$-complexity rate of $w$ is

$$C(w) = \limsup_{n \to \infty} \frac{C(w \upharpoonright n)}{n}.$$ 

If these are equal we may speak simply of the $C$-complexity rate. In the case where $C = A_N$ we may speak of automatic complexity rate.

**Definition 9** (Fibonacci constant). Let $\phi = \frac{1 + \sqrt{5}}{2}$, the positive root of $\phi^2 = \phi + 1$.

**Definition 10** (Tribonacci constant). Let

$$\xi = \frac{1}{3} \left( 1 + \sqrt[3]{19 - 3\sqrt{33}} + \sqrt[3]{19 + 3\sqrt{33}} \right).$$

**Theorem 11** ([9]). The critical exponent of the infinite Fibonacci word $f^{(2)}$ is $2 + \phi \approx 3.6$.

Tan and Wen [12] studied critical exponents, calling them free indices.

**Theorem 12** (Tan and Wen [12, Theorem 4.5]). The critical exponent of the infinite Tribonacci word $f^{(3)}$ is $3 + \frac{1}{2}(\theta^2 + \theta^4)$, where

$$\theta = \frac{1}{3} \left( -1 - \frac{2}{(17 + 3\sqrt{33})^{1/3}} + (17 + 3\sqrt{33})^{1/3} \right) \approx 0.54369012692076\ldots$$

is the unique real root of the equation $\theta^3 + \theta^2 + \theta = 1$.

**Lemma 13.** The critical exponent of $f^{(3)}$ is the real zero

$$2 + \frac{1}{6} \sqrt[3]{54 - 6\sqrt{33}} + \frac{\sqrt[3]{9 + \sqrt{33}}}{6^{2/3}}$$

of the polynomial $2x^3 - 12x^2 + 22x - 13$.

Lemma 13 follows from Theorem 12 by computer software (Wolfram Alpha).

**Definition 14.** Let $x$ be a word of length $n$, $x = x_1, \ldots, x_n$. Two occurrences of words $a$ (starting at position $i$) and $b$ (starting at position $j$) in a word $x$ are disjoint if $x = uavbw$ where $u, v, w$ are words and $|u| = i - 1$, $|uv| = j$. If in addition $|v| > 0$ then we say that these occurrences of $a$ and $b$ are strongly disjoint.

**Theorem 15** ([8]). If the critical exponent of a word $x$ is at most $\gamma \geq 2$ then there is an $m \geq 0$ and a set of $m$ many strongly disjoint at-least-square powers in $x$ with $A_N(x) \geq \frac{n + 1 - m}{\gamma}$.
**Theorem 16.** If the critical exponent of a word $x$ is at most $\gamma \geq 2$ then $A_N(x) \geq \frac{n + 1 - \sqrt{2n}}{\gamma}$.

**Proof.** By uniqueness of path the $m$ many powers in Theorem 15 must have distinct base lengths. Thus the base lengths add up to at least $\sum_{k=1}^{m} k = m(m + 1)/2$, which implies $m(m + 1)/2 \leq n$. Consequently $m \leq \sqrt{2n}$ and

$$A_N(x) \geq \frac{n + 1 - m}{\gamma} \geq \frac{n + 1 - \sqrt{2n}}{\gamma}.$$ 

**Theorem 17.** The $A_N$-complexity rate of the infinite Fibonacci word $f^{(2)}$ is at least

$$\frac{2}{5 + \sqrt{5}} = 0.27639\ldots$$

The $A_N$-rate of the infinite Tribonacci number $f^{(3)}$ is at least

$$0.31333478\ldots$$

the real root of $-2 + 12x - 22x^2 + 13x^3$.

**Proof.** These two facts now follow by applying Theorem 16 with Theorems 11 and 12 respectively.

Karhumäki [4] showed that the Fibonacci words contain no 4th power and this implies (the deterministic version is in Shallit and Wang 2001 [10] Theorem 9) $A_N(x) \geq (n + 1)/4$.

**Theorem 18.** The $A_N$-complexity rate of the infinite $k$-bonacci word $f^{(k)}$ is at least $1/4$ for any $k \geq 2$.

**Proof.** Glen [2] showed that the $k$-bonacci word has no fourth power for any $k \geq 2$. Thus, the critical exponent is at most 4 and by Theorem 16 we are done.

## 3 Upper bounds from factorizations

There are many factorization result possible for $k$-bonacci words. Even their definitions like $F_n = F_{n-1}F_{n-2}$ are factorizations. We shall prove some such results that help us obtain upper bounds on automatic complexity: Theorem 20 and 21. In the following, for convenience we renumber by defining $\tilde{T}_n = T_{n+3}$.

**Definition 19.** For $n \geq 0$ and $0 \leq k \leq n$, we let $\langle k \rangle_n = \tilde{T}_{n-k}$. We also write $\|k\|_n = |\langle k \rangle_n|$, the length of $\langle k \rangle_n$. When $n$ is understood from context we write $\langle k \rangle = \langle k \rangle_n$ and $\|k\| = \|k\|_n$.

**Theorem 20.** For large enough $n$, $\tilde{T}_{n-2}^2 \prod_{k=6}^{\lfloor n/3 \rfloor} \tilde{T}_{n-k}$ is a prefix of $\tilde{T}_n$. 


Proof. Note that the equation $\tilde{T}_n = \tilde{T}_{n-1} \tilde{T}_{n-2} \tilde{T}_{n-3}$ holds for $n \geq 4$. Thus we can write

$$
\langle n - 2 \rangle = \langle n - 1 \rangle \langle n \rangle \langle n + 1 \rangle
$$

(1)

but we cannot expand $\langle n - 1 \rangle$. The idea now is to use a loop of length 13 followed by one of length 6:

$$
\tilde{T}_n = \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle = (\langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) \langle 2 \rangle \langle 3 \rangle = (\langle 2 \rangle \langle 3 \rangle \langle 4 \rangle \langle 5 \rangle) \langle 3 \rangle = (\langle 2 \rangle \langle 3 \rangle \langle 4 \rangle \langle 5 \rangle \langle 6 \rangle) \langle 5 \rangle = (\langle 2 \rangle \langle 3 \rangle \langle 4 \rangle \langle 5 \rangle \langle 6 \rangle) \langle 5 \rangle \langle 6 \rangle \langle 4 \rangle \langle 5 \rangle \langle 3 \rangle
$$

Thus for $m = 4$ we have

$$
(0) = \langle 2 \rangle^2 \left( \prod_{k=6}^{13} \langle k \rangle \right) \langle 3m - 1 \rangle = \left( \prod_{M=m}^{3m+1} \langle M \rangle \langle 3M - 2 \rangle \langle 3M - 1 \rangle \right) \langle 3 \rangle.
$$

(2)

Here we use the notation $\prod_{M=m}^{3m+1} a_M = a_M a_{M-1} \ldots a_0$. To prove (2) for $m \geq 4$ by induction we expand:

$$
\langle 3m - 1 \rangle = \langle 3m + 2 \rangle \langle 3m + 3 \rangle \langle 3m + 4 \rangle \langle 3m + 5 \rangle \langle 3m + 6 \rangle \langle 3m + 7 \rangle \langle 3m + 8 \rangle \langle 3m + 9 \rangle \langle 3m + 10 \rangle \langle 3m + 11 \rangle 
$$

This is valid as long as $3m + 1 \leq n - 2$ by (1), i.e., as long as $3(m + 1) \leq n$.\end{proof}

**Theorem 21.** For any $m \geq 4$, $\prod_{k=6}^{3m+1} \langle k \rangle$ is a prefix of $\langle 2 \rangle$.

**Proof.** We have

$$
\langle 2 \rangle = \langle 3 \rangle \langle 4 \rangle \langle 5 \rangle = \langle 4 \rangle \langle 5 \rangle \langle 6 \rangle \langle 4 \rangle \langle 5 \rangle = \langle 6 \rangle \langle 7 \rangle \langle 8 \rangle \langle 6 \rangle \langle 7 \rangle \langle 5 \rangle \langle 6 \rangle \langle 4 \rangle \langle 5 \rangle
$$

By substitution into the proof it will suffice to show that $\langle 4 \rangle \langle 5 \rangle \ldots$ is a prefix of $\langle 0 \rangle$.

To show that, we first show that $\langle 3 \rangle \langle 4 \rangle \ldots$ is a prefix of $\langle 0 \rangle$:

$$
\langle 0 \rangle = \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle = \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle \langle 2 \rangle \langle 3 \rangle = \langle 3 \rangle \langle 4 \rangle \langle 5 \rangle \langle 3 \rangle \langle 4 \rangle \langle 2 \rangle \langle 3 \rangle
$$

Now by substitution, $\langle 6 \rangle \langle 7 \rangle \langle 8 \rangle \ldots$ is a prefix of $\langle 3 \rangle$ and we are done. (Incidentally this can now be used to show that $\langle 2 \rangle \langle 3 \rangle \ldots$ is a prefix of $\langle 0 \rangle$.) Finally, let us show that $\langle 4 \rangle \langle 5 \rangle \ldots$ is a prefix of $\langle 0 \rangle$:

$$
\langle 0 \rangle = \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle = \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle \langle 2 \rangle \langle 3 \rangle = \langle 3 \rangle \langle 4 \rangle \langle 5 \rangle \langle 3 \rangle \langle 4 \rangle \langle 2 \rangle \langle 3 \rangle
$$

It does not hold for $n = 1$: we have $\tilde{T}_2 = (01)\langle 0 \rangle \langle 2 \rangle = \tilde{T}_1 \tilde{T}_0 \tilde{T}_1$, but $\tilde{T}_1 = 01 \neq \langle 0 \rangle \langle 2 \rangle = \tilde{T}_0 \tilde{T}_1 \tilde{T}_0 \tilde{T}_2$.\end{proof}

6
By substitution and Theorem 20, \( \langle 7 \rangle \langle 8 \rangle \langle 9 \rangle \ldots \) is a prefix of \( \langle 4 \rangle \), and we are done.

We can also expand the tail of \( \tilde{T}_n \):
\[
\langle 0 \rangle = \langle 1 \rangle \langle 3 \rangle \langle 4 \rangle \langle 5 \rangle \langle 4 \rangle \langle 5 \rangle \langle 6 \rangle
\]
which ends in \((\langle 4 \rangle \langle 5 \rangle)^*\) with * a non-integer exponent.

**Theorem 22.** For \( n \geq 6 \),
\[
A^{-}(\tilde{T}_n) \leq \|1\| - 3\left\lfloor \frac{n}{3} \right\rfloor + 1 + \sum_{k=6}^{3\left\lfloor \frac{n}{3} \right\rfloor + 1} \|k\|.
\]

For example, when \( n = 6 \) this is \( 24 - (1 + 1) = 22 \), which fits with an 8-day long computation we performed.

**Proof.** Using now a \( \|2\| \)-cycle followed by a path and then a \( \|4\| + \|5\| \)-cycle, we can subtract the extra prefix from Theorem 20 and use only
\[
\|2\| + \|4\| + \|5\| + \left(\|0\| - 2\|2\| - 2(\|4\| + \|5\|) - \|6\| - \sum_{k=6}^{3\left\lfloor \frac{n}{3} \right\rfloor + 1} \|k\|\right)
\]
\[
= \|1\| - \sum_{k=6}^{3\left\lfloor \frac{n}{3} \right\rfloor + 1} \|k\|
\]
(since \( \|0\| = \|6\| + 2\|5\| + 3\|4\| + 2\|3\| + \|2\| \) states. Uniqueness is guaranteed by Theorem 24.)

**Lemma 23.** The Tribonacci constant (Definition 17) satisfies
\[
\xi = \lim_{n \to \infty} \frac{\|1\|_n}{\|0\|_n} = 1.83929 \ldots
\]
and is the unique real root of \( \xi^3 = \xi^2 + \xi + 1 \).

In particular \( \xi = 1/\theta \) with \( \theta \) as in Theorem 12.

**Theorem 24.** For large enough \( n \), the equation
\[
x\|2\| + y(\|4\| + \|5\|) = 2(\|2\| + \|4\| + \|5\|)
\]
for nonnegative integers \( x, y \), has the unique solution \( x = y = 2 \).

**Proof.** Suppose \((x, y)\) is a solution, not equal to \((2, 2)\). Then we have \( x = 0, \ x = 1, \ y = 0, \) or \( y = 1 \). If \( x = 0 \) then
\[
\Xi \ni y = 2 \left( \frac{\|2\| + \|4\| + \|5\|}{\|4\| + \|5\|} \right) = 2 \left( \frac{\|6\|}{\|4\| + \|5\|} + 3 \right)
\]
which is impossible as soon as $\|6\| > 0$ since $2\|6\| < \|4\| + \|5\|$. Similarly, if $x = 1$ then
\[
Z \ni y = \frac{2(\|2\| + \|4\| + \|5\|) - 2\|2\|}{\|4\| + \|5\|} = \frac{\|2\|}{\|4\| + \|5\|} + 2 = \frac{\|6\|}{\|4\| + \|5\|} + 4.
\]
If $y = 0$ then
\[
Z \ni x = \frac{2\|\|2\| + \|4\| + \|5\|\|}{\|2\|} = 2\left(1 + \frac{\|2\| - \|3\|}{\|2\|}\right) = 2\left(2 - \|3\|\right)
\]
which is impossible as soon as $\|3\| > 0$ since
\[
2 > \frac{2\|\|3\|\|}{\|2\|} = \frac{\|3\| + \|4\| + \|5\| + \|6\|}{\|3\| + \|4\| + \|5\|} > 1.
\]
Finally, if $y = 1$ then
\[
Z \ni x = \frac{2(\|2\| + \|4\| + \|5\|) - \|4\| - \|5\|\|}{\|2\|} = 2 + \frac{\|4\| + \|5\|\|}{\|2\|}
\]
which is impossible as soon as $0 < \|4\| + \|5\| < \|2\|$, i.e., $\|4\| > 0$.

**Theorem 25.** The $A^-$-complexity rate of the Tribonacci word satisfies
\[
\limsup_{n \to \infty} \frac{A^-(T_n)}{|T_n|} \leq \frac{1}{6}(-8 + (586 - 102\sqrt{33})^{1/3} + (2(293 + 51\sqrt{33}))^{1/3})
\]
\[
\approx 0.4870856 \ldots
\]

**Proof.** We calculate
\[
\limsup_{n \to \infty} \frac{A^-(\hat{T}_n)}{|\hat{T}_n|} \leq \lim_{n \to \infty} \frac{1\| - \sum_{k=6}^{2(\lfloor n/3 \rfloor + 1)} \|k\|}{\|0\|} = \frac{1}{\xi} \sum_{k=6}^{\infty} \frac{1}{\xi^k}
\]
\[
= \frac{1}{\xi} - \frac{1}{3\xi^2 + 3\xi + 2}
\]
\[
= \frac{1}{6}(-8 + (586 - 102\sqrt{33})^{1/3} + (2(293 + 51\sqrt{33}))^{1/3}).
\]

**Definition 26.** $A^\text{lower}_N(x)$ is the minimal $q$ such that for all sequences of strongly disjoint powers $x_1^{\alpha_1}, \ldots, x_m^{\alpha_m}$, in $x$, with the uniqueness condition that $\sum \alpha_i |x_i| = \sum \alpha_i y_i \implies y_i = |x_i|$, all $i$, we have
\[
2q \geq n + 1 - m - \sum_{i=1}^{m} (\alpha_i - 2)|x_i|.
\]
Figure 1: An automaton witnessing the automatic complexity of the Fibonacci word of length 55.

The definition of $A_{N}^{\text{lower}}(x)$ may seem very technical. The point is that

- $A_{N}^{\text{lower}}$ appears to be faster to compute than $A_{N}$,
- by [8, Theorem 19], we have $A_{N}^{\text{lower}}(x) \leq A_{N}(x)$ for all words $x$, and
- $A_{N}^{\text{lower}}(x)$ is a better lower bound than that obtained simply by the critical exponent considerations in Theorem 15.

We have implemented $A_{N}$ and $A_{N}^{\text{lower}}$ ([5]) with results in Table 1 and Table 2 (A lookup tool is also available for automatic complexity [6].) Note that

$$A^{-}(T_{n}) = A^{-}(\tilde{T}_{n-3}) \leq \|1\| - \sum_{k=6}^{3(n-3)/3+1} \|k\|$$

$$= \|1\| - \sum_{k=6}^{3(n/3)-2} \|k\|$$

$$= \|1\| - \sum_{k=6}^{7} \|k\| = t_{8} - t_{3} - t_{2} = 24 - 1 - 1 = 22, \quad (n = 9),$$

$$= \|1\| - \sum_{k=6}^{7} \|k\| = t_{9} - t_{4} - t_{3} = 44 - 2 - 1 = 41, \quad (n = 10).$$

Remark 27. Witnessing automata for $A_{N}$ are conveniently generated by state sequences. A state sequence is the sequence of states visited by the unique accepting path of length $n+1$ (having potentially up to $n$ edges and $n+1$ states). A week-long computer search for the length 55 Fibonacci word

$$0100101001001010010010010010100100100100101001001001010010010100101010010010010010100100101001001001010$$
revealed the witnessing state sequence, where states are given numerical labels, using letters A, B, C, . . . for the numbers 10, 11, 12, . . .:

0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 8, 9, A, B, C, D, E, F, G, H, I, J, K, L, 9, A, B, C, D, E, F, G, H, I, J, K, L, 9, A, B, C, D, E, F, G, H.

We illustrate the automaton induced by this state sequence in Figure 1. Generalizing this example gives

\[ \overline{A_N}(f^{(2)}) \leq \frac{1}{\varphi^2} + \frac{1}{\varphi^3} = 0.41640786499 \]

**Lemma 28.** For \( n \geq 6 \), the equation

\[ x f_{n-2} + y f_n = 2(f_{n-2} + f_n) \]

for nonnegative integers \( x, y \), has the unique solution \( x = y = 2 \).

**Proof.** If \( x = 0 \) then

\[ y = 2 \frac{f_{n-2} + f_n}{f_n} = 2 + 2 \frac{f_{n-2}}{f_n} = 2 + \frac{2f_{n-2}}{f_{n-1} + f_{n-2}} \in (2, 3) \]

is not an integer as long as \( n \geq 4 \). If \( x = 1 \) then

\[ y = 2 \frac{(f_{n-2} + f_n) - f_{n-2}}{f_n} = 2 + \frac{f_{n-2}}{f_n} \in (2, 3) \]

is not an integer as long as \( n \geq 3 \). If \( y = 0 \) then

\[ x = 2 \frac{f_{n-2} + f_n}{f_{n-2}} = 2 \frac{3f_{n-2} + f_{n-3}}{f_{n-2}} = 6 + \frac{f_{n-3} + f_{n-4} + f_{n-5}}{f_{n-3} + f_{n-4}} \in (7, 8) \]

is not an integer as long as \( f_{n-5} > 0 \), i.e., \( n \geq 6 \). If \( y = 1 \) then

\[ x = 2 \frac{(f_{n-2} + f_n) - f_n}{f_{n-2}} = 2 + \frac{f_n}{f_{n-2}} = 2 + \frac{f_{n-2} + f_{n-2} + f_{n-3}}{f_{n-2}} \in (4, 5) \]

as long as \( f_{n-3} > 0 \). \( \Box \)

**Theorem 29.** The upper automatic complexity rate of the infinite Fibonacci word \( \overline{A_N}(f^{(2)}) \) is at most \( \frac{2}{\varphi^3} \).

**Proof.** We exploit the lengths of Fibonacci words.

\[
\begin{align*}
  f_n &= f_{n-1} + f_{n-2} = f_{n-2} + f_{n-3} + f_n \\
  &= f_{n-3} + f_{n-4} + (f_{n-3} + f_{n-3} + f_{n-4}) \\
  &= f_{n-4} + f_{n-5} + f_{n-4} + (f_{n-3} + f_{n-3} + f_{n-4}) \\
  &= f_{n-4} + (f_{n-5} + f_{n-5} + f_{n-6}) + (f_{n-3} + f_{n-3} + f_{n-4}) \quad \text{hardcode this} \\
  &= f_{n-5} + f_{n-4} + f_{n-6} + f_{n-4} + f_{n-3} + f_{n-4}. \\
\end{align*}
\]
Table 1: Lower and limiting upper bounds on $A_N(T_n)$ and $A^{-}(T_n)$.

| $n$ | $t_n$ | $T_n$ | $0.313t_n$ | $A_N^{\text{lower}}$ | $A^{-}$ | $0.487t_n$ |
|-----|-------|-------|-------------|-----------------------|----------|------------|
| 0   | 0     | 0     | 0           | 1                     | 1        | 0          |
| 1   | 0     | 0     | 0.3         | 1                     | 1        | 0.49       |
| 2   | 1     | 2     | .6          | 2                     | 2        | 0.97       |
| 3   | 1     | 0     | .3          | 1                     | 1        | 0.49       |
| 4   | 2     | 01    | 1.3         | 3                     | 3        | 1.95       |
| 5   | 4     | 0102  | 2.2         | 4                     | 4        | 3.4        |
| 6   | 7     | 0102010| 4.1        | 7                     | 7        | 6.3        |
| 7   | 13    | 0102010010201| 7.5    | 12                    | 13       | 11.7       |
| 8   | 24    | 0102...0100102| 13.8   | 21                    | 22       | 21.4       |
| 9   | 44    | 0102...0102010| 25.4   | 36                    | 41       | 39.4       |

This way we obtain for a Fibonacci word $x$ of length $f_n$ that

$$A_N(x) \leq f_{n-4} + f_{n-5} + f_{n-3} = 2f_{n-3}.$$ 

In the limit, $f_n/f_{n-1} \sim \varphi = \frac{1+\sqrt{5}}{2} \approx 1.6$, so $f_n/f_{n-3} \sim \varphi^3 = 4.236$, so

$$A_N(x) \leq \frac{2}{\varphi^3} f_n = 0.47 f_n.$$

The cycles give a unique path of length $f_n$ for large enough $n$, since

$$f_{n-5}x + f_{n-3}y = 2(f_{n-5} + f_{n-3})$$

has a unique solution $x = y = 2$ by Lemma 28.

4 Conclusion

More can be done on automatic complexity rates of $k$-bonacci words. For instance, we conjecture that $A_N(f^{(2)}) \leq 1/\varphi^2 = 0.382$. More precisely, we conjecture that this can be shown by analyzing the decomposition

$$
\langle 0 \rangle = (1)\langle 2 \rangle = (2)\langle 1 \rangle = (3)\langle 4 \rangle = (4)\langle 5 \rangle = (5)\langle 4 \rangle = (6)\langle 5 \rangle = (7)\langle 6 \rangle = \langle 8 \rangle
$$

However, in the present article we are content to have proven that the Fibonacci word has intermediate automatic complexity rate in Theorem 29.
| $n$ | $f_n$ | $F_n$ | $0.276f_n$ | $A^\text{lower}_N$ | $0.382f_n$ |
|-----|------|------|-----------|----------------|---------|
| 0   | 0    | 0    | 0         | 1              | 0       |
| 1   | 1    | 1    | 0.3       | 1              | 0.4     |
| 2   | 1    | 0    | 0.3       | 1              | 0.4     |
| 3   | 2    | 01   | 0.6       | 2              | 0.8     |
| 4   | 3    | 010  | 0.8       | 2              | 1.1     |
| 5   | 5    | 01001| 1.4       | 3              | 1.9     |
| 6   | 8    | 01001010 | 2.2   | 4     | 3.1     |
| 7   | 13   | 0100101001001 | 3.6   | 6     | 5.0     |
| 8   | 21   | 010010100101001001010 | 5.8   | 9     | 8.0     |
| 9   | 34   | 0100101001010010010101010100101010010100101010010100101001010...1001001...1010 | 9.4   | 14    | 13.0    |
| 10  | 55   | 0100...1010 | 15.2  | 21    | 21.0    |

Table 2: Lower and limiting upper bounds on $A_N(F_n)$.

References

[1] Steven R. Finch. *Mathematical constants*, volume 94 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2003.

[2] Amy Glen. Powers in a class of $\mathcal{R}$-strict standard episturmian words. *Theoret. Comput. Sci.*, 380(3):330–354, 2007.

[3] Kayleigh K. Hyde and Bjørn Kjos-Hanssen. Nondeterministic automatic complexity of overlap-free and almost square-free words. *Electron. J. Combin.*, 22(3), 2015. Paper 3.22, 18.

[4] Juhani Karhumäki. On cube-free $\omega$-words generated by binary morphisms. *Discrete Appl. Math.*, 5(3):279–297, 1983.

[5] Sun Young Kim, Bjørn Kjos-Hanssen, and Clyde James Felix. Code for “Automatic complexity of Fibonacci and Tribonacci words”. [https://github.com/bjoernkjoshanssen/tetranacci](https://github.com/bjoernkjoshanssen/tetranacci), 2019.

[6] Bjørn Kjos-Hanssen. Complexity lookup. [http://math.hawaii.edu/wordpress/bjoern/complexity-of-0110100110010110/](http://math.hawaii.edu/wordpress/bjoern/complexity-of-0110100110010110/)

[7] Bjørn Kjos-Hanssen. Automatic complexity of shift register sequences. *Discrete Math.*, 341(9):2409–2417, 2018.

[8] Bjørn Kjos-Hanssen. An incompressibility theorem for automatic complexity, 2019. arXiv preprint 1908.10843v1.

[9] F. Mignosi and G. Pirillo. Repetitions in the Fibonacci infinite word. *RAIRO Inform. Théor. Appl.*, 26(3):199–204, 1992.
[10] Jeffrey Shallit and Ming-Wei Wang. Automatic complexity of strings. *J. Autom. Lang. Comb.*, 6(4):537–554, 2001. 2nd Workshop on Descriptional Complexity of Automata, Grammars and Related Structures (London, ON, 2000).

[11] Michael Sipser. A complexity theoretic approach to randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC ’83, pages 330–335, New York, NY, USA, 1983. ACM.

[12] Bo Tan and Zhi-Ying Wen. Some properties of the Tribonacci sequence. *European J. Combin.*, 28(6):1703–1719, 2007.