University of Hawaiʻi at Mānoa

Master of Arts in Mathematics

Plan B Project Report

Nondeterministic automatic complexity of Fibonacci words

Author:
Calvin K. Bannister

Supervisor:
Bjorn Kjos-Hanssen

August 11, 2023

Abstract

Automatic complexity rates can be thought of as a measure of randomness of a computer program for some given automaton (machine). By creating a scale on the closed unit interval $I = [0, 1]$, that ranges from periodic to random, we can quantify how complex a program can be. If the rate is strictly between 0 and 1/2 then we call it intermediate. In end of his paper [2] Dr. Kjos-Hanssen conjectured that the non deterministic automatic complexity of an infinite Fibonacci word (infinite fibonacci string) has a particular upper bound. In this paper we examine infinite fibonacci words and measure their complexity provided some corresponding non-deterministic finite automaton.

1 Introduction

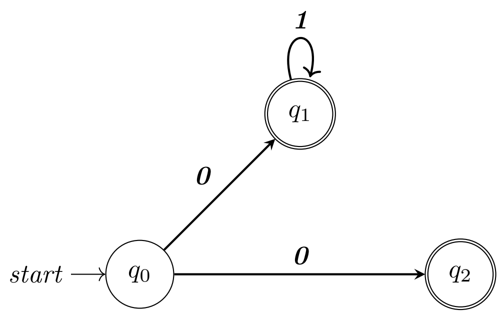
Shallit and Wang [3] introduced the theory and ideas behind automatic complexity, which has been studied in [1] and by Dr. Kjos-Hanssen and his students. Automatic complexity is studied in a branch of computer science called *automata theory* that deals with the study of abstract machines/automata along with computational problems that can be solved when using them. The idea of complexity (or randomness) deficiency comes from the study of Kolmogorov complexity. Kolmogorov complexity of an object, such as a piece of text, is the length of the shortest computer program (in a predetermined programming language) that produces the object as output. In this paper we prove the conjecture that was at the end of [2], and we show that for an infinite Fibonacci word the non-deterministic automatic complexity can be no greater than $1/\phi^2$. Let's begin by defining what is a non deterministic finite automaton.

Definition 1. *Let Σ be a finite set (an* alphabet*) and let Q be a finite set whose elements are states. A nondeterministic finite automaton (NFA) is a 5 - tuple $M = (Q, \Sigma, \delta, q_o, F)$. The transition function $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ maps each $(q, b) \in Q \times \Sigma$ to a subset of Q. Within Q we find the initial state $q_0 \in Q$ and the set of final states $F \subset Q$. The language accepted by M is*

$$L(M) = \{x \in \Sigma^* | \delta(q_0, x) \bigcap F \neq \emptyset\}$$

where $\Sigma^* := \bigcup_{n \in \mathbb{N}} \Sigma^n$ is the set of all words of length n for all natural numbers n.

Example 1. *Define the following sets $\Sigma = \{\mathbf{0,1}\}$, $Q = \{q_0, q_1, q_2\}$, $F = \{q_1, q_2\}$ and suppose we have the non-deterministic finite automaton below (For this rest of this paper we will refer to them as an NFA).*



For our transition function δ

$$\delta(q_0, \boldsymbol{0}) = \{q_1, q_2\}$$
$$\delta(q_0, \boldsymbol{1}) = \emptyset$$
$$\delta(q_1, \boldsymbol{1}) = \{q_1\}.$$

Looking at the first example, $\delta(q_0, \boldsymbol{0})$ *tells us that if we start at the initial state* q_0 *and take path* $\boldsymbol{0}$ *then the only set of states that contains such a path is the set* $\{q_1, q_2\}$*. The second set is empty since both edges have the* $\boldsymbol{0}$ *path coming from the start* q_0*. For the last set, the only path with such properties is if we take the loop at* q_1*.*

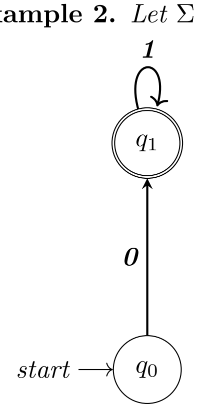
The language accepted by M *is the set*

$$L(M) = \{\boldsymbol{0}, \boldsymbol{01}^N : N \in \mathbb{Z}^+\}.$$

Definition 2. *Let* $L(M)$ *be the language recognized by the automaton* M*. Let* x *be a sequence of finite length* n*. The (unique-acceptance) nondeterministic automatic complexity* $A_N(w) = A_{N_u}(w)$ *of a word* w *is the minimum number of states of an NFA* M *such that* M *accepts* w *and the number of walks along which* M *accepts words of length* $|w|$ *is 1 i.e.*

the exact-acceptance nondeterministic automatic complexity $A_N(w) = A_{N_e}(w)$ *of a word* w *is the minimum number of states of an NFA* M *such that* M *accepts* w *and* $L(M) \bigcap \Sigma^{|w|} = \{w\}$*.*

Example 2. *Let* $\Sigma = \{\boldsymbol{0,1}\}$ *and* $x = 01111 \in \Sigma^*$



then $A_N(01111, \Sigma) \leq 2$.

Definition 3. *Fix a finite alphabet* Σ. *For an infinite word* $\boldsymbol{x} \in \Sigma^\infty$, *let* $\boldsymbol{x} \mid n$ *denote the prefix of* $\boldsymbol{x}$ *of length* n. *Let* $A_N : \Sigma^* \longrightarrow \mathbb{N}$. *Then the upper nondeterministic automatic complexity rate of* $\boldsymbol{x}$ *is*

$$\overline{A_N}(\boldsymbol{x}) = \limsup_{n\to\infty} \frac{A_N(\boldsymbol{x} \mid n)}{n}.$$

Nondeterministic automatic complexity has a rate of $0 < A_N(\boldsymbol{x}) < \frac{1}{2}$ *which is called intermediate. There is also lower nondeterministic automatic complexity, however for the purpose of this paper we will only focus on upper nondeterministic automatic complexity to help prove the theorem that is central to this paper.*

Definition 4. *(Fibonacci numbers)*

The nth Fibonacci number f_n *is defined by the following recursion*

$$f_n = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ f_n = f_{n-1} + f_{n-2} & n \geq 2 \end{cases}$$

This is the standard recursive definition for finding the nth Fibonacci number. Let's perform a few iterations of this function up to f_5.

$$f_0 = 0$$
$$f_1 = 1$$
$$f_2 = 1$$
$$f_3 = 2$$
$$f_4 = 3$$
$$f_5 = 5$$

Definition 5. *(Fibonacci words) For a Fibonacci word* F_n

$$F_n = \begin{cases} \varepsilon, & n = 0 \\ 1 & n = 1 \\ 0 & n = 2 \\ F_n = F_{n-1}F_{n-2} & n \geq 3 \end{cases}$$

In order to achieve the next word we concatenate the previous two. Similarly with the Fibonacci numbers let's compute some values of this function up to $n = 5$ starting at F_3.

$$F_3 = 01$$
$$F_4 = 010$$
$$F_5 = 01001$$

Definition 6. *For $n \geq 0$ and $0 \leq k \leq n$ we let $\langle k \rangle_n = F_{n-k}$ we also write $||k||_n = |\langle k \rangle_n|$ as the length of $\langle k \rangle_n$. When n is understood from context we write $\langle k \rangle = \langle k \rangle_n$ and $||k||_n = ||k||$.*

Example 3. *Suppose $n = 7$ and $k = 3$ then*

$$\langle 3 \rangle_7 = F_4 = 010 \ \ \text{and} \ \ |\langle 3 \rangle|_7 = |F_4| = 3.$$

Note that the string F_4 has the same length of Fibonacci number $f_4 = 3$. In general the string F_{n-k} has the length of the $n - k$ Fibonacci number. This result was proven in the paper [2] via induction by Dr. Kjos-Hanssen. Also to decompose Fibonacci words so that it's consistent with how we defined them we do the following

$$\langle k \rangle = \langle k+1 \rangle \langle k+2 \rangle$$

$$F_{n-k} = F_{n-(k+1)} F_{n-(k+2)}$$

Definition 7 (Fibonacci Constant)**.** *Define $\phi = \frac{1+\sqrt{5}}{2}$ to be to positive root of the polynomial $f(x) = x^2 - x - 1$.*

Definition 8 (Golden ratio)**.** *The ratio between terms of the Fibonacci sequence is*

$$\lim_{n\to\infty} \frac{f_n}{f_{n-1}} = \phi \Longleftrightarrow \lim_{n\to\infty} \frac{f_{n-1}}{f_n} = \frac{1}{\phi}.$$

Next we will need a lemma involving powers of $\frac{1}{\phi}$, that will be useful in the main result.

Definition 9 (The Infinite Fibonacci Word)**.** *The infinite Fibonacci word F_∞ is the fixed point of the morphism $\Phi : \Sigma^* \longmapsto \Sigma^*$ such that*

$$0 \underset{\Phi}{\longmapsto} 01$$

$$01 \underset{\Phi}{\longmapsto} 0$$

*and $\Phi(uv) = \Phi(u)\Phi(v)$ for $u, v \in \Sigma^*$. This word indeed exists and was proven in [2] as **Lemma 7.7**.*

Lemma 1. *For each $c \in \mathbb{N}$*

$$\lim_{n\to\infty} \prod_{k=0}^{c-1} \frac{f_{n-k}}{f_{n-(k-1)}} = \frac{1}{\phi^c}$$

Proof. Suppose $c \in \mathbb{N}$. It's a well known fact that

$$\lim_{n\to\infty} \frac{f_{n-1}}{f_n} = \frac{1}{\phi}. \tag{1}$$

Since we have a finite product that is independent of n, and the product of each of the limit exists provided Equation (1) we are permitted to swap the two yielding the following result

$$\lim_{n\to\infty}\prod_{k=0}^{c-1}\frac{f_{n-k}}{f_{n-(k-1)}}=\prod_{k=0}^{c-1}\lim_{n\to\infty}\frac{f_{n-k}}{f_{n-(k-1)}}=\prod_{k=0}^{c-1}\frac{1}{\phi}=\frac{1}{\phi^c}.$$

□

Theorem 1. *For $n \geq 5$ the equation*

$$xf_{n-1}+yf_n=2(f_{n-1}+f_n)$$

for non-negative integers x, y, has the unique solution $x = y = 2$.

Proof. For $x = 0$

$$y=\frac{2(f_{n-1+f_n})}{f_n}=\frac{2f_{n-1}+2f_n}{f_n}=2\frac{f_{n-1}}{f_n}+2\in(3,4)$$

provided that $n \geq 4$ otherwise y will be an integer.

For $x = 1$

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

provided that $n \geq 3$.

For $y = 0$

$$x=2+2\frac{f_n}{f_{n-1}}\in(5,6)$$

provided that $n \geq 5$.

For $y = 1$

$$x=2+\frac{f_n}{f_{n-1}}\in(3,4)$$

provided that $n \geq 4$.

For $x > 2$

$$y=\frac{f_{n-1}}{f_n}(2-x)+2\notin\mathbb{N}$$

provided that $n \geq 3$.

For $y > 2$

$$x=\frac{f_n}{f_{n-1}}(2-y)+2\notin\mathbb{N}$$

provided that $n \geq 4$.

For the cases when $x, y > 2$ and when n is large enough the respective outputs will be elements in the set $\mathbb{Q} \setminus \mathbb{Z}^+$. This can be seen more easily in desmos as a line graph where the ratio of Fibonacci numbers is the slope m, $2 = b$ is the y-intercept, and x, y are independent variables respectively. Thus we can conclude that $x = y = 2$ is the only solution with non-negative integers that satisfies the equation.

$\square$

Now we begin to show the main result. But first in order to build some intuition on how we can create the desired bound we first begin by looking at a simpler case.

Theorem 2. $\overline{A_N}(F_n) \leq \frac{1}{\phi^2} + \frac{1}{\phi^6}$.

Proof. We exploit the length of the Fibonacci words via decomposition. Recall the algorithm for breaking down an arbitrary Fibonacci word $\langle k \rangle = \langle k+1 \rangle \langle k+2 \rangle$. For this example we let $n = 9$ so we have $\langle k \rangle_9$. Decomposing we will have the following decomposition

$$\begin{aligned}
\langle 0 \rangle &= \langle 1 \rangle \langle 2 \rangle = \langle 2 \rangle \langle 3 \rangle \langle 2 \rangle \\
&= \langle 3 \rangle \langle 4 \rangle \langle 3 \rangle \langle 2 \rangle \\
&= \langle 4 \rangle \langle 5 \rangle \langle 4 \rangle \langle 3 \rangle \langle 2 \rangle \\
&= \langle 4 \rangle \langle 5 \rangle \langle 5 \rangle \langle 6 \rangle \langle 3 \rangle \langle 2 \rangle \\
&= \langle 4 \rangle \langle 5 \rangle \langle 6 \rangle \langle 7 \rangle \langle 6 \rangle \langle 3 \rangle \langle 2 \rangle \\
&= (\langle 4 \rangle \langle 4 \rangle \langle 7 \rangle) \ \ \langle 6 \rangle \ \ (\langle 3 \rangle \langle 3 \rangle \langle 4 \rangle)
\end{aligned}$$

Now we count the length of each $\langle k \rangle_9$ which will correspond to a certain number of states. This will be key in constructing the automaton.

$$\begin{aligned}
|\langle 4 \rangle|_9 &= 5 \\
|\langle 3 \rangle|_9 &= 8 \\
|\langle 7 \rangle|_9 &= 1 \\
|\langle 6 \rangle|_9 &= 2 \\
|\langle 0 \rangle|_9 &= 34
\end{aligned}$$

The corresponding automaton for this example would have at most $|\langle 0 \rangle| = 34$ states, but the minimum number of states needed comes from the sum

$$A_N(F_9) = |\langle 4 \rangle|_9 + |\langle 3 \rangle_9| + |\langle 6 \rangle|_9 = 15.$$

The automaton being built can be thought of as three pieces. For the first piece, in $\langle 4 \rangle \langle 4 \rangle \langle 7 \rangle$ we avoid over counting the number of states needed by looping around $\langle 4 \rangle$ twice, and then walking along one more edge which corresponds to $\langle 7 \rangle$. This will place us one state ahead of the start state, and is valid since

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

which tells us $\langle 7 \rangle$ is a prefix of $\langle 4 \rangle$. For rest of this paper we will use $\leq_p$ to denote *"is a prefix of"*.

Next ⟨6⟩ behaves like a bridge which will connect us to what we are about to see is another loop. Repeat a similar process in ⟨3⟩⟨3⟩⟨4⟩; go around ⟨3⟩ hitting the 8 states twice, and then end 5 states from the connected edge which is the accepted state. It should be clear that ⟨4⟩ is a prefix of ⟨3⟩. So via this decomposition we see that the corresponding automaton can be thought of as two cycles adjoined by a "bridge". More explicitly we can write a table for a transition function δ_9 as:

States (q)	Input {0,1}	Next State $\delta(q, a)$
q_o *(start)*	0	q_1
q_1	1	q_2
q_1	0	q_5
q_2	0	q_3
q_3	0	q_4
q_4	1	q_0
q_5	1	q_6
q_6	0	q_7
q_7	1	q_8
q_8	0	q_9
q_9	0	q_{10}
q_{10}	1	q_{11}
q_{11}	0	q_{12}
q_{12}	1	q_{13}
q_{13} (acpt. state)	0	q_6

This tells us the upper non-deterministic automatic complexity when $n = 9$ is

$$\frac{A_N(F_9)}{|\langle 0\rangle|} \leq \frac{5+8+2}{|\langle 0\rangle|} = \frac{15}{34} \approx 0.441$$

which gives us an intermediate complexity rate.

Now generalizing this

$$\langle 0\rangle = \underbrace{(\langle 4\rangle\langle 4\rangle\langle 7\rangle)}_{f_{n-4}\text{ state}} \underbrace{\langle 6\rangle}_{f_{n-6}\text{ state}} \underbrace{(\langle 3\rangle\langle 3\rangle\langle 4\rangle)}_{f_{n-3}\text{ state}}$$

note that the f_{n-4} state comes from the fact that $|\langle 4\rangle| = |F_{n-4}| = f_{n-4}$ which tells us that it indeed has f_{n-4} many states in that cycle. We can make similar calculations and deductions for ⟨6⟩ and ⟨4⟩ which will give us f_{n-6} and f_{n-4} many states respectively.

Now running through all $n \geq 9$, applying **definition 3** along with **Lemma** 1 we will have the following non deterministic automatic complexity:

$$\limsup_{n\to\infty} \frac{A_N(F_n)}{f_n} \leq \limsup_{n\to\infty} \frac{f_{n-3}+f_{n-6}+f_{n-4}}{f_n}$$

$$= \limsup_{n\to\infty} \frac{f_{n-2}+f_{n-6}}{f_n} = \frac{1}{\phi^2} + \frac{1}{\phi^6} \approx 0.412.$$

□

Theorem 3. $\overline{A_N}(F_\infty) \leq \frac{1}{\phi^2}$

Proof. We begin we with the last decomposition from **Theorem 2**, and manipulate the left side and middle term while $(\langle 3\rangle\langle 3\rangle\langle 4\rangle)$ remains fixed. Performing this algorithm will give us a corresponding sequence of automatons. Decomposing even further we have:

$$\langle 0\rangle = (\langle 4\rangle\langle 4\rangle\langle 7\rangle)\langle 6\rangle(\langle 3\rangle\langle 3\rangle\langle 4\rangle)$$

$$\langle 0\rangle = \langle 4\rangle\langle 4\rangle\langle 7\rangle\langle 7\rangle\langle 8\rangle(\langle 3\rangle\langle 3\rangle\langle 4\rangle)$$

$$\langle 0\rangle = \langle 4\rangle\langle 4\rangle\langle 7\rangle\langle 8\rangle\langle 9\rangle\langle 8\rangle(\langle 3\rangle\langle 3\rangle\langle 4\rangle)$$

$$= \underbrace{(\langle 4\rangle\langle 4\rangle\langle 6\rangle)}_{f_{n-4}\text{ state}}\ \underbrace{\langle 9\rangle\langle 8\rangle}_{f_{n-7}\text{ state}}\ \underbrace{(\langle 3\rangle\langle 3\rangle\langle 4\rangle)}_{f_{n-3}\text{ state}}$$

Even though concatenating Fibonacci words is not commutative (they are actually *almost commutative*) $\langle 9\rangle\langle 8\rangle$ still gives f_{n-7} states since

$$|\langle 9\rangle\langle 8\rangle| = |\langle 9\rangle| + |\langle 8\rangle| = |\langle 8\rangle| + |\langle 9\rangle| = |\langle 8\rangle\langle 9\rangle| = |\langle 7\rangle|.$$

Again letting $n \to \infty$ we will achieve a smaller bound of $A_N(F_n)$

$$\limsup_{n\to\infty} \frac{A_N(F_n)}{f_n} \leq \limsup_{n\to\infty} \frac{f_{n-3}+f_{n-7}+f_{n-4}}{f_n}$$
$$\limsup_{n\to\infty} \frac{f_{n-2}+f_{n-7}}{f_n} = \frac{1}{\phi^2} + \frac{1}{\phi^7}.$$

Also note that the left portion on the right hand side of this equation

$$\langle 0\rangle = \underbrace{(\langle 4\rangle\langle 4\rangle\langle 6\rangle)}_{f_{n-4}\text{ state}}\ \underbrace{\langle 9\rangle\langle 8\rangle}_{f_{n-7}\text{ state}}\ \underbrace{(\langle 3\rangle\langle 3\rangle\langle 4\rangle)}_{f_{n-3}\text{ state}}$$

will give us the f_{n-4} state in the limit, since $|\langle 4\rangle|$ is greater in length which implies more states. Similarly we have the f_{n-3} state that remains fixed on the far right when taking the limit, and then all the interesting changes happen in the middle state via construction. When computing the non deterministic automatic complexity this permits us to have the f_{n-2} state plus some extra bit that we expect to converge to 0 for n large enough (see Figure 1).

Regrouping we can again have a better bound

$$\langle 0\rangle = \underbrace{(\langle 4\rangle\langle 4\rangle\langle 6\rangle\langle 9\rangle)}_{f_{n-4}\text{ state}}\ \underbrace{\langle 8\rangle}_{f_{n-8}\text{ state}}\ \underbrace{(\langle 3\rangle\langle 3\rangle\langle 4\rangle)}_{f_{n-3}\text{ state}}.$$

This works since $\langle 6\rangle\langle 9\rangle = \langle 7\rangle\langle 8\rangle\langle 9\rangle \leq_p \langle 4\rangle$.

$$\limsup_{n\to\infty} \frac{A_N(F_n)}{n} \leq \limsup_{n\to\infty} \frac{f_{n-2}+f_{n-8}}{f_n} = \frac{1}{\phi^2} + \frac{1}{\phi^8}.$$

Expanding, regrouping, and taking the limit again, we achieve a better bound that will work since $\langle 6\rangle\langle 9\rangle\langle 9\rangle = \langle 7\rangle\langle 8\rangle\langle 9\rangle\langle 9\rangle \leq_p \langle 4\rangle$.

$$\langle 0\rangle = (\langle 4\rangle\langle 4\rangle\langle 7\rangle\langle 8\rangle\langle 9\rangle\langle 9\rangle)\ \langle 10\rangle\ (\langle 3\rangle\langle 3\rangle\langle 4\rangle)$$

$$\limsup_{n\to\infty} \frac{A_N(F_n)}{n} \leq \limsup_{n\to\infty} \frac{f_{n-2}+f_{n-10}}{f_n} = \frac{1}{\phi^2} + \frac{1}{\phi^{10}}.$$

From this point we only pull apart the middle piece, and of the two new parts the words labeled $\langle 2m-1\rangle$ for $m \geq 6$ are grouped with the f_{n-4} state while the other of the form $\langle 2m\rangle$ for $m \geq 6$ becomes isolated. Putting this into effect will give us the following

$$\langle 0\rangle = (\langle 4\rangle\langle 4\rangle\langle 7\rangle\langle 8\rangle\langle 9\rangle\langle 9\rangle\langle 11\rangle)\ \langle 12\rangle\ (\langle 3\rangle\langle 3\rangle\langle 4\rangle)$$

$$\limsup_{n\to\infty} \frac{A_N(F_n)}{f_n} \leq \limsup_{n\to\infty} \frac{f_{n-2}+f_{n-12}}{f_n} = \frac{1}{\phi^2} + \frac{1}{\phi^{12}}.$$

Again expanding, regrouping, and taking the limit

$$\langle 0\rangle = (\langle 4\rangle\langle 4\rangle\langle 7\rangle\langle 8\rangle\langle 9\rangle\langle 9\rangle\langle 11\rangle\langle 13\rangle)\ \langle 14\rangle\ (\langle 3\rangle\langle 3\rangle\langle 4\rangle)$$

$$\limsup_{n\to\infty} \frac{A_N(F_n)}{f_n} \leq \limsup_{n\to\infty} \frac{f_{n-2}+f_{n-14}}{f_n} = \frac{1}{\phi^2} + \frac{1}{\phi^{14}}.$$

One last time expanding, regrouping, and taking the limit will produce:

$$\langle 0\rangle = (\langle 4\rangle\langle 4\rangle\langle 7\rangle\langle 8\rangle\langle 9\rangle\langle 9\rangle\langle 11\rangle\langle 13\rangle\langle 15\rangle)\ \langle 16\rangle\ (\langle 3\rangle\langle 3\rangle\langle 4\rangle)$$

$$\limsup_{n\to\infty} \frac{A_N(F_n)}{f_n} \leq \limsup_{n\to\infty} \frac{f_{n-2}+f_{n-16}}{f_n} = \frac{1}{\phi^2} + \frac{1}{\phi^{16}}.$$

Generalizing this pattern we are able to write our decomposition as

$$\langle 0\rangle = (\langle 4\rangle\langle 4\rangle\langle 7\rangle\langle 8\rangle\langle 9\rangle\langle 9\rangle\langle 11\rangle\langle 13\rangle\langle 15\rangle\langle 17\rangle\ldots\langle 2k-1\rangle)\ \langle 2k\rangle\ (\langle 3\rangle\langle 3\rangle\langle 4\rangle)$$

$$\langle 0\rangle = (\langle 4\rangle\langle 4\rangle\langle 7\rangle\langle 9\rangle\langle 9\rangle \prod_{i=5}^{k-1}\langle 2i+1\rangle)\ \langle 2k\rangle\ (\langle 3\rangle\langle 3\rangle\langle 4\rangle)$$

We show this key part of our decomposition to be true via induction.

Claim 1.

$$\left(\prod_{i=5}^{k-1}\langle 2i+1\rangle\right)\ \langle 2k\rangle = \langle 10\rangle$$

Base Case: *If* $k=6$ *then*

$$\prod_{i=5}^{5}\langle 2i+1\rangle\ \ \langle 12\rangle = \langle 11\rangle\langle 12\rangle = \langle 10\rangle.\checkmark$$

Induction Hypothesis: *Suppose*

$$\prod_{i=5}^{k-1}\langle 2i+1\rangle)\ \ \langle 2k\rangle = \langle 10\rangle$$

is true for $k \geq 6$

Inductive step: $k \to k+1$
Substituting $k+1$ *for* k *we will have*

$$\prod_{i=5}^{k}\langle 2i{+}1\rangle)\ \ \langle 2(k{+}1)\rangle = \prod_{i=5}^{k-1}\langle 2i{+}1\rangle)\ \ \underbrace{\langle 2k+1\rangle\langle 2k+2\rangle}_{\langle 2k\rangle} = \prod_{i=5}^{k-1}\langle 2i{+}1\rangle\ \ \langle 2k\rangle = \langle 10\rangle.\checkmark$$

Continuing in this manner the bound for $\bar{A}_N(\mathbf{f}\ ^{(2)})$ can be achieved in the limit

$$\limsup_{n\to\infty}\frac{A_N(F_n)}{f_n} \leq \limsup_{k\to\infty}\left(\frac{1}{\phi^2}+\frac{1}{\phi^{2k}}\right) = \frac{1}{\phi^2}.$$

Since the NFA consists of a cycle of finite length containing the start state, followed by another cycle that also has finite length containing the final state and nothing else, then there will be a unique walk of finite length from the start state to the final state since

$$xf_{n-1} + yf_n = 2(f_{n-1} + f_n)$$

has the unique solution x = y = 2 for $n \geq 5$ by **Theorem 1**.

□

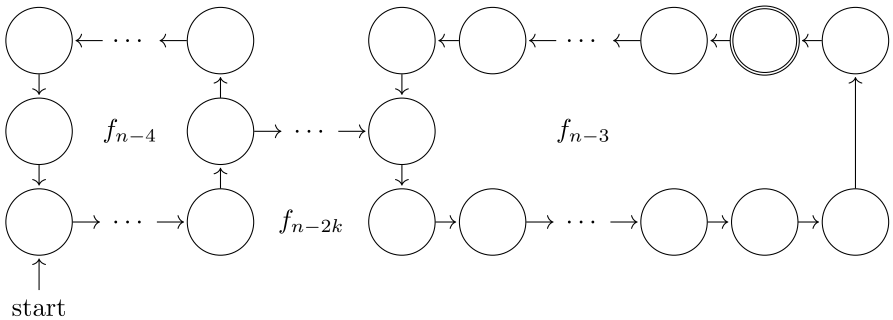


Figure 1: The corresponding automaton witnessing the automatic complexity of an infinite Fibonacci word F_∞. When computing $A_N(x)$ as $n \to \infty$ the corresponding automaton will have more states in the two big cycles while the edge with the single state that behaves as a bridge can be "ignored".

References

[1] Liam Jordon and Philippe Moser. Normal sequences with non-maximal automatic complexity. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 213 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 47, 16. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2021.

[2] Bjørn Kjos-Hanssen. Automatic complexity of Fibonacci and Tribonacci words. *Discrete Appl. Math.*, 289:446–454, 2021.

[3] Jeffrey Shallit and Ming-Wei Wang. Automatic complexity of strings. volume 6, pages 537–554. 2001. 2nd Workshop on Descriptional Complexity of Automata, Grammars and Related Structures (London, ON, 2000).