The Mathematics of Per Nørgård's Rhythmic Infinity System

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

September 10, 2003

1 Introduction

The Danish composer Per Nørgård (1932–) invented a procedure for generating rhythms which was described by Erling Kullberg [5]. Reworded in mathematical notation, this procedure is as follows:

Let the Fibonacci numbers $(F_n)_{n\geq 0}$ be defined as usual by $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$. Starting with the pair $(c_0, c_1) = (F_{2n}, F_{2n+1})$, perform the following operation $n-2$ times:

- If a number F_i appears in an even-indexed position, replace it with (F_{i-2}, F_{i-1})
- If a number F_i appears in an odd-indexed position, replace it with (F_{i-1}, F_{i-2})

Kullberg illustrates this procedure in the case $n = 5$, as follows:

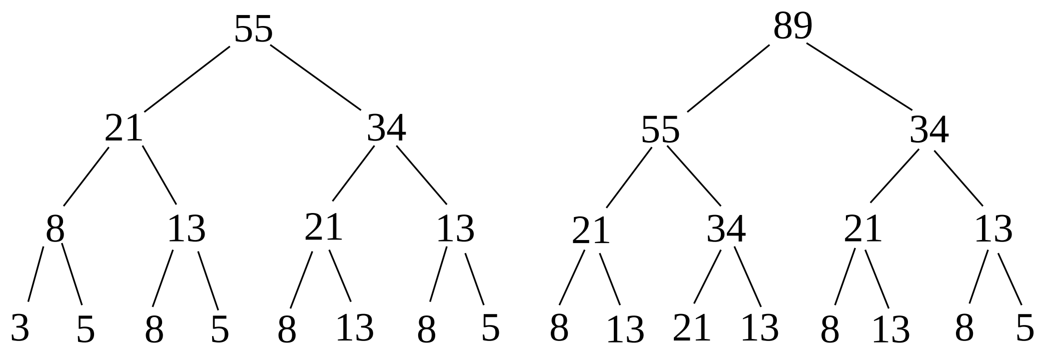

Figure 1: Generating the rhythmic infinity series

*Supported in part by a grant from NSERC.

Here, starting with the pair $(55, 89)$, we replace 55 by $(21, 34)$ and 89 by $(55, 34)$ to get the quadruple $(21, 34, 55, 34)$, and so forth.

After $n - 2$ iterations, the resulting sequence is of length 2^{n-1}. As $n \to \infty$ we get a limiting sequence $(a_i)_{i \geq 0}$:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	$\cdots$
a_i	3	5	8	5	8	13	8	5	8	13	21	13	8	13	8	5	8	13	$\cdots$

In this paper I obtain an explicit formula for the sequence $(a_i)_{i \geq 0}$ and show how it is related to binary Gray code.

We can see the structure of the sequence $(a_i)_{i \geq 0}$ more easily if we replace each number in Figure 1 by the corresponding Fibonacci number, as follows:

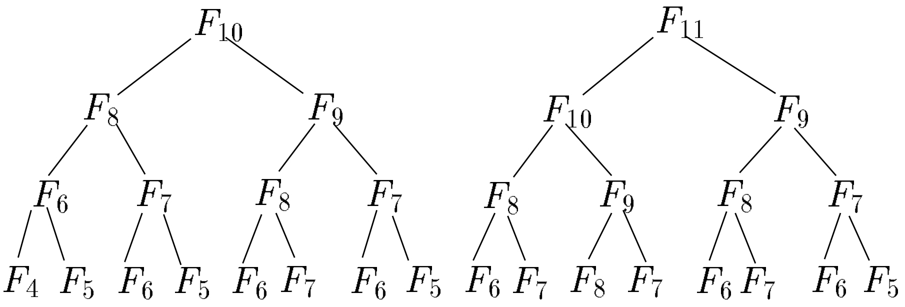

Figure 2: Generating the rhythmic infinity system

This gives us a sequence $(b_i)_{i \geq 0}$ defined by $a_i = F_{b_i}$:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	$\cdots$
b_i	4	5	6	5	6	7	6	5	6	7	8	7	6	7	6	5	6	7	$\cdots$

Finally, if we define $c_i = b_i - 4$, we get the following sequence:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	$\cdots$
c_i	0	1	2	1	2	3	2	1	2	3	4	3	2	3	2	1	2	3	$\cdots$

We now find another way to generate the sequence $(c_i)_{i \geq 0}$: through iterated morphisms.

Let Σ be a finite set of symbols, called an *alphabet*. Then Σ^* denotes the set of all finite strings with symbols chosen from Σ. For example,

$$\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}.$$

Here ϵ is the symbol for the empty string.

A *morphism* is a map $h : \Sigma^* \to \Sigma^*$ that satisfies the identity $h(xy) = h(x)h(y)$ for all strings $x, y \in \Sigma^*$. A morphism may be iterated by defining h^0 to be the identity map (i.e., $h^0(x) = x$ for all $x \in \Sigma^*$) and $h^i(x) = h^{i-1}(h(x))$ for $i \geq 1$.

Iterated morphisms have been used by the composer Tom Johnson in some of his work; for more details see [1, 2].

To generate $(c_i)_{i\geq 0}$ we may model Nørgård's transformation as follows: we define a map

$$\mu : [a, b] \to [a-2, a-1][b-1, b-2].$$

This map can be extended to a morphism on sequences of pairs using the rule $\mu(xy) = \mu(x)\mu(y)$. Then the first 2^{n-1} terms of the sequence $(b_i)_{i\geq 0}$ are given by $\mu^{n-2}([2n, 2n+1])$, and the first 2^{n+1} terms of the sequence $(c_i)_{i\geq 0}$ are given by $\mu^n([2n, 2n+1])$.

For example:

$$\begin{aligned}
\mu^0([6,7]) &= [6,7] \\
\mu^1([6,7]) &= [4,5][6,5] \\
\mu^2([6,7]) &= [2,3][4,3][4,5][4,3] \\
\mu^3([6,7]) &= [0,1][2,1][2,3][2,1][2,3][4,3][2,3][2,1]
\end{aligned}$$

This generates the sequence $(c_i)_{i\geq 0}$ in a "top-down" fashion.

To generate $(c_i)_{i\geq 0}$ in a "bottom-up" fashion we introduce a morphism φ defined by

$$\begin{aligned}
\varphi([a, a+1]) &= [a, a+1][a+2, a+1] \\
\varphi([a+1, a]) &= [a+1, a+2][a+1, a]
\end{aligned}$$

Theorem 1 *For $n \geq 0$ we have*

$$\mu^n([2n, 2n+1]) = \varphi^n([0,1]). \tag{1}$$

Proof. It turns out to be useful to prove something more general. Namely, we prove the following two equations simultaneously by mathematical induction on n:

$$\mu^n([k, k+1]) = \varphi^n([k-2n, k+1-2n]); \tag{2}$$

$$\mu^n([k+1, k]) = \varphi^n([k+1-2n, k-2n]); \tag{3}$$

for all integers k.

It is easy to see (2) and (3) hold for $n = 0$. Now assume (2) and (3) hold for n; we prove them for $n+1$.

$$\begin{aligned}
\mu^{n+1}([k, k+1]) &= \mu^n(\mu([k, k+1])) \\
&= \mu^n([k-2, k-1][k, k-1]) \\
&= \mu^n([k-2, k-1])\ \mu^n([k, k-1]) \\
&= \varphi^n([k-2-2n, k-1-2n])\ \varphi^n([k-2n, k-1-2n]) \\
&= \varphi^n([k-2-2n, k-1-2n][k-2n, k-1-2n]) \\
&= \varphi^n(\varphi([k-2-2n, k-1-2n])) \\
&= \varphi^{n+1}([k-2(n+1), k+1-2(n+1)]).
\end{aligned}$$

Similarly

$$\begin{aligned}
\mu^{n+1}([k+1,k]) &= \mu^n(\mu([k+1,k])) \\
&= \mu^n([k-1,k][k-1,k-2]) \\
&= \mu^n([k-1,k])\ \mu^n([k-1,k-2]) \\
&= \varphi^n([k-1-2n,k-2n])\ \varphi^n([k-1-2n,k-2-2n]) \\
&= \varphi^n([k-1-2n,k-2n][k-1-2n,k-2-2n]) \\
&= \varphi^n(\varphi([k-1-2n,k-2-2n]) \\
&= \varphi^{n+1}([k+1-2(n+1),k-2(n+1)]).
\end{aligned}$$

Finally, the desired result (1) follows by setting $k = 2n$ in (2). ■

It now follows that we can generate the sequence c_i by iterating the morphism φ starting with $[0,1]$. For example

$$\begin{aligned}
\varphi^0([0,1]) &= [0,1] \\
\varphi^1([0,1]) &= [0,1][2,1] \\
\varphi^2([0,1]) &= [0,1][2,1][2,3][2,1] \\
\varphi^3([0,1]) &= [0,1][2,1][2,3][2,1][2,3][4,3][2,3][2,1] \\
&\vdots
\end{aligned}$$

As a consequence we get

Corollary 2

$$\varphi([c_{2i},c_{2i+1}]) = [c_{4i},c_{4i+1}][c_{4i+2},c_{4i+3}].$$

We now introduce the so-called "pattern functions" $e_P(n)$. Let P be a string of 0's and 1's. Then $e_P(n)$ counts the number of (possibly overlapping) occurrences of P in the base-2 expansion of n. For example, $e_{10}(12) = 1$, since the base-2 representation of 12 is 1100, and this contains one occurrence of 10.

In the case where P starts with a 0, some additional elaboration is necessary. In this case we assume that the base-2 representation of n starts with $|P| - 1$ zeroes. For example, $e_{01}(12) = 1$.

We define $d_n = e_{01}(n) + e_{10}(n)$. It is easy to see that, for $n > 0$, the quantity d_n counts the number of distinct blocks of adjacent identical symbols in the binary expansion of n. For example, the binary expansion of 399 is 110001111, which has 3 blocks (namely 11, 000, and 1111). We have $d_{399} = e_{01}(399) + e_{10}(399) = 2 + 1 = 3$.

Here is a table:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	$\cdots$
$e_{01}(i)$	0	1	1	1	1	2	1	1	1	2	2	2	1	2	1	1	1	2	$\cdots$
$e_{10}(i)$	0	0	1	0	1	1	1	0	1	1	2	1	1	1	1	0	1	1	$\cdots$
d_i	0	1	2	1	2	3	2	1	2	3	4	3	2	3	2	1	2	3	$\cdots$

Theorem 3 *We have $c_n = d_n$ for $n \geq 0$.*

Proof. By comparing the binary expansions of $2n$, $2n+1$ with those of $4n$, $4n+1$, $4n+2$, $4n+3$, we easily see that

$$\begin{aligned} d_{4n} &= d_{2n} \\ d_{4n+1} &= d_{2n}+1 \\ d_{4n+2} &= d_{2n+1}+1 \\ d_{4n+3} &= d_{2n+1} \end{aligned}$$

for $n \geq 0$. Since $c_0 = d_0 = 0$, the equality $c_n = d_n$ for all $n \geq 0$ will follow if we can show that $(c_n)_{n\geq 0}$ satisfies the same relations as those for d given above.

To see this, we consider the case n even and n odd separately.

If n is even, then $c_{2n+1} = c_{2n}+1$. Using this fact and Corollary 2, we find

$$\begin{aligned} [c_{4n}, c_{4n+1}][c_{4n+2}, c_{4n+3}] &= \varphi([c_{2n}, c_{2n+1}]) \\ &= \varphi([c_{2n}, c_{2n}+1]) \\ &= [c_{2n}, c_{2n}+1][c_{2n}+2, c_{2n}+1] \\ &= [c_{2n}, c_{2n}+1][c_{2n+1}+1, c_{2n+1}], \end{aligned}$$

from which the desired relations follow.

If n is odd, then $c_{2n+1} = c_{2n}-1$. Using this fact and Corollary 2 again, we find

$$\begin{aligned} [c_{4n}, c_{4n+1}][c_{4n+2}, c_{4n+3}] &= \varphi([c_{2n}, c_{2n+1}]) \\ &= \varphi([c_{2n}, c_{2n}-1]) \\ &= [c_{2n}, c_{2n}+1][c_{2n}, c_{2n}-1] \\ &= [c_{2n}, c_{2n}+1][c_{2n+1}+1, c_{2n+1}], \end{aligned}$$

from which the desired relations follow. ■

The sequence $(d_n)_{n\geq 0}$ defined by $d_n = e_{01}(n) + e_{10}(n)$ is well-known: in addition to its characterization as the number of distinct blocks of adjacent identical symbols in the binary expansion of n, it is also the sum of the bits in the Gray code representation of n [4, 3]. From this, the identity $|d_n - d_{n-1}| = 1$ for $n \geq 1$ easily follows. This explains its attractiveness as a basis for music composition: the sequence $(d_n)_{n\geq 1}$ makes no large jumps, and hence when used as an index into the Fibonacci numbers it "alternately expands and contracts in a gently undulating form" [5].

We can now prove our closed-form for Nørgård's rhythmic infinity sequence:

Theorem 4 *We have $a_i = F_{d(i)+4} = F_{e_{01}(i)+e_{10}(i)+4}$ for $i \geq 0$.*

Proof. We have $c_i = d_i = e_{01}(i) + e_{10}(i)$ by Theorem 3. On the other hand, by definition we have $c_i = b_i - 4$ and $a_i = F_{b_i}$. Putting this all together gives the desired relation for a_i. ■

Next we give an additional method of generating the sequence $(c_i)_{i\geq 0}$. Define

$$\begin{aligned} X_n &= c_0 c_1 c_2 \cdots c_{2^n-1} \\ Y_n &= c_{2^n} c_{2^n+1} \cdots c_{2^{n+1}-1} \end{aligned}$$

for $n \geq 0$; thus X_n and Y_n are blocks of 2^n symbols. Let X be a block of symbols. By $X + a$ we mean the block that results by adding a to each symbol in X.

Theorem 5 *We have*

$$\begin{aligned} X_{n+1} &= X_n Y_n; \\ Y_{n+1} &= (X_n + 2) Y_n. \end{aligned}$$

Proof. The result for X_n follows immediately from the definition. Thus it suffices to show that

$$c_{2^{n+1}+a} = c_a + 2$$

and

$$c_{2^{n+1}+2^n+a} = c_{2^n+a}$$

for $0 \leq a < 2^n$. These identities follow immediately from Theorem 3 and consideration of the binary expansion. ■

Finally, we observe that the sequences $(b_i)_{i\geq 0}$ and $(c_i)_{i\geq 0}$ are members of a much more general class of sequences, the so-called 2-regular sequences [3]. In fact, even the sequence $(a_i)_{i\geq 0}$ is 2-regular, as our last theorem shows:

Theorem 6 *We have*

$$\begin{aligned} a_{4i} &= a_{2i} \\ a_{4i+2} &= -a_i + 2a_{2i} + 2a_{2i+1} - a_{4i+1} \\ a_{4i+3} &= a_{2i+1} \\ a_{8i+1} &= a_{4i+1} \\ a_{8i+5} &= -a_i + 2a_{2i} + 3a_{2i+1} - a_{4i+1} \end{aligned}$$

for all $i \geq 0$.

Proof. These relations follow easily from Theorem 4. For example, let us prove the identity for a_{4i+2}. There are two cases to consider: when i is even and when i is odd.

If i is even, say $i = 2k$, then

$$\begin{aligned} -a_{2k} + 2a_{4k} + 2a_{4k+1} - a_{8k+1} &= -F_{d_{2k}+4} + 2F_{d_{4k}+4} + 2F_{d_{4k+1}+4} - F_{d_{8k+1}+4} \\ &= -F_{d_{2k}+4} + 2F_{d_{2k}+4} + 2F_{d_{2k}+5} - F_{d_{2k}+5} \\ &= F_{d_{2k}+4} + F_{d_{2k}+5} \\ &= F_{d_{2k}+6} \\ &= F_{d_{8k+2}+4} \\ &= a_{8k+2}. \end{aligned}$$

Here we have used the identities $d_{8k+2} = d_{2k} + 2$, $d_{4k} = d_{2k}$, $d_{4k+1} = d_{2k} + 1$, $d_{8k+1} = d_{2k} + 1$, which are easily verified by considering the binary expansion of k.

If i is odd, say $i = 2k + 1$, then

$$\begin{aligned}
-a_{2k+1} + 2a_{4k+2} + 2a_{4k+3} - a_{8k+5} &= -F_{d_{2k+1}+4} + 2F_{d_{4k+2}+4} + 2F_{d_{4k+3}+4} - F_{d_{8k+5}+4} \\
&= -F_{d_{2k+1}+4} + 2F_{d_{2k+1}+5} + 2F_{d_{2k+1}+4} - F_{d_{2k+1}+6} \\
&= F_{d_{2k+1}+4} + F_{d_{2k+1}+3} \\
&= F_{d_{2k+1}+5} \\
&= F_{d_{8k+6}+4} \\
&= a_{8k+6}.
\end{aligned}$$

Verification of the remaining identities is left to the reader. ■

References

[1] J.-P. Allouche and T. Johnson. Finite automata and morphisms in assisted musical composition. *J. New Music Research* **24** (1995), 97–108.

[2] J.-P. Allouche and T. Johnson. Narayana's cows and delayed morphisms. In *Cahiers du GREYC, Troisièmes Journées d'Informatique Musicale (JIM 96)*, Vol. 4, pp. 2–7, 1996.

[3] J.-P. Allouche and J. O. Shallit. The ring of k-regular sequences. *Theoret. Comput. Sci.* **98** (1992), 163–197.

[4] P. Flajolet and L. Ramshaw. A note on Gray code and odd-even merge. *SIAM J. Comput.* **9** (1980), 142–158.

[5] Erling Kullberg. Beyond infinity: On the infinity series — the DNA of hierarchical music. In A. Beyer, editor, *The Music of Per Nørgård: Fourteen Interpretive Essays*, pp. 71–93. Scolar Press, 1996.

AMS 2000 Mathematics Subject Classifcations: Primary 11B39; Secondary 68Q45, 11A63.