A PROBLEM ON GENERATION SETS CONTAINING FIBONACCI NUMBERS

DANIELLE COX AND KARYN MCLELLAN

Abstract. At the Sixteenth International Conference on Fibonacci Numbers and Their Applications the following problem was posed by Clark Kimberling:

Let S be the set generated by these rules: Let $1 \in S$ and if $x \in S$, then $2x \in S$ and $1 - x \in S$, so that S grows in generations:

$$G_1 = \{1\}, G_2 = \{0, 2\}, G_3 = \{-1, 4\}, \ldots.$$

Prove or disprove that each generation contains at least one Fibonacci number or its negative.

In this paper we generalize the problem as follows. Let S be the set described above, $\mathcal{S}$ be a sequence and $\mathcal{P}$ the property that a generation contains a term of $\mathcal{S}$ or the negative of a term of $\mathcal{S}$. We will show that when $\mathcal{S}$ is the Fibonacci sequence there are many generations that fail to have property $\mathcal{P}$. Other sequences $\mathcal{S}$ will also be considered and shown to have at least one generation failing to have property $\mathcal{P}$. The proportion of generations failing to have property $\mathcal{P}$ is also investigated.

1. Introduction

The following problem was posed by Clark Kimberling at the Sixteenth International Conference on Fibonacci Numbers and Their Applications [1].

Problem 1.1. *Let S be the set generated by these rules: Let $1 \in S$ and if $x \in S$, then $2x \in S$ (Operation 1) and $1 - x \in S$ (Operation 2), so that S grows in generations:*

$$G_1 = \{1\}, G_2 = \{0, 2\}, G_3 = \{-1, 4\}, \ldots.$$

Prove or disprove that each generation contains at least one Fibonacci number or its negative.

In this paper, we will generalize this problem as follows. Let S be the set described in Problem 1.1, $\mathcal{S}$ be a sequence and $\mathcal{P}$ the property that a generation contains a term of $\mathcal{S}$ or the negative of a term of $\mathcal{S}$. We will show that when $\mathcal{S}$ is the Fibonacci sequence, there are many generations that fail to have property $\mathcal{P}$.

First, we introduce the necessary definitions and notation. We will slightly modify the generations given in the original problem to include a zeroth generation.

Definition 1.2. *Let S be the set generated by Operation 1 and Operation 2. Start with $G_0 = \{0\}$, and for $i \geq 1$ let G_i denote the set of new elements added to S at the ith iteration of its creation. We call this **generation** i and refer to i as the **generation index**.*

Note that by this definition, the sets G_i are pairwise disjoint, because they consist only of new elements added to S. In other words, an integer can belong to only one generation.

Example 1.3. *Here are the first several generations of S:*

- $G_0 = \{\mathbf{0}\}$
- $G_1 = \{\mathbf{1}\}$
- $G_2 = \{\mathbf{2}\}$
- $G_3 = \{\mathbf{-1}, 4\}$

- G_4 = {**-3**, **-2**, **8**}
- G_5 = {-7, -6, -4, **3**, 16}
- G_6 = {-15, -14, -12, **-8**, **5**, 6, 7, 32}
- G_7 = {-31, -30, -28, -24, -16, **-5**, 9, 10, 12, **13**, 14, 15, 64}

The Fibonacci numbers and their negatives are seen in bold and as the example demonstrates, property $\mathcal{P}$ holds true for these generations. We will show that every integer can be found in some G_i, and knowing that all integers appear in S, we will disprove the proposition in Problem 1.1 by finding an expression that, given any integer z, will compute the generation index for the generation containing z.

2. Generations Containing a Particular Integer

In order to find an expression for the generation index we need a better understanding of how the generations are computed. The following result outlines the recursive structure of the generations.

Theorem 2.1. *For $i \geq 3$, the elements of G_i come from doubling all terms of G_{i-1} and subtracting the double of all terms of G_{i-2} from 1, i.e., $G_i = \{2x \mid x \in G_{i-1}\} \cup \{1 - 2x \mid x \in G_{i-2}\}$.*

Proof. Consider an element $x \in G_i$, for $i \geq 3$. We will determine the sequence of applications of Operation 1 and Operation 2 that produce x. This sequence of operations is dependent on the parity of x, so we break our proof into two cases.
Case 1. x is even. Let $x = 2\ell$, where $\ell \in \mathbb{Z}$. Based on the operations for generating S, there are two possible scenarios for the creation of x.

The first is that x comes from Operation 1, so ℓ is in G_{i-1}.

The other is that x comes from Operation 2; that is, the number $1 - 2\ell$ is found in G_{i-1}. Notice that $1 - 2\ell$ is odd, thus it could not have been obtained from implementing Operation 1 on an element of G_{i-2}. It had to come from applying Operation 2 to some y in G_{i-2}, so $1 - 2\ell = 1 - y$. This implies that $y = 2\ell = x$, contradicting that x is in G_i. Therefore, all even integers in G_i must have come from doubling elements in G_{i-1}.
Case 2. x is odd. Since x is odd it could not have come from Operation 1, so we must have $x = 1 - y$ for some $y \in G_{i-1}$. Since y is even, it came from applying Operation 1 to some integer ℓ in G_{i-2}. Therefore, all odd numbers in G_i must be of the form $1 - 2\ell$, for $\ell \in G_{i-2}$. Notice that Operation 2 is applied only to even numbers in G_{i-1} and produces no new elements when applied to odd numbers in G_{i-1}.

The converse, that all terms in G_{i-1} and G_{i-2} produce terms in G_i according to the rules of the theorem, must hold. If not, we can suppose there exists an $\ell \in G_{i-1}$, such that $2\ell \notin G_i$. The only reason 2ℓ would not be found in G_i, according to the rules of the problem, would be if it already appeared in an earlier generation. If this is the case, then ℓ would appear in the generation previous to that, which contradicts the fact that it belongs to G_{i-1}. Similarly, this argument holds for elements of the form $1 - 2\ell$, coming from G_{i-2}. □

The following result about the sizes of generations is now immediate.

Corollary 2.2. *The size of the set G_i is the Fibonacci number F_i, for $i \geq 1$.*

Proof. This follows from a simple induction, because $|G_1| = 1$, $|G_2| = 1$, and by Theorem 2.1, $|G_i| = |G_{i-1}| + |G_{i-2}|$. □

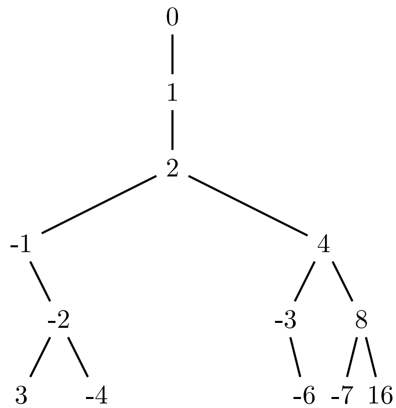

FIGURE 1. Binary Tree

An easy way to understand the structure and size of the generations is to use a binary tree, as seen in Figure 1. Generations 0, 1, and 2 each have one node, generation 3 has two nodes, namely -1 and 4, and so on, as Example 1.3 demonstrates. Theorem 2.1 tells us that each element in G_{i-1} has its double in G_i, for $i \geq 3$. We structure the tree so that these doubles are the right children. We also know that Operation 2 is applied only to even elements in G_{i-1}, and we let these be the left children of these elements. This means that even labeled nodes in the tree have two children, whereas odd labeled nodes only have a right child, as the left child of an odd labeled node would be precisely its parent, so is not included. We can easily visualize the fact that every node in the tree has a unique path (i.e., a unique sequence of operations) back to 0.

We can now state the following important result.

Theorem 2.3. *All integers belong to the set S.*

Proof. We have already seen that $0, 1, 2 \in S$. Let $z \in \mathbb{Z}$ and suppose that $z \notin S$. Recall from Theorem 2.1 that any even element $2\ell \in G_i$ comes from $\ell \in G_{i-1}$, for $i \geq 3$. Also, any odd element, $1 - 2\ell \in G_i$ comes from $\ell \in G_{i-2}$. With one exception, ℓ is less than both 2ℓ and $1 - 2\ell$ in absolute value, and so applying these operations to z gives a decreasing sequence (in absolute value) back to either 1 or 2 (see Figure 1). The exception is $z = -1$, which gives a non-increasing sequence to 1. Therefore, if $z \notin S$, it follows that 1 or 2 is not in S either, by the rules given in Problem 1.1. This is a contradiction, and so $z \in S$ for all $z \in \mathbb{Z}$. $\square$

TABLE 1. The generation indices for the first 7 whole numbers and their negatives.

k	i	k	i	k	i	k	i
		2	2	4	3	6	6
0	0	-2	4	-4	5	-6	5
1	1	3	5	5	6	7	6
-1	3	-3	4	-5	7	-7	5

We now turn our attention to investigating the generation index for each integer; see Table 1. The next five theorems completely characterize the patterns found in Table 1, and are dependent on the parity of the numbers involved.

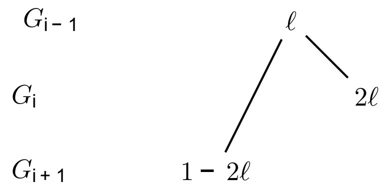

FIGURE 2. Patterns in the binary tree.

Theorem 2.4. *When moving from a negative odd number to a positive even number in Table 1, that is from* $(1-2\ell)$ *to* 2ℓ, $\ell \geq 1$, *the generation index decreases by 1.*

Proof. Consider a positive even number $k = 2\ell$ with $\ell \geq 1$. If 2ℓ appears in G_i, by Theorem 2.1 it came from ℓ in G_{i-1}. Let $1-2\ell$ be the negative odd number appearing just before 2ℓ in Table 1. By Theorem 2.1, $1-2\ell$ comes from ℓ, but appears two generations later, so $1-2\ell$ is in G_{i+1}. Therefore, moving from a negative odd number to a positive even number the generation index decreases by 1 (see Figure 2). □

Theorem 2.5. *When moving from a negative even number to a positive odd number in Table 1, that is from* -2ℓ *to* $2\ell+1$, $\ell \geq 1$, *the generation index increases by 1.*

Proof. Let $x = -2\ell$, where $\ell \geq 1$, be an negative even number. If -2ℓ appears in G_i, by Theorem 2.1 it came from $-\ell$ in G_{i-1}. Let $2\ell+1$ be the positive odd number appearing after -2ℓ in Table 1. By Theorem 2.1, $2\ell+1$ comes from $-\ell$, but appears two generations later, so $2\ell+1$ is in G_{i+1}. Therefore, moving from a negative even number to a positive odd number the generation index increases by 1. (See Figure 2 with ℓ replaced by $-\ell$.) □

We now consider moving from odd to odd or even to even.

Theorem 2.6. *When moving from a positive odd number* $2\ell-1$, *where* $\ell \geq 3$, *to a negative odd number* $1-2\ell$ *in Table 1, the generation index increases by 1 if* $2\ell-1 \equiv 1 \pmod 4$ *and decreases by 1 if* $2\ell-1 \equiv 3 \pmod 4$.

Proof. Let $2\ell-1$, where $\ell \geq 3$, be a positive odd number, and $1-2\ell$, be the negative odd number appearing after it in Table 1. By Theorem 2.1, $2\ell-1$ comes from $1-\ell$ two generations back, and $1-2\ell$ comes from ℓ two generations back.
Case 1. If $2\ell-1 \equiv 3 \pmod 4$, then ℓ is even, and positive. This makes $1-\ell$ a negative odd number. We have shifted our problem back by two steps (i.e., generations) for each number in question, and we are now moving from a negative odd number $1-\ell$ to a positive even number ℓ. By Theorem 2.4, the generation index decreases by 1. Therefore by moving from a positive odd number $2\ell-1$ to a negative odd number $1-2\ell$, the generation index also decreases by 1.
Case 2. If $2\ell-1 \equiv 1 \pmod 4$, then ℓ is odd, and positive. This makes $1-\ell$ a negative even number. We have shifted our problem back by two steps (i.e., generations) for each number in question, and we are now moving from a negative even number $1-\ell$ to a positive odd number ℓ. By Theorem 2.5, the generation index increases by 1. Therefore by moving from a positive odd number $2\ell-1$ to a negative odd number $1-2\ell$, the generation index also increases by 1. □

Theorem 2.7. *When moving from a positive even number* $2^j m$, *where* $j \geq 1$ *and* $m \geq 3$ *is an odd number, to the negative even number* $-2^j m$ *in Table 1, the generation index increases by 1 if* $m \equiv 1 \pmod 4$ *and decreases by 1 if* $m \equiv 3 \pmod 4$.

Proof. By Theorem 2.1, $2^j m$ comes from $2^{j-1}m$ one generation back, and similarly $-2^j m$ comes from $-2^{j-1}m$ one generation back. Therefore the difference in generation indices when we move from $2^{j-1}m$ to $-2^{j-1}m$ is the same as that when we move from $2^j m$ to $-2^j m$. We can continue dividing each term by 2, and moving back one generation, without changing the difference in generation indices between the two numbers. We do this j times until we reach the odd numbers m and $-m$. This is a familiar situation; by Theorem 2.6, the generation index increases by 1 if $m \equiv 1 \pmod 4$ and decreases by 1 if $m \equiv 3 \pmod 4$. Therefore, this is also the difference in indices we see when moving from $2^j m$ to $-2^j m$. □

We come to the last remaining scenario when dealing with the numbers in Table 1.

Theorem 2.8. *When moving from a positive power of two, 2^j, where $j \geq 0$, to the negative number -2^j in Table 1, the generation index increases by 2.*

Proof. We prove this using a simple induction. For the initial case, $j = 0$, we see that in Table 1, the number 1 first occurs in G_1 and the number -1 first occurs in G_3. The generation index increases by 2, as required. Now assume this result is true when we move from 2^{j-1} to -2^{j-1}. By Theorem 2.1, 2^j and -2^j occur one generation after 2^{j-1} and -2^{j-1}, respectively, because we are doubling. This preserves the difference in generation indices between 2^{j-1} and -2^{j-1}, and so moving from 2^j to -2^j means the generation index increases by 2 as well, completing the induction. □

3. An Expression for the Generation Index

Now that we have established the patterns found in Table 1, we will use them to achieve our goal of finding an expression for the generation index for any integer. Recall the sequence of generation indices found in Table 1, namely $i = 0, 1, 3, 2, 4, 5, 4, 3, 5, 6, 7, 6, \ldots$. We will denote this sequence by $f(n)$ for $n \geq 0$. Let $k \in \mathbb{Z}$, as found in columns $1, 3, 5, 7$ of Table 1. Then $n = 2k - 1$ if $k > 0$ and $n = -2k$ if $k \leq 0$, so that an integer k is found in the $f(n)$th generation.

Let us now consider the difference sequence of $f(n)$, which we will denote $f_d(n)$ for $n \geq 1$:

$$f_d(n) = f(n) - f(n-1). \tag{3.1}$$

This sequence is given in Table 2, and is read column-wise. Of course, taking partial sums of $f_d(n)$ will get us back to $f(n)$.

Example 3.1. *From Table 1 we see that $k = -5$ appears in G_7. For $k = -5$, $n = -2k = 10$ so $f(10) = 7$, and as expected, the tenth partial sum of $f_d(n)$ is 7.*

Table 2. $f_d(n)$: the difference sequence of $f(n)$.

0	2	2	-1	2	1	-1	-1
1	1	1	1	1	1	1	1
2	-1	1	-1	1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1

Recall that our goal is to find an expression for the sequence $f(n)$. Neither this sequence, nor the difference sequence $f_d(n)$ is found in the Online Encyclopedia of Integer Sequences

(OEIS); however, similar sequences are. If the powers of two are divided out of $f_d(n)$, we obtain the sequence in Table 3, which we will denote $a_d(n)$ for $n \geq 1$:

$$a_d(n) = \begin{cases} \dfrac{f_d(n)}{2}, & n = 2^k,\ k \geq 1 \\ f_d(n), & n \neq 2^k,\ k \geq 1. \end{cases}$$

We are ultimately interested in the partial sums of this sequence, so we define it as a difference sequence, analogous to $f_d(n)$.

TABLE 3. $a_d(n)$: $f_d(n)$ with powers of 2 divided out.

0	1	1	-1	1	1	-1	-1
1	1	1	1	1	1	1	1
1	-1	1	-1	1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1

The sequence $a_d(n)$ matches [2, A034947], which is the Jacobi symbol (-1/n). To be sure of this, consider the recursion given in [2, A034947]:

$$\alpha(4n+3) = -1, \qquad n \geq 0; \tag{3.2}$$

$$\alpha(4n+1) = 1, \qquad n \geq 0; \tag{3.3}$$

$$\alpha(2n) = \alpha(n), \qquad n \geq 1. \tag{3.4}$$

Theorem 3.2. *The sequences $a_d(n)$ and $f_d(n)$ follow the recursion given in equations (3.2), (3.3), and (3.4), with the exception that equation (3.4) holds for $f_d(n)$ for $n \geq 2$.*

Proof. Equations (3.2) and (3.3) follow from Theorems 2.4 and 2.5, respectively. Theorem 2.4 tells us that when moving from a negative odd number $-k$, for $k \geq 1$, to a positive even number $k+1$, the generation index decreases by 1. In terms of n, this means moving from $2k$ to $2k+1$. The change in generation indices here is denoted by $f_d(2k+1)$, as given by equation (3.1). Because k is odd, $n = 2k+1 \equiv 3 \pmod 4$, and so $f_d(n) = -1$, for $n \equiv 3 \pmod 4$, matching equation (3.2). We can similarly obtain equation (3.3) from Theorem 2.5, except now k is even, implying $n = 2k+1 \equiv 1 \pmod 4$, and the generation index increases by 1.

Equation (3.4) is the result of Theorems 2.6, 2.7, and 2.8. To start, Theorem 2.8 implies that for n a power of 2, $f_d(n) = 2$, and $a_d(n) = 1$, both in adherence to equation (3.4). Note that this is the only case of the proof where $f_d(n)$ and $a_d(n)$ differ, so we need not consider both sequences again.

The proof of Theorem 2.7 tells us that for positive even values of k (which are not powers of 2), we can divide out powers of 2 without changing the difference in generation indices between k and $-k$. In terms of n, this means moving from $2k-1$ to $2k$, and so $f_d(n) = f_d(n/2)$, as required. In this case $n \equiv 0 \pmod 4$ because k is even.

Lastly, the proof of Theorem 2.6 tells us something similar; for positive odd values of k, $k \geq 3$, the difference in generation indices between k and $-k$ doesn't change when we move back two steps in the tree. The term $-k$ comes from $(1+k)/2$, two generations back. In terms of n, this means $2k$ comes from k two generations back, (since, $(1+k)/2$ is positive, its corresponding n value is $2[(1+k)/2]-1 = k$) and so $f_d(n) = f_d(n/2)$, as given in equation (3.4). In this case $n \equiv 2 \pmod 4$ because k is odd. Finally, for $n = 1$, we have $f_d(2) \neq f_d(1) = 1$. □

If we replace -1 by 0 in $a_d(n)$, we obtain [2, A014577], the regular paper-folding sequence, also known as the dragon curve sequence. This sequence is fractal in nature, and the fractal properties also exist in $f_d(n)$, as demonstrated by the previous recurrence.

Let the partial sums of sequence $a_d(n)$ be denoted by $a(n)$ for $n \geq 1$, and let $a(0) = 0$; see Table 4. In other words $a_d(n)$ is the difference sequence of $a(n)$:

$$a_d(n) = a(n) - a(n-1).$$

The sequence $a(n)$ is [2, A005811], and is the number of runs in the binary expansion of n, which is the number of 1's in the Gray code of n. This was a critical observation, as this allowed us to define an expression for $f(n)$, the function that will output the generation index for an integer k.

TABLE 4. $a(n)$: The number of 1's in the Gray code of n.

k	n	$a(n)$
0	0	0
1	1	1
-1	2	2
2	3	1
-2	4	2
3	5	3
-3	6	2
4	7	1
-4	8	2
5	9	3
-5	10	4

The sequence $a(n)$ almost gives us $f(n)$. Recall, we removed the powers of 2 from $f_d(n)$ to get $a_d(n)$, so we need to add those powers of 2 back in by adding $\lfloor \log_2(n) \rfloor$. This gives the following result about $f(n)$; see also Table 5.

Theorem 3.3. *Let k be an integer,*

$$n = \begin{cases} 2k-1, & k > 0; \\ -2k, & k \leq 0, \end{cases}$$

and $a(n)$ represent the number of 1's in the Gray code of n. Then $f(0) = 0$, and for $n \geq 1$,

$$f(n) = a(n) + \lfloor \log_2(n) \rfloor, \tag{3.5}$$

where $f(n)$ is the generation index for k.

We conclude this section with a complete recurrence for $f(n)$.

Theorem 3.4. *For $n \geq 1$, the sequence $f(n)$ can be expressed recursively as follows:*

$$f(2n) = f(n) + \begin{cases} 1, & n \text{ even}; \\ 2, & n \text{ odd}. \end{cases}$$

$$f(2n+1) = f(n) + \begin{cases} 1, & n \text{ odd}; \\ 2, & n \text{ even}. \end{cases}$$

Proof. Let us start with the even-indexed case. Consider $f(2n)$, where n is even. Here, $k = -n/2$ appears in generation $f(n)$, and $-n$ appears in generation $f(2n)$. To get from $-n/2$ to $-n$ in the binary tree, we apply Operation 1, doubling. Therefore $f(2n) = f(n) + 1$.

TABLE 5. $f(n)$: The generation index for k.

k	n	$f(n)$
0	0	0
1	1	1
-1	2	3
2	3	2
-2	4	4
3	5	5
-3	6	4
4	7	3
-4	8	5
5	9	6
-5	10	7

Consider $f(2n)$, where n is odd. Here, $n = 2k - 1$ so $k = (n + 1)/2$ appears in generation $f(n)$, and $-n$ appears in generation $f(2n)$. To get from $(n + 1)/2$ to $-n$ in the binary tree, we apply Operation 1, then Operation 2. Therefore, $f(2n) = f(n) + 2$.

Now, let us consider the odd-indexed case. Consider $f(2n + 1)$, where n is even. Here, $k = -n/2$ appears in generation $f(n)$, and $n + 1$ appears in generation $f(2n + 1)$. To get from $-n/2$ to $n + 1$ in the binary tree, we apply Operation 1, then Operation 2. Therefore, $f(2n+1) = f(n)+2$. Consider $f(2n+1)$, where n is odd. Here $n = 2k-1$ and so $k = (n+1)/2$ appears in generation $f(n)$, and $n+1$ appears in generation $f(2n+1)$. To get from $(n+1)/2$ to $n+1$ in the binary tree, we apply Operation 1. Therefore, $f(2n + 1) = f(n) + 1$. $\square$

In theory, we could have used this recurrence to generate our sequence $f(n)$, instead of using equation (3.5). Keep in mind however, that we are ultimately only interested in terms $f(n)$ for which k is a Fibonacci number or its negative. Using the recurrence would require us to calculate $f(n)$ for every value of n up to the point required by our counterexample, which would be extensive, as will be seen in the next section.

4. The Counterexample

We are now able to provide a counterexample for the proposition Problem 1.1. We wrote a procedure in Maple using equation (3.5) to determine that the first generation which does not have property $\mathcal{P}$ for the Fibonacci sequence is 43. In fact, the first several generations which do not have property $\mathcal{P}$ are as follows: $43, 47, 53, 61, 66, 67, 73, 82, 107, 108, 124, 143, 150, \ldots$. Some details of the calculation follow.

The sequence $a(n)$ is neither increasing nor decreasing, although $\lfloor \log_2(n) \rfloor$ is non-decreasing so it can be used as a lower bound for $f(n)$. If we let k be F_{63}, i.e., $k = 6, 557, 470, 319, 842$, the corresponding n is $n = 13, 114, 940, 639, 683$. Then $\lfloor \log_2(n) \rfloor = 43$ and $a(n) = 19$. We computed the generation indices $f(n)$ for the first 63 Fibonacci numbers and their negatives and found that none of these occur in G_{43}. If we let $k = F_{64}$ then $\lfloor \log_2(n) \rfloor = 44$. This means that no other Fibonacci number is in G_{43} since they must occur in a generation with index at least 44. This is proof that property $\mathcal{P}$ does not hold for G_{43}, for the Fibonacci sequence. The generation indices of the first 64 Fibonacci numbers and their negatives can be found in Table 6 (read horizontally).

Table 6. The generation indices for the first 64 Fibonacci numbers and their negatives.

1	3	1	3	2	4	5	4	6	7	4	6
7	8	10	11	9	10	11	10	12	13	11	12
13	14	14	15	15	16	17	16	16	17	19	18
22	23	22	23	23	24	26	25	20	21	25	26
28	29	26	27	25	26	32	31	30	31	31	32
36	37	33	34	31	32	36	35	37	38	35	34
36	37	41	42	39	40	42	41	45	46	35	34
48	49	49	50	44	45	50	49	49	50	48	49
48	49	51	52	54	55	52	51	55	56	58	59
57	58	63	64	64	65	65	64	59	60	62	63
69	70	63	64	62	63	68	69				

G_{43} contains $1,836,311,903$ integers, so to find this counterexample using brute force, or the recurrence in Theorem 3.4, would have been computationally difficult, whereas using the Maple procedure based on the patterns described in this paper, it took seconds to compute.

5. Conclusions

We saw that the proposition in Problem 1.1 is false, with generation 43 as a counterexample. A natural question to then ask is: how often do generations fail to have property $\mathcal{P}$?

We computed the generation indices for the first 5000 Fibonacci numbers and their negatives. These Fibonacci numbers occur within the first 3471 generations. Maple computations show that approximately 14.66% (509) of the first 3741 generations fail to have property $\mathcal{P}$. Looking at the first 4161 generations, we see that 610 of them do not have property $\mathcal{P}$, which is again approximately 14.66%. Extending this to the first 4865 generations, 718 of them, or about 14.79%, do not have property $\mathcal{P}$. This is surprising, as we anticipated the failure rate to approach zero as the generations grow larger in size.

We also looked at generalizing $\mathcal{S}$ to a Fibonacci-like sequence, with starting terms $1, a$, where $a > 1$. Clearly if a is large enough, property $\mathcal{P}$ will fail to hold true for early generations. With various values of a, ranging from 1 to 12, we found that between approximately 13% and 15% of the generations failed to have property $\mathcal{P}$. The failure rate of these sequences warrant further investigation.

References

[1] C. Kimberling, *Problem proposals*, Proceedings of the Sixteenth International Conference on Fibonacci Numbers and Their Applications, P. Anderson, C. Ballot, and W. Webb, eds., The Fibonacci Quarterly, **52.5** (2014), 5–14.

[2] OEIS Foundation Inc. (2011), The On-Line Encyclopedia of Integer Sequences, `http://oeis.org`.

MSC2010: 11B39, 05C05, 11Y99

Department of Mathematics, Mount Saint Vincent University, Halifax, Nova Scotia, B3M 2J6, Canada

E-mail address: danielle.cox@msvu.ca

Department of Mathematics, Mount Saint Vincent University, Halifax, Nova Scotia, B3M 2J6, Canada

E-mail address: karyn.mclellan@msvu.ca