

Fall 2012

# Integer Compositions, Gray Code, and the Fibonacci Sequence

Linus Lindroos  
*Georgia Southern University*

Follow this and additional works at: <https://digitalcommons.georgiasouthern.edu/etd>



Part of the [Mathematics Commons](#)

---

## Recommended Citation

Lindroos, Linus, "Integer Compositions, Gray Code, and the Fibonacci Sequence" (2012). *Electronic Theses and Dissertations*. 13.  
<https://digitalcommons.georgiasouthern.edu/etd/13>

This thesis (open access) is brought to you for free and open access by the Graduate Studies, Jack N. Averitt College of at Digital Commons@Georgia Southern. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons@Georgia Southern. For more information, please contact [digitalcommons@georgiasouthern.edu](mailto:digitalcommons@georgiasouthern.edu).

**INTEGER COMPOSITIONS, GRAY CODE, AND THE FIBONACCI  
SEQUENCE**

by

**LINUS J. LINDROOS**

(Under the Direction of Andrew Sills)

**ABSTRACT**

In this thesis I show the relation of binary and Gray Code to integer compositions and the Fibonacci sequence through the use of analytic combinatorics, Zeckendorf's Theorem, and generating functions.

*Index Words:* Number Theory, Integer Compositions, Gray Code, Mathematics

INTEGER COMPOSITIONS, GRAY CODE, AND THE FIBONACCI  
SEQUENCE

by

LINUS J. LINDROOS

B.S. in Mathematics

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial  
Fulfillment  
of the Requirement for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

2012

©2012

LINUS J. LINDROOS

All Rights Reserved

INTEGER COMPOSITIONS, GRAY CODE, AND THE FIBONACCI  
SEQUENCE

by

LINUS J. LINDROOS

Major Professor: Andrew Sills

Committee: David Stone  
Hua Wang

Electronic Version Approved:

December 2012

# TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	vii
CHAPTER	
1 Introduction – Preliminaries . . . . .	1
1.1 Integer Compositions . . . . .	1
1.2 Binary Representations of Compositions . . . . .	2
1.3 Gray Code . . . . .	4
2 Generating Functions . . . . .	8
2.1 Generating functions . . . . .	8
2.2 Examples of Generating functions . . . . .	8
2.3 The Fibonacci Sequence . . . . .	10
3 Generating Functions of Selected Integer Compositions . . . . .	12
3.1 Generating function proofs . . . . .	12
4 Binary to Gray Code map and compositions . . . . .	17
4.1 Zeckendorf’s Theorem . . . . .	17
4.2 The Fibbinaries . . . . .	18
4.3 Binary and Gray Code Compositions . . . . .	20
4.4 Bijective Proof of Theorem 3.5 . . . . .	21
4.5 The occurrence of the Golden Ratio . . . . .	26
5 Conclusion . . . . .	28

REFERENCES . . . . . 29

## LIST OF TABLES

Table		Page
1.1	Composition Tails . . . . .	1
1.2	MacMahon Graphs and Compositions of 4 . . . . .	4
1.3	Binary . . . . .	5
1.4	Gray Code . . . . .	5
1.5	XOR Table . . . . .	6
1.6	Binary to Gray Code Conversion . . . . .	6
1.7	Counting to 8 in Binary and Gray Code . . . . .	7
4.1	Fibbinaries and Compositions of 1's and 2's . . . . .	20
4.2	Binary and Gray Code with Corresponding Compositions . . . . .	21
4.3	Highlighted Compositions Resulting from Fibbinary Numbers . . . . .	22



## CHAPTER 1

### INTRODUCTION – PRELIMINARIES

We start with basic definitions and concepts before continuing.

#### 1.1 Integer Compositions

*Definition 1.1.* A *composition*  $\mathcal{C}$  of an integer  $n$  is a vector in which the components are positive integers that sum to  $n$ . Each component of  $\mathcal{C}$  is called a *part* of  $\mathcal{C}$ .

For instance the compositions of 3 are  $(1, 1, 1)$ ,  $(1, 2)$ ,  $(2, 1)$ , and  $(3)$ .

*Definition 1.2.* Let  $(a_1, a_2, a_3, \dots, a_n)$  be a composition  $\mathcal{C}$ . The *tail* of  $\mathcal{C}$  is  $a_n$ . If  $a_n = 2m$  for some  $m \in \mathbb{Z}_+$ , then we say  $\mathcal{C}$  has an *even tail*. If  $a_n = 2m - 1$  for some  $m \in \mathbb{Z}_+$ , then we say that  $\mathcal{C}$  has an *odd tail*.

*Definition 1.3.* Let  $(a_1, a_2, a_3, \dots, a_{k-1}, a_k, a_{k+1}, \dots, a_n)$  be a composition  $\mathcal{C}$  with  $a_k = a_{k+1} = \dots = a_n$  and  $a_{k-1} \neq a_k$ . Then  $\mathcal{C}$  has a *repeating tail* of  $n - k + 1$   $a_n$ 's.

*Example 1.4.* If a composition  $\mathcal{C}$  is of the form  $(a_1, a_2, a_3, \dots, a_{k-1}, \overbrace{1, 1, \dots, 1}^n)$  with  $a_{k-1} \neq 1$ , we say that the composition has a *repeating tail* consisting of  $n$  1's.

Composition	Tail
$(3, 1, \mathbf{2})$	Even
$(3, 1, \mathbf{3})$	Odd
$(3, 1, \mathbf{2}, \mathbf{2})$	Two repeating 2's
$(3, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1})$	Four repeating 1's

Table 1.1: Composition Tails

## 1.2 Binary Representations of Compositions

Compositions of an integer  $n$  can be represented by binary strings of  $n - 1$  bits. To illustrate, let's look at the integer compositions of 4. We can represent the integer 4 with four dashes.

- - - -

By placing vertical lines between these dashes we can count each adjacent dash and use each vertical bar as a marker to indicate the next part in the composition.

Thus,

- - | - -

represents the composition  $(2, 2)$  and

- | - | - -

represents the composition  $(1, 1, 2)$ . Now replace the vertical bars in between the dashes with 0 and every other space between the dashes with a 1. So

- | - | - -

becomes

- 0 - 0 - 1 - .

Thus, 001 corresponds to  $(1, 1, 2)$ . Looking at another example we get

- - | - -

which becomes

- 1 - 0 - 1 - .

Thus, 101 corresponds to  $(2, 2)$ .

*Example 1.5.* To translate 1010 to a composition reverse the technique above to get

\_1\_0\_1\_0\_

Now remove the 1's and replace the 0's with vertical bars.

\_ \_ | \_ \_ | \_

So 1010 translates to the composition (2, 2, 1)

*Example 1.6.* Translating 0000 to a composition we get

\_0\_0\_0\_0\_

which goes to

\_ | \_ | \_ | \_ | \_

which translates to (1, 1, 1, 1, 1)

*Example 1.7.* Translating 1111 to a composition we get

\_1\_1\_1\_1\_

which translates to

- - - - -

which is simply the trivial composition- the integer itself: (5).

*Definition 1.8.* We shall call this dashed representation of a composition of  $n$  a *MacMahon graph*, and for the purposes of this thesis we shall just call it a *graph* in the upcoming tables [6].

By utilizing the Macmahon graphs we have a clear bijection between compositions of an integer  $n$  and binary strings of length  $n - 1$ , thus the following theorem is immediate.

**Theorem 1.9.** *For every positive integer  $n$  there are  $2^{n-1}$  compositions [5].*

The full list of compositions of 4 with binary representations are listed below.

Binary	Translation	Graph	Composition
000	.0_0_0_	-   -   -   -	(1, 1, 1, 1)
001	.0_0_1_	-   -   - -	(1, 1, 2)
010	.0_1_0_	-   - -   -	(1, 2, 1)
011	.0_1_1_	-   - - -	(1, 3)
100	.1_0_0_	- -   -   -	(2, 1, 1)
101	.1_0_1_	- -   - -	(2, 2)
110	.1_1_0_	- - -   -	(3, 1)
111	.1_1_1_	- - - -	(4)

Table 1.2: MacMahon Graphs and Compositions of 4

### 1.3 Gray Code

Gray Code was introduced by Bell Labs researcher Frank Gray in a 1947 patent application under the name ‘reflected binary code’. The name was derived from the fact that it “may be built up from the conventional binary code by a sort of reflection process” [1].

Much as the usual binary code provides a way to represent integers in base 2, Gray Code is a binary encoding method, but with an additional valuable property the Gray Code representations of two consecutive integers differ by only one bit. For instance, the integer representations of zero to three in binary is shown in table 1.3.

The integer representations of zero to three in Gray Code is shown in table 1.4.

Note that we can go back to the original state of ‘00’ with another 1 bit change. This makes the code cyclic in nature and forms a natural Hamiltonian path.

The method used in this thesis utilizes the exclusive or ‘XOR’ to change binary digits to Gray Code. This is illustrated by the following Matlab function, named

Binary	Bit Changes
00	Start
01	1 bit change
10	2 bit change
11	1 bit change

Table 1.3: Binary

Gray Code	Bit Changes
00	Start
01	1 bit change
11	1 bit change
10	1 bit change

Table 1.4: Gray Code

bi2gray [2] . The subroutine is as follows, where b is a given binary integer,

```
function g = bi2gray( b )

g(1) = b(1);

    for i = 2 : length(b),
        x = xor(str2num(b(i-1)), str2num(b(i)));
        g(i) = num2str(x);
    end

return;
```

First let's remind ourselves how the 'exclusive or' works.

0	XOR	0	→	0
1	XOR	0	→	1
0	XOR	1	→	1
1	XOR	1	→	0

Table 1.5: XOR Table

Going through the algorithm step by step, the first digit of the binary code is always the first digit in the corresponding Gray Code. Then we get the output of the second digit XOR'ed with the first. So if the first digit is 1 and second digit 0, we get a 1 for the second digit, if both first and second digit are 0 or 1, we just get a zero. This process is repeated until the end of the binary string is reached.

*Example 1.10.* To illustrate, let's use this algorithm to convert 1011 from binary to Gray Code.

Binary	1	0	1	1
Conversion	↓	0 XOR 1	1 XOR 0	1 XOR 1
Gray Code	1	1	1	0

Table 1.6: Binary to Gray Code Conversion

Here are the first eight integers starting at zero with their representations in both binary and Gray Code.

Integer	Binary	Gray Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Table 1.7: Counting to 8 in Binary and Gray Code

## CHAPTER 2

### GENERATING FUNCTIONS

#### 2.1 Generating functions

*Definition 2.1.* The generating function  $G(\{a_n\}_{n=0}^{\infty}; x)$  of a sequence  $\{a_n\}_{n=0}^{\infty}$  is

$$G(a_n; x) = G(\{a_n\}_{n=0}^{\infty}; x) = \sum_{n=0}^{\infty} a_n x^n.$$

If we have a generating function of the form

$$\sum_{n=0}^{\infty} a x^n$$

with  $a \in \mathbb{R}$ , then we can give a closed form representation of the generating function by using the well known formula for the sum of a geometric series:

$$G(a; x) = \sum_{n=0}^{\infty} a x^n = \frac{a}{1-x}.$$

When using generating functions for combinatorial analysis we do not need to worry about convergence conditions as  $x$  is used strictly as a formal variable. The generating function for the Fibonacci numbers will be calculated in Section 2.3.

#### 2.2 Examples of Generating functions

*Definition 2.2.* Let  $C_S(n)$  denote the number of compositions of  $n$  with parts in the set  $S$ .

*Definition 2.3.* Let  $C_S(n, m)$  denote the number of compositions of  $n$  with exactly  $m$  parts and all parts in the set  $S$ .

*Definition 2.4.* Let  $C_{S,t,k}(n)$  denote the number of compositions of  $n$  with all parts in the set  $S$  with a repeating tail of  $k$   $t$ 's. The third subscript is omitted when  $k = 1$ .

*Example 2.5.*  $C_{\mathbb{Z}_+}(n, 3)$  denotes the number of compositions of  $n$  into exactly 3 parts. The generating function for  $C_{\mathbb{Z}_+}(n, 3)$  is:



$$\begin{aligned}
& (x^1 + x^2 + x^3 + x^4 + \dots) \\
& \times (x^1 + x^2 + x^3 + x^4 + \dots) \\
& \times (x^1 + x^2 + x^3 + x^4 + \dots) \\
& = x^{1+1+1} + (x^{1+1+2} + x^{1+2+1} + x^{2+1+1}) \\
& \quad + (x^{1+1+3} + x^{1+3+1} + x^{3+1+1} + x^{1+2+2} + x^{2+1+2} + x^{2+2+1}) + \dots \\
& = x^3 + 3x^4 + 6x^5 + \dots
\end{aligned}$$

Multiply these series together and the coefficients indicate how many compositions consisting of 3 parts there are for each integer.

$$\begin{aligned}
& \left( x \sum_{n=0}^{\infty} x^n \right) \times \left( x \sum_{n=0}^{\infty} x^n \right) \times \left( x \sum_{n=0}^{\infty} x^n \right) \\
& = \left( \frac{x}{1-x} \right) \times \left( \frac{x}{1-x} \right) \times \left( \frac{x}{1-x} \right) \\
& = \left( \frac{x}{1-x} \right)^3 \\
& = \sum_{n=0}^{\infty} C_{\mathbb{Z}_+}(n, 3).
\end{aligned}$$

Let's look at another.

*Example 2.6.* Letting  $E$  denote the set of positive even integers,  $C_E(n, m)$  denotes the number of compositions of  $n$  into exactly  $m$  even parts only. The generating function for  $C_E(n, m)$  is:

$$\begin{aligned}
& m \text{ times } \left\{ \begin{array}{l} (x^2 + x^4 + x^6 + x^8 + \dots) \\ \times (x^2 + x^4 + x^6 + x^8 + \dots) \\ \times \dots \\ \times (x^2 + x^4 + x^6 + x^8 + \dots) \end{array} \right. \\
& = (x^2 + x^4 + x^6 + x^8 + \dots)^m = \left( \frac{x^2}{1-x^2} \right)^m
\end{aligned}$$

Thus the generating function for compositions into  $m$  even parts is

$$\begin{aligned}
 \sum_{m=0}^{\infty} \left( \frac{x^2}{1-x^2} \right)^m &= \frac{\frac{x^2}{1-x^2}}{1 - \frac{x^2}{1-x^2}} \\
 &= \frac{\frac{x^2}{1-x^2}}{1 - \frac{x^2}{1-x^2}} \times \frac{1-x^2}{1-x^2} \\
 &= \frac{x^2}{1-x^2-x^2} \\
 &= \frac{x^2}{1-2x^2} \\
 &= \sum_{n=0}^{\infty} C_E(n, m)x^n.
 \end{aligned}$$

### 2.3 The Fibonacci Sequence

The *Fibonacci numbers* are a sequence of numbers named after Leonardo Fibonacci, though it was known by Indian Mathematicians as early as the 6th century. Fibonacci's *Liber Abaci* introduced it to the west in 1202 and is why the sequence bears his name.

The *Fibonacci numbers* are defined by  $F_0 = 0$  and  $F_1 = 1$ , with  $F_{n+1} = F_n + F_{n-1}$  for  $n \geq 1$ . So the sequence starts: 0, 1, 1, 2, 3, 5, 8, 13, ....

The generating function for the Fibonacci numbers is calculated as follows.

$$\mathcal{F}(x) = G(F_n; x) = \sum_{n=1}^{\infty} F_n x^n = 0 + 1x + 1x^2 + 2x^3 + 3x^4 + 5x^5 + 8x^6 + \dots$$

Multiply  $F_{n+1} = F_n + F_{n-1}$  by  $x^n$  and obtain

$$F_{n+1}x^n = F_n x^n + F_{n-1}x^n.$$

Now sum each from  $n = 1$  to  $\infty$  to obtain

$$\sum_{n=1}^{\infty} F_{n+1}x^n = \sum_{n=1}^{\infty} F_n x^n + \sum_{n=1}^{\infty} F_{n-1}x^n.$$

In order to get the generating function we must put all of these sums in terms of  $\mathcal{F}(x)$  so a little algebra is needed:

$$\frac{1}{x} \left( \sum_{n=1}^{\infty} F_n x^n - 1x \right) = \sum_{n=1}^{\infty} F_n x^n + x \sum_{n=1}^{\infty} F_n x^n.$$

Substituting in  $\mathcal{F}(x)$  we get

$$\frac{1}{x} (\mathcal{F}(x) - x) = \mathcal{F}(x) + x\mathcal{F}(x),$$

so

$$\frac{\mathcal{F}(x) - x}{x} = \mathcal{F}(x) + x\mathcal{F}(x).$$

Solve for  $\mathcal{F}(x)$  :

$$\mathcal{F}(x) - x = x(\mathcal{F}(x) + x\mathcal{F}(x))$$

$$\mathcal{F}(x) - x = x\mathcal{F}(x) + x^2\mathcal{F}(x)$$

$$\mathcal{F}(x) - x\mathcal{F}(x) - x^2\mathcal{F}(x) = x$$

$$\mathcal{F}(x)(1 - x - x^2) = x$$

$$\mathcal{F}(x) = \frac{x}{1 - x - x^2} = G(F_n; x).$$

So the generating function for the Fibonacci sequence is

$$\sum_{n=0}^{\infty} F_n(x) = \frac{x}{1 - x - x^2}. \quad (2.1)$$

**CHAPTER 3**  
**GENERATING FUNCTIONS OF SELECTED INTEGER**  
**COMPOSITIONS**

**3.1 Generating function proofs**

**Theorem 3.1.** *The number of compositions of  $n$  consisting of only 1's and 2's equals the  $(n + 1)^{st}$  Fibonacci number [4].*

*Proof.* Let  $C_{\{1,2\}}(n)$  denote the number of compositions of  $n$  into parts consisting of only 1's and 2's. The generating function for compositions into 1's and 2's with exactly  $m$  parts is

$$(x^1 + x^2)^m.$$

Therefore,

$$\begin{aligned} \sum_{n=1}^{\infty} C_{\{1,2\}}(n)x^n &= \sum_{n=1}^{\infty} (x + x^2)^n = \frac{1}{1 - x - x^2} \\ &= \sum_{n=1}^{\infty} F_n x^{n-1} \\ &= \sum_{n=0}^{\infty} F_{n+1} x^n. \end{aligned}$$

□

**Theorem 3.2.** *The number of compositions of  $n$  consisting of only 1's and 2's with a repeating tail of  $k$  ones equals the  $(n - k + 1)^{st}$  Fibonacci number.*

*Proof.* Let  $C_{\{1,2\},1,k}(n)$ , denote the number of compositions of  $n$  into parts consisting of only 1's and 2's with a repeating tail of  $k$  1's. The generating function for compositions into 1's and 2's with a repeating tail of  $k$  1's with exactly  $m$  parts is

$$(x^1 + x^2)^{m-k} \times x^k.$$

Therefore,

$$\begin{aligned}
\sum_{n=k}^{\infty} C_{\{1,2\},1,k}(n)x^k &= \sum_{n=k}^{\infty} (x+x^2)^{n-k} x^k = \sum_{n=k-(k-1)}^{\infty} (x+x^2)^{n+(k-1)-k} x^k \\
&= \sum_{n=1}^{\infty} (x+x^2)^{n-1} x^k \\
&= \frac{x^k}{1-x-x^2} \\
&= x^{k-1} \frac{x}{1-x-x^2} \\
&= x^{k-1} \sum_{n=0}^{\infty} F_n x^n \\
&= \sum_{n=0}^{\infty} F_n x^{n+k-1} \\
&= \sum_{n=k-1}^{\infty} F_{n-k+1} x^n.
\end{aligned}$$

□

**Theorem 3.3.** *The number of compositions of  $n$  consisting of only 1's and 2's with a repeating tail of  $k$  twos equals the  $(n-2k+1)^{st}$  Fibonacci number.*

*Proof.* Let  $C_{\{1,2\},2,k}(n)$ , denote the number of compositions of  $n$  into parts consisting of only 1's and 2's with a repeating tail of  $k$  2's. The generating function for compositions into 1's and 2's with a repeating tail of  $k$  2's with exactly  $m$  parts is

$$(x^1 + x^2)^{m-k} \times (x^2)^k.$$

Therefore,

$$\begin{aligned}
\sum_{n=k}^{\infty} C_{\{1,2\},2,k}(n)x^k &= \sum_{n=k}^{\infty} (x+x^2)^{n-k} x^{2k} = \sum_{n=k-(k-1)}^{\infty} (x+x^2)^{n+(k-1)-k} x^{2k} \\
&= \sum_{n=1}^{\infty} (x+x^2)^{n-1} x^{2k} \\
&= \frac{x^{2k}}{1-x-x^2} \\
&= x^{2k-1} \frac{x}{1-x-x^2} \\
&= x^{2k-1} \sum_{n=0}^{\infty} F_n x^n \\
&= \sum_{n=0}^{\infty} F_n x^{n+2k-1} \\
&= \sum_{n=2k-1}^{\infty} F_{n-2k+1} x^n.
\end{aligned}$$

□

**Theorem 3.4.** *The number of compositions of  $n$  consisting of only odd parts equals the  $n^{\text{th}}$  Fibonacci number [4].*

*Proof.* Letting  $O$  denote the set of positive odd integers,  $C_O(n)$  denotes the number of compositions of  $n$  into odd parts. Note that the generating function for compositions of  $n$  into exactly  $m$  odd parts is:

$$(x^1 + x^3 + x^5 + x^7 + \dots)^m = \left( \frac{x}{1-x^2} \right)^m$$

Therefore,

$$\begin{aligned}
\sum_{n=1}^{\infty} C_O(n)x^n &= \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} C_O(n,m)x^n = \sum_{m=1}^{\infty} \left( \frac{x}{1-x^2} \right)^m = \frac{\frac{x}{1-x^2}}{1-\frac{x}{1-x^2}} \times \frac{1-x^2}{1-x^2} \\
&= \frac{x}{1-x-x^2} \\
&= \sum_{n=0}^{\infty} F_n x^n.
\end{aligned}$$

□

**Theorem 3.5.** *The number of compositions of  $n$  consisting of only odd parts with an even tail equals the  $(n - 1)^{st}$  Fibonacci number.*

*Proof.* Let  $C_{\{O\},E}(n)$  where  $E = (2a)$  with  $a \in \mathbb{Z}_+$  denote the number of compositions of  $n$  into odd parts with an even tail. Note that the generating function for compositions of  $n$  into exactly  $m$  odd parts with an even tail is:

$$(x^1 + x^3 + x^5 + x^7 + \dots)^{m-1} \times (x^2 + x^4 + x^6 + \dots) = \left( \frac{x}{1-x^2} \right)^{m-1} \times \frac{1}{1-x^2}$$

Therefore,

$$\begin{aligned} \sum_{n=1}^{\infty} C_{\{O\},E}(n)x^n &= \sum_{n=1}^{\infty} \left( \frac{x}{1-x^2} \right)^{n-1} \times \frac{1}{1-x^2} = \frac{1}{1-x^2} \times \frac{1}{1-x^2} \\ &= \frac{1}{1-x^2} \times \frac{1-x^2}{1-x^2} \\ &= \frac{1}{1-x-x^2} \\ &= \sum_{n=1}^{\infty} F_n x^{n-1} \\ &= \sum_{n=0}^{\infty} F_{n-1} x^n. \end{aligned}$$

□

**Theorem 3.6.** *The number of compositions of  $n$  consisting of only 1's and 2's equals the number of compositions of  $n$  into odd parts plus the number of compositions of  $n$  into odd parts with an even tail.*

*Proof.* The result is immediate we get  $C_{\{1,2\}}(n) = F_{n+1}$  from Theorem 3.1,  $C_{\{O\}}(n) = F_n$  from Theorem 3.4 and  $C_{\{O\},E}(n) = F_{n-1}$  from Theorem 3.5. Therefore,

$$C_{1,2}(n) = C_{\{O\}}(n) + C_{\{O\},E}(n)$$

is equivalent to:

$$F_{n+1} = F_n + F_{n-1}.$$



A combinatorial proof of this fact will be presented in the following chapter.



## CHAPTER 4

### BINARY TO GRAY CODE MAP AND COMPOSITIONS

#### 4.1 Zeckendorf's Theorem

**Theorem 4.1.** (Zeckendorf) *Every positive integer can be represented uniquely as the sum of one or more distinct Fibonacci numbers in such a way that the sum does not include any two consecutive Fibonacci numbers. More precisely, if  $N$  is any positive integer, there exist positive integers  $c_1, c_2, \dots, c_k$  for some  $k$  such that  $c_i \geq 2$ , with  $c_{i+1} > c_i + 1$ , such that*

$$N = \sum_{i=0}^k F_{c_i} \quad (4.1)$$

where  $F_n$  is the  $n^{\text{th}}$  Fibonacci number [3].

This is known as Zeckendorf's theorem, named after Belgian mathematician Edouard Zeckendorf. This finite sum (Equation 4.1) is called the *Zeckendorf representation* of  $N$ .

*Example 4.2.* The Zeckendorf representation of 10 is:

$$10 = 8 + 2 = F_6 + F_3.$$

*Example 4.3.* The Zeckendorf representation of 80 is:

$$80 = 55 + 21 + 3 + 1 = F_{10} + F_8 + F_4 + F_2.$$

There is more than one way of representing 80 as the sum of Fibonacci numbers, for example:

$$80 = 34 + 21 + 3 + 1,$$

$$80 = 55 + 13 + 8 + 3 + 1.$$

These are not Zeckendorf representations since 34 and 21 are consecutive Fibonacci numbers ( $F_9$  and  $F_8$ ), as are 13 and 8.

## 4.2 The Fibbinaries

From Zeckendorf's Theorem we can write any positive integer  $N$  uniquely as

$$N = F_{n_1} + F_{n_2} + \cdots + F_{n_k}$$

where  $n_i - n_{i+1} \geq 2$  for  $i = 1, 2, \dots, k - 1$ .

*Definition 4.4.* The  $N^{\text{th}}$  Fibbinary number [7],  $fib(N)$ , is:

$$fib(N) = \sum_{m=1}^k 2^{n_m-2}.$$

So from example 4.2 above we get:

$$10 = 8 + 2 = F_6 + F_3$$

so

$$fib(10) = 2^{6-2} + 2^{3-2} = 10010_2 = 18.$$

Note that we get the *Fibbinary Sequence* when we encode the positive integers with the Zeckendorf representation, change to base 2, and let 0 represent the null sum. The first few terms of the sequence are given by: (OEIS A003714)

$$1, 2, 4, 5, 8, 9, 10, 16, 17, 18, 20, 21, 32, 33, 34, 36, 37, 40, 41, 42, 64, 65, \dots$$

Also, since the Zeckendorf representation of a number does not allow consecutive Fibonacci numbers, the binary representation of Fibbinary numbers contain no consecutive 1's. Calculating  $fib(N)$  for  $N = 1, 2, 3, 4, 5, 6, 7$  results in:

$$1 = F_2 \rightarrow 2^{2-2} = 2^0 = 01_2$$

$$2 = F_3 \rightarrow 2^{3-2} = 2^1 = 10_2$$

$$3 = F_4 \rightarrow 2^{4-2} = 2^2 = 100_2$$

$$4 = F_4 + F_2 \rightarrow 2^{4-2} + 2^{2-2} = 2^2 + 2^0 = 101_2$$

$$5 = F_5 \rightarrow 2^{5-2} = 2^3 = 1000_2$$

$$6 = F_5 + F_2 \rightarrow 2^{5-2} + 2^{2-2} = 2^3 + 2^0 = 1001_2$$

$$7 = F_5 + F_3 \rightarrow 2^{5-2} + 2^{3-2} = 2^3 + 2^1 = 1010_2$$

In order to check if an integer  $n$  is a Fibbinary number, simply write the integer out in binary. Any consecutive 1's will indicate the integer is not a Fibbinary.

*Example 4.5.*

$$27 = 11011_2$$

27 is not a fibbinary number since there are 2 pairs of consecutive 1's.

$$21 = 10101_2$$

21 is a fibbinary number since there are no consecutive 1's.

In order to check where a Fibbinary number lies in the sequence of Fibbinary numbers, simply work out the Zeckendorf representation in reverse.

*Example 4.6.*

$$21 = 10101_2 = 2^{6-2} + 2^{4-2} + 2^{2-2} \rightarrow F_6 + F_4 + F_2 = 8 + 3 + 1 = 12$$

so

$$fib(12) \leftrightarrow 21.$$

The compositions of 5 can be represented by a binary string of length 4. The eight fibbinaries computed above map to all 8 compositions of 5 consisting of only 1's and 2's as shown in table 4.1. So the Fibbinaries with  $n$  bit binary representations map to the  $F_{n+1}$  compositions consisting of 1's and 2's of  $n$ . This result is immediate from Theorem 3.1.

k	fib(k)	Fibbinary Number	Translation	Graph	Composition
0	0	0000	._0._0._0._0_	-   -   -   -   -	(1, 1, 1, 1, 1)
1	1	0001	._0._0._0._1_	-   -   -   - -	(1, 1, 1, 2)
2	2	0010	._0._0._1._0_	-   -   - -   -	(1, 1, 2, 1)
3	4	0100	._0._1._0._0_	-   - -   -   -	(1, 2, 1, 1)
4	5	0101	._0._1._0._1_	-   - -   - -	(1, 2, 2)
5	8	1000	._1._0._0._0_	- -   -   -   -	(2, 1, 1, 1)
6	9	1001	._1._0._0._1_	- -   -   - -	(2, 1, 2)
7	10	1010	._1._0._1._0_	- -   - -   -	(2, 2, 1)

Table 4.1: Fibbinaries and Compositions of 1's and 2's

### 4.3 Binary and Gray Code Compositions

We can get a better picture of the relation between binary and Gray code by combining the first  $2^{n-1}$  nonnegative integers written in binary and Gray Code with the corresponding compositions of  $n$ . The compositions of 4 with each element listed is shown in Table 4.2.

With all these elements listed in this form we can get a better picture of the relations between each element. Moreover, for an integer  $n$ , we have clear bijection between the the set of integers from 0 to  $2^{n-1}$ , binary and Gray codes, along with the

Binary	Gray Code	Bin Comp	Gray Comp
000	000	(1, 1, 1, 1)	(1, 1, 1, 1)
001	001	(1, 1, 2)	(1, 1, 2)
010	011	(1, 2, 1)	(1, 3)
011	010	(1, 3)	(1, 2, 1)
100	110	(2, 1, 1)	(3, 1)
101	111	(2, 2)	(4)
110	101	(3, 1)	(2, 2)
111	100	(4)	(2, 1, 1)

Table 4.2: Binary and Gray Code with Corresponding Compositions

corresponding set of compositions of  $n$ .

#### 4.4 Bijective Proof of Theorem 3.5

If we take the list of compositions from above and highlight the Fibbinary numbers a clear pattern emerges (Table 4.3).

We see that the compositions corresponding to binary code are all the compositions of 4 consisting of 1's and 2's, with 5 compositions total, this is expected from Theorem 3.1. Also notice that the compositions corresponding to Gray Code are the 3 odd compositions of 4 which corresponds to Theorem 3.4 and the 2 remaining compositions are odd with an even tail by Theorem 3.5.

*Proof.* A Fibbinary number,  $fib(N)$ , may be even or odd. For  $fib(N)$  to be odd the units digit of the binary representation of  $fib(N)$  must be 1. The Fibbinary number is even otherwise. Let  $n, k, m \in \mathbb{Z}_+$ . We consider cases.

Integer	Binary	Gray Code	Bin Comp	Gray Comp
<b>0</b>	<b>000</b>	<b>000</b>	<b>(1,1,1,1)</b>	<b>(1,1,1,1)</b>
<b>1</b>	<b>001</b>	<b>001</b>	<b>(1,1,2)</b>	<b>(1,1,2)</b>
<b>2</b>	<b>010</b>	<b>011</b>	<b>(1,2,1)</b>	<b>(1,3)</b>
3	011	010	(1,3)	(1,2,1)
<b>4</b>	<b>100</b>	<b>110</b>	<b>(2,1,1)</b>	<b>(3,1)</b>
<b>5</b>	<b>101</b>	<b>111</b>	<b>(2,2)</b>	<b>(4)</b>
6	110	101	(3,1)	(2,2)
7	111	100	(4)	(2,1,1)

Table 4.3: Highlighted Compositions Resulting from Fibbinary Numbers

(a) Every odd Fibbinary number's binary representation will be of the form:

$$\dots 010 \dots 010 \dots 01.$$

Consider an odd Fibbinary consisting of alternating 1's and 0's with last digit

1. Its binary representation is thus:

$$\overbrace{10101 \dots 01}^{2k-1}. \quad (4.2)$$

Since it starts and ends with a 1, the number of bits must be odd, say  $2k - 1$  for some  $k$ . Clearly, this is an odd number of bits. Its Gray Code representation is a string of  $2k - 1$  1's, ie.:

$$\overbrace{111 \dots 1}^{2k-1}.$$

The composition corresponding to the Gray Code thus has a single part:

$$(2k - 1 + 1) = (2k).$$

$(2k)$  is an even number for all  $k$ , in other words a composition with a single even part with zero odd parts in front of it.

Fib. Number	Gray Code	Translation	Map	Composition
10101	11111	$\_1\_1\_1\_1\_1\_$	$\_ \_ \_ \_ \_ \_ \_$	(6)
$\overbrace{1010 \dots 101}^{2k-1}$	$\overbrace{111 \dots 1}^{2k-1}$	$\_1\_1\_ \dots \_1\_$	$\_ \_ \dots \_ \_ \_$	(2k)

Prepend a single 0 to equation 4.2:

$$0 \overbrace{10101 \dots 01}^{2k-1}.$$

Instead of a 1 for the first digit when we change to Gray Code we get a 0, so in the example above we would get:

$$0 \overbrace{111 \dots 1}^{2k-1}$$

which corresponds to the composition  $(1, 2k - 1 + 1) = (1, 2k)$ .

Fib. Number	Gray Code	Translation	Map	Composition
010101	011111	$\_0\_1\_1\_1\_1\_1\_$	$\_   \_ \_ \_ \_ \_ \_ \_$	(1,6)
01010...0101	0111...1	$\_1\_1\_ \dots \_1\_$	$\_   \_ \_ \dots \_ \_ \_$	(1,2k)

Prepending a string of zeros, ie.  $\overbrace{00 \dots 0}^n$  to 4.2 results in a binary string of the form

$$\overbrace{00 \dots 0}^n \overbrace{10101 \dots 01}^{2k-1}$$

Converting to Gray Code results in:

$$\overbrace{00 \dots 0}^n \overbrace{111 \dots 1}^{2k-1},$$

which corresponds to the composition:  $(\overbrace{1, 1, \dots, 1}^n, 2k)$ .

Prepending a binary string of the form

$$\overbrace{1010 \dots 01}^{2m-1}$$

to add an integer other than 1 to the beginning of the composition results in a binary string of the form

$$\overbrace{1010 \cdots 1}^{2m-1} \overbrace{00 \cdots 0}^n \overbrace{10101 \cdots 01}^{2k-1}, n \geq 2,$$

During the Gray Code conversion the first zero from the string of  $n$  zeroes will change to a 1 (since 1 XOR 0 results in a 1). Now regroup the former binary string to compensate for this fact:

$$\overbrace{1010 \cdots 10}^{2m} \overbrace{00 \cdots 0}^{n-1} \overbrace{10101 \cdots 01}^{2k-1},$$

we have shown that the tail of the composition must be even so we just need to worry about the new integer we tried to adjoin to the composition. When we convert to Gray Code we end up with

$$\overbrace{111 \cdots 1}^{2m} \overbrace{00 \cdots 0}^{n-1} \overbrace{111 \cdots 1}^{2k-1},$$

the first part of the composition corresponds to the integer  $(2m + 1)$ , an odd integer. Thus, the composition is:

$$(2m + 1, \overbrace{1, \cdots, 1}^{n-2}, 2k - 1 + 1) = (2m + 1, \overbrace{1, \cdots, 1}^{n-2}, 2k).$$

Fib. Number	Gray Code	Graph	Composition
$\overbrace{101}^{2m-1} \overbrace{00}^n \overbrace{101}^{2k-1}$	$\overbrace{1111}^{2m} \overbrace{0}^{n-1} \overbrace{111}^{2k-1}$	-----   -----	(5, 4)
101000101	111100111	-----   -   -----	(5, 1, 4)

Repeat the process above indefinitely and every odd Fibbinary number when converted to Gray Code will result in an odd composition with an even tail.



Fib. Number	Gray Code	Composition
$\overbrace{1}^{2m_1-1}$ $\overbrace{00}^{n_1}$ $\overbrace{101}^{2m_2-1}$ $\overbrace{00}^{n_2}$ $\overbrace{101}^{2k-1}$	$\overbrace{11}^{2m_1}$ $\overbrace{0}^{n_1-1}$ $\overbrace{1111}^{2m_2}$ $\overbrace{0}^{n_2-1}$ $\overbrace{111}^{2k-1}$	(3, 5, 4)
$\overbrace{1}^{2m_1-1}$ $\overbrace{000}^{n_1}$ $\overbrace{101}^{2m_2-1}$ $\overbrace{00}^{n_2}$ $\overbrace{101}^{2k-1}$	$\overbrace{11}^{2m_1}$ $\overbrace{00}^{n_1-1}$ $\overbrace{1111}^{2m_2}$ $\overbrace{0}^{n_2-1}$ $\overbrace{111}^{2k-1}$	(3, 1, 5, 4)
101000101	111100111	(5, 1, 4)

(b) Every even Fibbinary number will be of the form:

$$\dots 010 \dots 010 \dots 010 \dots 0.$$

Consider an even Fibbinary number of the form:

$$\overbrace{1010 \dots 1}^{2m-1} \overbrace{00 \dots 0}^n, n \geq 1.$$

Translating through Gray Code we get:

$$\overbrace{111 \dots 1}^{2m} \overbrace{00 \dots 0}^{n-1}.$$

Therefore we get the composition  $(2m + 1, \overbrace{1, \dots, 1}^{n-1})$ .

Fib. Number	Gray Code	Composition
1010	1111	(5)
$\overbrace{1010 \dots 1}^{2m-1} 0$	$\overbrace{111 \dots 1}^{2m}$	$(2m + 1)$
$\overbrace{1010 \dots 1}^{2m-1} \overbrace{0 \dots 0}^n$	$\overbrace{111 \dots 1}^{2m} \overbrace{0 \dots 0}^{n-1}$	$(2m + 1, \overbrace{1, \dots, 1}^{n-1})$

Repeat the process for adding parts of compositions as demonstrated above to see that every even Fibbinary number will translate to an odd composition.

Fib. Number	Gray Code	Composition
$\overbrace{1}^{2m_1-1}$ $\overbrace{00}^{n_1}$ $\overbrace{101}^{2m_2-1}$ $\overbrace{00}^{n_2}$	$\overbrace{11}^{2m_1}$ $\overbrace{0}^{n_1-1}$ $\overbrace{1111}^{2m_2}$ $\overbrace{0}^{n_2-1}$	(3, 5, 1)
$\overbrace{1}^{2m_1-1}$ $\overbrace{000}^{n_1}$ $\overbrace{101}^{2m_2-1}$ $\overbrace{000}^{n_2}$	$\overbrace{11}^{2m_1}$ $\overbrace{00}^{n_1-1}$ $\overbrace{1111}^{2m_2}$ $\overbrace{00}^{n_2-1}$	(3, 1, 5, 1, 1)
$\overbrace{1}^{2m_1-1}$ $\overbrace{0000}^{n_1}$ $\overbrace{101}^{2m_2-1}$ $\overbrace{000}^{n_2}$	$\overbrace{11}^{2m_1}$ $\overbrace{000}^{n_1-1}$ $\overbrace{1111}^{2m_2}$ $\overbrace{00}^{n_2-1}$	(3, 1, 1, 5, 1, 1)

Therefore the number of compositions of some integer  $M$  into 1's and 2's equals the number of compositions of  $M$  into compositions of all odd parts and compositions of all odd parts with an even tail. More specifically,

- (a) every composition of  $M$  into 1's and 2's with an odd tail equals the number of odd compositions of  $M$  with an even tail and,
- (b) every composition of  $M$  into 1's and 2's with an even tail equals the number of odd compositions of  $M$ .

□

#### 4.5 The occurrence of the Golden Ratio

Taking the Zeckendorf representation of the odd fibbinarys and letting  $\phi = \frac{1+\sqrt{5}}{2}$  we get:

$$5 = 101 = 2^2 + 2^0 \rightarrow F_{2+2} + F_{0+2} = 3 + 1 = 4 = \lfloor 2\phi^2 \rfloor - 1$$

$$9 = 1001 = 2^3 + 2^0 \rightarrow F_{3+2} + F_{0+2} = 5 + 1 = 6 = \lfloor 3\phi^2 \rfloor - 1$$

$$17 = 10001 = 2^4 + 2^0 \rightarrow F_{4+2} + F_{0+2} = 8 + 1 = 9 = \lfloor 4\phi^2 \rfloor - 1$$

$$21 = 10101 \rightarrow F_{4+2} + F_{2+2} + F_{0+2} = 8 + 3 + 1 = 12 = \lfloor 5\phi^2 \rfloor - 1$$

$$33 = 100001 \rightarrow F_{5+2} + F_{1+2} = 13 + 1 = 14 = \lfloor 6\phi^2 \rfloor - 1$$

When we list the odd Fibbinary's and work through the Zeckendorf representation method in reverse, it appears that we form the terms of the sequence:

$$1, 4, 6, 9, 12, 14, 17, 19, 22, 25, 27, 30, 33, 35, 38, 40, 43, \dots \text{ (OEIS A003622),}$$

which is defined as:  $\{\lfloor n\phi^2 \rfloor - 1\}_{n=1}^{\infty}$ ,  $\phi = \frac{1+\sqrt{5}}{2}$ .

*Conjecture 4.7.* The  $n^{\text{th}}$  odd fibbinary number maps to  $\lfloor n\phi^2 \rfloor - 1$  through the Zeckendorf representation.

$$fib(N)_n \rightarrow \{\lfloor (n)\phi^2 \rfloor - 1\}_{n=1}^{\infty},$$

where  $fib(N)_n =$  the  $n^{\text{th}}$  odd Fibbinary Number.

This remains an open question.

## CHAPTER 5

### CONCLUSION

It is doubtful that Frank Gray ever thought his “reflected binary code” would be used in the analysis of integer compositions, yet his code has produced an interesting bijection between certain sets of compositions. Also this relation made certain relations between sets of compositions and the Fibonacci numbers more clear. The occurrence of the Golden Ratio was completely unexpected and proving the relation between the odd Fibbinaries and the integer sequence formed by  $\{[n\phi^2] - 1\}_{n=1}^{\infty}$  is still an open question.

The natural Hamiltonian Path formed by Gray Code may have some interesting consequences when teamed up with Graph Theory. There appears to be a growing interest regarding Gray Code and its relation to integer compositions in various forms. This makes for an interesting fusion between computer science and pure mathematics. Computers have served a major role in the advancement of mathematics with use of programs such as Matlab, Maple, and Mathematica, just to name a few. However, computer code and logic at the most basic levels can have applications to higher mathematics, as has just been demonstrated.

## REFERENCES

- [1] F. Gray, *Pulse code communication*, March 17, 1953 (filed Nov. 1947). U.S. Patent 2,632,058
  
- [2] Adrian Bohdanowicz, *bi2gray*,  
<http://www.mathworks.com/matlabcentral/fileexchange/1743-bi2gray>.
  
- [3] Zeckendorf representation. G.M. Phillips (originator), Encyclopedia of Mathematics. URL:  
[http://www.encyclopediaofmath.org/index.php?title=Zeckendorf\\_representation](http://www.encyclopediaofmath.org/index.php?title=Zeckendorf_representation)
  
- [4] R. Stanley, *Enumerative Combinatorics*, Vol. 1, Cambridge University Press, Cambridge, England, 1997.
  
- [5] Silvia Heubach, Toufik Mansour, *Combinatorics of Compositions and Words*, CRC Press, 2009, ISBN 978-1-4200-7267-9.
  
- [6] A. Sills, *Compositions, Partitions, and Fibonacci Numbers*, The Fibonacci Quarterly, 2011, vol. 40.
  
- [7] C. Kimberling, *Affinely recursive sets and orderings of languages*, Discrete Math., 274 (2004), 147-160.