Clemson University
TigerPrints

All Theses | Theses

5-2019

The Calkin-Wilf Tree and Its Various Properties

Catherine Mary Kenyon
Clemson University, catherine1fff@gmail.com

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

Recommended Citation

Kenyon, Catherine Mary, "The Calkin-Wilf Tree and Its Various Properties" (2019). *All Theses*. 3075.
https://tigerprints.clemson.edu/all_theses/3075

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

The Calkin-Wilf Tree and Its Various Properties

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mathematical Science

---

by
Catherine Mary Kenyon
May 2019

Accepted by:
Dr. Neil Calkin, Committee Chair
Dr. William Bridges
Dr. Chris McMahan

Abstract

The Tree of All Fractions, otherwise known as the Calkin-Wilf tree, gets its name from the fact that every reduced positive rational appears once in this binary search tree. We look at the various properties this tree is known to hold and will explore a plethora of related, yet new, characteristics within the patterns of the tree.

One of the most interesting properties of the tree we will state is the numerators are the hyperbinary representation of that index in the sequence, as already observed by Calkin and Wilf. We will take look at this as well as the ordering of the numerators. We notice that some numbers in the sequence of numerators appeared "out of order," i.e. if $N+1$ appears before N in the sequence of numerators. We found these oddities and looked at the distance between these occurrences.

We also look closely at the "paths" of the rationals in the tree. The "path" of an element is the right and left steps it takes to reach said element. We let 0 represent a left step and 1 represent a right step. We explore various ways to change or look at the paths, as well as with matrices that represent the left and right steps.

We also question the claim, $\forall \epsilon > 0$ and $x \in \mathbb{R}^+$, there exists an n such that $\left| x - \frac{a_n}{a_{n+1}} \right| < \epsilon$. In other words, for any positive real number, there is a rational number that is incredibly close to the real number given. We use ideas from continued fraction approximations and computation to bring this claim to life. From this claim, we attempt to define a dsitrubtion over the real numbers.

Dedication

This Master's Thesis is dedicated in part to Herbert Wilf (June 13, 1931 - January 7, 2012). Without him and his mathematical creativity, this project would not have been possible.

Acknowledgments

I would like to first thank Dr. Calkin for his unfathomable patience during this research project. In addition, the other Mathematical Sciences Graduate Students here at Clemson have been extremely helpful in listening to me explain my research, helping me answer questions, and believing in me throughout this process. Their emotional support truly carried me through these past two years. A huge thank you is due to my partner, Trevor Squires, who continually listened to me talk about my research and improved many of my MATLAB codes. This thesis and degree wouldn't have been possible without my sisters, Elizabeth and Sarah, for continually supporting my endeavors. Most importantly, I would never have begun this journey without my parents, Thomas and Susan Kenyon; they have never given up on me and have always supported me- financially, emotionally, and spiritually- throughout my mathematical studies.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

There is a long history of labeled binary tree similar to the Calkin-Wilf tree; the oldest dating back to the Stern-Brocot tree discovered by Moritz Stern and Achille Brocot [7]. Then, a tree more similar to the Tree of All Fractions, the Raney tree came about. Finally, in 2000, Dr. Neil Calkin and Dr. Herbert Wilf produced the Tree of All Fractions, where we begin our research [5].

Unlike the Stern-Brocot tree, the Calkin-Wilf tree does not repeat any rational number. In fact, the Calkin-Wilf tree shows a one-to-one correspondence with the set of all positive rational numbers and the set of natural numbers. The way the Calkin-Wilf tree does this is it takes the "parent" rational, $\frac{a}{b}$ and it creates two "children" from this rational- the "left child", $\frac{a}{a+b}$, and the "right child", $\frac{a+b}{b}$, as so:

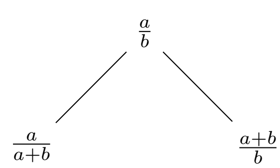

Figure 1.1: The Calkin-Wilf tree:a "parent" and its "child"

Calkin and Wilf explain how every positive rational appears only once in the tree in their initial paper, [2], by contradiction. One can also deduce that every rational appears once by considering building the tree. Of course, we must start at $\frac{1}{1}$, then using what we know about its children, build the tree going downwards. So, if we build the tree we see that a "left child" and the next rational in the tree is a "right child" share the same "parent" rational. So, for left children, we have that the numerator of the child is the same as the numerator of its parent and for right children, we have that their denominator is the same as the denominator of its parent. After listing a couple of rows of the Calkin-Wilf tree, it also becomes clear that the child to the leftmost position on each level has a numerator of 1 and the child to the rightmost position on each level has a denominator of 1, as seen below:

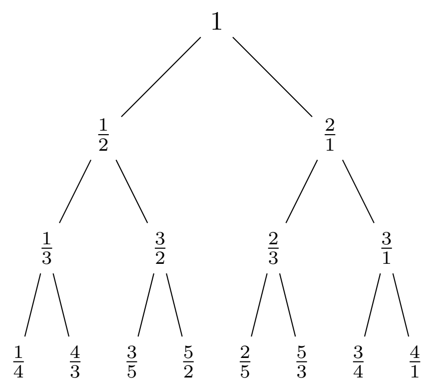

Figure 1.2: The first three levels of the Calkin-Wilf tree

The Calkin-Wilf tree has very useful properties that we took advantage of during our research, such as:

- Every reduced positive rational appears once, as mentioned above.
- The numerator and denominator at each vertex are relatively prime.
- The list of numerators is the number of hyperbinary representation of the corresponding index; we label nodes as $\frac{a_n}{a_{n+1}}$ where a_n is the number of hyperbinary representations of n.
- There is a concise matrix method to compute the labels of the tree; i.e. matrices to represent the "children" and build from there.
- The Fibonacci sequence appears along "proper zig-zags" down the tree.

Building on what Calkin and Wilf discovered in 1998, we explore more properties and patterns in this tree of rationals. We borrow ideas from combinatorics, probability, algebra, analysis, and computation to take our visions from ideas to conjectures to proofs.

Chapter 2

Calkin-Wilf Tree Basics

Our research project begins with creating simple, efficient codes to create the Calkin-Wilf tree. We were able to write code that did the following, which proved to be very useful later on during the project:

- create the Calkin-Wilf Tree
- create the list of numerators in the Calkin-Wilf Tree
- create the list of denominators in the Calkin-Wilf Tree
- given n, compute $\frac{a_n}{a_{n+1}}$
- given $\frac{a_n}{a_{n+1}}$, compute n

After writing these simple codes, we are able to explore many concepts of the Calkin-Wilf Tree. We begin with the basic ideas we wanted to check.

2.1 Reverse Paths and Palindromes

We first explore the consequences of "flipping" a path along the tree. The path to a rational $\frac{a}{b}$ in the tree is the sequence of left and right turns it takes to get to $\frac{a}{b}$. We assign 0 to represent a left turn and a 1 to represent a right turn.

Definition: A *reverse path* of a rational $\frac{a}{b}$ in the Tree of All Fractions is the sequence of 0's and 1's read in reverse order.

For example, if we want to get to the rational $\frac{3}{2}$, we would have to take a left turn from 1 and then a right turn from $\frac{1}{2}$:

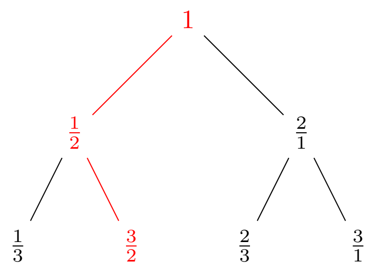

Figure 2.1: Example of a path in tree.

Therefore, the binary path to represent $\frac{3}{2}$ in the tree is 01. We discover if we reverse the path of the rational, instead of reached $\frac{a}{b}$ from the original path, we reach $\frac{b}{a}$. So, in our example, our reverse path would be 10; i.e. taking a right turn from 1, then a left turn from $\frac{2}{1}$. Therefore, we reach $\frac{2}{3}$. Our initial thought it that the reverse of a path to a certain rational gives us the reciprocal of the rational.

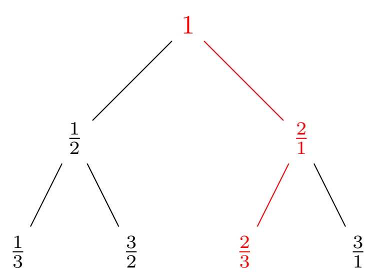

Figure 2.2: Example of a reverse path in tree.

However, it is not always the case that the reverse path gives the reciprocal of the rational. For example, if we start with the rational $\frac{4}{3}$, which has path 001, then reversing the path gives 100 which leads to the fraction $\frac{2}{5}$. One of our codes returns which rational you end up on when you reverse the path, which can be found in Appendix A. Table 2.1 shows a list of rationals and their reverse path rationals.

original fraction	path	reverse path	reverse path fraction
$1/1$	-	-	$1/1$
$1/2$	0	0	$1/2$
$2/1$	1	1	$2/1$
$1/3$	00	00	$1/3$
$3/2$	01	10	$2/3$
$2/3$	10	01	$3/2$
$3/1$	11	11	$3/1$
$1/4$	000	000	$1/4$
$4/3$	001	100	$2/5$
$3/5$	010	010	$3/5$
$5/2$	011	110	$3/4$
$2/5$	100	001	$4/3$
$5/3$	101	101	$5/3$
$3/4$	110	011	$5/2$
$4/1$	111	111	$4/1$
$1/5$	0000	0000	$1/5$
$5/4$	0001	1000	$2/7$
$4/7$	0010	0100	$3/8$
$7/3$	0011	1100	$3/7$
$3/8$	0100	0010	$4/7$

Table 2.1: The first few reverse path fraction pairs

Although interesting results didn't immediately come from our study in reverse paths, we were later able to deduce results about palindromic paths we see when looking back on our reserve paths. In the above table, the paths in blue are what we defined to be "palindromic paths", i.e. paths that are the same forward and backward.

Definition: A *palindromic path* in this context is a path that is equivalent if it is read the same forward as it is read backward.

For example, a palindromic path could be 010.

We also defined an "antipalindromic path", i.e. the path read in reverse is the inverse of the original path.

Definition: An *antipalindromic path* in this context occurs when the inverse of the path of $\frac{a}{b}$ is the same as the reverse path of $\frac{a}{b}$.

For example, an antipalindromic path would be 1100 since its reverse is 0011, which is the same as its inverse path, 0011. We found that on level n, where n is odd, and its subsequent level, level $n+1$, we see $2^{\frac{n+1}{2}}$ palindromes; i.e. on levels 3 and 4, we see $2^{\frac{4}{2}} = 2^2$ palindromes. Labeled palindromes (blue) in the tree gives a clearer representation of this result.

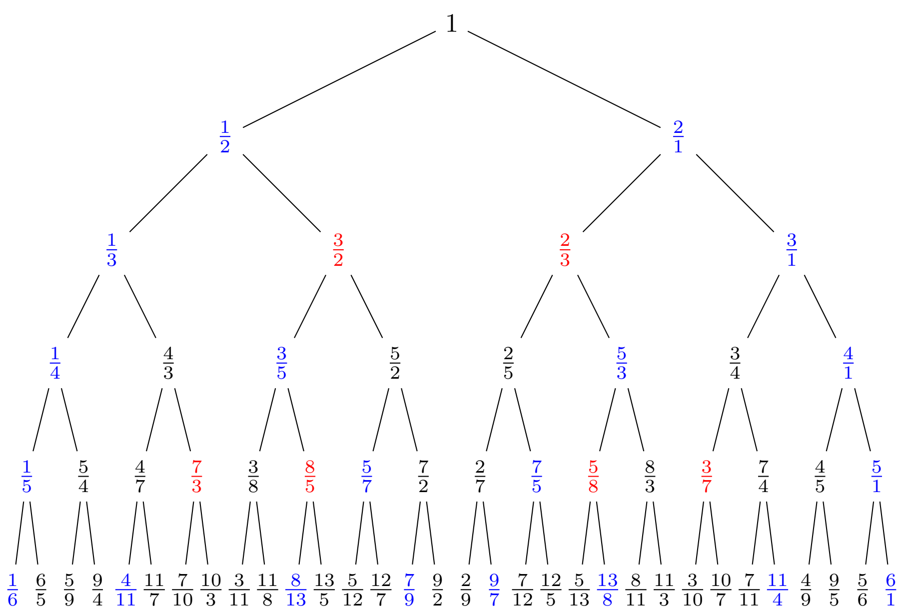

Figure 2.3: Palindromes (blue) and antipalindroms (red) on the first few levels of the tree

Also from our observations of these palindromes, we look at the spacing between the palindromes. For example, on level 3, we have a space of 1 rational between $\frac{1}{4}$ and $\frac{3}{5}$, then a space of two rationals between $\frac{3}{5}$ and $\frac{5}{3}$, and finally another space of one rational between $\frac{5}{3}$ and $\frac{4}{1}$. Table 2.1 provides the results of the list of spaces between each palindrome for the first few levels of the tree.

Observing the length of these spaces down several levels, we always observe that in the direct middle of the list of the length of space between each palindrome is 2. From the first few levels of lists of spaces between palindromes, we conjecture that the space between palindromes is always prime. However, upon listing a few more levels, we find this to not be true. However, further investigation shows that of this list of spaces between each palindrome, we find that out of this list, no element is square.

level	space between palindromes
0	0
1	0
2	2
3	1, 2, 1
4	5, 2, 5
5	3, 5, 3, 2, 3, 5, 3
6	11, 5, 11, 2, 11, 5, 11
7	7, 11, 7, 5, 7, 11, 7, 2, 7, 11, 7, 5, 7, 11, 7
8	23, 11, 23, 5, 23, 11, 23, 2, 23, 11, 23, 5, 23, 11, 23
9	15, 23, 15, 11, 15, 23, 15, 5, 15, 23, 15, 11, 15, 23, 15, 2, 15, 23, 15, 11, 15, 23, 15, 5, 15, 23, 15, 11, 15, 23, 15
10	47, 23, 47, 11, 47, 23, 47, 5, 47, 23, 47, 11, 47, 23, 47, 2, 47, 23, 47, 11, 47, 23, 47, 5, 47, 23, 47, 11, 47, 23, 47

Table 2.2: The spaces between palindromes on a given level.

We also observe that these levels once again occur in parities, like we found in our occurrences of the palindromes on levels. For example, levels 5 and 6 both have 7 spaces between palindromes. This is obvious since we know that levels 4 and 5 have $2^{\frac{6}{2}} = 2^3 = 8$ palindromes. However, we notices that the spaces between palindromes on level 5 appear to be in the order of magnitude:

small (3), large (5), small (3), center (2), small (3), large (5), small (3)

while the spaces between palindromes on level 6 appear to be in the opposite order of magnitude:

large (11), small (5), large (11), center (2), large (11), small (5), large (11)

The pairs of levels follow this parity for the spaces between palindromes.

We would love to include further findings from palindromic and antipalindromic paths in future research and these findings are welcome to open investigation from the reader.

2.2 Inverse Paths

After seeing what happens when we simply reverse a path, we want to explore what happens when we invert a path. This means that if we have a left turn, we change it to a right turn and vice versa. For example, if we have that path 011, the inverse path would be 100.

Definition: An *inverse path* of a rational $\frac{a}{b}$ in the Tree of All Fractions is the binary complement of the orignal path to $\frac{a}{b}$; i.e. the 0's and 1's are interchanged.

We quickly discovered that inverting a path will give you the reciprocal of the original rational. For example, let's take our same path 011. This leads to the fraction $\frac{5}{2}$.

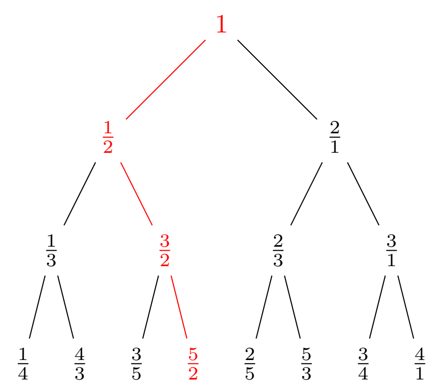

Figure 2.4: Example of a path in tree.

Note that the reverse path of this, 110, would lead to the fraction $\frac{3}{4}$. However, the inverse path, 100, leads to $\frac{2}{5}$, as seen below:

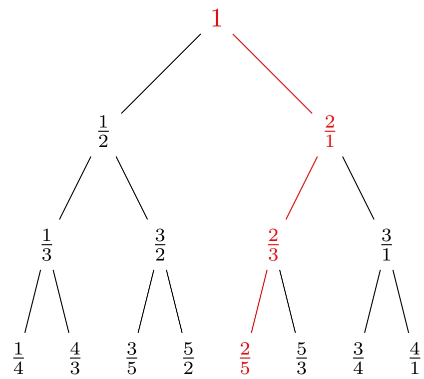

Figure 2.5: Example of an inverse path in tree.

The proof of this is simple and a sketch is given.

Proof. (Sketch) Given rational $\frac{a}{b}$, we know it has the path given by the sequence of 0's and 1's that we will call a_n. If we take the inverse of this path a_n, we change all the left turns (0) to right turns (1) and vice versa. Therefore the inverted path, a'_n, is the binary complement of a_n. Since the Tree of all Fractions has the property that the left half and the right half of the tree are related by their opposing rational being the reciprocal of the opposite side, we have that a path's binary complement will lead to the reciprocal of the original rational. Therefore, if $\frac{a}{b}$'s path is a_n, the inverted path, a'_n, leads to rational $\frac{b}{a}$. □

As with the reverse path, we created a code to give the fraction you end on if you invert the path of the original fraction to check that it is always the reciprocal, as seen in Appendix A.

2.3 Appearances in Numerators

Looking more closely at our list of numerators, we noticed that certain numbers appeared before their naturally consecutive number. An example of this is we see that 7 appears before 6 in the list of numerators. We see the list of numerators below and we can see that 7 certainly appears before 6.

$$1\ 1\ 2\ 1\ 3\ 2\ 3\ 1\ 4\ 3\ 5\ 2\ 5\ 3\ 4\ 1\ 5\ 4\ 7\ 3\ 8\ 5\ 7\ 2\ 7\ 5\ 8\ 3\ 7\ 4\ 5\ 1\ 6\ \cdots$$

We wondered how many occurrences we have of this and what numbers they were. Therefore, we created a code that returns if $N+1$ appears before N in the list of numerators.

We will call any numbers we see like 6 a "latecomer" since 6 appears after we see 7.

Definition: A *latecomer*, N, is a natural number that appears after $N+1$ or any of its subsequent numbers in the list of numerators from the Tree of All Fractions.

The list of the first few latecomers we see is

$$6\ 10\ 12\ 16\ 17\ 20\ 22\ 24\ 25\ 27\ 28\ \cdots$$

According to The Online Enclyclopedia of Integer Sequences (OEIS), we find that these are the numbers that do not appear in the running maxima of the Stern-Brocot tree, OEIS Reference A270362, yet another relation we see to the Stern-Brocot Tree.

Figure 2.6 displays a graphical representation of the order of occurrence of our list of the first $\frac{2^{20}}{4}$ numerators (the most we could fit without the plot looking too crowded) leads to interesting results. The plot below displays the each numerator as the x value and the order of its first appearance in the list of numerators.

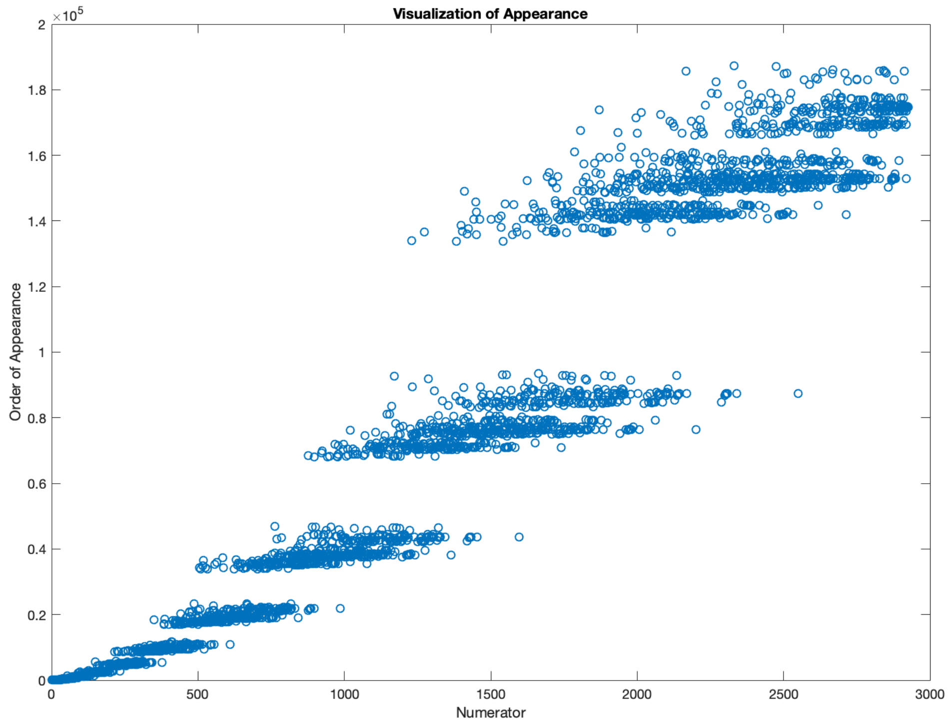

Figure 2.6: Visualization of Appearance: The numerator is plotted by its first occurrence

Since our plot only displays the first occurrence of each numerator, the clear breaks in the plot are instances in our list of numerators where we see repeats of certain numbers. It is also noteworthy to observe that these gaps in our plot occur at multiples of powers of 2.

The most interesting thing is that the upper bound of this plot is formed by the latecomers themselves; i.e. the left most endpoint of every "chunk" on the plot is a latecomer. So, our plot is bounded above by the latecomers. This makes sense since the latecomers literally come later in the sequence of numerators than their neighboring numerators. The lower bound of this plot is the Fibonacci numbers; i.e. the right most endpoint of every "chunk" on the plot is a Fibonacci number. This leads us to the result that Fibonacci numbers, which already have distinct properties in the tree as discovered by Calkin and Wilf, have the additional property that they are the first in their group to appear in the list of numerators. You could go as far as to say they are "early-comers" since they represent the opposite of our latecomers.

Chapter 3

Zig Zags and Complexities

We now want to observe how turns in the path affect the complexity of the rational where our path ends. We already know from previous work that a "perfectly zig-zagged" path, 01010101 or left, right, left, right, left, right, left, right, will end on a "Fibonacci" rational; i.e. where the numerator and the denominator are consecutive Fibonacci numbers and the rational appears in the middle of a level on the tree. To closely study these effects, we first had to define what we mean by complexity.

3.1 Defining Complexities

We defined our first complexity to be sum complexity. Just as it sounds, the sum complexity of a rational number, $\frac{a}{b}$, is $a+b$. So, the sum complexity of $\frac{7}{10}$ would be $\text{sum}(7,10) = 7+10 = 17$.

The second, and most simple complexity we defined, was max complexity. Here, we let the max complexity of a rational, $\frac{a}{b}$, be the maximum value between a and b, $\max(a,b)$. So, the max complexity of $\frac{3}{5}$ would be 5.

The final complexity we defined was path complexity. Path complexity is the product of the lengths of the runs of consecutive 0's and 1's plus 1. For example the path 000110000 would have a path complexity of $(3+1)\cdot(2+1)\cdot(4+1) = 60$ since we have 3 zeros followed by 2 ones and then 4 more zeros.

These definitions of complexity can all be useful for proving previous results about the tree by induction.

3.2 Sum Complexity Results

While playing around with the different complexities, we discovered a few interesting results. The first complexity that caught our interest was sum complexity and one of the questions we asked ourselves was if we could determine the levels that bound the occurrence of a certain sum complexity. Let's say you want to find the levels where a sum complexity of k occurs. We found that this is going to be bounded below by level $k-1$ and above by the level where the sum complexity occurs on. This above level bound is the index of where m occurs in the Fibonacci Sequence ($F_n = m$) minus one, $n-1$.

For example, let's say we want to find where sum complexity $k = 5$ occurs. We can find the levels that found this sum complexity using our results described above. The lower level bound is easy, it is simply level $5 - 1 = 4$. We can find that it is bounded above by level 3 since 3 is the largest Fibonacci number less than 5. So, $m = 3$. We know that 3 is the fourth Fibonacci number, $F_4 = 3$. So, $n = 4$. So, we have that 5 is bounded above by level $n - 1 = 4 - 1 = 3$.

Another interesting result from playing around with sum complexity was that the maximum sum complexity on level k is F_{k+2} where we start at $\frac{1}{1}$ and "properly" zig-zag down the tree.

For example, consider level 3 on the tree:

$$\frac{1}{3} \qquad \frac{3}{2} \qquad \frac{2}{3} \qquad \frac{3}{1}$$

The max sum complexity is 5 because $F_{k+2} = F_{3+2} = F_5 = 5$ since 5 is the fifth Fibonacci number. We can check this by evaluating each sum complexity of each rational on the third level and finding the maximum:

$\text{sum}(1,3) = 1 + 3 = 4$

$\text{sum}(2,3) = 2 + 3 = 5$

$\text{sum}(3,2) = 3 + 2 = 5$

$\text{sum}(3,1) = 3 + 1 = 4$

$\max(4,5) = 5$. So, it is true that the maximum sum complexity on the third level is 5.

3.3 Maximizing Complexity

We then asked the question, "If we have a path to a rational and we change one of the turns in the path, how does this affect the complexity?" and also "If we are just adding this one extra turn, where do we change directions in the path such that we will maximize our complexity?" The answer to these questions is the same no matter what defined complexity we use.

Let's first look how changing one turn in a path affects the complexity of a rational. Let's take a simple example and use path complexity, Look at the path 0000001111111. This gives us the rational $\frac{50}{7}$, which has sum complexity 57, max complexity 50, and path complexity 56. If we change the third 0 to a 1, this will obviously change the path and the rational the path leads to as well as the complexities. The path 0010001111111 leads to the rational $\frac{109}{15}$, which has a sum complexity of 124, max complexity of 109, and path complexity of 192.

Changing the turn in a path doesn't tell us too much about the changing complexities, so we looked further into the question of how do we choose the change such that we maximize our new complexity. Our first assumption for this question was that we are starting with a rational that is far away from a Fibonacci rational. Therefore, its path will have long strings of 0's and 1's instead of strictly zig-zagging paths as seen in the Fibonacci rationals and rationals close to the Fibonacci rationals.

We found that if we change one turn we can maximize the complexity. The turn we change will the the turn that occurs in the middle of the longest consecutive portion of the path. For example, if we have a path of 00000000011111, we would change the fifth zero, a left turn, to a 1, a right turn. So, the changed path that maximizes complexity is now 00001000011111. If the biggest portion of the path contains an even number of zeros or ones, the complexity is maximizes whether you change the $\frac{n}{2}-1$ or the $\frac{n}{2}+1$ term where n is the number of zeros or ones in the largest portion of the path. For example, if we have a path of 0000000011111, we could change the fourth or sixth zero to a one to maximize complexity. In summary, to maximize complexity we change the the turn that occurs in the middle of the longest consecutive portion of the path. If the longest consecutive portion is odd of length n, we change turn $\lceil \frac{n}{2} \rceil$. If the longest consecutive portion is even of length n, we can change turn $\frac{2}{n}-1$ or $\frac{2}{n}+1$.

Maximizing complexity brings the rational in question closer to the center of the tree. For example, the path 00011 leads to $\frac{9}{4}$. Maximizing complexity would change this path to 01011, which leads to $\frac{13}{5}$. Maximizing complexity allows you to draw nearer to a rational that contains a Fibonacci number, which, as we know, are found near the center of the tree according to their "proper zig-zag" paths.

Chapter 4

Distribution

Now, we'll figure what probabilities on each level look like and investigate distributions on the positive reals which arise naturally from the tree. We know that real numbers have their own continued fraction expansion and we also know that the Tree provides paths based off of a number's continued fraction expansion. Although the Tree of All Fractions contains only rationals, we try to find a distribution over the tree and its levels in hopes of discovering a distribution over the reals by this continued fraction relation. Therefore, in this section, we explore the relationship the Tree of All Fractions could have with a distribution on the real numbers.

4.1 Level Probabilities

Our interest in distributions within the tree begins with the observation that the probability that a real number less then or equal to r is the same as the probability that the n^{th} level's label is less than or equal to r; i.e.

$$P_n(x \leq r) = \text{probability } n^{th} \text{ level label} \leq r$$

In other words, we want to find the following in accordance to the n^{th} level:

$$P_n\left(x \leq \tfrac{a}{a+b}\right) = \tfrac{1}{2}P_{n-1}\left(x \leq \tfrac{a}{b}\right)$$

Better yet, we define $Q_n(x \leq r)$ to be the probability that the first n levels of the tree are $\leq r$. This was motivated by an observation made by moving down the levels and observing how many numbers are greater than or equal to 3. For example, on level 1, no numbers are greater or equal to 3 since the only number on level 1 is $\frac{1}{1}$. Therefore, $Q_1(x \geq 3) = 0$. Similarly, on level 2, we have $Q_n(x \geq 3) = 0$. On level 3, out of all the numbers on all the levels we have seen so far, only 1 number, $\frac{3}{1}$, out of the seven we have seen is greater or equal to 3. Therefore, $Q_3(x \geq 3) = \frac{1}{7} = \frac{1}{2^3-1}$. Similarly, on level 4, we obtain $Q_4(x \geq 3) = \frac{2}{2^4-1}$. From this, we figured that our probability Q_n to be

$$Q_n\left(x \leq \tfrac{a}{a+b}\right) = \tfrac{1}{2}Q_{n-1}\left(x \leq \tfrac{a}{b}\right)$$

If we let $Q(x \leq r) = \lim_{n\to\infty} Q_n(x \leq r)$ and $Q(x \leq r) = 2^{-r}$, we want to show that Q does indeed exists, which we prove in the following section.

4.2 Level Probability Proof

We want to show that $Q_n(x \leq \frac{a}{b}) \to Q(x \leq \frac{a}{b})$. So, we can just prove $\lim_{n\to\infty} Q_n(x \leq \frac{a}{b}) = Q(x \leq \frac{a}{b})$ where

$$Q_n(x \leq \tfrac{a}{b}) = \begin{cases} 0 & n < \frac{a}{b} \\ \frac{2^{n-a/b}}{2^n-1} & n \geq \frac{a}{b} \end{cases} \qquad \text{and} \qquad Q(x \leq \tfrac{a}{b}) = 2^{-\frac{a}{b}}$$

Proof. The case for when $n < \frac{a}{b}$ is easy, so we just show for when $n \geq \frac{a}{b}$.

For all $\epsilon > 0$, let $N = \log_2(2^{-\frac{a}{b}}\epsilon + 1)$ such that for all n, $n > N$. So, we have that

$$\begin{aligned}
|Q_n - Q| &= \left|\frac{2^{n-a/b}}{2^n - 1} - 2^{-a/b}\right| \\
&= \left|\frac{2^{n-a/b}}{2^n - 1} - \frac{2^{-a/b}(2^n - 1)}{2^n - 1}\right| \\
&= \left|\frac{2^{-a/b}}{2^n - 1}\right| \\
&< \left|\frac{2^{a/b}}{2^N - 1}\right| \\
&= \left|\frac{2^{-a/b}}{2^{\log_2(2^{-a/b}\epsilon+1)} - 1}\right| \\
&= \left|\frac{2^{-a/b}}{2^{-a/b}\epsilon + 1 - 1}\right| \\
&= \left|\frac{2^{-a/b}}{2^{-a/b}\epsilon}\right| \\
&= \left|\frac{1}{\epsilon}\right| < \epsilon
\end{aligned}$$

Since $\forall\epsilon > 0$, there exists an $N = \log_2(2^{-a/b}\epsilon + 1)$ such that for all $n > N$ we have $|Q_n(x \leq \frac{a}{b}) - Q(x \leq \frac{a}{b})| < \epsilon$, we have that $\lim_{n\to\infty} Q_n(x \leq \frac{a}{b}) = Q(x \leq \frac{a}{b})$. $\square$

4.3 Arbitrary Distribution

As mentioned previously, the Tree of All Fractions has a beautiful, symmetric property in which the two halves of the tree, the left half and the right half, are complements of each other. We have assumed that we can choose a left or right branch from a parent rational with equal probability of $\frac{1}{2}$. However, what happens when we change the probability of choosing a left or right branch? In this section, we'll assume we choose the left branch with probability p and the right branch with probability $q = 1 - p$. From this, we want to see if we can achieve a given distribution on $\mathbb{R}^+$.

Let's begin by thinking of this idea in reverse. Let's assume we have a distribution with $P(X < x) = F(x)$ as our cummulative distribution function. Let $F\left(\frac{a_n}{a_{n+1}}\right)$ for each label on our tree and split the left and right probabilities of each path to the left and right child as p_n and q_n, as shown in the following figure.

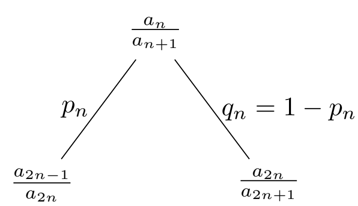

Figure 4.1: A parent and its child with chosen path probabilities.

We want to convert this cdf, $F\left(\frac{a_n}{a_{n+1}}\right)$, to a probability mass function on levels which get distributed appropriately to the children $\frac{a_{2n-1}}{a_{2n}}$ and $\frac{a_{2n}}{a_{2n+1}}$. We begin with a simple example, the first level of the tree, as build from there.

The first level of the tree with appropriate left and right probabilities would look like the figure below

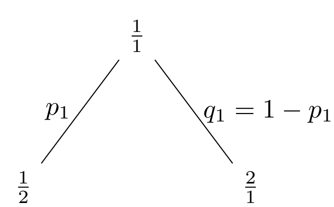

Figure 4.2: The first level of the tree with chosen path probabilities

Therefore, our cdf of the first level of the tree would be

$$F_1(x) = \begin{cases} 0 & x < \frac{1}{2} \\ p_1 & \frac{1}{2} \leq x < 2 \\ 1 & x \geq 2 \end{cases}$$

Let's add another level to see how our choice of path probabilities affect our cdf.

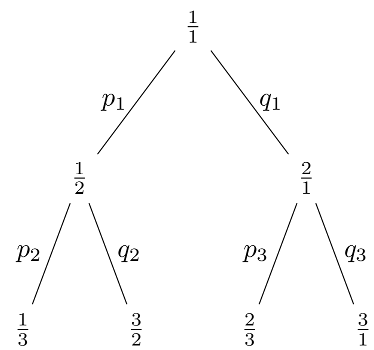

Figure 4.3: The first level of the tree with chosen path probabilities

Therefore, have the following cdf for *just* level 2:

$$F_2(x) = \begin{cases} 0 & x < \frac{1}{3} \\ p_1p_2 & \frac{1}{3} \leq x < \frac{2}{3} \\ p_1p_2 + q_1p_3 & \frac{2}{3} \leq x < \frac{3}{2} \\ p_1p_2 + q_1p_3 + p_1q_2 & \frac{3}{2} \leq x < 3 \\ 1 & x \geq 3 \end{cases}$$

Therefore, our pmf for the rationals on the second level is the following:

$$P\left(X = \frac{1}{3}\right) = p_1p_2$$
$$P\left(X = \frac{2}{3}\right) = q_1p_3$$
$$P\left(X = \frac{3}{2}\right) = p_1q_2$$
$$P\left(X = \frac{3}{1}\right) = q_1q_3$$

We look at one more level's cdf before making any conjectures. The cdf of *just* level 3 would be

$$F_3(x) = \begin{cases} 0 & x < \frac{1}{4} \\ p_1p_2p_3 & \frac{1}{4} \le x < \frac{2}{5} \\ p_1p_2p_3 + q_1p_2p_3 & \frac{2}{5} \le x < \frac{3}{5} \\ p_1p_2p_3 + q_1p_2p_3 + p_1q_2p_3 & \frac{3}{5} \le x < \frac{3}{4} \\ p_1p_2p_3 + q_1p_2p_3 + p_1q_2p_3 + q_1q_2p_3 & \frac{3}{4} \le x < \frac{4}{3} \\ p_1p_2p_3 + q_1p_2p_3 + p_1q_2p_3 + q_1q_2p_3 + p_1p_2q_3 & \frac{4}{3} \le x < \frac{5}{3} \\ p_1p_2p_3 + q_1p_2p_3 + p_1q_2p_3 + q_1q_2p_3 + p_1p_2q_3 + q_1p_2q_3 & \frac{5}{3} \le x < \frac{5}{2} \\ p_1p_2p_3 + q_1p_2p_3 + p_1q_2p_3 + q_1q_2p_3 + p_1p_2q_3 + q_1p_2q_3 + p_1q_2q_3 & \frac{5}{3} \le x < \frac{4}{1} \\ 1 & x \ge \frac{4}{1} \end{cases}$$

Now, we ask ourselves, "What can we choose for these path probabilities for the cdf and pmf to fit a distribution?"

Answers to this question are not clear. However, the plots, shown in figures 4.4 and 4.5 of the pmf and cdf of multiple levels provide some insight.

As Dr. Calkin conjectured, we definitely see some exponential tendencies arising. We are planning on looking further into these results to see if a distribution on the real numbers could result from these seen distributions.

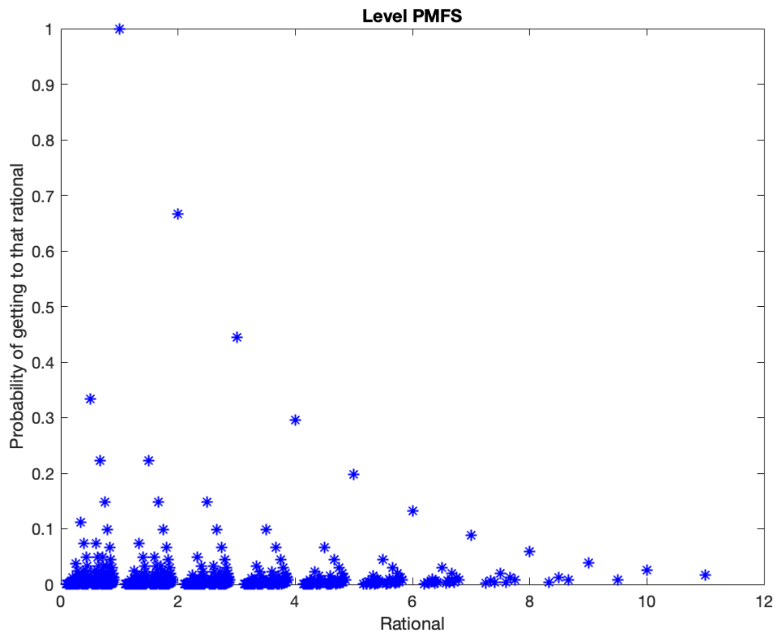

Figure 4.4: Probability Mass Function for Levels 0 through 12.

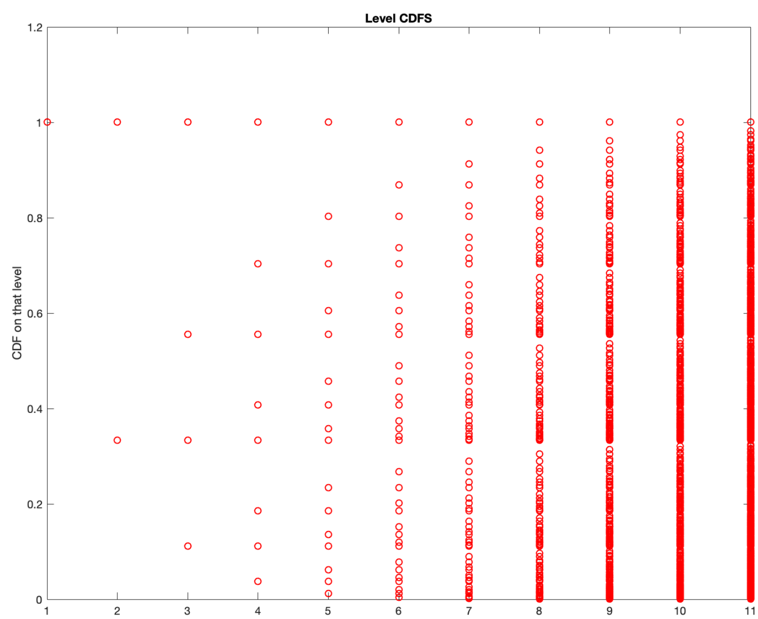

Figure 4.5: Cummulative Distribution Function for Levels 0 through 12.

While these plots above don't tell us much about distributions over the reals, we also generate random paths of varying lengths (distributed uniformly from length 1 to the length it takes to get to the desired level) with different path probabilities; with $p = 1/3$ (green), $p = 1/2$ (blue), and $p = 2/3$ (yellow).

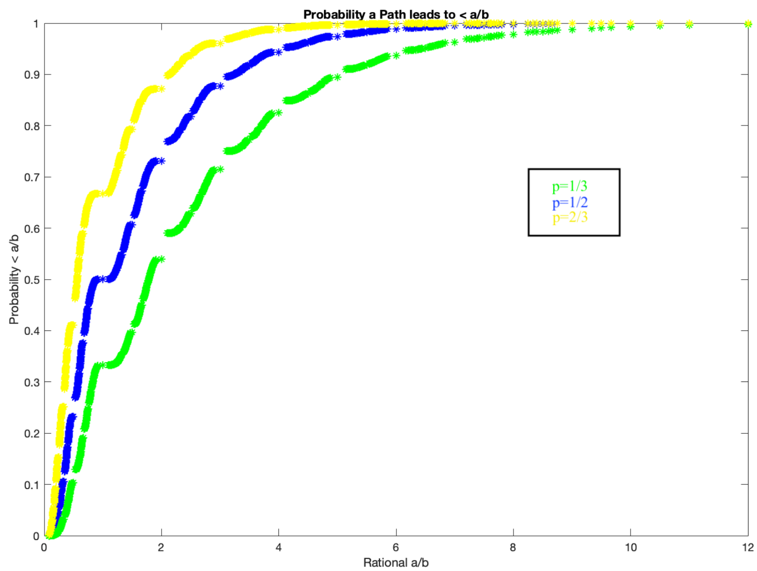

Figure 4.6: The probability a path leads to less than $\frac{a}{b}$ is shown above with $p = \frac{1}{3}$ in green, $p = \frac{1}{2}$ in blue, and $p = \frac{2}{3}$ in yellow. This plot reaches level 12.

We choose $p = \frac{1}{2}$ to make sure our plot is correct because this way we can check that at the rational $\frac{1}{1}$, we know we should have a y-value of $\frac{1}{2}$ since we know from the parity of the Tree of All Fractions that half of our rationals will be less than $\frac{1}{1}$ and half will be greater than $\frac{1}{1}$. When we make our left branch probability to be less than $\frac{1}{2}$; i.e. $p = \frac{1}{3}$, we increase our probability of our end value of our path to be greater than $\frac{1}{1}$. When we make our left branch probability to be greater than $\frac{1}{2}$; i.e. $p = \frac{2}{3}$, we decrease our probability of our end value of our path to be greater than $\frac{1}{1}$.

While finding a distribution over the real numbers is quite hard (if not impossible), we do find some interesting results when we look at rational fraction distributions. For this analysis, we look at Thomae's Function, defined as a function $t : \mathbb{R} \to [0, 1]$ where

$$t(x) = \begin{cases} \frac{1}{b} & x = \frac{a}{b}; a, b \text{ coprime}; x \text{ is rational} \\ 0 & x \text{ is irrational} \end{cases}$$

Trifonov, Pasqualucci, Dalla-Favera, and Rabadan found a way to create a distribution that is a convolution over the rational numbers [4]. They say to start with a given distribution, in our case, we will start with $f(x)$. Note that this distribution is unnamed. From this distribution, we are assuming we can generate rationals, $r = \frac{a}{a+b}$. These rationals will have a distribution, which we will call $F(r)$ on the rationals that is of the following form, where δ is the classic Kronecker Delta Function:

$$\begin{aligned} F(r) &= F\left(\frac{a}{a+b}\right) \\ &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} f(i)F(j)\delta\left(\frac{a}{a+b} - \frac{i}{i+j}\right) \\ &= \sum_{k=0}^{\infty} f(ka)f(kb) \end{aligned}$$

The authors found closed form solutions for exponential, uniform, and Poisson distributions. These solutions show similarities between their closed forms and Thomae's function [1]. These ideas from Trifonov, Pasqualucci, Dalla-Favera, and Rabadan may prove to be helpful when we attempt to identify a distribution over the real numbers.

Chapter 5

Future Questions

During our research, we asked ourselves a couple further reaching questions which we will discuss here. Hopefully, these lead to more research and deeper understanding.

5.1 Other Euclidean Domains

Paths in the Calkin-Wilf tree correspond to continued fractions of certain rationals. For example, the rational $^{53}/_{37}$ has a continued fraction expansion of $\begin{bmatrix}1 & 2 & 3 & 4 & 1\end{bmatrix}$. We already know that continued fractions also relate to the Euclidean algorithm in the fact that they are the quotients of the algorithm [6], like so: The Euclidean algorithm of $^{53}/_{37}$ is

$$\begin{aligned} 53 &= 37(1) + 16 \\ 37 &= 16(2) + 5 \\ 16 &= 5\ (3) + 1 \\ 5 &= 1\ (4) + 1 \\ 1 &= 1\ (1) + 0 \end{aligned}$$

If you read the continued fraction, or the quotients of the Euclidean algorithm backwards (and "chop" off the first value in some cases), you get the number of steps in a single direction; i.e. our path for $^{53}/_{37}$ is $\begin{bmatrix}0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1\end{bmatrix}$, the reverse of our continued fraction. So, we begin with 4 left turns, 3 right turns, 2 left turns, and 1 right turn in our path for $^{53}/_{37}$.

We then thought if it is possible to create similar trees for other Euclidean Domains. Our first thought was to try this on the Gaussian Integers. We did make a few assumptions before creating the brother tree to the Calkin-Wilf tree in the Gaussian Integers. One of the biggest assumptions we made was that some Gaussian Integers will have two unique forms when written in as $a = qb + r$. For example, $\frac{3+2i}{-1+3i}$ can be written as either

$$3 + 2i = -i(-1 + 3i) + i \qquad \text{or} \qquad 3 + 2i = (1 - i)(-1 + 3i) + (1 + 2i)$$

So, we decide to choose the one with the smallest remainder. Although this question about Gaussian integers hasn't been further explored, we would love to continue this question about possible analogous trees for Euclidean Domains.

5.2 Sequences in the Tree

We had the idea to take famous binary sequences, assign left and right path turns, and then walk down the tree according to the sequence. The sequences we looked at were: Thue-Morse, Kalakowski, Fibonacci-Word, Baum-Sweet, and the Paper-Folding sequence. Although, looking at the rationals that subsequences of these famous binary sequences gave didn't prove interesting results. However, this study did lead to our questions about continued fractions and other Euclidean Domains in relation to the tree.

5.3 New and Arbor-itrary Trees

Another interesting result arises when we ask, "What happens if we start the tree, with the same parent-child relationship, with the rational $\frac{0}{1}$ or $\frac{1}{0}$ instead of the classic $\frac{1}{1}$?" For the tree that begins with $\frac{0}{1}$, we have that the left most branches remain the fraction $\frac{0}{1}$, but the right branches give the Calkin-Wilf tree again. Similar results occur when we start with $\frac{1}{0}$.

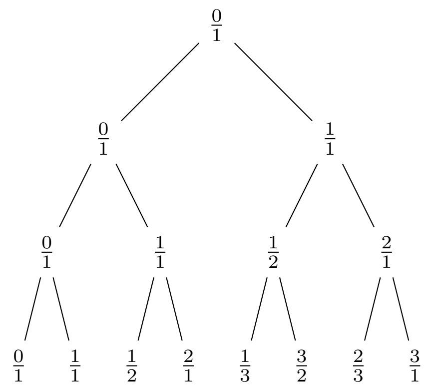

Figure 5.1: The tree formed by beginning at $\frac{0}{1}$

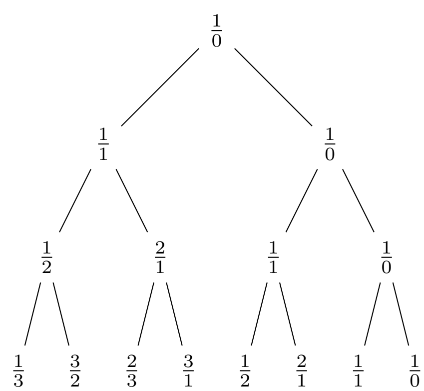

Figure 5.2: The tree formed by beginning at $\frac{1}{0}$

We extend this idea to ask the question, "What does the general $\frac{a}{b}$ tree look like? Is it a linear combination of these two trees?" To begin to answer this, we look at the general $\frac{a}{b}$ tree that follows the same rules as the Calkin-Wilf Tree.

Looking at the $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$-tree, we can see a relationship between these coefficients on a and b

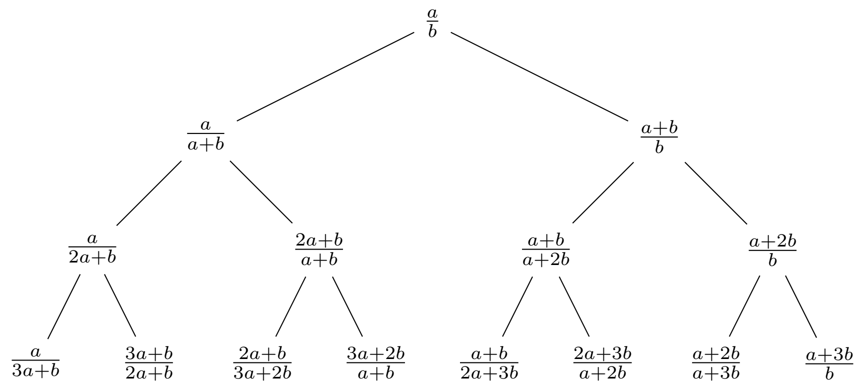

Figure 5.3: The tree formed by beginning at $\frac{a}{b}$

and the numerators and denominators of the $\begin{pmatrix}0\\1\end{pmatrix}$-Tree. The list of the first few numerators in the $\begin{pmatrix}0\\1\end{pmatrix}$-tree is

$$0\ 0\ 1\ 0\ 1\ 1\ 2\ 0\ 1\ 1\ 2\ 1\ 3\ 2\ 3$$

We want to write a functional equation for the sequence of numerators similar to the function generating function, $F(x) = (1+x+x^2)F(x^2)$, that was found for the hyperbinary numbers. Progress in this has been made by Mansour and Shattuck [3] and the reader is more than welcome to use the code for creating these trees and lists of numerators found in the Appendix to further research these questions. However, may it be noted that similar work has been done concerning the general $\begin{pmatrix}a\\b\end{pmatrix}$-tree by Mansour and Shattuck.

Chapter 6

Conclusions and Discussion

We have looked at modifications on the paths in the tree, "latecomers" in our numerators, complexity classes along the paths, level and arbitrary distributions in the tree, and the tree in reference to other Euclidean Domains, other sequences, and other trees. Overall, our study of The Calkin-Wilf Tree and its varying properties is never done. This Tree of All Fractions continuously shows us new patterns and properties the more we investigate. Any continued work on the results mentioned in this paper are welcome to all to explore and discover as the Tree of All Fractions can provide insight to a multitude of disciplines.

Appendices

Appendix A Codes for Project

Creating the Calkin-Wilf Tree

This function gives you the Calkin-Wilf Tree, the list of numerators, or the list of denominators depending on your input.

```
%If you enter TreeNum(n,1), you will get a list of length (2^n)-1 numerators.
%If you enter TreeNum(n,2), you will get a list of length (2^n)-1 denominators.
%If you enter TreeNum(n), you will get a list of length (2^n)-1 fractions.
function [ list ] = TreeNum( n,pos )
m = (2^n)-1; num = zeros(1,m); denom = zeros(1,m); num(1) = 1; denom(1) = 1;
for i =2:m
    if mod(i,2) == 0
        num(i) = num(i/2);
        denom(i) = denom(i/2)+num(i/2);
    else
        num(i) = num((i-1)/2)+denom((i-1)/2);
        denom(i) = denom((i-1)/2);
    end
end
if nargin>1
    if pos ==1
        list = num;
    elseif pos == 2
        list = denom;
    end
else
    list = num./denom;
end
end
```

Returning the path of a rational

This returns the path taken to get from 1/1 to a/b where 0=left, 1=right

```
function [ history ] = Genealogy( a,b )

history = -1*ones(1,max(a,b));

index = 1;

while a~=b

    if a*b == 0

        history(index-1) = -1;

        a =b;

    elseif a>b

        temp = floor(a/b);

        temp_vec = ones(1,temp);

        history(index:index+temp-1) = temp_vec;

        index = index+temp;

        a = a-temp*b;

    else

        temp = floor(b/a);

        temp_vec = zeros(1,temp);

        history(index:index+temp-1) = temp_vec;

        index = index+temp;

        b = b-temp*a;

    end

end

history = history(1:index-2);

history = fliplr(history);

end
```

Evaluating a Path This function takes a path input and gives you the rational that this path leads to on the tree.

```
function [ a,b ] = TreeEval( direction )
a = 1;
b = 1;

for i = 1:length(direction)
    if (direction(i) == 0)
        b = b+a;
    else
        a = a+b;
    end
end
end
```

Reverse Path

This function goes through the tree and gives you what the a/b becomes if you read the path backwards.

```
function [ matrix ] = reversePath( k,m )
num = TreeNum(k,1);
denom = TreeNum(k,2);
new_num = zeros(1,length(num));
new_denom = zeros(1,length(denom));
for i = 1:length(num)
    [flip_a, flip_b] = RatFlip(num(i),denom(i));
    new_num(i) = flip_a;
    new_denom(i) = flip_b;
    if (nargin>1)
        fprintf('The fraction %d/%d becomes %d/%d\n',num(i),denom(i),flip_a,flip_b);
    end
end
  matrix = [num; denom; new_num; new_denom];
```

Palindromes

This function inputs a fraction and will tell you if its path is palindromic.

```
function [output] = palindromecheck(a,b)

[path1] = Genealogy(a,b);

path2 = fliplr(path1);

[c,d] = TreeEval(path2);

if (a==c) && (b==d)

    output=1;

else

    output=0;

end
```

This function inputs a fraction and will tell you if its path is antipalindromic.

```
function [output] = antipalindromecheck(a,b)

[path1] = Genealogy(a,b);

path2 = fliplr(path1);

[c,d] = TreeEval(path2);

if (a==d) && (b==c)

    output=1

else

    output=0

end
```

Palindromes Continued

This script gives a list of palindromes in the tree, counts the spaces between palindromes, and can tell you information about the palindromes and their spaces.

```
n=15;
nums = TreeNum(n,1);
dems = TreeNum(n,2);
pcheck = [];
pals = [];

for i=1:length(nums)
    pcheck = [pcheck palindromecheck(nums(i),dems(i))];
end

for j=1:length(pcheck)
    if pcheck(j) == 1
        pals = [pals nums(j)*dems(j)];
    else
        pals = [pals];
    end
end
queue=[];
space=[];
for i=1:length(pcheck)
    if pcheck(i) == 0
        queue = queue+1;
    else
        space=[space queue];
        queue = 0;
    end
end
TF = isprime(space); TF = issquare(space);
```

Palindromes Continued

This function checks up to $(2^n) - 1$ elements in the tree if they are palindromic or antipalindromic

```
function[pals] = TreePalCheck(n)
nums = TreeNum(n,1);
dems = TreeNum(n,2);
pals = zeros((2^n)-1,1);
anti = zeros((2^n)-1,1);
for i=1:length(nums)
     pals(i) = palindromecheck(nums(i),dems(i));
     anti(i) = antipalindromecheck(nums(i),dems(i));
end

%gives descriptive table of which rationals are palindromes/antipalindromes
T=table(nums(:),dems(:),pals(:),anti(:),'VariableNames',{'Numerator','Denominator',
'Palindrome','Antipalindrome'})

%plots how many palindromes on each level
for i=1:n-1
    y=sum(pals((2^(i-1))+1:2^i));           %count the number of palindromes on level i-1
    plot(i,y,'*')                           %plot level with the number of palindromes
    hold on
end
```

Inverse Path

This function will invert the path you input (0-¿1) (1-¿0)

```
function [ inv_path ] = TreeInversion( path )

n = length(path);

inv_path = zeros(1,n);

for i = 1:n

    inv_path(i) = mod(path(i)+1,2);

end

end
```

This function will invert the path of a/b and give you b/a.

```
function [ inv_a,inv_b ] = inversePath( a,b )

o_path = Genealogy(a,b);

inv_path = TreeInversion(o_path);

[inv_a, inv_b] = TreeEval(inv_path);

end
```

Numerator Numbering

This script gives you the plot that was created to show the numerator and its corresponding order of appearance in the tree. This script require a file called `20Nums.txt` that is simply a text file of the first 2^{20} numerators found in the tree.

```
A=readtable('20Nums.txt');
C=table2array(A);
C=[1; C];
C=C(1:length(C)/14);
x=1:1:length(C)/14;
y=[];
for i=1:length(C)/14
    y=[y find(C==x(i),1,'first')];
end
x=x(1:length(y));


plot(x,y,'o')
title('Visualization of Appearance')
xlabel('Numerator')
ylabel('Order of Appearance')
```

Complexities

The following function, can be used to find sum, max, or product complexity depending on which lines you decide to comment out.

```
function [ c ] = Complexity(a,b)
c=a+b;                  %sum complexity
%c=max(a,b);            %max complexity
%c=a*b;                  %product complexity
end
```

The following function gives the path complexity of a certain rational's path.

```
function [ product ] = pathComplexity( path )
n=length(path);
product=1;
SPL=2;
for i=2:n
    if path(i)==path(i-1)
        SPL=SPL+1;
        if i==n
            product = product*SPL;
        else
        end
    else
        product = product*SPL;
        SPL=2;
    end
end
```

The following function checks that complexity is maximized when the middle of the longest consecutive path is changed.

```
function [ cnew  ] = complexityCheck( path, t )
path(t)=mod(1+path(t),2);
[anew,bnew] = TreeEval(path);
[cnew] = sumComplexity(anew,bnew);
%[ cnew ] = pathComplexity( path );
end
```

Creating the $\frac{0}{1}$ Tree

This function gives you the $\frac{0}{1}$ Tree, the numerators in the $\frac{0}{1}$ Tree, or the denominators in the $\frac{0}{1}$ Tree depending on your input.

```
%If you enter ZeroOneTree(n,1), you will get a list of length (2^n)-1 numerators.
%If you enter ZeroOneTree(n,2), you will get a list of length (2^n)-1 denominators.
%If you enter ZeroOneTree(n), you will get a list of length (2^n)-1 fractions.


function [ list ] = ZeroOneTree( n,pos )
m = (2^n)-1; num = zeros(1,m); denom = zeros(1,m); num(1) = 0; denom(1) = 1;
for i =2:m
    if mod(i,2) == 0
        num(i) = num(i/2);
        denom(i) = denom(i/2)+num(i/2);
    else
        num(i) = num((i-1)/2)+denom((i-1)/2);
        denom(i) = denom((i-1)/2);
    end
end


if nargin>1
    if pos ==1
        list = num;
    elseif pos == 2
        list = denom;
    end
else
    list = num./denom;
end
end
```

Creating the $\frac{1}{0}$ Tree

This function gives you the $\frac{1}{0}$ Tree, the numerators in the $\frac{1}{0}$ Tree, or the denominators in the $\frac{1}{0}$ Tree depending on your input.

```
%If you enter OneZeroTree(n,1), you will get a list of length (2^n)-1 numerators.
%If you enter OneZeroTree(n,2), you will get a list of length (2^n)-1 denominators.
%If you enter OneZeroTree(n), you will get a list of length (2^n)-1 fractions.


function [ list ] = OneZeroTree( n,pos )
m = (2^n)-1; num = zeros(1,m); denom = zeros(1,m); num(1) = 1; denom(1) = 0;
for i =2:m
    if mod(i,2) == 0
        num(i) = num(i/2);
        denom(i) = denom(i/2)+num(i/2);
    else
        num(i) = num((i-1)/2)+denom((i-1)/2);
        denom(i) = denom((i-1)/2);
    end
end


if nargin>1
    if pos ==1
        list = num;
    elseif pos == 2
        list = denom;
    end
else
    list = num./denom;
end
end
```

Creating the Probability Plots

The following function generates the probability plot as seen in Figure 4.6.

```
function [] = ProbPlots(probs,colors)


for i=1:length(probs)

    p=probs(i);

    r=colors(i);

    q=1-p;

    n=12;

nums = TreeNum(n,1);

dems = TreeNum(n,2);

rats=nums./dems;

srats=sort(rats);

j=1;

actualrats=[];


    for i=1:100000

        m = randi([1,n+1]);

        path = [];

        for j=1:m

            if rand <= p

                path = [path 0];

            else

                path = [path 1];

            end

        end

        [ a,b ] = TreeEval( path );

        actualrats=[actualrats a/b];

    end
```

```
    c=['b','g','y'];

    for i=1:length(srats)
        count = sum(actualrats<srats(i));
        plot(srats(i),count/length(actualrats),'color',c(r),'marker','*')
        hold on
    end

    title('Probability a Path leads to < a/b')
    xlabel('Rational a/b')
    ylabel('Probability < a/b')
    hold on
end
```

Bibliography

[1] (ANDART II Anders Sandberg). Rational fractal distributions — (ANDART II), , 2014. [Online; accessed 2019-03-14].

[2] Neil J. Calkin and Herbert S. Wilf. Recounting the rationals. *Amer. Math. Monthly*, 107(4):306–363, 2000.

[3] Toufik Mansour and Mark Shattuck. Two further generalization of the calkin-wilf tree. *Journal of Combinatorics*, 2(4):507–524, 2011.

[4] Vladimir Trifonov, Laura Pasqualucci, Riccardo Dalla-Favera, and Raul Rabadan. Fractal-like distributions over the rational numbers in high-throughput biological and clinical data. *Scientific Reports*, 1(4):191 EP, 2011.

[5] Wikipedia: The Free Encyclopedia. Calkin-Wilf tree, Last Updated: February 13, 2019. [Online; accessed 2017/2018/2019].

[6] Wikipedia: The Free Encyclopedia. Continued fraction, Last Updated: March 12, 2019. [Online; accessed 2017/2018/2019].

[7] Wikipedia: The Free Encyclopedia. Stern-Brocot tree, Last Updated: March 14, 2019. [Online; accessed 2017/2018/2019].