$(3x+1)$ Results

Bruce Adcock & Sucheta Soundarajan

November 23, 2004

Abstract

We show a method to determine arithmetic progressions based on their m-path with respect to the structure theorem for the $(3x+1)$-Map. This leads us to examine the required properties for m-paths of numbers that would contradict the $(3x+1)$-Conjecture. This examination includes describing necessary conditions for the m-paths, and how many such m-paths there could potentially be given certain criterion. Finally, in the appendix we make use of these results with an algorithm to find numbers less than a bound that might contradict the $(3x+1)$-Conjecture.

1 Introduction

1.1 The $(3x+1)$-Map, the $(3x+1)$-Conjecture, and the Structure Theorem

Let x be an odd integer greater than 0. Then $3x+1$ is even, and so we can find a unique $k>0$ such that $y=\frac{3x+1}{2^k}$ is odd. Furthermore, 3 does not divide y. If Π is the set of natural numbers that are not divisible by either 2 or 3, then we are therefore looking at a mapping T on the set Π. Clearly, only the number 1 is fixed by a single iteration of T. Through an ongoing search [**?**], we know all numbers less than $302 \cdot 2^{50}$ eventually are mapped to 1 after a finite number of iterations of T. This leads to the following conjecture.

Conjecture 1 (The $(3x+1)$-Conjecture) *If* $x \in \Pi$ *then* $\exists n \in \mathbb{N}$ *such that* $T^n(x)=1$.

There have been many papers [**?**] written working to determine the validity of the $(3x + 1)$-Conjecture. Instead, this paper assumes that in fact the conjecture is false, and examines the properties of a number that contradicts the $3x + 1$-Conjecture. First we require the following definition.

Definition 1 *Let x be given and k_i for $i \in \{1, \ldots, m\}$ be such that*

$$T^i(x) = \frac{3T^{i-1}(x) + 1}{2^{k_i}}.$$

*Then the m-*path *of x is $(k_1, \ldots, k_m)$. Further, we write $\gamma_m(x)$ as the function that gives the m-path for $x \in \Pi$.*

The remainder of the paper makes extensive use of the Structure Theorem [**?**].

Theorem 2 (The Structure Theorem) *Let $k_1, \ldots, k_m$ be given positive integers, and $\varepsilon \in \{1, 5\}$. Then $\exists q_m$ with $0 \leq q_m < 6 \cdot 2^{k_1 + \ldots + k_m}$ and $q_m \equiv_6 \varepsilon$ such that*

$$\{a \in \Pi : \gamma_m(a) = (k_1, \ldots, k_m)\} = \{6 \cdot 2^{k_1 + \ldots + k_m} p + q_m : p \in \mathbb{N}\}.$$

Hence we get two arithmetic progressions, one for $\varepsilon = 1$ and one for $\varepsilon = 5$. Further, we need only find the minimal representatives in order to completely determine the solutions.

In Section **??**, we discuss an algorithm to determine the minimal representative of an arbitrary m-path. In Section **??**, we determine the characteristics of m-paths that represent any number that does *not* eventually get sent to 1 under repeated iterations of T. Finally, the Appendix contains sample Mathematica code used to implement many of the items discussed in this paper.

1.2 Notation

Let $\alpha = (k_1, \ldots, k_m)$ be an m-path, $i \in \{1, \ldots, m\}$, and a a representative of α. The following conventions are used throughout this paper:

E	$\{1,5\}$
ε	Arbitrary element from E
Π	$\{a\in\mathbb{N}:a \equiv_6 1,5\}$
$\gamma_m(a)$	The m-path $(k_1,\ldots,k_m)$ that represents a
Q_i	$\sum\limits_{j=1}^{i} k_j$
α_i	The initial i-path of α, i.e. $\alpha_i=(k_1,\ldots,k_i)$
$T^{(\alpha)}(a)$	The result of applying the m-path α on a.
$\mathcal{M}$	The set of all possible m-paths, or $\cup_{m=0}^{\infty}\mathbb{N}_{>0}^{m}$
$R^{(\alpha)}$	The set of representatives for α

2 Algorithm to find the minimal representatives of any given m-path

The Structure Theorem tells us that, given any m-path, there are an infinite number of representatives for it, given by two arithmetic progressions. These arithmetic progressions are determined by both the minimal representatives of the m-paths and Q_m. The latter is easily determined. The minimal representatives can be found using the following algorithm.

2.1 Minimal representatives for 1-paths

In order to determine the two minimal representatives for a given m-path, we apply a recursive algorithm. The base case for this recursion occurs when we are given a 1-path (k). To find the minimal representatives for (k), consider first the case when $x_1 \equiv_6 1$, so $x_1 = 6y_1+1$, for some integer y_1. Then $T(x) = \frac{3(6y_1+1)+1}{2^k} = a_1$, for some positive odd integer a_1. Hence $18y_1+4 = 2^k a_1$ and so $9y_1+2 = 2^{k-1}a_1$, so we wish to find the smallest positive a_1 such that $2^{k-1}a_1 \equiv_9 2$. Notice that $2^6 \equiv_9 1$, so if $2^{k-1}a_1 \equiv_9 2$, then $2^{k-1+6}a_1 \equiv_9 2$. This implies that there are 6 possible minimal values for a_1, depending on k modulo 6.

Once a value for a_1 is obtained from the above information, it can be used to determine y_1, since

$9y_1 + 2 = 2^{k-1}a_1$, which in turn can be used to determine x_1, because $x_1 = 6y_1 + 1$. Because a_1 is minimal, y_1 and thus x_1 are minimal. The solutions are below in Table **??**.

Table 1: *Choice of initial values.*

$k \equiv_6$	a_1	a_5	x_1	x_5
0	13	7	$\frac{2^k \cdot 13 - 1}{3}$	$\frac{2^k \cdot 7 - 1}{3}$
1	11	17	$\frac{2^k \cdot 11 - 1}{3}$	$\frac{2^k \cdot 17 - 1}{3}$
2	1	13	$\frac{2^k - 1}{3}$	$\frac{2^k \cdot 13 - 1}{3}$
3	5	11	$\frac{2^k \cdot 5 - 1}{3}$	$\frac{2^k \cdot 11 - 1}{3}$
4	7	1	$\frac{2^k \cdot 7 - 1}{3}$	$\frac{2^k - 1}{3}$
5	17	5	$\frac{2^k \cdot 17 - 1}{3}$	$\frac{2^k \cdot 5 - 1}{3}$

A similar process is used to determine the minimal representative equivalent to 5 modulo 6. In this case, $x_5 = 6y_5 + 5$, meaning $T(x_5) = \frac{3(6y_5+5)+1}{2^k} = a_5$ for some positive odd integer a_5. This gives us $18y_5 + 16 = 2^k a_5$ and thus $9y_5 + 8 = 2^{k-1}a_5$. We are left to find the smallest positive integer a_5 with $2^{k-1}a_5 \equiv_9 8$. Again, we need only consider 6 cases, depending on what k is modulo 6. These solutions are in Table **??**.

Using this information, we can determine a based on k, and so we can find y and thus the minimal x with the given 1-path satisfying $x \equiv_6 5$.

2.2 Inductive step

Once both minimal representatives for each of two m-paths have been found, an algorithm utilizing the Chinese Remainder Theorem can be used to find minimal representatives of the two m-paths merged. Suppose we have found the minimal representatives (x_1, x_5) and (y_1, y_5) of m-paths $\alpha = (k_1, \ldots, k_i)$ and $\beta = (k_{i+1}, \ldots, k_m)$, respectively, and wish to find the minimal representatives (z_1, z_5) of $\gamma = (k_1, \ldots, k_m)$, the m-path formed by concatenating α with β, where x_1, y_1, and z_1 are equivalent to 1 modulo 6, and x_5, y_5, and z_5 are equivalent to 5 modulo 6.

Since α is an initial sequence of γ, any representative a of γ must also be a representative of α. Thus by the Structure Theorem, z_1 is of the form $6 \cdot 2^{k_1 + \cdots + k_i} p + x_1$, for some non-negative integer

p. By finding $T^{(\alpha)}(x_1)$, we can determine whether $T^{(\alpha)}(z_1)$ is equivalent to 1 or 5 modulo 6. In the former case, $T^{(\alpha)}(z_1)$ is of the form $6{\cdot}2^{k_{i+1}+\ldots+k_m}p + y_1$, and in the latter case, $T^{(\alpha)}(z_1)$ is of the form $6{\cdot}2^{k_{i+1}+\ldots+k_m}p + y_5$. Because we are given each k_j in α, we can determine an explicit equality between $T^{(\alpha)}(z_1)$ and $6{\cdot}2^{k_{i+1}+\ldots+k_m}p + y$,

$$\frac{3^i z_1}{2^{k_1+\ldots+k_i}} + \frac{3^{i-1}}{2^{k_1+\ldots+k_i}} + \frac{3^{i-2}}{2^{k_2+\ldots+k_i}} + \ldots + \frac{3^0}{2^{k_i}} = 6{\cdot}2^{k_{i+1}+\ldots+k_m}p + y,$$

where y is either y_1 or y_5, as appropriate. After multiplying each side by $2^{k_1+\ldots+k_i}$, we obtain

$$3^i z_1 + 3^{i-1} + 2^{k_1}3^{i-2} + \ldots + 2^{k_1+\ldots+k_{i-1}} = 6{\cdot}2^{k_1+\ldots+k^i+k_{i+1}+\ldots+k_m}p + 2^{k_1+\ldots+k_i}y.$$

Using the Chinese Remainder Theorem, we can solve for the smallest value z_1 satisfying both this equation (once we have simplified appropriately for z_1) and $z_1 = 6{\cdot}2^{k_1+\ldots+k_i}p + x_1$. This result clearly has both the initial path of α and must also satisfy that $T^{(\alpha)}(z_1)$ has the m-path of β. A process conceptually identical to this is used to find z_5.

2.3 Algorithmic improvements

The algorithm outlined in Sections **??** and **??** is the most intuitive method of determining minimal representatives of a given m-path; however, with a few modifications, it's running time can be improved approximately ten-fold.

First, suppose that we have an m-path $\alpha = (k_1, \ldots, k_m)$ with minimal representatives x_1 and x_5. Note that if $k_m \equiv_2 0$, then $T^{(\alpha)}(x_1)$ and $T^{(\alpha)}(x_5)$ are both equivalent to 1 modulo 6, and if $k_m \equiv_2 1$, then $T^{(\alpha)}(x_1)$ and $T^{(\alpha)}(x_5)$ are both equivalent to 5 modulo 6. Thus when inductively joining two m-paths we need only look at the last term in the left m-path to determine which minimal representative of the right m-path to use.

When implementing the previous algorithm, any method of "splitting" the original m-path was allowable. For example, it could be split in half, and the algorithm applied to each half before recursively joining the halves together. However, use of such a method results in repeated applications of T, which is inefficient. Therefore, the fastest method of splitting the original m-path is to let the left path be nothing more than the 1-path (k_1). We know that the minimal representatives x_1 and x_5 for a 1-path are of the form $\frac{2^{k_1}a-1}{3}$, where a = 1, 5, 7, 11, 13, or 17, and so T applied to these minimal representatives is simply a: T need not even be implemented in the code. Note that if the left path is only a 1-path, then the inverse of 3 in 2^{k_1} is either $\left\lfloor \frac{2^{k_1}}{3} \right\rfloor + 1$ or -$\left\lfloor \frac{2^{k_1}}{3} \right\rfloor$, which in the equivalency classes of 2^{k_1} are the same as $\left\lceil \frac{2^{k_1}}{3} \right\rceil$ and $2\cdot \left\lfloor \frac{2^{k_1}}{3} \right\rfloor + 1$, depending on the parity of k_1. Using this, we can eliminate the need to implement either the Euclidian Algorithm or the Chinese Remainder Theorem in order to determine the minimal representatives.

Though none of these optimizations change the generic algorithm described in the previous sections, when taken together, they speed up the running time of the process by approximately a factor of 10.

2.4 Example

As an example of this process, we find the representatives of the 6-path $(1, 2, 3, 4, 5, 6)$. By the Structure Theorem, this m-path has representatives of the form $6\cdot 2^{21}p + a_\varepsilon = 12,582,912p + a_\varepsilon$, where $p\in\mathbb{N}$ and $0<a_\varepsilon<12,582,912$. There are potentially $\frac{6\cdot 2^{21}}{3} = 4,194,304$ numbers to check for both a_1 and a_5.

By Section **??**, we can quickly calculate the representative of (6) to be $384p + 277$ and $384p + 149$. From there, we make use of the algorithm from Section **??** to determine $(5, 6)$, $(4, 5, 6)$, $(3, 4, 5, 6)$, $(2, 3, 4, 5, 6)$, and finally $(1, 2, 3, 4, 5, 6)$.

Now look at $(5, 6)$, using Section **??**. Since 5 is odd, we choose the representative 149 from (6) for the task. We also need to determine the inverse of 3 modulo 2^6. Since 6 is even, this is $2 \left\lfloor \frac{2^6}{3} \right\rfloor + 1 = 43$. We therefore know that the two minimal representatives of $(5, 6)$ are of the form

$$\frac{2^5\cdot 17 - 1}{3} + 2^5\ell_1$$

and

$$\frac{2^5\cdot 5 - 1}{3} + 2^5\ell_5,$$

where $0\leq\ell_1, \ell_5<6\cdot 2^6$ and $\ell_1 \equiv_{6\cdot 2^6} 43(149 - 17)$ and $\ell_5 \equiv_{6\cdot 2^6} 43(149 - 5)$. These yield $\ell_1 = 300$ and $\ell_5 = 48$. We hence find that the minimal representatives of $(5, 6)$ are $9,781$ and $1,589$, respectively. Further work gives us the results of Table **??**.

Table 2: *Determining representatives of* $(1,2,3,4,5,6)$.

α	a_1	a_5
(6)	$384p+277$	$384p+149$
$(5,6)$	$12,288p+8,781$	$12,288p+1,889$
$(4,5,6)$	$196,608p+52,165$	$196,608p+117,701$
$(3,4,5,6)$	$1,572,864p+1,362,445$	$1,572,864p+838,153$
$(2,3,4,5,6)$	$6,291,456p+6,010,897$	$6,291,456p+3,913,745$
$(1,2,3,4,5,6)$	$12,582,912p+6,803,467$	$12,582,912p+10,997,771$

To test these two solutions we see that $T^{(1,2,3,4,5,6)}(6,803,467)=2365$ and $T^{(1,2,3,4,5,6)}(10,997,771)=3823$, giving us that both solutions worked.

3 "Bad" m-paths

It is an interesting problem to try to find the smallest number (should it exist) that starts a loop, goes to a loop, or goes off unbounded to infinity. Such a number would possess the property that the images of repeated iterations of T must always be greater than or equal to the given number itself. That is, let a be any such number so that for arbitrary m, $\alpha=\gamma_m(a)$. Then

$$a \le T^{(\alpha_i)}(a),\ \forall i\in\{1,\ldots,m\}.$$

Direct calculation yields

$$a \le \frac{3^i}{2^{Q_i}}a+\frac{3^i}{2^{Q_i}}\sum_{j=1}^{i}\frac{2^{Q_{j-1}}}{3^j},\ \forall i\in\{1,\ldots,m\}.$$

If we assume the essential portion is $\frac{3^i}{2^{Q_i}}a$ (which is clearly true for "large enough" values of a), then we come upon the following definition.

Definition 2 *A* bad m-path *is an m-path such that*

$$Q_i \le i\log_2 3,\ \forall i\in\{1,\ldots,m\}.$$

We write $\mathcal{B}$ for the set of all bad m-paths, and $\mathcal{B}_m$ for all bad m-paths of length m.

An m-path that is not bad is a better *m-path.*

When generating such an m-path we make use of these quickly deduced facts: $k_1{=}1$; and $1{\leq}k_i{\leq}i\log_2 3{-}Q_{i-1}$.

3.1 Lower bounds on minimal representatives of bad m-paths

In the definitions of bad m-paths, we ignored the effect of adding 1 after multiplying by 3. This should only concern smaller numbers, but the question is how small. This is answered by the following theorem.

Theorem 3 *Let α be a better m-path, and $a{>}1$ be some representative of α such that $T^{(\alpha_j)}(a){\geq}a$, $\forall j{\in}\mathbb{N}{\cap}[1,m]$. Then*

$$a \leq \max_{j\in\{1,\ldots,m\}} \left\{ \frac{\sum_{i=1}^{j} 2^{\lfloor (i-1)\log_2 3\rfloor - i\log_2 3}}{2^{\lceil j\log_2 3\rceil - j\log_2 3} - 1} \right\}.$$

Proof: Let α and a be as given. We will do induction on $j{\in}\{1,\ldots,m\}$. If $j=1$, then α_1 must either be 2 when $a=1$, or otherwise 1 and $a{>}1$. So now assume α_{j-1} is a bad $(j{-}1)$-path, for some $j{\in}\{2,\ldots,m\}$. We get explicitly that

$$\begin{aligned} a &\leq T^{(\alpha)}(a_j) = \frac{3^j}{2^{Q_j}}a + \frac{3^j}{2^{Q_j}}\sum_{i=1}^{j}\frac{2^{Q_{i-1}}}{3^i} \\ a - \frac{3^j}{2^{Q_j}}a &\leq \frac{3^j}{2^{Q_j}}\sum_{i=1}^{j}\frac{2^{Q_{i-1}}}{3^i} \\ \left(2^{Q_j - j\log_2 3} - 1\right)a &\leq \sum_{i=1}^{j} 2^{Q_{i-1} - i\log_2 3} \\ a &\leq \frac{\sum_{i=1}^{j} 2^{Q_{i-1}-i\log_2 3}}{2^{Q_j - j\log_2 3} - 1} \end{aligned}$$

This is good for examining a single explicitly given m-path, but we still are not able to talk about a bound on all such m-paths. To better simplify this set $c = Q_j - \lfloor j\log_2 3\rfloor$, so now

$$a{\leq}\frac{\sum_{i=1}^{j} 2^{Q_{i-1}-i\log_2 3}}{2^{\lfloor j\log_2 3\rfloor + c - j\log_2 3} - 1}.$$

Since $c\geq 1$ due to $Q_j>j\log_2 3$ (by assumption), the *largest* bound on a would be if $c=1$. As $\log_2 3\notin\mathbb{Q}$, we know $\lfloor j\log_2 3\rfloor+1=\lceil j\log_2 3\rceil$. Further, because α_{j-1} is a bad $(j-1)$-path, therefore $Q_i\leq\lfloor i\log_2 3\rfloor$ for $i\in\{0,\ldots,j-1\}$. A maximum on this bound can be found by assuming $Q_i=\lfloor i\log_2 3\rfloor$. Ergo, we have now

$$a\leq\frac{\sum_{i=1}^{j} 2^{\lfloor (i-1)\log_2 3\rfloor-i\log_2 3}}{2^{\lceil j\log_2 3\rceil-j\log_2 3}-1}.$$

To find the possibilities for any better m-path, not just for all m-paths that are bad up to $j-1$, we now must take the maximum of all possibilities, with j ranging from 1 to m.

$$a\leq\max_{j\in\{1,\ldots,m\}}\left\{\frac{\sum_{i=1}^{j} 2^{\lfloor (i-1)\log_2 3\rfloor-i\log_2 3}}{2^{\lceil j\log_2 3\rceil-j\log_2 3}-1}\right\}.$$

This gives us exactly what we wanted. $\square$

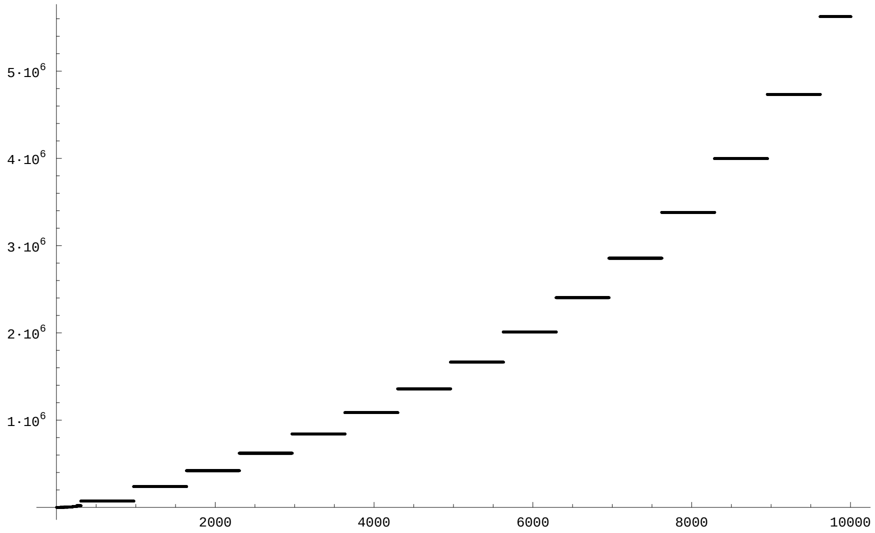

Figure 1: *Lower bounds for m ranging from 1 to 10,000, as calculated from Theorem* **??**.

Corollary 4 *Let $B\in\mathbb{N}$ be given such that $\forall b\in\Pi\cap[1,B)$, b eventually converges to 1 under repeated iterations of T. Let $M\in\mathbb{N}$ be the least M such that*

$$B<\max_{j\in\{1,\ldots,M\}}\left\{\frac{\sum_{i=1}^{j} 2^{\lfloor (i-1)\log_2 3\rfloor-i\log_2 3}}{2^{\lceil j\log_2 3\rceil-j\log_2 3}-1}\right\}.$$

If $a\in\Pi$ is such that for some $m\in\mathbb{N}$, the m-path $\alpha=\gamma_m(a)$ gives $T^{(\alpha)}(a)=a$, then either $a=1$ or $m\geq M$.

Proof: Let a be given. If $a=1$, then $\gamma_1(a)=(2)$ is a loop. Now assume that $a>1$. Let $m\in\mathbb{N}$ be the least number such that $T^m(a)=a$, and set α to $\gamma_m(a)$. We may also assume that

$a = \min\{T^n(a){:}n\in\mathbb{N}\}$, because any number $\{T^{(\alpha_i)}(a){:}i\in\{0,\ldots,m-1\}\}$ returns to itself after m iterations of T.

Now α is such that $T^{(\alpha)}(a) = a$, and $\forall i\in\{1,\ldots,m\}$, $T^{(\alpha_i)}(a) > a$. It is not possible that α is a bad m-path, because if α is such that $Q_m \leq m\log_2 3$ then in fact $T^{(\alpha)}(a) > a$. Hence, α is a better m-path. Further, by Theorem **??** we have that $a < C$, where

$$C = \max_{j\in\{1,\ldots,m\}} \left\{ \frac{\sum_{i=1}^{j} 2^{\lfloor (i-1)\log_2 3\rfloor - i\log_2 3}}{2^{\lceil j\log_2 3\rceil - j\log_2 3} - 1} \right\}.$$

However, all numbers less than B are assumed to converge. This means that it must be that $C{\geq}B$. For this to be true, m is forced to be at least as large as the first such number that gives this. ☐

To see a use of Corollary **??**, we note that it has been checked that all numbers less than $302 \cdot 2^{50}$ converge to 1. If this number is increased to 2^{60}, using the algorithm given in Appendix **??** we can determine that if $a{>}1$ has a loop of α (should it exist), then α is an m-path with $m{>}6,586,818,669$.

3.2 The number of bad m-paths for a fixed m

When generating bad m-paths, it is useful to know how many such m-paths we will in fact be dealing with. To do so, we consider a new function, $\delta(k,m)$, such that $\delta(k,m)$ is the number of bad m-paths such that $Q_m = m+k$. Note that $\delta{:}\mathbb{N}^2{\to}\mathbb{N}$, and must be such that

1. If $k > m(\log_2 3 - 1){=}m\log_2 \frac{3}{2}$ or $k < 0$, then $\delta(k,m) = 0$.
2. If $k{=}0$, then $\delta(k,m){=}1$.

How, then, do we define $\delta(k,m)$ for other values of k and m? We may consider the relation between bad m-paths and bad $(m-1)$-paths. Let α be a bad m-path with $Q_m = m+k$, meaning it is counted by $\delta(k,m)$. If $k_m = 1$, then $Q_{m-1} = m+k-1$ and so α_{m-1} was an $(m-1)$-path counted by $\delta(k,m-1)$. This can further be considered for higher values of k_m, so that we get as a recursive case

$$\delta(k,m) = \sum_{i=0}^{k} \delta(i,m-1).$$

This combines to give us the following definition.

Definition 3 *Let $\delta(k,m)$ be the number of bad m-paths such that $Q_m = m + k$. Then*

$$\delta(k,m) = \begin{cases} 0, & k > m(\log_2 3 - 1) \\ 1, & k = 0 \\ \sum_{i=0}^{k} \delta(i, m-1), & \text{otherwise} \end{cases} .$$

This is all very well and good, but in fact what we would like to determine is the *total* number of bad m-paths. We therefore define a new function.

Definition 4 *Let $\Delta(m)$ be the total number of bad m-paths. Therefore*

$$\Delta(m) = \sum_{k=0}^{\lfloor m(\log_2 3-1)\rfloor} \delta(k,m) = \delta(\lfloor m(\log_2 3 - 1)\rfloor, m+1) = \delta(\lfloor (m+1)(\log_2 3 - 1)\rfloor, m+1).$$

Note that $q_m = \lfloor m(\log_2 3 - 1)\rfloor$ is in fact the largest possible value for k to plug in to $\delta(k,m)$ for a fixed m. Hence, if $q_{m+1} = q_m$ then $\Delta(m) = \delta(q_m, m+1) = \delta(q_{m+1}, m+1)$. Otherwise, if they are not the same, then $q_{m+1} = q_m + 1$ and therefore $\delta(q_{m+1}, m) = 0$. By inspection we also get that $\Delta(m) = \delta(q_{m+1}, m+1)$. For our definition, we typically use $\Delta(m) = \delta(\lfloor (m+1)(\log_2 3 - 1)\rfloor, m+1)$.

Definition **??**, while correct, is inefficient. It can be seen that due to the summands and the recursive nature of this definition that a calculation for δ using this definition runs in exponential time. For example, timings of $\Delta(14) = \delta(8, 15)$, $\Delta(15) = \delta(9, 16)$, and $\Delta(16) = \delta(9, 17)$ on the OSU Math Department's server took 34.902 seconds, 75.256 seconds, and 215.604 seconds respectively, while on a 1GHz G4 iMac took 7.743 seconds, 17.017 seconds, and 48.311 seconds. Using δ as defined to determine the number of bad 60-paths could potentially take over 21 billion years (310 million years on the iMac), assuming the power does not go out during that period of time.

For all calculations, we assume that we are dealing with $0 < k \leq m(\log_2 3 - 1)$. The first simplification we can make is

$$\delta(k,m) = \sum_{i=0}^{k} \delta(i, m-1) = \delta(k, m-1) + \sum_{i=0}^{k-1} \delta(i, m-1) = \delta(k, m-1) + \delta(k-1, m).$$

This in itself is a nice relation, but more importantly gives us

$$\delta(k,m) \;=\; \sum_{i=1}^{m}\delta(k-1,i) = \sum_{i=m_k^*}^{m}\delta(k-1,i) \tag{1}$$

where $m_k^*=\lceil \frac{k}{\log_2 3-1}\rceil$ (Note $\delta(k-1,i)=0$, when $0\leq i<m_k^*$). The value of this can be seen in the following theorem.

Theorem 5 *Let k be fixed and $m_k^*=\lceil \frac{k}{\log_2 3-1}\rceil$, so that m_k^* is the smallest value for m such that $\delta(k,m)\neq 0$. Then $\delta(k,m)$ is a polynomial of degree k defined on $[m_k^*,\infty)$. Further, if $k>0$ then $(m-(m_k^*-1))\Big|\delta(k,m)$.*

Proof: The proof is by induction. For $k=0$, $\delta(k,m)=\delta(0,m)=1$. In this case, $m_0^*=0$, and indeed $\delta(0,m)$ is defined for all $m\in\mathbb{N}$.

Now assume that for each fixed integer k, $\delta(k,m)$ is a polynomial in m of degree k, and defined on $[m_k^*,\infty)$. Therefore,

$$\delta(k,m)=\sum_{n=0}^{k}a_n m^n.$$

Hence, by Equation **??** we see

$$\delta(k+1,m) \;=\; \sum_{i=m_{k+1}^*}^{m}\sum_{n=0}^{k}a_n i^n = \sum_{n=0}^{k}a_n\sum_{i=m_{k+1}^*}^{m} i^n. \tag{2}$$

The power sum $\sum\limits_{i=1}^{m} i^n$ is in fact well known to be

$$\sum_{\ell=1}^{n+1}\frac{(-1)^{n-\ell+1}B_{n-\ell+1}n!}{\ell!(n-\ell+1)!}m^{\ell}$$

from [**?**], and

$$\sum_{\ell=1}^{n+1}\binom{n+1}{\ell}\frac{B_{n-\ell+1}}{n+1}(m+1)^{\ell}$$

from [**?**], where B_r is the rth Bernoulli number. This ensures us a polynomial of degree $n+1$. Hence

$$\sum_{i=m_{k+1}^*}^{m} i^n = \sum_{i=1}^{m} i^n - \sum_{i=1}^{m_{k+1}^*-1} i^n = \sum_{\ell=1}^{n+1}\frac{(-1)^{n-\ell+1}B_{n-\ell+1}n!}{\ell!(n-\ell+1)!}(m^{\ell}-(m_{k+1}^*-1)^{\ell})$$

giving us a polynomial of degree $n+1$. Hence Equation **??** must indeed be a polynomial of degree $k+1$. The restriction of the domain to $[m^*_{k+1}, \infty)$ follows from only looking for non-zero values for $\delta(k,m)$ when $k+m \leq m \log_2 3$. Further, as a consequence of our construction we see that $m - (m^*_{k+1} - 1)$ is a factor of every term, and hence divides $\delta(k+1,m)$. ☐

Through application of this theorem, we get the following polynomials for small k:

$$\begin{aligned}
\delta(0,m) &= 1 \\
\delta(1,m) &= m-1 \\
\delta(2,m) &= \frac{1}{2}m^2 - \frac{1}{2}m - 3 = \frac{(m-3)(m+2)}{2} \\
\delta(3,m) &= \frac{1}{6}m^3 - \frac{19}{6}m - 5 = \frac{(m-5)(m+2)(m+3)}{6} \\
\delta(4,m) &= \frac{1}{24}m^4 + \frac{1}{12}m^3 - \frac{37}{24}m^2 - \frac{79}{12}m + 23 = \frac{(m-6)(m^3+8m^2+11m-92)}{24}
\end{aligned}$$

An implementation of how to use this in Mathematica is given in Appendix **??**. Now, instead of using the recursive definition we can generate the polynomial for $\delta(k,m)$ with the given k, and thusly plug in m. This new method is now quadratic. Using it to calculate $\Delta(14)$, $\Delta(15)$, $\Delta(16)$, and $\Delta(60)$ took 0.022 seconds, 0.031 seconds, 0.032 seconds, and 1.605 seconds on the math server, respectively, and 0.006 seconds, 0.007 seconds, 0.008 seconds, and 0.381 seconds on the iMac, respectively.[1]

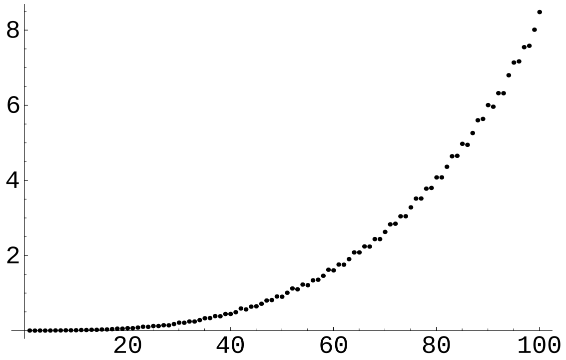

Figure 2: *Math server times for* $\Delta(1)$ *through* $\Delta(100)$.

---

[1] For those interested, the results are $\Delta(16) = 51{,}033$, $\Delta(17) = 108{,}950$, $\Delta(18) = 312{,}455$, and $\Delta(60) = 4{,}327{,}322{,}846{,}731{,}848{,}749{,}589{,}802$.

3.3 Determining how many numbers less than a bound have bad m-paths

One way that bad m-paths and the algorithm presented in Section **??** can prove useful is to generate bad m-paths, and determine what numbers less than a given bound are represented (an algorithm is given in Appendix **??**). For example, Figure **??** shows numbers less than $6{\cdot}2^{12} = 24,576$ represented by bad m-paths of lengths 1 through 66. Figure **??** shows the number of such numbers as we increase m. For this example we stop with an m of 66, because nothing less than 24,576 is represented by a bad m-path of length greater than 66.

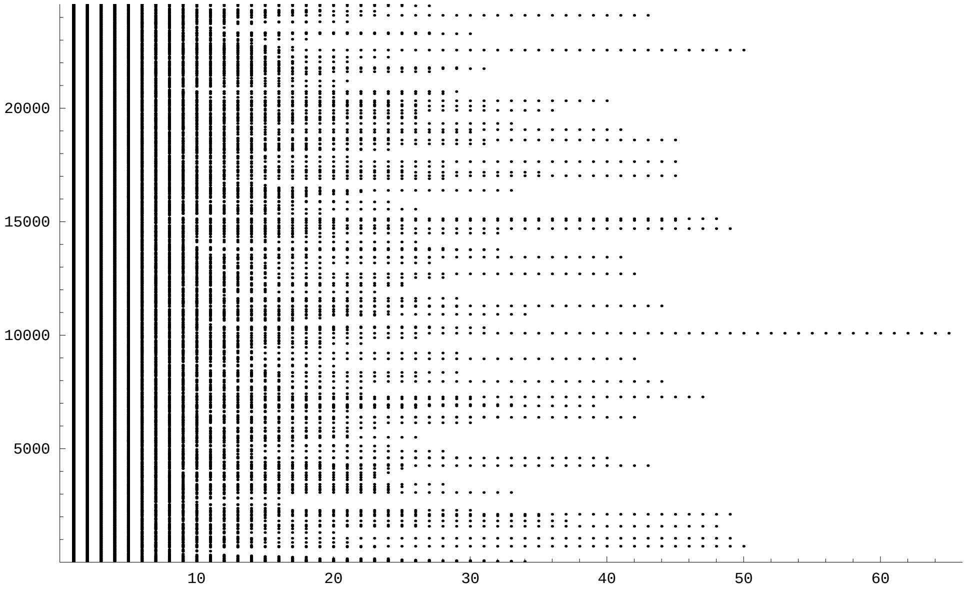

Figure 3: *Numbers less than 24,576 represented by bad m-paths of lengths 1 through 66.*

When trying to generate bad m-paths for numbers less than a bound B, it is of course useful to know how many such m-paths will be created. In this previous example we ruled out all numbers for $m = 67$. From Section **??** we see that this leads to potentially $5,547,417,647,649,572,762,833,925,633$ bad m-paths to deal with by the time we are done, which is clearly too many. Once an m-path α does not have any representatives less than a bound, then all m-paths that have α as an initial segment must be likewise and hence can be ignored. In fact, for a bound of 24,576 we only needed to generate 431 bad m-paths at its peak. We would like to have a better estimation of the total number of bad m-paths we will have to deal with, and Theorem **??** gives us such an estimate.

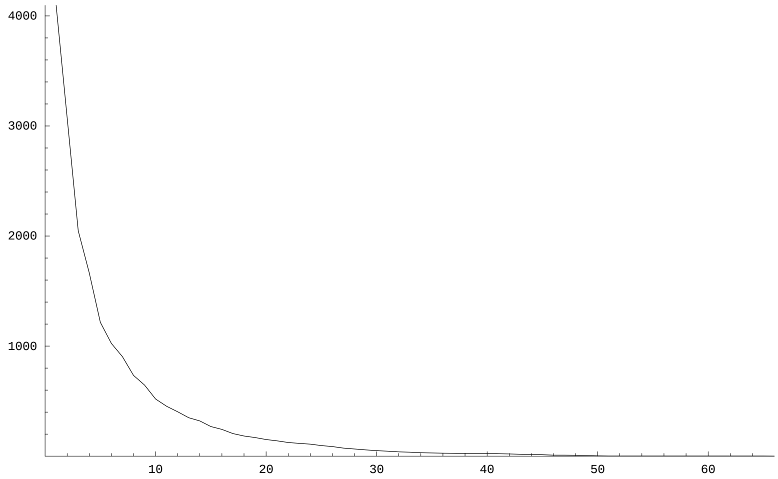

Figure 4: *Number of numbers less than 24,576 represented by bad m-paths of lengths 1 through 66.*

Theorem 6 *If $B, m \in \mathbb{N}$ and m is fixed, then*

$$\left|\{\alpha \in \mathcal{B}_m : (\exists a \in R^{(\alpha)})(a < B)\}\right| \leq A,$$

where

$$A = 2 + 2 \sum_{k=0}^{\lceil \log_2 \frac{B}{6} \rceil - 1} \left(\sum_{m=0}^{\lceil \log_2 \frac{B}{6} \rceil - k - 1} \delta(k, m) \left(\lfloor (m+1)(\log_2 3 - 1) \rfloor - k \right) \right).$$

Proof: Consider a bad m-path α of *any* length such that $6 \cdot 2^{Q_m} < B$. Such an m-path must have two or more representatives less than our bound B, and hence it is possible that each one of those representatives has an individual n-path for some $n > m$. To be able to talk about any kind of bound, it is necessary to ensure that the m-paths we are counting have at *most* two representatives less than our bound B. So now consider a bad m-path α of any length such that $6 \cdot 2^{Q_m} \geq B$. Such an m-path has at most two representatives less than B, just as we desire. However there are an infinite number of bad m-paths with $6 \cdot 2^{Q_m} \geq B$, making it rather difficult to find a finite bound.

In fact, we are not concerned about all of these m-paths. Instead look at all those bad m-paths α where m can range from 1 to $\lceil \log_2 \frac{B}{6} \rceil$, so that $6 \cdot 2^{Q_m} \geq B$ and $6 \cdot 2^{Q_{m-1}} < B$. These must each have at most two representatives less than B, and so we merely have to double this number to find a worse-case bound. We can do this by noting that the total number of bad m-paths (including the empty m-path) such that $6{\cdot}2^{Q_m}{<}B$ is

$$\sum_{q=0}^{\lceil \log_2 \frac{B}{6} \rceil - 1} \left(\sum_{m=0}^{q} \delta(q-m, m) \right).$$

On the other hand, we can also calculate the total number of bad m-paths such that $6{\cdot}2^{Q_{m-1}}{<}B$. We already know that $\delta(q{-}m, m)$ counts all bad m-paths such that $Q_m = q$. If we then try to extend the path to a bad $(m{+}1)$-path the possible values for k_{m+1} range from 1 to $\lfloor (m+1)\log_2 3 \rfloor {-} q$. That means the number of bad $(m+1)$-paths α such that $Q_m = q$ (the sum of the k_is for α_m) is $\delta(q-m, m)(\lfloor (m+1)\log_2 3 \rfloor - q)$.

It should be noted that this counts all bad m-paths stated before (and more), except for the empty 0-path. Thus, the total number of bad m-paths such that $6{\cdot}2^{Q_{m-1}}{<}B$ is at most

$$1 + \sum_{q=0}^{\lceil \log_2 \frac{B}{6} \rceil - 1} \left(\sum_{m=0}^{q} \delta(q-m, m)(\lfloor (m+1)\log_2 3 \rfloor - q) \right).$$

Our bound A is found by

$$\begin{aligned}
\frac{A}{2} &= 1 + \sum_{q=0}^{\lceil \log_2 \frac{B}{6} \rceil - 1} \left(\sum_{m=0}^{q} \delta(q-m, m)(\lfloor (m+1)\log_2 3 \rfloor - q) \right) - \sum_{q=0}^{\lceil \log_2 \frac{B}{6} \rceil - 1} \left(\sum_{m=0}^{q} \delta(q-m, m) \right) \\
&= 1 + \sum_{q=0}^{\lceil \log_2 \frac{B}{6} \rceil - 1} \left(\sum_{m=0}^{q} \delta(q-m, m)(\lfloor (m+1)\log_2 3 \rfloor - q - 1) \right) \\
&= 1 + \sum_{k=0}^{\lceil \log_2 \frac{B}{6} \rceil - 1} \left(\sum_{m=0}^{\lceil \log_2 \frac{B}{6} \rceil - k - 1} \delta(k, m)(\lfloor (m+1)\log_2 3 \rfloor - k - (m+1)) \right) \\
&= 1 + \sum_{k=0}^{\lceil \log_2 \frac{B}{6} \rceil - 1} \left(\sum_{m=0}^{\lceil \log_2 \frac{B}{6} \rceil - k - 1} \delta(k, m)(\lfloor (m+1)(\log_2 3 - 1) \rfloor - k) \right).
\end{aligned}$$

So finally we get

$$A = 2 + 2 \sum_{k=0}^{\lceil \log_2 \frac{B}{6} \rceil - 1} \left(\sum_{m=0}^{\lceil \log_2 \frac{B}{6} \rceil - k - 1} \delta(k, m)(\lfloor (m+1)(\log_2 3 - 1) \rfloor - k) \right)$$

where we make a change of $k = q - m$. $\square$

Through testing it is apparent that this bound is typically not very good. For example, the largest number of bad m-paths with representatives less than $6{\cdot}2^v$, with v ranging from 1 to 12, is stated below in Table **??**, along with the bounding given by Theorem **??**.

An interesting, and completely unproven here, way to estimate the total number of bad m-paths at any given m is as follows. If we assume that representatives of length m and $Q_m = m + k$ are nicely distributed between 0 and $6{\cdot}2^{m+k}$, then it must be that $\zeta(k, m, B) =$ the number of bad m-paths with representatives less than a bound B is approximately $\frac{B}{3\cdot 2^{m+k}}\delta(k, m)$ when $B < 6{\cdot}2^{k+m}$, and $\delta(k, m)$ otherwise. We can therefore try to approximate this with

$$\zeta(k, m, B) = \begin{cases} \delta(k, m), & 6{\cdot}2^{m+k} < B \\ \frac{B}{3\cdot 2^{m+k}}\delta(k, m), & \text{otherwise} \end{cases}$$

Table 3: *Maximum number of m-paths with representatives less than B.*

B	$6{\cdot}2^1$	$6{\cdot}2^2$	$6{\cdot}2^3$	$6{\cdot}2^4$	$6{\cdot}2^5$	$6{\cdot}2^6$	$6{\cdot}2^7$	$6{\cdot}2^8$	$6{\cdot}2^9$	$6{\cdot}2^{10}$	$6{\cdot}2^{11}$	$6{\cdot}2^{12}$
Empirical	2	2	2	5	9	15	29	49	80	145	248	431
$\zeta(k, m, B)$	2	3	4	6.5	10	16	29	53.5	88.75	158.875	292.75	494.5
Theorem **??**	2	4	6	10	18	30	54	106	200	362	648	1170

We thus know by Theorem **??** that there are at most 5,030,569,550,933,804 bad m-paths for numbers less than 2^{60}, while using $\zeta(k, m, B)$ that the actual number is possibly closer to 951,447,911,908,357. Theorem **??** tells us we may have to at up to 2.62% of all m-paths representing numbers less than 2^{60}, whereas if ζ is accurate, we would only need to look at at most 0.495% of all such m-paths.

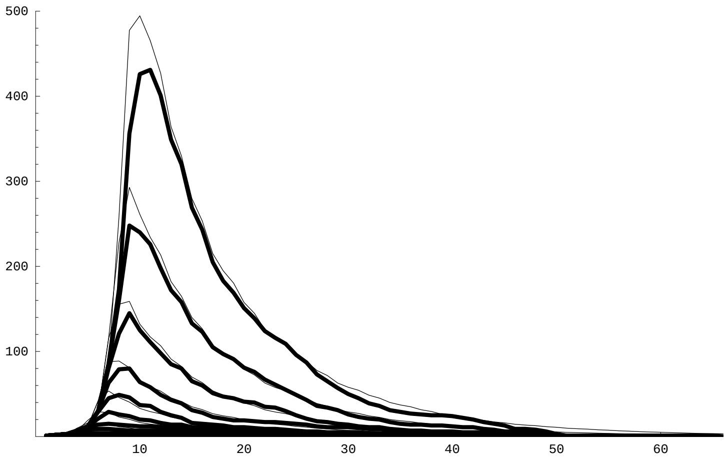

Figure 5: *Number of bad m-paths with ranging m (bold) compared with estimations with the ζ function, for bounds $6{\cdot}2^1$ to $6{\cdot}2^{12}$.*

APPENDIX

A Determining m-path representatives

An implementation of determining the minimal representatives of a given m-path, as discussed in Section **??**.

```
GetMPathRep[mpath_] :=
    Block[{a, b, c, p, rightPower, leftPower, i, theInv, k=0},
      leftPower = 2^mpath[[-1]];
      rightPower = 1;
      {a,b} = Switch[Mod[mpath[[-1]],6],
          0, {(leftPower*13-1)/3, (leftPower*7-1)/3},
          1, {(leftPower*11-1)/3, (leftPower*17-1)/3},
          2, {(leftPower-1)/3, (leftPower*13-1)/3},
          3, {(leftPower*5-1)/3, (leftPower*11-1)/3},
          4, {(leftPower*7-1)/3, (leftPower-1)/3},
          5, {(leftPower*17-1)/3, (leftPower*5-1)/3}
          ];
      For[i = -2, i >= -Length[mpath], i--,
        k += mpath[[i+1]];
        rightPower = rightPower*leftPower;
        leftPower = 2^mpath[[i]];
        If[0 == Mod[mpath[[i]], 2], c = a, c = b];
        If[0 == Mod[k, 2],
          theInv = 2*Floor[rightPower/3]+1,
          theInv = Ceiling[rightPower/3]
          ];
        Switch[Mod[mpath[[i]],6],
          0, {a, b} = (leftPower*{13, 7}-1)/3+
              leftPower*Mod[theInv*(c-{13, 7}), 6*rightPower],
          1, {a, b} = (leftPower*{11, 17}-1)/3+
              leftPower*Mod[theInv*(c-{11, 17}), 6*rightPower],
          2, {a, b} = (leftPower*{1, 13}-1)/3+
              leftPower*Mod[theInv*(c-{1, 13}), 6*rightPower],
          3, {a, b} = (leftPower*{5, 11}-1)/3+
              leftPower*Mod[theInv*(c-{5, 11}), 6*rightPower],
          4, {a, b} = (leftPower*{7, 1}-1)/3+
              leftPower*Mod[theInv*(c-{7, 1}), 6*rightPower],
          5, {a, b} = (leftPower*{17, 5}-1)/3+
              leftPower*Mod[theInv*(c-{17, 5}), 6*rightPower]
          ];
```

```
    ];
   {a, b}
   ];

GetMPathRep::usage = "Given an m-path, finds the two minimal representatives";
```

B Lower bounds on minimal representatives of bad m-paths

Given $A\in\mathbb{N}$ we can determine the largest possible m such that if $a\in\mathbb{N}$ and α is the m-path for a, then $a = 1$ or $a\geq A$, as determined in Theorem **??**.

```
(* How often to output current status *)
printHerePlease = 1000000;
(* Initial precision of Log[2,3] *)
theDecimalPlace = 16;
theBigLog = N[Log[2, 3], theDecimalPlace];
(* Our bound of interest *)
A = 2^60;
(* Precision in calculating the top *)
topPrecision = 25;
(* How much precision the bound given requires *)
necessaryPrecision = Ceiling[Log[2^60]/Log[10]];
(* We will use to make sure we are properly accurate *)
calcPrecision = 10^(-theDecimalPlace + necessaryPrecision);
(* Some initial conditions in calculating the bound *)
j = 0;
theMax = 0;
theBottom = 1;
theTop = 0;
While[theMax < A,
  If[Mod[j, printHerePlease] == 0,
   Print[j, ":  ", theMax, " (", N[100*theMax/A, 2], "
  j++;
  theBottom = 2^(Ceiling[j*theBigLog] - j*theBigLog) - 1;
  (* Here we check if we must update the precision *)
  If[(theBottom - calcPrecision) <= 0,
   Print["Need better precision..."];
   While[(theBottom - calcPrecision) <= 0,
    theDecimalPlace++;
    calcPrecision = 10^(-theDecimalPlace + necessaryPrecision);
    theBigLog = N[Log[2, 3], theDecimalPlace];
    theBottom = 2^(Ceiling[j*theBigLog] - j*theBigLog) - 1;
    ];
```

```
      Print["Precision required for j=", j, ":  ", theDecimalPlace];
      ];
    theTop += N[2^(Floor[(j - 1)*theBigLog] - j*theBigLog), topPrecision];
    theMax = Max[theMax, Floor[theTop/theBottom]];
    ];
j--
Print["The final answer is ", j, "."];
```

C The number of bad m-paths for a fixed m

An implementation of δ and Δ, as discussed in Section **??**.

```
δ[k_, m_] :=
    If[k < 0 || k > m (Log[2, 3] - 1),
      0,
      If[k == 0,
        1,
        (* Here is where we generate the coefficients of our polynomial *)
        Block[{A, B, mStar, i, n, q},
          A = {1};
          For[q = 1, q <= k, q++,
            mStar = Ceiling[q/(Log[2, 3] - 1)];
            B = Table[
                Sum[
                (A[[n + i]]*(-1)^n*BernoulliB[n]*(n + i - 1)!)/(n!*i!),
                {n, 0, q - i}
                ],
                {i, 1, q}
                ];
            A = Prepend[B, -B.Table[(mStar - 1)^n, {n, 1, q}]];
            ];
          (* Calculate actual answer after the polynomial has been created *)
          A.Table[m^n, {n, 0, k}]
          ]
        ]
      ];
δ::usage =
    "Calculates the number of bad m-paths such that the sum of the k's is k+m.";

Δ[n_] := δ[Floor[(n + 1)(Log[2, 3] - 1)], n + 1];

Δ::usage = "Calculates the number of bad m-paths.";
```

D Determining how many numbers less than a bound have bad m-paths

An implementation of β for determining an upper bound on the greatest possible number of m-paths less than a bound B, as discussed in Section **??**.

```
β[B_] :=
    Block[{m, A, C, mStar, i, n, q},
      A = {1};
      answer = 1;
      mStar = 0;
      For[k = 0, k < Log[2, B/6], k++,
        For[m = mStar, m < Log[2, B/6] - k, m++,
          answer +=
              A.Table[If[n > 0, m^n, 1], {n, 0,
                      k}]*(Floor[(m + 1)*Log[2, 3/2]] - k);
          ];
        mStar = Ceiling[(k + 1)/(Log[2, 3] - 1)];
        C = Table[
            Sum[(A[[n + i]]*(-1)^n*BernoulliB[n]*(n + i - 1)!)/(n!*i!), {n, 0,
                k - i + 1}], {i, 1, k + 1}];
        A = Prepend[C, -C.Table[(mStar - 1)^n, {n, 1, k + 1}]];
        ];
      2 answer
      ];
β::usage =

    "Determines an upper bound on the greatest number of m-paths with a representative

less than B.";
```

E Sample code for creating bad m-paths with representatives less than a given bound

By generating bad m-paths with representatives less than a bound, we may limit or even completely eliminate potential numbers to be checked for converging to 1. Possible variables are the final length of the bad m-paths, the bound that representatives should be below to be considered, the potential step size in creating the m-paths, and finally the name to use for temporary files created to save memory.

```
GBMPH[mGiven_, mUsed_, kUsed_] := If[mUsed == mGiven,
  {},
  Block[{k, j, temp, answer = {}},
   If[mGiven > mUsed + 1,
    For[k = 1, k < ((mUsed + 1)* MLC - kUsed), k++,
     temp = GBMPH[mGiven, mUsed + 1, kUsed + k];
     answer =
      Join[answer, Table[Join[{k}, temp[[j]]], {j, 1, Length[temp]}]];
     ],
    For[k = 1, k < ((mUsed + 1) MLC - kUsed), k++,
     answer = Append[answer, {k}];
     ]
    ];
   answer
   ]
  ];
GBMPH::usage =
  "Get Bad M Path Helper returns a list of bad m-paths of length
(mGiven-mUsed).  This means we need to know how many k_i have been used.";

GetBadMPaths[mGiven_, upperLimit_, stepSize_, theFileName_] :=
  Block[{i = Min[mGiven, stepSize], answer = {}, thisMP, temp, thisPathRep,
    oldLimit, j, k, n, p, trueStep, readFrom, writeTo, thatMP},
   (* First stepSize (or less) *)
   temp = GBMPH[i, 0, 0];
   Print[i, ":  1 of 1 - ", Length[temp]];
   For[j = 1, j ≤ Length[temp], j++,
    k = Sum[temp[[j]][[n]], {n, 1, i}];
    thisPathRep = Min[GetMPathRep[temp[[j]]]];
    If[thisPathRep < upperLimit,
     thisPathRep >>>
      theFileName <> "." <> ToString[i] <> "." <> ToString[k - i + 1] <>
       ".txt",
     openTo =
      OpenAppend[
        theFileName <> "." <> ToString[i] <> "." <>
         ToString[k - i + 1] <> ".txt"];
     Close[openTo]
     ];
    ];
   (* It is essential to try and save as much memory as we can,
because this can and will balloon *)
   Clear[temp];

   (* Next bunches of stepSize *)
   While[i < mGiven,
    If[i <= mGiven - stepSize,
```

```
    i = i + stepSize;
    trueStep = stepSize,
    trueStep = mGiven - i;
    i = mGiven
    ];
  oldLimit = (i - trueStep)*MLC;
  For[k = i - trueStep, k <= oldLimit, k++,
    Clear[temp];
    readFrom =
      OpenRead[
          theFileName <> "." <> ToString[i - trueStep] <> "." <>
            ToString[k - i + trueStep + 1] <> ".txt"];
    thatMP = Read[readFrom, RecordSeparators→{"\n"}];
    While[Head[thatMP] == Integer,
      thatMP = GetMPath[thatMP, i - trueStep];
      If[Length[temp] == 0,
        temp = GBMPH[i, i - trueStep, k];
        Print[i, ":  ", k - i + trueStep + 1,
          " of ", Floor[oldLimit - i + trueStep + 1],
          " - ", δ[k, (i - trueStep)]*Length[temp]];,
        ];
      For[j = 1, j <= Length[temp], j++,
        thisMP = Join[thatMP, temp[[j]]];
        n = Sum[thisMP[[p]], {p, -trueStep, -1}];
        thisPathRep = Min[GetMPathRep[thisMP]];
        If[thisPathRep < upperLimit,
          thisPathRep >>>
            theFileName <> "." <> ToString[i] <> "." <>
              ToString[k + n - i + 1] <> ".txt",
          openTo =
            OpenAppend[
                theFileName <> "." <> ToString[i] <> "." <>
                  ToString[k + n - i + 1] <> ".txt"];
          Close[openTo]
          ];
        ];
      thatMP = Read[readFrom, RecordSeparators→{"\n"}];
      ];
    Close[readFrom];
    DeleteFile[
        theFileName <> "." <> ToString[i - trueStep] <> "." <>
          ToString[k - i + trueStep + 1] <> ".txt"];
    ];
  ];
Print["Completed"];
oldLimit = mGiven*MLC;
For[k = mGiven, k < oldLimit, k++,
  readFrom =
```

```
          OpenRead[
              theFileName <> "." <> ToString[mGiven] <> "." <>
                ToString[k - mGiven + 1] <> ".txt"];
        thatMP = Read[readFrom, RecordSeparators→{"\n"}];
        While[Head[thatMP] == Integer,
          thatMP = GetMPath[thatMP, mGiven];
          answer =
            Join[answer, Select[GetMPathRep[thatMP], # < upperLimit &]];
          thatMP = Read[readFrom, RecordSeparators→{"\n"}];
          ];
        Close[readFrom];
        ];
      (* Cleanup files *)
      For[k = mGiven, k < oldLimit, k++,
        DeleteFile[
            theFileName <> "." <> ToString[mGiven] <> "." <>
              ToString[k - mGiven + 1] <> ".txt"];
        ];
      Sort[answer]
      ];
GetBadMPaths::usage =

    "Generates bad m-paths of length mGiven with representatives less than upperLimit.";
```

References

[1] Lagarias, J.C., *The* $3x + 1$ *problem and its generalizations*, American Mathematical Monthly, **92** (1985), No. 1, 3-23.

[2] Roosendaal, Eric, *On the* $3x + 1$ *problem*, [Online] available `http://personal.computrain.nl/eric/wondrous/`, November 23, 2004

[3] Sinai, Ya. G., *The* $(3x+1)$ *Problem*, Communications on Pure and Applied Mathematics, **LVI** (2003), 1016-1028.

[4] *Power Sum – from MathWorld*, [Online] available `http://mathworld.wolfram.com/PowerSum.html`, November 23, 2004.

[5] *Sum of* r*th powers of the first* n *positive integers*, [Online] available `http://planetmath.org/encyclopedia/SumOfKthPowersOfTheFirstNPositiveIntegers.html`, November 23, 2004.