The Inner Structure of the Collatz Iteration Sequence

Abstract

We present a new Collatz Structure Table which guarantees that any value within the table iterates down to 1 in accordance with the Collatz Conjecture. Proving that all natural numbers are within the bounds of this new table is considerably more difficult.

We present a new Collatz Rank Table which explicitly considers every odd number, and shows their first Collatz sequence steps to the next odd number. This table has mathematically elegant, and regular (repeating) features.

We can now (v1.50) show where the Rank values occur within the Structure Table.

By using features from both tables we can prove that either there are no exceptions at all to the Collatz Conjecture, or there is an infinity of infinite chains of counter-examples to the Collatz Conjecture.

Other names for this same problem / sequence include:

> Ulam conjecture, Kakutani’s problem, Thwaites conjecture,
> Hasse’s algorithm, Syracuse problem, 3n + 1 problem,
> 3x + 1 problem, Hailstone sequence.

Introduction

Much has been written about the Collatz iteration sequence. We assume that readers fall into two broad camps: (1) those who have barely heard of it, and (2) those who know it quite well.

For those who have barely heard of it, we have provided a very readable introduction, written primarily for school children.[1]

For those who know it quite well, a full historical context and prior work is either redundant, or can readily be found online. It is also redundant to quote work which was not consulted in the course of writing this paper.

Since there are subtly different variants of the Collatz sequence in the literature, we define the variant being used here for the sake of clarity.

- Start from any positive integer
- **If** the value is even then divide it by two,
  else multiply it by 3 and add 1.
- Repeat the previous step if the resultant value is greater than one.

We are led to believe that professional mathematicians dislike explanatory drawings as a matter of principle. However, we know for a fact that professional engineers are very fond of explanatory drawings, even if these may be considered as a mental crutch by giant-brained mathematicians. We therefore make no apologies for using such a drawing as the opening explanatory tool for this paper.

[1] *Introduction to Convergence of the Collatz Sequence*, 2021, Green, L.O.

It will be highly beneficial to have both the text and the drawing overleaf visible at the same time, either by having two instances of a PDF viewer on a dual-monitor computer, or even going ‘old-school’ and printing the drawing out, especially on A3 paper.

Collatz Structure Table

L0 is the terminating sequence. Note that L5 columns values 'jump over' the darker blue L4 columns to get to the inner L4 columns. L5 4437 ↗ L4 13312.

								L5
✗	L6	141984	70992	35496	17748	8874	►	4437
			►		►			
	L6	35488	17744	8872	4436	2218	►	1109
			►		►			
	L6	8864	4432	2216	1108	554	►	277
			►		►			
✗	L6	2208	1104	552	276	138	►	69
			►		►			
	L6	544	272	136	68	34	►	17
		▲		▲		▲		
	L7	181		45		11		

	✗						✗
	L4		L4		L4		L4
►	3072		13312		54272		218112
	1536	▼	6656	▼	27136	▼	109056
►	768		3328		13568		54528
	384	▼	1664	▼	6784	▼	27264
►	192		832		3392		13632
	96	▼	416	▼	1696	▼	6816
►	48		208		848		3408
	24	▼	104	▼	424	▼	1704
►	12		52		212		852
	6		26		106		426
	▼		▼		▼		▼

	L5								
◄	9045	◄	18090	36180	72360	144720	289440	L6	✗
				◄		◄			
◄	2261	◄	4522	9044	18088	36176	72352	L6	
				◄		◄			
◄	565	◄	1130	2260	4520	9040	18080	L6	
				◄		◄			
◄	141	◄	282	564	1128	2256	4512	L6	✗
				◄		◄			
◄	35	◄	70	140	280	560	1120	L6	
			▲		▲		▲		
			23		93		373	L7	

L3	3		13		53		213
	▼		▼		▼		▼

		L0												
1	◄	2												
		4		L1										
	▲	8												
		16	◄	5	◄	10	20	40	80	160	320	640	L2	
	▲	32					◄		◄		◄			
		64	◄	21	◄	42	84	168	336	672	1344	2688	L2	✗
	▲	128					◄		◄		◄			
		256	◄	85	◄	170	340	680	1360	2720	5440	10880	L2	
	▲	512					◄		◄		◄			
		1024	◄	341	◄	682	1364	2728	5456	10912	21824	43648	L2	
	▲	2048					◄		◄		◄			
		4096	◄	1365	◄	2730	5460	10920	21840	43680	87360	174720	L2	✗
	▲	8192					◄		◄		◄			
		16384	◄	5461	◄	10922	21844	43688	87376	174752	349504	699008	L2	
	▲	32768					◄		◄		◄			
		65536	◄	21845	◄	43690	87380	174760	349520	699040	1398080	2796160	L2	

Key to the Collatz Structure Table

Throughout this paper we will make *claims* (assertions) which will typically be justified immediately below in brown text if the explanation is adequately brief. This enables the reader to quickly scan over uninteresting claims to find 'the good stuff'.

The claims are numbered so they can be easily quoted in case any are disputed, inadequately convincing, or circular.

The Structure Table is defined by iteration ***Levels***, ranging from L0 to some indefinitely large value. L0 is the iteration Level closest to the termination point, 1. It does not take any great intelligence to realise that the L0 values are all powers of 2.

Clearly every Level shown extends infinitely in the increasing direction, but we have not explicitly shown ellipses at these points.

C01: If you start from any L0 value it is a **known** direct path to termination.

Any 2^k value reaches 1 after k divide-by-2 steps.

C02: Whilst there are infinitely many of these 'easy' known starting points, they have a very low density.

In a range up to some huge value, H, we have $\log_2(H) / H$ per-unit L0 starting points, which is *almost none* in this asymptotic sense.

C03: Unless you actually start from an L0 value, the **only** way to get to an L0 value is from an L1 value.

$(3\times + 1)$ from all integer values does not produce all possible 2^k values. The L1 set is defined as *all those* that do produce 2^k values.

C04: Only half of L0 values can be reached from L1.

$(3\times + 1)$ from all integer values produces all possible 2^{2k} values. The L1 set is *defined* as all those that produce 2^{2k} values.

C05: Unless you actually start from that specific L1 value, it cannot be reached except from an L2 value.

L1 values are all odd. The only way to reach an odd value is from an even value using the divide-by-two operation. L2 values are all the power-of-two multiples of their respective L1 values.

C06: All L1 values are reachable.

All L1 values can be used as the starting value. Each L1 value is the *root* of its own power-of-two multiplied *chain*, known as L2.

C07: All L2 values are power-of-two multiples of their respective L1 *root* values.

By definition.

C08: One third of L2 chains cannot be reached from L3.

These L2 values are highlighted by a darker blue shade, and with a red cross on that row. You can start on say 42, or you can get to 42 by starting on 168, then dividing down to 42.

C09: 'Unreachable' even-Level chains are all multiples of 3.

Even-Levels can only be reached by a $(3\times + 1)$ step from an odd Level (unless you start from within that even Level. There is no possible integer $\boldsymbol{n}$ such that $(3\boldsymbol{n} + 1)$ is evenly divisible by 3.

C10: A value, V, in an odd Level can be reached in exactly two ways.

This is a generalisation of **C05** above.

(1) You start from that value.

(2) A divide-by-two step from 2V in the even-Level above.

C11: Any even-Level, LE, (E = 2k) contains only even numbers.

This is how the table is structured.

Watch Out! We are using the capital letters **O** and E for odd and even numbers respectively. L0 and **LO** are different things, and easy to confuse. Hopefully the context will be a clue, as will the colour!

C12: Any odd numbered Level **LO**, (**O** = 2k+1) contains only odd numbers.
This is how the table is structured.

C13: If an iteration sequence starts on, or reaches, an even Level, the sequence can *always* iterate down to an odd value in an odd Level with the next lower index number (eg L32 → L31)
Even values can always be divided by 2 until the result is odd.

C14: If an iteration sequence starts on, or reaches, an odd value (necessarily in an odd Level), the sequence will always iterate down to an even value (necessarily in an even Level) with the next lower index number (eg L31 → L30)
An odd value, (2k + 1), is transformed into an even value by the (3× + 1) step. $(2k + 1) \to 3(2k + 1) + 1 = 6k + 4 = 2(3k + 2)$
The iteration is 'down' in the sense that it is closer to termination, and has a lower L-index. The actual value is of course larger.

C15: The value 1, the termination point, is not considered to be on a Level.
This is just a definition. Originally 1 was in the L0 set, but that gives an odd value in an even level, which is against the general 'evenness rule' for an even Level.

Definitions: We define any value in an odd-Level as the ***root*** of the infinite ***chain*** of power-of-two multiples of it which occur in the next Level up. So, for example, 5, which occurs in L1 is the *root* of the *chain* { 10, 20, 40, 80, 160, 320, 640, … } in L2.

Infinite chains also occur within odd Levels, related by the (4× + 1) recurrence relation, for example: { **3**, 13, 53, 213, 853, 3413, … }.

The lowest value in such a chain is also called a *root*.

Key Claims

You should appreciate that the table becomes increasingly difficult to draw as the number of Levels increases. In our drawing, L3 is very poorly represented as we should have five L3 rows and we only have one.

However, the values in each set are easily computer generated, and we list some of them out in a later section.

Below are the most difficult issues to resolve:

C20: Any natural number as a starting point can be found on this table, provided the table has been drawn with an adequate number of Levels.
At this stage it is not clear if or how we could justify this claim.

C21: Any natural number on this table is unique (never duplicated).
We justify this later, starting with claim **C30**.

C22: Having found the unique starting point within the table, there is a guaranteed monotonic progression from one Level to the next, eventually arriving at the termination point, 1.
Monotonic progression is built-in to the fabric of the table. See also **C13**, **C14**. If **C20** is true then this claim is immediately true.

C23: All possible starting points terminate on 1 (The Collatz Conjecture).
This is simply a restatement of **C22**. The "wandering off to infinity" possibility is expressly excluded.

C24: No value within the table can form a looped sequence.
Having demonstrated a *monotonic* progression through the Levels in **C22**, it is not necessary to separately address loops.

Level L0

This is a very easy set to generate and count. Up to some definite value, H, we have $2^k \leq H$. $k = \lfloor \log_2(H) \rfloor$

> The funny brackets around the log( ) function above are called the *floor( )* function. For positive values it just means throw away any fractional part in the decimal value.

The count will be k as we do not include $2^0 = 1$ within L0.

However, we also need values in the table as intermediate points, since the Collatz sequence is known to internally generate values which exceed the initial starting value 'considerably'. (See the Appendix)

We therefore have two values to consider. Firstly we would ideally like an exact count of numbers in a table of starting values, in order to at least demonstrate that all starting values exist within the bounds of the table. Secondly we need some larger value to give confidence that all intermediate values have been considered. For this purpose we use $10H^2$, the requirement for which is demonstrated in the Appendix.

As an initial 'confidence builder' we will create a small table to handle all starting values up to $8^2 = 64$.

L0 = 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768.

The count is 6. $k = \lfloor \log_2(8^2) \rfloor = \lfloor 6.000 \rfloor = 6$

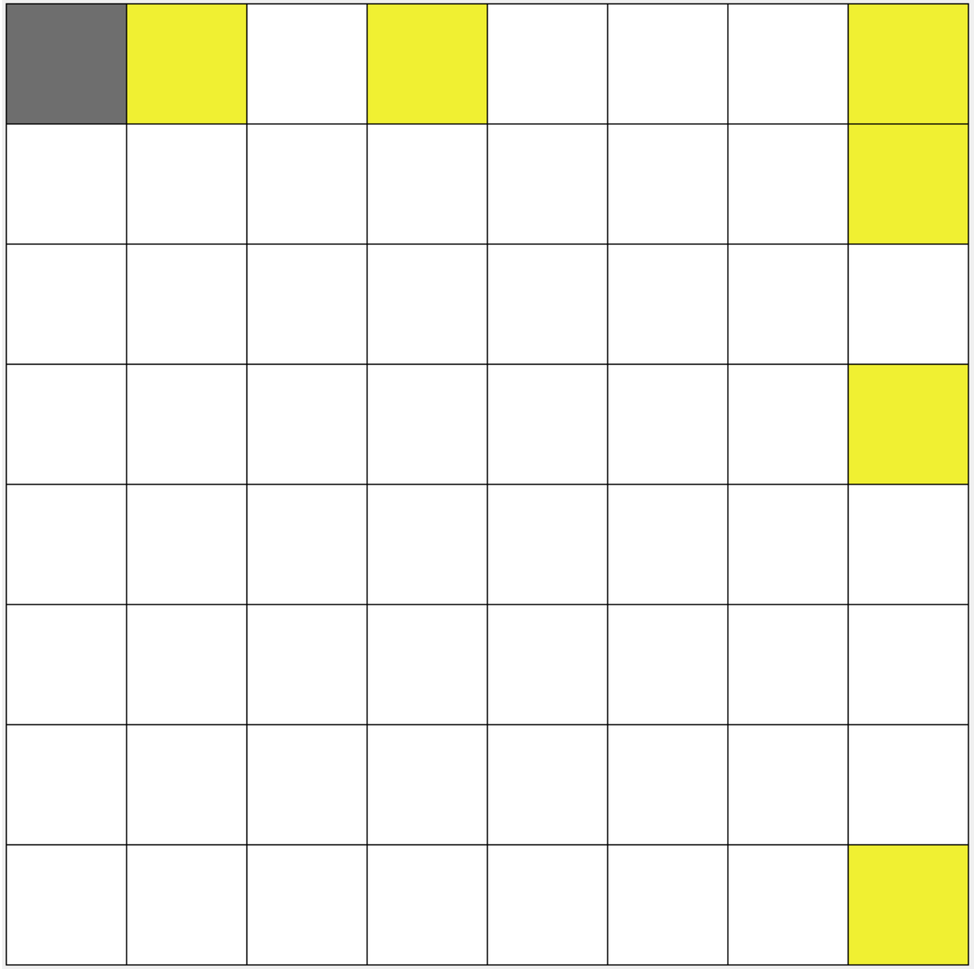
In the 8 x 8 table, these values are already very sparse.

Initially the visual presentation was done in MS Excel. It was then realised that only two colours could be created using the built-in *conditional field shading*. It was therefore necessary to write a C++ program from scratch to produce the presentations.

Number 1, the termination point, is at the top left in grey.

Number 9 would be on the left, one row down from the top.

L0 = 2, 4, 8, 16, 32, 64.



It should be evident that L0, L2, L4, … only contain even values (**C11**) Likewise L1, L3, L5, … only contain odd values (**C12**) It should also come as no surprise that an even L-index number corresponds to even values since the indexing was set up that way.

Uniqueness and coverage are the two key aspects of the Structure Table which need to be established. It is essential that every counting number up to some limit, H, is available exactly once within the Levels structure. Certainly we are 100% confident that within the L0 section there are no duplicated values, since we have a strictly increasing set of powers-of-two (**C31**). It remains to be shown that no such duplicated values occur in any other even-indexed Level (**C39**).

Level L1

L1 values are necessarily odd. There is a simple formula for their values:

$$\frac{2^{2k}-1}{3} \qquad \text{for } k > 1$$

Note that k > 1. For k = 1 we get an L1 value of 1, which would be a duplicate, and this is strictly forbidden.

There are two things to demonstrate here:

(1) $3|\left(2^{2k}-1\right)$ so all the values are integers [2]

The assertion 3 | (2^{2k} – 1) reads as "3 *evenly* divides (2^{2k} – 1)"

(2) $\frac{2^{2k}-1}{3}$ is odd, as required [3]

Both of these have been posed and answered on a maths puzzle website for school-kids, linked below. Both are also proved in **C60** further on.

In Level L1, and indeed in every odd Level, successive values in a chain can be found using (4× + 1) steps, as discussed in a later section.

$$4\times\left(\frac{2^{2k}-1}{3}\right)+1 = \left(\frac{2^{2k+2}-4}{3}\right)+1 = \left(\frac{2^{2k+2}-1}{3}\right)-\frac{3}{3}+1 = \left(\frac{2^{2k+2}-1}{3}\right)$$

We can see that the odd Levels have an 'every-other-value' connection to the even-Level below (closer to termination). Why?

[2] https://aplusclick.org/t.htm?q=10886
[3] https://aplusclick.org/t.htm?q=10913

Considering mod 3 values, we have exactly 3 possibilities: { 0, 1, 2 }.

We can represent some general value V, optimised for mod 3 arithmetic as V = 3k + r, where all symbols here and below are natural numbers (including zero).

For r = 0:

$V = 3k$
$2V = 6k = 3k_1$
$4V = 6k_1 = 3k_2$

Doubling, as happens in every even-Level set, does not change the mod 3 residue. A number which is divisible by 3, remains divisible by 3 after being doubled any number of times.

For r = 1:

$V = 3k + 1$
$2V = 6k + 2 = 3k_1 + 2$
$4V = 6k_1 + 4 = 3k_2 + (3 + 1) = 3(k_2 + 1) + 1 = 3k_3 + 1$

Doubling, as happens in every even-Level set, changes a mod 3 residue of 1 to 2, and back again, in an infinitely repeating pattern. Clearly any value of the form 3k + 1 is evenly divisible by 3 after you subtract 1 from it. It is also clear that any value of the form 3k is no longer divisible by 3 when you subtract one from it, as seen in **C09** earlier.

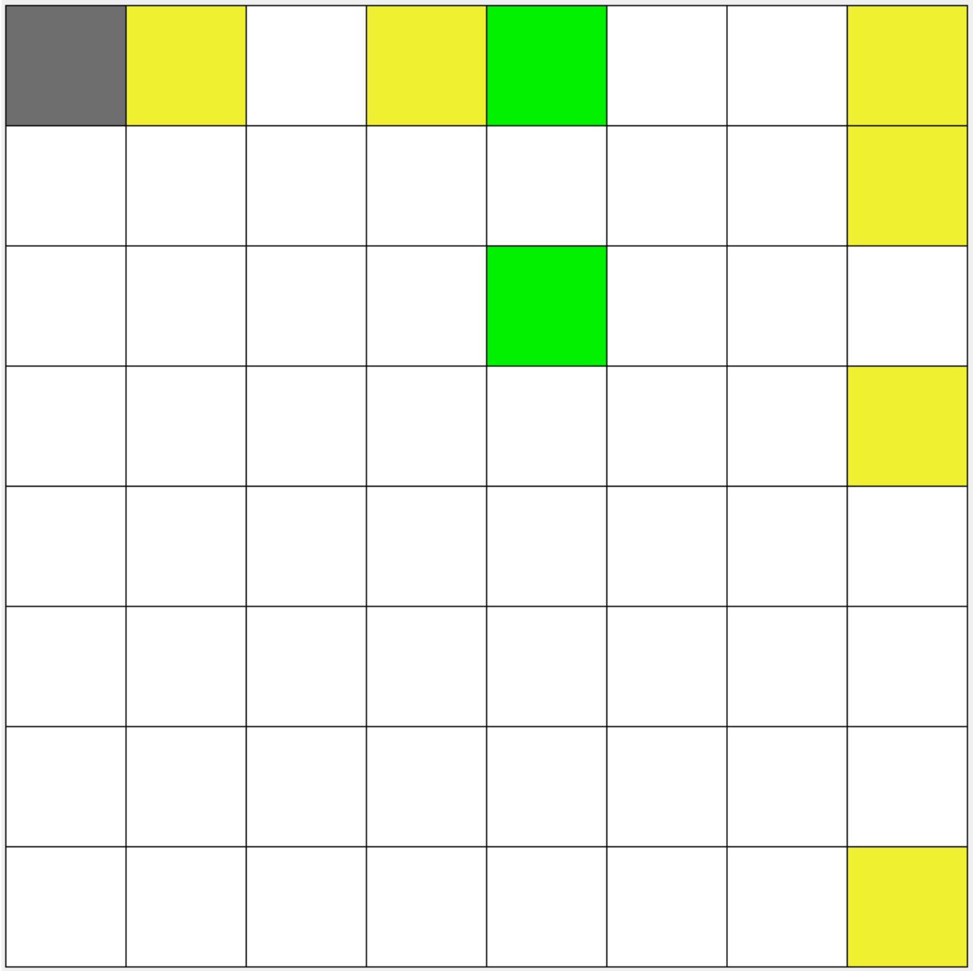
Asymptotically, there are half as many values in L1 as in L0.

There are actually only 2 L1 values in our 8 x 8 starting value grid.

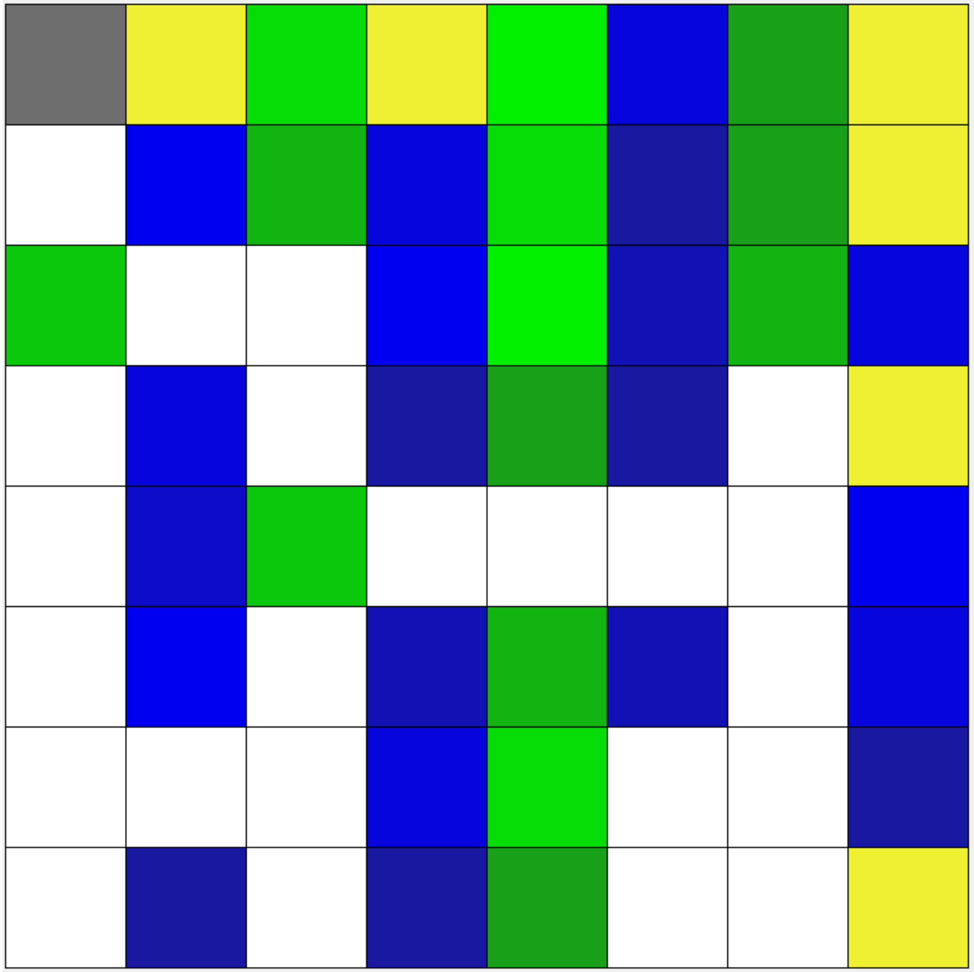
L1 = 5, 21, 85, 341, 1365, 5461.

L0 = 2, 4, 8, 16, 32, 64 (in yellow)

L1 = 5, 21 (in green)



Even up to L10 the grid is still not very full. Shades of green are odd-Level values. Shades of blue are even-Level values.



In the tabulation which follows, each even ***Level*** has been printed in powers-of-two ***chains***. L0 has exactly one chain. L1 has exactly one chain. All the rest have multiple chains. Odd Levels give rise to (4× + 1) chains.

First 7 Levels for a starting grid of 8 x 8

--------- LEVEL 0
2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768,

--------- LEVEL 1
5, 21, 85, 341, 1365, 5461,

--------- LEVEL 2
10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120, 10240, 20480, 40960,
42, 84, 168, 336, 672, 1344, 2688, 5376, 10752, 21504,
170, 340, 680, 1360, 2720, 5440, 10880, 21760,
682, 1364, 2728, 5456, 10912, 21824,
2730, 5460, 10920, 21840,
10922, 21844,

--------- LEVEL 3
3, 13, 53, 213, 853, 3413, 13653,
113, 453, 1813, 7253,
227, 909, 3637, 7281,

--------- LEVEL 4
6, 12, 24, 48, 96, 192, 384, 768, 1536, 3072, 6144, 12288, 24576,
26, 52, 104, 208, 416, 832, 1664, 3328, 6656, 13312, 26624,
106, 212, 424, 848, 1696, 3392, 6784, 13568, 27136,
426, 852, 1704, 3408, 6816, 13632, 27264,
1706, 3412, 6824, 13648, 27296,
6826, 13652, 27304, 27306,
226, 452, 904, 1808, 3616, 7232, 14464, 28928,
906, 1812, 3624, 7248, 14496, 28992,
3626, 7252, 14504, 29008,
14506, 29012,

454, 908, 1816, 3632, 7264, 14528, 29056,
1818, 3636, 7272, 14544, 29088,
7274, 14548, 29096,
14562, 29124,

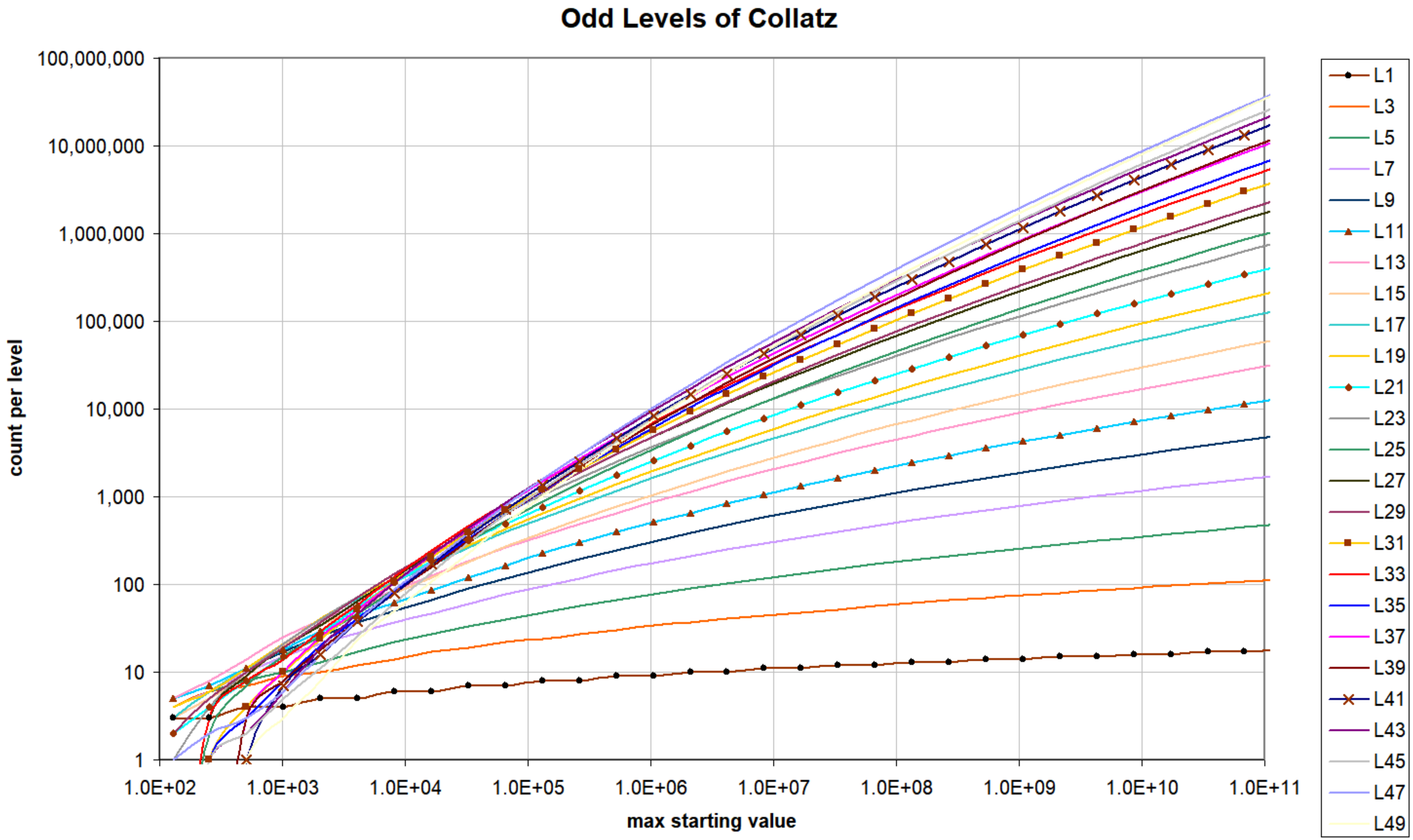
--------- LEVEL 5
17, 69, 277, 1109, 4437,
35, 141, 565, 2261, 9045,
1137, 4549,
2275, 9101,
75, 301, 1205, 4821,
2417, 9669,
4835,
151, 605, 2421, 9685,
4849,

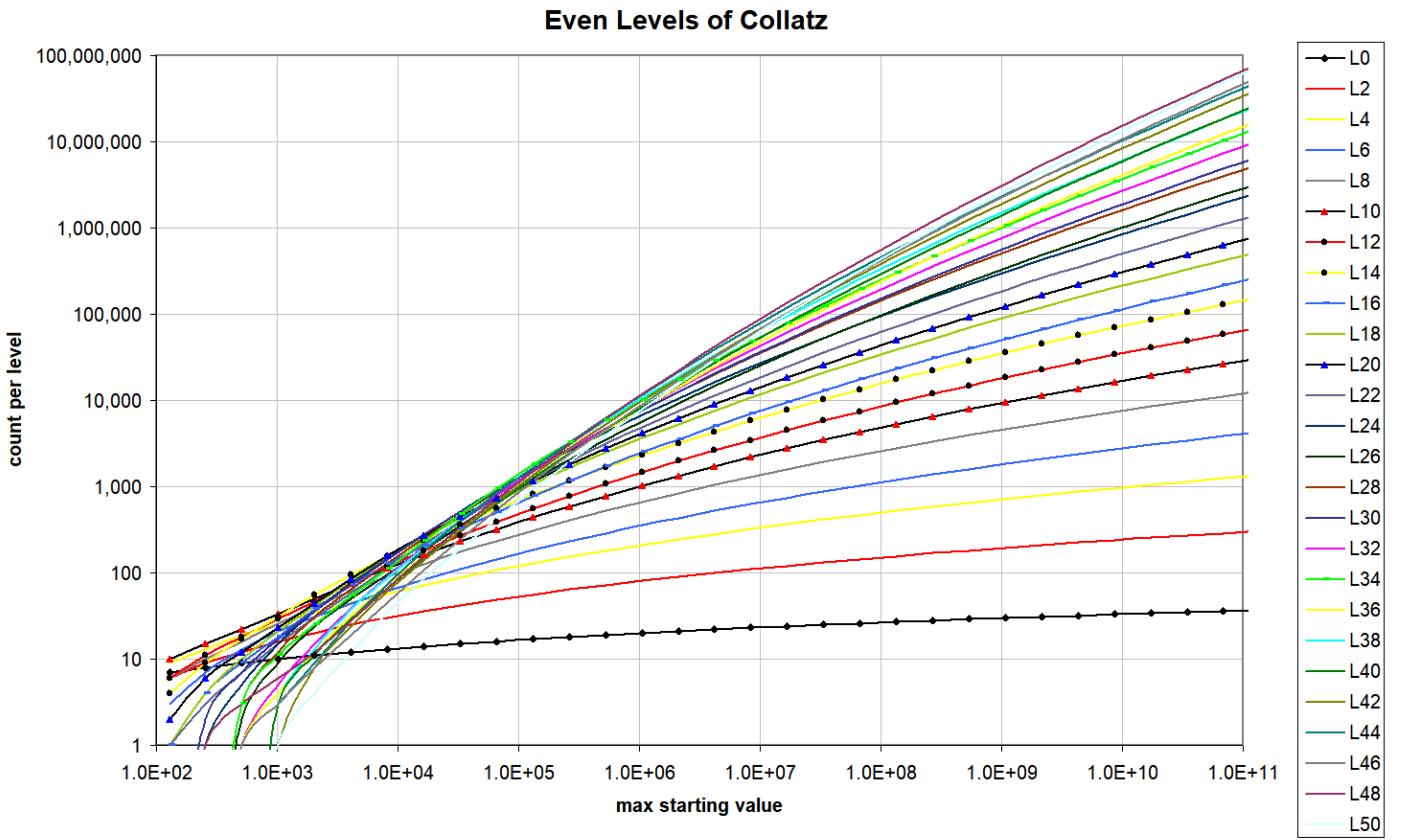
--------- LEVEL 6
34, 68, 136, 272, 544, 1088, 2176, 4352, 8704, 17408, 34816,
138, 276, 552, 1104, 2208, 4416, 8832, 17664, 35328,
554, 1108, 2216, 4432, 8864, 17728, 35456,
2218, 4436, 8872, 17744, 35488,
8874, 17748, 35496,
70, 140, 280, 560, 1120, 2240, 4480, 8960, 17920, 35840,
282, 564, 1128, 2256, 4512, 9024, 18048, 36096,
1130, 2260, 4520, 9040, 18080, 36160,
4522, 9044, 18088, 36176,
18090, 36180,
2274, 4548, 9096, 18192, 36384,
9098, 18196, 36392,
4550, 9100, 18200, 36400,
18202, 36404,
150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400,
602, 1204, 2408, 4816, 9632, 19264, 38528,
2410, 4820, 9640, 19280, 38560,
9642, 19284, 38568,
4834, 9668, 19336, 38672,
19338, 38676,
9670, 19340, 38680,
302, 604, 1208, 2416, 4832, 9664, 19328, 38656,
1210, 2420, 4840, 9680, 19360, 38720,
4842, 9684, 19368, 38736,
19370, 38740,
9698, 19396, 38792,

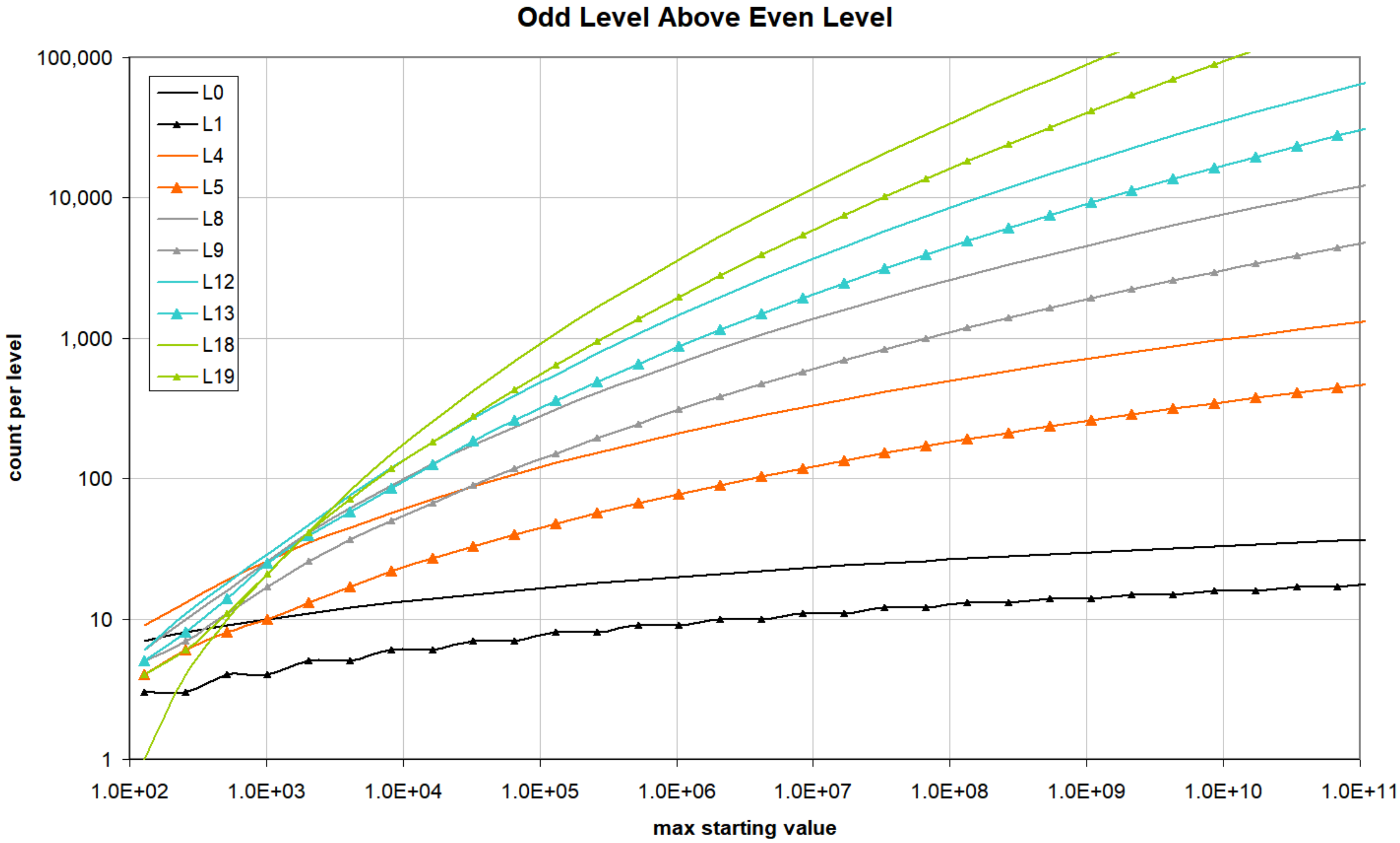
--------- LEVEL 7
11, 45, 181, 725, 2901, 11605,
369, 1477, 5909,
739, 2957, 11829,
23, 93, 373, 1493, 5973,
753, 3013, 12053,
1507, 6029, 6065,
3033, 12133,
6067,
401, 1605, 6421,
803, 3213, 12853,
1611, 6445,
3223, 12893,
201, 805, 3221, 12885,
403, 1613, 6453, 12913,
6465,

It should be noted that to save space on the odd Levels, not all values on the same line are in the same (4× + 1) chain.

Numbers in Levels




Even Levels of Collatz
count per level
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
max starting value
1.0E+02
1.0E+03
1.0E+04
1.0E+05
1.0E+06
1.0E+07
1.0E+08
1.0E+09
1.0E+10
1.0E+11
L0
L2
L4
L6
L8
L10
L12
L14
L16
L18
L20
L22
L24
L26
L28
L30
L32
L34
L36
L38
L40
L42
L44
L46
L48
L50



Whilst there are hardly any L0 starting values, each successive even Level has increasingly more starting values in it.

Above we show an even Level with the next Level up (which is odd) in the same colour, but with a triangular marker. There are roughly half as many starting values in the odd Level immediately above an even Level.

Uniqueness

C30: No value in an even Level (eg L32) can ever equal a value in any odd Level (eg L31, L39, L27).

An odd number cannot equal an even number, (C11, C12)

C31: No value is duplicated within the L0 Level itself.

The L0 Level consists of strictly increasing powers of 2.

C32: No value within an even Level is duplicated within that Level (unless there is a duplication in the odd Level from which it is formed).

Without loss of generality, pick an example even-Level such as L32. Each chain consists of 2^k multiples of the odd *root* of that *chain* in the L31 Level. Taking two distinct roots in the L31 Level, such as $\boldsymbol{p}$ and $\boldsymbol{q}$,

$\boldsymbol{p} \times 2^{\boldsymbol{m}} \neq \boldsymbol{q} \times 2^{\boldsymbol{n}}$ by the *Fundamental Theorem of Arithmetic*.

C33: Chains in even-numbered Levels cannot cross, intersect, or meet, regardless of them being within the same Level, or otherwise.

Working up from a root value using a multiply-by-two at each step, there is no possibility to go two separate ways. From any value $\boldsymbol{k}$, the $\times 2$ step is always $2\boldsymbol{k}$, which is then deterministically in the same chain, and within the same Level.

C34: No even-value is duplicated within the table unless an odd root value is duplicated within the table.

Taking two distinct roots anywhere within the table, such as $\boldsymbol{p}$ & $\boldsymbol{q}$, $\boldsymbol{p} \times 2^{\boldsymbol{m}} \neq \boldsymbol{q} \times 2^{\boldsymbol{n}}$ by the *Fundamental Theorem of Arithmetic*.

C35: No L0 value is duplicated within the table.

L0 values could only be duplicated in even-Levels (see **C30**). Even-Levels are odd-multiples of 2^n values.

$2^{\boldsymbol{m}} \neq (2\boldsymbol{k}+1) \times 2^{\boldsymbol{n}}$ for $\boldsymbol{k} > 0$

by the *Fundamental Theorem of Arithmetic*.

C36: All L1 values are unique within the L1 set.

L1 values are uniquely generated from the distinct and strictly increasing values in the L0 set.

C37: All L1 values are unique within the table.

L1 values are uniquely generated from the distinct and strictly increasing values in the L0 set. The L0 values are distinct within the table according to claim **C35**. There is no other way to create an L1 value, so the L1 values are unique within the table.

C38: All L2 values are unique within the table.

According to **C34**, all values in L2 are unique within the table since they are uniquely generated from L1 values which are themselves unique according to **C37**.

C39: All values within all Levels are distinct (unique) within the table.

L0 values are all unique within the table according to claim **C35**. All L1 values are distinct within the table as they are uniquely derived from L0. All L2 values are distinct within the table as they are uniquely derived from L1. Repeating this same process indefinitely, every higher Level is uniquely derived from the Level below. It means all values within the Levels are distinct.

This result means that the Uniqueness claim of **C21** is now justified.

Collatz Rank Table, R1 values

We temporarily put aside the Structure Table, and consider a new table, the Rank Table. We consider all odd numbers in the first column, and all (3× + 1) Collatz values generated from the odd numbers. We then divide by 2 in successive columns until an odd number is reached. It is this odd number which defines the rank of the *initial odd number*.

In this table a ***chain*** starts with an odd value, and is guaranteed to ***terminate*** in a *different* odd value, all on the same row of the table. Note that *terminate / termination* have a completely different meaning for this table compared to the Structure Table.

To be clear, starting with a 9 in the left-most column, the 9 is a Rank 2 (R2) value because the chain terminates at 7 in the R2 column.

Starting from 3, every other odd number is an R1 value, { 3, 7, 11, 15, 19, … }. R1 values are of the form (4k – 1) for k > 0.

C50: The Rank table has 100% coverage over the odd natural numbers.

The table is listed for all odd natural numbers.

The R1 termination values in the R1 column are separated by numerical gaps of 6, { 5, 11, 17, 23, 29, … }. R1 termination values are of the form (6k – 1) for k > 0.

We will use ↗ (up) to represent the Collatz (3× + 1) step, and ↘ (down) to represent a divide-by-two step.

C51: An R1 value is of the form (4k – 1) and terminates in the odd number (6k – 1), where k has the same value throughout.

(4k – 1) ↗ 3(4k – 1) + 1 = 12k – 3 + 1 = (12k – 2) ↘ (6k – 1)

The value (6k – 1) is clearly odd, so the (4k – 1) chain had only one possible divide-by-two step, and hence was in R1.

An R1 chain ends at a greater value than its start, since (6k – 1) > (4k – 1)

odd	3x+1	R1	R2	R3	R4	R5	R6	R7	R8
1									
3	10	5							
5	16	8	4	2	1				
7	22	11							
9	28	14	7						
11	34	17							
13	40	20	10	5					
15	46	23							
17	52	26	13						
19	58	29							
21	64	32	16	8	4	2	1		
23	70	35							
25	76	38	19						
27	82	41							
29	88	44	22	11					
31	94	47							
33	100	50	25						
35	106	53							
37	112	56	28	14	7				
39	118	59							
41	124	62	31						
43	130	65							
45	136	68	34	17					
47	142	71							
49	148	74	37						
51	154	77							
53	160	80	40	20	10	5			
55	166	83							
57	172	86	43						
59	178	89							
61	184	92	46	23					
63	190	95							
65	196	98	49						
67	202	101							
69	208	104	52	26	13				
71	214	107							
73	220	110	55						
75	226	113							
77	232	116	58	29					
79	238	119							
81	244	122	61						
83	250	125							
85	256	128	64	32	16	8	4	2	1
87	262	131							
89	268	134	67						
91	274	137							
93	280	140	70	35					
95	286	143							
97	292	146	73						
99	298	149							
101	304	152	76	38	19				
103	310	155							
105	316	158	79						
107	322	161							
109	328	164	82	41					
111	334	167							
113	340	170	85						
115	346	173							
117	352	176	88	44	22	11			
119	358	179							
121	364	182	91						
123	370	185							
125	376	188	94	47					
127	382	191							
129	388	194	97						
131	394	197							
133	400	200	100	50	25				
135	406	203							

Collatz Rank Table, R2 values

Note that a Rank Table chain is distinct from a Sequence Table L-chain. In the Rank Table, a chain starts from an odd number, gets boosted by the Collatz (3× + 1) operation, gets all the factors of two divided out, and ends up at a different odd value to the starting value. In an L-set, values are either all even or all odd (**C11**, **C12**).

An R2 chain also has a systematic repeating pattern. The starting R2 values are { 9, 17, 25, 33, 41, 49, 57, … }, being of the form (8k + 1). The resultant values are { 7, 13, 19, 25, 31, 37, 43, 49, … }, having the value (6k + 1).

C52: An R2 value is of the form (8k + 1) and terminates in the odd number (6k + 1), where k has the same value throughout.

(8k + 1) ↗ 3(8k + 1) + 1 = (24k + 4) ↘ (12k + 2) ↘ (6k + 1)

The value (6k + 1) is clearly odd, so the (8k + 1) chain had exactly two divide-by-two steps, and hence was in R2.

The value at the end of an R2 chain is always lesser than at the start since (6k+1) < (8k+1)

It should be observed that developed values within the Rank Table are not unique. (The '3' chain contains {10, 5} and so does the '13' chain.)

C53: The Rank of an odd value is uniquely determined within the Collatz Rank Table.

Each odd value occurs exactly once in the odd number column. The developed sequence of even numbers must necessarily terminate on an odd number when the number of factors of two has been depleted. This occurs exactly once within the table.

odd	3x+1	R1	R2	R3	R4	R5	R6	R7	R8
1									
3	10	5							
5	16	8	4	2	1				
7	22	11							
9	28	14	7						
11	34	17							
13	40	20	10	5					
15	46	23							
17	52	26	13						
19	58	29							
21	64	32	16	8	4	2	1		
23	70	35							
25	76	38	19						
27	82	41							
29	88	44	22	11					
31	94	47							
33	100	50	25						
35	106	53							
37	112	56	28	14	7				
39	118	59							
41	124	62	31						
43	130	65							
45	136	68	34	17					
47	142	71							
49	148	74	37						
51	154	77							
53	160	80	40	20	10	5			
55	166	83							
57	172	86	43						
59	178	89							
61	184	92	46	23					
63	190	95							
65	196	98	49						
67	202	101							
69	208	104	52	26	13				
71	214	107							
73	220	110	55						
75	226	113							
77	232	116	58	29					
79	238	119							
81	244	122	61						
83	250	125							
85	256	128	64	32	16	8	4	2	1
87	262	131							
89	268	134	67						
91	274	137							
93	280	140	70	35					
95	286	143							
97	292	146	73						
99	298	149							
101	304	152	76	38	19				
103	310	155							
105	316	158	79						
107	322	161							
109	328	164	82	41					
111	334	167							
113	340	170	85						
115	346	173							
117	352	176	88	44	22	11			
119	358	179							
121	364	182	91						
123	370	185							
125	376	188	94	47					
127	382	191							
129	388	194	97						
131	394	197							
133	400	200	100	50	25				
135	406	203							

Collatz Rank Table, R3 values

R3 values are { 13, 29, 45, 61, 77, 93, … }, all of the form (16k – 3).

R3 chains terminate in { 5, 11, 17, 23, 29, … }, all having the value (6k – 1).

C54: An R3 value is of the form (16k – 3) and terminates in the odd number (6k – 1), where k has the same value throughout.

(16k – 3) ↗ 3(16k – 3) + 1 =

(48k – 8) ↘ (24k – 4) ↘ (12k – 2) ↘ (6k – 1)

The (6k – 1) terminating value is clearly odd after exactly 3 divide-by-two operations, so the initial (16k – 3) value was in R3.

C55: The Rank Table has 100% implicit (but not explicit) coverage over the natural numbers.

C50 shows explicit coverage for all odd natural numbers. The even values {6, 12, 18, 24, 30, 36, … } are all missing from within the body of the Rank Table. We can imagine such values as being power-of-two multiples of the odd values, extending off to the left. In any case, even values must eventually become odd after a sufficient number of divide-by-two steps, so even values are implicitly covered, but not explicitly.

odd	3x+1	R1	R2	R3	R4	R5	R6	R7	R8
1									
3	10	5							
5	16	8	4	2	1				
7	22	11							
9	28	14	7						
11	34	17							
13	40	20	10	5					
15	46	23							
17	52	26	13						
19	58	29							
21	64	32	16	8	4	2	1		
23	70	35							
25	76	38	19						
27	82	41							
29	88	44	22	11					
31	94	47							
33	100	50	25						
35	106	53							
37	112	56	28	14	7				
39	118	59							
41	124	62	31						
43	130	65							
45	136	68	34	17					
47	142	71							
49	148	74	37						
51	154	77							
53	160	80	40	20	10	5			
55	166	83							
57	172	86	43						
59	178	89							
61	184	92	46	23					
63	190	95							
65	196	98	49						
67	202	101							
69	208	104	52	26	13				
71	214	107							
73	220	110	55						
75	226	113							
77	232	116	58	29					
79	238	119							
81	244	122	61						
83	250	125							
85	256	128	64	32	16	8	4	2	1
87	262	131							
89	268	134	67						
91	274	137							
93	280	140	70	35					
95	286	143							
97	292	146	73						
99	298	149							
101	304	152	76	38	19				
103	310	155							
105	316	158	79						
107	322	161							
109	328	164	82	41					
111	334	167							
113	340	170	85						
115	346	173							
117	352	176	88	44	22	11			
119	358	179							
121	364	182	91						
123	370	185							
125	376	188	94	47					
127	382	191							
129	388	194	97						
131	394	197							
133	400	200	100	50	25				
135	406	203							

Collatz Rank Table, R4 values & beyond

R4 values are { 5, 37, 69, 101, 133, … }, all of the form (32k + 5).

R4 chains terminate in { 1, 7, 13, 19, 25, }, all having the value (6k + 1).

C56: An R4 value is of the form (32k + 5) and terminates in the odd number (6k + 1), where k has the same value throughout.

(32k + 5) ↗ 3(32k + 5) + 1 =

(96k + 16) ↘ (48k + 8) ↘ (24k + 4) ↘ (12k + 2) ↘ (6k + 1)

The (6k + 1) terminating value is clearly odd after exactly 4 divide-by-two operations, so the initial (32k + 5) value was in R4.

C57: An R5 value is of the form (64k – 11) and terminates in the odd number (6k – 1), where k has the same value throughout.

(64k – 11) ↗ 3(64k – 11) + 1 =

(192k – 32) ↘ (96k – 16) ↘ (48k – 8)

↘ (24k – 4) ↘ (12k – 2) ↘ (6k – 1)

The (6k – 1) terminating value is clearly odd after exactly 5 divide-by-two operations, so the initial (64k – 11) value was in R5.

C58: An R6 value is of the form (128k + 21) and terminates in the odd number (6k + 1), where k has the same value throughout.

(128k + 21) ↗ 3(128k + 21) + 1 =

(384k + 64) ↘ (192k + 32) ↘ (96k + 16) ↘ (48k + 8)

↘ (24k + 4) ↘ (12k + 2) ↘ (6k + 1)

The (6k + 1) terminating value is clearly odd after exactly 6 divide-by-two operations, so the initial (128k + 21) value was in R6.

odd	3x+1	R1	R2	R3	R4	R5	R6	R7	R8
1									
3	10	5							
5	16	8	4	2	1				
7	22	11							
9	28	14	7						
11	34	17							
13	40	20	10	5					
15	46	23							
17	52	26	13						
19	58	29							
21	64	32	16	8	4	2	1		
23	70	35							
25	76	38	19						
27	82	41							
29	88	44	22	11					
31	94	47							
33	100	50	25						
35	106	53							
37	112	56	28	14	7				
39	118	59							
41	124	62	31						
43	130	65							
45	136	68	34	17					
47	142	71							
49	148	74	37						
51	154	77							
53	160	80	40	20	10	5			
55	166	83							
57	172	86	43						
59	178	89							
61	184	92	46	23					
63	190	95							
65	196	98	49						
67	202	101							
69	208	104	52	26	13				
71	214	107							
73	220	110	55						
75	226	113							
77	232	116	58	29					
79	238	119							
81	244	122	61						
83	250	125							
85	256	128	64	32	16	8	4	2	1
87	262	131							
89	268	134	67						
91	274	137							
93	280	140	70	35					
95	286	143							
97	292	146	73						
99	298	149							
101	304	152	76	38	19				
103	310	155							
105	316	158	79						
107	322	161							
109	328	164	82	41					
111	334	167							
113	340	170	85						
115	346	173							
117	352	176	88	44	22	11			
119	358	179							
121	364	182	91						
123	370	185							
125	376	188	94	47					
127	382	191							
129	388	194	97						
131	394	197							
133	400	200	100	50	25				
135	406	203							

C59: An even-Ranked number, RE, is of the form $(2^{E+1}k + (2^E - 1)/3)$ and terminates in the value $(6k + 1)$.

$(2^{E+1} \times k + (2^E - 1)/3) \nearrow 3(2^{E+1} \times k + (2^E - 1)/3)) + 1 = 3 \times 2^{E+1} \times k + 2^E$

$= 2^E (6k + 1) \searrow \ldots \searrow 2^2 (6k + 1) \searrow 2^1 (6k + 1) \searrow (6k + 1)$

The E-index denotes the number of divide-by-two operations necessary before the odd value $(6k + 1)$ is reached.

C60: $(2^E - 1)/3$ is an odd integer

$(2^E - 1) = (2^{2k} - 1) = (2^k - 1)(2^k + 1)$

Exactly one of the three contiguous integers $\{ (2^k - 1), 2^k, (2^k + 1) \}$ is divisible by 3. Clearly 2^k is not divisible by 3, as it has no factors of 3 in its prime factorisation. Therefore one of the two factors $(2^k - 1)$ & $(2^k + 1)$ is divisible by 3, making $(2^{2k} - 1)$ divisible by 3. This shows that $(2^E - 1)/3$ is an integer. 2^E is even. $(2^E - 1)$ is therefore odd. Dividing by 3 removed the factor of 3. It did not remove a factor of 2. If $(2^E - 1)$ were even, it would have stayed even. It is odd, and remains odd after being divided by 3.

C61: An odd-Ranked number, **RO**, is of the form $(2^{O+1}k - (2^O + 1)/3)$ and terminates in the value $(6k - 1)$.

$(2^{O+1} \times k - (2^O + 1)/3) \nearrow 3(2^{O+1} \times k - (2^O + 1)/3)) + 1 =$

$3 \times 2^{O+1} \times k - 2^O = 2^O (6k - 1) \searrow \ldots$

$\searrow 2^2 (6k - 1) \searrow 2^1 (6k - 1) \searrow (6k - 1)$

The **O**-index denotes the number of divide-by-two operations necessary before the odd value $(6k - 1)$ is reached.

C62: $(2^O + 1)/3$ is an odd integer

$3 \mid (2^{2k} - 1)$ from **C60** above

$3 \mid 2 \times (2^{2k} - 1) = (2^{2k+1} - 2)$ as $\times 2$ does not affect divisibility by 3

$3 \mid (2^{2k+1} + 1)$ adding 3 does not affect divisibility by 3

Doubling $(2^{2k} - 1)/3$ made it even. Adding 3/3 makes it odd.

C63: Only values in R1 are increased at the end of a chain. All other R*N* values are strictly decreased.

See the results in this table. Note that the value of k at the end is the same as that at the start.

	start		end
R1	$4k - 1$	$2^2k - 1$	$6k - 1$
R2	$8k + 1$	$2^3k + 1$	$6k + 1$
R3	$16k - 3$	$2^4k - 3$	$6k - 1$
R4	$32k + 5$	$2^5k + 5$	$6k + 1$
R5	$64k - 11$	$2^6k - 11$	$6k - 1$
R6	$128k + 21$	$2^7k + 21$	$6k + 1$
RE	------	$2^{E+1}k + (2^E - 1)/3$	$6k + 1$
RO	------	$2^{O+1}k - (2^O + 1)/3$	$6k - 1$

C64: Since one half of all odd values are in R1, half of all odd values give rise to increasing end-of-chain values.

The odd naturals form an infinite set. Given that the modern mathematics of infinity [4] has not yet penetrated the consciousness of the majority of readers, we can talk instead about the relative densities of values.

In this asymptotic sense, one half of all natural numbers are odd $(2k - 1)$, with half of these being in R1 and therefore of the form $(4k - 1)$. The rest { R2, R3, R4, R5, R6, R7, R8, R9, … } sum to

$$\tfrac{1}{4} = \{1/8 + 1/16 + 1/32 + 1/64 + 1/128 + 1/256 + \ldots \},$$

meaning they form the other half which are not in R1.

[4] *The TRUE Mathematics of Infinity for Scientists and Engineers*, 2019, Green, L.O. (v1.20, 2021)

C65: All R1 chains can be converted into longer equivalent [5] R3 chains by applying a (4× + 1) operation to the starting value. Two extra even values are added to the start of the even-part of the chain.

Example: **R1**: 3 – { 10, 5 } goes to **R3:** 13 – { 40, 20, 10, 5 }

Start from a specific R1 value, (4k – 1). Apply the (4× +1) operation to get 4(4k – 1) + 1 = 16k – 3. This is indeed an R3 value from an R1 value, but does the chain end in the same way?

(4k – 1) ↗ 3(4k – 1) + 1 = (12k – 2) *start of the even values*

(16k – 3) ↗ 3(16k – 3) + 1 = 48k – 8 = 4(12k – 2)

Note that the value k is consistent between the two lines above, and is not just a dummy variable.

C66: All R2 chains can be converted into longer equivalent R4 chains by applying a (4× + 1) operation to the starting value. Two extra even values are added to the start of the even-part of the chain.

Example: **R2**: 9 – { 28,14,7 } goes to **R4:** 37 – { 112,56,28,14,7 }

Start from a specific R2 value, (8k + 1). Apply the (4× +1) operation to get 4(8k + 1) + 1 = 32k + 5 This is indeed an R4 value from an R2 value, but does the chain end in the same way?

(8k + 1) ↗ 3(8k + 1) + 1 = (24k + 4) *start of the even values*

(32k + 5) ↗ 3(32k + 5) + 1 = 96k +16 = 4(24k + 4)

As before it is the same value of k in both lines above.

[5] Whilst we do not wish to endorse the general concepts presented in the following paper, we do wish to acknowledge the (4× + 1) inspiration from section 2 of Maya Ahmed's paper "*A window to the Convergence of a Collatz Sequence*", (2015).

C67: All R3 chains can be converted into longer equivalent R5 chains by applying a (4× + 1) operation to the starting value. Two extra even values are added to the start of the even-part of the chain.

Example: **R3:** 13 – {40, 20, 10, 5} goes to
R5: 53 – {160, 80, 40, 20, 10, 5}

Start from a specific R3 value, (16k – 3). Apply the (4× +1) operation to get 4(16k – 3) + 1 = (64k – 11). This is indeed an R5 value from an R3 value, but does the chain end in the same way?

(16k – 3) ↗ 3(16k – 3) + 1 = (48k – 8) *start of the even values*

(64k – 11) ↗ 3(64k – 11) + 1 = (192k – 32) = 4(48k – 8)

As before it is the same value of k in both lines above.

What we have said is that up to some huge value, 2H, we have H odd numbers, H/2 are in R1, H/4 are in R2, H/8 are in R3, and so on. By converting R1 values to R3 values, R2 values to R4 values, and R3 values to R5 values, and so on, we seem to have reduced the amount of starting values we need to consider. This is a fallacy. All we have done is shove the R1, R2, and R3 values further up the number line (beyond 2H) where they are not so visible. It is a typical 'infinity paradox' waiting to happen, so we must ensure that we don't fall into such an obvious and embarrassing trap.

All we are actually trying to do is to consolidate the Ranks to make them more similar to the Structure Table Levels.

C68: All RE chains can be converted into longer equivalent RE′ chains by applying a (4× + 1) operation to the starting value. Two extra even values are added to the start of the even-part of the chain. E′ = E+2.

Start from a generic RE value, $(2^{E+1}k + (2^{E} - 1)/3)$

Apply the (4× +1) operation to get $4(2^{E+1}k + (2^{E} - 1)/3) + 1 =$

$(2^{E+3}k + (2^{E+2} - 4)/3) +1 = (2^{E'+1}k + (2^{E'} - 1)/3) - 3/3 + 1 =$

$(2^{E'+1}k + (2^{E'} - 1)/3)$

This is indeed an RE′ value from an RE value.

$(2^{E+1}k + (2^{E} - 1)/3) \nearrow 3(2^{E+1}k + (2^{E} - 1)/3) + 1 = (3\times2^{E+1}k + 2^{E}) - 1 + 1$

$(2^{E'+1}k + (2^{E'} - 1)/3) \nearrow 3\times2^{E'+1}k + 2^{E'} = 4(3\times2^{E+1}k + 2^{E})$

The values of E and k are the same for both lines above.

C69: All **RO** chains can be converted into longer equivalent **RO′** chains by applying a (4× + 1) operation to the starting value. Two extra even values are added to the start of the even-part of the chain. **O′** = **O**+2.

Start from a generic **RO** value, $(\mathbf{2^{O+1}}k - (\mathbf{2^{O}} + 1)/3)$.

Apply the (4× +1) operation to get $4(\mathbf{2^{O+1}}k - (\mathbf{2^{O}} + 1)/3) + 1 =$

$(\mathbf{2^{O+3}} - (\mathbf{2^{O+2}} + 4)/3) + 1 = (\mathbf{2^{O'+1}}k - (\mathbf{2^{O'}} + 1)/3) - 3/3 + 1 =$

$(\mathbf{2^{O'+1}}k - (\mathbf{2^{O'}} + 1)/3)$

This is indeed an **RO′** value from an **RO** value.

$(\mathbf{2^{O+1}}k - (\mathbf{2^{O}} + 1)/3) \nearrow 3(\mathbf{2^{O+1}}k - (\mathbf{2^{O}} + 1)/3) + 1 = (3\times\mathbf{2^{O+1}}k - \mathbf{2^{O}}) - 1 + 1$

$(\mathbf{2^{O'+1}}k - (\mathbf{2^{O'}} + 1)/3) \nearrow (3\times\mathbf{2^{O'+1}}k - \mathbf{2^{O'}}) = 4(3\times\mathbf{2^{O+1}}k - \mathbf{2^{O}})$

The values of **O** and k are the same for both lines above.

Relating the Rank Table to the Structure Table

The Rank Table is guaranteed to give 100% coverage over the natural numbers (**C55**). Every R*N* is guaranteed to terminate in a different value to its starting value. If we can relate the 100% coverage to the Structure Table then our work is complete.

If we consider the L0 set in the Structure Table, we see those values in many different chains within the Rank Table. For example { 5, 21, 85, … } go increasingly far up the L0 chain. There are infinitely many starting points within the Rank Table, each of successively increasing ranks (two ranks at a time), which end up in L0. The (odd) starting points are of course within L1. More accurately, they compose L1, with the values $(2^{2k} - 1)/3$.

To be clear about what we actually need, there is no point in finding Levels in the Structure Table and relating them back to the Rank Table. What we want to do is to find any value in the Rank Table and then ensure that this value is available in the Structure Table. If we can do this, then the 100% coverage of the Rank Table passes over to 100% coverage by the Structure Table.

As an example, can we find a home for all R1 values within the Structure Table? They are of the form (4k – 1), but indeterminately far from the L0 termination sequence.

R1 = { 3, 7, 11, … }. 3 is easy to find in L3, but 7 is not even seen in the Structure Table up to L7. In fact 7 is in L9, whilst 11 is in L7.

The problem is that there are infinitely many starting values within the Rank Table, so this does not reduce the difficulty of proving that we have 100% coverage of the naturals within the Structure Table.

Rank Values within the Structure Table

We have seen that L1 starts at 5, and subsequent elements in L1 are related by the (4× + 1) recurrence relation. There is only one such *chain* in L1. In L3 we have an indefinite number of such chains, each ending in a distinct L1 element, although L1 elements which are divisible by 3 do not support such chains (shaded in blue). Of course the blue elements in L1 do support infinite chains in L2.

Level 3					Level 2	Level 1
853	213	53	13	3	↗↘	5
						21
29013	7253	1813	453	113	↗↘↘	85
58197	14549	3637	909	227	↗↘	341
						1365
1864021	466005	116501	29125	7281	↗↘↘	5461
3728213	932053	233013	58253	14563	↗↘	21845

The yellow shaded *root* elements in L3 are R1 values. The unshaded root elements in L3 are R2 values.

We have already seen (**C65**) that an R1 value is converted to a *related* R3 value by using the (4× + 1) operation. Likewise an R2 value is converted to a related R4 value (**C66**) using the same operation. It means we can additionally label the Structure Table values with Ranks.

Level 3					Level 2	Level 1
853	213	53	13	3	↗↘	5
R9	**R7**	**R5**	**R3**	**R1**		

Level 3					Level 2	Level 1
29013	7253	1813	453	113	↗↘↘	85
R10	**R8**	**R6**	**R4**	**R2**		

We now know how to place all Rank values within the Structure Table pattern. This is sadly not the same as saying that all Rank values are actually in the Structure Table.

C70: The root elements in odd Levels are either R1 or R2 values.

Suppose the root element had a Rank above 2. Each odd Level chain increases from the root value using the (4× + 1) relation. This means that a lower value could be achieved by subtracting 1, then dividing by 4. In such a case the identified value could not in fact be a root, the root being the lowest value in the chain. In the same way that the (4× + 1) operation gives a resultant two Ranks above the starting value, the reverse operation lowers the Rank by two. Neither R1 nor R2 can be reduced by two Ranks. Therefore the roots of odd Levels are either R1 or R2 values.

The yellow shaded L5 root elements are R1 values. The unshaded root elements in L5 are R2 values.

The blue shaded elements in L3 are divisible by 3, and therefore do not support chains in L5. There is of course an infinite chain from these values in L4.

Level 5					Level 4	Level 3
4437	1109	277	69	17	↗↘↘	13
9045	2261	565	141	35	↗↘	53
						213
291157	72789	18197	4549	1137	↗↘↘	853
19285	4821	1205	301	75	↗↘	113
						453
618837	154709	38677	9669	2417	↗↘↘	1813
1237845	309461	77365	19341	4835	↗↘	7253
38741	9685	2421	605	151	↗↘	227
						909
1241429	310357	77589	19397	4849	↗↘↘	3637
2483029	620757	155189	38797	9699	↗↘	14549

Stray Values

We define a ***stray value*** as an example of a natural number which cannot be found within the Structure Table. This contradicts claim **C20**, which remains unproven.

If there is even one stray value, then we can label ***p*** as the minimum such value. We suppose that such a value *could* exist and consider the implications.

Clearly ***p*** cannot be even as ***p***/2 < ***p***, contradicting the original definition. Whilst we have said that ***p*** is not within the Structure Table, the rules that form the Structure Table nevertheless have to be upheld as they come directly from the Collatz rules. Specifically, there has to be an infinite chain of ×2 even values numerically above ***p***. Additionally, unless ***p*** is divisible by 3, there has to be a chain of (4× + 1) odd values above ***p*** as well.

From **C70**, ***p*** has to be of Rank 1 (R1) or Rank 2 (R2) as it is the lowest element of the odd chain (by definition). However, we can further limit ***p*** to being an R1 value because an R2 value would make a new stray value ***q*** with ***q*** < ***p***, which is not allowed by our definition of ***p*** as the minimum stray value.

...		...
1024***p***		1024(3***p*** + 1)
512***p***		512(3***p*** + 1)
256***p***		256(3***p*** + 1)
128***p***		128(3***p*** + 1)
64***p***		64(3***p*** + 1)
32***p***		32(3***p*** + 1)
16***p***		16(3***p*** + 1)
8***p***		8(3***p*** + 1)
4***p***		4(3***p*** + 1)
2***p***		2(3***p*** + 1)
p = (4k-1)	↗	(3***p*** + 1)
		q = (3***p*** + 1)/2 = 6k-1

In summary, if ***p*** exists, it has to be an R1 value and therefore of the form (4k − 1) according to **C51**. We can also explicitly write down the value of ***q*** as (6k − 1). Needless to say, there is also a power-of-twos multiple chain above ***q*** as well.

⇐ Here is what we know so far.

We have assumed for simplicity that ***p*** is divisible by 3 so the (4× + 1) odd chain is not required.

The ellipses at the top of the power-of-two multiples chains mean carry on in the same way indefinitely (forever).

q can be in R1 or R2, but not in R3+ as that would make the resultant ***r*** < ***p***, which is not allowed by our prior definition of ***p***.

...		...		
1024***p***		1024(3***p*** + 1)		...
512***p***		512(3***p*** + 1)		1024(3***q*** + 1)
256***p***		256(3***p*** + 1)		512(3***q*** + 1)
128***p***		128(3***p*** + 1)		256(3***q*** + 1)
64***p***		64(3***p*** + 1)		128(3***q*** + 1)
32***p***		32(3***p*** + 1)		64(3***q*** + 1)
16***p***		16(3***p*** + 1)		32(3***q*** + 1)
8***p***		8(3***p*** + 1)		16(3***q*** + 1)
4***p***		4(3***p*** + 1)		8(3***q*** + 1)
2***p***		2(3***p*** + 1)		4(3***q*** + 1)
p = (4k-1)	↗	(3***p*** + 1)		2(3***q*** + 1)
		q = (3***p*** + 1)/2 = 6k - 1	↗	(3***q*** + 1) = 2(9k - 1)
				9k - 1
				4.5k - 0.5
				r ≠ 2.25k - 0.25

There are two possibilities for ***r***: ***r*** = 9k – 1 and ***r*** = 4.5k – 0.5.

If ***p*** and ***r*** were equal we would have a simple loop. In fact a longer loop, if it were possible, would have to be pretty large, as we shall see later.

Of course we have already demonstrated that the Structure Table has no loops (**C24**). That demonstration does not relate to our application here because we have postulated the existence of a stray value which does not exist within the Structure Table.

Notice that the stray table we built up is not actually the Structure Table since it is not *rooted* at 1. The key point of the Structure Table is that as you proceed through its Levels, you make definite and steady progress towards the goal of termination.

The stray table we have partially built has exactly three possibilities:

1) it terminates eventually
2) it loops indefinitely
3) it diverges to infinity

C72: If any part of the stray table meets any part of the Structure Table, the stray table ceases to exist.

> The stray table follows all the rules of a path through the Structure Table, with the exception that stray table has no route to termination. If any part of the stray table meets any part of the Structure Table, a route to termination has been found. Each branch of the stray table can then be labelled with a Level, and that branch will already exist within a Structure table of adequate size. The stray table is no longer stray, and ceases to exist as defined.

From **C72** above, a stray table cannot have any branch which terminates (in the sense of reaching 1), hence it must either loop or diverge.

So far we have a stray table with 4 odd root values in it: ***p***, ***q***, ***r***, and a so-far unnamed one, say ***s***. Each has an infinite power-of-two set of multiples *above it*. At least three of these infinite chains also have an infinite set of odd values coming off from every other even value.

The infinite power-of-two multiples of ***p*** chain may or may not have an infinite chain of odd values coming off from it, depending on its divisibility by 3. As seen in **C09**, a chain of even values which is also divisible by 3 cannot be reached by the Collatz (3× + 1) operation.

Below we show the current state of the stray table.

Even-value chains are shaded light blue. Odd-value chains are shaded light green.

We have not added an odd chain to the left of the even column rooted on ***p*** on the (pessimistic) assumption that ***p*** is evenly divisible by 3.

We have already demonstrated that ***p***, ***q***, ***r***, and ***s*** are distinct values. Other than possibly ***p***, they are also not multiples of 3, since they are all of the form (6k ± 1). As a result, all values shown in this (infinite) table are also distinct, again using the Fundamental Theorem of Arithmetic argument.

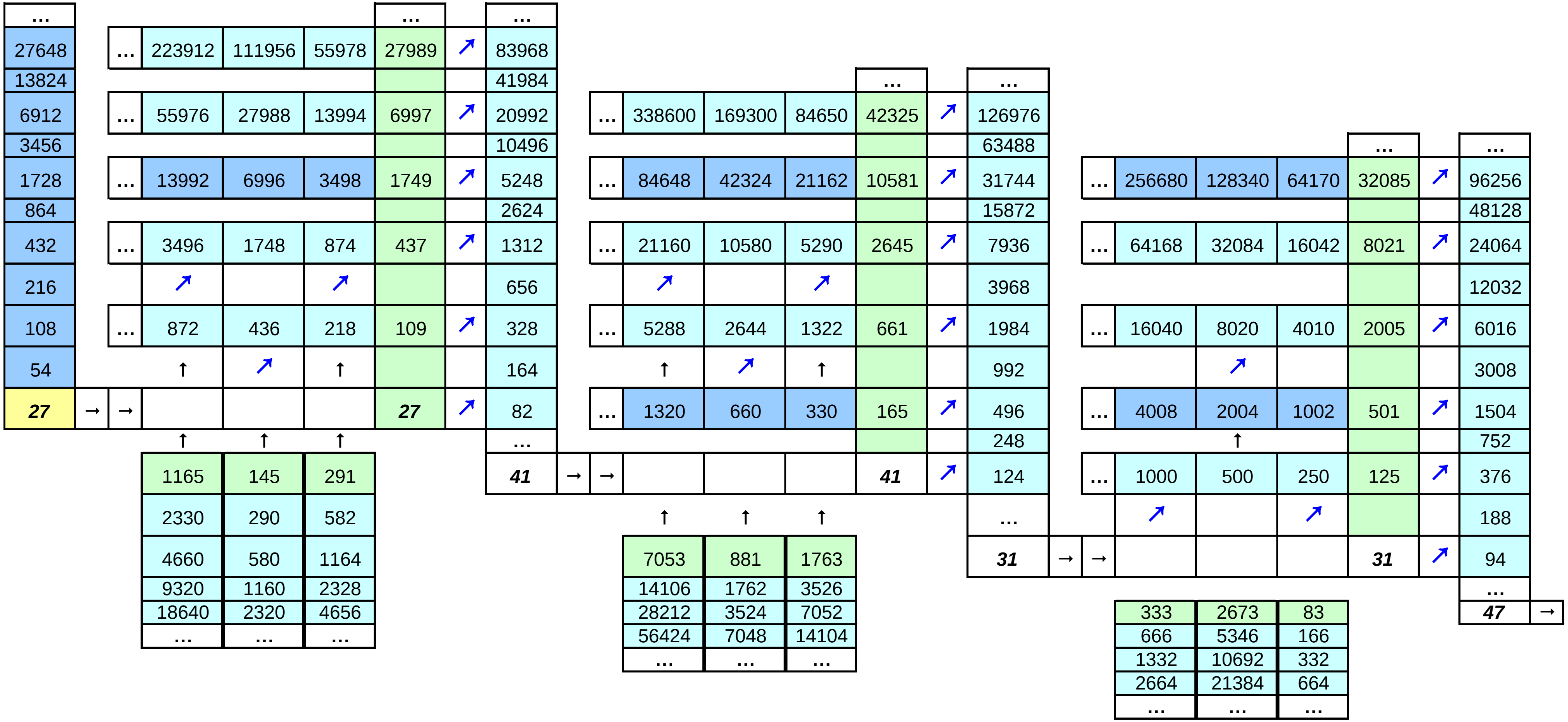
There is an infinite column of power-of-two multiples of ***q*** going upwards. From this column there is another infinite odd column to the left where the (3× + 1) steps feed into the even column. From each odd value we get an infinite row of power-of-two multiples of the odd values.

On the next page we show another set of (3× + 1) steps to every other even value, but the table is getting difficult to read. Some actual example numbers, and arrows (overleaf), make the drawing less difficult to understand, but it nevertheless takes some work.

...						...		...																	
1024p		...	8(1024p + 341)	4(1024p + 341)	2(1024p + 341)	1024p + 341	↗	1024(3p + 1)																	
512p								512(3p + 1)						...		...									
256p		...	8(256p + 85)	4(256p + 85)	2(256p + 85)	256p + 85	↗	256(3p + 1)		...	8(1024q + 341)	4(1024q + 341)	2(1024q + 341)	1024q + 341	↗	1024(3q + 1)									
128p								128(3p + 1)								512(3q + 1)						...		...	
64p		...	8(64p + 21)	4(64p + 21)	2(64p + 21)	64p + 21	↗	64(3p + 1)		...	8(256q + 85)	4(256q + 85)	2(256q + 85)	256q + 85	↗	256(3q + 1)		...	8(1024r + 341)	4(1024r + 341)	2(1024r + 341)	1024r + 341	↗	1024(3r + 1)	
32p								32(3p + 1)								128(3q + 1)								512(3r + 1)	
16p		...	8(16p + 5)	4(16p + 5)	2(16p + 5)	16p + 5	↗	16(3p + 1)		...	8(64q + 21)	4(64q + 21)	2(64q + 21)	64q + 21	↗	64(3q + 1)		...	8(256r + 85)	4(256r + 85)	2(256r + 85)	256r + 85	↗	256(3r + 1)	
8p								8(3p + 1)								32(3q + 1)								128(3r + 1)	
4p		...	8(4p + 1)	4(4p + 1)	2(4p + 1)	4p + 1	↗	4(3p + 1)		...	8(16q + 5)	4(16q + 5)	2(16q + 5)	16q + 5	↗	16(3q + 1)		...	8(64r + 21)	4(64r + 21)	2(64r + 21)	64r + 21	↗	64(3r + 1)	
2p								2(3p + 1)								8(3q + 1)								32(3r + 1)	
p	→	→	→	→	→	p	↗	(3p + 1)		...	8(4q + 1)	4(4q + 1)	2(4q + 1)	4q + 1	↗	4(3q + 1)		...	8(16r + 5)	4(16r + 5)	2(16r + 5)	16r + 5	↗	16(3r + 1)	
								...								2(3q + 1)								8(3r + 1)	
								q	→	→	→	→	→	q	↗	(3q + 1)		...	8(4r + 1)	4(4r + 1)	2(4r + 1)	4r + 1	↗	4(3r + 1)	
																...								2(3r + 1)	
																r	→	→	→	→	→	r	↗	(3r + 1)	
																								...	
																								s	→

Showing a numerical example is helpful, even though it cannot be real in the sense of finding a genuine stray starting point (*which may not actually exist*). 27 is a multiple of 3, like our postulated value of ***p***, so there are no feed-in points to the left of the power-of-two multiples of ***p*** (=27) column.

Notice that we do have some darker blue rows (and the darker blue column above ***p***), which are the ‘unreachable’ divisible-by-3 sets, also seen in the Structure Table. These were mentioned earlier with respect to L2 (**C08**, **C09**).



C73: Every third power-of-two multiple chain rooted in odd values which feed in (via a $(3\times + 1)$ step) to a power-of-two multiple of an odd value are evenly divisible by 3. (In other words the darker blue rows in the table above)

We know that all end values from a Rank chain are of the form $(6k \pm 1)$. Suppose $\boldsymbol{q} = (6k - 1)$. Using the direct $(4\times + 1)$ method we then get

$$4q + 1 = 4(6k - 1) + 1 = 24k - 3 = 3k' \quad \text{(divisible by 3)}$$

Suppose $\boldsymbol{q} = (6k + 1)$. Using the direct $(4\times + 1)$ method we then get

$$4q + 1 = 4(6k + 1) + 1 = 24k + 5 = 3k' + 2$$

$$16q + 5 = 16(6k + 1) + 5 = 96k + 21 = 3k' \quad \text{(divisible by 3)}$$

k' is a *dummy variable* in all the lines above (inconsistent values).

This has showed the starting values, but not yet the 1-in-3 rule.

The even number chain above $\boldsymbol{q}$ is $2^n \times \boldsymbol{q}$. The every-other-one divisible by 3 rule gives the $2^{2n} \times \boldsymbol{q}$ rule we have seen before. The new 1-in-3 of these divisible by 9 rule is effectively for $2^{6n} \times \boldsymbol{q}$.

$\mathbf{9 \mid (2^{6n} - 1)}$ is easy to demonstrate numerically, and is proved in **C74** below.

C74: $9 \mid (2^{6n} - 1)$ or $(2^{6n} - 1) = 9k$

$$\mathbf{(2^6 - 1)} = (2^3 + 1)(2^3 - 1) = \mathbf{9}(2^3 - 1)$$
$$\mathbf{(2^{12} - 1)} = (2^6 + 1)(2^6 - 1) = \mathbf{9}(2^3 - 1)(2^6 + 1)$$

$$\mathbf{(2^{24} - 1)} = (2^{12} + 1)(2^{12} - 1) = \mathbf{9}(2^3 - 1)(2^6 + 1)(2^{12} + 1)$$
$$\mathbf{(2^{48} - 1)} = (2^{24} + 1)(2^{24} - 1) = \mathbf{9}(2^3 - 1)(2^6 + 1)(2^{12} + 1)(2^{24} + 1)$$

$$(2^6 - 1)(2^{12} - 1) = 2^{18} - 2^{12} - 2^6 + 1 = (2^{18} - 1) - (2^{12} - 1) - (2^6 - 1)$$
$$\mathbf{(2^{18} - 1)} = (2^6 - 1)(2^{12} - 1) + (2^{12} - 1) + (2^6 - 1), \qquad \textit{all divisible by 9}$$

$$(2^6 - 1)(2^{6n} - 1) = 2^{6n+6} - 2^{6n} - 2^6 + 1 = (2^{6n+6} - 1) - (2^{6n} - 1) - (2^6 - 1)$$
$$\mathbf{(2^{6(n+1)} - 1)} = (2^6 - 1)(2^{6n} - 1) + (2^{6n} - 1) + (2^6 - 1)$$

When n=1, all the RHS terms are $(2^6 - 1)$, and we have shown above that these are all divisible by 9. Since all the terms on the RHS are divisible by 9, their sum, and hence the LHS, is also divisible by 9. We have shown that ***if*** $9 \mid (2^{6n} - 1)$, then $9 \mid (2^{6(n+1)} - 1)$, and also shown that the n=1 case is true. Using mathematical induction we have therefore proved that the general case is true.

We now proceed to count the numbers produced in the stray table based on the single value $\boldsymbol{q}$ and up to some huge limit, H, assuming that $\boldsymbol{q}$ is relatively small compared to H.

(1) Even numbers above $\boldsymbol{q}$, amounting to $\log_2(H)$
(2) Odd numbers to every-other even number in (1) above = ½ $\log_2(H)$
(3) Even numbers from each odd number in (2) above = ½ $\log_2(H) \times \log_2(H)$
(4) Odd numbers from ⅓ of the even numbers from (3) ⅙ $\log_2(H) \times \log_2(H)$

We lose 1 in 3 chains, and then every other term from each remaining chain, making only ⅓ of the even values accessible from an odd value.

(5) Even numbers from each odd number in (4) above =

⅙ $\log_2(H) \times \log_2(H) \times \log_2(H)$

(6) Odd numbers from 1/3 of the even numbers from (5) above =
…

We have the sum:

$$S = (3/2)\log_2(H) + (2/3)[\log_2(H)]^2 + (2/3^2)[\log_2(H)]^3 + \ldots$$

Since $\log_2(H) \gg 10$ we have a *strictly* divergent series.[6]

It is important to note that all of the values we are counting here are strictly greater than $\boldsymbol{q}$. It means that as we iterate through the counting sequence above, the values get increased steadily by some factor, say k.

At some point the previous implicit assumption that $H \gg k^n \boldsymbol{q}$ will break down, making the rate of divergence at least slow down, and possibly making the series no longer divergent.

(1) becomes $\log_2(H) - \log_2(\boldsymbol{q}) = \log_2(H/\boldsymbol{q})$
(2) becomes ½ $\log_2(H/\boldsymbol{q})$
(3) becomes ½ $\log_2(H/\boldsymbol{q}) \times \log_2(H/(k\boldsymbol{q}))$
(4) becomes ⅙ $\log_2(H/\boldsymbol{q}) \times \log_2(H/(k\boldsymbol{q}))$
(5) becomes ⅙ $\log_2(H/\boldsymbol{q}) \times \log_2(H/(k\boldsymbol{q})) \times \log_2(H/(k^2\boldsymbol{q}))$

An approximated value of 8 for k would be both relevant and convenient. This is nevertheless a *difficult* sum.

Of course S was just for $\boldsymbol{q}$. We demonstrated that there had to be at least 2 similar root values, and actually as many more as were needed to form a loop (if such a loop could exist).

As you can easily see, the hope was to show that a stray value, and hence a stray infinite chain, and hence an infinite collection of infinitely long chains, demonstrated that such a set must necessarily intersect with the Structure Table, and therefore could not be stray. But we have not achieved this goal. Infinity does not mean "all".

Consider the natural numbers. There are infinitely many of them. Now consider the natural numbers which are evenly divisible by 7. There are infinitely many. Nobody would be so idiotic as to assert that because there are infinitely many natural numbers, and infinitely many natural numbers evenly divisible by 7, that all natural numbers are evenly divisible by 7. The infinities are distinct.

One could argue this case using either the Φ-rules or density, neither of which are especially well known.

[6] *On the Summation of Divergent series using the Φ rules*, 2018, Green, L.O. (v1.80, 2021).

Here we could (unintentionally) make the logical fallacy less easy to spot.

5×2^n generates an infinite chain (set) of even numbers.

7×2^m also generates an infinite chain (set) of even numbers. But these infinite sets are disjoint and do not cover the field of even numbers.

It would be very reasonable to argue that the Structure Table is an 'after-the-fact' construct. Only values which are known to be convergent appear on such a table.

It is also very reasonable to consider that the Structure Table is a Reverse Collatz table, in the sense that it is built upwards from the root value 1. Of course in the actual construction we search all possible "upwards" paths (further from termination, but not always greater in numerical value).

At any branch point, we always go in *both* directions. In the normal iteration sequence these branch points are merge points. They only occur in even Levels at values of the form (3k + 1).

More on Loops

Since numbers up to 10^{20} are known to be convergent under Collatz, we can simplify an UP step, $(3\times + 1)$, to $3\times$ without too much error. In this case loops occur when the product of UP steps $(3\times)$ and DOWN steps, (/2), is close to one.

Clearly $3^n \neq 2^m$ for n ≥ 1, due to the Fundamental Theorem of Arithmetic. However, the addition of 1 into the Collatz UP step means the inequality cannot be definitely excluded. If we try to look for cases where

$\frac{3^n}{2^m} \cong 1$ we soon get a problem because even using **doubles** we get an overflow when n > 646. Instead we take natural logs and expect a result close to zero, then round ***m*** to the nearest integer, and see how close we got. In C++ we have:

```
const double f = log(3.0)/log(2.0);

for( int n=1; n<100; n++ ){
    int m = int(n * f + 0.5);

    double delta = fabs( n * log(3) - m * log(2) );
    double ratio = exp( delta );

    if( ratio < 1.05 )
        printf("\n\tn=%d, m=%d\tratio=%4.3f", n, int(m), ratio );
}
```

The **log( )** function gives the natural log.

Notice that without the 1 in the UP steps, the product of steps can be calculated in any order. When the 1 is included, the exact order of UP and DOWN steps becomes relevant if enough significant digits are considered.

First we allow quite a large ratio in order to check the maths 'manually' using a calculator.

```
n=12, m=19      ratio=1.014
n=17, m=27      ratio=1.039
n=24, m=38      ratio=1.027
n=29, m=46      ratio=1.025
n=36, m=57      ratio=1.041
n=41, m=65      ratio=1.012
n=53, m=84      ratio=1.002
n=65, m=103     ratio=1.016
n=70, m=111     ratio=1.037
n=77, m=122     ratio=1.030
n=82, m=130     ratio=1.023
n=89, m=141     ratio=1.044
n=94, m=149     ratio=1.009
```

Then we can reduce the ratio to get a better approximation.

```
n=79335, m=125743       ratio=1.00000366
n=111202, m=176251      ratio=1.00000360
n=190537, m=301994      ratio=1.00000006
n=269872, m=427737      ratio=1.00000373
n=301739, m=478245      ratio=1.00000354
n=381074, m=603988      ratio=1.00000013
n=460409, m=729731      ratio=1.00000379
n=492276, m=780239      ratio=1.00000347
n=571611, m=905982      ratio=1.00000019
n=650946, m=1031725     ratio=1.00000386
n=682813, m=1082233     ratio=1.00000341
n=762148, m=1207976     ratio=1.00000026
n=841483, m=1333719     ratio=1.00000392
n=873350, m=1384227     ratio=1.00000334
n=952685, m=1509970     ratio=1.00000032
```

Since the variable **delta** is obtained by subtracting two very nearly equal values, after having multiplied the logs by some large numbers, we must be losing accuracy from the supposed 15 digit accuracy of the original log function.

The point is we have to do some significant error analysis to work out the uncertainty in the ratio values, but it certainly looks like loops, if any, have to be longer than hundreds of thousands of steps.

If we tweak the calculation for improved accuracy

```
double delta2= fabs( n * log(3.0/2.0) - ( m - n ) * log(2) );
```

it makes no visible difference at the displayed resolution shown above.

The point is we could use a full **long double** library, with long double logs, and we could then reduce the ratios closer and closer to one, but all we get is a higher bound for any possible loop size. That is not very interesting from an analytic point of view since it does not tell us whether or not loops can occur.

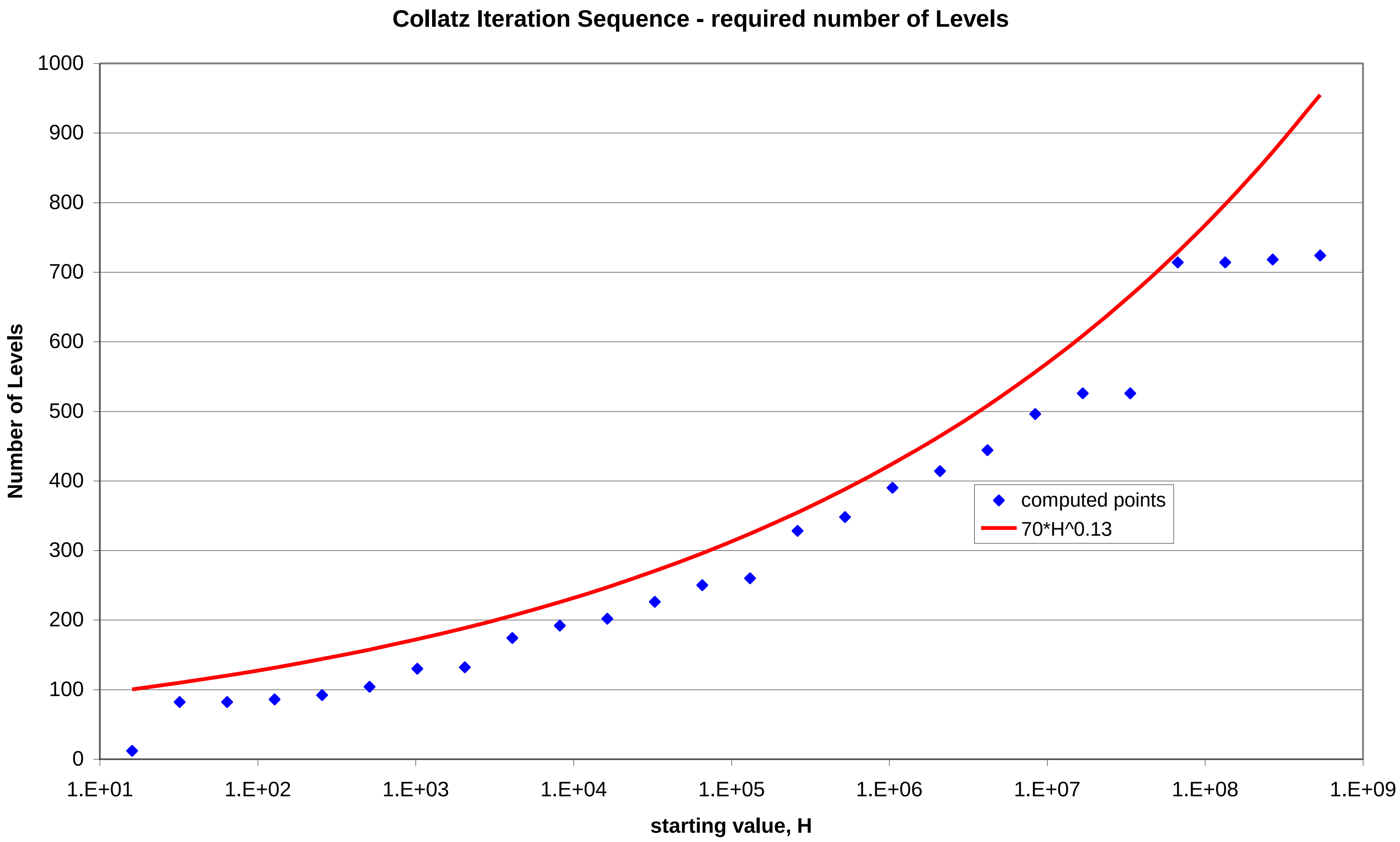
Conclusion

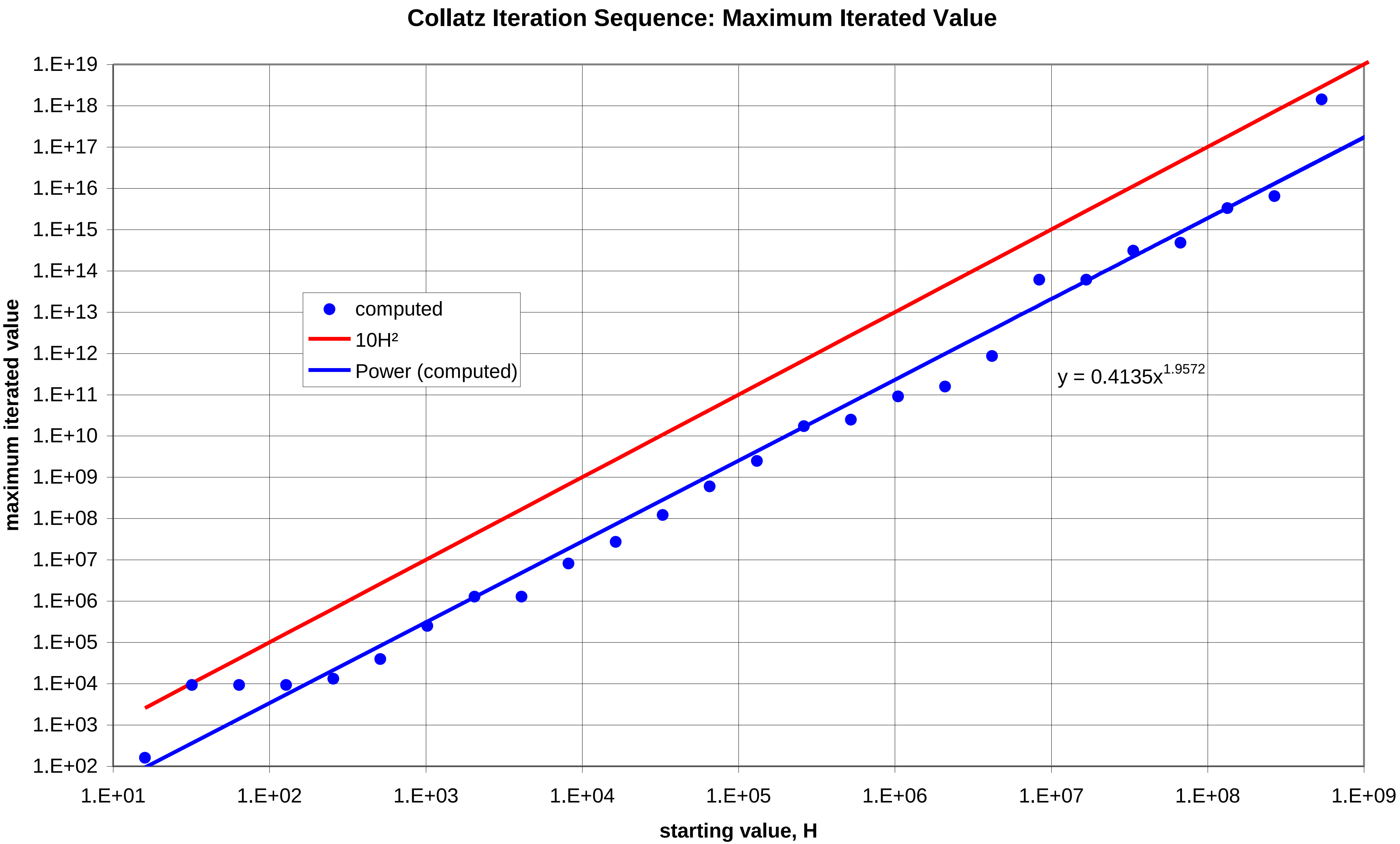
The numerical testing for loops shows that any loop must be at least hundreds of thousands of steps long, but no amount of numerical testing is likely to settle the question concerning loops (unless an actual loop is found).

Using **C72** we can certainly say that there are *either* infinitely many counter-examples to the Collatz Conjecture, or absolutely none at all. No in-between cases are possible.

Appendix: Computational Results

We can never hope to prove anything with computation alone. However, we can gain insight into the problem.




Collatz Iteration Sequence: Maximum Iterated Value
maximum iterated value
1.E+19
1.E+18
1.E+17
1.E+16
1.E+15
1.E+14
1.E+13
1.E+12
1.E+11
1.E+10
1.E+09
1.E+08
1.E+07
1.E+06
1.E+05
1.E+04
1.E+03
1.E+02
computed
10H²
Power (computed)
y = 0.4135x^1.9572
1.E+01
1.E+02
1.E+03
1.E+04
1.E+05
1.E+06
1.E+07
1.E+08
1.E+09
starting value, H

Version History

v1.72: 13 August: Update graphs in Numbers in Levels to include 2 extra decades of starting values.

v1.70: 4 Aug 2022: New section, ***Numbers in Levels***.

v1.60: 5 Nov 2021. Complete the proof of C74.

v1.50: 23 Oct 2021. New section, ***Rank Values within the Structure Table***.

Page 5, **Definitions**. Add a definition for a chain *within* an odd level.

Page 9, only **even** Levels are power-of-two chains. Add a sentence about odd Levels.

Rework the section on **Stray Values** with a *minimum* stray value, ***p***. Remove older work on loops and add a new section. ***More on Loops***, considering only values above 10^{15}.

v1.00: 8 Feb 2021. First publication on

http://lesliegreen.byethost3.com