Collatz conjecture structural and logical proof .

Author: Samiya Akhtar

Note for reader.

I know it's hard to accept new perspectives. When we come across something new, especially if it goes against what we've always believed in, it can feel strange. Sometimes our first instinct is to reject it, or to put it aside until we've had time to think it over. If that happens while you're reading this, that's okay. Just give yourself space, stay curious, and let logic take the lead instead of emotions. You can always take a break and come back when you feel ready.

Here are some points to note before you move ahead

- The Collatz conjecture isn't really a usual number theory problem.It's a whole system that processes any number using its own fixed rules. (17 state based transitional rules stated in the document)
- It does have "**invariants**," but they're structural invariants, not numerical.
- Once you look at it through the right lens, it becomes predictable and decidable in fact quite innocent.
- It works like an algorithm, like a computer program, coherently.

So, let's explore the system step by step , class by class and see how it really works under the hood.

1. Structure of 3n + 1 (First invariant)

Self-similar recursive structure of function (function's heart)

Position	Odd numbers	3n + 1	Power of 2 × Odd	Resulting mod class	Scaled copy of function
1	1	4	$2^2 \times 1$	1mod8	
2	3	10	$2^1 \times 5$	5mod8	
3	5	16	$2^2 \times 1$	Move back to 1st position	4(3n+1)
4	7	22	$2^1 \times 11$	3mod8	
5	9	28	$2^2 \times 7$	7mod8	
6	11	34	$2^1 \times 17$	1mod8	
7	13	40	$2^3 \times 5$	Move back to 2nd position	4(3n+1)
8	15	46	$2^1 \times 23$	7mod8	
9	17	52	$2^2 \times 13$	5mod8	
10	19	58	$2^1 \times 29$	5mod8	
11	21	64	$2^6 \times 1$	Move back to 3rd position	4(3n+1)
12	23	70	$2^1 \times 35$	3mod8	
13	25	76	$2^2 \times 19$	3mod8	
14	27	82	$2^1 \times 41$	1mod8	
15	29	88	$2^3 \times 11$	Move back to 4th position	4(3n+1)
16	31	94	$2^1 \times 47$	7mod8	
17	33	100	$2^2 \times 25$	1mod8	
18	35	106	$2^1 \times 53$	5mod8	
19	37	112	$2^4 \times 7$	Move back to 5th position	4(3n+1)
20	39	118	$2^1 \times 59$	3mod8	
21	41	124	$2^2 \times 31$	7mod8	
22	43	130	$2^1 \times 65$	1mod8	
23	45	136	$2^3 \times 17$	Move back to 6th position	4(3n+1)
24	47	142	$2^1 \times 71$	7mod8	
25	49	148	$2^2 \times 37$	5mod8	
26	51	154	$2^1 \times 77$	5mod8	
27	53	160	$2^5 \times 5$	Move back to 7th position	4(3n+1)
28	55	166	$2^1 \times 83$	3mod8	
29	57	172	$2^2 \times 43$	3mod8	

Explanation

If we apply 3n+1 to odd numbers the sequence we will get will be 4mod6 class . (Containing 2 distinct sequences and a self-similar loop like structure sequences are colored for understanding) now look at the class we get.

Like this (4, 10, 16, 22, 28, 34, 40, 46, 52, 58, 64, 70, 76, 82, 88….)

If we give position (start indexing terms) to each term of the sequence and extract the 3mod4 indices and divide those indices terms with 4 this is what will happen.

Terms on 3mod4 indices (16, 40, 64, 88, 112, 136, 160, 184, 208, 232, 256, 280, 304, 328, 352….)

Divide this with 4 (16/4, 40/4, 64/4, 88/4, 112/4, 136/4, 160/4…..)

After division we are again in the base sequence (4, 10, 16, 22, 28….) its kind of recursive loop in structure itself. See above table to understand and visualize it.

If we extract 3mod4 indices from our 2nd sequence (16, 40, 64, 88…) and divide with 16 we will again be in the base class.

3mod4 indices terms of 2nd sequence (64, 160, 256, 352, 448, 544, 640, 736, 832, 928, 1024, 1120, 1216, 1312, 1408)

Divide these with 16 we will again in the base class or base function .

Take 3mod4 indices of 3rd sequence divide with 64 again we will move back to the first sequence generated by the 3n+1 function.

And if we keep doing this, this function keeps giving us back its own copies with higher powers of 4. No matter where we are from which number we start with we always have this position in a system that will throw us back to the main function or let say smaller position. Note this, we will use it later.

Now here's how Collatz conjecture behave in different modular classes. For ease and understanding we will divide all odd numbers in 4 modular classes and the 2nd reason is these are the classes that hold the main blueprint of this function. We can't understand the function and its behavior without this division. Classes are

1 mod 8, 3 mod 8, 5 mod 8 and 7 mod 8 classes.

We will present classes in the same order in which it would be easy to understand the function's behavior, paths it takes and how it transitions between classes .

2. 7mod8 class behavior (Second invariant)

Collatz behavior in 7 mod 8 class, when we apply function to 7 mod 8 class it keeps alternating between 7mod8 and 3mod8 class based on its position. See table.

Position	7 mod 8 Class (n = 8i - 1)	Collatz Function Applied	Resulting Mod 8 Class
1	7	3×7+1 = 22 → 11	3 mod 8
2	15	3×15+1 = 46 → 23	7 mod 8
3	23	3×23+1 = 70 → 35	3 mod 8
4	31	3×31+1 = 94 → 47	7 mod 8
5	39	3×39+1 = 118 → 59	3 mod 8
6	47	3×47+1 = 142 → 71	7 mod 8
7	55	3×55+1 = 166 → 83	3 mod 8
8	63	3×63+1 = 190 → 95	7 mod 8
9	71	3×71+1 = 214 → 107	3 mod 8
10	79	3×79+1 = 238 → 119	7 mod 8
11	87	3×87+1 = 262 → 131	3 mod 8
12	95	3×95+1 = 286 → 143	7 mod 8
13	103	3×103+1 = 310 → 155	3 mod 8
14	111	3×111+1 = 334 → 167	7 mod 8
15	119	3×119+1 = 358 → 179	3 mod 8

On odd positions it moves to 3mod8 class and on even positions it moves forward in 7mod8 class this is where most consecutive odd even steps happen. Here the 3n+1 use 2 distinct rules positions based to map to the next class. We will present the formulas and shortcuts of calculations on how to calculate from which position it will move out from 7mod8 to 3mod8 class in the last section for now we have observed 2 invariants so far in its structure, a self recursive structure and its behavior in 7mod8 class.

3. 3mod8 class behavior (Third invariant)

Now we will see how this function behaves in the 3mod8 class. See table

Position	3 mod 8 Class (n = 8i - 5)	Collatz Function Applied	Resulting Mod 8 Class
1	3	3×3+1 = 10 → 5	5 mod 8
2	11	3×11+1 = 34 → 17	1 mod 8
3	19	3×19+1 = 58 → 29	5 mod 8
4	27	3×27+1 = 82 → 41	1 mod 8
5	35	3×35+1 = 106 → 53	5 mod 8
6	43	3×43+1 = 130 → 65	1 mod 8
7	51	3×51+1 = 154 → 77	5 mod 8
8	59	3×59+1 = 178 → 89	1 mod 8
9	67	3×67+1 = 202 → 101	5 mod 8
10	75	3×75+1 = 226 → 113	1 mod 8
11	83	3×83+1 = 250 → 125	5 mod 8
12	91	3×91+1 = 274 → 137	1 mod 8
13	99	3×99+1 = 298 → 149	5 mod 8
14	107	3×107+1 = 322 → 161	1 mod 8
15	115	3×115+1 = 346 → 173	5 mod 8

Again a very neat, clean and mappable structure. Very predictable alternation between 1mod8 and 5mod8 classes. We will have 2 distinct rules here position based to map to the next class. This class immediately sends a function for “convergence” on odd position it must be 3 or 3 plus consecutive divisions from odd positions the function will move to 5mod8 class (this is the main decision class; once we will see its structure we will understand) and 1mod8 class will provide two consecutive divisions. Here function will converge but on a lesser scale. Now here's how we will do mapping.

The 1 position of 7mod8 class will map to the 2nd position of 3mod8 class. (On 2nd position in 3mod8 class we have 11 and 7 returns 11 when we apply collatz function to 7)

The 3rd position of 7mod8 class will map to 5th position of 3mod8 class

The 5th position of 7mod8 class will map to 8th position of 3mod8 class.

Very neat clean mapping with clear arithmetic progression.

Now this is how we will map all 4 classes with each other.

4. 1mod8 class behavior (Fourth invariant)

Here is the structure of the 1mod8 class.

Position	1 mod 8 Class (n = 8i - 7)	Collatz Function Applied	Resulting Mod 8 Class
1	1	3×1+1 = 4 → 1	1 mod 8
2	9	3×9+1 = 28 → 7	7 mod 8
3	17	3×17+1 = 52 → 13	5 mod 8
4	25	3×25+1 = 76 → 19	3 mod 8
5	33	3×33+1 = 100 → 25	1 mod 8
6	41	3×41+1 = 124 → 31	7 mod 8
7	49	3×49+1 = 148 → 37	5 mod 8
8	57	3×57+1 = 172 → 43	3 mod 8
9	65	3×65+1 = 196 → 49	1 mod 8
10	73	3×73+1 = 220 → 55	7 mod 8
11	81	3×81+1 = 244 → 61	5 mod 8
12	89	3×89+1 = 268 → 67	3 mod 8
13	97	3×97+1 = 292 → 73	1 mod 8
14	105	3×105+1 = 316 → 79	7 mod 8
15	113	3×113+1 = 340 → 85	5 mod 8

Again very ordered class move function based on positions quite predictably based on the current position of the function. Here we will have 4 distinct functions through which class will map to other classes.

Now it's time for the 5mod8 class, this class structure is a bit different and tricky as in the beginning we saw the self recursive loop type structure in 3n + 1 function. Now this class represents the loop structure of the 3n +1 function. See table how.

5. 5mod8 class (Loop or self-similar structure)

n	3n + 1	Simplified 4(3n+1)	Match in 3n + 1 Series
5	16	4 × 4	4
13	40	4 × 10	10
21	64	4 × 16	16
29	88	4 × 22	22
37	112	4 × 28	28
45	136	4 × 34	34

Here's how the 5mod8 class looks when we apply the Collatz function on it. And in next table we will see how to tame it.

Position	5 mod 8 Class (n = 8i - 3)	Collatz Function Applied	Resulting Mod 8 Class
1	5	3×5+1 = 16 → 1	1 mod 8
2	13	3×13+1 = 40 → 5	5 mod 8
3	21	3×21+1 = 64 → 1	1 mod 8
4	29	3×29+1 = 88 → 11	3 mod 8
5	37	3×37+1 = 112 → 7	7 mod 8
6	45	3×45+1 = 136 → 17	1 mod 8
7	53	3×53+1 = 160 → 5	5 mod 8
8	61	3×61+1 = 184 → 23	7 mod 8
9	69	3×69+1 = 208 → 13	5 mod 8
10	77	3×77+1 = 232 → 29	5 mod 8
11	85	3×85+1 = 256 → 1	1 mod 8
12	93	3×93+1 = 280 → 35	3 mod 8
13	101	3×101+1 = 304 → 19	3 mod 8
14	109	3×109+1 = 328 → 41	1 mod 8
15	117	3×117+1 = 352 → 11	3 mod 8

Seems unpredictable and chaotic now let's see what's inside, just apply the 3n + 1 function and start writing it with the powers of 2 here how.

6. 5mod8 class (2 distinct sequences, loop structure invariants)

Position	5 mod 8 (n)	3n + 1	Power of 2 × Odd	Resulting mod class
1	5	16	$2^4 \times 1$	1mod8
2	13	40	$2^3 \times 5$	5mod8
3	21	64	$2^6 \times 1$	Move back to 1st position
4	29	88	$2^3 \times 11$	3mod8
5	37	112	$2^4 \times 7$	7mod8
6	45	136	$2^3 \times 17$	1mod8
7	53	160	$2^5 \times 5$	Move back to 2nd position
8	61	184	$2^3 \times 23$	7mod8
9	69	208	$2^4 \times 13$	5mod8
10	77	232	$2^3 \times 29$	5mod8
11	85	256	$2^8 \times 1$	Move back to 3rd position
12	93	280	$2^3 \times 35$	3mod8
13	101	304	$2^4 \times 19$	3mod8
14	109	328	$2^3 \times 41$	1mod8
15	117	352	$2^5 \times 11$	Move back to 4th position
16	125	376	$2^3 \times 47$	7mod8
17	133	400	$2^4 \times 25$	1mod8
18	141	424	$2^3 \times 53$	5mod8
19	149	448	$2^6 \times 7$	Move back to 5th position
20	157	472	$2^3 \times 59$	3mod8
21	165	496	$2^4 \times 31$	7mod8
22	173	520	$2^3 \times 65$	1mod8
23	181	544	$2^5 \times 17$	Move back to 6th position
24	189	568	$2^3 \times 71$	7mod8
25	197	592	$2^4 \times 37$	5mod8
26	205	616	$2^3 \times 77$	5mod8
27	213	640	$2^7 \times 5$	Move back to 7th position
28	221	664	$2^3 \times 83$	3mod8
29	229	688	$2^4 \times 43$	3mod8

The structure is clearly visible now. On even positions we have a sequence that looks like 2^3 (5, 11, 17…..) coloured.

On 1mod4 positions its has a distinct sequence 2^4 (1, 7, 13…) coloured.

These both sequences return mod8 classes in the same predictable rhythmic manner (it's beautiful just like music beats) like we saw in our previous 3 classes.. Now 3mod4 positions are left?

Its again the loop function and looks like 16(3n +1) and can be divided in 3 classes like we did for 5mod8 (4(3n+1)) class but we don't need to do so because we can simply write a function that these positions will feed back into 5mod8 class like a nested loop feedback to outer loop in computer programs.

first 3mod4 position will feedback to 1st position of the 5mod8 class

2nd 3mod4 position will feedback to 2nd position of the 5mod8 class

3rd 3mod4 position will feedback to the 3rd position of the 5mod8 class and here the powers of 2 will keep increasing so this class will keep becoming a massive black hole that will keep forcing numbers inside to reach back to 1.

So here is how we can calculate positions and correspondence numbers. And yes you can add even classes if you want to. I did only with odd numbers because it's less messy. Even classes can be added and functions could be changed accordingly based on the same position mapping. The main point is this is the blueprint of this machine and this is how this finite state machine is working. For position and number calculations look at the table.

Class	1 mod 8	3 mod 8	5 mod 8	7 mod 8
Calculate Position "p"	(n+7)/8	(n+5)/8	(n+3)/8	(n+1)/8
Calculate Number "n"	8p-7	8p-5	8p-3	8p-1

As we have observed earlier, only odd positions in the 7mod8 class will send a function to the 3mod8 class so here's how we can calculate the odd position if the function returns an even number as position.

Look at this Example:

Let's say p is 4. Function is 3p/2 until odd.

Apply: 4 × 3 / 2 = 6 (p is still even, apply the rule again)

6 × 3 / 2 = 9 , now p is an odd position it will move out of the class and go to 3mod8 class.

There's another way to calculate p. Let's say p is 56.

Divide 56 by 2 until it becomes odd: 56 = 2 × 2 × 2 × 7.

Now, p = 3 (number of times you divided by 2) × 7.

So, p = 3^3 × 7.

Once you calculate the position you can start applying these rules.

Class	Position	Rule Apply	Return Position in	Next Rule to Apply	Position dynamics current	Position dynamics next
7 mod 8	If Odd	(3p+1)/2	3 mod 8	Apply rules of 3 mod 8 class	Divergent	Divergent
7 mod 8	If Even	3p/2 until odd	7mod8	7 mod 8 rules.	Divergent	Divergent
3 mod 8	if odd	(3p-1)/2	5 mod 8	5 mod 8 rules	Divergent	Convergent
3 mod 8	if even	3p/2	1 mod 8	1 mod 8 rules	Divergent	Convergent
1 mod 8	if 1 mod 4	(3p+1)/4	1 mod 8	1 mod 8 rules	Convergent	Convergent
1 mod 8	if 2 mod 4	(3P-2)/4	7 mod 8	7 mod 8 rules	Convergent	Divergent
1 mod 8	if 3 mod 4	(3p-1)/4	5 mod 8	5 mod 8 rules	Convergent	Convergent
1 mod 8	if 0 mod 4	3p/4	3 mod 8	3 mod 8 rules	Convergent	Divergent
5 mod 8	if 3 mod 4	(p+1)/4	5 mod 8	5 mod 8 rules	Convergent	Convergent
5 mod 8	if 2 mod 8	(3p+2)/8	5 mod 8	5 mod 8 rules	Convergent	Convergent
5 mod 8	if 4 mod 8	(3p+4)/8	3 mod 8	3 mod 8 rules	Convergent	Divergent
5 mod 8	if 6 mod 8	(3p+6)/8	1 mod 8	1 mod 8 rule	Convergent	Convergent
5 mod 8	if 0 mod 8	3p/8	7 mod 8	7 mod 8 rules	Convergent	Divergent
5 mod 8	if 1 mod 16	(3p+13)/16	1 mod 8	1 mod 8 rules	Convergent	Convergent
5 mod 8	if 5 mod 16	(3p+1)/16	7 mod 8	7 mod 8 rules	Convergent	Divergent
5 mod 8	if 9 mod 16	(3p+5)/16	5 mod 8	5 mod 8 rules	Convergent	Convergent
5 mod 8	if 13 mod 16	(3p+9)/16	3 mod 8	3 mod 8 rules	Convergent	Divergent

With these rules we can see the current number mod class, position, everything, we can predict behavior like number is on even position of 7mod8 class so function will rise and so even we can predict after how many iterations number will drop.

- As 3mod4 position is a loop position in 5mod8 class so if it returns 3mod4 position again, keep applying the rule again until either it reaches 1 or return any other position.

We are almost done here. We have seen the whole underlying structure, working of the function now we will talk about non trivial cycles and convergence most favorite topics.

7. Non trivial cycles

We now have the whole set of rules, invariants and we know that no number can deviate from them, whether a single digit number or a million digit number every number has to follow the defined 17 rules (in table) and will follow the same defined paths.

So let's see how many possible combinational paths (tree) we can have that this function uses to reach back to 1. 7mod8 (2 paths but only 1 will interact outside the class) 3mod8 (2 paths) 1mod8 (4 paths) 5mod8 (9 paths)

So multiply them 2*4*9 = 72 paths in total. (Hopefully I am not doing it wrong)

Weak argument: while exploring the function upto 10^{18} we must have explored all these and if they didn't produced any cycle there they will not go in cycles for any number. (I like such arguments lazy yet confident, and I am at least providing the rule before making this argument)

Strong argument. We have no valid combination (with constraints these rules have)of these 17 rules that could produce the output that is equal to input in the same class we started with other than one exception where our little naughty cycle resides. The only function that outputs the same number as input is 3p+1/ 4 (1mod8 class rule for 1mod4 position) and this function only returns the same output when p=1. Ambitious souls can plz write all possible 72 paths just for fun and see if we can have any such combination where input = output in the same class. Tyring? Let's take another and easier approach.

And now here is the **strongest argument**. We will use the concept of predecessors and this time plz pay attention. So every class has defined paths as we have seen, but every number also has a “defined” predecessor that can never be equal to the number itself.

Start with first class 7omd8 we have predecessor for this class one predecessor is from 1mod8 class and 2nd is 7mod8 class, all the rest infinite many predecessors are in 5mod8 class in our loop function.

So when predecessor will be from 7mod8 class the number will be raising and no previous number will be equal to the current number (cycle isn't possible)

When predecessor will be from 1mod8 class of course this is distinct class and both 7mod8 and 1mod8 class don't share any common number so predecessor will never be equal of the number of 7mod8 class (cycles impossible because the number that will reach the next number should must be different) all the rest of predecessors will be from 5mod8 class. And this is our loop function looks like 3p+1 /4 here input and output could never be same other the one input that is 1 itself so this class can't map back to original input (cycles impossible)

Same is true for the rest of 2 classes.

Predecessors for 3mod8 will either be from class 7mod8 or 1mod8. Both classes don't share common numbers with the 3mod8 class so it will always be accessed from the numbers that can never be equal to the number that is being accessed. (Cycles impossible) all the rest predecessor will be in our loop function in 5mod8 class infinite many.

For 1mod8 class we have 2 predecessors one is from 3mod8 class (distinct class don't share common numbers, cycles impossible) and from 1mod8 class and when we have predecessors from 1mod8 class this is where class is moving back and finally dropping to our trivial cycle.

The 5mod8 class is a loop function and represents the whole 3n+1 function with higher divisibility so it always feeds back like an inner loop that terminates and feeds back to the outer loop. Cycles are not possible in any position other than 3mod4; the same logic is applicable as above, for 3mod4 positions they move backwards and finally drop to trivial cycles.

General Explanation.

Every number in collatz map has a defined set of predecessors, and no 2 numbers share the same predecessors.

For any cycle to exist we must have a number that can be its own predecessor or come in a chain at some point while except 1 no number can fulfill this condition. Here's why

The reverse function of collatz for any number n. Where n is a set of odd integers.

Function is.

(n.2^m -1)/3

When this function will take 2 different numbers as input, the output will definitely be different as well, no 2 numbers can satisfy the function for the same values. Hence the predecessor set for any number will be unique. And for any power of 2 when the function will generate the integer output that will be a valid predecessor of given n.

For example for 7 set of predecessors will be (this is infinite set)

7*2-1=13/3 .. no integer solution

7*4-1 =27/3 = 9 valid predecessor

7*8-1 = 55/3 invalid

7*16-1 = 111/3 = 37 valid predecessors and so on..

And as we have 3 in the denominator so this equation could only be equal to n when n<3 for any n≥3 this equation couldn't be equal to n itself, because no multiple of 3 can satisfy this equation for integer solution (we know that Collatz function don't return us any multiple of 3 in its sequence) but can start from any multiple of 3. So any multiple of 3 can itself be predecessor of any number but no number will be predecessor of any multiple of 3 (this is a cute little trick this function is playing to avoid cycles)

Now see example.

Let's take any easy Collatz sequence and start applying this predecessor thing to that. Let's take the function, let say 15. (We will deal only with odd numbers of sequence.)

1: 15 (no predecessor)

2: 23 (15, 61, 245, 981….) predecessor set

3: 35 (23, 93, 373, 1493…) predecessor set

4: 53 (35, 141, 565….) predecessor set

5: 5 (13, 53, 213…) predecessor set

6: 1 (1, 5, 21, 85…) predecessor set

- Notice how the predecessors are generating their own sequences? I just realized it at 2:30am in the night while trying to complete the document. Math is seductive, it won't let you sleep until satisfaction.

Now the argument is that when every number has a unique (disjoint, I hope I am using the right term) set of predecessors , no number can be its own predecessor except 1. How are cycles possible?

For any cycle to form, we need a number that appears twice in sequence but this unique predecessor set doesn't allow it to happen. So cycles are impossible.

Now anyone who thinks that it is about the predecessor and not the successor? Let's have a party. (I wish I could tell you that current successor will become a predecessor in next step, but I don't want to corrupt the innocence of your pure thoughts ❤️)

Just for fun let's see what kind of sequence it will make for 7 and any number ending on 9 (9 itself is orphan, predecessor-less) we already have seen for numbers ending in 1,3,5 let's explore the remaining 2 numbers.

7 (9, 37, 149, 597, 2389…..) predecessor set

19 (25, 101, 405, 1621…) predecessor set

So infinite many numbers, with infinite many disjoint predecessor sets, sets that looks like cousins of arithmetic progressions growing predictably with the difference of previous term into 4 (what we call this in standard maths? I don't know the term so let's call it CAP (Cousin of arithmetic progression). How are cycles possible when every number can be in the predecessor set just for once, and just for one unique number and can't be in its own predecessor set?

Think of it like a function that has infinitely many disjoint sets and every time it picks one number from any set and moves forward to reach back to 1. It doesn't use any set twice in one go.

- "second-order linear homogeneous recurrence relation" chatgpt told me this name for the above sequence based on its property but I am much better and easy with CAP.

8. Convergence vs divergence

We can take 3 approaches to prove that the system will always converge.

1. We list all possible paths based on previously discussed 17 rules and exactly find out the longest route a number can take. Hard choice ambitious souls report plz.
2. We simply list the divergence and convergence rate for any position in the above table and look for the overall trend, after all it is a system working coherently so definitely we can easily say convergence rates are higher so the system will converge over time if no non-trivial cycle exists. Better approach than the first one.
3. We just look at the table and say "hamm, more convergence points than divergence points, also nested loop like structure, it will definitely converge bro maths is democratic." chill, and move on.. laziest choice but easiest one I will take this one.

Okay let's generalize 3n+1 and see the convergence funda.

Generalization of (3n + 1) Functions

We explore functions of the form f(n) = (2^m - 1) + 1, and their self-similar behavior on specific modular positions. These functions exhibit structured repetition depending on the exponent m. This study provides a generalized rule for identifying such self-similarities. (See wrote by chatgpt, boring academic language gives me headache, like aliens talking to each other in secret code language, I read I mean try to read some Collatz papers and literally get terrified, bahi ko example to do pata to chaly keh kya rahy ho 🤔)

General Rule

For a function of the form f(n) = (2^m - 1) + 1:

- When m = 2 → Self-similarity occurs at positions congruent to 3 mod 4.
- When m = 3 → Self-similarity occurs at positions congruent to 5 mod 8.
- When m = 4 → Self-similarity occurs at positions congruent to 9 mod 16.
- When m = 5 → Self-similarity occurs at positions congruent to 17 mod 32.
- And so on.

Generalized Rule:
For exponent m, self-similarity occurs at positions congruent to (2^(m-1) + 1) mod 2^m.

Notes

1. Every system has exactly one fixed cycle at the starting position. No system will enter into any other cycle due to modular constraints.

2. Every system is convergent if it is of the form:
 (((2^m + 1) - 1) * n + 1)) when odd, and divide by 2^m when even

3. Adjust the rules in accordance with denominator eg. For 7n+1 , apply 7n+1 when odd, divide with 4 when even, if not divisible by 4, add 2 to make it divisible. (Or simply apply floor division to make sure numbers don't climb to roof (infinity)). In both cases it will come back home. (Want an example of that one drunk person who always comes back home?)

4. These systems are bounded systems , predictably transitioning in modular classes under given function.

Convergence vs divergence.

We will analyze two systems (3n+1 and 7n+1) with respect to 2 in basement (denominator) and see how they behave in terms of convergence and divergence.

As we know that both systems exhibit self-similarity so we will take one block and analyze that because every block will behave similarly in the system due to its self-similarity (in terms of convergence and divergence). If they don't, they don't belong to 3n+1 family, we look other families like 5n+1 later, but remember convergence always depends on denominator and not on self-similarity (i initially thought it do, i was mathematical kid that time just a week in mathematics, now i am bit grown up, like 2 and half month old, i can eat solid and can give bit solid statements, so yes self-similarity allow us to close (bound) the system and ensures non-trivial cycles exclusion, and denominator tell us weather system will drop or not).

3n + 1 system.

D = divergent

C = convergent

Position	Odd numbers	3n + 1	Power of 2 × Odd	Convergence vs divergence rate w.r.t 2
1	1	4	$2^2 \times 1$	-0.25 C
2	3	10	$2^1 \times 5$	+0.5 D
3	5	16	$2^3 \times 1$	-0.62 C
4	7	22	$2^1 \times 11$	+0.5 D
5	9	28	$2^2 \times 7$	-0.25 C
6	11	34	$2^1 \times 17$	+0.5 D
7	13	40	$2^3 \times 5$	-0.62 C
8	15	46	$2^1 \times 23$	+0.5 D
9	17	52	$2^2 \times 13$	-0.25 C
10	19	58	$2^1 \times 29$	+0.5 D
11	21	64	$2^6 \times 1$	-0.95 C

So our 3mod4 positions are not acting stable like other positions, we will simply expand the 3mod4 positions of the function and will average their behavior. (Average behavior still show a proportion of a convergence actual convergence of function will still remain higher than our calculations)

Our 5mod8 class is the representative of these 3mod4 positions..

So here is an expanded view. And we can see for half of positions its giving 3 divisions, for left 2 quarters its giving 4, and at least 5 divisions (if we expand further it will keep increasing but we average just first expansion (as our function is working on 4 modular classes so we don't need to go beyond 5mod8 class)

Position	5 mod 8 (n)	3n + 1	Power of 2 × Odd
1	5	16	$2^4 \times 1$
2	13	40	$2^3 \times 5$
3	21	64	$2^6 \times 1$
4	29	88	$2^3 \times 11$
5	37	112	$2^4 \times 7$
6	45	136	$2^3 \times 17$
7	53	160	$2^5 \times 5$
8	61	184	$2^3 \times 23$
9	69	208	$2^4 \times 13$
10	77	232	$2^3 \times 29$

11	85	256	$2^8 \times 1$
12	93	280	$2^3 \times 35$
13	101	304	$2^4 \times 19$
14	109	328	$2^3 \times 41$
15	117	352	$2^5 \times 11$

For 50% , 2^3 positions drop ratio will be 1 - (3/8) = 1 - 0.375 = 0.625 .

For 25% 2^4 drop ratio will be 1- (3/16) = 1- 0.1875 = 0.8125 .

For 25% 2^5 drop ration will be 1 - (3/32) = 1- 0.09375 = 0.90625 (or higher, as soon as number hit this position it will further go into 3,4,5 divisions right after one odd step, so it's kind of black hole.)

Lets take average. (0.90625+0.8125)/2 = 0.86

(0.86+0.63)/2 = 0.745 (round it to 0.75)

So on average 3mod4 position gives us 0.75 convergence.

Now we will take just one chunk from position 4 to 7 (because the function has self-similarity and it's going to repeat this pattern infinitely) and apply these values to see if the system is convergent or divergent

Position	Odd numbers	3n + 1	Power of 2 × Odd	Convergence vs divergence rate w.r.t 2
1	1	4	$2^2 \times 1$	
2	3	10	$2^1 \times 5$	
3	5	16	$2^3 \times 1$	
4	7	22	$2^1 \times 11$	+0.5 D
5	9	28	$2^2 \times 7$	-0.25 C
6	11	34	$2^1 \times 17$	+0.5 D
7	13	40	$2^3 \times 5$	-0.75 C
8	15	46	$2^1 \times 23$	
9	17	52	$2^2 \times 13$	
10	19	58	$2^1 \times 29$	
11	21	64	$2^6 \times 1$	

0.5+0.5 = 0.25+0.75 😂😂

Divergence = convergence 😂 I think they will cancel each other out. And system will stay on base singing (yh to wohe jaga hy guzaray thy ham jahan se)

The system is very stable !! So why does it converge ? First see how i calculated these ratios, i will calculate for 7n+1 as well and you will understand why that system diverges and this system doesn't. Here's an explanation.

When we multiply any number with 3 we have 3 similar chunks (ignore +1 as it doesn't add up much) so when we halve any number for once, one of its chunks is reduced and the 2nd chunk gets halved. The output number will always be around 50% (i wrote it as +0.5 in table) higher than input number and the system will be divergent here.

E.g input = 7, output = 11 (both 7mod8 and 3mod8 class will exhibit this behavior).

Now when we divide twice this is where the system actually starts converging and output number will be smaller than input number roughly 75% of input number. This is why I am using plus and minus signs because the system will only converge where output < input.

E.g input = 41, output = 31 (roughly 75% of input number, both 1mod8 and 5mod8 classes exhibit this behavior of decay)

Now what will this system do?? Mathematically it should converge, (the maximum convergence rate of 3mod4 positions is one that's why) it can only diverge if it doesn't believe in mathematics.

Let's see what 7n+1 does..

The system has self-similarity on 5mod8 positions so we will again analyze one chunk from position 6 to 13.

Position	Odd n	7n+1	Power of 2×odd	Con vs div w.r.t 2
1	1	8	$2^3 \times 1$	
2	3	22	2×11	
3	5	36	$2^2 \times 9$	
4	7	50	2×25	
5	9	64	$2^6 \times 1$	
6	11	78	2×39	1- (7/2) = 2.5 D
7	13	92	$2^2 \times 23$	1- (7/4) = 0.75 D
8	15	106	2×53	2.5 D
9	17	120	$2^3 \times 15$	1 - (7/8) = 0.125 C
10	19	134	2×67	2.5 D
11	21	148	$2^2 \times 37$	0.75 D
12	23	162	2×81	2.5 D
13	25	176	$2^4 \times 11$	1- (7/64) = 0.89 C

Here 5mod8 is our self-similar position no matter what power of 2 we choose to (i did with 2^6 because there is no point to expand it), you can try with any power because if 5mod8 positions even reach to 1 (it can never exceed 1) we still have higher divergent ratio than convergence.

Let's calculate

4(2.5) + 2(0.75) > 0.125 + 1 (assuming the maximum convergence on 5mod8 positions)

With respect to 4 the system will immediately become stable and reach back to 1, we just need to do floor division.

Formalism

Neither am I a mathematician nor a computer scientist, little knowledge that I have, I have from my old school days (25 years ago) so I have forgotten most of the stuff though I used to enjoy cracking puzzles but of course that was a past. This work is kind of a compensation for my failed math exams and incomplete cs courses. Even if I can't formalise it at the moment, the “logical structure” and rules and logic itself are still clear, as are the calculations.

I believe today my work on Collatz conjecture has been completed. It took longer than expected because firstly I hate hard work, secondly I had to do all this on a mobile device. So plz keep ignoring formatting, table’s heart melting break ups (across pages) and inconsistency of commitment in writing.

References

My pen, my notebook, my brain, chatgpt for converting my hand written tables in pdf, (though it made some terrible mistakes that i tried my best to correct in this document but if any left in any function you can map positions yourself, you can calculate yourself) deepseek for testing the rules on 27 (nightmare number for Collatz Conjecture fans) and some other small numbers.

Request

Plz prove me wrong about the structure and of course non trivial cycles arguments 😊.

Waiting for your insights (infact I am only waiting to be praised 😂)

On a serious note, work is completely serious though I myself prefer to do boring work in a cheerful manner else you know everyone hates math for a reason being dry.

samiyaparacha@gmail.com