A Modular Encoding and Symbolic Structure in the 3n + 1 Process

Author: Samiya Akhtar

Abstract
This paper presents a logic-based approach to exploring the structure of the Collatz Conjecture (also known as the 3n + 1 problem). The goal is to reveal the hidden modular patterns, relationships, and encodings within the sequence and it operates underneath.

1. Introduction
The Collatz Conjecture has long been considered resistant to structure. Despite countless attempts, no formal proof has yet resolved its mystery. This work takes a different path — instead of chasing a proof while testing on numbers, it focuses on decoding structure through modular reasoning.

This method was developed independently, without formal affiliation to an academic institution.

2. Method and Modular Structure
3n+1 structure.

Apply 3n+1 to odd numbers the sequence you will get will be 4mod6 class .

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

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

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…..)

After division we are again in the base sequence (4, 10, 16, 22, 28….) its kind of recursive loop in structure itself.

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

3mod4 indices 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. For now just note that 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.

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 and how it transitions between classes .

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. 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 things so far in its structure, a self recursive structure and its behavior in 7mod8 class.

On a lighter note as its structural fact no odd position of 7mod8 can say "oh again that boring 3mod8 class let me step out somewhere"

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	$3×3+1 = 10 \to 5$	5 mod 8
2	11	$3×11+1 = 34 \to 17$	1 mod 8
3	19	$3×19+1 = 58 \to 29$	5 mod 8
4	27	$3×27+1 = 82 \to 41$	1 mod 8
5	35	$3×35+1 = 106 \to 53$	5 mod 8
6	43	$3×43+1 = 130 \to 65$	1 mod 8
7	51	$3×51+1 = 154 \to 77$	5 mod 8
8	59	$3×59+1 = 178 \to 89$	1 mod 8
9	67	$3×67+1 = 202 \to 101$	5 mod 8
10	75	$3×75+1 = 226 \to 113$	1 mod 8
11	83	$3×83+1 = 250 \to 125$	5 mod 8
12	91	$3×91+1 = 274 \to 137$	1 mod 8
13	99	$3×99+1 = 298 \to 149$	5 mod 8
14	107	$3×107+1 = 322 \to 161$	1 mod 8
15	115	$3×115+1 = 346 \to 173$	5 mod 8

Again a very neat, clean and mappable structure. Very predictable alternation between 1mod8 and 5mod8 classes. This class immediately sends a function for "convergence" on odd position it must be 3 or 3 plus consecutive divisions from odd positions function will move to 5mod8 class (this the main decision class once we will see its structure we will understand for now just trust me) and 1mod8 class will provide two consecutive divisions. Here function will converge but on a lesser scale. Now take out your pens, laptops and strat mapping positions yourself it would be like

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. Now we will see the structure of 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.

Yes you can think of cycle or loop here but in the first position this class is mapping back to itself and returning the same position and this is the only position where we get 4 - 2 - 1 loop. No other position has such similarity or characteristics.

All positions afterwards will move outwards with a clear arithmetic progression so no cycle will be possible based on this outward movements of position or in other words position 1 is the only position that return itself no other position do this and every class has distinct entry and exit points like function move to class from different position and function return back that position from different position of other class. This is why cycles don't happen.

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 that loop structure of the 3n +1 function. See table how.

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

So why cycles aren't possible here is because this movement is linear not cyclic. This class moves back to the original function like if you throw a ball high in the sky the ball will come back not go into loop this is how this function works. Like if you write the 3n + 1 and start extracting 3mod4 positions and keep dividing with powers of 4 you will keep getting back the main function its kind of straight backwards movement not cyclic backwards movement like a nested loop terminate and move back to original loop (Cs loop structure nested-loop its work in the same manner). Now let's see its structure first about how it looks after we apply the Collatz function and in the next table we will understand 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.

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	$2^5 \times 11$
16	125	376	$2^3 \times 47$
17	133	400	$2^4 \times 25$
18	141	424	$2^3 \times 53$
19	149	448	$2^6 \times 7$
20	157	472	$2^3 \times 59$
21	165	496	$2^4 \times 31$
22	173	520	$2^3 \times 65$
23	181	544	$2^5 \times 17$
24	189	568	$2^3 \times 71$
25	197	592	$2^4 \times 37$

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

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

These both sequences return mod8 classes in the same predictable manner 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 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	(n+7)/8	(n+5)/8	(n+3)/8	(n+1)/8
Calculate Number	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 move out of the class to 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
7 mod 8	Odd	(3p+1)/2	3 mod 8	Apply rules of 3 mod 8 class
3 mod 8	if odd	(3p-1)/2	5 mod 8	5 mod 8 rules
3 mod 8	if even	3p/2	1 mod 8	1 mod 8 rules
1 mod 8	if 1 mod 4	(3p+1)/4	1 mod 8	1 mod 8 rules
1 mod 8	if 2 mod 4	(3P-2)/4	7 mod 8	7 mod 8 rules
1 mod 8	if 3 mod 4	(3p-1)/4	5 mod 8	5 mod 8 rules
1 mod 8	if 0 mod 4	3p/4	3 mod 8	3 mod 8 rules
5 mod 8	if 3 mod 4	(3p+1)/8	5 mod 8	5 mod 8 rules
5 mod 8	if 2 mod 8	(3p+2)/8	5 mod 8	5 mod 8 rules
5 mod 8	if 4 mod 8	(3p+4)/8	3 mod 8	3 mod 8 rules
5 mod 8	if 6 mod 8	(3p+6)/8	1 mod 8	1 mod 8 rule
5 mod 8	if 0 mod 8	3p/8	7 mod 8	7 mod 8 rules
5 mod 8	if 1 mod 16	(3p+13)/16	1 mod 8	1 mod 8 rules
5 mod 8	if 5 mod 16	(3p+1)/16	7 mod 8	7 mod 8 rules
5 mod 8	if 9 mod 16	(3p+5)/16	5 mod 8	5 mod 8 rules
5 mod 8	if 13 mod 16	(3p+9)/16	3 mod 8	3 mod 8 rules

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

All these functions are reversible of course there is a short cut for reverse tree as well but let you yourself find that out. 😊 It's your homework.

3. Observations

We have just one class 7mod8 that makes the function raise, and even in that one class every 3mod4 position moves to the 5mod8 class after just one odd step in the 3mod8 class means instant convergence.

Even positions make more noise but still with limitations. On the other hand, the 5mod8 class has no limitations; its loop function can force you to reach back to base even from astronomically large numbers.

Convergence is inevitable because of the nested-loop function in the structure of the 3n+1 function. The 3mod8 class doesn't play any significant role in the function to raise; rather it acts like a bridge between raise and convergence, 1mod8 class plays a role in convergence but not that massively as the 5mod8 class does.

4. Closing lighter Note

Collatz function turns out to be a simple finite state machine operating on modular classes rather than directly on numbers . This work is compensation for my 2 failed math exams and my incomplete CS course.

Besides i want to say that as i am a bed bound patient (for now) and is unable to sit for longer times, i did all this by hand while laying on bed without laptop so mostly depending on chatgpt to convert my hand written tables in to pdf so yes mistakes happen that i corrected in this version. Hopefully it won't have any mistake else i believe anyone who is investing his/her time in understanding will spot mistakes and will be able to correct them if any are left.

References

My pen, my notebook and my brain that always failed to understand formalism.