Projekt JOVIALISMUS

How to proof the Collatz Conjecture

Joerg Drescher
Kyiv / Berlin
February / March 2022

Projekt Jovialismus
Kotowskaia Uliza 11, Ap. 5
04060 Kyiv - Ukraine
http://www.3n-1.de
iovialis@gmx.de

Hello world!

My name is Joerg Drescher. I am originally from Germany, but I used to live in Kyiv/Ukraine. Some days before the war began, I announced at Facebook an approach to proof the Collatz conjecture, which is said to be one of the hardest unsolved mathematic problems in the world. But I had to flee from my home in Kyiv and was not able to publish my solution, yet. Who does not know about the problem, there is a link to a video, which explains the problem.

In mid-January 2022 I watched exactly this video and started to write PHP scripts to understand, why it is so hard to solve this problem. By accident I saw an approach for a solution and was wondering, why it is so difficult and no one found an answer, yet.

I want to thank Dr. Edmund Weitz, a professor for mathematics in Hamburg for his inspiring videos, in which he explains mathematics for everyone. My brother contacted him while I was fleeing from Kyiv, but he refused to look at this paper.

And yes, I understood him, because why should an unknown person with no further reputation on this problem find a reasonable solution in such a short time. I did not read any papers on the problem and even I doubt, that a professional journal would ever look at a paper from somebody like me.

This is why I choose the way of publishing this paper, in which I use relatively simple mathematic methods. It should be possible for everyone with basic mathematic skills to understand, how the Collatz conjecture behaves. You should only know about modulo and binary numbers.

Finally, I want to thank the open source community, which gave me free tools to analyse the Collatz conjecture. This is why I publish my approach under the Creative Commons license, so everyone can check this approach.

OK, let's get started.

cc BY SA

A brief explanation of the problem

The mathematician Lothar Collatz introduced 1937 the idea, that repeating two simple arithmetic operations – if a natural number is even, half it until it becomes odd; if a natural number is odd, multiply it by 3 and add 1 – will lead always to a loop of 4-2-1. There are several other names for this conjecture named after the mathematicians Stanislaw Ulam, Shizuo Kakutani, Sir Bryan Thwaites or Helmut Hasse. The conjecture is also simply called 3n+1 problem. You can read more on this at Wikipedia.

Let's check my birth month's number 7 and what is happening:

7 is odd, so multiply it by 3 which gives us 21 and add 1 which result is 22. 22 is even, so divide it by 2 – 11. 11 is odd, multiplied by 3 plus 1 – 34. 34 even, divided by 2 – 17. 17 odd, multiplied by 3 plus 1 – 52. 52, even, divided by 2, 26, even, divided by 2, 13, odd, multiplied by 3 plus 1, 40, even, divided by 2, 20, even, divided by 2, 10, even, divided by 2, 5, odd, multiplied by 3 plus 1, 16, even, divided by 2, 8, even, divided by 2, 4, even, divided by 2, 2, even, divided by 2, 1, odd, multiplied by 3 plus 1, 4, even, divided by 2, 2, even, divided by 2, 1 and so on.

You can do this with any number and the respective sequence will always end up at the loop 4-2-1. However, there are numbers, e.g. like 27, which lead to surprisingly high turning points.

To proof the conjecture for every natural numbers, you need to show, that all end up in this loop and that there is no sequence, which runs into infinity or into another loop. The last happens, if you use negative numbers – there are three loops. This instruction is also known as 3n-1.

Table of content

1. Numbers within the Collatz sequences to be watched

It is evident, that you have even and odd numbers while executing the Collatz instruction. However, there are triples, too, to be watched. Odd triples appear only once by halving even triple numbers until, they are odd. After they are odd, they are multiplied by 3 and after adding 1, they are no triples anymore.

So far, so good.

CC BY SA

2. Reverse of the Collatz Conjecture

Reversing the Collatz conjecture means, that you can reach any natural number by starting with 1. But what are the instructions? Doubling an odd or even number leads both times to an even number. However, some even numbers have odd parents, which were multiplied by 3 and their result increased by 1. Thus, we need to identify such even numbers decreased by 1 and which then can be divided by 3:

n	(n-1) / 3
1	0
2	
3	
4	**1**
5	
6	
7	2
8	
9	
10	**3**
11	
12	
13	4
14	
15	
16	**5**
17	
18	
19	6
20	
21	
22	**7**
23	
24	
25	8
26	
27	
28	**9**
29	
30	
31	10

We see, every third number, starting at 4 can be divided by 3 after the number was decreased by 1. However, the result is sometimes odd and sometimes even. Only every sixth number starting with 4 is even and can be divided by 3 after the number was decreased by 1. In math you can express this by using modulo:

```
n mod 6 = 4
```

So far, so good.

CC BY SA

3. Analysing even numbers within Collatz sequences

Even numbers of the form $n*2^x$ (x from 0 to ∞) end up after x iterations in the odd number n. You can delete all ending zeros of binary numbers without influencing its leading binary structure:

n (dec)	n (bin)	n (cleaned bin)	n (cleaned dec)
2	**10**	**1**	**1**
4	100	1	1
6	**110**	**11**	**3**
8	1000	1	1
10	**1010**	**101**	**5**
12	1100	11	3
14	**1110**	**111**	**7**
16	1000	1	1
18	**10010**	**1001**	**9**
20	10100	101	5
22	**10110**	**1011**	**11**
24	11000	11	3
26	**11010**	**1101**	**13**
28	11100	111	7
30	**11110**	**1111**	**15**
32	100000	1	1

This method shortens up the iterations and shows, that there are some even numbers, decreasing more than others, but at least by 2. And it is happening regularly:

- Each second number is even and decreases only by 2 => $n \bmod 4 = 2$
- Each forth even number is even and decreases by more than 2 => $n \bmod 4 = 0$

So far, so good.

CC BY SA

4. Analysing odd numbers within a Collatz sequence

Following, what we experienced with even numbers, we check, what is happening with odds, when we apply the Collatz rules (3n+1; halving until its result is odd):

n	3n+1 (cleaned)	n (bin)	3n+1 (cleaned bin)
1	4 (1)	1	100 (1)
3	**10 (5)**	**11**	**1010 (101)**
5	16 (1)	101	10000 (1)
7	**22 (11)**	**111**	**10110 (1011)**
9	28 (7)	1001	11100 (111)
11	**34 (17)**	**1011**	**100010 (10001)**
13	40 (5)	1101	101000 (101)
15	**46 (23)**	**1111**	**101110 (10111)**
17	52 (13)	10001	110100 (1101)
19	**58 (29)**	**10011**	**111010 (11101)**
21	64 (1)	10101	1000000 (1)
23	**70 (35)**	**10111**	**1000110 (100011)**
25	76 (19)	11001	1001100 (10011)
27	**82 (41)**	**11011**	**1010010 (101001)**
29	88 (11)	11101	1011000 (1011)
31	**94 (47)**	**11111**	**1011110 (101111)**

We can say, that applying the Collatz rule for odd numbers with:

- `n mod 4 = 3` leads to even numbers with `n mod 4 = 2`
- `n mod 4 = 1` leads to even numbers with `n mod 4 = 0`

We recognize, that there are two kinds of odd numbers: those, which increase the next “net” odd number, and those, which decrease the next “net” odd number. And it is happening regularly:

- There are odd number increasing the “net” result => `n mod 4 = 3`
- There are odd number decreasing the “net” result => `n mod 4 = 1`

It means, that we have also with “cleaned net” odd numbers an up and down of the sequence. Thus, we can ignore even numbers and concentrate on “net” odd numbers, which binary structures do change after applying the Collatz rule.

So far, so good.

CC BY SA

5. Using n+(n+1)/2 instead of (3n+1)/2

What we experienced by analysing the relation of odd to even numbers, leads us to the question, why and how this is happening. The "secret" lies in how multiplying odd numbers with 3 and then adding 1. There are several possibilities to do this (here as example with the binary number for 27 and 255):

1) 3 * n + 1 = n + n + n + 1

11011_2	(27_{10})	11111111_2	(255_{10})
+ 11011_2	(27_{10})	+ 11111111_2	(255_{10})
+ 11011_2	(27_{10})	+ 11111111_2	(255_{10})
+ 1_2	(1_{10})	+ 1_2	(1_{10})
--------		-----------	
1010010_2	(82_{10})	1011111110_2	(766_{10})

2) 2n + n + 1

110110_2	(54_{10})	111111110_2	(510_{10})
+ 11011_2	(27_{10})	+ 11111111_2	(255_{10})
+ 1_2	(1_{10})	+ 1_2	(1_{10})
--------		-----------	
1010010_2	(82_{10})	1011111110_2	(766_{10})

As we see, the last digit is always 0, which means, the number is even. Therefore, we can divide the result suddenly with 2. For the last case, this is interesting, because we receive:

3.1) n + (n + 1) / 2

11011_2	(27_{10})	11111111_2	(255_{10})
+11110_2	(14_{10})	+10000000_2	(128_{10})
-------		----------	
101001_2	(41_{10})	101111111_2	(383_{10})

All ending ones (until the next 0, if there is one) change to zeros and they act as "transparent mask".

To make it clearer, here the sequence of only odds from 255 to 1

Iteration	n (dec) (net)	n (bin) (net)	(n+1)/2 (bin)
1.	255	11111111	1000000
3.	383	101111111	11000000
5.	575	1000111111	100100000
7.	863	1101011111	110110000
9.	1295	10100001111	1010001000
11.	1943	11110010111	1111001100
13.	2915	101101100011	10110110010
15.	4373	1000100010101	1000100010011
22.	205	11001101	1100111
26.	77	1001101	100111
30.	29	11101	1111
34.	11	1011	110
36.	17	10001	1001
39.	13	1101	111
43.	5	101	11
48.	1	1	

cc BY SA

The ending ones in binary of n are erased by executing the algorithm until the number ends up with `zzzzzzz01`. This means, that a number `yyyyyyy10x11` (`x` = as many ones as you like) stays `n mod 4 = 3` until the number reaches `zzzzzzz01 (n mod 4 = 1)`. However, the structures `yyyyyyy` and `zzzzzzz` change by each iteration while the decimal value of the number grows.

The reason for this behaviour is, by adding `1` to a number with a structure like `yyyyyyy0x1 (x` = as many ones as you wish, even none), the ending one(s) change(s) to zero(s) until they reach the next `0` (which will change to `1`). And because of halving them, the number to be added has the structure `yyyyyyy1X (X` = as many zeros as before were one(s)).

It looks like this in binary system:

  `yyyyyyy0x1 (x` = as many ones as you wish, even none)
\+ `yyyyyyyy1X (X` = one less zero as before were one(s))

Thus, the next number of the iteration keeps its ending structure decreased by one 1 until the next 0.

The decimal value of numbers n, ending with `yyyyyyy01` shrink until the last two digest ends up with `zzzzzzz11`. They change the structure `yyyyyyy` by each iteration while the decimal value of n decreases after cleaning the ending zeros. The amount of how many zeros are deleted, depends on the structure of `yyyyyyy01` and decreases its length at least by one. There are four rules for decreasing odd numbers (n mod = 4):

`yyyyyyy0001`	changes after one iteration to	`zzzzzzz01(00)`	`mod 4 = 1`
`yyyyyyy1001`	changes after one iteration to	`zzzzzzz11(00)`	`mod 4 = 3`
`yyyyyyy1101`	changes after one iteration to	`zzzzzzz101(000)`	`mod 4 = 1`
`yyyyyyy(01)01`	changes after one iteration to	`zzzzzzz1((00)00)`	`mod 4 = 1 or 3`

The fourth rule needs an explanation: If you have a repeating 01 pattern at the end of a binary number, each 01 becomes 00.

So far, so good.

6. How to get from odds to odds in both directions within a Collatz sequence

We know the path from one odd number to another odd number within a Collatz sequence: multiplying the odd number with 3 and adding 1 and afterwards we delete the zeros of the binary number. However, how to get from this new odd number backwards, if we cannot say, how many zeros were deleted? How often do we need to multiply an odd number by 2?

We know for sure: odd triples cannot be reached by another odd number and thus there is no action for them necessary. The other odd numbers can be identified by `n mod 3 = 1 or 2`.

Adding zeros to a binary number means multiply them by 2^x. The condition must be: $((n*(2^x))-1)/3$ or in math expression: $((n*(2^x))-1) \bmod 3 = 1, 2$. Thus, it depends on `n` and `x`, if the result of $(n*(2^x))-1$ can be divided by `3`.

First, the condition for x and if 2^x-1 (it is the case for `n=1` with `n mod 3 = 1`) can be divided by `3`. This is true for all even `x`. And all further n with the condition `n mod 3 = 1` multiplied by 2^x (with even `x>1`) stay `n mod 3 = 1`. If we subtract 1, they can be divided by `3`.

Further: 2^x with odd `x` multiplied by `n mod 3 = 2` change to `n mod 3 = 1.` Thus, if we subtract 1, we have again an even number, which can be divided by 3.

We receive this table for all odd `n` and all `x>=1`:

n (mod 3)	**$(n*(2^x)-1)/3$ (even x)**	**$(n*(2^x)-1)/3$ (odd x)**
1 (1)	1 (2), 5 (4), 21 (6) …	None
3 (0)	None	None
5 (2)	None	**3 (1),** 13 (3), 53 (5) …
7 (1)	9 (2), 37 (4), 149 (6) …	None
9 (0)	None	None
11 (2)	None	**7 (1),** 29 (3), 117 (5) …
13 (1)	17 (2), 69 (4), 277 (6) …	None
15 (0)	None	None
17 (2)	None	**11 (1),** 45 (3), 181 (5) …
19 (1)	25 (2), 101 (4), 405 (6) …	None
21 (0)	None	None
23 (2)	None	**15 (1),** 61 (3), 245 (5) …
25 (1)	33 (2), 133 (4), 533 (6) …	None
27 (0)	None	None
29 (2)	None	**19 (1),** 77 (3), 309 (5) …
31 (1)	41 (2), 165 (4), 661 (6) …	None
n mod 3 = 0	None	None
n mod 3 = 2	None	**$row_{column=0} = n-(n+1)/3$ $row_{column+1} = row_{column}*4+1$**
n mod 3 = 1	**$row_{column=0} = n+(n-1)/3$ $row_{column+1} = row_{column}*4+1$**	None

Written by Joerg Drescher, March.2022

Each given number of column 2 or 3 multiplied by 3 and its result plus 1 and then divided by 2 until the number is odd, will result in the number of column 1.

We can call this table a map, which works for the opposite direction like this: for example, you start at 1 and look for any number in column 2 (e.g. 5) than you go to the respective number (5) in the first column and there you choose the next number you wish (e.g. 13). Then, you look for this number (13) in column 1 and select a new number (e.g. 17). In column 1 at this new number (17) you select again a number (e.g. 45). If you end up in a triple, it means, you can only double the result.

You can use the map also in the original direction: for example, you want to explore the path from 7 to 4-3-1. For this approach, search for number 7 in column 2 or 3, select its number from column 1 (7 is at 11) and search again for this number in column 2 or 3 (for 11 it is 17). You can repeat this as you end up at 1 (`17 => 13, 13 => 5, 5 => 1`).

Further we see, that every third odd number (`n mod 3 = 2`) has only one parent smaller than its origin (we remember, this behaviour appears with numbers `n mod 4 = 3`). All other results are `n mod 4 = 1`.

This map shows also, that all odd numbers are unique represented.

So far, so good.

1. Numbers within 3n-1 sequences to be watched

It is evident, that you have even and odd numbers while executing the Collatz instruction. However, there are triples, too, to be watched. Odd triples appear only once by halving even triple numbers until, they are odd. After they are odd, they are multiplied by 3 and after subtracting 1, they are no triples anymore.

There is no difference of numbers to be watched between 3n+1 and 3n-1 sequences

So far, so good.

2. Reverse of the 3n-1

Reversing 3n-1 means, that you can reach any natural number by starting with 1 (or from another loop beginning). But what are the instructions? Doubling an odd or even number leads both times to an even number. However, some even numbers have odd parents, which were multiplied by 3 and their result decreased by 1. Thus, we need to identify such even numbers increased by 1 and which then can be divided by 3:

n (decimal)	(n+1) / 3 (decimal)
1	0
2	**1**
3	
4	
5	
6	
7	
8	**3**
9	
10	
11	
12	
13	
14	**5**
15	
16	
17	
18	
19	
20	**7**
21	
22	
23	
24	
25	
26	**9**
27	
28	
29	
30	
31	

We see, every third number, starting at 2 can be divided by 3 after the number was increaded by 1. However, the result is sometimes odd and sometimes even. Only every sixth number starting with 2 is even and can be divided by 3 after the number was increased by 1. In math you can express this by using modulo:

```
n mod 6 = 2
```

The difference from 3n-1 to 3n+1 is, that odd numbers start to appear earlier, but in the same frequency (every sixth number).

So far, so good.

3. Analysing even numbers within 3n-1 sequences

Even numbers of the form $n*2^x$ (x from 0 to ∞) end up after x iterations in the odd number n. You can delete all ending zeros of binary numbers without influencing its leading binary structure:

n (dec)	n (bin)	n (cleaned bin)	n (cleaned dec)
2	**10**	**1**	**1**
4	100	1	1
6	**110**	**11**	**3**
8	1000	1	1
10	**1010**	**101**	**5**
12	1100	11	3
14	**1110**	**111**	**7**
16	1000	1	1
18	**10010**	**1001**	**9**
20	10100	101	5
22	**10110**	**1011**	**11**
24	11000	11	3
26	**11010**	**1101**	**13**
28	11100	111	7
30	**11110**	**1111**	**15**
32	100000	1	1

This method shortens up the iterations and shows, that there are some even numbers, decreasing more than others, but at least by 2. And it is happening regularly:

- Each second number is even and decreases only by 2 => $n \bmod 4 = 2$
- Each forth even number is even and decreases by more than 2 => $n \bmod 4 = 0$

There is no difference between 3n+1 and 3n-1

So far, so good.

4. Analysing odd numbers within a 3n-1 sequence

Following, what we experienced with even numbers, we check, what is happening with odds, when we apply the rule of 3n-1 and halving its result until it is odd:

n	3n-1 (cleaned)	n (bin)	3n-1 (cleaned bin)
1	2 (1)	1	10 (1)
3	**8 (1)**	**11**	**1000 (1)**
5	**14 (7)**	**101**	**1110 (111)**
7	**20 (5)**	**111**	**10100 (101)**
9	26 (13)	1001	11010 (1101)
11	**32 (1)**	**1011**	**100000 (1)**
13	38 (19)	1101	100110 (10011)
15	**44 (11)**	**1111**	**101100 (1011)**
17	50 (25)	10001	110010 (11001)
19	**56 (7)**	**10011**	**111000 (111)**
21	62 (31)	10101	111110 (11111)
23	**68 (17)**	**10111**	**1000100 (10001)**
25	74 (37)	11001	1001010 (100101)
27	**80 (5)**	**11011**	**1010000 (101)**
29	86 (43)	11101	1010110 (101011)
31	**92 (23)**	**11111**	**1011100 (10111)**

We can say, that applying the 3n-1 rule for odd numbers with:

- `n mod 4 = 1` leads to even numbers with `n mod 4 = 2`
- `n mod 4 = 3` leads to even numbers with `n mod 4 = 0`

We recognize, that there are two kinds of odd numbers: those, which increase the next "net" odd number, and those, which decrease the next "net" odd number. And it is happening regularly:

- There are odd number increasing the "net" result => `n mod 4 = 1`
- There are odd number decreasing the "net" result => `n mod 4 = 3`

It means, that we have also with "cleaned net" odd numbers an up and down of the sequence. Thus, we can ignore even numbers and concentrate on "net" odd numbers, which binary structures do change after applying the 3n-1 rule.

The behaviour differs from 3n+1 to 3n-1 while the result of `n mod 4` is inverted.

Further, there are two ending loops visible:

1-1
5-7-5

So far, so good.

cc BY SA

5. Using n+(n-1)/2 instead of (3n-1)/2

What we experienced by analysing even and odd numbers, leads us to the question, why and how their up and down is happening. When we use within the binary system n+(n-1)/2 instead of (3n-1)/2, we can recognize following relation between n and (n-1)/2:

Iteration	n (dec) (net)	n (bin) (net)	(n-1)/2 (bin)
1.	85	1010101	10101001
3.	127	1111111	111111
6.	95	1011111	101111
9.	71	1000111	100011
12.	53	110101	1101001
14.	79	1001111	100111
17.	59	111011	11101
22.	11	1011	101
28.	1	1	1

The ending ones in binary of n are erased by executing the algorithm until the number ends up with `zzzzzzz01`. This means, that a number `yyyyyy0x11` (x = as many ones as you like) stays `n mod 4 = 3` until the number reaches `zzzzzzz01 (n mod 4 = 1)`. However, the binary structures `yyyyyyy` and `zzzzzzz` change by each iteration while the decimal value of the number decreases.

The reason for this behaviour is, by subtracting `1` of a number with a binary structure like `yyyyyyy0x11 (x` = as many ones as you wish, even none), the ending x11 changes to x10. And because of halving it, the number to be added has the structure `yyyyyyyx1 (x` = as many ones as before were chosen). Adding `yyyyyyy0x1` to `yyyyyyy0x11` has the result `zzzzzzz10(X)10 (`the new X has two ones less).

The decimal value of numbers n, ending with a binary structure like `yyyyyyy01` grows until the last two digest ends up with `zzzzzzz11`. They change the binary structure `yyyyyyy` by each iteration while the decimal value of n increases after cleaning the ending zeros. The amount of how many zeros are deleted, depends on the structure of `yyyyyyy01`. There are four rules:

`yyyyyyy1101`	changes after one iteration always to	`zzzzzzz011(0)`	`mod 4 = 3`
`yyyyyyy0101`	changes after one iteration always to	`zzzzzzz111(00)`	`mod 4 = 3`
`yyyyyyy0001`	changes after one iteration always to	`zzzzzzz001(0)`	`mod 4 = 1`
`yyyyyyy1001`	changes after one iteration always to	`zzzzzzz101(0)`	`mod 4 = 1`

So far, so good.

6. How to get from odds to odds in both directions within a 3n-1 sequence

We know the path from one odd number to another odd number within a Collatz sequence: multiplying the odd number with 3 and adding 1 and afterwards we delete the zeros of the binary number. However, how to get from this new odd number backwards, if we cannot say, how many zeros were deleted? How often do we need to multiply an odd number by 2?

We know for sure: odd triples cannot be reached by another odd number and thus there is no action for them necessary. The other odd numbers can be identified by `n mod 3 = 1 or 2`.

Adding zeros to a binary number means multiply them by 2^x. The condition must be: $((n*(2^x))+1)/3$ or in math expression: $((n*(2^x))+1) \bmod 3 = 1, 2$. Thus, it depends on n and x, if the result of $(n*(2^x))-1$ can be divided by 3.

First, the condition for x and if 2^x+1 (it is for the case n=1 with `n mod 3 = 1`) can be divided by 3. This is true for all odd x. And all further n with the condition `n mod 3 = 1` multiplied by 2^x (with odd x>=1) change to `n mod 3 = 2`. If we add 1, they can be divided by 3.

Further 2^x with even x multiplied by `n mod 3 = 2` stay `n mod 3 = 2`. Thus, if we add 1, we have again an even number, which can be divided by 3.

We receive this table for all odd n and all x>=1:

n (mod 3)	**$(n*(2^x)+1)/3$ (odd x)**	**$(n*(2^x)+1)/3$ (even x)**
1 (1)	1 (1), 3 (3), 11 (5) …	None
3 (0)	None	None
5 (2)		**7 (2), 27 (4), 107 (6)**
7 (1)	5 (1), 19 (3), 75 (5) …	
9 (0)	None	None
11 (2)		**15 (2), 59 (4), 235 (6)**
13 (1)	9 (1), 35 (3), 139 (5)	
15 (0)	None	None
17 (2)		**23 (2), 91 (4), 363 (6)**
19 (1)	13 (1),51 (3),203 (5)	
21 (0)	None	None
23 (2)		**31 (2), 83 (4), 331 (6)**
25 (1)	17 (1),67 (3),267 (5)	
27 (0)	None	None
29 (2)		**39 (2),155 (4),619 (6)**
31 (1)	21 (1),83 (3),331 (5)	
n mod 3 = 0	None	None
n mod 3 = 2	None	**$row_{column=0} = n+(n+1)/3$ $row_{column+1} = row_{column}*4-1$**
n mod 3 = 1	**$row_{column=0} = n-(n-1)/3$ $row_{column+1} = row_{column}*4-1$**	None

CC BY SA

Each given number of column 2 or 3 multiplied by 3 minus 1 and then divided by 2 until the number is odd, will result in the number of column 1.

We can call this table a map, which works for the opposite direction like this: for example, you start at 1 and look for any number in column 2, than you go to the respective number in the first column and there you choose the next number you wish. Then, you look for this number in column 1 and select a new number. In column 1 at this new number you select again a number. And so on and so on. If you end up in a triple, it means, you can only double the result.

You can use the map also in the original direction: search for the number you wish in column 2 or 3, select its referring number from column 1 and search again for this number in column 2 or 3. You can repeat this as long as you end up at one of the three loops.

Further we see, that every first odd number ($n \bmod 3 = 1$) has only one parent smaller than its origin (we remember, this behaviour appears with numbers $\bmod 4 = 1$). All other results are $\bmod 4 = 3$.

This map shows, that all odd numbers are unique represented.

So far, so good.

Conditions for endless growth

We know for both sequences, that even numbers do not let a sequence grow. Halving an even number shrink the sequence.

We know for the sequences of 3n+1 that all odd n mod 4 = 3 grows even, after they are halved until they reach after some iterations an odd n mod 4 = 1. The further increase depends on the structure of the binary number.

We know for the sequences of 3n-1, that all odd n mod 4 = 1 grows even, after they are halved until they reach after some iterations an odd n mod 4 = 3. The further increase depends on the structure of the binary number.

This is why we can say, that a direct growth of both sequences is impossible, because there will be always a point, after the sequence decreases. Thus, we can say, that an endless growth, if there is any, would take place in waves.

So far, so good.

Conditions for loops

The start of a loop must grow and thus, it cannot be an even number. The start must be odd and its net result must grow and it must be reachable by the end of the loop. Thus, the start cannot be a triple. We know these conditions:

For 3n+1:
$n_{Startloop}$ with n mod 4 = 3 AND n mod 3 = 1

For 3n-1:
$n_{Startloop}$ with n mod 4 = 1 AND n mod 3 = 2

The end of a loop must be larger than the start of the loop and it must fall. Thus, it can be an even number and it can be a triple. We know these conditions:

For 3n+1:
$n_{Endloop} > n_{Startloop}$
$n_{Endloop}$ with n mod 4 = 0, 2 or 1

For 3n-1:
$n_{Endloop} > n_{Startloop}$
$n_{Endloop}$ with n mod 4 = 0, 2 or 3

So far, so good.

Conclusion

Summarizing the analysis of the last pages, we can say about the Collatz sequences:

1. Any number n, which leads with the Collatz-rules to the end loop 4-2-1, multiplied by 2^x (x from 1 to ∞), will also lead to 4-2-1; the result will be even and you can either repeat this instruction or, if the result's mod 6 = 4, you can subtract 1 and divide its result by 3 for a new number (which will be odd)
2. Any odd number n, which leads with the Collatz-rules to the end loop 4-2-1, multiplied by 4 and then added 1, will also lead to 4-2-1; and because the result is again an odd number, this instruction or instruction 1 can be applied, or depending on the result's mod 3, instruction 3 or 4
3. Any odd number n with n mod 3 = 2, which leads with the Collatz-rules to the end loop 4-2-1, subtracted `(n+1)/3`, will also lead to 4-2-1; and because the result is again an odd number, instruction 1 or 2 can be applied or, depending on the result's mod 3, this instruction or instruction 4
4. Any odd number n with n mod 3 = 1, which leads with the Collatz-rules to the end loop 4-2-1, added `(n−1)/3`, will also lead to 4-2-1; and because the result is again an odd number, instruction 1 or 2 can be applied or, depending on the result's mod 3, this instruction or instruction 3

Question: Are there any numbers, which do not lead with the Collatz-rules to the end loop 4-2-1?

If you can answer this question with no, you proofed the Collatz conjecture

And we can say about the 3n-1 sequences:

1. Any number n, which leads with the 3n-1-rules to one of the end loops with 1, 5 or 17, multiplied by 2^x (x from 1 to ∞), will also lead to the respective loop; the result will be even and you can either repeat this instruction or, if the result's mod 6 = 2, you can add 1 and divide its result by 3 for a new number (which will be odd)
2. Any odd number n, which leads with the Collatz-rules to one of the end loops with 1, 5 or 17, multiplied by 4 and then subtracted 1, will also lead to the respective loop; and because the result is again an odd number, this instruction or instruction 1 can be applied, or depending on the result's mod 3, instruction 3 or 4
3. Any odd number n with n mod 3 = 2, which leads with the Collatz-rules to one of the end loops with 1, 5 or 17, added `(n+1)/3`, will also lead to the respective loop; and because the result is again an odd number, instruction 1 or 2 can be applied or, depending on the result's mod 3, this instruction or instruction 4
4. Any odd number n with n mod 3 = 1, which leads with the Collatz-rules to one of the end loops with 1, 5 or 17, subtracted `(n−1)/3`, will also lead to the respective loop; and because the result is again an odd number, instruction 1 or 2 can be applied or, depending on the result's mod 3, this instruction or instruction 3

Question: Are there any numbers, which do not lead with the 3n-1-rules to one of the end loops with 1, 5 or 17?

If you can answer this question with no, you additionally proofed the Collatz conjecture for negative natural numbers

cc BY SA

Fun fact on special numbers:

Starting with a binary number with only ones (2^x-1) leads within the sequence to a trinary number with the same count of ones. The iterations increase by 2 for each new one. Examples:

3_{10}	$= 11_2$	$= 10_3$	→ 6 iterations	→ 4_{10}	$= 100_2$	$= 11_3$
7_{10}	$= 111_2$	$= 21_3$	→ 8 iterations	→ 13_{10}	$= 1100_2$	$= 111_3$
15_{10}	$= 1111_2$	$= 120_3$	→ 10 iterations	→ 40_{10}	$= 10100_2$	$= 1111_3$
31_{10}	$= 11111_2$	$= 1011_3$	→ 12 iterations	→ 121_{10}	$= 1111001_2$	$= 11111_3$
63_{10}	$= 111111_2$	$= 2100_3$	→ 14 iterations	→ 364_{10}	$= 101101100_2$	$= 111111_3$
127_{10}	$= 1111111_2$	$= 11201_3$	→ 16 iterations	→ 1093_{10}	$= 10001000101_2$	$= 1111111_3$
255_{10}	$= 11111111_2$	$= 100110_3$	→ 18 iterations	→ 3289_{10}	$= 110011010000_2$	$= 11111111_3$

cc BY SA