PROOF OF COLLATZ' CONJECTURE

ROBERT DELOIN

Abstract. Collatz' conjecture (stated in 1937 by Collatz and also named Thwaites conjecture, or Syracuse, 3n+1 or oneness problem) can be described as follows:

Take any positive whole number N. If N is even, divide it by 2. If it is odd, multiply it by 3 and add 1. Repeat this process to the result over and over again. Collatz' conjecture is the supposition that for any positive integer N, the sequence will invariably reach the value 1.

The main contribution of this paper is to present a new approach to Collatz' conjecture. The key idea of this new approach is to clearly differentiate the role of the division by two and the role of what we will name here the jump: $a = 3n + 1$.

With this approach, the proof of the conjecture is given as well as generalizations for jumps of the form $qn + r$ and for jumps being polynomials of degree m >1.

MSC 2010 : 11Y16; 11A51
Keywords: Collatz; 3n+1; Syracuse; Thwaites; oneness; conjecture

Contents

1. Introduction

A Collatz' sequence is obtained from a start integer N to which one applies the iterative function f defined by:

- $f_0 = N$ with the integer $N \neq 0$;
- $f_{i+1} = f_i/2$ if f_i is even;
- $f_{i+1} = 3f_i + 1$ if f_i is odd.

For example, if we start with N = 7, we obtain the infinite list of numbers:
7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, ...

The sequence falls into an endless loop (1, 4, 2) but it is arbitrarily accepted that the sequence is considered done when the first number 1 is reached. This means that the symbolic test "if (f_i=1) done" is added to the function f, transforming it into an algorithm that is usually named Collatz' algorithm.

Collatz' conjecture [1][2] is now the supposition that for any positive integer N, Collatz' algorithm will always end up at 1.

Let's just notice two points:

- No information is known about the choice made by Collatz of the function $f_{i+1} = 3f_i + 1$;
- The sequence of numbers obtained for any N, also named a trajectory, is often considered in the literature as a list of undifferentiated numbers.

2. Preliminary notes

2.1. New terms: main function and jump.

In order to obtain a list of differentiated numbers and so, a new vision on Collatz' problem, we first introduce two new terms:

- *the main function*: for Collatz' algorithm, the main function is the division by two of even values of function f: $f_{i+1} = f_i/2$;
- *the jump*: for Collatz' algorithm, the jump is the special treatment $f_{i+1} = 3f_i+1$ that is used to replace odd values $f_i = n$ by an even value a usable by the main function and that we will write for convenience and from now on: $a = 3n + 1$.

2.2. The new vision: series of numbers S_i.

The new vision on Collatz' problem is now given by representing the use of the main function by commas and the use of jumps by semi-colons in Collatz' sequence. Then, Collatz' algorithm for $N = 7$ gives the list of "*series of numbers*" S_i:

7 ; 22,11 ; 34,17 ; 52,26,13 ; 40,20,10,5 ; 16,8,4,2,1

We notice that, if for $N = 7$ another algorithm is used, based by instance on the jump $a = n + 17$ instead of $a = 3n + 1$, this new algorithm falls into an endless loop as we have:

7 ; 24,12,6,3 ; 20,10,5 ; 22,11 ; 28,14,7 ;
24,12,6,3 ; 20,10,5 ; 22,11 ; 28,14,7 ;
24,12,6,3 ; 20,10,5 ; 22,11 ; 28,14,7 ;
...

where the last sequence (24,12,6,3 ; 20,10,5 ; 22,11 ; 28,14,7 ;) is looping on itself without reaching 1. It proves that the condition that detects a loop in an algorithm different or not of Collatz' one, is defined by:

Loop condition: if one odd number obtained by the main function divides the product of all the previously obtained odd numbers.

So, the question that has to be answered to prove Collatz' conjecture is:

Does Collatz' algorithm using the jump $a = 3n + 1$, always ends up at 1 whatever is the start number N and why?

To answer this question, we must first remind a general property of natural numbers and put forward three new ones.

2.3. Property 1 of natural integers N.

From the fundamental theorem of arithmetic, any natural number N can be factorized in only one way when the factorization is ordered by increasing primes, as:

$$N = 2^w p_1^{\alpha_1} p_2^{\alpha_2}\ \mathrm{p}_3^{\alpha_3}\ ...$$

where w and α_i are positive or null integers and p_i are increasing odd primes or:

Property 1: Any natural number N can be factorized as: $N = n2^w$ where $n > 0$ is an *odd* integer, composite or prime and w is a positive or null integer.

2.4. Property 2 of series of numbers S_i.

Property 2: For any given natural number N, the series of numbers S_i are parts of invariant branches B_i of general form: $B_i = B(n, w) = n2^w$ with the odd integer $n > 0$ and the natural integer $w \geqslant 0$

Proof. Let's build the following table filled bottom to top by an odd n in column 0 and adding numbers left to right by recurrently multiplying these ones by 2.

Table 1. S_i are parts of branches $B(n, w) = n2^w$

for odd $n > 0$ and $w \geqslant 0$

Br\Cols:	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	...
...	...	...	...	...	...	...	...	...	...
$B(15,w)$:	15	30	60	120	240	480	960	1920	...
$B(13,w)$:	13	26	52	104	208	416	832	1664	...
$B(11,w)$:	11	22	44	88	176	352	704	1408	...
$B(9,w)$:	9	18	36	72	144	288	576	1152	...
$B(7,w)$:	7	14	28	56	112	224	448	896	...
$B(5,w)$:	5	10	20	40	80	160	320	640	...
$B(3,w)$:	3	6	12	24	48	96	192	384	...
$B(1,w)$:	1	2	4	8	16	32	64	128	...
$B(n,w) \uparrow$	$B(n,0) = n \uparrow$	$w \to 1$	2	3	4	5	6	7	...

Reading from right to left, each line is a list of numbers that are divided by 2 until they reach an odd number: this is exactly the first part of the definition of the main function of the algorithm. □

2.5. Property 3 on branches B(n,w).

Property 3: The infinite set of branches $B(n, w)$ is
a covering system of the natural number set $\mathbb{N}$
or:
Any positive integer (even or odd) is present in Table 1.

Proof. This is because all branches $B(n, w)$ are of the form $B(n, w) = n2^w$ where n is odd, which is exactly the general definition of natural numbers according to the fundamental theorem of arithmetic. □

In Table 1, property 3 is true only for numbers up to 16 as odd numbers are limited to 15, but it suffices to expand the table upwards to any odd number $2^w - 1$ to complete the list up to any number 2^w.

For $N = 7$, the result given by Collatz' algorithm can then be represented as follows, with a last column indicating the part of branch used by each series of numbers:

Table 2. Collatz' trajectory of $N = 7$
using parts of branches $B(n, w = list)$

a_i	, ... ,	n_i	branch
$a_1 = N = 7$	, ... ,	$n_1 = 7$;	$B_1 = B(7, w = 0)$
$a_2 = 3n_1 + 1 = 22$	, ... ,	$n_2 = 11$;	$B_2 = B(11, w = 1, 0)$
$a_3 = 3n_2 + 1 = 34$	, ... ,	$n_3 = 17$;	$B_3 = B(17, w = 1, 0)$
$a_4 = 3n_3 + 1 = 52$	, ... ,	$n_4 = 13$;	$B_4 = B(13, w = 2, 1, 0)$
$a_5 = 3n_4 + 1 = 40$	, ... ,	$n_5 = 5$;	$B_5 = B(5, w = 3, 2, 1, 0)$
$a_6 = 3n_5 + 1 = 16$	, ... ,	$n_6 = 1$	$B_6 = B(1, w = 4, 3, 2, 1, 0)$

The trajectory for $N = 7$ can thus be summed up by the list of parts of branches:

$$B(7, w = 0)\ ;\ B(11, w = 1, 0)\ ;\ B(17, w = 1, 0)\ ;$$
$$B(13, w = 2, 1, 0)\ ;\ B(5, w = 3, 2, 1, 0)\ ;\ B(1, w = 4, 3, 2, 1, 0)$$

where we notice that, for a given N, the n_i's of the branches $B(n_i, w)$ are somewhat erratic.

2.6. **Property 4: Condition to reach 1.** From the example for $N = 7$ and *if the series S_i is supposed to be the last one*, we deduce that it is always the last jump from n_{i-1} to $a_i = 3n_{i-1} + 1$ that leads to $n_i = 1$. Therefore:

Property 4: Collatz' algorithm ends up at 1
if there exists a triplet solution (i, n_{i-1}, m) to the equation:

$$3n_{i-1} + 1 = 2^m$$

2.7. **Solutions n(m) to reach 1.** To solve the last equation, we can consider that either n_{i-1} (or n_i) is a function of m or the converse. We can look for solutions $m(n_i)$ but as n_i can be known only by running the algorithm to its end, this is not a mathematical solution. We have then to look for solutions $n_i(m)$. As the first a_i of each branch (except the first when N is odd) is of the general form:

$$a_i = 3n_{i-1} + 1$$

we have first to check if the equation:

$$3n + 1 = 2^m \tag{1}$$

where n and m are independent of i (and thus of N), has always at least one solution $n(m)$ or not. From equation (1), the solutions $n(m)$ always verify:

$$n(m) = \frac{2^m - 1}{3}$$

In the two following subsections we will prove that they can be either integer or fractional (but, in the last case, invisible because Collatz' algorithm works only with integer n's).

2.7.1. *The integer solution n(m).* To study this integer solution is equivalent to study the factorization of the numbers $2^m - 1$ as equation (1) implies that:

$$2^m - 1 = 3n$$

We know that when m is even ($m = 2k$), we algebraically have:

$$2^m - 1 = 2^{2k} - 1 = 4^k - 1^k$$
$$4^k - 1^k = (4 - 1)(4^{k-1} + 4^{k-2} + 4^{k-3} + ... + 4 + 1)$$

so that:

$$2^{2k} - 1 = 3(4^{k-1} + 4^{k-2} + 4^{k-3} + ... + 16 + 4 + 1)$$
$$2^{2k} = 3(4^{k-1} + 4^{k-2} + 4^{k-3} + ... + 16 + 4 + 1) + 1$$

Therefore, whatever is N, the integer solution $n(m)$ of the equation $3n+1 = 2^m$ when $m = 2k$ is:

$$m = 2k$$
$$n(m) = 4^{k-1} + 4^{k-2} + 4^{k-3} + ... + 16 + 4 + 1$$

2.7.2. *The fractional solution n(m).* We know that when m is odd ($m = 2k + 1$):

$$2^m - 1 = 2^{2k+1} + (-1)^{2k+1}$$
$$2^{2k+1} + (-1)^{2k+1} = (2 + (-1))(2^{2k} + 2^{2k-1} + ... + (4 + 2) + 1)$$
but as:
$$2^{2k} + 2^{2k-1} = (2 + 1)2^{2k-1} = 3 \times 2^{2k-1}$$
we have:
$$2^{2k+1} - 1 = 3(2^{2k-1} + 2^{2k-3} + ... + 2^3 + 2^1) + 1$$
$$2^{2k+1} = (3n + 1) + 1 = 3n + 2$$
which can be written:
$$2^{2k+1} = 3(n + 1/3) + 1$$

This shows that when $m = 2k + 1$ is odd, the equation $2^m = 3n + 1$ has no integer solution $n(m)$ but always a fractional one that verifies:

$$n(m) = \frac{2^m - 1}{3}$$
$$n(m) = (2^{2k-1} + 2^{2k-3} + ... + 2^3 + 2^1) + 1/3$$
which always gives:
$$3n + 1 = 3(2^{2k-1} + 2^{2k-3} + ... + 2^3 + 2^1) + 2 = 3(n - 1/3) + 2 = 2^{2k+1}$$

We have therefore:
Property 5: Independently of N and for any $m \geqslant 0$, the general equation $3n + 1 = 2^m$ has always a solution for n. This solution is:

either the integer solution of $3n + 1 = 2^m$ for any even m
or, for any odd m:
the fractional solution of $3n + 1 = 2^m$
or the integer solution of $3n + 2 = 2^m$

Remark. The fractional solution happens by instance for $N = 7$ as Collatz' algorithm ends up at f=1 for $n = 682 + 1/3$ because:

$$3(682 + 1/3) + 1 = 3 \times 682 + 2 = 2048 = 2^{11}$$

Thus, after $m = 11$ divisions (or commas), the full trajectory for $N = 7$ being:

7 ; 22,11 ; 34,17 ; 52,26,13 ; 40,20,10,5 ; 16,8,4,2,1

we get the following Table by retaining only the values f^* that follow a division or comma:

Table 3. Details for $N = 7$

variables	val	val	val	val	val	val	val	val	val	val	val
m	1	2	3	4	5	6	7	8	9	10	11
f^*	11	17	26	13	20	10	5	8	4	2	1
2^m	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
$n(m)$	1/3	1	7/3	5	31/7	21	...	...	...	...	$(2^{11} - 1)/3$

This shows that jumps (semi-colons) have not to be taken into account in the calculation of m. This can be illustrated by placing each division in a column and the jumps a_i *under* the last odd n_{i-1}. This creates a 2-dimensional table for the trajectory where each branch is isolated in a line, as follows:

Table 4. Trajectory of $N = 7$ in 11 divisions (commas)

Br\\m	0	1	2	3	4	5	6	7	8	9	10	11
1	7											
2	22,	11										
3	(68)	34,	17									
4	(208)	(104)	52,	26,	13							
5	(640)	(320)	(160)	(80)	40,	20,	10,	5				
6	(2^{11})	(2^{10})	(2^9)	(2^8)	(2^7)	(2^6)	(2^5)	16,	8,	4,	2,	1.

The numbers in parenthesis are not part of the trajectory but show the prolongation of each branch on the left. The observation of this table gives three other properties:

Property 6. All n_i's of the trajectory of $N = 7$ have a trajectory that ends up at 1.

Property 7. Each $a_i = 2^{\alpha_i} n_i$.

Property 8. The number B of branches is equal to the number of lines in Table 4.

We then have:

$$B = 1 + J \text{ where } J \text{ is the number of jumps}$$

Now, let's consider the product of the first a_i of each branch i (let's notice that for $N = 7$, in the first line of Table 4, 7 is both an a_i and an n_i):

$$A = \prod_{i=1}^{B} a_i = \prod_{i=1}^{B} 2^{\alpha_i} n_i = 2^{\sum_{i=1}^{B} \alpha_i} \prod_{i=1}^{B} n_i$$

As the number of divisions m to go from N to 1 by Collatz' algorithm is the same as the number of multiplications by 2 to go back from 1 to N, we have:

$$m = \sum_{i=1}^{B} \alpha_i \tag{2}$$

so that:

$$\prod_{i=1}^{B} a_i = 2^m \prod_{i=1}^{B} n_i$$

and:

$$m = log_2 \frac{\prod_{i=1}^{B} a_i}{\prod_{i=1}^{B} n_i} \tag{3}$$

Verification for our case:

$$m = log_2 \frac{7.22.34.52.40.16}{7.11.17.13.5} = log_2 2^{11} = 11$$

But due to the erratic values n_i ending the successive branches, we are still not sure that one of these erratic values will verify equation (1). The next section examines this problem.

2.8. **Capability of the algorithm to reach 1.** We have seen with Property 3 that the branches $B(n,w) = n2^w$ with odd integers n are a covering system of the set $\mathbb{N}$ of natural numbers. But we have also seen that *the sequence of branches* used by a trajectory is somewhat erratic, so that it cannot be mathematically expressed.

Fortunately, there is another set of mathematical objects, different from the set of branches $B(n,w)$, that give another way to cover the set $\mathbb{N}$ of natural numbers and that can be mathematically expressed.

2.8.1. *Cut-out of* $\mathbb{N}$ *by numbers* 2^m. If we cut out the set $\mathbb{N}$ of the natural integers using the successive powers of 2, we can write the whole set in 2^m-type columns as follows:

Table 5. Cut out of $\mathbb{N}$ by powers of 2

.	.	.	.	.	.	.	$2^{m+1}-1$
...	...	...	...	...	...	...	.
.	.	.	15	23	39	...	.
.	.	.	14	22	38	...	.
.	.	.	13	21	37	...	.
.	.	.	12	20	36	...	.
.	.	7	11	19	35	...	.
.	.	6	10	18	34	...	.
.	3	5	9	17	33	...	2^m+1
1	2	4	8	16	32	...	2^m
0	1	2	3	4	5	...	m

where each column $m \geqslant 0$ begins at 2^m, ends at $(2^{m+1}-1)$ and contains:

$$(2^{m+1}-1) - (2^m - 1) = 2^m \text{ numbers.}$$

2.8.2. *Right shifts implied by jumps.* With this cut out of $\mathbb{N}$, we can see that:

• the second term $(n = 2^m + 1)$ in column m is always transferred by the jump $a = 3n + 1$ into column $m + 1$ because for $n = 2^m + 1$, we always have:

$$a = 3n + 1 = 3(2^m + 1) + 1 = (2 + 1)(2^m) + 4 = 2^{m+1} + 2^m + 2^2$$

which proves that this number a is in column $m + 1$.

• the upper term $(n = 2^{m+1} - 1)$ of a column $m \geqslant 0$ is always transferred by $a = 3n + 1$ into column $m + 2$ because for $n = 2^{m+1} - 1$, we always have:

$$a = 3n + 1 = 3(2^{m+1} - 1) + 1 = (2 + 1)(2^{m+1}) - 2 = 2^{m+2} + 2^{m+1} - 2$$

which proves that this number a is in column $m + 2$.

These two points prove that the first (even) number a_i of a series $i > 1$, produced by a jump $a = 3n + 1$, is always obtained by *an always existing right shift of 1 or 2* 2^m*-type columns* in $\mathbb{N}$.

2.8.3. *Left shifts implied by the main function.* All the even terms t of a 2^m-type column with $m > 0$ can always be written as:

$$t = 2^m + 2s \quad \text{with } 0 \leqslant s \leqslant 2^{m-1} - 1$$

so that:

$$t/2 = 2^{m-1} + s \quad \text{with } 0 \leqslant s \leqslant 2^{m-1} - 1$$

This means that a division by two of an even number in column m always places the result in column $m-1$, producing *an always existing left shift of 1 column* in $\mathbb{N}$.

2.8.4. *Property 9 of Collatz' algorithm.* For Collatz' algorithm, we have seen that:
• each jump $a = 3n+1$ between two branches corresponds to an *always existing right shift* of 1 or 2 columns in $\mathbb{N}$;
• each division by two corresponds to an *always existing left shift* of 1 column in $\mathbb{N}$;
These two points prove that the right and left shifts of 1 column are always possible and we have:

Property 9a. Collatz' algorithm provides *a continuous screening of the 2^m-type columns of* $\mathbb{N}$, these columns being a covering system of $\mathbb{N}$.

or:

Property 9b. No 2^m-type column of $\mathbb{N}$ is left unreachable by Collatz' algorithm,
particularly column C0 and its number 1.

This property proves the capability of Collatz' algorithm to end up at 1.

3. Main Result: Proof of Collatz' conjecture

Now, we have all the necessary properties to prove Collatz' conjecture:

Proof. • From property 2 we know that Collatz' algorithm (CA) is an erratic screening of branches $B_i = B(n, w) = n2^w$. This can be symbolically written:

$$CA = ErrScr(B_i) \tag{4}$$

• From property 3 we know that the branches $B_i = B(n, w)$ are a covering system of $\mathbb{N}$. This can be symbolically written:

$$B_i = CovSys(\mathbb{N})$$

Then, with this result, equation (4) becomes:

$$CA = ErrScr(CovSys(\mathbb{N})) \tag{5}$$

• From property 9a we know that the jumps of Collatz' algorithm are equivalent to a *continuous* screening of the 2^m-type columns of $\mathbb{N}$. Then, equation (5) can be symbolically written:

$$CA = ContScr(2^m columns(\mathbb{N}))$$

We know at this point, from property 9b, that Collatz' algorithm has the capability to end up at 1. But does it always do it? The answer comes from the three last facts:
1− Jumps *always* provide the even numbers necessary to the main function, which ensures the continuity of the algorithm from branch to branch;
2− The main function (division by 2) is *always* a left shift of one 2^m-type column, that is to say a move towards the goal of the algorithm constituted by column C0 where the number 1 is located.
3− From properties 4 and 5 we know that for any m and independently of Collatz' algorithm, the general equation $3n+1 = 2^m$ has *always* a solution for n. We also know from equation (3) that, in Collatz' algorithm, m depends on $i = B$, the number of branches B_i necessary to the trajectory. Therefore, an $i = B$ value *always* exists such that for any given number N, the equation $3n_{j,i-1}+1 = 2^m$ has a solution in $n_{j,i-1}$ when the $n_{j,i}$ are computed by the algorithm or by incrementing one by one i or m, which proves Collatz' conjecture. □

4. A generalization for even jumps

A more general approach on Collatz' problem is obtained by keeping the division by 2 as main function but by considering the general even jump $a = qn + r$ where q and r verify $\gcd(q, r) = 1$.

As in Collatz' algorithm a jump is used only when n is odd, we choose to have only odd n's. As this makes a to be even, this implies that q and r have to be of same parity. For simplicity, we will use hereafter only odd q's and odd r's with $\gcd(q, r) = 1$.

We will now look for the conditions that odd q's and r's have to verify to make the general algorithm end up at 1 and show that Collatz' algorithm verifies them. This almost mimics what has been done for the jumps $a = 3n+1$ but it enables us to prove the uniqueness of Collatz' algorithm and other results.

4.1. **Condition 1 to end up at 1.** To reach the branch $B(1, w)$ from a given N and so end up at 1, we know from property 4 that for a given N and at the end of the branch $B_{i-1}(n_{i-1}, w)$, the general algorithm has to verify the condition:

$$qn_{i-1} + r = 2^m$$

To solve this condition is equivalent to study the factorization of $2^m - r$ as, ignoring the index of n_{i-1}, we must have:

Condition 1: $2^m - r = qn$

This shows that the condition that makes the general algorithm reach the branch $B(1, w)$ and end up at 1 for a given N is that q must be a divisor of $2^m - r$, which then implies that $n = (2^m - r)/q$ is an integer. It appears that only two cases have to be differentiated.

4.1.1. *Case where $q = 1$ with any odd $r > 0$.* With $q = 1$, $a = qn + r$ can be written $a = n + r$ and, with odd n and r, condition 1 can be written:

$$2^m - r = n$$

We see now that the problem of the factorization of $2^m - r$ is transferred from its factorization qn to the factorization of n only. As, according to the fundamental theorem of arithmetic, any positive *odd* number n can be written:

$$n = p_1^{\alpha_1} p_2^{\alpha_2} p_3^{\alpha_3} \ \dots \ \text{all } p_i\text{'s being odd}$$

we must have:

$$2^m - r = p_1^{\alpha_1} p_2^{\alpha_2} p_3^{\alpha_3} \ \dots \ \text{all } p_i\text{'s being odd}$$

This makes appear the result that:

When the trajectory from N to n is possible and j is the number of divisors d_j of n, for each divisor d_j of $2^m - r$, condition 1 is verified and j couples (m_j, r_j) determine j couples (q_j, n_j), or j couples $(q_j, r_j = 1)$ determine j couples (m_j, n_j). This is because for j values, we have:

$$2^{m_j} - r_j = d_j(n/d_j) = n$$

Then, the last two series always appear as:

$$2^k n,\ \dots,\ 4n,\ 2n,\ n\ ;\ n + r = 2^m,\ 2^{m-1},\ \dots,\ 4,\ 2,\ 1$$

and the j algorithms based on j couples $(q_j, r_j = 1)$ always end up at 1.

4.1.2. *Case with odd q>1 and odd r>0.* This case is identical to the first case when we change n into qn.

This makes appear the result that:

When the trajectory from N to n is possible and j is the number of divisors d_j of qn, for each divisor d_j of $2^m - r$, condition 1 is verified and j couples (m_j, r_j) determine j couples (q_j, n_j), or j couples (m_j, n_j) determine j couples (q_j, r_j). This is because for j values, we always have:

$$2^{m_j} - r_j = d_j(qn/d_j) = qn$$

Again, the last two series always appear as:

$$2^k\ n,\ \dots,\ 4n,\ 2n,\ n\ ;\ qn + r{=}2^m,\ 2^{m-1},\ \dots,\ 4,\ 2,\ 1$$

and the j algorithms based on j couples $(q_j, r_j = 1)$ always end up at 1.

As this case includes the couple $(q = 3, r = 1)$, it includes Collatz' algorithm and we have the result that:

For a given N, if the trajectory from N to n is possible, condition 1 is verified for j couples (m_j, r_j) that determine j couples (q_j, n_j), or j couples (q_j, r_j) whatever is m, and always among them the couple $(q_1 = 3, r_1 = 1)$, as Collatz' conjecture has been proved.

4.2. **Condition 2 on the trajectory from N to n$_i$.** The two last results are still conditional to the fact that:

Condition 2: the trajectory from N to $n = n_i$ has to be possible.

Proof. This condition is always verified by Collatz' algorithm because:
• independently of N, the main function (the division by 2 applied only to even numbers) and the jumps $a = qn + r$, are always defined functions;
• and because Collatz' algorithm verifies Properties 9a and 9b. □

The last remaining point is to generalize Condition 1 from one given N to all N's, which will give the final result.

4.3. **Uniqueness of Collatz' algorithm.**

Proof. We know from 4.1.2 that for a given N, condition 1 is verified for j couples (q_j, r_j) that always include $(q = 3, r = 1)$ which defines Collatz' jump.

But for different N's, the *number of couples* j is generally different from one N to another. This is because the number of divisors j that divide qn is generally different from one n to another.

The *involved couples* in the lists of couples are also generally different from one list to another.

As we have seen in 4.1.2 that the couple $(q = 3, r = 1)$ is always present in these lists, independently of N, it proves that for all N's, *the unique couple* (q, r) *with odd* r *common to all lists that make a general algorithm end up at 1*, is the couple $(q = 3, r = 1)$ which defines Collatz' jump and algorithm. □

Examples:

Table 6. Different jumps $a = qn + 1$ that make the algorithm end up at 1 for different N's with q checked up to 199

N	jumps	nb divs $= m$	jump #
N=7	$a = 3n + 1$	11	1
N=7	$a = 9n + 1$	6	2
N=7	$a = 17n + 1$	11	3
N=7	$a = 73n + 1$	9	4
...	...	...	...
N=11	$a = 3n + 1$	10	1
N=11	$a = 3.31n + 1$	10	2
...	...	...	...
N=24	$a = 3n + 1$	8	1
N=24	$a = 5n + 1$	7	2
N=24	$a = 9n + 1$	11	3
N=24	$a = 3.7n + 1$	9	4
N=24	$a = 5.17n + 1$	11	5
...	...	...	...
N=1000	$a = 3n + 1$	72	1
...	...	...	...

5. Other results for general even jumps

5.1. **A fast check of a general even jump a = qn+r.** A fast method to check if an algorithm using $a = qn + r$ ends up at 1 is as follows:

- 1- Factorize $2^m - 1$ for all m's up to any wanted limit;
- 2- All factors appearing in these factorizations are possible q's but the only true solutions are those for which no loop happens for a given N.
  If q appears in the factorizations generated by $2^m - 1$, the given algorithm will potentially end up at 1.

Example: Check of Collatz' algorithm where r=1. For $2^m \leq 1000$, we have:

Table 7. Check of Collatz' algorithm

m	$2^m - 1$	factorization
9	512-1=511	7.73
8	256-1=255	3.5.17
7	128-1=127	127
6	64-1=63	3^2.7
5	32-1=31	31
4	16-1=15	3.5
3	8-1=7	7
2	4-1=3	3
1	2-1=1	

As $q = 3$ appears in the factorizations generated by $2^m - 1$ for any m, Collatz' algorithm potentially ends up at 1.
Table 7 also confirms the results of Table 6 as, by instance for $N = 7$, the incomplete list of jumps making Collatz' algorithm end up at 1 are obtained with $q = 3, 9, 17, 73$ which are values of Table 7.

5.2. **The fastest algorithm based on divisions by 2.** On one hand, when $a = qn + r > n$, the jump is a "rear jump" with respect to 1 as the distance from a to 1 is greater than that of n to 1.

On the other hand, when $a = qn + r < n$, the jump is a "front jump" towards 1. Therefore, with $q = 1$, a front jump $a = n + r$ is obtained if and only if $r < 0$.

As in this case we have $q = 1$ and $a = n + r$ with $n = (2^m - r)/q$ depending on m, for some small values of m (the column in $\mathbb{N}$ where n is located) it may happen, if r is too much negative, that $a = n + r$ becomes a big front jump that skips one or several 2^m-type columns of $\mathbb{N}$, leaving them unreachable and making the algorithm a *not continuous screening* of the columns.

As the smallest odd n_i that does not stop the algorithm is 3, it thus appears that the only acceptable negative odd value of r that makes the jump $a = n+r$ to be an acceptable front jump, is $r = -1$. It gives the exceptional jump:

$$a = n - 1$$

the unique and fastest algorithm that contains only front jumps and so, the fastest decreasing sequence towards 1.

For $N = 1000$, this jump $a = n - 1$ gives:
1000,500,250,125 ; 124,62,31 ; 30,15 ; 14,7 ; 6,3 ; 2,1
with only 9 divisions, much less than the 72 divisions necessary for Collatz' algorithm with jump $3n + 1$ as mentioned in Table 6. As a comparison:
• the jump $a = n + 1$ gives:
1000,500,250,125 ; 126,63 ; 64,32,16,8,4,2,1 with 10 divisions,
• the jump $a = n + 3$ gives:
1000,500,250,125 ; 128,64,32,16,8,4,2,1 with 10 divisions,
• the jumps $a = n + 5$ and $a = n + 7$ give loops on 5,
• the jump $a = n + 9$ gives:
1000,500,250,125 ; 134,67 ; 76,38,19 ; 28,14,7 ; 16,8,4,2,1 with 12 divisions,
• the jump $a = n + 11$ gives:
1000,500,250,125 ; 136,68,34,17 ; 28,14,7 ; 16,8,4,2,1 with 12 divisions,
• the jump $a = n + 13$ gives:
1000,500,250,125 ; 138,69 ; 82,41 ; 54,27 ; 40,20,10,5 ; 18,9 ;
22,11 ; 24,12,6,3 ; 16,8,4,2,1 with 18 divisions.

6. A generalization for jumps being polynomials of degree m >1

As we have seen that Collatz' algorithm is made of an integer main function f such that $f_{i+1} = f_i/2$ and an integer jump function $a_i = 3f_i + 1$ used to replace odd f_i values, a full generalization would have to take into account any combination of any two functions.

Here, we will only consider main functions f that are divisions by any integer polynomial g_i:

$$f_{i+1} = f_i/g_i$$

and jumps are integer polynomials:

$$a_{i+1} = a(f_i)$$

used to replace f_{i+1} when this value is less than 1. To prove the method in a simple way, we will do it first on an instance where g_i and a_{i+1} are known.

6.1. A first step.

In a first step, let's choose the divisor function:

$$g_i = i^2 + 1$$

where i is an integer (not a complex number). If we choose that this algorithm ends up at 1 when $m = 4$, we consider the four first values of g_i: $g_{1,4} = \{2, 5, 10, 17\}$ whose product is 1700. Let's generate the sequence with $f_0 = N = 1700$. We get the sequence with no jumps:

$$f_0 = N, f_1 = f_0/g_1, f_2 = f_1/g_2, f_3 = f_2/g_3, f_4 = f_3/g_4$$

which gives:

$$f_0 = 1700, f_1 = 1700/2 = 850, f_2 = 850/5 = 170,$$
$$f_3 = 170/10 = 17, f_4 = 17/17 = 1$$

and we get that the sequence ends up at 1 with f_4 as expected.

This proves that there always exists an algorithm beginning with any number N and ending at 1 when the divisor function $g(i)$ is an integer polynomial that generates the exact list of the factors g_i of N.

6.2. A possible second step.

A possible second step can be to find which jumps can be associated with f that can allow to start the sequence with an f_0 different from $N = 1700$.

To do that, we have to choose a value of i that makes the jump a_{i+1} replace a disqualified $f_{i+1} = f_i/g_{i+1} < 1$ coming from an integer N' different of N. By instance, let's choose $i = 2$ such that $a_2 = f_2 = 170$ replaces a disqualified value $f_2 = f_1/g_2 < 1$. Here, the divisor function $g_i = i^2 + 1$ is already defined but not the jump. Let's choose by instance the jump:

$$a_{i+1} = (f_i)^2 + b$$

which fixes b to the odd complements to $f_2 = 170$ of these squares:

$$b = 170 - (f_i)^2$$

All the possible jumps are then:

$$\begin{gathered}
a_{i+1} = (f_i)^2 + 161 \text{ for } f_i = 3\\
a_{i+1} = (f_i)^2 + 145 \text{ for } f_i = 5\\
a_{i+1} = (f_i)^2 + 121 \text{ for } f_i = 7\\
a_{i+1} = (f_i)^2 + 89 \text{ for } f_i = 9\\
a_{i+1} = (f_i)^2 + 49 \text{ for } f_i = 11\\
a_{i+1} = (f_i)^2 + 1 \text{ for } f_i = 13
\end{gathered}$$

Choosing $f_i = f_1 = 3$, we have $b = 161$ and the sequence is:

$$\begin{gathered}
f_0 = 2f_1 = 6, f_1 = 3, f_2 = 3/5 < 1 \text{ replaced by } a_2 = 3^2 + 161 = 170,\\
f_3 = 170/10 = 17, f_4 = 17/17 = 1
\end{gathered}$$

For all the possible odd values of b above and their associated values f_i, the sequences are:

$$\begin{array}{cccc}
\text{for } f_i = 3, & b = 161: & a_{i+1} = (f_i)^2 + 161 & f_i = 6,\ 3;\ 170,\ 17,\ 1\\
\text{for } f_i = 5, & b = 145: & a_{i+1} = (f_i)^2 + 145 & f_i = 10,\ 5;\ 170,\ 17,\ 1\\
\text{for } f_i = 7, & b = 121: & a_{i+1} = (f_i)^2 + 121 & f_i = 14,\ 7;\ 170,\ 17,\ 1\\
\text{for } f_i = 9, & b = 89: & a_{i+1} = (f_i)^2 + 89 & f_i = 18,\ 9;\ 170,\ 17,\ 1\\
\text{for } f_i = 11, & b = 49: & a_{i+1} = (f_i)^2 + 49 & f_i = 22,\ 11;\ 170,\ 17,\ 1\\
\text{for } f_i = 13, & b = 1: & a_{i+1} = (f_i)^2 + 1 & f_i = 26,\ 13;\ 170,\ 17,\ 1
\end{array}$$

This proves that for all the odd values 3 to 13 and all the even values $\{3.\text{to}.13\}2^1$, all different of $N = 1700$, the algorithm defined by:

$$f_{i+1} = f_i/g_{i+1}, \quad g_i = i^2 + 1 \quad \text{and } a_i = i^2 + b$$

ends up at f=1.

6.3. Proof of the generalization.

Proof. The above proof has been built upon chosen instances of N, $g(i)$ and $a(i)$. It does not allow, at this stage, to generalize to all combinations of integer polynomials $f(i)$, $g(i)$ and $a(i)$.

But, as according to the fundamental theorem of arithmetic, any integer number $f(i)$ generated by an integer function f can be factorized in only one way when the factorization is ordered by increasing primes, it is also true for any number:

$$N = \prod_{i=1}^{m} f(i)$$

So, as it is always possible by a system of m equations to find a rational polynomial function $g(i)$ that generates the list of divisors of N, it is always possible to find an algorithm ending up at 1 for any value of m and N, which proves the generalization. $\square$

Acknowledgements. This work is dedicated to my family. As I am a hobbyist, I wish to express my gratitude towards the Editors of this journal as well as towards the team of Reviewers for having welcomed, reviewed and accepted my article.

References

[1] Weisstein, Eric W., Collatz Problem, *From MathWorld–A Wolfram Web Resource* at: http://mathworld.wolfram.com/CollatzProblem.html

[2] Collatz conjecture, *From Wikipedia, the free encyclopedia* at: https://en.wikipedia.org/wiki/Collatz_conjecture

Robert Deloin, Bouc Bel Air, France
E-mail address: `rdeloin@free.fr`