Collatz Conjecture: Properties and Algorithms

Evangelos Tzanis

Computation and Reasoning Laboratory (Corelab)
School of Electrical and Computer Engineering
National Technical University of Athens
15780 Zografou, Athens, Greece
Email: vtzanis@cs.ntua.gr

Abstract. The mathematical problem on which this paper focuses is a chestnut from the 1930's known as: the $3x + 1$ problem, or the Collatz problem, or Ulam's problem; various other names exist. The eminent mathematician Paul Erdös suggested: "mathematics is not ready for this kind of problems" [1]. Here we try to approach the problem by classifying natural numbers in order to investigate which numbers reach a smaller number with certainty. We associate each such class with a binary tree and we show a relation of these trees to the Pascal's Triangle. We present two algorithms: The first generates the corresponding binary tree for each class, after a reasonably small number of steps. The second algorithm provides an efficient way for an overwhelming majority of input numbers, to estimate the total number of steps needed to terminate. This paper combines mathematical methods and programming techniques. For our current knowledge the Collatz problem is computationally hard. But for most of the numbers it is mathematically certain that they reach a smaller number quite fast. Our contribution is a method to distinguish the majority of these numbers. Practically speaking, this means that for most input numbers we can generate the Collatz sequence in reasonable time, while for few numbers we have to execute the iteration process exhaustively.

1 Introduction

One of the most tantalizing conjectures in number theory is the so-called $3x + 1$ conjecture, attributed to L. Collatz (1937) [10]. The problem can be stated quite simply. Start with any positive integer. If it is an even number, halve it.[1] Otherwise, multiply the number by 3, add 1 to it, and then halve it (e.g., if the starting number was 11, the next integer in the sequence would be 17). Take the result and repeat the process.

Any such sequence seems to end up in the cycle 2, 1, 2, 1, etc., no matter the choice of starting number. By convention, one terminates the iterations at 1. More formally:

Let T: $\mathbb{Z} \to \mathbb{Z}$ be defined by:

$$T(x) = \begin{cases} \frac{x}{2}, & \text{if x is even} \\ \frac{3x+1}{2}, & \text{if x is odd} \end{cases}$$

[1] The original function described by L. Collatz used $(3x + 1)$ for the transformation of odd numbers. Without incurring any loss of generality, it is convenient to replace the odd transformation by $\frac{3x+1}{2}$.

Collatz conjectured that if $x \in \mathbb{N}$, then the trajectory: $x, T(x), T^2(x), \ldots$, eventually reaches the cycle 1, 2, 1.

Consider the application of $T(x)$ to the starting number $x_0 = 17$. Since 17 is odd we apply the $\frac{3n+1}{2}$ component and find that the next iterate $x_1 = 26$. Since 26 is even we apply the $\frac{n}{2}$ component again and find $x_2 = 13$. The next iteration produces $x_3 = 20$, which is even. Division by 2 yields $x_4 = 10$, which is even and so on until the value 1 is reached; at which point we terminate application of the $T(x)$ function. The complete sequence is shown in below table.

The $3x + 1$ sequence produced by $x_0 = 17$

Step	0	1	2	3	4	5	6	7	8	9
Value	17	26	13	20	10	5	8	4	2	1

The sequence has no obvious pattern. Moreover, we do not have an easy explanation why the sequence should take 9 iterations to reach 1 and why the maximum value taken by any iterate is 26.

The wildly varying nature of the iterates is displayed more precisely in [1], [2] and [3]. For instance, an extreme case is when the starting value is $x_0 = 27$ and 70 iterations are needed to reach 1. In general, the number of iterations is not proportional to the magnitude of the starting number.

2 Mathematical Approach

Several remarkable features are noteworthy about the behavior of the iterative function:

- Being easy to program on a computer [4, 5], it has been shown to hold for all starting integers up to 10^{13}.
- The length of any sequence bears no simple relationship to the magnitude of the starting number.
- The values in any sequence appear to be completely erratic and unpredictable.

In this paper we will try to present briefly one method which gives a better idea how the Collatz function behaves. Our aim is to give a proof of the following statement:

The Collatz conjecture is true for most of numbers, without using a program to evaluate the number of iterations needed for a given number to terminate (i.e. without computing the Collatz function).

2.1 Classification

We proceed by classifying all numbers in $\mathbb{N}$ into four different classes A, B, C and D.

$$A = \{x | x \equiv\ 1\ (mod\ 4)\},\ B = \{x | x \equiv\ 2\ (mod\ 4)\},$$

$$C = \{x | x \equiv\ 3\ (mod\ 4)\}, D = \{x | x \equiv\ 0\ (mod\ 4)\}$$

A-class	B-class	C-class	D-class
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

$$A : 1 + 4 \cdot k,\ B : 2 + 4 \cdot k,\ C : 3 + 4 \cdot k,\ D : 4 + 4 \cdot k,\ (k = 0, 1, 2, \ldots)$$

2.2 Yielding a Smaller Number

By this classification we will try to show that almost all numbers reach 1. One trivial fact is that even numbers yield a smaller number, since we divide them with 2. So, if we give a proof that any odd number in classes A and C can reach a smaller number then immediately we conclude that any number can reach the number 1. But this seems to be the hard part of our method.

Proposition 1 *All the numbers in class A reach a smaller number using precisely two steps of Collatz function.*

Consider the application of T(x) to the starting number $x_0 = 1 + 4 \cdot k \in A$. Since x_0 is odd we apply the $\frac{3x+1}{2}$ component and find that the next iterate $x_1 = 2 + 6 \cdot k$. Since x_1 is even we apply the $\frac{x}{2}$ component and find $x_2 = 1 + 3 \cdot k$ and so on:

The $3x + 1$ sequence produced by $x_0 = 1 + 4 \cdot k$

Step	0	1	2
Value	1+4·k	2+6·k	1+3·k

But $1 + 4 \cdot k > 1 + 3 \cdot k$. So the numbers of A class reach a smaller number using two Collatz function steps.

2.3 New Classification

We proceed by classifying all numbers in C class into four different classes C_1, C_2, C_3 and C_4.

$$C_1\!:\ 3 + 16 \cdot k\ ,\ C_2 : 7 + 16 \cdot k,\ \ C_3\!:\ 11 + 16 \cdot k,\ \ C_4\!:\ 15 + 16 \cdot k \quad (k = 0, 1, 2, \ldots)$$

2.4 Yielding a Smaller Number (continue)

Proposition 2 *All the numbers in class C_1 reach a smaller number using precisely four steps of Collatz function.*

Consider the application of $T(x)$ to the starting number $x_0 = 3 + 16 \cdot k \in \mathrm{C}_1$. Since x_0 is odd we apply the $\frac{3x+1}{2}$ component and find that the next iterate $x_1 = 5 + 24 \cdot k$. Since x_1 is odd we apply the $\frac{3x+1}{2}$ component again and find $x_2 = 8 + 36 \cdot k$, x_2 is odd we apply $\frac{x}{2}$ and so on:

The $3x + 1$ sequence produced by $x_0 = 3 + 16 \cdot k$

Step	0	1	2	3	4
Value	3+16·k	5+24·k	8+36·k	4+18·k	2+9·k

But $3 + 16 \cdot k > 2 + 9 \cdot k$. So the numbers of C_1 class reach a smaller number using four Collatz function steps. The only classes that are still open for our statement are: C_2, C_3 and C_4. We will study each class distinctly:

2.5 Tree of $\mathbf{C_2}$ class

Proposition 3 *All the numbers in class C_2 $(7 + 16 \cdot k)$ reach the number $13 + 27 \cdot k$ after four Collatz's function steps. So the first four steps of the number are depend on the magnitude of k.*

Consider the application of $T(x)$ to the starting number $x_0 = 7 + 16 \cdot k \in \mathrm{C}_2$. Since x_0 is odd we apply the $\frac{3x+1}{2}$ component and find that the next iterate $x_1 = 11 + 24 \cdot k$. Since x_1 is odd we apply the $\frac{3x+1}{2}$ component again and find $x_2 = 17 + 36 \cdot k$, x_2 is odd we apply $\frac{3x+1}{2}$ and so on:

The $3x + 1$ sequence produced by $x_0 = 7 + 16 \cdot k$

Step	0	1	2	3	4
Value	7+16·k	11+24·k	17+36·k	26+54·k	13+27·k

So each number of the form: $7 + 16 \cdot k$ reach the number $13 + 27 \cdot k$. For the number $13 + 27 \cdot k$ is not known immediately if it is either even or odd.

Case	k	$13 + 27 \cdot k$	$T(13 + 27 \cdot k)$
1	even	odd	$20 + 3^4 \cdot \rho$
2	odd	even	$20 + 3^3 \cdot \rho$

- Let k is an even number, then the $13 + 27 \cdot k = 13 + 27 \cdot (2 \cdot \rho)$ is odd. Applying Collatz function: $T(13 + 27 \cdot 2 \cdot \rho) = 20 + 3^4 \cdot \rho$
- If k is odd, then the $13+27 \cdot k$ is even. $13+27 \cdot k = 13+27 \cdot (2 \cdot \rho+1) = 40+3^3 \cdot 2 \cdot \rho$ and applying Collatz function for even case: $T(40 + 3^3 \cdot 2 \cdot \rho) = 20 + 3^3 \cdot \rho$.

Note: $T(40 + 3^3 \cdot 2 \cdot \rho) = 20 + 3^3 \cdot \rho < 7 + 16 \cdot k$ since $\rho = \frac{k-1}{2}$ and $\frac{3^3}{2} < 16$.

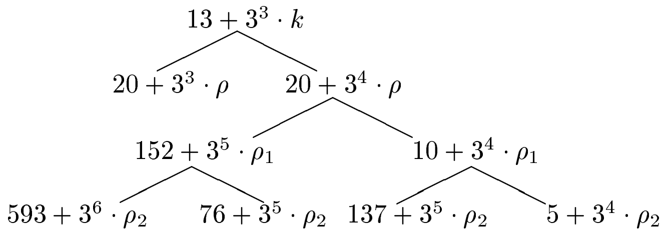
Proposition 4 *Numbers in class C_2 $(7 + 16 \cdot k)$ with $k = 2 \cdot \rho + 1$ (is odd) reach the smaller number $20 + (\frac{27}{2}) \cdot (k - 1)$ after five Collatz's function steps. So at least the fifty percent of the C_2-numbers reduce a smaller number.*

2.6 Binary Tree of C_2 Class (height = 3)

- Right nodes: $k = 2 \cdot \rho$, $\rho_i = 2 \cdot \rho_{i-1}$
- Left nodes: $k = 2 \cdot \rho + 1$, $\rho_i = 2 \cdot \rho_{i-1} + 1$

The terminal nodes (reach a smaller number) are: $20 + 3^3 \cdot \rho$, $5 + 3^4 \cdot \rho_2$.

Step	0	1	2	3	4
Value	7+16·k	11+24·k	17+36·k	26+54·k	13+27·k



2.7 Structure of C_2's Class Tree

Let us assume that each node of the binary tree has a real (Re) and a imaginary (Im) part, like complex analysis. For instance: $13 + 27 \cdot k$ has Re=13 and Im=27=3^3. So the pair (13,3^3) is one abbreviation of the $13 + 27 \cdot k$. Each number of the form $(b, 3^l) = b + 3^l \cdot k$ whether $l \in \mathbb{N}$ has the follow properties:

Case	b	k	(b, 3^l)	Collatz function T(x)
1	even	even	even	(T(b), 3^l)
2	even	odd	odd	(T(b+3^l), 3^{l+1})
3	odd	even	odd	(T(b), 3^{l+1})
4	odd	odd	even	$(T(b+3^l), 3^l)$

Case 1: The b and k are even numbers, so the number $(b, 3^l)$ is also even and applying the Collatz function we get:

$$\frac{b + 3^l \cdot k}{2} = \frac{b}{2} + \frac{3^l \cdot k}{2} = T(b) + \frac{3^l \cdot 2 \cdot \rho}{2} = T(b) + 3^l \cdot \rho = (T(b), 3^l)$$

Case 2: The b and k is an even and an odd number respectively, so the number $(b, 3^l)$ is an odd number and applying the $\frac{3x+1}{2}$ component we found that:

$$\frac{3 \cdot (b + 3^l \cdot k) + 1}{2} = \frac{3 \cdot (b + 3^l \cdot (2 \cdot \rho + 1)) + 1}{2} = \frac{3 \cdot (b + 3^l + 3^l \cdot 2 \cdot \rho) + 1}{2} =$$

$$= \frac{3 \cdot (b + 3^l) + 1}{2} + \frac{3^{l+1} \cdot 2 \cdot \rho}{2} = \frac{3 \cdot (b + 3^l) + 1}{2} + 3^{l+1} \cdot \rho = (T(b + 3^l), 3^{l+1})$$

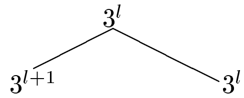
Case 3: The b and k is an odd and an even number respectively, so the number $b+3^l \cdot k$ is odd. We apply the $\frac{3x+1}{2}$ component and find that:

$$\frac{3\cdot(b+3^l\cdot k)+1}{2} = \frac{3\cdot(b+3^l\cdot 2\cdot\rho)+1}{2} = \frac{3\cdot b+1}{2} + \frac{3^{l+1}\cdot 2\cdot\rho}{2} = (T(b), 3^{l+1})$$

Case 4: The b and k are odd numbers, so the $b+3^l\cdot k$ is even. We apply the $\frac{x}{2}$ component and find that:

$$\frac{b+3^l\cdot k}{2} = \frac{b+3^l\cdot(2\cdot\rho+1)}{2} = \frac{b+3^l}{2} + \frac{3^l\cdot 2\cdot\rho}{2} = \frac{b+3^l}{2} + 3^l\cdot\rho = (T(b+3^l), 3^l)$$

For the imaginary part we can note:



2.8 Pascal's Triangle and Collatz Conjecture

Using the previous results about the imaginary part, we can easily represent the C_2's class Tree. We must note in each level of binary tree the terminal nodes:

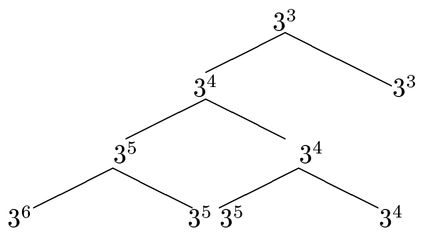
If $\frac{3^y}{2^h} < 16$ where h is the current height of binary tree, y is the smallest power of 3 in that height, 16 is the imaginary part of the root, then the node with 3^y is a terminal node, because this node represents a class of numbers which reach a smaller number.

Imaginary part of C_2's binary class

h =1 $(3^3/2) < 16$

h =2 $(3^4/2^2) > 16$

h =3 $(3^4/2^3) < 16$



Level	Condition	Tree	Terminal Nodes	Percent	Total
0	$\frac{3^3}{2^0} > 16$	3_1	$a_0 = 0$	0	0
1	$\frac{3^3}{2^1} < 16$	$4_1 3_1$	$a_1 = 1$	$\frac{1}{2}$	50
2	$\frac{3^4}{2^2} > 16$	$5_1 4_1$	$a_2 = 0$	0	50
3	$\frac{3^4}{2^3} < 16$	$6_1 5_2 4_1$	$a_3 = 1$	$\frac{1}{8}$	62,5
4	$\frac{3^5}{2^4} < 16$	$7_1 6_3 5_2$	$a_4 = 2$	$\frac{2}{16}$	75
5	$\frac{3^6}{2^5} > 16$	$8_1 7_4 6_3$	a_5=0	0	75
6	$\frac{3^6}{2^6} < 16$	$9_1 8_5 7_7 6_3$	$a_6 = 3$	$\frac{3}{2^6}$	79.69
7	$\frac{3^7}{2^7} > 16$	$10_1 9_6 8_{12} 7_7$	$a_7 = 0$	0	79.69

Let us discuss the previous table:

- In named column Tree the notation 3_1 means: for the specific height we have **one** number with the imaginary part 3^3. In the same way the notation $4_1 3_1$ means: in the first level we have one number with imaginary part 3^4 and one number with imaginary part 3^3. And so on.
- Percent is the percent of nodes that are terminal nodes.
- Total Percent means the percent of numbers in C_2 class, which reach a smaller number.

From the previous table maybe the most interesting is the sequence of pointers, the last sequence shows the Pascal's Triangle:

3_1				1				1				
4_1	3_1			1	1			1	1			
5_1	6_1			1	1			1	2	1		
6_1	5_2	4_1	$\rightarrow$	1	2	1		1	3	3	1	
7_1	6_3	5_2		1	3	2		1	4	6	4	1
8_1	7_4	6_3		1	4	3		1	...	...	...	...

The sequence of pointers it is not the same sequence as Pascal's Triangle, but some part of it is exactly the same. The steps of C_2's construction are:

1. Each element of next row follows the Pascal's Triangle
2. In each row we check for terminal node. If it is exist then we reduce the length of the row and we continue with first step. If it is not exist then we continue with first step.

Thus the difference between Pascal's Triangle and the sequence of pointers is that the length of each row is not the same. In the Pascal's Triangle, in each step the length of row is increased by 1. But the length of each row in sequence of pointers varies from the fact if exists terminal node or not.

```
// C code: sequence of pointers
b=27; l=1; p=0; k=0; // initialization
for(i=2; i<=21; i++) // 21
the height of binary tree
   {
   for (j=2; j<=a+1; j++) // a is the length of ith row
   // Pascal's triangle
   buffer0[i][j]= buffer0[i - 1][j - 1] + buffer0[i - 1][j];
   l=2*l; // power of two
   if(b/l<16) // condition for terminal node
   {
   k=buffer0[i][j-2]; buffer0[i][j-2]=0; // terminal node
   b=3*b; p=(float) k/l+p; // estimation of total percent
   }
   else a=a+1;
}
```

Executing the previous C program we took the following experimental results:

Height	Percent of numbers which reach a smaller number
9	86,13
12	87,93
15	91,54
20	94,19
31	98,21

2.9 Tree of C_3 class

Proposition 5 *All the numbers in class C_3 (11+16·k) reach the number* $20 + 27 \cdot k$ *after four Collatz's function steps. So the first four steps of the number are depend on the magnitude of* k.

The $3x + 1$ sequence produced by $x_0 = 11 + 16 \cdot k$

Step	0	1	2	3	4
Value	11+16·k	17+24·k	26+36·k	13+18·k	20+27·k

2.10 Structure of C_3's class Tree

The root for C_3's class Tree is $20 + 27 \cdot k$. Since $b = 20$ and $l = 3$, arise the same sequence of pointers for the imaginary part as we saw in C_2's class Tree.

Experimental results

Height	Percent of numbers which reach a smaller number
9	86,13
12	87,93
15	91,54
20	94,19
31	98,21

2.11 Tree of C_4 class

Proposition 6 *All the numbers in class C_4 (*$15 + 16 \cdot k$*) reach the number* $80 + 81 \cdot k$ *after four Collatz's function steps. So the first four steps of the C_4-elements are depend on the magnitude of* k.

The $3x + 1$ sequence produced by $x_0 = 15 + 16 \cdot k$

Step	0	1	2	3	4
Value	15+16·k	23+24·k	35+36·k	53+18·k	80+81·k

2.12 Structure of C_4's Class Tree

The root for C_4's class Tree is $80 + 81 \cdot k$. So $b = 80$ and $l = 4$.

Experimental results

Height	Percent of numbers which reach a smaller number
9	56,05
12	67,68
15	71,17
20	82,45
31	93,17

2.13 Conclusion

From the mathematicians standpoint, the fact that it has been shown the conjecture to hold for starting integers exceeding 10^{13} *[4–6] it is not so interesting because empirical evidence is not the same thing as a rigorous mathematical proof and* 10^{13} *is not a large number in mathematics.*

In the previous considerations we tried to face this problem. We conclude for a specific height of our classification, without running the Collatz function for each number, that the majority of numbers reach a smaller number quite fast.

3 An Efficient Algorithm

The $3x + 1$ *has been numerically checked for a large range of values of* x*. It is an interesting problem to find efficient algorithms to test the conjecture on a computer*[1].

The previous Mathematical work gives us the following facts:

1. Each number of A class reaches a smaller number by two Collatz's function steps. Each number of the form $1 + 4 \cdot k$ reaches the number $1 + 3 \cdot k$ after two steps.
2. Each number of B and D class reaches a smaller number by one Collatz's function step.
3. Each number of C_1 class reaches a smaller number by four Collatz's function steps. Each number of the form $3 + 16 \cdot k$ reaches the number $2 + 9 \cdot k$ after four steps.
4. The Binary trees of classes C_2, C_3 and C_4 provide several cases of numbers which reach a smaller number. We will assume that we have constructed the binary tree with height $h = 4$ for each class and we note all the terminal nodes.

Precisely we have the following cases:
For C_2 class we will use the following cases: (starting number $7 + 16 \cdot k$)

Type of k	Smaller Number	Number of steps
k=2·ρ+1	20+($\frac{27}{2}$)·(k-1)	5
k=8·ρ	5+($\frac{81}{8}$)·k	7
k=16·ρ+12	190+($\frac{243}{16}$)·(k-12)	8
k=16·ρ+2	38+($\frac{243}{16}$)·(k-2)	8

For C_3 class we will use the following cases: (starting number $11 + 16 \cdot k$)

Type of k	Smaller Number	Number of steps
k=2·ρ	10+($\frac{27}{2}$)·k	5
k=8·ρ+3	38+($\frac{81}{8}$)·(k-3)	7
k=16·ρ+13	209+($\frac{243}{16}$)·(k-13)	8
k=16·ρ+7	118+($\frac{243}{16}$)·(k-7)	8

For C_4 class we will use the following cases: (starting number $15 + 16 \cdot k$)

Type of k	Smaller Number	Number of steps
k=8·ρ	10+($\frac{81}{8}$)·k	7
k=16·ρ+4	76+($\frac{243}{16}$)·(k-4)	8
k=16·ρ+5	91+($\frac{243}{16}$)·(k-5)	8
k=16·ρ+10	167+($\frac{243}{16}$)·(k-10)	8

Our Algorithm seems to be very similar with mathematical induction. The steps of the algorithm are:

1. Assume that the steps of first 16 numbers are known.
2. For each next number try to find if we have the information that this number reaches a smaller number. The sum of steps until this number reaches the smaller number is stored. If we don't have this information try to run the Collatz function with some improvements.

For example let us begin with 17: 17 belongs in A class and $k = 4$, so 17 reaches the number $1 + 3 \cdot 4 = 13$ after two Collatz's function steps. The stopping time of 13 is 7 so the stopping time for 17 is 9. We continue with 18. Obviously 18 reaches the 9 so the stopping time is 14 since the stopping time of 9 is 13. The stopping time of 19 ($19 \in C_1$: $3 + 16 \cdot 1$) is 14 since the stopping time of $(2 + 9 \cdot 1)$ 11 is 10. And so on.

Some experimental results:

Range Number	C_2-C_3-C_4	Percent
1-176	3-3-7-13	7,3=$\frac{13}{176}$
1-336	5-5-15-25	7,44 = $\frac{25}{336}$
1-1050	12-13-38-63	6 = $\frac{63}{1050}$
1-1616	25-25-69-119	7,36 = $\frac{119}{1616}$
1-9600	150-150-449-749	7,78 = $\frac{749}{9600}$
1-12816	200-200-600-1000	7,80 = $\frac{1000}{12816}$
1-96016	1500-1500-4125-7125	7,42 = $\frac{7125}{96016}$

The second column shows for how many numbers we need to run the Collatz function because we don't have enough information about their reduction to a smaller number. The sequence for example 3-3-7-13 means that we have three numbers from C_2 class, three numbers from C_3 class and seven numbers from C_4 class, total 13 numbers that they don't reach a smaller number. We also know which these numbers are.

- If we extend the cases about how many numbers are reach smaller number from the C_2, C_3 ,C_4 binary trees then we will see that our algorithm is more efficient. But we don't have the opportunity to expand each time the binary trees because that it is equivalent with the execution of Collatz function.

Range Number	Collatz Function (sec)	Our Algorithm (milli sec)
1-12816	1	50
1-32016	1	220
1-64032	3	380
1-96016	5	550

4 Intuition

4.1 Triangle and Intuition

Taking into consideration the odd numbers only, we can construct triangles of the mentioned general type:

Level	a_1	a_2	...	...	a_n
1	a				
2	2·a	3·a			
3	4·a	6·a	9·a		
...	...	...	...	...	...
n	$2^n \cdot a$	$2^{n-1} \cdot 3 \cdot a$	$2^{n-2} \cdot 3^2 \cdot a$	...	$3^{n-1} \cdot a$

Algorithm

- 1st step: For each odd number we construct a triangle as above. We don't construct triangle for odd number which is an element of a previous one.
- 2nd step: For each specific triangle we use the following transformation,
  Transformation (T_1): for each element of the triangle we imply the 2·x-1 function where x is the triangles' element.

So we have obtained the corresponding triangle to the specific one. For example $(a = 1, n = 3)$:

1				1		
2	3		→	3	5	
4	6	9		7	11	17

As it can be easily observed the corresponding triangle in each line give the partial sequence of Collatz function. If we expand the level of triangle we take many steps of Collatz function for a specific number.

4.2 Tables and Intuition

Let us assume the sequence of tables-matrixes with 16 elements, where k=0,1,2, ...:

A_k

A-class	B-class	C-class	D-class
$1 + 16 \cdot k$	$2 + 16 \cdot k$	$3 + 16 \cdot k$	$4 + 16 \cdot k$
$5 + 16 \cdot k$	$6 + 16 \cdot k$	$7 + 16 \cdot k$	$8 + 16 \cdot k$
$9 + 16 \cdot k$	$10 + 16 \cdot k$	$11 + 16 \cdot k$	$12 + 16 \cdot k$
$13 + 16 \cdot k$	$14 + 16 \cdot k$	$15 + 16 \cdot k$	$16 + 16 \cdot k$

We will present a group of movements which show the Collatz's function steps without to estimate for each number the Collatz function.

For A-class elements we know that:

- **Step 1**: An A-element reaches one B element.
- **Step 2**: The next A-element reaches the D-element of the next row.
- **Step 3**: We leave one row and we return in step 1.

The next sequence of numbers shows the previous description, note that the symbol $\square\ \alpha$ means: the specific number is reached from α number.

A-class	B-class	C-class	D-class
1	2 $\square$ **1**	3	4
5	6	7	8 $\square$ **5**
9	10	11	12
13	14 $\square$ **9**	15	16
17	18	19	20 $\square$ **13**
21	22	23	24
25	26 $\square$ **17**	27	28
29	30	31	32 $\square$ **21**

For B-class elements we know that:

- **Step 1**: An B-element reaches one A- element.
- **Step 2**: The next B-element reaches the C-element of the of the same row.
- back again in step 1, in the next row.

For D-class elements we know that:

- **Step 1**: An D-element reaches one A- element.
- **Step 2**: The next D-element reaches the B-element of the same row.
- **Step 3**: The next D-element reaches the C-element of the same row.
- **Step 4**: The next D-element reaches the D-element of the same row.
- back again in step 1, in the next row.

4.3 C_1-class Moving Rules:

- **Step 1:** An C_1-element reaches the second A- element of one specific matrix.
- **Step 2:** The next C_1-element
  reaches the fourth A-element of the next matrix.
- **Step 3**: Leaving the next matrix we return to the step 1.

4.4 C_2-class Moving Rules:

- **Step 1:** An C_2-element reaches a C_1- element of one specific matrix.
- **Step 2:** The next C_2-element reaches a C_3- element of the next matrix.
- **Step 3**: Leaving the next matrix we return to the step 1.

4.5 C_3-class Moving Rules:

- **Step 1:** An C_3-element reaches the first A- element of one specific matrix.
- **Step 2:** The next C_3-element reaches the third A- element of the next matrix.
- **Step 3:**Leaving the next matrix we return to the step 1.

4.6 C_4-class Moving Rules:

- **Step 1:** An C_4-element reaches a C_2- element of one specific matrix.
- **Step 2:** The next C_4-element reaches a C_4-element of the next matrix.
- **Step 3:** Leaving the next matrix we return to the step 1.

- The symbol $\clubsuit\alpha$ means: the specific number is reached from α where α is C_1-element number.
- The symbol $\diamondsuit\alpha$ means: the specific number is reached from α where α is C_2-element number.
- The symbol $\spadesuit\alpha$ means: the specific number is reached from α where α is C_3-element number.
- The symbol $\heartsuit\alpha$ means: the specific number is reached from α where α is C_4-element number.

A-class	B-class	C-class	D-class
1	2	3	4
5 $\clubsuit$3	6	7	8
9	10	11 $\diamondsuit$7	12
13	14	15	16
17 $\spadesuit$11	18	19	20
21	22	23 $\heartsuit$15	24
25	26	27	28
29 $\clubsuit$19	30	31	32
33	34	35 $\diamondsuit$23	36
37	38	39	40
41 $\spadesuit$27	42	43	44
45	46	47 $\heartsuit$31	48

Conclusion: After the previous work is easy to observe that using moving rules of each class we can simulate the Collatz function.

Acknowledgments: I would like to particularly name Prof. Efstathios Zachos. Professor Zachos was the first person that told me about Collatz Conjecture and he gave me the chance to present my work in his research seminar. Many thanks also to: Dr. Aris Pagourtzis and Prof. Keith R. Matthews, for their useful remarks and ideas.

References

1. J.C. Lagarias, The 3x+1 Problem and its Generalizations, *The American Mathematical Monthly*, 92(1):3-23, (1985)
2. T. Oliveira e Silva, Maximum Excursion and Stopping Time Record-holders for the 3x+1 Problem: Computational Results, *Mathematics of Computation*, 68(225):371-384, (1999)
3. T. Oliveira e Silva, http://www.ieeta.pt/~tos/3x+1.html
4. M.A. Herkommer, *Number Theory: a Programmer's Guide*, McGraw-Hill, New York, (1999)
5. E. Roosendaal, http://personal.computrain.nl/eric/wondrous/
6. D.B. Meadle, C.A. Nicol, Maple Tools for Use in Conjecture Testing and Iteration Mappings in Number Theory. http://www.math.sc.edu/~meade/papers/imi93:06.ps
7. T. Mimuro, On Certain Simple Cycles of the Collatz Conjecture *SUT Journal of Mathematics* 37(2):79-89 (2001)
8. J.L. Pe. A Heuristic Argument for the Collatz Problem, iDEN System Engineering Tools and Statistics. http://www.geocities.com/SoHo/Exhibit/8033/collatz/collatz.htm
9. P. Schorer, The Structure of the 3x+1 Function, Hewlett-Packard Laboratories, Palo Alto, CA (ret.). http://www.occampress.com
10. K.R. Matthews, The Generalized 3x+1 Mapping, http://www.maths.uq.edu.au/~krm/.
11. K.G. Monks. 3x +1 Minus the +, Department of Mathematics, Univ. of Scranton. Discrete Mathematics and Theoretical Computer Science 5, 47-54.
12. P. Cheeseman, B. Kanefsky, W.M. Taylor, Where the Really Hard Problems Are. *Proceedings IJCAI Conference*, (1991) 163-169
13. K. Rosen, Elementary Number Theory and Its Applications Fourth Edition, ATT Bell Laboratories, http://www.aw-bc.com/rosen/
14. J.H. Conway. Unpredictable Iterations. *Proceedings Conference on Number Theory*, University of Colorado, Boulder, CO, (1972) 49-52
15. R. Terras. A Stopping Time Problem on the Positive Integers. *Acta Arith.* 30:241-252, (1976)
16. R. Terras. On the Existence of a Density. *Acta Arith.* 35:101-102, (1979)