

Collatz Dynamics is Partitioned by Residue Class Regularly

Wei Ren¹

¹*School of Computer Science, China University of Geosciences, Wuhan, China, weirencs@cug.edu.cn*

Abstract

We propose Reduced Collatz Conjecture that is equivalent to Collatz Conjecture, which states that every positive integer can return to an integer less than it, instead of 1. Reduced Collatz Conjecture should be easier because some properties are presented in reduced dynamics, rather than in original dynamics (e.g., ratio and period). Reduced dynamics is a computation sequence from starting integer to the first integer less than it, and original dynamics is a computation sequence from starting integer to 1. Reduced dynamics is a component of original dynamics. We denote dynamics of x as a sequence of either computations in terms of “I” that represents $(3*x+1)/2$ and “O” that represents $x/2$. Here $3*x+1$ and $x/2$ are combined together, because $3*x+1$ is always even and followed by $x/2$. We formally prove that all positive integers are partitioned into two halves and either presents “I” or “O” in next ongoing computation. More specifically, (1) if any positive integer x that is i module 2^t (i is an odd integer) is given, then the first t computations (each one is either “I” or “O” corresponding to whether current integer is odd or even) will be identical with that of i . (2) If current integer after t computations (in terms of “I” or “O”) is less than x , then reduced dynamics of x is available. Otherwise, the residue class of x (namely, i module 2^t) can be partitioned into two halves (namely, i module 2^{t+1} and $i+2^t$ module 2^{t+1}), and either half presents “I” or “O” in intermediately forthcoming $(t+1)$ -th computation.

Keywords: Collatz Conjecture, $3x+1$ Problem, Arithmetics, Residue Class, Computational Number Theory, Discrete Dynamics Systems

September 5, 2022

The Collatz conjecture can be stated simply as follows: Take any positive integer number x . If x is even, divide it by 2 to get $x/2$. If x is odd, multiply it by 3 and add 1 to get $3 * x + 1$. Repeat the process again and again. The Collatz conjecture is that no matter what the number (i.e., x) is taken, the process will always eventually reach 1.

The Collatz conjecture can be stated simply as follows: Take any positive integer number x . If x is even, divide it by 2 to get $x/2$. If x is odd, multiply it by 3 and add 1 to get $3 * x + 1$. Repeat the process again and again. The Collatz conjecture is that no matter what the number (i.e., x) is taken, the process will always eventually reach 1.

The current known integers that have been verified are about 60 bits by T.O. Silva using normal personal computers [1, 2]. They verified all integers that are less than 60 bits, but it is not clear whether their method is able to check extremely large integers, e.g., integers with length more than 100000 bits.

Wei Ren et al. [3] verified $2^{100000} - 1$ can return to 1 after 481603 times of $3 * x + 1$ computation, and 863323 times of $x/2$ computation, which is the largest integer being verified in the world. Wei Ren [4] also propose a new approach on proving Collatz conjecture by exploring reduced dynamics on Collatz conjecture. This proposed approach provides the linkage between Collatz conjecture and reduced Collatz conjecture, and the rationale why exploring empirical and experimental results on reduced Collatz conjecture dynamics can facilitate the proofing of reduced Collatz conjecture. Wei Ren [5] proposed to use a tree-based graph to reveal two key inner properties in reduced Collatz dynamics: one is ratio of the count of $x/2$ over the count of $3*x+1$ (for any reduced Collatz dynamics, the count of $x/2$ over the count of $3*x+1$ is larger than $\ln 3/\ln 2$), and the other is partition (all positive integers are partitioned regularly corresponding to ongoing dynamics). Wei Ren et al. [6] also proposed an automata method for fast computing Collatz dynamics. All source code and output data by computer programs in those related papers can be accessed in public repository [7].

1. Preliminaries

Notation 1.1.

- (1) \mathbb{N}^* : positive integers;
- (2) $\mathbb{N} = \mathbb{N}^* \cup \{0\}$;
- (3) $[1]_2 = \{x | x \equiv 1 \pmod{2}, x \in \mathbb{N}^*\}$; $[0]_2 = \{x | x \equiv 0 \pmod{2}, x \in \mathbb{N}^*\}$.
- (4) $[i]_m = \{x | x \equiv i \pmod{m}, x \in \mathbb{N}^*, m \geq 2, m \in \mathbb{N}^*, 0 \leq i \leq m-1, i \in \mathbb{N}\}$.

$3*x + 1$ is always followed by $x/2$, we thus can represent required computations as $(3*x + 1)/2$ and $x/2$, which are denoted by $I(x)$ and $O(x)$, respectively.

Note that, $I(x)$ and $O(x)$ can be simply denoted as $I(\cdot)$ and $O(\cdot)$, or I and O , respectively. Obviously, $\forall x \in \mathbb{N}^*$, $I(x) = (3*x + 1)/2 > x$, $O(x) = x/2 < x$. That is the reason we call the notations as I (to represent “Increase”) and O (to represent “dOwn”).

Definition 1.2. Collatz transformation, denoted as $f(\cdot)$, where $f(\cdot) = I(\cdot) = (3*x + 1)/2$ if $x \in [1]_2$, and $f(\cdot) = O(\cdot) = x/2$ if $x \in [0]_2$.

Remark 1.3.

- (1) We assume $f^0(x) = x$.
- (2) $f_n(f_{n-1}(\dots f_2(f_1(x))))$ can be written as $f_1 \| f_2 \| \dots \| f_n(x)$, where $f_i(\cdot) \in \{I(\cdot), O(\cdot)\}$ ($i = 1, 2, \dots, n$) and “ $\|$ ” is the concatenation of two Collatz transformations (either “ I ” or “ O ”). For simplicity, we just denote $f_i(\cdot)$ as $f \in \{I, O\}$.

Example 1.4.

- (1) The Collatz transformations for 1 from starting to 1 is IO , because $1 \rightarrow 4 \rightarrow 2 \rightarrow 1$.
- (2) The Collatz transformations for 3 from starting to 1 is $IIOOO$, because $3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$.

Notation 1.5. $|\cdot|$. “ $|x|$ ” returns the length of $x \in \{I, O\}^{\geq 1}$, in terms of the total count of I and O . E.g., $|IIOO| = 4$.

Definition 1.6. $IsMatched : x \times c \rightarrow \text{bool}$. It takes as input $x \in \mathbb{N}^*$ and $c \in \{I, O\}$, and outputs $\text{bool} \in \{\text{True}, \text{False}\}$. If $x \in [1]_2$ and $c = I$, or if $x \in [0]_2$ and $c = O$, then output $\text{bool} = \text{True}$; Otherwise, output $\text{bool} = \text{False}$.

Remark 1.7. Simply speaking, this function checks whether the forthcoming Collatz transformation (i.e., $c \in \{I, O\}$) matches with the current integer x . That is the reason we call it “Is Matched”.

Definition 1.8. $GetS : s \times i \times j \rightarrow s'$. It takes as input s, i, j , where $s \in \{I, O\}^{|s|}$, $1 \leq i \leq |s|$, $0 \leq j \leq |s| - (i - 1)$, $i \in \mathbb{N}^*$, $j \in \mathbb{N}$, and outputs s' where s' is a segment in s that starts from the location i in s and $|s'| = j$. Note that, $GetS(\cdot, \cdot, 0)(x) = x$ when $j = 0$.

Example 1.9. $GetS(IIOO, 1, 1) = I$, $GetS(IIOO, 1, 2) = II$, $GetS(IIOO, 1, 3) = IIO$, $GetS(IIOO, 1, 4) = IIOO$.

Remark 1.10.

(1) s' is a selected segment in s that starts from the location i and has the length of j . That is the reason we call this function as “Get Substring”.

(2) Simply speaking, this function outputs the Collatz transforms from i to $i + j - 1$ in an inputting transformation sequence $s \in \{I, O\}^{|s|}$.

(3) Especially, $GetS(s, 1, |s|) = s$. $GetS(s, |s|, 1)$ returns the last transformation in s . $GetS(s, 1, 1)$ returns the first transformation in s . $GetS(s, j, 1)$ returns the j -th transformation in s .

(4) Note that, $GetS(\cdot)$ itself is a function. In other words, it can be looked as $GetS(\cdot)(\cdot)$. For example,

$$GetS(IIOO, 1, 1)(3) = I(3) = (3 * 3 + 1) / 2 = 5,$$

$$GetS(IIOO, 1, 2)(3) = II(3) = I(I(3)) = I(5) = (3 * 5 + 1) / 2 = 8,$$

$$GetS(IIOO, 1, 3)(3) = IIO(3) = O(II(3)) = O(8) = 8 / 2 = 4,$$

$$GetS(IIOO, 1, 4)(3) = IIOO(3) = O(IIO(3)) = O(4) = 4 / 2 = 2 < 3.$$

Besides,

$IsMatched(GetS(IIOO, 1, i)(x), GetS(IIOO, i+1, 1)) = \text{True}$ ($i = 0, 1, \dots, 3$).

Notation 1.11. Original dynamics of x . It is the sequence of occurred Collatz transformations from x to 1.

For example, the original dynamics of 5 is $IODOO$ due to $5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$.

Notation 1.12. Reduced dynamics of x . It is the sequence of occurred Collatz transformations from x to the first integer that is less than x .

For example, the reduced dynamics of 5 is IO due to $5 \rightarrow 16 \rightarrow 8 \rightarrow 4$.

Definition 1.13. *Collatz Conjecture.* $\forall x \in \mathbb{N}^*, \exists L \in \mathbb{N}^*,$ such that

- (1) $s(x) = 1$ where $s \in \{I, O\}^L;$
- (2) $IsMatched(GetS(s, 1, i)(x), GetS(s, i+1, 1)) = True$ where $i = 0, 1, \dots, L-1.$

Obviously, Collatz conjecture is held when $x = 1$. In the following, we mainly concern $x \geq 2, x \in \mathbb{N}^*$.

Definition 1.14. *Reduced Collatz Conjecture.* $\forall x \in \mathbb{N}^*, x \geq 2, \exists L \in \mathbb{N}^*,$ such that

- (1) $s(x) < x$ where $s \in \{I, O\}^L;$
- (2) $GetS(s, 1, i)(x) \not\leq x, i = 1, \dots, L-1;$
- (3) $IsMatched(GetS(s, 1, j)(x), GetS(s, j+1, 1)) = True$ where $j = 0, 1, \dots, L-1.$

Obviously, L must be the minimal positive integer such that $s(x) < x.$

Theorem 1.15. *Collatz Conjecture is equivalent to Reduced Collatz Conjecture.*

Straightforward. Please check our another paper[4].

Remark 1.16.

(1) *Ordered sequence $s = s_0 \| s_1 \| \dots \| s_{n-1}, s \in \{I, O\}^L$ in above proof is original dynamics (referring to $s(x) = 1$), which consists of $L = \sum_{i=0}^{n-1} |s_i|$ Collatz transformations during the computing procedure from a starting integer (i.e., x) to 1.*

(2) *In contrast, s_0 in above proof is reduced dynamics (referring to $s_0(x) < x$), which is represented by a sequence of occurred Collatz transformations during the computing procedure from a starting integer (i.e., x) to the first transformed integer that is less than the starting integer.*

(3) *Obviously, reduced dynamics is more primitive than original dynamics, because original dynamics consists of reduced dynamics. Simply speaking, reduced dynamics are building blocks of original dynamics.*

(4) We thoroughly study the relation between Collatz conjecture and Reduced Collatz conjecture [4]. Especially, we also extensively study why Reduced Collatz conjecture will be much easier to explore, or why reduced dynamics presents better properties than original dynamics, e.g., period [8], and ratio [9]. Due to above theorem, we thus only need to concentrate on reduced dynamics.

Notation 1.17. $\text{RD}[x]$. It denotes reduced dynamics of x that are represented by $\{I, O\}^{\geq 1}$. Formally, $\forall x \in \mathbb{N}^*, x \geq 2$, if $\exists L \in \mathbb{N}^*$ such that $s(x) < x$, $s \in \{I, O\}^L$, $\text{GetS}(s, 1, i)(x) \not\prec x$, $i = 1, \dots, L - 1$, and $\text{IsMatched}(\text{GetS}(s, 1, j)(x), \text{GetS}(s, j + 1, 1)) = \text{True}$ where $j = 0, 1, \dots, L - 1$, then s is called reduced dynamics of x , and denoted as $\text{RD}[x] = s$.

Remark 1.18.

(1) Obviously, $\text{RD}[x \in [0]_2] = O$.

(2) $IIOO$ can be denoted in short as I^2O^2 . $IIIOIOO$ can be denoted in short as I^3OIO^2 . In other words, we denote $\underbrace{I \dots I}_n$ as I^n , and we denote $\underbrace{O \dots O}_n$ as O^n where $n \in \mathbb{N}^*, n \geq 2$. We also assume $I^1 = I$, $O^1 = O$.

(3) For example, $\text{RD}[3] = IIOO$, $\text{RD}[5] = IO$, $\text{RD}[7] = IIIOIOO$, $\text{RD}[9] = IO$, $\text{RD}[11] = IIIOIO$. Indeed, we design computer programs [7] that output all $\text{RD}[x]$ for $\forall x \in [1, 99999999]$.

(4) In fact, we proved some results on $\text{RD}[x]$ for specific x , e.g., $\text{RD}[x \in [1]_4] = IO$, $\text{RD}[x \in [3]_{16}] = IIOO$, $\text{RD}[x \in [11]_{32}] = IIIOIO$, et al. [5].

(5) In fact, we formally proved that the ratio exists in any reduced Collatz dynamics [9]. That is, the count of $x/2$ over the count of $3*x + 1$ is larger than $\log_2 3$. We also proved that reduced dynamics is periodical and its period equals 2 to the power of the count of $x/2$ [8]. More specifically, if there exists reduced dynamics of x , then there exists reduced dynamics of $x + P$, where $P = 2^L$ and L is the total count of $x/2$ computations in reduced dynamics of x (i.e., $L = |\text{RD}[x]|$). Moreover, the ratio and period can also be observed and verified in our proposed tree-based graph [5].

Example 1.19. $\text{RD}[5] = IO$, if and only if

$$I(5) = (3 * 5 + 1)/2 = 8 \not\prec 5;$$

$$IO(5) = O(I(5)) = O(8) = 8/2 = 4 < 5;$$

$IsMatched(GetS(1, 1, i)(5), GetS(1, i + 1, 1)) = True, i = 0, 1.$

Example 1.20. $x \in \mathbb{N}^*, x \geq 2$. If $\text{RD}[x]$ exists, then

- (1) $s(x) < x$, where $s = \text{RD}[x]$;
- (2) $GetS(s, 1, i)(x) \not< x$, where $i = 1, 2, \dots, |s| - 1$;
- (3) $IsMatched(GetS(s, 1, j)(x), GetS(s, j+1, 1)) = True$ where $j = 0, 1, \dots, |s| - 1$.

Proposition 1.21. $\forall x \in \mathbb{N}^*, x \geq 2$, if $\text{RD}[x]$ exists, then $\text{RD}[x]$ is unique.

Proof Straightforward. Given x , either $I(x)$ or $O(x)$ is deterministic and unique. Similarly, given x , $s'(x)$ is deterministic and unique, where $s' = GetS(s, 1, i)$, $s = \text{RD}[x]$, $i = 1, 2, \dots, |s|$. Thus, s is unique for any given x . \square

Remark 1.22.

(1) We assume $\text{RD}[x = 1] = IO$, although $IO(1) = O((3*1+1)/2) = O(2) = 2/2 = 1 \not< x$. In other words, we assume the reduced dynamics of $x = 1$ is IO . In the following, we always concern $x \geq 2, x \in \mathbb{N}^*$.

(2) As Collatz conjecture is equivalent to Reduced Collatz conjecture (recall Theorem 1.15), we thus only need to prove Reduced Collatz conjecture is true. By using notation $\text{RD}[x]$, we thus only need to prove $\forall x \in \mathbb{N}^*, \exists \text{RD}[x]$.

Proposition 1.23. Given $x \in \mathbb{N}^*$, if $\text{RD}[x]$ exists, then $\text{RD}[x]$ ends by O .

Proof Straightforward due to $I(x) = (3*x + 1)/2 > x$. Suppose $\exists x \in \mathbb{N}^*, x \geq 2, s(x) \not< x$, $\text{RD}[x] = s \parallel I$. Then, $\{s \parallel I\}(x) = I(s(x)) > s(x)$, thus $\text{RD}[x] = \{s \parallel I\}(x) \not< x$. Contradiction occurs. \square

Proposition 1.24. $\text{RD}[x \in [0]_2] = O$, $\text{RD}[x \in [1]_4] = IO$.

Proof Straightforward. \square

Notation 1.25. $S_{\text{RD}} = \{s | x \in \mathbb{N}^*, \exists \text{RD}[x], s = \text{RD}[x], s \in \{I, O\}^{\geq 1}\}$.

That is, $\forall x \in \mathbb{N}^*$, if $\exists \text{RD}[x]$, then $\text{RD}[x] = s$ will be included in S_{RD} , which is a set of existing reduced dynamics.

Proposition 1.26. $O \in S_{\text{RD}}$, $IO \in S_{\text{RD}}$.

Proof Straightforward. $\text{RD}[x \in [0]_2] = O$ and $\text{RD}[x \in [1]_4] = IO$ by Proposition 1.24. \square

Theorem 1.27. (Subset Theorem.) Suppose $s \in S_{\text{RD}}$, $|s| \geq 2$, $x \in \mathbb{N}^*$, $i = 0, 1, \dots, |s| - 2$. We have

- (1.1) $\{x | \text{GetS}(s, 1, i+1)(x) \in [1]_2\} \subset \{x | \text{GetS}(s, 1, i)(x) \in [0]_2\}$;
- (1.2) $\{x | \text{GetS}(s, 1, i+1)(x) \in [0]_2\} \subset \{x | \text{GetS}(s, 1, i)(x) \in [0]_2\}$;
- (2.1) $\{x | \text{GetS}(s, 1, i+1)(x) \in [1]_2\} \subset \{x | \text{GetS}(s, 1, i)(x) \in [1]_2\}$;
- (2.2) $\{x | \text{GetS}(s, 1, i+1)(x) \in [0]_2\} \subset \{x | \text{GetS}(s, 1, i)(x) \in [1]_2\}$.

Proof When $i = 0$, there exists two and only two cases as follows:

(1) If $\text{GetS}(s, i+1, 1) = O$, then $\text{GetS}(s, 1, i)(x) \in [0]_2$. There exists two subcases as follows:

- (1.1) If $\text{GetS}(s, i+2, 1) = I$, then

$$O(\text{GetS}(s, 1, i)(x)) \in [1]_2$$

$$\Rightarrow \text{GetS}(s, 1, i)(x)/2 \in [1]_2$$

$$\Rightarrow \text{GetS}(s, 1, i)(x) \in [2]_4 \subset [0]_2.$$

Thus, $\{x | \text{GetS}(s, 1, i+1)(x) \in [1]_2\} \subset \{x | \text{GetS}(s, 1, i)(x) \in [0]_2\}$.

- (1.2) If $\text{GetS}(s, i+2, 1) = O$, then

$$O(\text{GetS}(s, 1, i)(x)) \in [0]_2$$

$$\Rightarrow \text{GetS}(s, 1, i)(x)/2 \in [0]_2$$

$$\Rightarrow \text{GetS}(s, 1, i)(x) \in [0]_4 \subset [0]_2.$$

Thus, $\{x | \text{GetS}(s, 1, i+1)(x) \in [0]_2\} \subset \{x | \text{GetS}(s, 1, i)(x) \in [0]_2\}$.

(2) If $\text{GetS}(s, i+1, 1) = I$ then $\text{GetS}(s, 1, i)(x) \in [1]_2$. There exists two subcases as follows:

- (2.1) If $\text{GetS}(s, i+2, 1) = I$, then

$$I(\text{GetS}(s, 1, i)(x)) \in [1]_2$$

$$\Rightarrow (3 * \text{GetS}(s, 1, i)(x) + 1)/2 \in [1]_2$$

$$\Rightarrow 3 * \text{GetS}(s, 1, i)(x) + 1 \in [2]_4$$

$$\Rightarrow 3 * \text{GetS}(s, 1, i)(x) \in [1]_4$$

$$\Rightarrow \text{GetS}(s, 1, i)(x) \in [3]_4 \subset [1]_2.$$

Thus, $\{x | \text{GetS}(s, 1, i+1)(x) \in [1]_2\} \subset \{x | \text{GetS}(s, 1, i)(x) \in [1]_2\}$.

- (2.2) If $\text{GetS}(s, i+2, 1) = O$, then

$$I(\text{GetS}(s, 1, i)(x)) \in [0]_2$$

$$\Rightarrow (3 * \text{GetS}(s, 1, i)(x) + 1)/2 \in [0]_2$$

$$\Rightarrow 3 * \text{GetS}(s, 1, i)(x) + 1 \in [0]_4$$

$$\Rightarrow 3 * \text{GetS}(s, 1, i)(x) \in [3]_4$$

$$\Rightarrow \text{GetS}(s, 1, i)(x) \in [1]_4 \subset [1]_2.$$

Thus, $\{x|GetS(s, 1, i+1)(x) \in [0]_2\} \subset \{x|GetS(s, 1, i)(x) \in [1]_2\}$.

We can prove similarly for $i = 1, 2, \dots, |s| - 2$. \square

Corollary 1.28. Suppose $s \in S_{RD}$, $|s| \geq 2$, $i = 0, 1, \dots, |s| - 2$, $x \in \mathbb{N}^*$. We have

- (1.1) $\{x|GetS(s, 1, i+1)(x) \in [1]_2\} = \{x|GetS(s, 1, i)(x) \in [2]_4\}$;
- (1.2) $\{x|GetS(s, 1, i+1)(x) \in [0]_2\} = \{x|GetS(s, 1, i)(x) \in [0]_4\}$;
- (2.1) $\{x|GetS(s, 1, i+1)(x) \in [1]_2\} = \{x|GetS(s, 1, i)(x) \in [3]_4\}$;
- (2.2) $\{x|GetS(s, 1, i+1)(x) \in [0]_2\} = \{x|GetS(s, 1, i)(x) \in [1]_4\}$.

Proof Straightforward by Theorem 1.27. \square

Remark 1.29.

(1) Corollary 1.28 states that residue classes are partitioned regularly into halves and either half will present either I or O in next intermediate transformation.

(2) Note that, Theorem 1.27 is not only guaranteed for reduced dynamics, but also for original dynamics.

A new notation $I'(\cdot)$ is introduced hereby to reveal the relations among $I(x+P)$, $I(x)$ and $I'(P)$.

Notation 1.30. $I'(x) = (3 * x)/2$.

Example 1.31.

(1) $I(3+16) = (3(3+16)+1)/2 = (3*3+1)/2 + 3*16/2 = I(3) + I'(16)$, $I(3) = (3*3+1)/2 = 5$, $I'(16) = 3*16/2 = 24 \in [0]_2$, $5 > 3$, $5+24 > 3+16$. Thus, either next transformation for $3+16$ and 3 is I .

(2) $II(3+16) = I(I(3) + I'(16)) = (3 * (I(3) + I'(16)) + 1)/2 = (3I(3) + 1)/2 + 3I'(16)/2 = II(3) + I'I'(16)$,

$II(3) = I(5) = (3*5+1)/2 = 8$, $I'I'(16) = I'(24) = 3*24/2 = 36 \in [0]_2$, $8 > 3$, $8+36 > (3+16)$. Thus, either next transformation is O .

(3) $IIO(3+16) = O(II(3) + I'I'(16)) = IIO(3) + I'I'O(16)$,

$IIO(3) = 8/2 = 4$, $I'I'O(16) = 36/2 = 18 \in [0]_2$, $4 > 3$, $4+18 = 22 > (3+16)$. Thus, either next transformation is O .

(4) $IIOO(3+16) = O(IIO(3) + I'I'O(16)) = IIOO(3) + I'I'OO(16)$

$IIOO(3) = 4/2 = 2$, $I'I'OO(16) = 18/2 = 9$, $2 < 3$, $2+9 = 11 < (3+16)$.

Thus, either reduced dynamics ends.

For better presentation, we thus introduce two functions as follows:

Definition 1.32. *IsEven : $x \rightarrow \text{bool}$. It takes as input $x \in \mathbb{N}^*$, and outputs $\text{bool} \in \{\text{True}, \text{False}\}$, where $\text{bool} = \text{True}$ if $x \in [0]_2$ and $\text{bool} = \text{False}$ if $x \in [1]_2$.*

Definition 1.33. *Replace : $s \rightarrow s'$. It takes as input $s \in \{I, O\}^{\geq 1}$, and outputs $s' \in \{I', O\}^{\geq 1}$, where $\text{GetS}(s', i, 1) = I'$ if $\text{GetS}(s, i, 1) = I$, and $\text{GetS}(s', i, 1) = O$ if $\text{GetS}(s, i, 1) = O$, for $i = 1, 2, \dots, |s|$.*

Remark 1.34.

(1) *Simply speaking, replacing all “I” in “s” respectively by “I’” will result in “s’”.*

(2) *Obviously, $\forall s \in \{I, O\}^{\geq 1}$, $|s'| = |s|$ where $s' = \text{Replace}(s)$.*

Lemma 1.35. (1) *If $P \in [0]_2, x \in \mathbb{N}^*$, then $\text{IsEven}(x + P) = \text{IsEven}(x)$.*

(2) *If $P \in [0]_2, x - P > 0, x \in \mathbb{N}^*$, then $\text{IsEven}(x - P) = \text{IsEven}(x)$.*

Proof Straightforward. Due to $P \in [0]_2$, if $x \in [1]_2$, then $x + P, x - P \in [1]_2$; If $x \in [0]_2$, then $x + P, x - P \in [0]_2$. Thus, $\text{IsEven}(x + P) = \text{IsEven}(x) = \text{IsEven}(x - P)$. \square

Remark 1.36. *Above lemma states that if $P \in [0]_2$, the first Collatz transformation of $x + P$ (or $x - P > 0$) is identical with that of x .*

Lemma 1.37. *$s(x + P) = s(x) + s'(P)$, where $s \in \{I, O\}$, $s' = \text{Replace}(s)$, $x \in \mathbb{N}^*$, $P \in [0]_2$.*

Proof $\text{IsEven}(x + P) = \text{IsEven}(x)$ because $P \in [0]_2$, due to Lemma 1.35. Thus, the first Collatz transformation of $x + P$ and that of x are identical.

(1) Suppose $x \in [1]_2$, so $s = I$. Thus, $s' = \text{Replace}(s) = I'$.

$s(x + P) = I(x + P) = 3((x + P) + 1)/2 = (3x + 1)/2 + 3 * P/2 = I(x) + I'(P) = s(x) + s'(P)$.

(2) Suppose $x \in [0]_2$, so $s = O$. Thus, $s' = \text{Replace}(s) = O$.

$s(x + P) = O(x + P) = (x + P)/2 = x/2 + P/2 = O(x) + O(P) = s(x) + s'(P)$.

Summarizing (1) and (2), $s(x + P) = s(x) + s'(P)$. \square

Remark 1.38. *Similarly, we can prove that $s(x - P) = s(x) - s'(P)$, where $s \in \{I, O\}$, $s' = \text{Replace}(s)$, $x \in \mathbb{N}^*$, $P \in [0]_2$.*

Lemma 1.39. (Separation Lemma.) *Suppose $x \in \mathbb{N}^*$, $s \in \{I, O\}^{\geq 2}$, $s' = \text{Replace}(s)$. If $\text{GetS}(s', 1, j)(P) \in [0]_2$, $j = 0, 1, 2, \dots, |s| - 1$, then*

- (1) $\text{IsEven}(\text{GetS}(s, 1, j)(x + P)) = \text{IsEven}(\text{GetS}(s, 1, j)(x))$;
- (2) $\text{GetS}(s, 1, j + 1)(x + P) = \text{GetS}(s, 1, j + 1)(x) + \text{GetS}(s', 1, j + 1)(P)$.

Proof (1) $j = 0$.

(1.1) $\text{GetS}(s', 1, j)(P) \in [0]_2$. That is, $\text{GetS}(s', 1, 0)(P) = P \in [0]_2$. Thus, $\text{IsEven}(x + P) = \text{IsEven}(x)$ due to Lemma 1.35. Thus, the intermediate next Collatz transformation of $x + P$ and x are identical.

$$\begin{aligned} & (1.2) \text{GetS}(s, 1, j + 1)(x + P) \\ &= \text{GetS}(s, 1, 1)(x + P) \because j = 0 \\ &= \text{GetS}(s, 1, 1)(x) + \text{GetS}(s', 1, 1)(P) \because \text{Lemma 1.37} \\ &= \text{GetS}(s, 1, j + 1)(x) + \text{GetS}(s', 1, j + 1)(P). \end{aligned}$$

(2) $j = 1$.

(2.1) Due to (1), $\text{GetS}(s, 1, 1)(x + P) = \text{GetS}(s, 1, 1)(x) + \text{GetS}(s', 1, 1)(P)$. Besides, $\text{GetS}(s', 1, 1)(P) \in [0]_2$. Thus, $\text{IsEven}(\text{GetS}(s, 1, 1)(x + P)) = \text{IsEven}(\text{GetS}(s, 1, 1)(x))$. Thus, the intermediate next Collatz transformation of $x + P$ and x are identical.

(2.2) There exists two cases as follows:

(2.2.1) If $\text{GetS}(s, 1, j + 1) = \text{GetS}(s, 1, j) \parallel I$, then

$$\begin{aligned} & \text{GetS}(s, 1, j + 1)(x + P) \\ &= (\text{GetS}(s, 1, j) \parallel I)(x + P) \\ &= I(\text{GetS}(s, 1, j)(x + P)) \\ &= I(\text{GetS}(s, 1, j)(x) + \text{GetS}(s', 1, j)(P)) \because (1.2) \\ &= (3(\text{GetS}(s, 1, j)(x) + \text{GetS}(s', 1, j)(P)) + 1)/2 \\ &= (3 * \text{GetS}(s, 1, j)(x) + 1)/2 + 3 * \text{GetS}(s', 1, j)(P)/2 \\ &= I(\text{GetS}(s, 1, j)(x)) + I'(\text{GetS}(s', 1, j)(P)) \\ &= (\text{GetS}(s, 1, j) \parallel I)(x) + (\text{GetS}(s', 1, j) \parallel I')(P) \\ &= \text{GetS}(s, 1, j + 1)(x) + \text{GetS}(s', 1, j + 1)(P). \end{aligned}$$

(2.2.2) If $\text{GetS}(s, 1, j + 1) = \text{GetS}(s, 1, j) \parallel O$, then

$$\begin{aligned} & \text{GetS}(s, 1, j + 1)(x + P) \\ &= \text{GetS}(s, 1, j) \parallel O(x + P) \\ &= O(\text{GetS}(s, 1, j)(x + P)) \\ &= O(\text{GetS}(s, 1, j)(x) + \text{GetS}(s', 1, j)(P)), \because (1.2) \\ &= (\text{GetS}(s, 1, j)(x) + \text{GetS}(s', 1, j)(P))/2 \end{aligned}$$

$$\begin{aligned}
&= O(\text{GetS}(s, 1, j)(x)) + O(\text{GetS}(s', 1, j)(P)) \\
&= O(\text{GetS}(s, 1, j)(x)) + O(\text{GetS}(s', 1, j)(P)) \\
&= (\text{GetS}(s, 1, j) \| O)(x) + (\text{GetS}(s', 1, j) \| O)(P) \\
&= \text{GetS}(s, 1, j + 1)(x) + \text{GetS}(s', 1, j + 1)(P).
\end{aligned}$$

(Note that, here $j + 1 = 2$. Recall that “ $\|$ ” is concatenation.)

(3) Similarly, $j = 2$.

Due to (2), $\text{GetS}(s, 1, 2)(x + P) = \text{GetS}(s, 1, 2)(x) + \text{GetS}(s', 1, 2)(P)$.

Besides, $\text{GetS}(s', 1, 2)(P) \in [0]_2$. Thus,

$\text{IsEven}(\text{GetS}(s, 1, j)(x + P)) = \text{IsEven}(\text{GetS}(s, 1, j)(x))$. Thus, the intermediate next Collatz transformation of $x + P$ and x are identical.

(Note that, here $j = 2$).

Again, we can prove the following similar to (2.2).

$$\text{GetS}(s, 1, j + 1)(x + P) = \text{GetS}(s, 1, j + 1)(x) + \text{GetS}(s', 1, j + 1)(P).$$

(Note that, here $j + 1 = 3$.)

(4) Similarly, we can prove $j = 3, \dots, |s'| - 1$, respectively and especially in an order. \square

Remark 1.40.

(1) Obviously, above conclusion can be extended to include $|s| = 1$ by Lemma 1.35 and Lemma 1.37.

(2) Separation Lemma states the sufficient condition for guaranteeing that all intermediate parities of transformed integers $x + P$ are exactly identical with those of x .

(3) Separation Lemma is general, as s could be either original dynamics or reduced dynamics. Besides, the length of s can be omitted by using condition “ $j = 0, 1, 2, \dots$ ” instead of “ $j = 0, 1, 2, \dots, |s| - 1$ ”.

We next explore how to compute $\text{GetS}(s', 1, j)(x)$.

Definition 1.41. Function $\text{CntI}(\cdot)$. $\text{CntI} : s \rightarrow n$. It takes as input $s \in \{I, O\}^{\geq 1}$, and outputs $n \in \mathbb{N}$ that is the count of I in s .

Example 1.42. $\text{CntI}(IIOO) = 2$, $\text{CntI}(III) = 3$. Obviously, the function name stems from “Count the number of I ”.

Lemma 1.43. Suppose $s \in \{I, O\}^{\geq 1}$, $s' = \text{Replace}(s)$, $x \in \mathbb{N}^*$, we have

$$\text{GetS}(s', 1, j)(x) = \frac{3^{\text{CntI}(\text{GetS}(s, 1, j))}}{2^j} * x, \quad j = 1, 2, \dots, |s|.$$

Proof (1) $|s| = 1$. Thus, $j = 1$.

(1.1) If $s = I$, then $s' = Replace(s) = I'$.

$$GetS(s', 1, j)(x) = GetS(I', 1, 1)(x) = I'(x) = 3 * x/2 = 3^1 * x/2^1 \\ = 3^{CntI(I)} * x/2^{|I|} = 3^{CntI(GetS(s,1,j))} / 2^j * x.$$

(1.2) If $s = O$, then $s' = Replace(s) = O$.

$$GetS(s', 1, j)(x) = O(x) = x/2 = 3^0 * x/2^1 = 3^{CntI(O)} * x/2^{|O|} \\ = 3^{CntI(GetS(s,1,j))} / 2^j * x.$$

(2) $|s| \geq 2$.

(2.1) $j = 1$.

(2.1.1) If $GetS(s, 1, 1) = I$, then

$$GetS(s', 1, j)(x) = I'(x) = 3 * x/2 = 3^1 * x/2^1 = 3^{CntI(I)} * x/2^{|I|} = \\ 3^{CntI(GetS(s,1,j))} / 2^j * x.$$

(2.1.2) If $GetS(s, 1, 1) = O$, then

$$GetS(s', 1, j)(x) = O(x) = x/2 = 3^0 * x/2^1 = 3^{CntI(O)} * x/2^{|O|} \\ = 3^{CntI(GetS(s,1,j))} / 2^j * x.$$

(2.2) Iteratively, for $j = 1, 2, \dots, |s'| - 1$ in an order (recall that $|s'| = |s|$).

(2.2.1) If $GetS(s, 2, 1) = I$, then

$$GetS(s', 1, j+1)(x) = (GetS(s', 1, j) \| I')(x) \\ = I'(GetS(s', 1, j)(x)) = I'(GetS(s', 1, j)(x)) \\ = 3 * GetS(s', 1, j)(x)/2 \\ = 3 * \frac{3^{CntI(GetS(s,1,j))}}{2^j} * x/2 \quad \because (2.1) \text{ for } j = 1, (2.2) \text{ for } j = 2, \dots, |s'| - 1 \\ = \frac{3^{CntI(GetS(s,1,j))} + 1}{2^{j+1}} * x \\ = \frac{3^{CntI(GetS(s,1,j+1))}}{2^{j+1}} * x. \quad \because GetS(s, 1, j+1) = GetS(s, 1, j) \| I$$

(2.2.2) If $GetS(s, 2, 1) = O$, then

$$GetS(s', 1, j+1)(x) = (GetS(s', 1, j) \| O)(x) \\ = O(GetS(s', 1, j)(x)) = GetS(s', 1, j)(x)/2 \\ = \frac{3^{CntI(GetS(s,1,j))}}{2^j} * x/2 = \frac{3^{CntI(GetS(s,1,j))}}{2^{j+1}} * x \\ = \frac{3^{CntI(GetS(s,1,j+1))}}{2^{j+1}} * x. \quad \because GetS(s, 1, j+1) = GetS(s, 1, j) \| O \quad \square$$

Remark 1.44. Recall that $s' = GetS(s', 1, |s|)$ and $s = GetS(s, 1, |s|)$. Thus, when $j = |s|$, then $s'(x) = GetS(s', 1, |s|)(x) = \frac{3^{CntI(GetS(s,1,|s|))}}{2^{|s|}} * x = \frac{3^{CntI(s)}}{2^{|s|}} * x$.

Similar to Lemma 1.39 (Separation Lemma), a variant of Separation Lemma can be given as follows:

Lemma 1.45. Suppose $x \in \mathbb{N}^*$, $s \in \{I, O\}^{\geq 2}$, $s' = Replace(s)$. If $GetS(s', 1, j)(P) \in [0]_2$, $j = 0, 1, 2, \dots, |s| - 1$, then

- (1) $\text{IsEven}(\text{GetS}(s, 1, j)(x - P)) = \text{IsEven}(\text{GetS}(s, 1, j)(x));$
- (2) $\text{GetS}(s, 1, j + 1)(x - P) = \text{GetS}(s, 1, j + 1)(x) - \text{GetS}(s', 1, j + 1)(P).$

Proof The proof is similar to Lemma 1.39. \square

2. Partition Theorem

Notation 2.1. $\text{DYNM}(x, n) \in \{I, O\}^n$. It denotes the first $n \in \mathbb{N}^*$ transformations (in terms of I or O) of $x \in \mathbb{N}^*$ no matter whether $\text{RD}[x]$ exists or not.

For example, $\text{DYNM}(19, 2) = II$, $\text{DYNM}(19, 3) = IIO$, and $\text{DYNM}(19, 4) = IIOO$.

Interestingly, we observe facts as follows:

$\text{DYNM}(x \in [3]_8, 3) = IIO$. Let $s = IIO$.

$\forall x \in [3]_8$, if $s(x) \in [0]_2$, then $s(x + 8) \in [1]_2$; If $s(x) \in [1]_2$, then $s(x + 8) \in [0]_2$.

For example, $x = 3 \in [3]_8$, $s(3) = IIO(3) = IO(5) = O(8) = 4 \in [0]_2$, $s(3+8=11) = IIO(11) = IO(17) = O(26) = 13 \in [1]_2$. Moreover, $s(11+8=19) = IIO(19) = IO(29) = O(44) = 22 \in [0]_2$.

Remark 2.2. For better understanding our further proofs, a graph called reduced dynamics graph is proposed and explained in Appendix, which can visualize the following conclusions, but not mandatory for the proof. The details on tree-based dynamics graph is presented in another paper [5].

Next, we will prove above key observations.

Recall that by *Subset Theorem* (Theorem 1.27), if $\text{DYNM}(x \in r_1, n) = s$, $\text{DYNM}(x \in r_2, n + 1) = s\|I$ or $s\|O$, then $r_2 \subset r_1$.

Indeed, in this section we will prove that not only $r_2 \subset r_1$, but also r_2 is either half partition of r_1 . More specifically, the first distinction between transformations of x and transformations of $x + 2^t$ occurs at the $(t + 1)$ -th transformation (if $|\text{RD}[x]| > t + 1$). That is, $x \equiv x + 2^t \equiv i \pmod{2^t}$. $[i]_{2^{t+1}} \cup [i + 2^t]_{2^{t+1}} = [i]_{2^t}$, $[i]_{2^{t+1}} \cap [i + 2^t]_{2^{t+1}} = \emptyset$. If $x \in [i]_{2^{t+1}}$, then $x + 2^t \in [i + 2^t]_{2^{t+1}}$. If $x \in [i + 2^t]_{2^{t+1}}$, then $x + 2^t \in [i]_{2^{t+1}}$.

Lemma 2.3. If $\text{DYNM}(x, t) = s \in \{I, O\}^t$, $x, t \in \mathbb{N}^*$, $j = 1, 2, \dots, t - 1$, then

- (1) $t = 1$. $\text{IsEven}(s(x)) \neq \text{IsEven}(s(x + 2^t))$; Or,
- (2) $t \geq 2$. $\text{IsEven}(\text{GetS}(s, 1, j)(x)) = \text{IsEven}(\text{GetS}(s, 1, j)(x + 2^t))$, and $\text{IsEven}(s(x)) \neq \text{IsEven}(s(x + 2^t))$.

Proof (1) $t = 1$. $2^t = 2 \in [0]_2$. $s = I$ or O .

$$I(x+2) = (3(x+2)+1)/2 = (3x+1)/2 + 3 = I(x) + 3.$$

Thus, $IsEven(I(x+2)) \neq IsEven(I(x))$.

$$O(x+2) = (x+2)/2 = O(x) + 1. \text{ Thus, } IsEven(O(x+2)) \neq IsEven(O(x)).$$

Hence, $IsEven(s(x)) \neq IsEven(s(x+2^t))$.

(2) $t \geq 2$. Let $s' = Replace(s)$.

$$\text{By Lemma 1.43, } GetS(s', 1, j)(x) = \frac{3^{CntI(GetS(s, 1, j))}}{2^j} * x, \quad j = 1, 2, \dots, |s|.$$

$GetS(s', 1, j)(2^t) = \frac{3^{CntI(GetS(s, 1, j))}}{2^j} * 2^t = 3^{CntI(GetS(s, 1, j))} * 2^{t-j} \in [0]_2$, due to $j \leq t-1$ ($t-j \geq 1$).

Due to *Separation Lemma* (i.e., Lemma 1.39), we have

$$(1) \quad GetS(s', 1, j)(x+2^t) = GetS(s, 1, j)(x) + GetS(s', 1, j)(2^t), \text{ and}$$

$$(2) \quad IsEven(GetS(s, 1, j)(x+2^t)) = IsEven(GetS(s, 1, j)(x)), \text{ and}$$

$$(3) \quad s(x+2^t) = s(x) + s'(2^t).$$

$$\text{Due to Remark 1.44, } s'(2^t) = \frac{3^{CntI(s)}}{2^{|s|}} * 2^t = \frac{3^{CntI(s)}}{2^t} * 2^t = 3^{CntI(s')} \in [1]_2.$$

Thus, $IsEven(s(x+2^t)) \neq IsEven(s(x))$. \square

Remark 2.4. By assuming $GetS(\cdot, \cdot, 0)(x) = x$, above lemma can be restated as $IsEven(GetS(s, 1, j)(x)) = IsEven(GetS(s, 1, j)(x+2^t))$ and $IsEven(s(x)) \neq IsEven(s(x+2^t))$ where $j = 0, 1, \dots, t-1, t \geq 1$.

Above lemma states that the first t transformations of x and $x+2^t$ are identical. It can be extended to include $x-2^t$ upon $x-2^t > 0$. In other words, the first t transformations of x and $x \pm 2^t$ (i.e., $x \in [x \bmod 2^t]_{2^t}$) are identical. Or, the first t transformations of x is determined by $x \bmod 2^t$. Or, all $x \in [i]_{2^t}$ ($i = x \bmod 2^t$) have the same first t transformations.

Lemma 2.5. If $DYNM(x, t) = s \in \{I, O\}^t$, $x, t \in \mathbb{N}^*$, then

$DYNM(x \in [i]_{2^t}, t) = s$, where $i = x \bmod 2^t$.

Proof Let $j = 0, 1, 2, \dots, t-1$.

(1) By Lemma 2.3 and Remark 2.4,

$$IsEven(GetS(s, 1, j)(x)) = IsEven(GetS(s, 1, j)(x+2^t)).$$

Thus, $DYNM(x+2^t, t) = DYNM(x, t) = s$.

Similarly,

$$IsEven(GetS(s, 1, j)(x+2^t)) = IsEven(GetS(s, 1, j)(x+2^t+2^t)). \text{ Thus,}$$

$$DYNM(x+2^t+2^t, t) = DYNM(x+2^t, t) = s.$$

Iteratively and hence,

$$DYNM(x+m*2^t, t) = DYNM(x, t) = s, \quad m \in \mathbb{N}^*.$$

(2) Due to the variant of Separation Lemma (i.e., Lemma 1.45), $IsEven(GetS(s, 1, j)(x)) = IsEven(GetS(s, 1, j)(x - 2^t))$.

When $x - 2^t > 0$, we have

$$DYNM(x - 2^t, t) = DYNM(x, t) = s.$$

Iteratively, when $x - m * 2^t > 0, m \in \mathbb{N}^*$, we have

$$DYNM(x - m * 2^t, t) = DYNM(x, t) = s.$$

Therefore, due to (1) and (2),

$$DYNM(x \in [i]_{2^t}, t) = s, \text{ where } i = x \bmod 2^t. \quad \square$$

Remark 2.6.

(1) Above lemma includes the special case $i = 0$ or $x \in [0]_2$. However, $DYNM(x \in [0]_2, 1) = RD[x \in [0]_2] = O$ is trivial, $DYNM(x \in [1]_2, n \in \mathbb{N}^*)$ is thus of more interest. That is, we mainly concentrate on $DYNM(x \in [i]_{2^t}, n \in \mathbb{N}^*), i, t \in \mathbb{N}^*, 1 \leq i \leq 2^t - 1, i \in [1]_2$.

(2) In above lemmas (Lemma 2.3 and Lemma 2.5), s is general. That is, both reduced dynamics and original dynamics satisfy above lemmas.

(3) Recall that, Lemma 2.5 states that the first t transformations for $x \in [x \bmod 2^t]_{2^t}$ are identical. Lemma 2.3 states that the $(t + 1)$ -th transformation for x and $x + 2^t$ is distinct (so-called “forking”). Note that, it can be observed that either x or $x + 2^t$ falls in either partition of $[x \bmod 2^t]_{2^t}$ (i.e., $[x \bmod 2^t]_{2^{t+1}}, [(x \bmod 2^t) + 2^t]_{2^{t+1}}$), respectively. Moreover, all natural numbers in $x \in [x \bmod 2^t]_{2^t}$ are further partitioned into two halves, and either partition results in either transformation (i.e., “I” or “O”) due to “forking”, iteratively.

Similar to Lemma 2.3 and Lemma 2.5 but more specifically for whether “forking” transformation is “I” or “O” and why, we have following theorem. Roughly speaking, partition residue class $[i]_{2^t}$ determines the first t transformations. If current integer is less than the starting integer, then reduced dynamics is obtained. Otherwise, further transformation occurs, and whether it is “I” or “O” is determined by and only by either further half partition of the current residue class of x .

Theorem 2.7. (Partition Theorem.)

(1) $DYNM(x \in [0]_2, 1) = RD[x \in [0]_2] = O$.

(2) Suppose $DYNM(i, t) = s \in \{I, O\}^t, t, i \in \mathbb{N}^*, 1 \leq i \leq 2^t - 1$. We have

(2.1) $DYNM(x \in [i]_{2^t}, t) = s$.

(2.2) If $s(x) < x$, then $\text{RD}[x \in [i]_{2^t}] = s$.

(2.3) If $s(x) \not\prec x$ and $s(i) \in [0]_2$, then

$\text{DYNM}(x \in [i]_{2^{t+1}}, t+1) = s\|O$ and $\text{DYNM}(x \in [i+2^t]_{2^{t+1}}, t+1) = s\|I$;

If $s(x) \not\prec x$ and $s(i) \in [1]_2$, then

$\text{DYNM}(x \in [i]_{2^{t+1}}, t+1) = s\|I$ and $\text{DYNM}(x \in [i+2^t]_{2^{t+1}}, t+1) = s\|O$.

Proof (1) Straightforward.

(2.1) The proof is similar to Lemma 2.3 (2) (except that here $t = 1$ is possible), or by Lemma 2.5 (1).

(2.2) If $s(x) < x$, then reduced dynamics ends and $\text{RD}[x \in [i]_{2^t}] = \text{DYNM}(x \in [i]_{2^t}, t) = s$.

(2.3) We inspect the value of $\text{IsEven}(s'(m * 2^t))$ where $s' = \text{Replace}(s)$ to decide whether $\text{IsEven}(s(i + m * 2^t)) = \text{IsEven}(s(i))$ or not, since $s(i + m * 2^t) = s(i) + s'(m * 2^t)$, by Separation Lemma (i.e., Lemma 1.39).

By Remark 1.44 (recall that $t \in \mathbb{N}^*$),

$$s'(m * 2^t) = \frac{3^{C_{\text{nt}}I(s)}}{2^{\lfloor s \rfloor}} * m * 2^t = \frac{3^{C_{\text{nt}}I(s)}}{2^t} * m * 2^t = 3^{C_{\text{nt}}I(s)} * m.$$

$$m \in [1]_2 \Rightarrow 3^{C_{\text{nt}}I(s)} * m \in [1]_2 \Rightarrow \text{IsEven}(s(i + m * 2^t)) \neq \text{IsEven}(s(i)).$$

$$\text{Otherwise, } m \in [0]_2 \Rightarrow \text{IsEven}(s(i + m * 2^t)) = \text{IsEven}(s(i)).$$

$$(1) m \in [1]_2 \Leftrightarrow x = i + (m-1) * 2^t + 2^t = i + \frac{m-1}{2} * 2^{t+1} + 2^t \in [i+2^t]_{2^{t+1}}.$$

$$\text{Thus, } x \in [i+2^t]_{2^{t+1}} \Rightarrow \text{IsEven}(s(x)) \neq \text{IsEven}(s(i)).$$

$$(2) m \in [0]_2 \Leftrightarrow x = i + m * 2^t = i + \frac{m}{2} * 2^{t+1} \in [i]_{2^{t+1}} \setminus \{i\}.$$

(Recall that $[0]_2 = \{a | a \in \mathbb{N}^*, a \equiv 0 \pmod{2}\}$, thus $m \geq 2$.)

$$\text{Thus, } x \in [i]_{2^{t+1}} \setminus \{i\} \Rightarrow \text{IsEven}(s(x)) = \text{IsEven}(s(i)).$$

Obviously, $x = i$ can be included too, because

$x = i \Rightarrow \text{IsEven}(s(x)) = \text{IsEven}(s(i))$. Therefore,

$x \in [i]_{2^{t+1}} \Rightarrow \text{IsEven}(s(x)) = \text{IsEven}(s(i))$.

(3) Finally, if $s(i) \in [0]_2$, then $\text{DYNM}(x \in [i]_{2^{t+1}}, t+1) = s\|O$ due to (2), and $\text{DYNM}(x \in [i+2^t]_{2^{t+1}}, t+1) = s\|I$ due to (1);

If $s(i) \in [1]_2$, then $\text{DYNM}(x \in [i]_{2^{t+1}}, t+1) = s\|I$ due to (2), and $\text{DYNM}(x \in [i+2^t]_{2^{t+1}}, t+1) = s\|O$ due to (1).

Note that, $[i+2^t]_{2^{t+1}} \cap [i]_{2^{t+1}} = \emptyset$, and $[i+2^t]_{2^{t+1}} \cup [i]_{2^{t+1}} = [i]_{2^t}$. \square

Remark 2.8.

(1) We call the theorem as “Partition Theorem” because $[i]_{2^t}$ is partitioned into two halves for deciding the $(t+1)$ -th transformation. Recall that, $[i]_{2^t} = [i]_{2^{t+1}} \cup [i+2^t]_{2^{t+1}}$, $[i]_{2^{t+1}} \cap [i+2^t]_{2^{t+1}} = \emptyset$.

(2) The level of partitions (in terms of residue modules) is the power of 2. In other words, the partitions always have the form like $x \in [0]_2 \cup [i]_{2^t}, t \in \mathbb{N}^*, 1 \leq i \leq 2^t - 1, i \in [1]_2, i = x \pmod{2^t}$.

For $x \in [0]_2$, the further partition is not of interest because $\text{RD}[x \in [0]_2] = O$ is obtained.

For $x \in [i]_{2^t}$, the further partitions are $x \in [i]_{2^{t+1}}$ and $x \in [i + 2^t]_{2^{t+1}}$, as either partition results in either transformation in the next (recall Lemma 2.3). More specifically, to determine whether $s(x) \in [1]_2$ or not, $x \in [i]_{2^t}$ is partitioned into two halves - $x \in [i]_{2^{t+1}}$ and $x \in [i + 2^t]_{2^{t+1}}$. $[i]_{2^{t+1}} \cap [i + 2^t]_{2^{t+1}} = \emptyset$, $[i]_{2^{t+1}} \cup [i + 2^t]_{2^{t+1}} = [i]_{2^t}$. Therefore, they always have the form like $[i]_{2^t}, t \in \mathbb{N}^*, 1 \leq i \leq 2^t - 1$, and especially, $i \in [1]_2$.

(3) This theorem also reveals the link between $\text{RD}[x]$ and $\text{DYNM}(x, t)$ as follows: Suppose $\text{DYNM}(x, t) = s \in \{I, O\}^t$. Once $s(x) < x$, then $\text{DYNM}(x, t) = \text{RD}[x] = s$ (which means current transformed number has already been less than starting number for the first time); Otherwise, $\text{DYNM}(x, t + 1) = s\|I$ or $\text{DYNM}(x, t + 1) = s\|O$.

Next corollary states that $i \in [0, 2^t - 1]$ determines the first t transformations of all $x \in [i]_{2^t}$.

Corollary 2.9. Suppose $t, i_1, i_2 \in \mathbb{N}^*, 0 \leq i_1, i_2 \leq 2^t - 1$.

$i_1 = i_2$, if and only if $\text{DYNM}(x \in [i_1]_{2^t}, t) = \text{DYNM}(x \in [i_2]_{2^t}, t)$.

Proof (1) Necessity.

(1.1) $t \geq 2$, thus $1 \leq i_1, i_2 \leq 2^t - 1$ due to Theorem 2.7 (2) and Remark 2.8 (2).

$$i_1 = i_2 \Rightarrow \text{DYNM}(i_1, t) = \text{DYNM}(i_2, t)$$

$$\Rightarrow \text{DYNM}(x \in [i_1]_{2^t}, t) = \text{DYNM}(x \in [i_2]_{2^t}, t)$$

$$\because \text{DYNM}(i_1, t) = \text{DYNM}(x \in [i_1]_{2^t}, t), \text{DYNM}(i_2, t) = \text{DYNM}(x \in [i_2]_{2^t}, t)$$

$$(1.2) t = 1, \text{ thus } 0 \leq i_1, i_2 \leq 2^t - 1 = 1.$$

$$i_1 = i_2 = (0 \vee 1)$$

$$\Rightarrow \text{DYNM}(x \in [i_1]_{2^t}, t) = \text{DYNM}(x \in [i_1]_2, 1)$$

$$= \text{DYNM}(x \in [i_2]_2, 1) = \text{DYNM}(x \in [i_2]_{2^t}, t).$$

(2) Sufficiency. It is proved by proving converse negative proposition.

$$(2.1) t \geq 2. \text{ Thus } 1 \leq i_1, i_2 \leq 2^t - 1.$$

$$i_1 \neq i_2, 1 \leq i_1, i_2 \leq 2^t - 1 \Rightarrow \text{DYNM}(i_1, t) \neq \text{DYNM}(i_2, t)$$

$$\Rightarrow \text{DYNM}(x \in [i_1]_{2^t}, t) \neq \text{DYNM}(x \in [i_2]_{2^t}, t).$$

$$\because \text{DYNM}(x \in [i_1]_{2^t}, t) = \text{DYNM}(i_1, t), \text{DYNM}(x \in [i_2]_{2^t}, t) = \text{DYNM}(i_2, t).$$

(2.2) $t = 1$.
 $i_1 \neq i_2 \Rightarrow (i_1 = 0, i_2 = 1) \vee (i_1 = 1, i_2 = 0)$
 $\Rightarrow \text{DYNM}(x \in [i_1]_{2^t}, t) = \text{DYNM}(x \in [i_1]_2, 1)$
 $\neq \text{DYNM}(x \in [i_2]_2, 1) = \text{DYNM}(x \in [i_2]_{2^t}, t)$. \square

In above corollary, t is categorized for tackling the case $i = 0$ to avoid $\text{DYNM}(i = 0, t)$.

Corollary 2.10. Suppose $x_1, x_2, t \in \mathbb{N}^*, x_1 \neq x_2$.
 $\text{DYNM}(x_1, t) = \text{DYNM}(x_2, t)$, if and only if $x_1 \equiv x_2 \pmod{2^t}$.

Proof Straightforward due to Corollary 2.9. \square

Next corollary states that if the first different transformation of x_1, x_2 occurs at the $(t + 1)$ -th transformation, $t \in \mathbb{N}^*$, then $x_1 \equiv x_2 \pmod{2^t}$ and $x_1 \equiv x_2 + 2^t \pmod{2^{t+1}}$. As $\text{DYNM}(x \in [0]_2, 1) = \text{RD}[x \in [0]_2] = O$, only $x \in [1]_2$ is of interests.

Corollary 2.11. (Forking Corollary.) Suppose $x_1, x_2 \in [1]_2, x_1 \neq x_2, t \in \mathbb{N}^*$.

$\text{DYNM}(x_1, t) = \text{DYNM}(x_2, t)$ and $\text{DYNM}(x_1, t + 1) \neq \text{DYNM}(x_2, t + 1)$, if and only if $x_1 \equiv x_2 \pmod{2^t}$ and $x_1 \equiv x_2 + 2^t \pmod{2^{t+1}}$.

Proof It is straightforward due to Corollary 2.10. \square

Corollary 2.12. (Extended Forking Corollary.) Suppose

$\text{DYNM}(x \in [i_p]_{2^t}, t) = s_p \in \{I, O\}^t, p = 1, \dots, m$.
 $\text{DYNM}(x \in [j_q]_{2^{t+1}}, t + 1) = s_q \in \{I, O\}^{t+1}, q = 1, 2, \dots, n$.
 $t \geq 3, t \in \mathbb{N}^*. \forall p \in [1, m], \exists q \in [1, n] \text{ such that}$
 $(s_q = s_p \| I, s_{q+1} = s_p \| O) \vee (s_q = s_p \| O, s_{q+1} = s_p \| I)$.
We have following conclusions:

- (1) $\bigcup_{q=1}^n [j_q]_{2^{t+1}} = \bigcup_{p=1}^m [i_p]_{2^t}$. $i_p \in [1]_2$. $j_q \in [1]_2$. $n = 2 * m$.
 $s_p = \text{GetS}(s_q, 1, t) = \text{GetS}(s_{q+1}, 1, t)$, where $q = 2 * p - 1$.
- (2) $j_{2*p-1} = i_p, j_{2*p} = i_p + 2^t$.
- (3) $\text{IsEven}(s_p(x \in [i_p]_{2^{t+1}})) \neq \text{IsEven}(s_p(x \in [i_p + 2^t]_{2^{t+1}}))$.
- (4) $\text{DYNM}(x \in [i_p]_{2^{t+1}}, t + 1) = s_p \| I \Leftrightarrow \text{DYNM}(x \in [i_p + 2^t]_{2^{t+1}}, t + 1) = s_p \| O$.
- (5) $\text{DYNM}(x \in [i_p]_{2^{t+1}}, t + 1) = s_p \| O \Leftrightarrow \text{DYNM}(x \in [i_p + 2^t]_{2^{t+1}}, t + 1) = s_p \| I$.
- (6) $s_p(x \in [i_p]_{2^t}) \in [1]_2 \Leftrightarrow \text{DYNM}(x \in [i_p]_{2^{t+1}}, t + 1) = s_p \| I$.

$s_p(x \in [i_p]_{2^t}) \in [0]_2 \Leftrightarrow \text{DYNM}(x \in [i_p]_{2^{t+1}}, t+1) = s_p \| O.$

(7) $\|\{q|GetS(s_q, 1, t)(x \in [j_q]_{2^{t+1}}) \in [1]_2\}\| = n/2 = m;$

$\|\{q|GetS(s_q, 1, t)(x \in [j_q]_{2^{t+1}}) \in [0]_2\}\| = n/2 = m.$

$GetS(s_q, 1, t)(x \in [j_q]_{2^{t+1}}) \in [1]_2 \Leftrightarrow GetS(s_q, 1, t)(x \in [j_{q+1}]_{2^{t+1}}) \in [0]_2.$

$GetS(s_q, 1, t)(x \in [j_q]_{2^{t+1}}) \in [0]_2 \Leftrightarrow GetS(s_q, 1, t)(x \in [j_{q+1}]_{2^{t+1}}) \in [1]_2.$

Proof By Theorem 2.7 (Partition Theorem),

if $s_p(x \in [i_p]_{2^t}) \in [1]_2, p \in [1, m]$, then $\exists q = 2 * p - 1 \in [1, n]$, such that $\text{DYNM}(x \in [j_q]_{2^{t+1}}, t+1) = s_p \| I = s_q$. Besides, $j_q = i_p$ due to $IsMatched(GetS(s_q, 1, t)(i_p) \in [1]_2, GetS(s_q, t+1, 1) = I) = \text{True}$.

If $s_p(x \in [i_p]_{2^t}) \in [0]_2, p \in [1, m]$, then $\exists q = 2 * p - 1 \in [1, n]$, such that $\text{DYNM}(x \in [j_q]_{2^{t+1}}, t+1) = s_p \| O = s_q$. Besides, $j_q = i_p$ due to $IsMatched(GetS(s_q, 1, t)(i_p) \in [0]_2, GetS(s_q, t+1, 1) = O) = \text{True}$.

If $s_p(x \in [i_p]_{2^t}) \in [0]_2, p \in [1, m]$, then $\exists q = 2 * p \in [1, n]$, such that $\text{DYNM}(x \in [j_q]_{2^{t+1}}, t+1) = s_p \| I = s_q$. Besides, $j_q = i_p + 2^t$ due to $IsMatched(GetS(s_q, 1, t)(j_q = i_p + 2^t) \in [1]_2, GetS(s_q, t+1, 1) = I) = \text{True}$.

If $s_p(x \in [i_p]_{2^t}) \in [1]_2, p \in [1, m]$, then $\exists q = 2 * p \in [1, n]$, such that $\text{DYNM}(x \in [j_q]_{2^{t+1}}, t+1) = s_p \| O = s_q$. Besides, $j_q = i_p + 2^t$ due to $IsMatched(GetS(s_q, 1, t)(j_q = i_p + 2^t) \in [0]_2, GetS(s_q, t+1, 1) = O) = \text{True}$.

Obviously, $s_p = GetS(s_q, 1, t)$, as $s_p \| \{I, O\} = s_q$ where $q = 2 * p - 1$ or $2 * p$ for $p = 1, 2, \dots, m$.

$t \geq 3$, thus $i_p \in [1]_2, p = 1, 2, \dots, m$. $j_q = i_p \in [1]_2$ where $q = 2 * p - 1$, and $j_q = i_p + 2^t \in [1]_2$ where $q = 2 * p$.

$p = 1, 2, \dots, m$, thus $q = 1, 2, \dots, n$ and $n = 2 * m$.

Besides, $\forall p \in [1, m], [i_p]_{2^t} = [i_p]_{2^{t+1}} \cup [i_p + 2^t]_{2^{t+1}}$, $[i_p]_{2^{t+1}} = [j_q]_{2^{t+1}}$ where $q = 2 * p - 1$, $[i_p + 2^t]_{2^{t+1}} = [j_q]_{2^{t+1}}$ where $q = 2 * p$. Therefore, $\bigcup_{q=1}^n [j_q]_{2^{t+1}} = \bigcup_{p=1}^m [i_p]_{2^t}$.

In summary, (1) is proved.

Besides, $j_{2*p-1} = i_p, j_{2*p} = i_p + 2^t$ where $p = 1, 2, \dots, m$.

$2 * p - 1 = q$, thus $j_q = i_p = i_{(q+1)/2}$ where $q = 1, 3, \dots, 2 * m - 1$.

$j_{q+1} = j_{2*p} = i_p + 2^t = j_q + 2^t$ where $q = 1, 3, \dots, 2 * m - 1$.

Thus, (2) is proved.

(3),(4),(5) are due to Lemma 2.3.

(6) is due to Partition Theorem, and the definition of $\text{DYNM}(\cdot, \cdot)$.

(7) is due to (2). \square

Roughly speaking, above corollary states the general (iteratively and accumulatively) effect of Forking Corollary (Corollary 2.11).

Lemma 2.13. *Given $\forall n \in \mathbb{N}^*$, $\exists x \in \mathbb{N}^*$ such that $\text{DYNM}(x, n) = I^n$.*

Proof Straightforward. Given $n \in \mathbb{N}^*$, $\exists x = 2^n - 1 \in \mathbb{N}^*$, such that $\text{DYNM}(x, n) = I^n$. Indeed, given $\forall n \in \mathbb{N}^*$, $\exists x \in [2^n - 1]_{2^n}$ such that $\text{DYNM}(x, n) = I^n$. \square

Lemma 2.14. *$\forall n \in \mathbb{N}^*$, $\exists x = 2^n - 1 \in \mathbb{N}^*$ such that $\text{DYNM}(x, n)(x) > x$.*

Proof Straightforward.

$$\begin{aligned}
 \text{DYNM}(x, n)(x) &= \text{DYNM}(2^n - 1, n)(x) = I^n(x) \\
 &= (3(\dots(3(3x + 1)/2) + 1)/2\dots) + 1)/2 \\
 &= \frac{3}{2}(\frac{3}{2}(\dots\frac{3}{2}(\frac{3}{2}x + \frac{1}{2}) + \frac{1}{2}) + \dots + \frac{1}{2}) + \frac{1}{2} \\
 &= (\frac{3}{2})^n x + \frac{1}{2}((\frac{3}{2})^{n-1} + (\frac{3}{2})^{n-2} + \dots + 1) \\
 &= (\frac{3}{2})^n x + \frac{1}{2}(\frac{(\frac{3}{2})^n - 1}{\frac{3}{2} - 1}) = (\frac{3}{2})^n x + (\frac{3}{2})^n - 1 = (\frac{3}{2})^n(x + 1) - 1 \\
 &= (\frac{3}{2})^n * x + (\frac{3}{2})^n - 1 > (\frac{3}{2})^n * x > x.
 \end{aligned}$$

\square

Lemma 2.15. *$\forall n \in \mathbb{N}^*$, $\exists x = 2^n - 1 \in \mathbb{N}^*$, if $\text{RD}[x]$ exists, then $|\text{RD}[x]| > n$.*

Proof Straightforward. $\text{DYNM}(x, n)(x) = \text{DYNM}(2^n - 1, n)(x) = I^n(x) > x$. If $\text{RD}[x]$ exists, then $|\text{RD}[x]| \geq n + 1 > n$. \square

By above lemmas, we can draw a conclusion that maximal length of reduced dynamics is nonexistent. Hence, it is *impossible* to enumerate all reduced dynamics. Indeed, original dynamics is also held for above two conclusions.

Theorem 2.16. *If Collatz conjecture is true, then the maximal count of Collatz transformations for all positive integers to return to 1 is nonexistent.*

Proof Collatz conjecture is true, hence $\forall x \in \mathbb{N}^*$, there exists original dynamics of x . Let $S = \{n | \forall x \in \mathbb{N}^*, \text{DYNM}(x, n)(x) = 1\}$. That is, S is a set of all $x \in \mathbb{N}^*$ that can return to 1 after finite times of Collatz transformations (i.e., n , in terms of $(3 * x + 1)/2$ denoted as I and $x/2$ denoted as O). In other words, $\forall x \in S, \exists n \in \mathbb{N}^*$ such that $\text{DYNM}(x, n)(x) = 1$.

Next, we will prove that $\max(S)$ cannot exist.

Suppose $\max(S)$ exists. We next construct a contradiction as follows: Let $N_{\max} = \max(S)$. We can create $y = 2^{N_{\max}} - 1 \in \mathbb{N}^*$. As Collatz conjecture is true, y can return to 1 after finite times of Collatz transformations. Thus, $y \in S$. However, y needs at least $N_{\max} + 1$ times of transformations to return to 1 due to $\text{DYNM}(y, N_{\max})(y) = I^{N_{\max}}(y) > y$ (i.e., $\text{DYNM}(y, N_{\max}) = I^{N_{\max}}$), which contradicts with $N_{\max} = \max(S)$. \square

Corollary 2.17. *If Collatz conjecture is true, the set of all $x \in \mathbb{N}^*$ that can return to 1 after finite times of Collatz transformations cannot be enumerated.*

Proof Straightforward due to Theorem 2.16.

Suppose S is a set of $x \in \mathbb{N}^*$ that can return to 1 after finite times of Collatz transformations in terms of $(3*x + 1)/2$ (denoted as I) and $x/2$ (denoted as O). Initially, S is empty. If $\forall x \in S, \exists n \in \mathbb{N}^*$ such that $\text{DYNM}(x, n)(x) = 1$, then let x be included into S .

Let $N_{max} = \max(\{n | \forall x \in S, \text{DYNM}(x, n)(x) = 1\})$. We can create $y = 2^{N_{max}} - 1$. As Collatz conjecture is true, y can return to 1 after finite times of Collatz transformations. $y \notin S$, because y needs at least $N_{max} + 1$ times of transformations to return to 1 due to $\text{DYNM}(y, N_{max})(y) = I^{N_{max}}(y) > y$ (i.e., $\text{DYNM}(y, N_{max}) = I^{N_{max}}$). That is, the set S cannot be enumerated in finite times, there always exists a positive integer that is not in current S . \square

Next, we will provide an approach to prove Reduced Collatz Conjecture.

Definition 2.18. *Function $CntO(\cdot)$. $CntO : s \rightarrow n$ takes as input $s \in \{I, O\}^{\geq 1}$, and outputs $n \in \mathbb{N}^*$ that is the count of “O” in s .*

Example 2.19. $CntO(IIOO) = 2$. Obviously, the function name comes from “counting the number of O”.

We can create an one-to-one mapping between residue class and reduced dynamics. (Indeed, in our another paper [9], the formula to compute residue class for a given reduced dynamics is derived. Sufficient and necessary condition for reduced dynamics is proved. The period due to residue class is extensively explored and proved in our another paper [8]. Those major properties for reduced dynamics such as partition, period, and ratio can be showed visually in our proposed graph [5], which is provided in Appendix).

A general algorithm can be proposed by Theorem 2.7, which outputs residue class of x , and takes as input first t transformations in (partial) reduced dynamics of x (i.e., $s = \text{DYNM}(x, t)$, $|\text{RD}[x]| \geq t$).

Given specific $s \in \{I, O\}^t$, residue class $[i]_{2^t}$ can be determined by Algorithm 1 (called D2R Algorithm) such that $\text{DYNM}(x \in [i]_{2^t}, t) = s$ as follows:

Remark 2.20.

(1) If $\text{IsMatched}(\text{GetS}(s, 1, j)(i), \text{GetS}(s, j+1, 1)) = \text{False}$, then $i \Leftarrow 2^j + i$. It is due to Theorem 2.7.

```

Data:  $s$ 
Result:  $i$ 
 $i \Leftarrow 1;$ 
for  $j = 1; j \leq t - 1; j++$  do
  if  $IsMatched(GetS(s, 1, j)(i), GetS(s, j + 1, 1)) == False$  then
     $i \Leftarrow 2^j + i;$ 
  end
end
return  $[i]_{2^t};$ 

```

Algorithm 1: D2R Algorithm. Input $s \in \{I, O\}^t, t \in \mathbb{N}^*, t \geq 2, CntO(GetS(s, 1, j)) < \lceil \log_2 1.5 * CntI(GetS(s, 1, j)) \rceil$ where $j = 1, 2, \dots, t-1, CntO(s) \leq \lceil \log_2 1.5 * CntI(s) \rceil$. Output $[i]_{2^t}, 1 \leq i \leq 2^t - 1, i \in [1]_2$, such that $DYNM(x \in [i]_{2^t}, t) = s$.

(2) Indeed, i in outputs $[i]_{2^t}$ is the major computation result, as t is available in the input s due to $t = |s|$.

(3) Obviously, above algorithm can be terminated and the time cost is $O(t)$.

(4) Besides, $t = 1$ is trivial and can be easily included in the algorithm as follows: If input is I , then let the output of D2R algorithm be $[1]_{2^1}$. If input is O , then let the output be $[0]_{2^1}$. Consequently, Corollary 2.21 can be extended to include $t = 1$ easily.

Corollary 2.21. (1) Given $s \in \{I, O\}^t, t \in \mathbb{N}^*, t \geq 2, CntO(GetS(s, 1, j)) < \lceil \log_2 1.5 * CntI(GetS(s, 1, j)) \rceil, j = 1, 2, \dots, t - 1, CntO(s) \leq \lceil \log_2 1.5 * CntI(s) \rceil$, there exists one and only one $i \in [1, 2^t - 1], i \in [1]_2$ such that $DYNM(x \in [i]_{2^t}, t) = s$.

(2) Inversely, given $x \in [i]_{2^t}, t \in \mathbb{N}^*, t \geq 2, i \in [1, 2^t - 1], i \in [1]_2$, there exists one and only one $s \in \{I, O\}^t$ such that

$DYNM(x, t) = s$,

$CntO(GetS(s, 1, j)) < \lceil \log_2 1.5 * CntI(GetS(s, 1, j)) \rceil, j = 1, 2, \dots, t - 1, CntO(s) \leq \lceil \log_2 1.5 * CntI(s) \rceil$.

(3) If $N_1 = \|\{s | s \in \{I, O\}^t, t \in \mathbb{N}^*, t \geq 2, i \in [1, 2^t - 1], i \in [1]_2, CntO(GetS(s, 1, j)) < \lceil \log_2 1.5 * CntI(GetS(s, 1, j)) \rceil, j = 1, 2, \dots, t - 1, CntO(s) \leq \lceil \log_2 1.5 * CntI(s) \rceil, DYNM(x \in [i]_{2^t}, t) = s\}\|$, and $N_2 = \|\{i | s \in \{I, O\}^t, t \in \mathbb{N}^*, t \geq 2, i \in [1, 2^t - 1], i \in [1]_2\}$,

$CntO(GetS(s, 1, j)) < \lceil \log_2 1.5 * CntI(GetS(s, 1, j)) \rceil$, $j = 1, 2, \dots, t - 1$,
 $CntO(s) \leq \lceil \log_2 1.5 * CntI(s) \rceil$, $DYNM(x \in [i]_{2^t}, t) = s \rceil \rceil$,
then $N_1 = N_2$.

Proof (1) $[i]_{2^t} \Rightarrow s$. It is due to the computation for the first t transformations of i , which is deterministic, and all $x \in [i]_{2^t}$ have the same first t transformations due to Lemma 2.5.

(2) $s \Rightarrow [i]_{2^t}$. As all transformations of $[j]_{2^t}$, $j = 1, \dots, 2^t - 1$, $j \in [1]_2$ can be enumerated, one and only one of them equals s due to Lemma 2.5. That j is i . Alternatively, non-trivial algorithm outputting i for given s is proposed in Algorithm 1.

(3) Due to (1), we have $N_1 = N_2$. (Recall Corollary 2.9 and Corollary 2.10.) \square

Remark 2.22.

(1) Simply speaking, above corollary states that s and $[i]_{2^t}$ can be mutually determined such that $DYNM(x \in [i]_{2^t}, t) = s$, and thus the number of types are identical.

(2) For better understanding above corollary, we can explain or observe above N_1 and N_2 in our proposed reduced dynamics graph in Appendix. That is, N_1 is the count of paths consisting of either “I” or “O” with length t from starting integer; N_2 is the count of distinct residue classes whose first t transformations equal these paths.

Proposition 2.23. (An approach to prove Reduced Collatz Conjecture.) Given $n \in \mathbb{N}^*$, the number of integers x ($x \in \mathbb{N}^*$) such that $|\mathbf{RD}[x]| \leq n$ is infinite due to periodical property of x . We denote this set as $S(x, n) = \{x(n) | n, x \in \mathbb{N}^*, |\mathbf{RD}[x]| \leq n\}$. The ratio of the number of this set over the number of all positive integers is $R(x, n) = \frac{\|S(x, n)\|}{\|\mathbb{N}^*\|}$. It is finite and obviously $R(x, n) \in (0, 1]$. E.g., $S(x, 1) = [0]_2$; $S(x, 2) = \{x(2) | |\mathbf{RD}[x]| \leq 2\} = \{x(2) | |\mathbf{RD}[x]| = 1, |\mathbf{RD}[x]| = 2\} = [0]_2 \cup [1]_4$; $R(x, 1) = \frac{\|[0]_2\|}{\|\mathbb{N}^*\|} = 1/2$; $R(x, 2) = \frac{\|[0]_2 \cup [1]_4\|}{\|\mathbb{N}^*\|} = 1/2 + 1/4 = 3/4$.

We hereby state that, if $R(x, i)$ goes to 1 with the growth of n to infinite, then Reduced Collatz Conjecture is true (and thus Collatz Conjecture is true). That is, if

$$\lim_{n \rightarrow +\infty} R(x, n) = 1,$$

where $R(x, n) = \frac{\|S(x, n)\|}{\|\mathbb{N}^*\|}$, then Reduced Collatz Conjecture is true and thus Collatz Conjecture is true.

Proof Straightforward. If $\lim_{n \rightarrow +\infty} R(x, n) = 1$, then $\|S(x, n)\| = \|\mathbb{N}^*\|$, when $n \rightarrow +\infty$. Thus, $\forall x \in \mathbb{N}^*, \exists \text{RD}[x]$. \square

3. Conclusion

This paper discovered and proved that, all positive integers, especially odd numbers, are partitioned regularly. If the first different $f \in \{I, O\}$ transformation of x_1, x_2 occurs at the $(t+1)$ -th transformation, $t \in \mathbb{N}^*$, then $x_1 \equiv x_2 \pmod{2^t}$ and $x_1 \equiv x_2 + 2^t \pmod{2^{t+1}}$. The first t transformations for all $x \in [x \pmod{2^t}]_{2^t}$ are identical. The $(t+1)$ -th transformation for x and $x + 2^t$ is distinct. $\forall x \in [1]_2$, if $s(x) \not\propto x$, $s = \text{DYNM}(x, t)$, $t \geq 2$, then $[i]_{2^t}, i \in [1]_2$ is partitioned into two halves and either half presents I or O in the $(t+1)$ -th transformation. Otherwise, reduced dynamics is obtained, i.e., $\text{DYNM}(x, t) = \text{RD}[x]$.

Acknowledgement

The research was financially supported by the Provincial Key Research and Development Program of Hubei (No. 2020BAB105), the Foundation of National Natural Science Foundation of China (No. 61972366), Open Research Project of The Hubei Key Laboratory of Intelligent Geo-Information Processing (KLIGIP-2021B06), and Knowledge Innovation Program of Wuhan-Basic Research (No. 2022010801010197).

References

- [1] Tomas Oliveira e Silva, *Maximum excursion and stopping time record-holders for the 3x+1 problem: computational results*, *Mathematics of Computation*, vol. 68, no. 225, pp. 371-384, 1999.
- [2] Tomas Oliveira e Silva, *Empirical Verification of the 3x+1 and Related Conjectures*. In *The Ultimate Challenge: The 3x+1 Problem*, (book edited by Jeffrey C. Lagarias), pp. 189-207, AMS, 2010.
- [3] Wei Ren, Simin Li, Ruiyang Xiao and Wei Bi, *Collatz Conjecture for $2^{100000} - 1$ is True - Algorithms for Verifying Extremely Large Numbers*, Proc. of IEEE UIC, Oct. 2018, Guangzhou, China, 411-416, 2018

- [4] Wei Ren, *A New Approach on Proving Collatz Conjecture*, Journal of Mathsmatics, Hindawi, April 2019, ID 6129836, <https://www.hindawi.com/journals/jmath/2019/6129836/>.
- [5] Wei Ren, *Ratio and Partition are Revealed in Proposed Graph on Reduced Collatz Dynamics*, Proc. of IEEE ISPA, pp. 474-483, 16-28 Dec. 2019, Ximen, China
- [6] Wei Ren, Ruiyang Xiao, *How to Fast Verify Collatz Conjecture by Automata*, Proc. of IEEE HPCC, pp. 2720-2729, 10-12 Aug. 2019, Zhangjiajie, China
- [7] Wei Ren, *Reduced Collatz Dynamics Data Reveals Properties for the Future Proof of Collatz Conjecture*, Data, MDPI, 2019, 4, 89.
- [8] Wei Ren, *Reduced Collatz Dynamics is Periodical and the Period Equals 2 to the Power of the Count of $x/2$* , Computational and Mathematical Methods, Wiley and Hindawi, Submitted, 2022.
- [9] Wei Ren, *A Reduced Collatz Dynamics Maps to a Residue Class, and its Count of $x/2$ over Count of $3*x+1$ is larger than $\ln 3/\ln 2$* , International Journal of Mathematics and Mathematical Sciences, Hindawi, Volume2020, Article ID 5946759, <https://doi.org/10.1155/2020/5946759>, 2020.

Conflicts of Interest

The author has no conflicts of interest to declare.

Appendix: Dynamics Graph with Partition Labels

Reduced dynamics graph [5] with partition labels can be obtained by adding each branch of paths a partition of $x \in \mathbb{N}^*$ (see Fig.1), which provides a visualization and smooth understanding for our proofs of Partition Theorem.

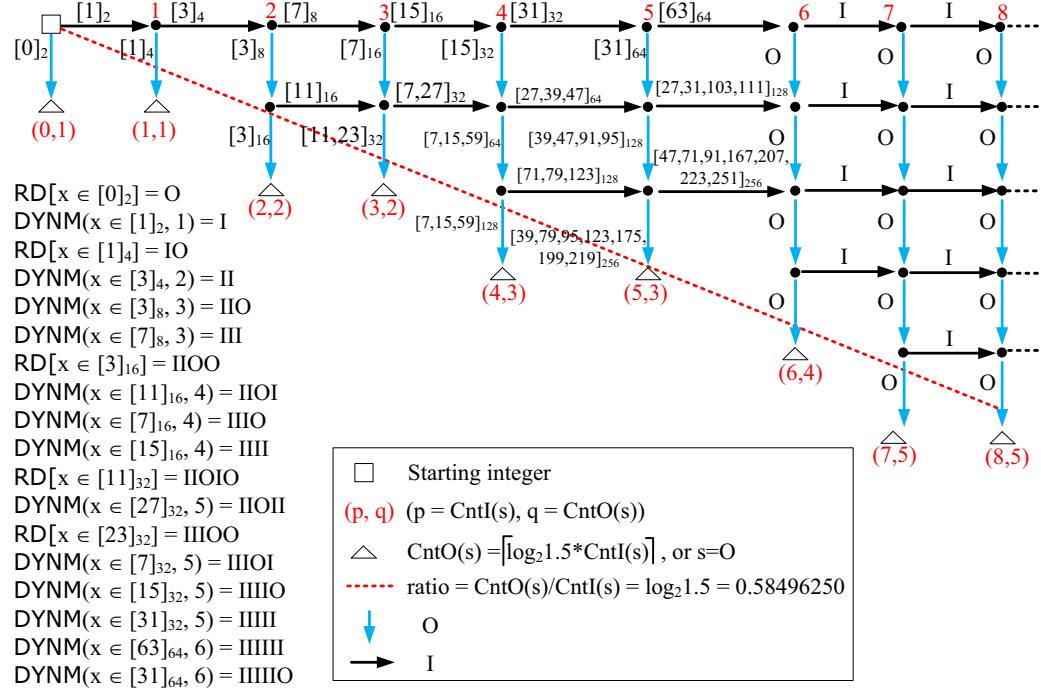


Figure 1: Reduced dynamics graph with partition labels. “□” represents starting integer. “△” represents transformed integer that is less than starting integer. Any “△” is below ratio line, whose slope is $\lambda = \log_2 1.5$. A reduced dynamics is a path consisting of edges in terms of “I” or “O” from “□” to “△”. If and only if $|s| = 1, s = O$ or $\text{CntO}(s) = \lceil \lambda * \text{CntI}(s) \rceil$ and $\text{CntO}(s') < \lceil \lambda * \text{CntI}(s') \rceil, s' = \text{GetS}(s, 1, j), j = 1, 2, \dots, |s| - 1, |s| \geq 2$, then s is a reduced dynamics, i.e., $s(x) < x$, $\text{GetS}(s, 1, j)(x) \not< x, j = 1, 2, \dots, |s| - 1$. All reduced dynamics intersect the ratio line in the last edge, which means transformed integer is the first one that is less than corresponding starting integer.

Remark 3.1.

(1) Either partition of current x presents the same previous dynamics, but will present either edge (I or O) in the next transformation, iteratively.

More specifically, either half of $x \in [i]_{2^t}$ (namely, $x \in [i]_{2^{t+1}}$ or $x \in [i + 2^t]_{2^{t+1}}$) has either $\text{DYNM}(x, t+1) = \text{DYNM}(x, t)\|I$ or $\text{DYNM}(x, t+1) = \text{DYNM}(x, t)\|O$. Besides, $\forall x \in [i]_{2^t}$, $\text{DYNM}(x, t)$ are identical.

For example, $\text{DYNM}(x \in [3]_8, 3) = IIO$. Whether the next transformation is I or O depends on either half partition of $x \in [3]_8$. That is, if $x \in [3]_{16}$, then $IIO(x) \in [0]_2$ and the next one is O ; If $x \in [11]_{16}$, then $IIO(x) \in [1]_2$ and the next one is I . Obviously, $[3]_8 = [3]_{16} \cup [11]_{16}$ and $[3]_{16} \cap [11]_{16} = \emptyset$.

(2) Let $\text{DYNM}(x, L) = s \in \{I, O\}^L, L \in \mathbb{N}^*$. If $\text{CntO}(s) = \lceil \lambda * \text{CntI}(s) \rceil$ (or $s = O$), then final transformed integers that are denoted as “ \triangle ” is available (i.e., $\text{DYNM}(x, L) = \text{RD}[x]$). In other words, $s(x) < x$, $\text{GetS}(s, 1, j)(x) \not\propto x, j = 1, 2, \dots, L-1$).

(3) Each edge (namely, I or O) is labeled with one residue class or a union of multiple residue classes, whose last transformation is this edge (I or O), and x in this residue class presents the dynamics represented by the path from “ \square ” (representing starting integer) to this edge.

For example, $[11, 23]_{32}$ is labeled due to $\text{DYNM}(x \in [11, 23]_{32}, 5) = s_{11}, s_{23} \in \{I, O\}^5$ where $\text{GetS}(s_{11}, 5, 1) = O, \text{GetS}(s_{23}, 5, 1) = O$. More specifically, $\text{DYNM}(x \in [11]_{32}, 5) = s_{11} = IIOIO, \text{DYNM}(x \in [23]_{32}, 5) = s_{23} = IIIOO$.

More specifically, if $[i]_{2^t}$ (or $[i_1, i_2, \dots, i_p]_{2^t}, p \in \mathbb{N}^*$) is (are) residue class (classes) as a label located at the last edge of path $s_i \in \{I, O\}^t$ (or paths $s_{i_1}, s_{i_2}, \dots, s_{i_p}$), then $\text{DYNM}(x \in [i]_{2^t}, t) = s_i$ (or $\text{DYNM}(x \in [i_j]_{2^t}, t) = s_{i_j}, j = 1, 2, \dots, p$).

For reduced dynamics, the ratio - the count of $x/2$ (or “ O ”) over the count of $(3*x+1)/2$ (or “ I ”) - is larger than $\log_2 1.5$ (denoted as λ). That is, when and only when the count of “ O ” is larger than the count of “ I ” times λ , current transformed integer will be less than the starting integer (i.e., the reduced dynamics will be available). Indeed, we prove following sufficient and necessary condition in which any $s \in \{I, O\}^{\geq 1}$ is a reduced dynamics formally in another paper [9].

Corollary 3.2. (Form Corollary.) $s \in \{I, O\}^{\geq 1}$ is a reduced dynamics, if and only if

- (1) $|s| = 1, s = O$; Or,
- (2) $|s| \geq 2$,

$$\begin{cases} \text{CntO}(s) = \lceil \lambda * \text{CntI}(s) \rceil, \\ \text{CntO}(s') < \lceil \lambda * \text{CntI}(s') \rceil, \quad s' = \text{GetS}(s, 1, i), i = 1, 2, \dots, |s| - 1. \end{cases} \quad (1)$$