2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

Ratio and Partition are Revealed in Proposed Graph on Reduced Collatz Dynamics

Wei Ren
School of Computer Science, China University of Geosciences, Wuhan, China
Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guizhou, China
weirencs@cug.edu.cn

***Abstract*—Collatz conjecture is also known as 3X+1 problem. The original dynamics is occurred 3x+1 and x/2 during the process from any given starting integer to 1. We but propose to study reduced dynamics, which are occurred (3*x+1)/2 and x/2 during the process from a starting integer to the first integer less than the starting integer, because reduced dynamics is the building blocks of original dynamics and x/2 always follows 3*x+1. We thus propose a graph (directed binary tree) to represent all possible reduced dynamics, in which (3*x+1)/2 is denoted as "I" with right directed edge and x/2 is denoted as "O" with down directed edge. The graph can show some inner laws in reduced dynamics directly and visually as follows: (1) The ratio, which is the count of x/2 over the count of (3*x+1)/2, is always larger than a constant value, namely, ln1.5/ln2. (2) The regular partition of integers whose reduced dynamics equals a path consisting of "I" and/or "O" can also be observed. That is, all integers are partitioned regularly in the graph. Given any positive integer x that is i module 2^t (i is an odd integer), the first t computations in terms of "I" or "O" can be determined. If current x, which is computed after t (t is greater or equal to 2) computations of "I" or "O", is less than x, then reduced dynamics is obtained or available. Otherwise, the residue class of x (namely, i module 2^t) is partitioned further 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 the forthcoming (t+1)-th computation. We finally propose an algorithm that takes as input reduced dynamics and outputs a residue class who presents this reduced dynamics.**

***Index Terms*—Collatz Conjecture, 3x+1 Problem, Graph, Residue Class, Algorithm, Discrete Dynamics Systems**

I. Introduction

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. W. 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. W. 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.

Reduced dynamics thus will be the focus for the proof of Collatz conjecture. We propose to create a graph to represent reduced dynamics, which can reveal some inner properties in reduced dynamics, directly and visually. Those observations from the graph certainly are also formally and mathematically proved in our other papers (e.g., ratio [6] and partition [5]), which are sketched in Appendix for interested readers.

Supported by Major Scientific and Technological Special Project of Guizhou Province (20183001), and the Open Funding of Guizhou Provincial Key Laboratory of Public Big Data (2018BDKFJJ009,2017BDKFJJ006)

II. Preliminaries

A. Notation

Some major notations used in the paper are listed as follows:

1) $\mathbb{Z}$: Integer.
2) $\mathbb{N} = \{a|a \in \mathbb{Z}, a \geq 0\}$.
3) $\mathbb{N}^* = \mathbb{N}\backslash\{0\} = \{a|a \in \mathbb{Z}, a \geq 1\}$.
4) $[1]_2 = \{x|x \equiv 1 \mod 2, x \in \mathbb{N}^*\}$; $[0]_2 = \{x|x \equiv 0 \mod 2, x \in \mathbb{N}^*\}$.
5) $[i]_m = \{x|x \equiv i \mod m, x \in \mathbb{N}^*, m \geq 2, m \in \mathbb{N}^*, 0 \leq i \leq m-1, i \in \mathbb{N}\}$.
6) $f(\cdot)$: Collatz transformation, $f(x) = 3*x+1$ or $f(x) = x/2$. $f^0(x) = x$. $f^n(x) = f(f^{n-1}(x)), n \in \mathbb{N}^*$.
7) $I(\cdot)$: $I(x) = (3*x+1)/2$, $I^0(x) = x, I^n(x) = I^{n-1}(I(x))$, $n \in \mathbb{N}^*$. It can be written as I in short.
8) $O(\cdot)$: $O(x) = x/2$, $O^0(x) = x$, $O^n(x) = O^{n-1}(O(x))$, $n \in \mathbb{N}^*$. It can be written as O in short.
9) Reduced Dynamics of starting number x: a sequence of occurred "I" and "O" during the process from x to the first transformed number that is less than x.
10) $\mathsf{RD}[x]$: reduced dynamics of x.
11) $Set_c = \{c|x \in \mathbb{N}^*, \exists \mathsf{RD}[x], c = \mathsf{RD}[x] \in \{I,O\}^{\geq 1}\}$.
12) $|\{I,O\}^L|$: the length of $\{I,O\}^L$.
13) $A\|B$: concatenation of $A \in \{I,O\}^{\geq a}$ and $B \in \{I,O\}^{\geq b}$. $a \in \mathbb{N}^*, b \in \mathbb{N}$, or $a \in \mathbb{N}, b \in \mathbb{N}^*$. $A\|\{I,O\}^0 = A$, $\{I,O\}^0\|B = B$.
14) $\|S = \{...\}\|$ returns the number of elements in a set S.
15) $\lambda = \log_2 1.5$.

978-1-7281-4328-6/19/$31.00 ©2019 IEEE
DOI 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00077

16) $\min(S = \{...\})$: The minimal element in a set S.
17) $[i_1, i_2, ..., i_n]_m = \bigcup_{j=1}^{n} [i_j]_m$.
18) $\mathsf{DYNM}(x, n)$: The first n transformations in terms of $\{I, O\}$ for x.
19) $Match(\cdot, \cdot)$: a function returns true or false that indicates whether the parity of an inputting integer matches an inputting transformation in terms of "I" or "O".
20) $Substr(\cdot, \cdot, \cdot)$: a function returns a sequence of "I" or "O" that is partial of an inputting sequence.
21) $CntI(\cdot)$: a function returns the count of "I" in an inputting sequence.
22) $CntO(\cdot)$: a function returns the count of "O" in an inputting sequence.

B. Reduced Dynamics

Let $f(\cdot)$ denote Collatz transformation. $f(x) = 3 * x + 1$ if $x \in [1]_2$, and $f(x) = x/2$ if $x \in [0]_2$. We assume $f^0(x) = x$. Let $f^n(x) = f(f^{n-1}(x)), n \in \mathbb{N}^*$.

Collatz Conjecture can be stated as follows: $\forall x \in \mathbb{N}^*$, $\exists L \in \mathbb{N}^*$, such that $f^L(x) = 1$. Obviously, Collatz conjecture is held when $x = 1$. In the following, we mainly concern $x \geq 2, x \in \mathbb{N}^*$.

Reduced Collatz Conjecture can be stated as follows: $\forall x \in \mathbb{N}^*, x \geq 2$, $\exists L \in \mathbb{N}^*$, such that $f^L(x) < x$ and $f^i(x) \not< x, i = 0, 1, ..., L-1$. That is, the minimal L such that $f^L(x) < x$ is of interest.

Theorem 2.1: Collatz conjecture is equivalent to Reduced Collatz conjecture.

Proof Please see reference [4]. ∎

Remark
1) *original dynamics* is a sequence of occurred Collatz transformations during the procedure from a starting number to 1.
2) *reduced dynamics* is a sequence of occurred Collatz transformations during the procedure from a starting number to the *first* transformed number that is less than the starting number.

Note that, reduced dynamics is more primitive than original dynamics, because original dynamics consists of reduced dynamics. It can be easily observed from the proof of Theorem 2.1.

Proposition 2.2: $x/2$ always follows after $3 * x + 1$.

Proof Straightforward. ∎

Hence, we introduce two notations: $I(x) = (3 * x + 1)/2$ (I means increase as $(3 * x + 1)/2 > x$) and $O(x) = x/2$ (O means down).

For example, reduced dynamics of $x = 5$ is $\langle I(\cdot), O(\cdot)\rangle$ or "IO" in short.

Besides, $O(I(x))$ can be simply written as $IO(x)$. That is, $IO(x) = O(I(x))$, where $IO(\cdot)$ is a composite function. For example, $IO(5) = O(I(5)) = O((3 * 5 + 1)/2) = O(8) = 8/2 = 4 < 5$. Formally, $f_1\|f_2\|...\|f_n(x) = f_n(f_{n-1}(...f_2(f_1(x))))$, where $f_i(\cdot) \in \{I(\cdot), O(\cdot)\}, i = 1, 2, ..., n$. For simplicity, we may denote them as $f \in \{I, O\}$ in the following.

Moreover, to denote reduced dynamics of x, we introduce a notation $\mathsf{RD}[x]$. That is, $\forall x \in \mathbb{N}^*$, $x \geq 2$, if $\exists L \in \mathbb{N}^*$ such that $f^L(x) < x$ and $f^i(x) \not< x, i = 0, 1, ..., L-1$, where $f(\cdot) \in \{I(\cdot), O(\cdot)\}$, then let $c = f^L \in \{I, O\}^L$ and call c as reduced dynamics of x, denoted as $\mathsf{RD}[x] = f^L = c$.

For example, $\mathsf{RD}[7] = IIIOIOO$, $\mathsf{RD}[11] = IIOIO$. (Indeed, we design a computer program that outputs all $\mathsf{RD}[x], x \in [1, 99999999]$, which is observed and we discover some laws that will be proved in the following. Those data will be provided upon request.)

Obviously, $f^L \in \{I, O\}^L$ is an ordered sequence consisting of I and O. Besides, $f^L = f^{L-1}\|f, f^L(x) = f(f^{L-1}(x))$. Recall that $f^0(x) = x$. Furthermore, this sequence implicitly matches the *parity* of all intermediate transformed numbers that are taken as input of $f(\cdot)$.

Example 2.3:
$\mathsf{RD}[5] = IO$. It implies following results:
1) $IO(5) = (3 * 5 + 1)/2/2 = 4 < 5$;
2) $I(5) = (3 * 5 + 1)/2 = 8 \not< 5$;
3) "I" is due to $5 \in [1]_2$;
4) "O" is due to $I(5) = (3 * 5 + 1)/2 = 8 \in [0]_2$.

Proposition 2.4: $\forall x \in \mathbb{N}^*$, $x \geq 2$, if $\exists L \in \mathbb{N}^*$, such that $f^L(x) < x$ and $f^i(x) \not< x, i = 0, 1, ..., L-1$, where $f(\cdot) \in \{I(\cdot), O(\cdot)\}$, let $c = f^L \in \{I, O\}^L$, then c is unique.

Proof See reference [4]. ∎

Remark
1) We assume $\mathsf{RD}[x = 1] = IO$, 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) If $L = |c|$ is finite for x ($|\cdot|$ returns the length of c, or the number of I and O in the ordered sequence $c \in \{I, O\}^{\geq 1}$), then $\mathsf{RD}[x]$ exists; If $\mathsf{RD}[x]$ exists, then L is finite.
3) If Reduced Collatz conjecture is true, then $\forall x \in \mathbb{N}^*$, $\mathsf{RD}[x]$ exists; If $\forall x \in \mathbb{N}^*$, $\mathsf{RD}[x]$ exists, then Reduced Collatz conjecture is true.
4) In $\mathsf{RD}[x] = f^L$, x is called starting integer. $f^i(x), i = 1, 2, ..., L, L = |c|$ are called transformed integers. $f^L(x)$ is the first transformed integer that is less than the starting integer x. In other words, $f^i(x) \not< x, i = 0, 1, ..., L-1$, and $f^L(x) < x$. ($f^0(x) = x$.) Besides, the parity of $f^i(x)$ determines the selection of the intermediately next $f(\cdot) \in \{I(\cdot), O(\cdot)\}$ after f^i.
5) We do not assume the existence of $\mathsf{RD}[x]$ for $\forall x \in \mathbb{N}^*$, which is exactly what needs to be proved in Reduced Collatz Conjecture.

Proposition 2.5: Given $x \in \mathbb{N}^*$, if $\mathsf{RD}[x]$ exists, then $\mathsf{RD}[x]$ ends by O.

Proof Straightforward due to $I(x) = (3 * x + 1)/2 > x$. Suppose $\exists x \in \mathbb{N}^*$, $x \geq 2$, $c(x) \not< x$, $\mathsf{RD}[x] = c\|I$.

$\{c\|I\}(x) = I(c(x)) = (3 * c(x) + 1)/2 = 1.5c(x) + 0.5 = c(x) + 0.5c(x) + 0.5 > c(x)$, thus $\mathsf{RD}[x] = \{c\|I\}(x) \not< x$. Contradiction occurs. ∎

Proposition 2.6: RD$[x \in [0]_2] = O$, RD$[x \in [1]_4] = IO$.

Proof Straightforward. ∎

Proposition 2.7: Given $x \in [3]_4$, if RD$[x]$ exists, then RD$[x] \in I^pO \| \{I, O\}^{\geq 1}$, $p \geq 2$.

Proof See reference [4]. ∎

Next, we will analyze some reduced dynamics with short length, e.g., less than 7.

Lemma 2.8: $m \in \mathbb{N}^*, m \geq 2, 0 \leq i \leq m-1, a, b, x \in \mathbb{N}^*$.
(1) $x \in [i]_m \Rightarrow a*x+b \in [(a*i+b) \mod m]_m$;
(2) $m, i \in [0]_2, x \in [i]_m \Rightarrow x/2 \in [i/2]_{m/2}$;
(3) $m, (a*i+b)/2 \in [0]_2, x \in [i]_m \Rightarrow (a*x+b)/2 \in [((a*i+b) \mod m)/2]_{m/2}$.

Proof Please see reference [4]. ∎

Proposition 2.9: RD$[x \in [3]_{16}] = IIOO$.

Proof Straightforward. ∎

Example 2.10: $115 \rightarrow 346 \rightarrow 173 \rightarrow 520 \rightarrow 260 \rightarrow 130 \rightarrow 65 < 115$. Thus, RD$[115] = IIOO$. $x = 115 \in [3]_{16}$.

Proposition 2.11: RD$[x \in [11]_{32}] = IIOIO$.

Proof Straightforward, by analyzing whether $IIOIO(x) < x$, $IIO(x) \not< x$ (i.e., the first time to be less than x). ∎

Example 2.12: $11 \rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 < 11$; RD$[11] = IIOIO$, $x = 11 \in [11]_{32}$.

Similarly, we can prove following results.

Proposition 2.13: RD$[x \in [59]_{128}] = IIOIIOO$.

Proposition 2.14: RD$[x \in [15]_{128}] = IIIIOOO$.

Following Eq. (1) and Eq. (2), which can be proved similarly as Proposition 2.13.

$$\text{RD}[x] = \begin{cases} IIOO & x \in [3]_{16}, \\ IIOIO & x \in [11]_{32}, \\ IIIOO & x \in [23]_{32}, \\ IIIOIOO & x \in [7]_{128}, \\ IIIIOOO & x \in [15]_{128}, \\ IIOIIOO & x \in [59]_{128}. \end{cases} \tag{1}$$

$$\text{RD}[x] = \begin{cases} I^3OI^2OO & x \in [39]_{256}, \\ I^4OOIO & x \in [79]_{256}, \\ I^5OOO & x \in [95]_{256}, \\ I^2OI^2OIO & x \in [123]_{256}, \\ I^4OIOO & x \in [175]_{256}, \\ I^3OIOIO & x \in [199]_{256}, \\ I^2OI^3OO & x \in [219]_{256}. \end{cases} \tag{2}$$

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

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

E.g., $CntI(IIOO) = 2$, $CntO(IIOO) = 2$.

III. Representing Dynamics by Graph

A. Reduced Dynamics Graph

We propose to present reduced dynamics as a graph (called reduce dynamics graph), e.g., Fig. 1.

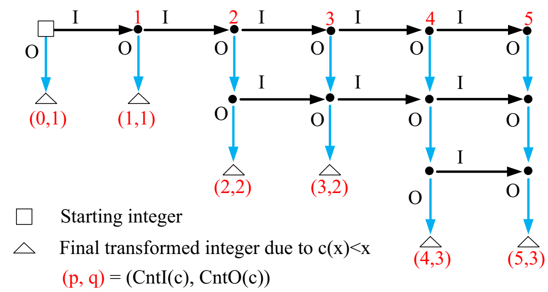

Fig. 1. Reduced dynamics graph.

Notations are used in the graph as follows:

1) According to Theorem 2.2, the occurred computation in (reduced) dynamics is either "I" or "O". Reduced dynamics for x can be represented by a path consisting of edge(s) labeled with I and O;
2) To distinguish "I" and "O", "I" is represented by "$\rightarrow$" in black, and "O" is represented by "$\downarrow$" in green in the graph;
3) Starting integers are represented by "□". Final transformed integers are represented by "△". Intermediate transformed integers are represented by black solid points located between edges;
4) Reduce dynamics can be looked (or organized in the graph) as one and only one path from "□" to "△". Inversely, each path from "□" to "△" represents one and only one residue class $c \in Set_c$, where Set_c is a set of all reduced dynamics. That is, $Set_c = \{c | x \in \mathbb{N}^*, \exists \text{RD}[x], c = \text{RD}[x] \in \{I, O\}^{\geq 1}\}$. "△" means reduced dynamics ends thereafter because current transformed integer is less than corresponding starting integer for the first time. In other words, $CntO(c) = \lceil \lambda * CntI(c) \rceil$, $\lambda = \log_2 1.5$. (This result is observed from the graph and proved in another paper by us formally; the proof is too long to be stated here.)
5) The label (p, q) under "△" points out the count of "I" and "O" in a path $c \in Set_c$, respectively. That is, $p = CntI(c), q = CntO(c)$. When $p = 0$, $q = 1$; When $p \geq 1$, $q = \lceil \lambda * p \rceil$, because $c \in Set_c$. (Indeed, p is the count of $3*x+1$ computation, q is the gap that is the count of $x/2$ computation minus the count of $3*x+1$ computation. $p+q$ is the count of $x/2$ computation.) All paths from "□" to the same "△" have identical (p, q). Each different path to (p, q) is associated to a different residue class rs where RD$[x \in rs] = c, p = CntI(c), q = CntO(c)$, $p \in \mathbb{N}$, $q = \max(1, \lceil \lambda * p \rceil)$.

For example, paths in reduced dynamics graph representing $IO, IIOO$ are depicted in Fig. 2, paths representing $IIOIO, IIIOO$ are depicted in Fig. 3, and paths representing $IIIOIOO$, $IIIIOOO$, $IIOIIOO$ are depicted in Fig. 4, respectively.

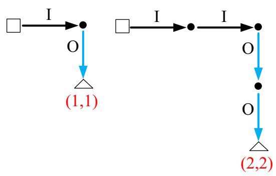

Fig. 2. $\mathsf{RD}[x \in [1]_4] = IO$. $\mathsf{RD}[x \in [3]_{16}] = IIOO$.

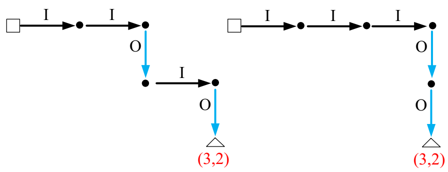

Fig. 3. $\mathsf{RD}[x \in [11]_{32}] = IIOIO, \mathsf{RD}[x \in [23]_{32}] = IIIOO$.

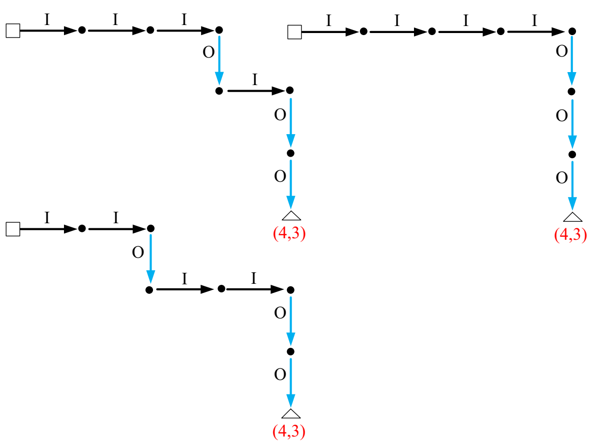

Fig. 4. $\mathsf{RD}[x \in [7]_{128}] = IIIOIOO, \mathsf{RD}[x \in [15]_{128}] = IIIIOOO, \mathsf{RD}[x \in [59]_{128}] = IIOIIOO$.

B. Dynamics Graph with Partition Labels

We introduce a new notation - $\mathsf{DYNM}(x, n) = \{I, O\}^n$ to denote the first $n \in \mathbb{N}^*$ transformations (i.e., I or O) of $x \in \mathbb{N}^*$ no matter whether $\mathsf{RD}[x]$ exists or not.

For example, $\mathsf{DYNM}(19, 2) = II$, $\mathsf{DYNM}(19, 3) = IIO$, and $\mathsf{DYNM}(19, 4) = IIOO$. Interestingly, we observe facts as follows:

$\mathsf{DYNM}(x \in [3]_8, 3) = IIO$. Let $f^3 = IIO$. $\forall x \in [3]_8$, if $f^3(x) \in [0]_2$, then $f^3(x+8) \in [1]_2$; If $f^3(x) \in [1]_2$, then $f^3(x+8) \in [0]_2$. For example, $x = 3 \in [3]_8$, $f^3(3) = IIO(3) = IO(5) = O(8) = 4 \in [0]_2$, $f^3(3+8 = 11) = IIO(11) = IO(17) = O(26) = 13 \in [1]_2$. Moreover, $f^3(11 + 8 = 19) = IIO(19) = IO(29) = O(44) = 22 \in [0]_2$.

Indeed, we can prove above key observation (which is called Partition Theorem and proved formally in our another paper [5]).More specifically, the first distinction between transformations of x and transformations of $x+2^t$ occurs at the $(t+1)$-th transformation (if $|\mathsf{RD}[x]| > t+1$). That is, $x \equiv x + 2^t \equiv i \mod 2^t$. $[i]_{2^{t+1}} \cup [i+2^t]_{2^{t+1}} = [i]_{2^t}$, $[i]_{2^{t+1}} \cap [i+2^t]_{2^{t+1}} = \emptyset$. 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 $x + 2^t \in [i]_{2^{t+1}}$.

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

The graph presents information as follows:

1) Either partition of current x presents the same previous dynamics, but will present either branch (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 $\mathsf{DYNM}(x, t+1) = \mathsf{DYNM}(x, t) \| I$ or $\mathsf{DYNM}(x, t+1) = \mathsf{DYNM}(x, t) \| O$. Besides, $\forall x \in [i]_{2^t}, \mathsf{DYNM}(x, t)$ are identical.
   For example, $\mathsf{DYNM}(x \in [3]_8, 3) = IIO$. 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 $\mathsf{DYNM}(x, L) = f^L \in \{I, O\}^L, L \in \mathbb{N}^*$. If $CntO(f^L) = \lceil \lambda * CntI(f^L) \rceil$ (or $CntO(f^L) = 1$, $CntI(CT^L) = 0$), then branches reach to final transformed numbers that are denoted as "△" (i.e., $\mathsf{DYNM}(x, L) = \mathsf{RD}[x]$; Or, $f^L(x) < x$, $Substr(f^L, 1, k)(x) \not< x, k = 1, 2, ..., L-1$).
3) Each edge (i.e., I or O) is labeled with one residue class or a union of multiple residue classes, whose last transformation is consistent with this edge (I or O), and x in which present the dynamics represented by the path from "□" to this edge.
   For example, $[11, 23]_{32}$ is labeled due to $\mathsf{DYNM}(x \in [11, 23]_{32}, 5) = f^5_{11}, f^5_{23} \in \{I, O\}^5$ where $Substr(f^5_{11}, 5, 1) = O, Substr(f^5_{23}, 5, 1) = O$. Hereby, in notation f^a_b, "a" represents its length, and "b" differentiate it among $f^a \in \{I, O\}^a$. More specifically, $\mathsf{DYNM}(x \in [11]_{32}, 5) = f^5_{11} = IIOIO$, $\mathsf{DYNM}(x \in [23]_{32}, 5) = f^5_{23} = IIIOO$.
   More specifically, if $[i]_{2^t}$ ($i \in ListI = \{i_1, i_2, ..., i_p\}, p \in \mathbb{N}^*$) is (are) residue class (classes) as a label located at the last edge of paths $f^t_i \in \{I, O\}^t$ (i.e., $Substr(f^t_i, t, 1)$), then $\mathsf{DYNM}(x \in [i \in ListI]_{2^t}, t) = f^t_i$.

C. A General Algorithm for Computing Residue Class from Dynamics

Definition Function $Match(\cdot, \cdot)$. $Match : x \times c \to b$ takes as input $x \in \mathbb{N}^*$ and $c \in \{I, O\}$, and outputs $b \in \{true, false\}$. When $x \in [1]_2, c = I$, or $x \in [0]_2, c = O$, $b = true$; Otherwise, $b = false$.

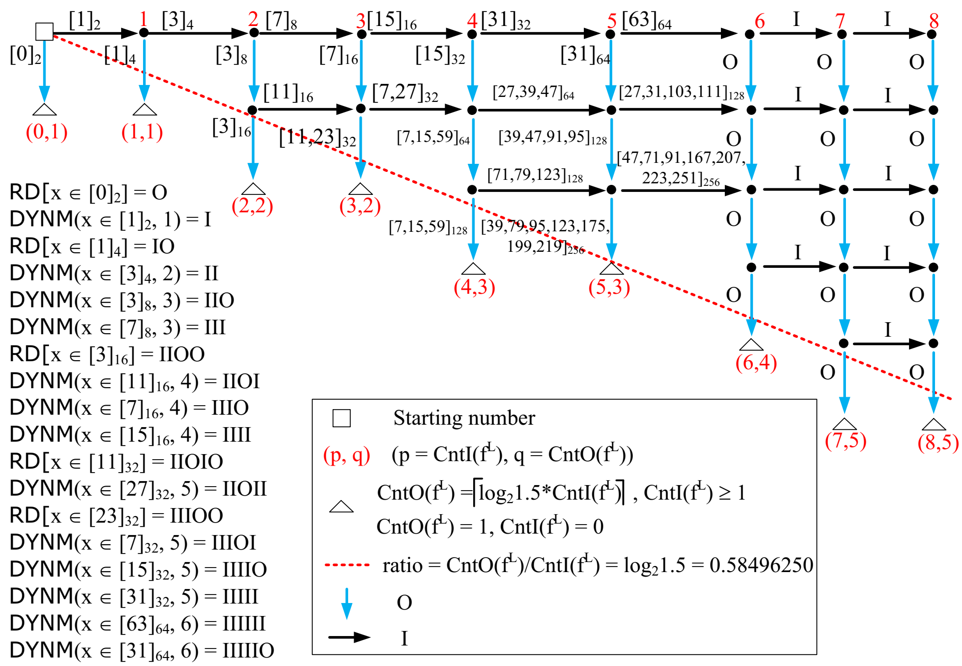

Fig. 5. Reduced dynamics graph with partition labels.

Definition Function $Substr(\cdot)$. $Substr : c \times i \times j \to c'$ takes as input c, i, j, where $c \in \{I,O\}^p$, $1 \le i \le p$, $1 \le j \le p-i+1$, and outputs $c' \in \{I,O\}^j$, $1 \le j \le p$ where $|c'| = j$ and $c \in \{I,O\}^{i-1} \| c' \| \{I,O\}^{(p-j)-(i-1)}$. (In other words, c' is a *segment* in c starting from i and the length of c' is j.)

For example, $Substr(IIOO,1,4) = IIOO$, $Substr(IIOO,1,3) = IIO$. Note that, $Substr(\cdot)$ itself is a function. In other words, it can be looked as $(Substr(\cdot))(\cdot)$. E.g., $Substr(IIOO,1,1)(3) = I(3) = (3*3+1)/2 = 5$,
$Substr(IIOO,1,2)(3) = II(3) = I(I(3)) = I(5) = (3*5+1)/2 = 8$,
$Substr(IIOO,1,3)(3) = IIO(3) = O(II(3)) = O(8) = 8/2 = 4$,
$Substr(IIOO,1,4)(3) = IIOO(3) = O(IIO(3)) = O(4) = 4/2 = 2 < 3$.

It is worth to stress that although in above definition $j \ge 1$, it can be extended to $j \ge 0$ by assuming $Substr(\cdot,\cdot,0)(x) = x$.

Example 3.1:

By using above two functions, if RD$[x]$ $(x \in \mathbb{N}^*, x \ge 2)$ exists, then

(1) $c(x) < x$, where $c = \text{RD}[x]$;

(2) $Substr(c,1,i)(x) \not< x$, where $i = 0,1,2,...,|c|-1$;

(3) $Match(Substr(c,1,i-1)(x), Substr(c,i,1)) = true$, where $i = 1,2,...,|c|$.

A general algorithm can be given by Theorem 4.12, which outputs residue class of x, and takes as input first t transformations in reduced dynamics of x (i.e., $f^t = \text{DYNM}(x,t)$, $|\text{RD}[x]| \ge t$).

Given $f^t \in \{I,O\}^t$, $t \in \mathbb{N}^*, t \ge 2$, $CntO(s) < \lceil \lambda * s \rceil$, $s = Substr(f^t,1,k)$, $k = 1,2,...,t-1$, $CntO(f^t) \le \lceil \lambda * CntI(f^t) \rceil$, $[i]_{2^t}$ can be determined by Algorithm 1 (called Dynm2Rs Algorithm) such that DYNM$(x \in [i]_{2^t}, t) = f^t$ as follows:

Data: f^t
Result: i
$i \Leftarrow 1$;
for $j = 1; j \le t-1; j++$ **do**
  if $Match(Substr(f^t,1,j)(i), Substr(f^t,j+1,1)) == false$ **then**
    $i \Leftarrow 2^j + i$;
  end
end
return $[i]_{2^t}$;

Algorithm 1: Dynm2Rs Algorithm. Input $f^t \in \{I,O\}^t, t \in \mathbb{N}^*$, $t \ge 2$, $CntO(Substr(f^t,1,k)) < \lceil \lambda * CntI(Substr(f^t,1,k)) \rceil$ where $k = 1,2,...,t-1$, $CntO(f^t) \le \lceil \lambda * CntI(f^t) \rceil$. Output $[i]_{2^t}$, $1 \le i \le 2^t - 1$, $i \in [1]_2$, such that DYNM$(x \in [i]_{2^t}, t) = f^t$.

Remark

(1) If $Match(Substr(f^t,1,j)(i), Substr(f^t,j+1,1)) = false$, then $i \Leftarrow 2^j + i$. It is due to Theorem 4.12.

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

(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 Dynm2Rs algorithm be $[1]_{2^1}$. If input is O, then let the output be $[0]_{2^1}$. Consequently, Corollary 3.2 can be extended to include $t=1$ easily.

Corollary 3.2:
(1) Given $f^t \in \{I,O\}^t, t \in \mathbb{N}^*, t \geq 2$,
$CntO(s) < \lceil \lambda * CntI(s) \rceil,\ s = Substr(f^t,1,k),\ k = 1,2,...,t-1$,
$CntO(f^t) \leq \lceil \lambda * CntI(f^t) \rceil$,
exists one and only one $i \in [1, 2^t-1], i \in [1]_2$ such that $\mathsf{DYNM}(x \in [i]_{2^t}, t) = f^t$.
Inversely, given $x \in [i]_{2^t}, t \in \mathbb{N}^*, t \geq 2, i \in [1, 2^t-1], i \in [1]_2$,
exists one and only one $f^t \in \{I,O\}^t$ such that $\mathsf{DYNM}(x,t) = f^t,\ CntO(s) < \lceil \lambda * CntI(s) \rceil,\ s = Substr(f^t,1,k),\ k = 1,2,...,t-1,\ CntO(f^t) \leq \lceil \lambda * CntI(f^t) \rceil$.
(2) If $N_1 = \|\{f^t | f^t \in \{I,O\}^t,\ t \in \mathbb{N}^*, t \geq 2,$ $i \in [1, 2^t-1], i \in [1]_2,$
$CntO(Substr(f^t,1,k)) < \lceil \lambda * CntI(Substr(f^t,1,k)) \rceil,$ $k = 1,2,...,t-1,\ CntO(f^t) \leq \lceil \lambda * CntI(f^t) \rceil,\ \mathsf{DYNM}(x \in [i]_{2^t}, t) = f^t\}\|$, and
$N_2 = \|\{i | f^t \in \{I,O\}^t,\ t \in \mathbb{N}^*, t \geq 2,\ i \in [1, 2^t-1], i \in [1]_2,$
$CntO(Substr(f^t,1,k)) < \lceil \lambda * CntI(Substr(f^t,1,k)) \rceil, k = 1,2,...,t-1,$
$CntO(f^t) \leq \lceil \lambda * CntI(f^t) \rceil,\ \mathsf{DYNM}(x \in [i]_{2^t}, t) = f^t\}\|$,
then $N_1 = N_2$.

Proof (1) Prove equivalency.
(1.1) $[i]_{2^t} \Rightarrow f^t$. 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 4.11.
(1.2) $f^t \Rightarrow [i]_{2^t}$. As all transformations of $[j]_{2^t}, j = 1,...,2^t-1, j \in [1]_2$ can be enumerated, one and only one of them equals f^t due to Lemma 4.11. That j is i. Alternatively, non-trivial algorithm outputting i for given f^t is proposed in Algorithm 1.
(2) Due to (1), we have $N_1 = N_2$. (Recall Corollary 4.13 and Corollary 4.14.) ∎

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

IV. Conclusion

We use $\mathsf{RD}[x]$ to denote the reduced dynamics of x, which is represented by a sequence of computation of I or O, where I denotes $(3*x+1)/2$ and O denotes $x/2$.

This paper propose reduced dynamics graph, and reduced dynamics graph with partition labels, to observe the properties of reduced dynamics.

The reduced dynamics graph shows that the ratio - the count of I over the count of O, is bounded by $\log_2 1.5$. Once the count of O is sufficient large, namely, equals $\lceil \log_2 1.5 * CntI(s) \rceil$, then $s(x) < x$ where $\mathsf{DYNM}(x,t) = s, t = |s|$, i.e., $\mathsf{RD}[x] = s$.

The reduced dynamics graph with partitions show that all natural numbers, 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 $x_1 \equiv x_2 \mod 2^t$ and $x_1 \equiv x_2 + 2^t \mod 2^{t+1}$. The first t transformations for all $x \in [x \mod 2^t]_{2^t}$ are identical. The $(t+1)$-th transformation for x and $x+2^t$ is distinct. $\forall x \in [1]_2$, if $s(x) \not< x,\ s = \mathsf{DYNM}(x,t),\ t \geq 2$, 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., $\mathsf{DYNM}(x,t) = \mathsf{RD}[x]$.

We finally propose an algorithm that takes as inputs the first t transformation in terms of I and O in reduced dynamics, and outputs corresponding reside class in which all natural numbers will present those t transformations. More specifically, it takes as input $f^t \in \{I,O\}^t, t \in \mathbb{N}^*,\ t \geq 2,\ CntO(Substr(f^t,1,k)) < \lceil \lambda * CntI(Substr(f^t,1,k)) \rceil$ where $k = 1,2,...,t-1,\ CntO(f^t) \leq \lceil \lambda * CntI(f^t) \rceil$. Output $[i]_{2^t},\ 1 \leq i \leq 2^t-1,\ i \in [1]_2$, such that $\mathsf{DYNM}(x \in [i]_{2^t}, t) = f^t$.

References

[1] T.O. 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] T.O. 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, American Mathematical Society, 2010.
[3] W. Ren, S. Li, R. Xiao and W. Bi, *Collatz Conjecture for $2^{100000}-1$ is True - Algorithms for Verifying Extremely Large Numbers*, Proc. Of IEEE UIC 2018, Oct. 2018, Guangzhou, China, 411-416, 2018
[4] W. 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] W. Ren, *Collatz Dynamics is Partitioned by Residue Class Regularly*, Information and Computation, Elsevier, Submitted, 2019.
[6] W. 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 ln3/ln2*, Mathematical Reports, Romanian Academy, Submitted, 2019.

Appendix

A. Some Conclusions on Ratio [6]

Corollary 4.1: (**Form Corollary.**) $\forall c \in \{I,O\}^{\geq 1}$. When $|c| = 1,\ c = O$; When $|c| \geq 2,\ s = Substr(c,1,i),\ i = 1,2,...,|c|-1$,

$$\begin{cases} CntO(c) = \lceil \lambda * CntI(c) \rceil \\ CntO(s) < \lceil \lambda * CntI(s) \rceil. \end{cases} \tag{3}$$

$\Leftrightarrow c \in Set_c$.

Theorem 4.2: (**Inverse Theorem.**)
(1) If $|c| = 1, c = O$, then $\exists x \in [0]_2$ such that $\mathsf{RD}[x] = O$.
(2) If $|c| \geq 2,\ c = I^{p_1}O^{q_1}I^{p_2}O^{q_2}...I^{p_n}O^{q_n},\ p_i, q_i \in \mathbb{N}^*,$ $i = 1,2,...,n, n \in \mathbb{N}^*,$
$CntO(c) = \lceil \lambda * CntI(c) \rceil,$
$CntO(s) < \lceil \lambda * CntI(s) \rceil,\ s = Substr(c,1,k), k = 1,2,...,|c|-1.$
then $\exists x \in [-\sum_{i=1}^{n} A_i B_i C_{i-1}]_{C_n} \wedge x > \Psi *$

$\frac{2^{\sum_{i=1}^{n}(p_i+q_i)}}{2^{\sum_{i=1}^{n}(p_i+q_i)}-3^{\sum_{i=1}^{n}p_i}}$
such that $\mathsf{RD}[x] = c$
where $A_i = 3^{p_i} - 2^{p_i}$, $B_i = (3^{\sum_{j=1}^{i} p_j})^{-1} \mod C_n$, $C_i = 2^{\sum_{j=1}^{i}(p_j+q_j)}, C_0 = 1$, $\Psi = \prod_{i=2}^{n} a_i b_1 + \prod_{i=3}^{n} a_i b_2 + ... + \prod_{i=n-1}^{n} a_i b_{n-2} + \prod_{i=n}^{n} a_i b_{n-1} + b_n$, $a_i = \frac{3^{p_i}}{2^{p_i+q_i}}, b_i = \frac{3^{p_i}-2^{p_i}}{2^{p_i+q_i}}$.

The following corollary states that any valid reduced dynamics (e.g., Set_c) can be constructed by generating by an algorithm, instead of by computing concrete Collatz transformations step by step.

Corollary 4.3: $Set_c = \{O\} \cup \{c | c \in \{I,O\}^L, L \in \mathbb{N}^*, L \geq 2, CntO(c) = \lceil \lambda * CntI(c) \rceil, CntO(s) < \lceil \lambda * CntI(s) \rceil, s = Substr(c,1,i), i = 1,2,...,L-1, \lambda = \log_2 1.5\}$.

Theorem 4.4: (**Simplified Inverse Theorem**.)
(1) If $|c| = 1, c = O$, then $\exists x \in [0]_2$ such that $\mathsf{RD}[x] = O$.
(2) If $|c| \geq 2$, $c = I^{p_1}O^{q_1}I^{p_2}O^{q_2}...I^{p_n}O^{q_n}$, $p_i, q_i \in \mathbb{N}^*$, $i = 1,2,...,n, n \in \mathbb{N}^*$,
$CntO(c) = \lceil \lambda * CntI(c) \rceil$,
$CntO(s) < \lceil \lambda * CntI(s) \rceil, s = Substr(c,1,k), k = 1,2,...,|c|-1, \lambda = \log_2 1.5$.
then $\exists x \in [-\sum_{i=1}^{n} A_i B_i C_{i-1}]_{C_n}$
such that $\mathsf{RD}(x) = c$,
where $A_i = 3^{p_i} - 2^{p_i}$, $B_i = (3^{\sum_{j=1}^{i} p_j})^{-1} \mod C_n$, $C_i = 2^{\sum_{j=1}^{i}(p_j+q_j)}, C_0 = 1$, $\Psi = \prod_{i=2}^{n} a_i b_1 + \prod_{i=3}^{n} a_i b_2 + ... + \prod_{i=n-1}^{n} a_i b_{n-2} + \prod_{i=n}^{n} a_i b_{n-1} + b_n$,
$a_i = \frac{3^{p_i}}{2^{p_i+q_i}}, b_i = \frac{3^{p_i}-2^{p_i}}{2^{p_i+q_i}}$.

Definition Function $Invrs(\cdot)$ (Simplified Version).
Function $Invrs : c \to rs$ takes as input $c = O$ or
$c = I^{p_1}O^{q_1}I^{p_2}O^{q_2}...I^{p_n}O^{q_n} \in \{I,O\}^{\geq 2}$, $p_i, q_i \in \mathbb{N}^*, i = 1,2,...,n, n \in \mathbb{N}^*$
$CntO(c) = \lceil \lambda * CntI(c) \rceil$,
$CntO(s) < \lceil \lambda * CntI(s) \rceil, s = Substr(c,1,k), k = 1,2,...,|c|-1$,
and outputs $rs = [0]_2$ when $c = O$, or
$rs = [-\sum_{i=1}^{n} A_i B_i C_{i-1}]_{2^{|c|}}$ when $|c| \geq 2$,
where $A_i = 3^{p_i} - 2^{p_i}$, $B_i = (3^{\sum_{j=1}^{i} p_j})^{-1} \mod C_n$, $C_i = 2^{\sum_{j=1}^{i}(p_j+q_j)}, C_0 = 1$,
$\Psi = \prod_{i=2}^{n} a_i b_1 + \prod_{i=3}^{n} a_i b_2 + ... + \prod_{i=n-1}^{n} a_i b_{n-2} + \prod_{i=n}^{n} a_i b_{n-1} + b_n$,
$a_i = \frac{3^{p_i}}{2^{p_i+q_i}}, b_i = \frac{3^{p_i}-2^{p_i}}{2^{p_i+q_i}}$.

Therefore, thanks to Simplified Inverse Theorem, it becomes much easier to prove Reduced Collatz conjecture by proving $\bigcup_{c \in Set_c} Invrs(c) = \mathbb{N}^*$, where $Set_c = \{O\} \cup \{c | c \in \{I,O\}^L, L \in \mathbb{N}^*, L \geq 2, CntO(c) = \lceil \lambda * CntI(c) \rceil$,
$CntO(s) < \lceil \lambda * CntI(s) \rceil, s = Substr(c,1,i), i = 1,2,...,L-1\}$ (recall Corollary 4.3).

B. The Proof of Partition Theorem [5]

A new notation $I'(\cdot)$ is introduced - $I'(x) = (3*x)/2$.

Example 4.5: $I(3+16) = (3(3+16)+1)/2 = (3*3+1)/2 + 3*16/2 = I(3) + I'(16)$, $I(3) = (3*3+1)/2 = 5, I'(16) = 3*16/2 = 24 \in [0]_2$;
$II(3+16) = I(I(3)+I'(16)) = (3*(I(3)+I'(16))+1)/2 = (3I(3)+1)/2 + 3I'(16)/2 = II(3) + I'I'(16), II(3) = I(5) = (3*5+1)/2 = 8, I'I'(16) = I'(24) = 3*24/2 = 36 \in [0]_2$;
$IIO(3+16) = O(II(3)+I'I'(16)) = IIO(3) + I'I'O(16) = 8/2 + 36/2$
$= 4 + 18 = 22 > (3+16), 18 \in [0]_2$;
$IIOO(3+16) = O(IIO(3) + I'I'O(16)) = IIOO(3) + I'I'OO(16)$
$= 4/2 + 18/2 = 11 < (3+16)$.

In above example $I'(16), I'I'(16), I'I'O(16)$, and $I'I'OO(16)$ are always remained *even*. Thus, they do not influence the next transformation selection (i.e., "I" or "O") during reduced dynamics of starting number 3. That is, the occurrence of I and O in $\mathsf{RD}[3+16]$ is only determined by the computation results of 3.

Definition Function $IsOdd(\cdot)$. $IsOdd : x \to c \in \{1,0\}$ takes as input $x \in \mathbb{N}^*$, outputs $c = 1$ if $x \in [1]_2$ and $c = 0$ if $x \in [0]_2$.

Definition Function $Replace(\cdot)$. $Replace : str \to str'$ takes as input $str \in \{I,O\}^{\geq 1}$, outputs $str' \in \{I',O\}^{\geq 1}$, where $i = 1,2,...,|str|$ and

$$Substr(str', i, 1) = \begin{cases} I' & Substr(str, i, 1) = I, \\ O & Substr(str, i, 1) = O. \end{cases} \quad (4)$$

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

Next lemma states that if $P \in [0]_2$, the first Collatz transformation of $x + P$ (or $x - P > 0$) is the same as that of x.

Lemma 4.6: (1) $P \in [0]_2, x \in \mathbb{N}^* \Rightarrow IsOdd(x+P) = IsOdd(x)$. (2) $P \in [0]_2, x - P > 0, x \in \mathbb{N}^* \Rightarrow IsOdd(x-P) = IsOdd(x)$.

Proof Straightforward. Due to $P \in [0]_2$, if $x \in [1]_2$, then $x + P \in [1]_2$; if $x \in [0]_2$, then $x + P \in [0]_2$. Thus, the first transformation of $x + P$ is the same as that of x. Similarly, conclusion (2) can be proved. ∎

Next lemma states the *separation* property of computation for $|STR| = 1$. Note that, we mainly concentrate on STR that guarantees $Match(Substr(STR,1,i)(x), Substr(STR, i+1, 1)) = true$, $i = 0,1,...,|STR|-1$, although the conclusion is general for any $STR \in \{I,O\}^{\geq 1}$. (Recall that $Substr(\cdot,\cdot,0)(x) = x$.)

Lemma 4.7: $STR \in \{I,O\}$, $STR' = Replace(STR)$, $x \in \mathbb{N}^*, P \in [0]_2$. We have $STR(x+P) = STR(x) + STR'(P)$.

Proof As $P \in [0]_2$, $IsOdd(x+P) = IsOdd(x)$ due to Lemma 4.6.

$|STR| = 1$, thus $STR = I$ or O.

(1) Suppose $x \in [1]_2$, $STR = I$. Thus, $STR' = Replace(STR) = I'$.

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

(2) Suppose $x \in [0]_2$, $STR = O$. Thus $STR' = Replace(STR) = O$.

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

Summarizing (1) and (2), $STR(x+P) = STR(x) + STR'(P)$. ∎

Next lemma states the *separation* property of computation for $|STR| \geq 2$.

Lemma 4.8: (**Separation Lemma**.) $STR \in \{I, O\}^{\geq 2}$, $STR' = Replace(STR)$, $x \in \mathbb{N}^*, P \in [0]_2$.

(1) $Substr(STR,1,1)(x+P) = Substr(STR,1,1)(x) + Substr(STR',1,1)(P)$.

(2) $j = 1, 2, ..., |STR'| - 1$. $Substr(STR,1,j)(x+P) = Substr(STR,1,j)(x) + Substr(STR',1,j)(P) \wedge Substr(STR',1,j)(P) \in [0]_2 \Rightarrow IsOdd(Substr(STR,1,j)(x+P)) = IsOdd(Substr(STR,1,j)(x)) \wedge Substr(STR,1,j+1)(x+P) = Substr(STR,1,j+1)(x) + Substr(STR',1,j+1)(P)$.

Proof (1) $j = 1$. Indeed, it is the same as Lemma 4.7. Nonetheless, we still give the details as follows:

$P \in [0]_2$, thus $IsOdd(x+P) = IsOdd(x)$ due to Lemma 4.6. The first transformation of $x+P$ and x are identical.

(1.1) $Substr(STR,1,j) = I$.

$Substr(STR,1,j)(x+P) = I(x+P) = 3((x+P)+1)/2$
$= (3x+1)/2 + 3*P/2 = I(x) + I'(P)$
$= Substr(STR,1,j)(x) + Substr(STR',1,j)(P)$.

(1.2) $Substr(STR,1,j) = O$.

$Substr(STR,1,j)(x+P) = O(x+P) = (x+P)/2 = x/2 + P/2 = O(x) + O(P) = Substr(STR,1,j)(x) + Substr(STR',1,j)(P)$.

Thus, $Substr(STR,1,j)(x+P) = Substr(STR,1,j)(x) + Substr(STR',1,j)(P)$.

(2) Due to (1), for $j = 1$,
$IsOdd(Substr(STR,1,j)(x+P))$
$= IsOdd(Substr(STR,1,j)(x) + Substr(STR',1,j)(P))$.

Besides, if $Substr(STR',1,j)(P) \in [0]_2$, then
$IsOdd(Substr(STR,1,j)(x+P)) = IsOdd(Substr(STR,1,j)(x))$.

Next, we prove the other concluding item as follows:
$Substr(STR,1,j+1)(x+P)$
$= (Substr(STR,1,j)\|I)(x+P) \vee (Substr(Str,1,j)\|O)(x+P)$
$= I(Substr(STR,1,j)(x+P)) \vee O(Substr(STR,1,j)(x+P))$
$= I(Substr(STR,1,j)(x) + Substr(STR',1,j)(P)) \vee O(Substr(STR,1,j)(x) + Substr(STR',1,j)(P))$
$= (3(Substr(STR,1,j)(x) + Substr(STR',1,j)(P)) + 1)/2 \vee (Substr(STR,1,j)(x) + Substr(STR',1,j)(P))/2$
$= (3*Substr(STR,1,j)(x)+1)/2 + 3*Substr(STR',1,j)(P)/2 \vee O(Substr(STR,1,j)(x)) + O(Substr(STR',1,j)(P))$
$= I(Substr(STR,1,j)(x)) + I'(Substr(STR',1,j)(P)) \vee O(Substr(STR,1,j)(x)) + O(Substr(STR',1,j)(P))$
$= (Substr(STR,1,j)\|I)(x) + (Substr(STR',1,j)\|I')(P) \vee (Substr(STR,1,j)\|O)(x) + (Substr(STR',1,j)\|O)(P)$
$= Substr(STR,1,j+1)(x) + Substr(STR',1,j+1)(P)$.
(Note that, here $j+1 = 2$. Recall that "$\|$" is concatenation.)

(3) Next, $j = 2, 3, ..., |STR'| - 1$, respectively and especially in an order.

$Substr(STR,1,j)(x+P) = Substr(STR,1,j)(x) + Substr(STR',1,j)(P) \wedge Substr(STR',1,j)(P) \in [0]_2$
$\Rightarrow IsOdd(Substr(STR,1,j)(x+P)) = IsOdd(Substr(STR,1,j)(x))$.

Again, $Substr(STR,1,j+1)(x+P)$
$= (Substr(STR,1,j)\|I)(x+P) \vee (Substr(Str,1,j)\|O)(x+P)$
$= I(Substr(STR,1,j)(x+P)) \vee O(Substr(STR,1,j)(x+P))$
$= I(Substr(STR,1,j)(x) + Substr(STR',1,j)(P)) \vee O(Substr(STR,1,j)(x) + Substr(STR',1,j)(P))$
$= (3(Substr(STR,1,j)(x) + Substr(STR',1,j)(P)) + 1)/2 \vee (Substr(STR,1,j)(x) + Substr(STR',1,j)(P))/2$
$= (3*Substr(STR,1,j)(x)+1)/2 + 3*Substr(STR',1,j)(P)/2 \vee O(Substr(STR,1,j)(x)) + O(Substr(STR',1,j)(P))$
$= I(Substr(STR,1,j)(x)) + I'(Substr(STR',1,j)(P)) \vee O(Substr(STR,1,j)(x)) + O(Substr(STR',1,j)(P))$
$= (Substr(STR,1,j)\|I)(x) + (Substr(STR',1,j)\|I')(P) \vee (Substr(STR,1,j)\|O)(x) + (Substr(STR',1,j)\|O)(P)$
$= Substr(STR,1,j+1)(x) + Substr(STR',1,j+1)(P)$. ∎

Remark

(1) Separation Lemma can be extended to include $|STR| = 1$. By assuming $Substr(\cdot,\cdot,0)(x) = x$, $Substr(STR,1,0)(x+P) = x+P$
$= Substr(STR,1,0)(x) + Substr(STR',1,0)(P)$. Thus, by Lemma 4.6, Lemma 4.7 and Lemma 4.8, we have
$Substr(STR',1,j)(P) \in [0]_2 \Rightarrow$
$Substr(STR,1,j+1)(x+P) = Substr(STR,1,j+1)(x) + Substr(STR',1,j+1)(P), j = 0, 1, ..., |STR'| - 1$.

(2) Indeed, Separation Lemma states the sufficient condition (i.e., $Substr(STR',1,j)(P) \in [0]_2, j = 1, 2, ..., |STR'| - 1$) for guaranteeing that all intermediate parities of transformed numbers or ordered parity sequence except for the first one (i.e., $Substr(STR,1,1)$) of $x+P$ are exactly the same as those of x. If the first one is included as well, then the sufficient condition is $Substr(STR',1,j)(P) \in [0]_2, j = 0, 1, ..., |STR'| - 1$.

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

Obviously, $CntI(STR) = CntI'(STR')$ where $STR' = Replace(STR), STR \in \{I, O\}^{\geq 1}$.

Lemma 4.9: $STR'(x) = \frac{3^{CntI'(STR')}}{2^{|STR'|}} * x$, where $STR' \in \{I', O\}^{\geq 1}$.

Proof (1) When $|STR'| = 1$, $STR' = I' \vee O$.

$STR'(x) = I'(x) \vee O(x) = 3*x/2 \vee x/2 = 3^1 * x/2^1 \vee x/2^1 = \frac{3^{CntI'(STR')}}{2^{|STR'|}}$.

(2) Suppose $|STR'| = i, i \geq 1, i \in \mathbb{N}^*$, we have $STR'(x) = \frac{3^{CntI'(STR')}}{2^{|STR'|}} * x$.

Next, we explore $\overline{STR'} \in \{I', O\}^{i+1}, |\overline{STR'}| = i + 1$.

$\forall \overline{STR'} \in \{I', O\}^{i+1}$, $\exists STR' \in \{I', O\}^i$, such that $\overline{STR'} = STR' \| I' \vee STR' \| O$. Recall that "$\|$" is concatenation. Besides, $STR'(x) = \frac{3^{CntI'(STR')}}{2^{|STR'|}} * x$.

$(STR' \| I')(x) = I'(STR'(x)) = 3(STR'(x))/2$
$= \frac{3^{CntI'(STR')+1}}{2^{|STR'|+1}} * x = \frac{3^{CntI'(STR' \| I')}}{2^{|STR' \| I'|}} * x$.

$(STR' \| O)(x) = O(STR'(x)) = STR'(x)/2$
$= \frac{3^{CntI'(STR')}}{2^{|STR'|+1}} * x = \frac{3^{CntI'(STR' \| O)}}{2^{|STR' \| O|}} * x$.

Thus, $\overline{STR'}(x) = \frac{3^{CntI'(\overline{STR'})}}{2^{|\overline{STR'}|}} * x$.

In summary, $STR'(x) = \frac{3^{CntI'(STR')}}{2^{|STR'|}} * x$ for any length of STR' due to (1) and (2). ∎

Lemma 4.10: If $\mathsf{DYNM}(x, t) = f^t \in \{I, O\}^t$, $x, t \in \mathbb{N}^*$, then
(1) $t = 1$. $IsOdd(f^t(x)) \neq IsOdd(f^t(x + 2^t))$; Or,
(2) $t \geq 2$.

$$\begin{cases} IsOdd(f^k(x)) = IsOdd(f^k(x + 2^t)), \\ IsOdd(f^t(x)) \neq IsOdd(f^t(x + 2^t)), \end{cases} \tag{5}$$

where $f^k = Substr(f^t, 1, k), k = 1, 2, ..., t - 1$.

Proof (1) $t = 1$. $2^t = 2 \in [0]_2$. $f^t = I$ or O.
$I(x + 2) = (3(x + 2) + 1)/2 = (3x + 1)/2 + 3 = I(x) + 3$.
Thus, $IsOdd(I(x + 2)) \neq IsOdd(I(x))$.
$O(x+2) = (x+2)/2 = O(x)+1$. Thus, $IsOdd(O(x+2)) \neq IsOdd(O(x))$.
Hence, $IsOdd(f^t(x)) \neq IsOdd(f^t(x + 2^t))$.

(2) $t \geq 2$. Let $f'^k = Replace(f^k)$.
$f'^k(2^t) = \frac{3^{CntI'(f'^k)}}{2^{|f'^k|}} * 2^t = \frac{3^{CntI'(f'^k)}}{2^k} * 2^t \quad \because$ Lemma 4.9.
$= 3^{CntI'(f'^k)} * 2^{t-k} \in [0]_2, \quad \because 1 \leq k \leq t - 1 \Rightarrow t - k \geq 1$.

Due to Separation Lemma (i.e., Lemma 4.8), we have
$f^k(x + 2^t) = f^k(x) + f'^k(2^t), k = 1, 2, ..., t - 1 \ \wedge$
$IsOdd(f^k(x + 2^t)) = IsOdd(f^k(x)), k = 1, 2, ..., t - 1 \ \wedge$
$f^t(x + 2^t) = f^t(x) + f'^t(2^t)$.

Due to Lemma 4.9, $f'^t(2^t) = \frac{3^{CntI'(f'^t)}}{2^{|f'^t|}} * 2^t = \frac{3^{CntI'(f'^t)}}{2^t} * 2^t$
$= 3^{CntI'(f'^t)} \in [1]_2$.

Thus, $IsOdd(f^t(x + 2^t)) \neq IsOdd(f^t(x))$. ∎

Remark If we assume $f^0(x) = Substr(\cdot, \cdot, 0)(x) = x$ and $f'^0(x) = x$, then above lemma can be restated as $IsOdd(f^k(x)) = IsOdd(f^k(x + 2^t))$ and $IsOdd(f^t(x)) \neq IsOdd(f^t(x + 2^t))$ where $f^k = Substr(f^t, 1, k), k = 0, 1, ..., t - 1, t \geq 1$.

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., $x \in [x \mod 2^t]_{2^t}$) are identical. Or, the first t transformations of x is determined by $x \mod 2^t$. Or, all $x \in [x \mod 2^t]_{2^t}$ have the same first t transformations.

Lemma 4.11: If $\mathsf{DYNM}(x, t) = f^t \in \{I, O\}^t$, $x, t \in \mathbb{N}^*$, then
$\mathsf{DYNM}(x \in [i]_{2^t}, t) = f^t$, where $i = x \mod 2^t$.

Proof Let $f^k = Substr(f^t, 1, k), k = 0, 1, 2, ..., t-1$. (Recall Lemma 4.10 Remark.)

(1) By Lemma 4.10 and Remark, $IsOdd(f^k(x)) = IsOdd(f^k(x + 2^t))$.

Thus, $\mathsf{DYNM}(x + 2^t, t) = \mathsf{DYNM}(x, t) = f^t$.

Similarly, $IsOdd(f^k(x + 2^t)) = IsOdd(f^k(x + 2^t + 2^t))$. Thus,
$\mathsf{DYNM}(x + 2^t + 2^t, t) = \mathsf{DYNM}(x + 2^t, t) = f^t$.

Iteratively and hence,
$\mathsf{DYNM}(x + m * 2^t, t) = \mathsf{DYNM}(x, t) = f^t, m \in \mathbb{N}^*$.

(2) Due to the variant of Separation Lemma (i.e., Lemma 4.8),
$IsOdd(f^k(x)) = IsOdd(f^k(x - 2^t))$.

When $x - 2^t > 0$, we have
$\mathsf{DYNM}(x - 2^t, t) = \mathsf{DYNM}(x, t) = f^t$.

Iteratively, when $x - m * 2^t > 0, m \in \mathbb{N}^*$, we have
$\mathsf{DYNM}(x - m * 2^t, t) = \mathsf{DYNM}(x, t) = f^t$.

Therefore, due to (1) and (2),
$\mathsf{DYNM}(x \in [i]_{2^t}, t) = f^t$, where $i = x \mod 2^t$. ∎

Remark
(1) Above lemma includes the special case $i = 0$ or $x \in [0]_2$. However, $\mathsf{DYNM}(x \in [0]_2, 1) = \mathsf{RD}[x \in [0]_2] = O$ is trivial, $\mathsf{DYNM}(x \in [1]_2, n \in \mathbb{N}^*)$ is thus of more interest. That is, we mainly concentrate on
$\mathsf{DYNM}(x \in [i]_{2^t}, n \in \mathbb{N}^*), i, t \in \mathbb{N}^*, 1 \leq i \leq 2^t - 1, i \in [1]_2$.
(2) In above lemmas (Lemma 4.10 and Lemma 4.11), f^t is general. Besides, both reduced dynamics and original dynamics satisfy above lemmas.
(3) Recall that, Lemma 4.11 states that the first t transformations for $x \in [x \mod 2^t]_{2^t}$ are identical. Lemma 4.10 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 $[x \mod 2^t]_{2^t}$ (i.e., $[x \mod 2^t]_{2^{t+1}}, [(x \mod 2^t) + 2^t]_{2^{t+1}}$), respectively. Moreover, all natural numbers in $x \in [x \mod 2^t]_{2^t}$ 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 4.10 and Lemma 4.11 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 4.12: (**Partition Theorem**.)
(1) $\mathsf{DYNM}(x \in [0]_2, 1) = \mathsf{RD}[x \in [0]_2] = O$.
(2) Suppose $\mathsf{DYNM}(i, t) = f^t \in \{I, O\}^t$, $t, i \in \mathbb{N}^*$, $1 \leq i \leq 2^t - 1$.
(2.1) $\mathsf{DYNM}(x \in [i]_{2^t}, t) = f^t$.
(2.2) If $f^t(x) < x$, then $\mathsf{RD}[x \in [i]_{2^t}] = f^t$.
(2.3) If $f^t(x) \not< x$, then
$f^t(i) \in [0]_2 \Rightarrow$
$\mathsf{DYNM}(x \in [i]_{2^{t+1}}, t + 1) = f^t \| O \wedge \mathsf{DYNM}(x \in [i +$

$2^t]_{2^{t+1}}, t+1) = f^t \| I$;
$f^t(i) \in [1]_2 \Rightarrow$
$\mathsf{DYNM}(x \in [i]_{2^{t+1}}, t+1) = f^t \| I \wedge \mathsf{DYNM}(x \in [i + 2^t]_{2^{t+1}}, t+1) = f^t \| O$.

Proof (1) Straightforward.

(2.1) The proof is similar to Lemma 4.10 (2) (except that here $t = 1$ is possible), or by Lemma 4.11 (1).

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

(2.3) We inspect the value of $IsOdd(f'^t(m * 2^t))$ to decide whether
$IsOdd(f^t(i + m * 2^t)) = IsOdd(f^t(i))$ or not, since $f^t(i + m * 2^t) = f^t(i) + f'^t(m * 2^t)$, by Separation Lemma (i.e., Lemma 4.8).

By Lemma 4.9 (recall that $t \in \mathbb{N}^*$),
$f'^t(m * 2^t) = \frac{3^{CntI'(f'^t)}}{2^{|f'^t|}} * m * 2^t = \frac{3^{CntI'(f'^t)}}{2^t} * m * 2^t = 3^{CntI'(f'^t)} * m$.

$m \in [1]_2 \Rightarrow 3^{CntI'(f'^t)} * m \in [1]_2 \Rightarrow IsOdd(f^t(i + m * 2^t)) \neq IsOdd(f^t(i))$.

Otherwise, $m \in [0]_2 \Rightarrow IsOdd(f^t(i + m * 2^t)) = IsOdd(f^t(i))$.

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

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

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

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

Obviously, $x = i$ can be included too, because $x = i \Rightarrow IsOdd(f^t(x)) = IsOdd(f^t(i))$. Therefore, $x \in [i]_{2^{t+1}} \Rightarrow IsOdd(f^t(x)) = IsOdd(f^t(i))$.

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

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

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

Remark
(1) The theorem is called "Partition Theorem" because $[i]_{2^t}$ is partitioned for deciding $(t+1)$-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 $x \in [0]_2 \cup [i]_{2^t}, t \in \mathbb{N}^*, 1 \leq i \leq 2^t - 1, i \in [1]_2, i = x \mod 2^t$.
For $x \in [0]_2$, the further partition is not of interest because $\mathsf{RD}[x \in [0]_2] = O$ 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 4.10). More specifically, to determine whether $f^t(x) \in [1]_2$ 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 $\mathsf{RD}[x]$ and $\mathsf{DYNM}(x, t)$ as follows: Suppose $\mathsf{DYNM}(x, t) = f^t \in \{I, O\}^t$. If $f^t(x) < x$, then $\mathsf{DYNM}(x, t) = \mathsf{RD}[x] = f^t$ (which means current transformed number has already been less than starting number for the *first* time); Otherwise, $\mathsf{DYNM}(x, t+1) = f^t \| I$ or $\mathsf{DYNM}(x, t+1) = f^t \| O$.

In following notation f^a_b, "a" represents its length, and "b" differentiates among $f^a \in \{I, O\}^a$. Next corollary states that $i \in [0, 2^t - 1]$ determines the first t transformations of all $x \in [i]_{2^t}$.

Corollary 4.13: If $\mathsf{DYNM}(x \in [i_1]_{2^t}, t) = f^t_{i_1}$, $\mathsf{DYNM}(x \in [i_2]_{2^t}, t) = f^t_{i_2}$. $t, i_1, i_2 \in \mathbb{N}^*$, $0 \leq i_1, i_2 \leq 2^t - 1$, then $i_1 = i_2 \Leftrightarrow f^t_{i_1} = f^t_{i_2}$.

Proof (1) ($\Rightarrow$).

(1.1) $t \geq 2$, thus $1 \leq i_1, i_2 \leq 2^t - 1$ due to Theorem 4.12 (2) and Remark (2).

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

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

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

(2) ($\Leftarrow$). It is proved by proving converse negative proposition.

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

$i_1 \neq i_2$, $1 \leq i_1, i_2 \leq 2^t - 1 \Rightarrow \mathsf{DYNM}(i_1, t) \neq \mathsf{DYNM}(i_2, t)$
$\Rightarrow \mathsf{DYNM}(x \in [i_1]_{2^t}, t) \neq \mathsf{DYNM}(x \in [i_2]_{2^t}, t)$.
$\because \mathsf{DYNM}(x \in [i_1]_{2^t}, t) = \mathsf{DYNM}(i_1, t), \mathsf{DYNM}(x \in [i_2]_{2^t}, t) = \mathsf{DYNM}(i_2, t)$.

(2.2) $t = 1$.

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

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

Corollary 4.14: $\mathsf{DYNM}(x_1, t) = \mathsf{DYNM}(x_2, t) \Leftrightarrow x_1 \equiv x_2 \mod 2^t$, where $x_1, x_2, t \in \mathbb{N}^*, x_1 \neq x_2$.

Proof Straightforward due to Corollary 4.13. ∎

Next corollary states that 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 $x_1 \equiv x_2 \mod 2^t$ and $x_1 \equiv x_2 + 2^t \mod 2^{t+1}$. As $\mathsf{DYNM}(x \in [0]_2, 1) = \mathsf{RD}[x \in [0]_2] = O$, only $x \in [1]_2$ is of interests.

Corollary 4.15: (**Forking Corollary**.)
$\mathsf{DYNM}(x_1, t) = \mathsf{DYNM}(x_2, t) \wedge$
$\mathsf{DYNM}(x_1, t+1) \neq \mathsf{DYNM}(x_2, t+1), x_1, x_2 \in [1]_2, x_1 \neq x_2, t \in \mathbb{N}^*$
$\Leftrightarrow x_1 \equiv x_2 \mod 2^t \wedge x_1 \equiv x_2 + 2^t \mod 2^{t+1}$.

Proof It is straightforward due to Corollary 4.14. ∎