Collatz Tree Expansions and Equivalence under Compression

Farhad Aliabdali

The Collatz map $T(n) = n/2$ for even n and $T(n) = 3n + 1$ for odd n admits classical affine descriptions via parity vectors, but these typically compress each odd event into the macro-step $(3n + 1)/2$, obscuring intermediate algebraic states. We introduce a two-stage expansion that separates an odd event into a rewrite step R (expressing $n = 2x + 1$) followed by a forced follow-up C (sending $x \mapsto 3x + 2$), alongside the even halving step E. This yields a word system over $\{E, R, C\}$ and a uniform normal form

$$X_N(w) = \frac{3^{k(w)}X_0 + 2^{D(w)} - 3^{k(w)} + \sigma_N(w)}{2^{D(w)}},$$

where $\sigma_N(w)$ admits an explicit signed monomial expansion in powers of 3 and 2. We prove that complete two-stage words (those with every R immediately followed by C) compress under $RC \mapsto O$ to the standard parity-vector affine form, giving a precise equivalence criterion and a canonical matching rule (k, D, Σ). Consequently, removing the standard-image equations from the two-stage enumeration leaves exactly the truncated (dangling-R) equations corresponding to intermediate states not representable in the standard form. Finally, we derive residue-class "locking" conditions modulo $2^{D(w)}$, clarifying integrality constraints and connecting the framework naturally to 2-adic formulations.

Introduction

This manuscript is an *algebraic/combinatorial* study of Collatz iterates: it introduces a two-stage branching formalism that makes intermediate states explicit, provides a canonical deduplication rule that recovers the standard affine "parity-vector" form, and reformulates integrality constraints as residue-class conditions modulo powers of 2, naturally connecting the framework to 2-adic viewpoints. No claim is made here to resolve the Collatz conjecture; rather, the goal is to supply a clean normal form and bookkeeping tools that can support cycle- and structure-focused investigations.

Motivation for the two-stage expansion.

In the usual shortcut form, an odd event is immediately compressed into $(3x + 1)/2$, which hides an intermediate "even-base" representation $x = 2y + 1$ and the forced follow-up producing $2(3y + 2)$. By separating these stages into the symbols R (rewrite) and C (forced follow-up), alongside E (halving), the two-stage tree tracks intermediate nodes that are otherwise invisible and reveals systematic algebraic redundancies.

Context and related work.

Affine descriptions in terms of parity words (or parity vectors) and their associated linear-fractional maps are classical in the literature; see Terras' stopping-time analysis and the survey of Lagarias for broader context. The extension of Collatz dynamics to the 2-adic integers and conjugacy-based formulations are also well developed; see Wirsching and Bernstein . Our contribution is orthogonal to these works: we supply a two-stage normal form that (i) makes the intermediate states explicit, (ii) yields an explicit monomial expansion for $\sigma_N(w)$, and (iii) gives an exact and computable compression-equivalence criterion via the compression map $RC \mapsto O$.

Contributions.

- **Two-stage word model:** a ternary alphabet $\{E, R, C\}$ with a clean distinction between complete (admissible) and truncated words, encoding intermediate states.
- **Closed normal form:** a uniform affine expression for $X_N(w)$ and an explicit monomial-sum representation of $\sigma_N(w)$.
- **Compression and equivalence theorem (core novelty):** complete two-stage words compress under $RC \mapsto O$ to the standard affine form, yielding a rigorous deduplication rule and canonical matching triple (k, D, Σ).
- **Residue-class locking:** for each finite route word, integrality of $X_N(w)$ is equivalent to membership of X_0 in a unique residue class modulo $2^{D(w)}$, connecting naturally to 2-adic formulations.

Significance and potential applications

The two-stage refinement is introduced to increase the resolution of symbolic bookkeeping. Making intermediate (dangling-R) states explicit provides:

a canonical way to enumerate and deduplicate affine equations via the compression map RC ↦ O, while preserving compatibility with classical parity-vector data;

a clean separation between complete words (standard-image equations) and truncated words (intermediate states), useful when auditing equation-generation pipelines or comparing normal forms;

route-dependent residue-class locking conditions modulo powers of 2 that can be read directly from $\sigma N(w)$, supporting cycle-candidate filtering and 2-adic consistency checks.

Classical vs. novel components.

Several ingredients are classical, notably the existence of affine forms attached to parity vectors and standard modular constraints. We re-derive these where needed for completeness and to keep the exposition self-contained. The novelty here is structural: the explicit R/C refinement of odd events, the induced classification of intermediate (truncated) states, and the proof that the standard affine equations arise *exactly* as the compressed image of complete two-stage words, enabling a rigorous compression-equivalence.

$$X_0\begin{cases} if\ (Even\ )\xrightarrow{\frac{1}{2}}(Even\ or\ Odd\ ) & \begin{cases} if\ (Even\ )\xrightarrow{\frac{1}{2}} & \begin{cases} if\ (Even\ )\xrightarrow{\frac{1}{2}}(Even\ or\ Odd\ )\\ if\ (Odd\ )\xrightarrow{\times 3+1}(Even)\end{cases}\\ if\ (Odd\ )\xrightarrow{\times 3+1}(Even)= & \begin{cases} if\ (Even\ )\xrightarrow{\frac{1}{2}}(Even\ or\ Odd\ )\\ (Even\ \neq Odd)\end{cases}\end{cases}\\ if\ (Odd\ )\xrightarrow{\times 3+1}(Even)= & \begin{cases} if\ (Even\ )\xrightarrow{\frac{1}{2}}(Even\ or\ Odd\ ) & \begin{cases} if\ (Even\ )\xrightarrow{\frac{1}{2}}(Even\ or\ Odd\ )\\ if\ (Odd\ )\xrightarrow{\times 3+1}(Even)\end{cases}\\ (Even\ \neq Odd)= & \begin{cases}[-]\\ [-]\end{cases}\end{cases}\end{cases}$$

$$X_0\begin{cases} 2X_1\xrightarrow{\frac{1}{2}}X_1= & \begin{cases} 2X_2\xrightarrow{\frac{1}{2}}X_2= & \begin{cases} 2X_3\xrightarrow{\frac{1}{2}}X_3\\ 2X_3+1\xrightarrow{\times 3+1}2(3X_3+2)\end{cases}\\ 2X_2+1\xrightarrow{\times 3+1}2(3X_2+2)= & \begin{cases} 2X_3\xrightarrow{\frac{1}{2}}X_3\\ 2X_3+1\ (Even\ \neq Odd)\end{cases}\end{cases}\\ 2X_1+1\xrightarrow{\times 3+1}\mathbf{2(3X_1+2)}= & \begin{cases} 2X_2\xrightarrow{\frac{1}{2}}X_2= & \begin{cases} 2X_3\xrightarrow{\frac{1}{2}}X_3\\ 2X_3+1\xrightarrow{\times 3+1}2(3X_3+2)\end{cases}\\ \mathbf{2X_2+1(Even\ \neq Odd)}= & \begin{cases}[-]\\ [-]\end{cases}\end{cases}\end{cases}$$

$$X_0\begin{cases} \frac{X_0}{2}=X_1 & \begin{cases} \frac{X_1}{2}=X_2 & \begin{cases} \frac{X_2}{2}=X_3\\ \frac{X_2-1}{2}=X_3\end{cases}\\ \frac{X_1-1}{2}=X_2 & \begin{cases} 3X_2+2=X_3\\ (Even\ \neq Odd)\end{cases}\end{cases}\\ \frac{X_0-1}{2}=X_1 & \begin{cases} 3X_1+2=X_2 & \begin{cases} \frac{X_2}{2}=X_3\\ \frac{X_2-1}{2}=X_3\end{cases}\\ (Even\ \neq Odd) & \begin{cases}[-]\\ [-]\end{cases}\end{cases}\end{cases}$$

$$X_0\begin{cases} \frac{X_0}{2}=X_1 & \begin{cases} \frac{X_0}{2^2}=X_2 & \begin{cases} \frac{X_0}{2^3}=X_3\\ \frac{X_0-2^2}{2^3}=X_3\end{cases}\\ \frac{X_0-2}{2^2}=X_2 & \begin{cases} \frac{3X_0-3\times 2+2^3}{2^2}=X_3\\ (Even\ \neq Odd)\end{cases}\end{cases}\\ \frac{X_0-1}{2}=X_1 & \begin{cases} \frac{3X_0-3+2^2}{2}=X_2 & \begin{cases} \frac{3X_0-3+2^2}{2^2}=X_3\\ \frac{3X_0-3+2^2-2}{2^2}=X_3\end{cases}\\ (Even\ \neq Odd) & \begin{cases}[-]\\ [-]\end{cases}\end{cases}\end{cases}$$

- It shows that X_n splits into X_{n+1}which splits intoX_{n+2} ...X_N.
- The notation $\{\frac{X_0-1}{2} = X_1, 3X_2 + 2 = X_3\}$ identifies that "Odd" branch, the number grows $\left\{\ldots \rightarrow \frac{1}{2} \rightarrow 3 \rightarrow \cdots\right\}$ and "Even" branch decay $\left\{\ldots \rightarrow \frac{1}{2} \rightarrow \frac{1}{2} \rightarrow \cdots\right\}$ rule derived above.

The symbols $(Even \neq Odd)$ and [-] represent a number cannot be both even and odd simultaneously. This confirms the Determinism: a number has only one valid path through the tree.

The Growth map:

- 1/2: Represents a decay step.
- 3: Represents a growth step.

The structure ½ –> $\left\{\frac{1}{2}, 3\right\}$ shows that after a decay step, the number is even or odd.

The structure $\left(\frac{1}{2} \text{–>} \text{ ½ –> ½}\right)$show that for a number to decay continuously, it routes specific sequence of "Even" checks.

Evolution of the integer index $\begin{cases} \frac{1}{2} & \begin{cases} \frac{1}{2} \\ \frac{1}{2} \end{cases} \\ \frac{1}{2} & \begin{cases} 3 \\ - \end{cases} \end{cases}$

Standard vs. two-stage branching. In the standard tree, nodes branch via E (even step) and O (odd macro-step). In the two-stage tree, odd events are expanded into R (rewrite) followed by forced C; the composite RC path compresses to O. Truncated nodes ending in R correspond to intermediate states absent in the standard representation.

Section 2 defines the two-stage operations and word model. Section 3 proves the closed affine normal form and derives the explicit monomial expansion for $\sigma_N(w)$. Section 4 formalizes the compression map $RC \mapsto O$ and the compression-equivalence criterion. Section 5 discusses cycle equations and includes worked examples. Section 6 develops residue-class (and 2-adic) constraints for fixed route words. We close with directions for further work.

Two-stage operations and branch words

Two-stage operations

Let $(X_n)_{n\geq 0}$ be a sequence of reals (eventually specialized to integers/rationals). We define the two-stage branching operations:

- **(E)** If X_n is even, write $X_n = 2X_{n+1}$ so that

$$X_{n+1} = \frac{X_n}{2}.$$

- **(R then C)** If X_n is odd, write $X_n = 2X_{n+1} + 1$, equivalently

  $$\text{(R)} \quad X_{n+1} = \frac{X_n - 1}{2}.$$

  Then apply the forced follow-up

  $$\text{(C)} \quad X_{n+2} = 3X_{n+1} + 2,$$

  which is consistent with $3(2X_{n+1} + 1) + 1 = 2(3X_{n+1} + 2)$.

Words and admissibility

Definition 1 (Branch word). A branch is encoded by a finite word $w = w_0 w_1 \cdots w_{N-1}$ over the alphabet $\{E, R, C\}$.

Definition 2 (Admissible (complete) and truncated words). A word is *admissible/complete* if every occurrence of R is immediately followed by C. A word is *truncated* if it ends in R (so it represents an intermediate "needs C next" node).

Counters

Definition 3 (Counters D and k). For a word w, define

$$D(w) := \#\{t : w_t \in \{E, R\}\}, \qquad k(w) := \#\{t : w_t = C\}.$$

For prefixes $w^{(t)} := w_0 \cdots w_{t-1}$ we write $D_t := D\left(w^{(t)}\right)$ and $k_t := k\left(w^{(t)}\right)$.

Two-stage closed form and proof for all nodes

Theorem 1 (Two-stage affine closed form). *For every word w of length N (admissible or truncated), there exists an integer expression $\sigma_N(w)$ representable as a signed sum of monomials $\pm 3^a 2^b$ such that* $\boxed{X_N(w) = \frac{3^{k(w)} X_0 + 2^{D(w)} - 3^{k(w)} + \sigma_N(w)}{2^{D(w)}}.}$

Proof. We induct on N.

Base $N = 0$. For the empty word $\varnothing$ we have $D(\varnothing) = k(\varnothing) = 0$. Setting $\sigma_0(\varnothing) = 0$ yields $X_0 = X_0$ in [eq:two-stage-closed].

Inductive step. Assume [eq:two-stage-closed] holds for a word w of length N, and denote its parameters by

$$D := D(w), \quad k := k(w), \quad \sigma := \sigma_N(w), \quad X := X_N(w) = \frac{3^k X_0 + 2^D - 3^k + \sigma}{2^D}.$$

We show the form is preserved under appending one symbol.

(i) Append E**.** Then $X' = \frac{X}{2}$, so

$$X' = \frac{3^k X_0 + 2^D - 3^k + \sigma}{2^{D+1}} = \frac{3^k X_0 + 2^{D+1} - 3^k + (\sigma - 2^D)}{2^{D+1}}.$$

Hence $D' = D + 1$, $k' = k$, and $\sigma' = \sigma - 2^D$.

(ii) Append R**.** Then $X' = \frac{X-1}{2}$, so

$$X' = \frac{3^k X_0 + 2^D - 3^k + \sigma - 2^D}{2^{D+1}} = \frac{3^k X_0 + 2^{D+1} - 3^k + (\sigma - 2^{D+1})}{2^{D+1}}.$$

Hence $D' = D + 1$, $k' = k$, and $\sigma' = \sigma - 2^{D+1}$.

(iii) Append C**.** Then $X' = 3X + 2$, so

$$X' = \frac{3^{k+1} X_0 + 3(2^D - 3^k + \sigma) + 2^{D+1}}{2^D} = \frac{3^{k+1} X_0 + 2^D - 3^{k+1} + (3\sigma + 2^{D+2})}{2^D}.$$

Hence $D' = D$, $k' = k + 1$, and $\sigma' = 3\sigma + 2^{D+2}$.

Thus the invariant form [eq:two-stage-closed] holds for all allowed extensions, completing the induction. □

Example 1 (Worked word $w = RCE$). Let $w = RCE$. Starting from X_0, the two-stage updates give

$$X_1 = \frac{X_0 - 1}{2} \quad (R), \qquad X_2 = 3X_1 + 2 = \frac{3X_0 + 1}{2} \quad (C), \qquad X_3 = \frac{X_2}{2} = \frac{3X_0 + 1}{4} \quad (E).$$

For this word one has $D(w) = 2$ (letters R and E) and $k(w) = 1$ (letter C). The closed form [eq:two-stage-closed] therefore predicts

$$X_3 = \frac{3^1 X_0 + 2^2 - 3^1 + \sigma_3(w)}{2^2} = \frac{3X_0 + 1 + \sigma_3(w)}{4}.$$

Comparing with $X_3 = (3X_0 + 1)/4$ yields $\sigma_3(w) = 0$. This also follows from the monomial sum [eq:sigma-sum]: the three letter-contributions are

$$(R){:}\ -3^{1-0}2^{0+1} = -6, \qquad (C){:}\ +3^{1-1}2^{1+2} = +8, \qquad (E){:}\ -3^{1-1}2^{1} = -2,$$

which sum to 0.

Explicit monomial sum for $\sigma_N(w)$

Proposition 1 (Monomial sum representation). *Let w be a word of length N and let (D_t, k_t) be the prefix counters. Then $\sigma_N(w)$ can be written explicitly as*

$$\boxed{\sigma_N(w) = \sum_{t:\, w_t = E} \left(-\,3^{k_N - k_t}\, 2^{D_t}\right) + \sum_{t:\, w_t = R} \left(-\,3^{k_N - k_t}\, 2^{D_t + 1}\right) + \sum_{t:\, w_t = C} \left(+\,3^{k_N - k_{t+1}}\, 2^{D_t + 2}\right),}$$ *where $k_N := k(w)$.*

Proof. We proceed by induction on N using the update rules for σ proved in Theorem 1.

Base case $N = 0$. For the empty word $\varnothing$, all sums are empty and [eq:sigma-sum] gives $\sigma_0(\varnothing) = 0$.

Inductive step. Assume [eq:sigma-sum] holds for a word w of length N with counters $D_N := D(w)$ and $k_N := k(w)$. Consider a one-letter extension $w' := wa$ of length $N + 1$.

Case $a = E$. Then $k_{N+1}(w') = k_N(w)$ and $D_{N+1}(w') = D_N(w) + 1$, and Theorem 1 gives $\sigma_{N+1}(w') = \sigma_N(w) - 2^{D_N(w)}$. In the right-hand side of [eq:sigma-sum], all contributions from positions $t < N$ are unchanged because the exponent $k_{N+1} - k_t$ equals $k_N - k_t$. A single new term appears at $t = N$ in the E-sum:

$$-\,3^{\,k_{N+1}-k_N}\,2^{\,D_N} = -\,3^0 2^{D_N} = -2^{D_N},$$

which matches the recursion.

Case $a = R$. Again $k_{N+1} = k_N$ and $D_{N+1} = D_N + 1$, while Theorem 1 yields $\sigma_{N+1}(w') = \sigma_N(w) - 2^{D_N(w)+1}$. As above, all existing terms (for $t < N$) are unchanged, and the new term at $t = N$ now lies in the R-sum:

$$-\,3^{\,k_{N+1}-k_N}\,2^{\,D_N+1} = -\,3^0 2^{D_N+1} = -2^{D_N+1},$$

agreeing with the recursion.

Case $a = C$. Now $k_{N+1} = k_N + 1$ and $D_{N+1} = D_N$, and Theorem 1 yields $\sigma_{N+1}(w') = 3\sigma_N(w) + 2^{D_N(w)+2}$. In [eq:sigma-sum], all contributions from positions $t < N$ acquire one extra factor of 3 because $k_{N+1} - k_t = (k_N - k_t) + 1$ and likewise $k_{N+1} - k_{t+1} = (k_N - k_{t+1}) + 1$. Thus the sum over $t < N$ becomes $3\sigma_N(w)$. The new terminal letter contributes one additional term at $t = N$ in the C-sum:

$$+\,3^{\,k_{N+1}-k_{N+1}}\,2^{\,D_N+2} = +\,3^0 2^{D_N+2} = 2^{D_N+2},$$

yielding exactly $\sigma_{N+1}(w') = 3\sigma_N(w) + 2^{D_N+2}$.

These three cases cover all extensions allowed in the two-stage generation, completing the induction. □

Cycle equation in two-stage form

Proposition 2 (Cycle equation). *Let w be any word of length N and define $D := D(w)$, $k := k(w)$, and $\sigma := \sigma_N(w)$. Then the fixed-point condition $X_N(w) = X_0$ is equivalent to* $\boxed{X_0 = 1 + \frac{\sigma}{2^D - 3^k}.}$ *In particular,* $X_0 \in \mathbb{Z} \quad \Leftrightarrow \quad 2^D - 3^k \mid \sigma$.

Proof. Set $X_N(w) = X_0$ in [eq:two-stage-closed] and rearrange:

$$X_0 = \frac{3^k X_0 + 2^D - 3^k + \sigma}{2^D} \Leftrightarrow (2^D - 3^k)X_0 = 2^D - 3^k + \sigma \Leftrightarrow X_0 = 1 + \frac{\sigma}{2^D - 3^k}.$$

The divisibility criterion follows immediately. □

Standard Collatz form as a compression of the two-stage tree

Standard affine form

A standard Collatz parity sequence yields an affine expression

$$\boxed{X_N = \frac{3^k X_0 + \Sigma}{2^D}}$$

for integers k, D, Σ.

Compression map $\boldsymbol{RC} \mapsto \boldsymbol{O}$

Definition 4 (Compression). Define a partial map $\pi\colon \{E, R, C\}^* \to \{E, O\}^*$ by $\pi(E) = E$ and $\pi(RC) = O$, extended by concatenation. It is defined precisely on admissible (complete) words (no dangling final R).

Proposition 3 (Equivalence on complete words). *Let w be complete and let $D := D(w)$ and $k := k(w)$. Define* $\boxed{\Sigma_N(w) := 2^D - 3^k + \sigma_N(w).}$ *Then the two-stage form [eq:two-stage-closed] becomes exactly the standard affine form:* $\boxed{X_N(w) = \frac{3^k X_0 + \Sigma_N(w)}{2^D}.}$ *Moreover this affine map matches the standard map associated to the compressed word $\pi(w)$.*

Proof. Substitute [eq:Sigma-def] into [eq:two-stage-closed]:

$$3^k X_0 + 2^D - 3^k + \sigma_N(w) = 3^k X_0 + \Sigma_N(w).$$

Thus $X_N(w) = \left(3^k X_0 + \Sigma_N(w)\right)/2^D$, which is [eq:standard-affine]. Since D counts E and R steps and k counts C steps, on complete words these agree with the corresponding counts in the compressed word $\pi(w)$. □

Why some equations are removed (equivalence)

Proposition 4 (Redundancy of complete two-stage equations). *Every complete two-stage equation generated by [eq:two-stage-closed] is algebraically identical to a standard Collatz affine equation after the change of constant $\Sigma = 2^D - 3^k + \sigma$. Therefore, removing all complete-word equations from the two-stage list removes no affine maps beyond those already represented in the standard list; it performs a deduplication.*

Proof. This is immediate from Proposition 3: on complete words w the two-stage representation is exactly the standard representation [eq:standard-affine]. Thus the complete-word portion of the two-stage list is contained in the standard list (up to parameter relabeling of the constant). □

Corollary 1 (Characterization of the "leftover" equations). *After removing the standard-equation matches (i.e. all complete words), the remaining equations correspond precisely to truncated words that end in a dangling R.*

Proof. A word is compressible by π if and only if every R is paired with a following C, i.e. the word is complete. If a word ends in a dangling R, it encodes an intermediate "even-base" state immediately after rewriting an odd value as $2X + 1$, before applying the forced follow-up C. Such intermediate states are not representable by the standard parity-vector (complete-step) affine form, and therefore remain after removing the complete-word redundancies. Conversely, every non-compressible word produced by the two-stage generation must end with such a dangling R. □

Canonical matching rule (implementation)

To decide whether a two-stage equation matches a standard equation, convert it to the canonical triple

$$(k,\ D,\ \Sigma) \quad \text{where} \quad \Sigma := 2^D - 3^k + \sigma.$$

Two equations match if and only if these triples coincide (equivalently, they define the same affine map).

Strictly monotone growth along consecutive odd macro-steps

This section isolates a *restricted* regime: trajectories whose evolution consists of consecutive odd→even macro-steps only. Algebraically, this corresponds to iterating the map

$$O(x) := \frac{3x + 1}{2},$$

and additionally requiring that every intermediate value remains odd (so that each application of O corresponds to a legitimate odd-step in the Collatz dynamics). In this regime the odd subsequence is strictly increasing, since $O(x) > x$ for all $x > -1$.

$$X_0 \begin{cases} 2X_1 \xrightarrow{\frac{1}{2}} X_1 = \begin{cases} \xrightarrow{\frac{1}{2}} = & \begin{cases} 2X_2 \xrightarrow{\frac{1}{2}} X_2 \\ 2X_2 + 1 \xrightarrow{\times 3\ +1} 2(3X_2 + 2) \end{cases} \\ \xrightarrow{\times 3\ +1} = & \begin{cases} 2X_2 \xrightarrow{\frac{1}{2}} X_2 \\ 2X_2 + 1\ (Even\ \neq Odd) \end{cases} \end{cases} \\ 2X_1 + 1 \xrightarrow{\times 3\ +1} 2(3X_1 + 2) = \begin{cases} \xrightarrow{\frac{1}{2}} = & \begin{cases} 2X_2 \xrightarrow{\frac{1}{2}} X_2 \\ 2X_2 + 1 \xrightarrow{\times 3\ +1} 2(3X_2 + 2) \end{cases} \\ (Even\ \neq Odd) = & \begin{cases} [-] \\ [-] \end{cases} \end{cases} \end{cases}$$

Closed form for the N-step odd-macro iterate

Proposition 5 (Odd-macro closed form). *For any* $N \geq 0$ *and any* $x \in \mathbb{Q}$, $O^N(x) = \frac{3^N x + \sum_{n=1}^{N} 3^{N-n} 2^{n-1}}{2^N} = (x+1)\left(\frac{3}{2}\right)^N - 1.$

Proof. A straightforward induction on N using $O(x) = (3x+1)/2$ yields the first equality. The summation is a finite geometric series:

$$\sum_{n=1}^{N} 3^{N-n}\, 2^{n-1} = \sum_{j=0}^{N-1} 3^{N-1-j}\, 2^j = \frac{3^N - 2^N}{3-2} = 3^N - 2^N,$$

which gives the second equality in [eq:O-iterate]. □

Congruence characterization for consecutive odd terms

Theorem 2 (Consecutive odd-step constraint). *Fix* $N \geq 1$. *Let* $x_0 \in \mathbb{Z}$ *be odd and define* $x_{n+1} = O(x_n)$ *for* $0 \leq n \leq N-1$. *Then the following are equivalent:*

1. $x_0, x_1, \dots, x_{N-1}$ *are all odd (i.e. N consecutive odd Collatz steps occur).*
2. $x_0 \equiv -1 \pmod{2^{N+1}}$ *(equivalently,* $2^{N+1} \mid (x_0+1)$*).*

In particular, the set of integer starts that realize N consecutive odd steps is exactly $\{\, x_0 = 2^{N+1}m - 1 :\ m \in \mathbb{Z} \,\}$.

Proof. (ii)⇒(i). Write $x_0 = 2^{N+1}m - 1$. Using [eq:O-iterate] with $x = x_0$ gives

$$x_j = O^j(x_0) = (x_0+1)\left(\frac{3}{2}\right)^j - 1 = 2^{N+1}m \cdot \frac{3^j}{2^j} - 1 = 3^j 2^{N+1-j} m - 1.$$

For $0 \leq j \leq N$, the quantity $3^j 2^{N+1-j} m$ is even, hence x_j is odd. In particular $x_0, \dots, x_{N-1}$ are odd.

(i)⇒(ii). Since $x_N = O^N(x_0)$ is obtained from N successive odd steps, we have $x_N \in \mathbb{Z}$ and x_N is odd. Using [eq:O-iterate] we can write

$$x_N = \frac{3^N(x_0+1) - 2^N}{2^N}.$$

The condition that x_N is odd is equivalent to

$$3^N(x_0+1) - 2^N \equiv 2^N \pmod{2^{N+1}},$$

i.e.

$$3^N(x_0+1) \equiv 2^{N+1} \equiv 0 \pmod{2^{N+1}}.$$

Since 3^N is invertible modulo 2^{N+1}, it follows that $x_0 + 1 \equiv 0 \pmod{2^{N+1}}$, proving $x_0 \equiv -1 \pmod{2^{N+1}}$.

Corollary 2 (No infinite all-odd growth from a natural start). *There is no $x_0 \in \mathbb{N}$ for which the Collatz trajectory exhibits infinitely many consecutive odd steps (i.e. an infinite iterate of [eq:Omap] with all intermediate values odd). Equivalently, the unique* 2*-adic solution to the nested congruences $x_0 \equiv -1 \pmod{2^{N+1}}$ for all N is the* 2*-adic integer $x_0 = -1$, which is not a natural number.*

Proof. If $x_0 \in \mathbb{N}$ produced N consecutive odd steps for every N, then by Theorem 2 we would have $x_0 \equiv -1 \pmod{2^{N+1}}$ for all $N \geq 1$. The only 2-adic integer satisfying these nested congruences is -1, which is not in $\mathbb{N}$. □

Remark 1 (Scope). The results of this section rule out an *infinite run of consecutive odd steps* from a natural start. They do not, by themselves, exclude the possibility of other forms of divergence that involve mixed parity patterns.

Residue-class constraints for fixed two-stage routes

This section reformats the "mixed route" congruence statements in a precise way that is suitable for publication. The key idea is that *fixing a finite route word w* forces the starting value X_0 to lie in a specific residue class modulo $2^{D(w)}$ in order for the resulting $X_N(w)$ to be integral.

Integrality as a congruence in the two-stage normal form

Recall the two-stage closed form from Theorem 1:

$$X_N(w) = \frac{3^{k(w)}X_0 + 2^{D(w)} - 3^{k(w)} + \sigma_N(w)}{2^{D(w)}}.$$

Lemma 1 (Invertibility of odd integers modulo powers of two). *If a is odd and $D \geq 1$, then* $\gcd(a, 2^D) = 1$*, hence there exists an integer a^{-1} such that $a\,a^{-1} \equiv 1 \pmod{2^D}$. In particular, $(3^k)^{-1} \bmod 2^D$ exists for every $k \geq 0$.*

Proposition 6 (Integrality criterion and residue class). *Fix a word w of length N and write $D := D(w)$ and $k := k(w)$. Then $X_N(w) \in \mathbb{Z}$ if and only if $3^k(X_0 - 1) + \sigma_N(w) \equiv 0 \pmod{2^D}$. Equivalently, since* $\gcd(3^k, 2^D) = 1$*, there is a unique residue class $C(w) \in \mathbb{Z}/2^D\mathbb{Z}$ such that*

$$\boxed{X_0 \equiv 1 - \sigma_N(w) \cdot (3^k)^{-1} \pmod{2^D}.}$$

Proof. The value $X_N(w)$ is integral if and only if the numerator is divisible by 2^D. Reducing the numerator modulo 2^D eliminates the $+2^D$ term, giving [eq:integrality-congruence]. Since 3^k is odd, it is invertible modulo 2^D, so [eq:X0-residue] follows by multiplying [eq:integrality-congruence] by $(3^k)^{-1}$ modulo 2^D.

Remark 2 (Complete-word (standard) form). If w is complete, one may equivalently use the standard constant $\Sigma_N(w) = 2^D - 3^k + \sigma_N(w)$ from Proposition 3. Then

$$X_N(w) = \frac{3^k X_0 + \Sigma_N(w)}{2^D}, \qquad X_N(w) \in \mathbb{Z} \Leftrightarrow 3^k X_0 + \Sigma_N(w) \equiv 0 \pmod{2^D},$$

and hence $X_0 \equiv -\Sigma_N(w)\,(3^k)^{-1} \pmod{2^D}$.

Nested residue classes and 2-adic limits

Consider an infinite admissible route (parity pattern) represented by an infinite word w_∞ and let $w^{(N)}$ denote its length-N prefix. For each N, Proposition 6 defines a residue class

$$X_0 \equiv C(w^{(N)}) \pmod{2^{D(w^{(N)})}}$$

that is necessary and sufficient for $X_N(w^{(N)})$ to be integral.

Proposition 7 (2-adic consistency). *Assume $D(w^{(N)}) \to \infty$ as $N \to \infty$. If the congruences $X_0 \equiv C\left(w^{(N)}\right) \pmod{2^{D(w^{(N)})}}$ are mutually consistent (i.e. each refines the previous), then they determine a unique 2-adic integer $X_0^{(2)} \in \mathbb{Z}_2$. Any integer $X_0 \in \mathbb{Z}$ satisfying all these congruences must equal that 2-adic limit.*

Proof. A nested sequence of residue classes modulo 2^{m_N} with $m_N \to \infty$ determines a unique element of $\mathbb{Z}_2$. If an ordinary integer satisfies all congruences, it represents the same element of $\mathbb{Z}_2$, hence must coincide with the limit. □

Remark 3 (What this does and does not imply). The residue-class constraint provides a strong *arithmetic locking* of possible starts for any fixed finite route. However, the existence (or nonexistence) of an ordinary integer in the corresponding 2-adic class for a given infinite route is a separate question. In particular, these congruence constraints alone do not by themselves preclude all forms of divergence; they reformulate the problem in 2-adic terms.

Conclusion

We introduced a two-stage refinement of Collatz branching in which even halving is represented by E, while each odd event is decomposed into a rewrite step R(expressing $n = 2x + 1$) followed by a forced follow-up C(sending $x \mapsto 3x + 2$). This yields a word model over $\{E, R, C\}$and a uniform affine normal form

$$X_N(w) = \frac{3^{k(w)}X_0 + 2^{D(w)} - 3^{k(w)} + \sigma_N(w)}{2^{D(w)}},$$

together with an explicit signed monomial expansion for the offset $\sigma_N(w)$. We then proved that complete two-stage words compress under $RC \mapsto O$to recover the classical parity-vector affine form, establishing a precise equivalence (and hence a canonical deduplication rule) between the high-resolution two-stage enumeration and the standard low-resolution description.

Beyond providing a consistency bridge to classical formulations, the two-stage view isolates the genuinely new intermediate (truncated) states—words ending in a dangling R—that have no direct representation in the standard model. The residue-class locking criterion for integrality, expressed as a congruence modulo $2^{D(w)}$, places these equations naturally into a 2-adic framework and offers a practical interface for modular filtering of route candidates. This suggests several directions for further work, including computationally efficient cycle-candidate screening based on (k, D, σ)invariants, refined coverage statements linking truncated states to complete trajectories, and deeper links between nested congruence classes and periodicity in $\mathbb{Z}_2$.

Notes for further work

The results above establish the algebraic equivalence of the two-stage and standard affine forms (on complete words) and explain the compression-equivalence step. To use this framework for cycle exclusion, one must specify (and prove) a coverage statement ensuring that all integer cycle candidates correspond to complete words (or are otherwise represented in the retained set).

J. C. Lagarias, *The* $3x + 1$ *problem and its generalizations*, The American Mathematical Monthly 92(1) (1985), 3–23. doi:10.2307/2322189.

J. C. Lagarias, *The* $3x + 1$ *Problem: An Overview* (as of 2010), available as arXiv:2111.02635 (version posted 2021).

R. Terras, *A stopping time problem on the positive integers*, Acta Arithmetica 30(3) (1976), 241–252. doi:10.4064/aa-30-3-241-252.

G. J. Wirsching, *The Dynamical System Generated by the* $3n + 1$ *Function*, Lecture Notes in Mathematics 1681, Springer, 1998. doi:10.1007/BFb0095985.

D. J. Bernstein, *A non-iterative* 2*-adic statement of the* $3x + 1$ *conjecture*, Proceedings of the American Mathematical Society 121(2) (1994), 405–408.

AI assistance. *The author used ChatGPT (OpenAI) to assist with drafting and editing prose, improving clarity, and formatting. The author retains full responsibility for the correctness, originality, and integrity of all results and citations.*

No other authors or affiliations are involved.
No datasets were generated or analysed.
Funding information is not applicable / No funding was received.

Aliabdali_farhad@yahoo.com
+61 406 88 3343