Alternating Binary Notation and Collatz Phase Expression: Canonical Representations and Exact Structural Dynamics

Y.Ueoka
Independent Researcher

Tsumugi (ChatGPT)
AI Co-researcher

28 November 2026 — Preprint, version 1.0.1

Abstract

Alternating binary notation (ABN) represents an integer by alternating signed powers of two. Its support defines the Collatz phase expression (CPE), a word of maximal occupied and vacant runs. The purpose of this representation is not merely to encode integers, but to make the structural degrees of freedom of the accelerated Collatz map explicit. Signed chains admit parity-dependent tripling laws, long vacant runs have a bounded penetration depth under one affine step before division, and the lowest occupied run directly determines the division exponent. The resulting run description naturally introduces structural quantities and distributions. We prove canonicality, give the exact update of every vacant run and the full output word, and derive threshold-count and finite-time balance identities for the vacancy count. The zero-vacancy and single-chain conditions characterize one-step arrival at 1; universal finite-time strict descent of either quantity outside this base set is equivalent to the Collatz conjecture. These results provide a deterministic framework for investigating structural descent independently of integer-value descent. The full-word update is equivalent to an earlier ordinary binary run rewrite; our focus is the signed-chain formulation and explicit threshold and finite-time vacancy balances. General finite-time descent is not established here.

1 Introduction and scope

The research programme studies the accelerated map

$$F(n) = \frac{3n+1}{2^{s(n)}}, \qquad s(n) = \nu_2(3n+1), \qquad n \in \{1, 3, 5, \ldots\},$$

where the domain here consists of strictly positive odd integers. The Collatz conjecture is equivalent to every such orbit reaching 1. The ordinary integer value is the state of the dynamical system, but is not the only possible quantity with which to organize its dynamics. We study an alternative description in which the elementary structural units are chains of alternating signed cells and intervening vacant runs. Our aim is to turn the evolution of these units into exact formulas and to identify quantities suitable for finite-time descent arguments.

Why ABN/CPE? The representation has four complementary roles. First, it makes occupied chains and vacant intervals explicit and recovers the signed integer without ambiguity. Second, the arithmetic of an entire signed chain under multiplication by three reduces to a parity-dependent expression containing at most three signed cells; the subsequent boundary normalization and the interaction across short gaps can be handled separately. The direct multiplication carry takes values in $\{0, 1, 2\}$, and a vacant interval can be penetrated by at most two cells from its lower boundary in one affine step before division. These are separate bounds: neither asserts a two-cell bound on ordinary binary carry propagation. Third, the chain and gap lengths supply natural observables: counts, length distributions, joint distributions of neighbouring runs, and threshold-dependent totals. Fourth, the exact run update connects these observables to structural generation, consumption, cancellation and shielding, and yields explicit finite-time balance identities. The point

is not that these observables are inaccessible in ordinary binary notation, but that they and their evolution are directly expressed in the chosen structural coordinates.

Scope and antecedents. The public ABN/CPE programme includes a historical preprint on directed rings and finite peeling [1]; the present paper neither adopts nor certifies that preprint's proof claim. We reconstruct the basic representation and exact run dynamics from the historical Phase 6 research record [2], principally P6-13, P6-54, P6-389, P6-452, P6-522 and P6-551–554. The formulas below are not claimed to be new relative to that record. Existing representations and related run-based Collatz methods are discussed in Section 3. Mathematical equivalence of encodings does not decide their usefulness for a particular analysis; nor does usefulness establish historical priority. The full-word update is equivalent to the earlier run rewrite of Sinyor; the explicit threshold and time-integrated balances are the principal structural results of this paper. Priority for the Collatz-specific formulas remains to be assessed.

Sections 2–4 give the canonical representations, their relation to existing encodings, and the structural characterization of the base set. Sections 5–6 establish local arithmetic and boundary rules. Section 7 derives the complete run update and the exact structural balances, and Section 8 explains how finite-time descent of structural quantities would imply convergence. Appendix A gives a separate digit-difference identity for prime valuations.

Let $b_j(n)$ be ordinary binary digits for $n \geq 0$, with $b_{-1} = 0$ and all sufficiently high digits zero. All cell positions start at zero. We use $\oplus$ for exclusive OR of bits, and XOR of integers for the corresponding bitwise operation. We distinguish signed ABN digits $A_j(n)$ from their support bits $e_j(n) = b_j(n) \oplus b_{j-1}(n)$ and from ordinary binary digits $b_j(n)$. We use $b_j(n) = 0$ for $j < 0$. The indicator $\mathbf{1}_P$ is one when proposition P holds and zero otherwise.

2 Canonical representations

Theorem 2.1 (Alternating Mersenne sum)**.** *With $M_k = 2^k - 1$, every positive integer has a unique expansion*

$$n = M_{k_1} - M_{k_2} + \cdots + (-1)^{\ell-1} M_{k_\ell}, \qquad k_1 > \cdots > k_\ell \geq 1.$$

Proof. Choose the unique k with $2^{k-1} \leq n \leq M_k$. The remainder $M_k - n$ lies between 0 and M_{k-1} (where $M_0 = 0$). Repeat for the positive remainder; the leading index strictly decreases. For uniqueness, induction on the number of terms shows that every nonempty alternating sum with leading index k lies in $[2^{k-1}, M_k]$: a nonempty tail has positive value at most M_{k-1} by the induction hypothesis. Thus the first index and the remainder are forced, and repetition proves uniqueness. □

Theorem 2.2 (Canonical ABN)**.** *Every positive integer has a unique finite expansion with digits in $\{-1, 0, 1\}$, indexed by nonnegative integers, with nonzero signs alternating from the highest positive sign to the lowest negative sign. Its digits are*

$$A_j(n) = b_{j-1}(n) - b_j(n), \qquad n = \sum_{j \geq 0} A_j(n) 2^j.$$

The nonzero mask is

$$E(n) = \sum_{j \geq 0} |A_j(n)| 2^j = n \operatorname{XOR} (2n).$$

Proof. The sum of the shifted binary digits equals $2n$; subtracting the original sum gives n. The nonzero digits are precisely binary transitions, hence alternate in sign and occur in pairs. Conversely, every alternating signed expansion of the stated kind can be paired into terms $2^u - 2^v$, $u > v$, whose ordinary binary digits are ones on $[v, u-1]$. Strictly ordered nonzero positions ensure that these intervals are disjoint and separated by at least one zero. Thus they are exactly the maximal one-runs of n, proving uniqueness. Taking absolute values of the adjacent differences gives the XOR formula. □

For example, $103 = (2^7 - 2^5) + (2^3 - 2^0)$. The two positive terms are not asserted to have independent Collatz dynamics. The alternating Mersenne expansion and the paired ABN expansion are distinct representations, though both use Mersenne numbers.

Definition 2.3 (CPE and structural quantities)**.** For a positive odd integer n, write the mask in *increasing* position order as

$$C(r_0)K(d_0)C(r_1)\cdots K(d_{m-1})C(r_m),$$

where $C(r)$ denotes r consecutive occupied cells and $K(d)$ denotes d consecutive vacant cells, each run maximal and of positive length. Here m counts K-runs and $\mu = m + 1$ counts C-runs. Define

$$B = \text{bitlength}(E(n)), \quad H_C = \sum_{i=0}^{m} r_i, \quad Z = \sum_{i=0}^{m-1} d_i = B - H_C.$$

The CPE of a positive odd integer starts and ends with an occupied cell. Conversely any finite binary support word with both endpoints occupied and a positive even number of occupied cells defines a unique positive odd integer by assigning alternating signs from the bottom, starting with -1. This follows from the paired representation above. In particular $B = \text{bitlength}(n) + 1$; B is not the ordinary binary length.

Structural observables. The run data naturally give the length-count functions, for positive integers r, d,

$$N_C(r; n) = \#\{i : 0 \le i \le m,\ r_i = r\}, \qquad N_K(d; n) = \#\{i : 0 \le i < m,\ d_i = d\}.$$

The quantities already defined are their zeroth and first moments (with counting, rather than probability, normalization):

$$\mu = \sum_{r\ge1} N_C(r; n), \qquad H_C = \sum_{r\ge1} r N_C(r; n),$$
$$m = \sum_{d\ge1} N_K(d; n), \qquad Z = \sum_{d\ge1} d N_K(d; n).$$

All sums have finite support. The K-run tail counts used in Section 7 are $N_D(n) = \sum_{d\ge D} N_K(d; n)$. One may also observe the joint counts of neighbouring triples (r_i, d_i, r_{i+1}), or longer subwords. Such joint counts retain local associations lost in separate length distributions. Neither the marginal distributions nor a fixed collection of joint counts is assumed to determine the next word: the exact update below uses the ordered run data and the lowest-boundary flag. The study of joint distributions and order sensitivity already belongs to the historical ABN/CPE programme: R178-A developed a pure-tripling chain-count balance using the empirical CKC distribution; R180-A showed that even the complete CKC triple counts need not determine the next pure-tripling C-length distribution; P6-338 gave a separate order-sensitive accelerated-map example [3, 2]. These results are mentioned for context, not added to the theorem scope of this paper. Statistical laws, correlations and asymptotic behaviour are research directions, not assumptions of the deterministic results in this paper.

3 Relations to signed binary, Booth recoding and Gray code

3.1 Relation to signed binary and Booth recoding

A signed binary expansion uses digits in $\{-1, 0, 1\}$ and weights 2^j. The digit set alone does not determine a unique representation: for instance, $1 = 2^0 = 2^1 - 2^0$. Here "balanced binary" refers to this symmetric digit set, not to a particular normalization. The non-adjacent form (NAF), associated with Reitwiesner [7], is a distinct canonical convention: adjacent nonzero digits are

forbidden, and the representation minimizes the number of nonzero digits among signed binary expansions. ABN instead requires alternation of the nonzero signs, with the highest sign positive and the lowest negative. Adjacent nonzero digits are allowed, and no minimum-weight property is asserted.

The distinction is substantial for the family $n_a = (4^a-1)/3$, $a \geq 1$. Its ordinary binary expansion is $\sum_{i=0}^{a-1} 2^{2i}$, already in NAF, of weight a. Its ABN is

$$n_a = \sum_{i=0}^{a-1} (2^{2i+1} - 2^{2i}),$$

with $2a$ occupied cells forming the single C-run $C(2a)$. For example, $21 = 16 + 4 + 1$ in NAF but $21 = 32 - 16 + 8 - 4 + 2 - 1$ in ABN. By Theorem 4.1, this entire occupied run encodes $s(n_a) = 2a$ and $F(n_a) = 1$. In NAF every maximal occupied run has length one, so the C-run length and the vacancy count defined here cannot be transferred unchanged to NAF. This does not assert that NAF cannot be used to study Collatz dynamics.

The signed digit rule $A_j = b_{j-1} - b_j$ is also precisely the radix-2 Booth recoding rule (with a zero below the least significant bit and sufficient zero extension above a positive integer), originating in signed binary multiplication [5, 6]. Indeed, the four possible ordered pairs $(b_j, b_{j-1}) = (0,0), (0,1), (1,0), (1,1)$ give $0, +1, -1, 0$, respectively. Thus the canonical ABN digit sequence on the present domain is not a distinct encoding from this Booth convention. The term ABN is retained to maintain continuity with the historical project. CPE is the run-length description of the support of that sequence, not an alternative choice of signed digits. Its use here is lossless for positive odd integers because the alternating signs are determined by the support. The contribution investigated in this paper is the Collatz-specific run dynamics and the resulting threshold and finite-time balance identities, not priority for signed binary digits or Booth recoding. Whether related Collatz-specific identities have appeared elsewhere remains a separate literature question.

3.2 Relation to reflected Gray code

The binary-reflected Gray encoding [8] is $g(n) = n \,\mathrm{XOR}\, \lfloor n/2 \rfloor$. For every nonnegative integer,

$$E(n) = g(2n) = 2g(n) + (n \bmod 2), \qquad g(n) = \lfloor E(n)/2 \rfloor.$$

Here the mask formula is extended by $E(0) = 0$. These identities follow by shifting the adjacent-XOR digits; the lowest bit of $E(n)$ is $b_0(n)$. Conversely, $b_j(n) = \bigoplus_{k>j} e_k(n)$ follows by telescoping the binary transitions with zero extension above. The signed Booth digits, Gray-related support and CPE run lengths thus determine the same integer on the stated domain, but make different structural features explicit. We claim no novelty for the classical Gray encoding or the adjacent-bit XOR rule. The Collatz dynamics below is a separate operation, not ordinary multiplication of Gray code words.

3.3 Relation to ordinary binary run-length dynamics

Binary run-length approaches to the accelerated Collatz map predate this work: Sinyor [10] gives a rewriting system for ordinary binary runs, building on Błażewicz and Pettorossi [9] (the latter cited here through Sinyor). Kaufman [11] also describes a binary-string algorithm for the accelerated map. These are directly relevant antecedents for comparing the structure and arithmetic of the respective representations.

The relation is exact at the representation level. If the ordinary binary runs of a positive odd integer have lengths $\ell_1, \ldots, \ell_q$ in low-to-high order, then q is odd and the occupied ABN positions are $0, \ell_1, \ell_1 + \ell_2, \ldots, \sum_{j=1}^{q} \ell_j$. Consequently the nonempty K-runs have lengths $\ell_j - 1$ for $\ell_j \geq 2$, in the same order, and $Z = \sum_j (\ell_j - 1)$. The C-runs group consecutive occupied positions separated by unit distances. Thus CPE is a lossless regrouping of the ordinary binary run data. The regrouping gives different elementary units: a sequence of unit-length ordinary runs becomes one C-run, and

an ordinary run of length greater than one gives a K-run between transitions. In particular, the signed-chain tripling law and the characterization $s = r_0$ below are expressed directly in these units. This is a mathematical reason to use CPE even when its relation to ordinary runs is known. The exact K-interval formulas and time-integrated threshold balances below are the particular results preserved here. The relationship to two antecedents can be made more precise.

Sinyor's rewriting system (2010, Section 2, Eq. (2.1)) has two states, f and g. A CPE word corresponds to $r_0 - 1$ unit ordinary runs, followed by alternating nonunit runs of length $d_i + 1$ and strings of $r_{i+1} - 1$ unit runs. Processing a nonunit run resets the state to g; processing a unit run toggles the state and increases the last output run length (or discards the increment when no output exists). At the initial nonunit run the state is g precisely when r_0 is even, including the removal of initial unit pairs. At each later nonunit run it is g precisely when r_i is odd. Thus g corresponds to $\epsilon_i = 1$ in Theorem 7.1. In ordinary run-length coordinates, the output distances contributed by the block V_i of Section 7 are

	$\epsilon_i = 0$	$\epsilon_i = 1$
$d_i \geq 2$	(d_i, r_{i+1})	$(1, d_i - 1, r_{i+1})$
$d_i = 1$	$(1, r_{i+1})$	$(r_{i+1} + 1)$

These are exactly the nonunit rewrite followed by $r_{i+1} - 1$ unit increments. A final distance of one is appended if the top state is g, i.e. if r_m is odd. The all-unit case is the base set. Consequently the full word rule here is a grouped, equivalent realization of that earlier rewrite, rather than a different one-step map. The grouping directly exposes the vacant intervals and their threshold contributions.

Kaufman's Algorithm 1 (2023, Section 2) likewise scans adjacent ordinary digits with a binary flag. Its flag, initialized to one for the affine step, is the addition carry in Eq. (1) below: unequal digits leave the flag unchanged, whereas equal digits reset it to the common digit. Its Method 1 counts initial alternating digits and the final flag to obtain the change in ordinary width. The width formula in Theorem 7.1 is the corresponding expression in CPE coordinates (with the base set treated separately). These relations identify the shared local arithmetic; the present treatment organizes it around signed chains, maximal vacant intervals, length distributions, and the explicit threshold and time-integrated balances. We do not infer historical priority for those balances from their absence in the two examined texts. Lagarias's annotated bibliography [12] describes the 1983 work as studying the ordinary binary patterns 1^k, 0^k and $(10)^k$. Hew [13] is also relevant to binary-width evolution; his full text has not been compared here. The 1983 original has not yet been obtained.

4 Structural characterization of the base set

Theorem 4.1 (One-step arrival at the base point)**.** *For positive odd n,*

$$\begin{aligned} F(n) = 1 &\iff Z(n) = 0 \iff \mu(n) = 1 \\ &\iff H_C(n) = B(n) \iff E(n) = 2^{B(n)} - 1 \\ &\iff s(n) = B(n) \iff n = (4^a - 1)/3 \ (a \geq 1). \end{aligned}$$

Moreover $s(n) = r_0$ for every positive odd n. In CPE notation the base words are exactly $C(2a)$, $a \geq 1$.

Proof. If $Z = 0$, the entire word is occupied and has even length $2a$. Pairing the signs gives $\sum_{i=0}^{a-1} 4^i = (4^a - 1)/3$. Conversely this sum has precisely that canonical word. The equation $F(n) = 1$ is equivalent to $3n + 1 = 2^s$; reduction modulo 3 forces s even. For the final assertion, the infinite binary sequence with bits $q_j = 1$ for even j and 0 for odd j represents $-1/3$ modulo every power of two: the finite geometric sum through any even position satisfies $3 \sum_{i=0}^{a-1} 4^i + 1 = 4^a$. For each $k \geq 1$, the lower k bits of n agree with this sequence exactly when 2^k divides $3n + 1$, because 3 is invertible modulo 2^k. The bottom run of occupied cells means the ordinary bits alternate starting with 1 until the first zero mask cell, at position r_0. At that position the ordinary bit equals the preceding bit and differs from q_{r_0}. This remains true if the run reaches the end of the mask, using

zero extension. Thus the first disagreement is at r_0, proving $s = r_0$. Since all K-runs have positive length, $Z = 0$ is equivalent to $m = 0$ and hence to $\mu = 1$. The identity $Z = B - H_C$ gives the next equivalence, and a word of width B has $H_C = B$ precisely when its mask is $2^B - 1$. Finally $r_0 = B$ means that the initial C-run is the entire word, so $s = B$ is equivalent to $Z = 0$. □

5 Parity-dependent C/K tripling laws

We now give a direct signed-cell form of the pure multiplication law, using the historical chain value (P6-54)

$$C(r) = \sum_{j=0}^{r-1} (-1)^{j+1} 2^j = \frac{(-2)^r - 1}{3}.$$

In this section $C(r)$ denotes the integer value of a chain when used in arithmetic expressions, and the run itself when used in a word. This value assumes a negative lowest cell; a chain with positive lowest cell has the opposite value. Throughout this subsection words are displayed in increasing position order, as in the definition of CPE. The symbols $p, m, 0$ mean $+1, -1, 0$, respectively, and 0^0 is the empty word.

Lemma 5.1 (Parity-dependent tripling of a signed chain and its upper gap)**.** *For a C-run of length r starting with m at position v, pure multiplication by three replaces its contribution by*

$$3\,2^v C(r) = \begin{cases} 2^{v+r} - 2^v, & r \text{ even}, \\ -2^{v+r+1} + 2^{v+r} - 2^v, & r \text{ odd}. \end{cases}$$

If a K-run of length d lies immediately above the chain, the resulting word on the original $r + d$ cells, provided the indicated replacement fits, is

$$\begin{aligned} r \text{ even},\ d \geq 1: &\quad m\,0^{r-1} p\,0^{d-1}, \\ r \text{ odd},\ d \geq 2: &\quad m\,0^{r-1} p m\,0^{d-2}. \end{aligned}$$

For a chain starting with p, reverse all signs. When r is odd and $d = 1$, the last emitted signed cell lies exactly at the first cell of the next input chain. It cancels the lowest emitted cell of that next chain. These are the only overlaps; apply all such cancellations before regrouping into maximal C- and K-runs. For the top chain use zero extension, with no upper cancellation. The resulting whole signed word is precisely the canonical ABN of $3n$.

Proof. The even formula follows from $3C(r) = 2^r - 1$, and the odd formula from $3C(r) = -2^r - 1 = -2^{r+1} + 2^r - 1$. Write the input as the sum of its signed chain contributions $\sigma_i 2^{v_i} C(r_i)$, where $\sigma_i \in \{-1, 1\}$ and $\sigma_{i+1} = (-1)^{r_i} \sigma_i$ by sign alternation. Each replacement has alternating signs internally, with lowest sign $-\sigma_i$ and highest sign $(-1)^{r_i}\sigma_i$. The next replacement begins with the opposite sign. Their support intervals are ordered, with an overlap possible only at $v_{i+1} = v_i + r_i + 1$ for an odd chain and $d_i = 1$. At that position the two opposite signs cancel. Every replacement contains at least two cells, so overlaps at the two ends of a replacement are distinct; the cancellation pairs are disjoint. Deleting adjacent opposite-sign pairs from the concatenated alternating list preserves alternation. The lowest and highest cells of the whole list survive, are negative and positive respectively, and all remaining positions are distinct. The sum is $3n$ by linearity and cancellation; canonical uniqueness gives the result. □

For even r the local unsigned replacement is $C(1)K(r-1)C(1)K(d-1)$. For odd $r \geq 3$ and $d \geq 2$ it is $C(1)K(r-1)C(2)K(d-2)$; for $r = 1$ it is $C(3)K(d-2)$. A zero-length K placeholder is deleted and neighbouring C-runs are merged. These formulas are local contributions before all boundary cancellations and mergers: an isolated signed odd chain need not itself be the canonical ABN of a positive integer. In particular one cannot infer the full output by adding the masks as integers. In the short-gap case $d = 1$ the cancellation rule, not a negative-length K-run, is used.

6 Boundary normalization, carry and shielding

Lemma 6.1 (Lowest-boundary update; P6-452)**.** *For odd $x > 0$, put $r = \nu_2(x+1)$ and $y = (x+1)/2^r$. Then*

$$A_0(y) = -1, \quad A_1(y) = 1 - b_{r+1}(x), \quad A_j(y) = A_{j+r}(x) \ (j \geq 2),$$

$$Z(y) - Z(x) = -r + 2b_{r+1}(x) + \mathbf{1}_{x+1=2^r}.$$

Proof. The low r bits of x are all one and the next is zero. Adding one changes just these $r+1$ bits. Division shifts the result; thus $b_0(y) = 1$ and $b_j(y) = b_{j+r}(x)$ for $j \geq 1$. Adjacent differences give the first formula. Below position $r+1$, the old mask has exactly two occupied positions, 0 and r, and position $r+1$ is occupied precisely if $b_{r+1}(x) = 1$. The new mask has $1 + (1 - b_{r+1}(x))$ occupied positions below position 2. Higher contributions match, giving $\Delta H_C = -2b_{r+1}(x)$. The change in B is $-r$ except when $x = 2^r - 1$, where it is $1 - r$. Subtract to obtain the result. □

Two-cell renormalization after shifting. Apply the preceding lemma to $x = 3n$ and $s = \nu_2(x+1)$. Define the shifted and truncated signed word $\widetilde{A}_j = A_{j+s}(x)$ for $j \geq 0$. Since $b_{s-1}(x) = 1$ and $b_s(x) = 0$, its lowest digit is $\widetilde{A}_0 = 1$, and $\widetilde{A}_1 = -b_{s+1}(x) \in \{0, -1\}$. Consequently

$$A_j(F(n)) = \widetilde{A}_j - 2\mathbf{1}_{j=0} + \mathbf{1}_{j=1} \qquad (j \geq 0).$$

Writing $p = +1$ and $m = -1$ and displaying digits high-to-low, the only boundary rewrites are $0p \to pm$ and $mp \to 0m$. The numerical value is preserved by the rewrite because $-2 \cdot 2^0 + 2^1 = 0$. Equivalently, for the support, only cell 1 is toggled: $E(F(n)) = (E(3n) >> s)$ XOR 2, where $>>$ is the ordinary right shift of a nonnegative integer. This also holds when $3n+1$ is a power of two, with zero extension. This statement concerns the normalization *after pure tripling and the shift*, not the full spatial extent of tripling itself. It separates the unbounded shift from a fixed two-cell signed correction (a one-cell support correction), independently of the length of the word or the division exponent.

For the pure tripling $n \mapsto 3n$, let $x_j = b_j(n)$, $y_j = b_j(3n)$ and c_j be the ordinary addition carry into position j, with $c_0 = x_{-1} = 0$. Set $e_j = x_j \oplus x_{j-1}$ and $e'_j = y_j \oplus y_{j-1}$, where $y_{-1} = 0$. Ordinary addition of $n + 2n$ gives

$$y_j = e_j \oplus c_j, \qquad c_{j+1} = \begin{cases} c_j, & e_j = 1, \\ x_j, & e_j = 0. \end{cases} \tag{1}$$

Indeed, when the input bits differ their sum is 1 and the outgoing carry equals the incoming carry; when they agree the carry equals either of them. The outgoing CPE is not obtained by multiplying the mask as an integer. For example $E(3) = 5$ but $3E(1) = 9$.

Bounded carries and one-step shielding of K-runs

Carry magnitude (P6-522 for pure tripling). For $\eta \in \{0, 1\}$, compute $3n + \eta$ directly by the recurrence

$$h_0 = \eta, \quad a_j = 3x_j + h_j, \quad z_j = a_j \bmod 2, \quad h_{j+1} = \lfloor a_j/2 \rfloor.$$

Induction gives $0 \leq h_j \leq 2$ at every position, since $0 \leq a_j \leq 5$ whenever $0 \leq h_j \leq 2$. The bound is sharp (take $n = 1, \eta = 1$ at position 1). The recurrence produces the ordinary binary digits of $3n+\eta$: summing $3x_j + h_j = z_j + 2h_{j+1}$ with weights 2^j telescopes, and the carry vanishes after finitely many zero input digits. This direct multiplication carry h_j is distinct from the addition carry c_j in (1). In fact, the same identity (1) computes $3n + \eta$ if $c_0 = \eta$ instead of zero. The addition carry is binary at every position, and $h_j = x_{j-1} + c_j$ follows by induction from the two recurrences. Thus these two bounds refer to different computational conventions, not to contradictory descriptions.

K-run shielding (P6-389). If $e_j(n) = 0$ on an interval $[L, U]$ with $1 \le L \le U$, then, for either $\eta = 0$ or $\eta = 1$,

$$e_j(3n + \eta) = 0 \qquad (L + 2 \le j \le U). \tag{2}$$

Indeed, the ordinary bits $x_{L-1}, \ldots, x_U$ have one common value q. Equation (1), with the initial carry appropriate to η, resets $c_{L+1} = q$ independently of c_L. It gives $z_j = q$ on $[L + 1, U]$, so the adjacent-XOR mask vanishes on $[L + 2, U]$. Empty intervals impose no conditions. Consequently an original K-run of length d retains at least $\max(d - 2, 0)$ vacant cells at the same positions before division. For the accelerated map the exact interval table below gives the corresponding surviving run after translation. The bound does not say that ordinary binary carry chains have length at most two: within a C-run the addition carry can propagate unchanged across an arbitrarily long interval. It is the penetration into an original vacant ABN interval that is bounded by two cells. The distinction is essential to using long K-runs as temporary shielding regions.

7 Complete run dynamics

The following theorem uses the ordered lengths in the CPE, with index zero at the lowest chain. K-runs are internal vacant intervals within the finite support, not the infinite exterior zero extension. All cell positions are counted from zero, and the position u_i immediately above a chain is not part of that chain. The coordinates U_i are measured after division by 2^s. The flag ϵ_i equals the parity of r_i except at the lowest chain, where it is reversed by the affine boundary correction. In the interval table an interval whose lower endpoint exceeds its upper endpoint is empty and is omitted.

Theorem 7.1 (All K-runs in one accelerated step; P6-551)**.** *Use the CPE convention above and suppose $m \ge 1$. Let u_i denote the position just above $C(r_i)$, including $u_m = B$. Set*

$$s = r_0, \quad U_i = u_i - s, \quad \epsilon_i = (r_i \bmod 2) \oplus \mathbf{1}_{i=0} \quad (i < m).$$

All maximal vacant runs of $F(n)$ are exactly the nonempty intervals in the following table (endpoints included), taken in increasing order of i and of position:

Condition	*Interval*	*Length*
$d_i \ge 2$	$[U_i + 1 + \epsilon_i, U_i + d_i - 1]$	$d_i - 1 - \epsilon_i$
	$[U_i + d_i + 1, U_{i+1} - 1]$	$r_{i+1} - 1$
$d_i = 1$	$[U_i + 2 - \epsilon_i, U_{i+1} - 1]$	$r_{i+1} - 1 + \epsilon_i$

The new width is $B(F(n)) = B - s + 1 + (r_m \bmod 2)$. For $m = 0$, $F(n) = 1$ and its mask has width 2 and no vacant runs.

Proof. At the first cell v of any C-run, $c_v = x_{v-1}$, because either $v = 0$ or the preceding cell is vacant. By (1), the carry stays constant throughout this run, and $y_j = 1 \oplus c_v$ on it. At its upper boundary u_i, the mask bit is zero and $y_{u_i} = c_v$, so $e'_{u_i} = 1$. The carry into position $u_i + 1$ is x_{u_i}, which equals $x_{v-1} \oplus (r_i \bmod 2)$ by alternating through the C-run. If $d_i \ge 2$, the first subsequent pure-tripling transition is $e'_{u_i+1} = r_i \bmod 2$. The carry is constant through the rest of the K-run and the next C-run, so another transition occurs exactly at $u_i + d_i$; there are no other transitions strictly between u_i and u_{i+1}. If $d_i = 1$, the change of carry and change of input mask occur together and the sole possible transition inside the interval is $e'_{u_i+1} = 1 \oplus (r_i \bmod 2)$. The same boundary argument with zero extension at B gives $e'_B = 1$, $e'_{B+1} = r_m \bmod 2$, and zero above.

The low s ordinary bits of $3n$ are one and bit s is zero. Adding one flips these $s + 1$ bits, so its adjacent-XOR mask flips exactly at positions 0 and $s + 1$. Right shifting by s gives the new odd integer; its position zero is occupied (the old mask bit at s was one). For positions $j \ge 1$, the mask is the shifted mask of $3n + 1$. Since $u_0 = s$ and $m \ge 1$, the change at $s + 1$ precisely reverses the lowest interval's parity flag, leaving the other intervals unchanged. The two possible patterns inside successive occupied boundary cells therefore give exactly the table. The uppermost

occupied pure-tripling cell is unaffected by the flip because $B \geq s+2$. After shifting, the top index is $B - s + (r_m \bmod 2)$, yielding the width. Every listed nonempty interval is bounded by occupied cells, and all positions outside them are occupied within the new width. Hence they are exactly the maximal K-runs. The case $m = 0$ is the base-set theorem. □

Thus the entire next support word follows from the old CPE, without reconstructing an ordinary integer: fill every cell within the new width except the listed intervals. The signed ABN follows by alternating signs. This is an exact finite-word description of F, not a probabilistic approximation.

7.1 Complete output CPE word (P6-554)

The interval theorem also has a direct word form. Let 1^a and 0^a mean repetitions of occupied and vacant cells, not powers representing integer values. For $m \geq 1$, use the coordinates and flags of Theorem 7.1; let V_i denote the output on $[U_i, U_{i+1} - 1]$. Then

$$V_i = \begin{cases} 1^{1+\epsilon_i} 0^{d_i - 1 - \epsilon_i} 1\, 0^{r_{i+1}-1}, & d_i \geq 2, \\ 1^{2-\epsilon_i} 0^{r_{i+1}-1+\epsilon_i}, & d_i = 1. \end{cases}$$

The entire low-to-high word is $V_0 \cdots V_{m-1} 1^{1+(r_m \bmod 2)}$; for $m = 0$ it is 11. Every exponent is nonnegative. To prove the formula, place zeros at the intervals of Theorem 7.1 and ones elsewhere; the word length is $U_{i+1} - U_i = d_i + r_{i+1}$. The top word follows from the width formula. The word V_i contains no zero if and only if $r_{i+1} = 1$ and $d_i = 1 + \epsilon_i$, as is seen by setting its zero exponents to zero. Regrouping the concatenation into maximal runs yields all C- and K-lengths, including merges across interval boundaries. This is the full accelerated word rule, rather than the pure-tripling contribution of a single signed chain in Lemma 5.1.

Worked word update. For $n = 27$, the low-to-high support is 101101, or

$$C(1)K(1)C(2)K(1)C(1).$$

Here $s = 1$, $(u_0, u_1, u_2) = (1, 4, 6)$, and $(\epsilon_0, \epsilon_1) = (0, 0)$. Both gaps have length one, so the second word rule gives $V_0 = 110$ and $V_1 = 11$; the top word is 11. Concatenation gives $1101111 = C(2)K(1)C(4)$, the support of 41. This example applies the structural rule without computing the integer $3n + 1$ or reconstructing its binary expansion; the integer equality $F(27) = 41$ is a separate check. Section 8 discusses the resulting structural descent.

7.2 Exact balances and finite-time descent criterion

Let $N_D(n) = \#\{i : d_i \geq D\}$ for an integer threshold $D \geq 1$, and put $b_n(D) = N_D(F(n)) - N_{D+2}(n)$. These quantities concern K-runs only.

Corollary 7.2 (Threshold balance; P6-552)**.** *For every positive odd n and every $D \geq 1$,*

$$\begin{aligned} b_n(D) = {} & \#\{i < m : r_{i+1} \geq D + 1\} \\ & + \#\{i < m : d_i = 1, \epsilon_i = 1, r_{i+1} = D\} \\ & + \#\{i < m : d_i = D + 1, \epsilon_i = 0\}. \end{aligned}$$

Proof. For $d_i \geq 2$, the first output length is $d_i - 1 - \epsilon_i$, so subtracting the input indicator $\mathbf{1}_{d_i \geq D+2}$ leaves precisely $\mathbf{1}_{d_i = D+1, \epsilon_i = 0}$. The other length contributes $\mathbf{1}_{r_{i+1} \geq D+1}$. For $d_i = 1$, only the output length $r_{i+1} - 1 + \epsilon_i$ contributes, producing the first and second terms. Sum over i. Empty sums cover the base set. □

Corollary 7.3 (Exact one-step and time-integrated balances; P6-553)**.**

$$Z(F(n)) - Z(n) = \sum_{i<m} \left[r_{i+1} - 2 + \epsilon_i (2\mathbf{1}_{d_i = 1} - 1) \right]. \tag{3}$$

Write $n_t = F^t(n_0)$ *and use* $r_{t,i}, d_{t,i}, m_t, \epsilon_{t,i}$ *for its data. For* $T \geq 1$ *define* $L_t = 2(T-1-t)$ *and* $(x)_+ = \max(x, 0)$. *Then*

$$\begin{aligned}
\mathcal{A}_T &= \sum_{i<m_0} \min(d_{0,i}, 2T), \\
\mathcal{R}_T &= \sum_{t=0}^{T-1} \sum_{i<m_t} \big[(r_{t,i+1} - 1 - L_t)_+ \\
&\qquad + \mathbf{1}_{d_{t,i}=1, \epsilon_{t,i}=1, r_{t,i+1}>L_t} + \mathbf{1}_{\epsilon_{t,i}=0, d_{t,i}\geq L_t+2}\big], \\
Z(n_T) - Z(n_0) &= \mathcal{R}_T - \mathcal{A}_T.
\end{aligned}$$

In particular strict Z*-descent at time* T *holds exactly when* $\mathcal{R}_T < \mathcal{A}_T$.

Proof. Adding output lengths in Theorem 7.1 and subtracting input lengths gives (3). By the definition of $b_n(D)$, iterating the threshold recurrence gives

$$N_D(n_T) = N_{D+2T}(n_0) + \sum_{t=0}^{T-1} b_{n_t}(D + L_t).$$

Sum over $D \geq 1$; all sums are finite for fixed T. The first term becomes $\sum_i (d_{0,i} - 2T)_+ = Z(n_0) - \mathcal{A}_T$. Summing the three indicator terms in Corollary 7.2 gives the displayed $\mathcal{R}_T$. Rearrangement proves the formula and criterion. □

For 599, the low-to-high word is $C(1)K(2)C(5)K(1)C(2)$, and $F(599) = 899$. The old K-lengths $(2, 1)$ become $(1, 4, 2)$, so Z increases from 3 to 7. In particular one-step Z-descent is not unconditional. The threshold contributions distinguish shortening of existing runs from generation associated with C-runs; the third term in the threshold balance is a correction to a two-cell shortening baseline, not an independent birth event. In particular, $\mathcal{R}_T$ is a nonnegative trajectory-dependent sum of generation and correction terms, not an external forcing or an independently specified birth count. The identity does not close the evolution in Z alone and does not establish universal descent.

8 Structural induction and the Collatz conjecture

Let $\mathcal{B} = \{n > 0 : n \text{ odd}, \ F(n) = 1\}$. In particular, $1 \in \mathcal{B}$ and $F(1) = 1$. Theorem 4.1 shows that $\mathcal{B}$ is the common minimum set of the nonnegative integer quantities Z and $\mu - 1$. The following elementary induction principle makes explicit why neither ordinary integer descent nor ordinary binary-length descent is required.

Theorem 8.1 (Equivalent structural descent problems)**.** *The following statements are equivalent, where all initial values are positive odd integers and all times are integers:*

1. *Every orbit reaches* 1 *in finite time.*
2. *Every orbit reaches* $Z = 0$ *in finite time (allowing time zero).*
3. *Every orbit reaches* $\mu = 1$ *in finite time (allowing time zero).*
4. *For every* n *with* $Z(n) > 0$, *some* $T \geq 1$ *satisfies* $Z(F^T(n)) < Z(n)$.
5. *For every* n *with* $\mu(n) > 1$, *some* $T \geq 1$ *satisfies* $\mu(F^T(n)) < \mu(n)$.
6. *For every* n *with* $Z(n) > 0$, *some* $T \geq 1$ *satisfies* $\mathcal{R}_T < \mathcal{A}_T$, *with the balances evaluated along the orbit starting at* n.

Proof. The base-set theorem and $F(1) = 1$ give the equivalence of (1), (2) and (3). Arrival at the common base set from outside it yields strict descent for both quantities, so (1) implies (4) and (5). For (4) implies (1), use strong induction on $Z(n) \in \mathbb{Z}_{\geq 0}$. At level zero the next iterate is 1. At any positive level, (4) supplies a finite orbit segment ending at a strictly lower level, where the induction hypothesis supplies a further finite segment reaching 1. Concatenation proves the claim. The same proof with $\mu(n) - 1 \in \mathbb{Z}_{\geq 0}$ gives (5) implies (1). The exact time-integrated balance gives the equivalence of (4) and (6). □

The induction does not require finitely many integers at each level, since only the nonnegative integer level must decrease. The individual descent statements (4) and (5) must each hold for all starting points outside the base set; a mixture of local descents of different quantities is not by itself a proof. Nor is descent of either quantity asserted at every single step. The outstanding requirement is to control the evolving C-run contributions along each fixed initial orbit, or establish another global descent mechanism. Finite-window statistics, finite verification, or individual sufficient conditions do not alone establish that control. No unconditional convergence theorem is claimed here. The separate study of sufficient descent conditions and the integration of different descent witnesses are deferred to later papers.

Structural descent as a distinct observable. For example, $n = 27$ has the low-to-high CPE $C(1)K(1)C(2)K(1)C(1)$ and $F(27) = 41$ has $C(2)K(1)C(4)$. Thus the integer increases, whereas Z decreases from 2 to 1 and μ decreases from 3 to 2. The mask width increases from 6 to 7. This example illustrates why the structural quantities are not simply proxies for numerical or length descent; it does not prove descent for other initial values.

The methodological role of CPE is to link the exact local rules with the observables used in induction. The chain tripling formula, the carry reset in vacant intervals, and the boundary flag in Theorem 7.1 lead to a deterministic description of the changing run lengths. The threshold corollary and the finite-time balance convert this description into quantities suitable for testing or proving descent. The same ordered word allows more detailed distributional or spatial questions, including whether shielding persists through successive steps and how neighbouring structures interact. Such questions can be studied independently of a full convergence proof; their answers need not be known to use the identities established here. No closed statistical evolution for a reduced set of observables is asserted.

A Gray differences and prime valuations

This appendix is an independent digit-identity supplement; none of the structural dynamics or descent equivalences in the main text uses it. For an integer base $p \geq 2$, write $b_j^{(p)}(x)$ for the ordinary base-p digits of a nonnegative integer x, extended by zeros above. Let $[z]_p$ denote the least nonnegative residue modulo p. Define the digitwise subtraction (without borrowing) and a Gray-type difference encoding by

$$x \ominus_p y = \sum_{j \geq 0} [b_j^{(p)}(x) - b_j^{(p)}(y)]_p p^j,$$
$$g_p(x) = \sum_{j \geq 0} [b_j^{(p)}(x) - b_{j+1}^{(p)}(x)]_p p^j.$$

The encoding is invertible on finite words by summing the difference digits from the top modulo p. For $p = 2$, $\ominus_p$ is XOR and $g_p = g$ from Section 3.2. For prime p, let $\nu_p(x)$ be the ordinary prime valuation of $x > 0$; for composite bases the same formulas hold with the largest exponent $k \geq 0$ for which p^k divides x.

Lemma A.1 (Prime valuation as a single Gray-type digit)**.** *For $x \geq 1$ and a prime p, with $k = \nu_p(x)$,*

$$x \ominus_p (x-1) = 1 + p + \cdots + p^k = \frac{p^{k+1}-1}{p-1},$$
$$g_p(x) \ominus_p g_p(x-1) = p^k.$$

The same formulas with $x = n+1$ give the increment identities for $n \geq 0$.

Proof. In the ordinary digits of x, the first k digits are zero and digit k is nonzero. Subtracting one changes those zero digits to $p-1$, decreases digit k by one, and leaves all higher digits unchanged. The digitwise modular difference is consequently one at positions $0, \ldots, k$ and zero elsewhere. The difference operation defining g_p acts linearly on the digit vectors over $\mathbb{Z}/p\mathbb{Z}$: it replaces this prefix of ones by a single one at position k. This proves both identities. □

In binary this gives $x \operatorname{XOR} (x-1) = 2^{\nu_2(x)+1} - 1$ and $g(x) \operatorname{XOR} g(x-1) = 2^{\nu_2(x)}$. In particular $g(3n+1) \operatorname{XOR} g(3n) = 2^{s(n)}$ for the accelerated Collatz map. The ordinary changed digit interval has length $s(n)+1$, whereas the Gray difference is supported only at position $s(n)$.

For a generalized affine step with integers a, b, n such that $an + b \geq 1$, define $s_p(n) = \nu_p(an+b)$ and $F_{a,b,p}(n) = (an+b)/p^{s_p(n)}$ (on any chosen positive-integer domain where the step is defined). Then

$$g_p(an+b) \ominus_p g_p(an+b-1) = p^{s_p(n)}.$$

This expresses the division exponent; it does not assert convergence or extend the binary run dynamics of the preceding sections automatically.

For positive integers $a + b = c$, let $D_p(x) = g_p(x) \ominus_p g_p(x-1)$ for $x > 0$. The same identity gives $D_p(a+b) = p^{\nu_p(c)}$. In particular, writing $\operatorname{rad}(m) = \prod_{p \mid m} p$, we have

$$\operatorname{rad}(abc) = \prod_{p:\, D_p(a) D_p(b) D_p(c) > 1} p.$$

The product is finite. For coprime a, b the three prime supports are pairwise disjoint. This provides a common digit-difference language for affine iteration and the prime support in the *abc* conjecture. It is only a reformulation: it supplies no inequality between c and $\operatorname{rad}(abc)$, and no claim of progress on that conjecture or of priority for the digit identity is made. The extension to prime division and its connection with generalized Collatz and *abc* were suggested by Y.Ueoka in BATON-R06; the definitions and elementary derivation here are by ChatGPT.

Contributions, provenance and reproducibility

Y.Ueoka initiated the ABN/CPE research programme and proposed its structural research direction, including boundary normalization, isolation and transport, and the use of structural observables and joint distributions. Tsumugi (ChatGPT) is the research pen name adopted in BATON-R14 for the AI co-researcher in this programme and the present reconstruction. Earlier records use the designation ChatGPT; their original attribution is retained, and the pen name is not a claim that all earlier ChatGPT sessions were the same model or individual. Tsumugi's research role includes mathematical formalization, proofs, finite verification, literature comparison and preparation of the English manuscript. The co-researcher designation describes this research role, separately from the human submitter's responsibility for publication.

The research underlying this paper was developed in a wider human–AI collaboration. We retain the historical records' specific assignments where they are documented; no unverified individual contribution is assigned to other collaborators. The Phase 6 record credits Y.Ueoka with the impetus for lowest-boundary normalization and the isolation/transport programme and ChatGPT with formalization and proofs of P6-452 and P6-551–553. In the BATON reconstruction, Y.Ueoka proposed emphasizing the carry bound and two-cell renormalization (R04), including Gray relations

and parity-dependent formulas (R05), and explicitly stating the base-set equivalences and separate inductions (R07). ChatGPT supplied the reconstructed proofs and explanatory details, including the signed-chain cancellation argument and the explicit word reconstruction using P6-554. In the earlier CKC research, R178-A credits Y.Ueoka with the research question; R180-A credits Y.Ueoka with the approach and ChatGPT with formalization, proof and finite verification. A source concordance appears in Appendix B; historical identifiers are retained to enable later source recovery, not to establish external publication priority.

Eight archived Python scripts verify the representation, boundary formulas, complete update, threshold counts, finite-time balances, and comparisons with earlier algorithms. They include checks of 65,536 positive odd integers below 2^{17}, 1,000 seeded large odd integers (32–512 bits), 5,200 trajectory-time pairs, and an independent enumeration of 16,384 admissible support words and 6,912 two-chain words (overlapping sets). All eight scripts were re-executed for version 1.0; the principal verification script is reproduced in Appendix B, together with a summary of the tests. The other seven scripts and full logs remain in the research archive. Finite tests supplement, rather than replace, the proofs.

A separate AI referee, given the preceding R15 manuscript PDF alone, reported no major mathematical defect and recommended minor revisions for a foundational preprint. R16 and this version address the report's expository and literature qualifications. This is not formal human peer review, nor an endorsement of the historical attribution records. No general finite-time descent or proof of the Collatz conjecture is claimed. A complete literature priority audit remains pending.

B Source concordance and reproducibility

The historical identifiers below are from the Phase 6 R270 archive unless indicated otherwise. They identify the original research records, not additional assumptions for the proofs. The historical archives are not attached to this PDF; the mathematical statements and proofs in this article are self-contained.

Present paper	Historical source
Sections 2–4	P6-13; BATON-R01–R07 reconstruction
Section 2, joint distributions	Phase 4 R178-A, R180-A; P6-338
Section 5, signed-chain value	P6-54; BATON reconstruction
Lemma 6.1	P6-452 (R69)
Section 6, carry and shielding	P6-522; P6-389
Theorem 7.1	P6-551 (R99)
Section 7.1	P6-554 (R99)
Corollaries 7.2–7.3	P6-552–553 (R99)
Section 8	BATON reconstruction of the descent criterion
Appendix A	BATON-R06

Finite verification summary. Eight Python scripts in the research archive were re-executed for public version 1.0, all with exit status zero. The principal script below tests the complete interval rule, width, base equivalence, threshold counts, signed boundary correction and finite-time vacancy balance. Additional archived tests cover the signed-chain rule, carry and shielding, Gray inverse and valuation, comparison with Sinyor and Kaufman, and independent support-word enumeration. The samples are not disjoint. No finite test is used as a proof of a universal statement.

Executable core check. Copy the following listing into a plain-text Python file and run with Python 3.10 or later, without the optimization flag `-O`. No third-party libraries are required. The listing reproduces the main run and balance tests described above. Long source lines are wrapped for display only.

```
"""Independent finite consistency checks for BATON-R02; Python 3.10+.
Does not prove the universal assertions. No external dependencies.
"""
import random
from itertools import groupby
from collections import Counter

def v2(x): return (x&-x).bit_length()-1
def bit(x,j): return (x>>j)&1 if j>=0 else 0
def E(n): return n^(n<<1)
def F(n):
    x=3*n+1
    return x>>v2(x)
def Z(n):
    e=E(n);return e.bit_length()-e.bit_count()
def data(n):
    e=E(n); runs=[(int(k),len(list(g))) for k,g in groupby(bin(e)[:1:-1])]
    assert runs[0][0]==runs[-1][0]==1
    r=[l for k,l in runs if k];d=[l for k,l in runs if not k]
    return r,d,e.bit_length()
def intervals(n):
    e=E(n);j=0;out=[]
    for k,g in groupby(bin(e)[:1:-1]):
        l=len(list(g))
        if k=='0':out.append((j,j+l-1))
        j+=l
    return out

def check(n):
    r,d,B=data(n);m=len(d);s=r[0];f=F(n)
    assert s==v2(3*n+1)
    assert (Z(n)==0)==(f==1)
    eps=[(a%2)^(i==0) for i,a in enumerate(r[:-1])]
    u=[r[0]]
    for a,b in zip(d,r[1:]):u.append(u[-1]+a+b)
    assert u[-1]==B
    U=[a-s for a in u]
    pred=[]
    for i,di in enumerate(d):
        p=eps[i]
        if di>=2:pred.extend([(U[i]+1+p,U[i]+di-1),(U[i]+di+1,U[i+1]-1)])
        else:pred.append((U[i]+2-p,U[i+1]-1))
    pred=[(a,b) for a,b in pred if a<=b]
    assert pred==intervals(f),(n,pred,intervals(f))
    assert E(f).bit_length()==(B-s+1+r[-1]%2 if m else 2)
    assert Z(f)-Z(n)==sum(a-2+p*(2*(b==1)-1) for a,b,p in zip(r[1:],d,eps))
    _,df,_=data(f); old=Counter(d); new=Counter(df)
    # Count exact threshold differences independently of predicted intervals.
    for D in range(1,max(d+df+[0])+2):
        lhs=sum(v for k,v in new.items() if k>=D)-sum(v for k,v in old.items() if k>=D+2)
        rhs=sum((a>=D+1)+(b==1 and p==1 and a==D)+(b==D+1 and p==0) for a,b,p in zip(r[1:],d,eps
    ))
        assert lhs==rhs,(n,D,lhs,rhs)
    return r,d,eps

def check_boundary(x):
    r=v2(x+1);y=(x+1)>>r
    A=lambda n,j:bit(n,j-1)-bit(n,j)
    assert A(y,0)==-1 and A(y,1)==1-bit(x,r+1)
    for j in range(2,y.bit_length()+2):assert A(y,j)==A(x,j+r)
```

```
    assert Z(y)-Z(x)==-r+2*bit(x,r+1)+(x+1==1<<r)

def trajectory(n,T):
    initial=n;_,ds,_=data(n);A=sum(min(d,2*T) for d in ds);R=0
    for t in range(T):
        r,d,_=data(n); L=2*(T-1-t)
        for i,di in enumerate(d):
            p=(r[i]%2)^(i==0);a=r[i+1]
            R+=max(a-1-L,0)+(di==1 and p==1 and a>L)+(p==0 and di>=L+2)
        n=F(n)
    assert Z(n)-Z(initial)==R-A,(initial,T)

if __name__=='__main__':
    rng=random.Random(20261128)
    for n in range(1,2**17,2):check(n);check_boundary(n)
    print('PASS: all 65,536 positive odd integers below 2^17, complete K-intervals, width, s,
    one-step and all relevant thresholds, boundary.')
    for k in [32,64,128,256,512]:
        for _ in range(200):
            n=rng.getrandbits(k)|(1<<(k-1))|1;check(n);check_boundary(n)
    print('PASS: 1,000 seeded large odd integers, 200 each at 32/64/128/256/512 bits.')
    for n in range(1,2000,2):
        for T in [1,2,5,10,20]:trajectory(n,T)
    for _ in range(200):trajectory(rng.getrandbits(128)|1,30)
    print('PASS: 5,200 trajectory-time balance cases (5,000 small; 200 seeded 128-bit-or-less).'
    )
```

References

[1] Ueoka, Y., et al. *A Finite-Peeling Proof of the Collatz Conjecture via the Directed ABN Baumkuchen Representation*. Historical preprint, version v34, 28 June 2026. https://doi.org/10.5281/zenodo.20968007. Metadata and abstract were reported checked in BATON-R07; the record could not be retrieved in R08. Proof claim not adopted.

[2] ABN/CPE research collaboration. Phase 6, R270 anti-stagnation policy archive, 26 November 2026. Internal source: MAIN_NOTE.tex, P6-13, P6-54, P6-389, P6-452, P6-522, P6-551–554. Historical archive, not publicly reproduced here; selected source identifiers are listed in Appendix B.

[3] ABN/CPE research collaboration. Phase 4, R178-A and R180-A (November 2026), preserved in *Collatz_MAIN_R201C4_A112.pdf*, physical pp. 26 and 24 (summary pages). Historical research record; not a claim of external publication priority.

[4] ABN/CPE research collaboration. BATON-R01, 28 November 2026. Internal reconstruction of the canonical representation and publication plan. Historical archive; source identifiers are retained in the text and Appendix B. Its historical reference to Zenodo record 15870107 could not be re-fetched in R02.

[5] A. D. Booth, A signed binary multiplication technique, *Quarterly Journal of Mechanics and Applied Mathematics* 4(2) (1951), 236–240. https://doi.org/10.1093/qjmam/4.2.236.

[6] MIT, 6.175 Constructive Computer Architecture, Lab 3: Multipliers, section on Booth's multiplication algorithm. https://csg.csail.mit.edu/6.175/labs/lab3-multipliers.html. Recoding table consulted 28 November 2026.

[7] G. W. Reitwiesner, Binary arithmetic, *Advances in Computers* 1 (1960), 231–308. https://doi.org/10.1016/S0065-2458(08)60610-5.

[8] F. Gray, *Pulse Code Communication*, US Patent 2,632,058, granted 17 March 1953. https://patents.google.com/patent/US2632058A/en.

[9] J. Błażewicz and A. Pettorossi, Some properties of binary sequences useful for proving Collatz's conjecture, *Foundations of Control Engineering* 8(2) (1983), 53–63. Bibliographic data via Sinyor; original not yet examined.

[10] J. Sinyor, The $3x+1$ problem as a string rewriting system, *International Journal of Mathematics and Mathematical Sciences* (2010), Article 458563, 6 pp. https://doi.org/10.1155/2010/458563.

[11] R. Kaufman, A reduced forward Collatz algorithm: how binary strings change their length under $3x+1$, arXiv:2301.07466 (2023). https://arxiv.org/abs/2301.07466. Algorithm 1 and Method 1 compared in BATON-R15.

[12] J. C. Lagarias, The $3x+1$ problem: An annotated bibliography (1963–1999), arXiv:math/0309224. Author-hosted version, p. 10, item 27: https://websites.umich.edu/~lagarias/doc/3x%2B1bib.pdf.

[13] P. C. Hew, Collatz on the dyadic rationals in $[0.5, 1)$ with fractals: How bit strings change their length under $3x+1$, *Experimental Mathematics*, published online 2019. https://doi.org/10.1080/10586458.2019.1577765. Bibliographic reference only.