MATHDISCOVERIES
Mathematical Preprint

Working Draft

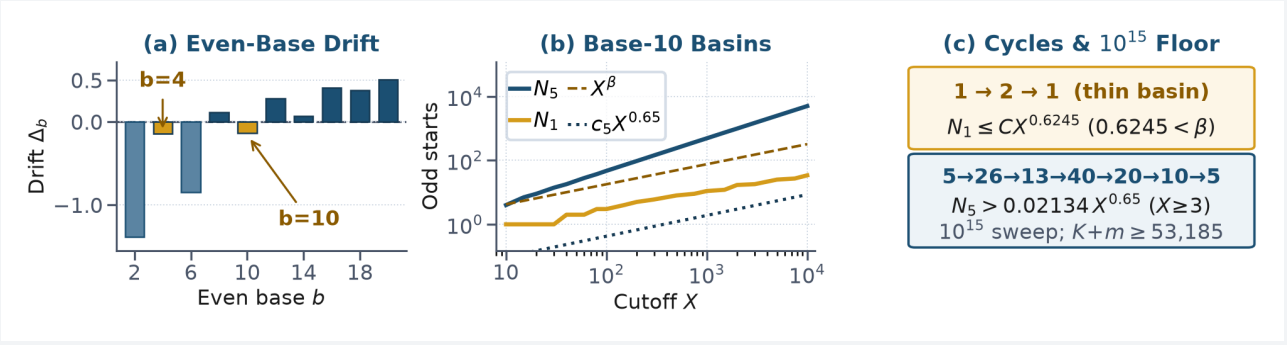
The Ouroboros Conjecture: Competing Attractors in Base 10 Among Last-Digit Collatz Maps

Tyler B. Hughes†

†Independent Researcher, United States

Supporting Information: `checker.zip`, `recompute.py`, *and* `lean/Ouroboros/` (*Lean 4 + Mathlib, rev.* `797e3413d`)

Disclosure: AI-assisted working draft directed by Tyler B. Hughes. Not for submission as is. No human mathematician has yet refereed the proof.



ABSTRACT: For $b \ge 2$, set $f_b(n) = n/2$ (even) and $(n \bmod b)\,n + 1$ (odd). Via $n = 2y + 1$, base 4 recovers shortcut Collatz, equating $\{1,3\}$-attraction to Collatz. Base 10 (f_{10}) splits into infinite 1- and 5-basins. We prove exponential density-one descent ($p_k(10) \le 1.124953\,(0.996423)^k$) and basin separation: $N_1(X) \le (244/100)\,X^\beta$ and $N_5(X) \ge (605/10^6)\,X^{13/20}$ $(X \ge 3)$ give $N_1(X)/N_5(X) \to 0$ (in Lean 4: $0.0145\,X^{0.49875} \le N_1(X) \le C\,X^{0.6245}$, $N_5(X) > 0.02134\,X^{0.65}$ for $X \ge 3$, $O(X^{-0.0255})$). In $\mathbb{Z}_2$, the odd 1-basin closes to a Cantor set of dimension $\beta \simeq 0.62638214 > 0.6245$. Every $a \bmod m$ hits $\mathcal{B}_5$ infinitely often; a halving test governs $\mathcal{B}_1$, whose $b = 2^t u$ endings have dimension $(\log u + t \log \lambda)/\log b$ (0.62638214, 0.88752982, 1 at $b = 2, 10, 7$). We exclude further cycles of ≤ 8 odd steps, verify $n \le 10^{15}$ ($K \ge 34{,}905$, $m \ge 15{,}496$, $K + m \ge 53{,}185$), classify even-base nonpositive drift and zero non-descent as $b \in \{2,4,6,10\}$, prove the rootless odd trapdoor law, and resolve 905/999 odd bases through 1,999; the Ouroboros conjecture and $N_1(X)$ exponent remain open.

Verification Scope (G11): WARN: Human referee: pending
Main results: 15 of 16 in Lean Kernel (94%) · 1 Cert Only · 1 open (not counted)
Abstract claims: 14 of 15 in Lean Kernel (93%) · 1 Cert Only · 2 open (not counted)

Full Trust Boundary: Section 8.

MSC 2020: 11B83, 11Y16, 11Y55, 37P99, 60J10 **Keywords:** Generalized Collatz maps, 2-adic Cantor sets, stopping times, Markov chains, Lean 4

1. ONE RULE, MANY BASES

For any integer base $b \ge 2$, let $d_b(n) = n \bmod b \in \{0, 1, \ldots, b-1\}$ denote the final base-b digit of a positive integer n. Consider the iteration

$$f_b(n) = \begin{cases} n/2, & n \text{ even}, \\ d_b(n)\,n + 1, & n \text{ odd}. \end{cases} \tag{1}$$

An even integer halves as in the classical $3x + 1$ problem; an odd integer reads its own last digit as its multiplier. Unlike fixed-multiplier generalizations of Collatz, the rule carries no external parameters beyond the base b.

Hold sixty starting integers fixed in $[10^9, 10^{10})$ and watch their *orbit values* $f_b^k(n)$ unfold as b steps from 2 through 20 (Figure 1). Three sharply distinct regimes appear at a glance. In bases 2 and 6, every curve collapses almost immediately. In bases 4 and 10, the curves wander irregularly for hundreds of steps before landing in a cycle (median entry times 279 and 353.5). In every other even base—8, 12, 14, 16, 18, 20—trajectories climb away, with base 14 hovering so close to the threshold that 6 of the 60 starts still drop into a cycle while all 54 crossings above 10^{60} happen after step 1,500. Odd bases behave altogether differently: orbits either drop straight to 1 on a green triangle (an odd multiple of b, whose last digit is 0, sending $0 \cdot n + 1 = 1$ in a single step) or shoot nearly vertically past 10^{60}.

Now fix the starting range $1 \le n \le 20{,}000$ and look inward toward the cycles themselves (Figure 2). Across bases 2 through 11, two panels wear bright solid rings: *only bases 4 and 10 send every tested start into exactly two loops with zero escapes.* Bases 2, 5, and 6 send all 20,000 starts into the single loop $1 \to 2 \to 1$; bases 3, 7, 8, 9, 11 lose large fractions past 2^{256}, with bases 7 and 8 already hosting three and four cycles. Yet the two brightly ringed panels—labeled *base* 4 (*Collatz*) and *base* 10 (*Ouroboros*)—pose an immediate visual puzzle. Compare the amber loop of 1 at the heart of each panel:

- In base 4, the loop of 1 attracts just 15 of the 20,000 starts along a single unbranched amber spike, while the re-

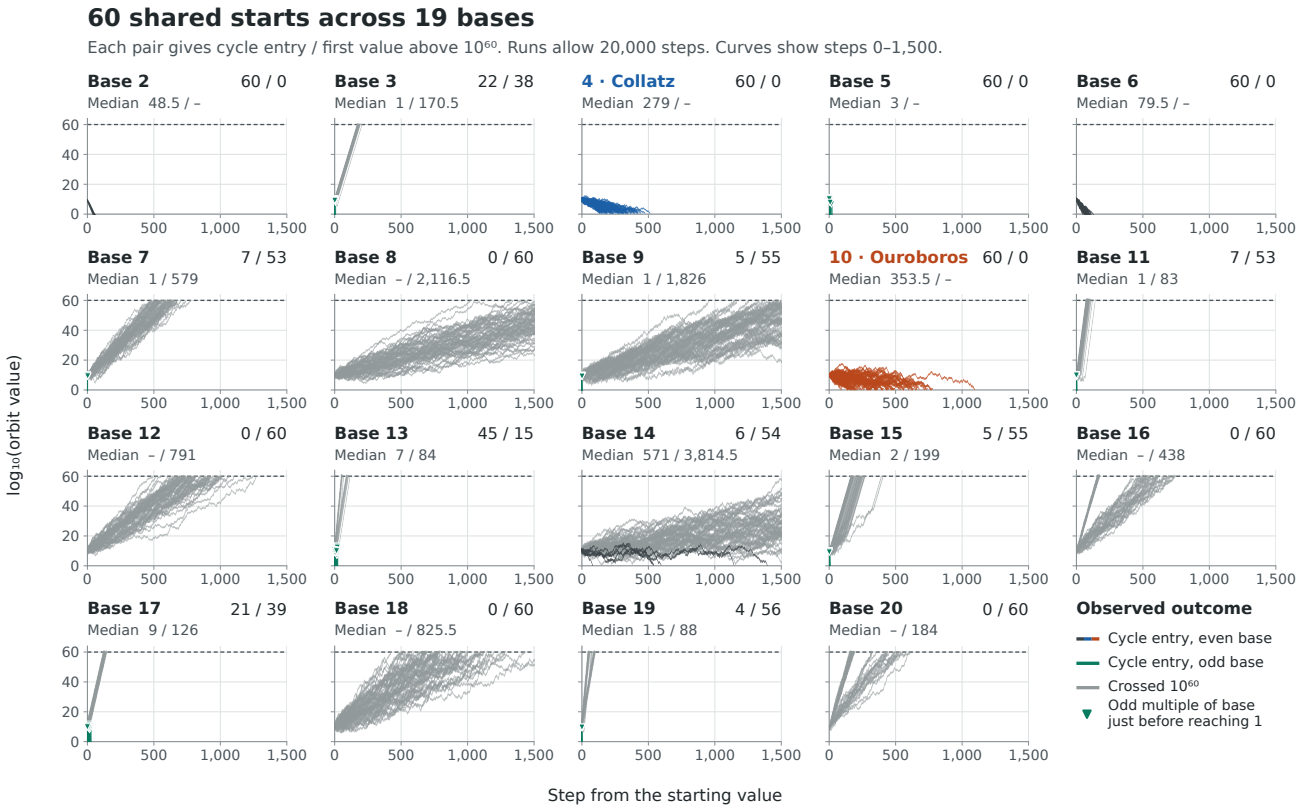


Figure 1. Sixty shared starts in $[10^9, 10^{10})$ iterated across bases $b = 2$ through 20. At step k, the *orbit value* is the k-th iterate $f_b^k(n)$, where $f_b(n) = n/2$ for even n and $f_b(n) = (n \bmod b)\, n + 1$ for odd n. Horizontal distance counts ordinary steps k; vertical height plots $\log_{10} f_b^k(n)$. Curves are drawn through step 1,500, while panel summaries report the full run until cycle entry or first crossing above 10^{60} (limit 20,000 steps). In base 14, six starts enter a cycle within the window and all 54 crossings above 10^{60} occur beyond step 1,500. A threshold crossing records a finite excursion, not a proof of divergence.

maining 19,985 starts spiral into the blue loop of 3.

- In base 10, the loop of 1 attracts 167 starts arranged in a *branching* amber tree, while 19,833 starts enter the blue loop of 5.

Why base ten is the last contracting even base. Figures 1 and 2 raise an immediate question: why does the even-base behavior flip upward at base 8, reverse back downward one final time at base 10, and then climb from base 12 onward? Part of the answer can be seen by hand (Figure 3). Whenever $4 \mid b$, an odd integer n and its last base-b digit $d_b(n)$ agree modulo 4, so $d_b(n)\, n + 1 \equiv 2 \pmod 4$ is never divisible by 4: every odd step gets trapped at a single halving (such as $19 \to 58 \to 29$ in base 8), capping its halving budget at $-\log 2$. Whenever $b \equiv 2 \pmod 4$, by contrast, an odd step can purchase cascades of halvings—in base 10, $17 \to 120 \to 60 \to 30 \to 15$ drops below its start even after multiplying by 7—doubling the average halving budget to $-2\log 2$. Section 6 proves that this extra halving budget pulls base 10 back below zero one last time: drift returns positive at base 12 (where $4 \mid b$ caps halvings again), growing digits overtake even the doubled $b \equiv 2 \pmod 4$ halving budget at base 14, and drift remains positive from base 12 onward.

Base four as Collatz and the Ouroboros Conjecture. That exact same modulo-4 congruence solves the visual puzzle in Figure 2. Calling base 4 "Collatz" looks paradoxical at first sight: the classical Collatz map $C(y) = y/2$ (even y), $C(y) = 3y + 1$ (odd y), with shortcut $T_C(y) = y/2$ (even y), $T_C(y) = (3y+1)/2$ (odd y), is famous for having *one* known positive cycle, $1 \to 4 \to 2 \to 1$, whereas Figure 2 displays *two* cycles in base 4, namely $1 \to 2 \to 1$ and $3 \to 10 \to 5 \to 6 \to 3$. Figure 4 dissolves this paradox completely.

Proposition 1 (Base four is equivalent to the Collatz conjecture). *Every positive integer $n \geq 1$ reaches 1 or 3 under f_4 if and only if every positive integer $y \geq 1$ reaches 1 under the classical Collatz shortcut map T_C:*

$$(\forall n \geq 1,\ \exists k \geq 0,\ f_4^k(n) \in \{1,3\}) \iff (\forall y \geq 1,\ \exists k \geq 0,\ T_C^k(y) = 1).$$

Proof. Write every odd positive integer as $n = 2y + 1$ with $y \geq 0$. Direct substitution verifies both branches: for $y = 2k$ ($n = 4k + 1$, base-4 digit 1), $f_4(n) = 4k + 2 = 2(2k + 1)$ and $f_4^2(n) = 2(y/2) + 1 = 2T_C(y) + 1$; for $y = 2k + 1$ ($n = 4k + 3$, base-4 digit 3), $f_4(n) = 12k + 10 = 2(6k + 5)$ and $f_4^2(n) = 2((3y + 1)/2) + 1 = 2T_C(y) + 1$. In both cases the intermediate even integer is congruent to 2 (mod 4), so f_4 per-

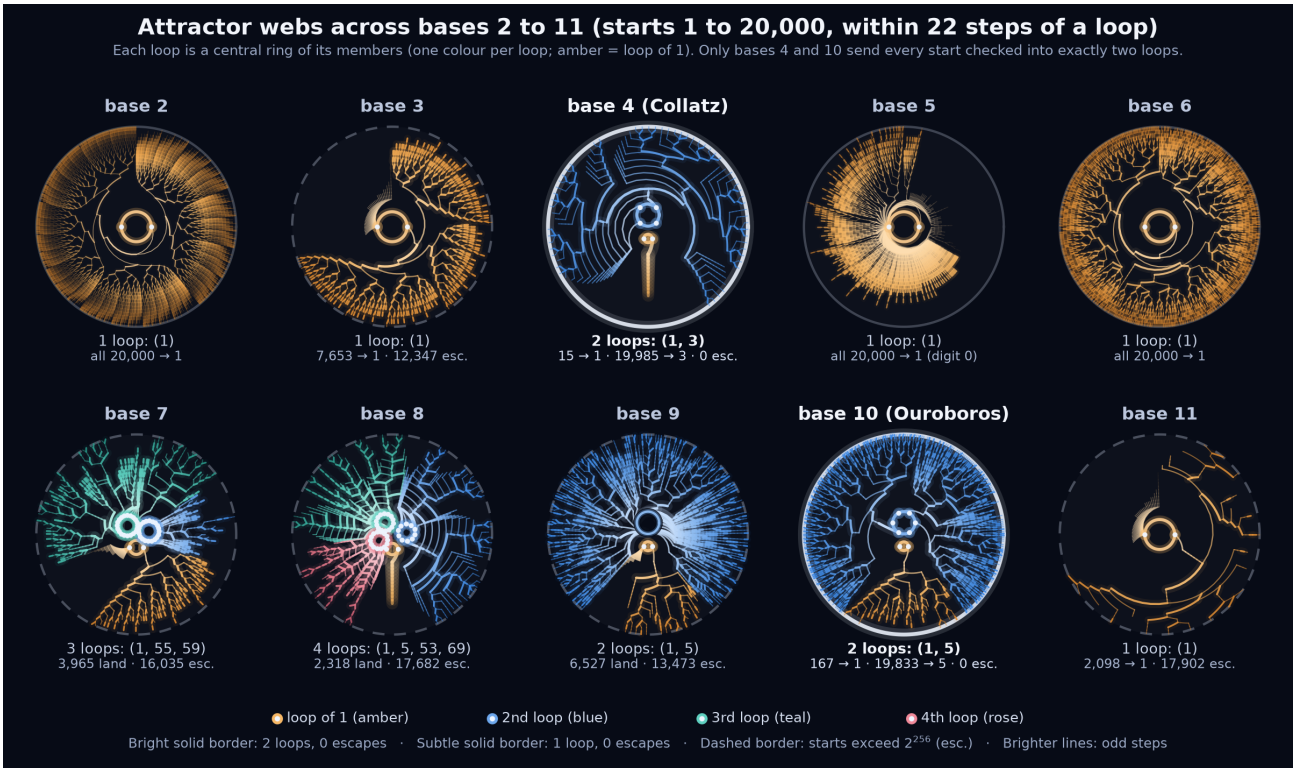


Figure 2. Attractor webs across bases 2 through 11, starting from all $1 \le n \le 20{,}000$ and tracing up to 22 ordinary steps near each observed cycle. Central rings depict the cycles (amber marks the loop of 1; blue, teal, and rose mark further cycles), and concentric outer rings count backward steps. Only bases 4 and 10 carry bright solid borders (two loops, zero escapes above 2^{256}). Odd-base panels sample shallow spokes so deeper branches stay visible. Neither experiment settles starts beyond its finite range.

forms exactly one halving and induction gives $f_4^{2k}(2y+1) = 2T_C^k(y)+1$ for all $k \ge 0$ (`prop_base4_step`, `prop_base4_iter`). Because T_C maps $\mathbb{Z}_{\ge 1}$ into itself, no odd $n = 2y+1 \ge 3$ ever reaches 1 (`prop_base4_odd_gt1_ne_one`), so the basin of $1 \leftrightarrow 2$ consists precisely of $\{2^a : a \ge 0\}$. Every even integer $2^a(2y+1)$ halves a times onto its odd part $2y+1$, which reaches 3 under f_4 if and only if y reaches 1 under T_C (`prop_base4_reaches_three_iff`). □

The same change of variables $n = 2y+1$ appears in Yolcu, Aaronson, and Heule's analysis of Farkas's map [8, 11] [to check]. Here it exposes the exact wall separating the two basins (Figure 4). On $\mathbb{Z}_{\ge 0}$, ordinary Collatz *already* has two loops: the trivial fixed point $0 \to 0$ (omitted on positive integers) and the 3-step cycle $1 \to 4 \to 2 \to 1$. Shifting by $n = 2y+1$ brings $y = 0$ into $\mathbb{N}$ as $n = 1$, turning $0 \to 0$ into $1 \to 2 \to 1$ and turning the 15 powers of two up to 20,000 $(1, 2, 4, \dots, 16{,}384)$ into the unbranched amber spike of Figure 2. Meanwhile, on odd y, Base 4 and ordinary Collatz march in lockstep ($3x+1$, then $/2$); on even y, Collatz divides by 2 in a single step while Base 4 pauses for one $+1$ move because $2y+1$ is odd. That extra pause explains why Collatz's 3-step loop $4 \to 2 \to 1 \to 4$ becomes Base 4's 4-step loop $5 \to 6 \to 3 \to 10 \to 5$.

In Base 10—our *Ouroboros map* $f = f_{10}$ on $\mathbb{Z}_{\ge 1}$, found during a screen of 16 digit-driven Collatz rules—that wall collapses. Because 31 ends in 1, its odd step gives $1 \cdot 31 + 1 = 32 = 2^5$, vaulting straight onto the powers-of-two highway to 1. Meanwhile 7 follows $7, 50, 25, 126, 63, 190, \dots$, climbs to a peak of 2,576, and settles after 40 steps into a second loop:

$$\begin{aligned} &1 \longrightarrow 2 \longrightarrow 1, \\ &5 \longrightarrow 26 \longrightarrow 13 \longrightarrow 40 \longrightarrow 20 \longrightarrow 10 \longrightarrow 5. \end{aligned} \tag{2}$$

Define the full (even and odd) attracting basins of these two loops by

$$\mathcal{B}_i := \{n \in \mathbb{Z}_{\ge 1} : f_{10}^k(n) = i \text{ for some } k \ge 0\} \qquad (i \in \{1, 5\}).$$

Because the two cycles (2) are disjoint and closed under f_{10}, $\mathcal{B}_1 \cap \mathcal{B}_5 = \emptyset$. The central open question of the base-10 dynamics is whether these two basins exhaust all positive integers.

> **Conjecture 2** (The Ouroboros Conjecture). *For every integer $n \ge 1$, there exists an integer $k \ge 0$ such that $f_{10}^k(n) \in \{1, 5\}$. Equivalently, $\mathbb{Z}_{\ge 1} = \mathcal{B}_1 \sqcup \mathcal{B}_5$, meaning that $1 \leftrightarrow 2$ and $5 \to 26 \to 13 \to 40 \to 20 \to 10 \to 5$ are the sole positive periodic cycles of f_{10} and every positive orbit is bounded.*

In our Lean 4 library (`lean/Ouroboros/Defs.lean`), this statement is defined verbatim as `OuroborosConjecture` using `f`, `Reaches`, `InBasin1`, and `InBasin5`. Conjecture 2 remains unsettled and is tracked explicitly as open outside the proved-claim denominators in Section 8.

Santi [12] independently recorded the six-cycle (2) as "cycle A" in a different Collatz-type map that alternates multipliers 3 and 5 on odd steps without reading a digit: his multiplier-5 step at 5 and multiplier-3 step at 13 match the two odd moves of (2) (verified in Lean 4 by `santi_cycleA_`

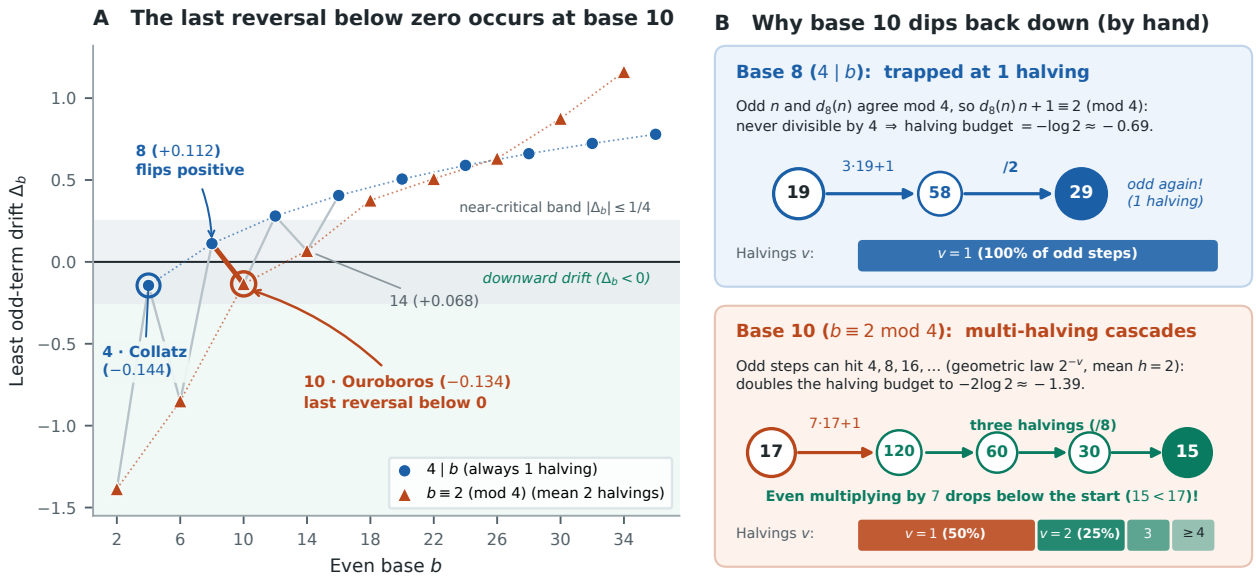


Figure 3. Why base 10 is the last even base with downward drift. **A:** Least odd-term drift Δ_b across even bases $b \le 36$. Drift is negative at $2, 4, 6$, turns positive at 8 $(+0.112)$, reverses back below zero one last time at 10 (-0.134), and remains positive from 12 onward. **B:** The hand-checkable mechanism: in base 8 $(4 \mid b)$, every odd step satisfies $d_8(n)n + 1 \equiv 2 \pmod 4$ and gets stuck at 1 halving $(19 \to 58 \to 29)$, whereas in base 10 $(b \equiv 2 \pmod 4)$ odd steps can buy cascades of halvings $(17 \to 120 \to 60 \to 30 \to 15 < 17)$, doubling the mean halving budget from $-\log 2$ to $-2\log 2$.

shared), whereas his other two listed cycles send 7 to 22 or 36 instead of $f(7) = 50$. The following eight lines verify our four opening decimal starts (counting ordinary steps to the first positive visit to 1 or 5):

Opening Decimal Starts (8 lines, verified by `recompute.py`)

```
def step(n):
    return n // 2 if n % 2 == 0 else (n % 10) * n + 1
for start in (1, 5, 7, 31):
    n, path = start, [start]
    while len(path) == 1 or n not in (1, 5):
        n = step(n)
        path.append(n)
    print(start, n, len(path) - 1, max(path))
```

Its four output lines end in $(1, 2, 2)$, $(5, 6, 40)$, $(5, 40, 2576)$, and $(1, 6, 32)$, respectively.

Main results and roadmap. Because Figure 2 singles out base 10 as the sole twin of Collatz base 4 among small bases, we follow the zoom straight into base 10 first (Sections 2–5), pull back across all bases $b \ge 2$ in Section 6 to prove rigorously that no larger base produces a third twin, and close the mathematical narrative in Section 7 by holding the decimal rule fixed while varying the base in which starting integers are written. Section 8 presents the machine-generated Reader's Trust Boundary and per-claim verification ledger; Section 9 collects open problems, reproducibility receipts, and prior work.

Write $N_i(X)$ for the count of *odd* integers $n \in [1, X]$ whose f_{10}-orbit visits $i \in \{1, 5\}$, and $N_i^{\text{all}}(X)$ for the count of *all* positive integers $n \in [1, X]$ visiting $i \in \{1, 5\}$. Write $d(S) = \lim_{X\to\infty} |S \cap [1, X]|/X$ for natural density (with $\underline{d}, \overline{d}$ for lower and upper densities). For any even base b, fold the guaranteed first halving after an odd step into the *shortcut map* $T_b(n) = n/2$ (even n), $T_b(n) = (d_b(n)\, n + 1)/2$ (odd n), and put

$$D_k(b) = \{n \ge 1 : T_b^j(n) \ge n \text{ for } 1 \le j \le k\},$$
$$p_k(b) = d(D_k(b)), \qquad p_\infty(b) = \lim_{k\to\infty} p_k(b),$$

noting that including or excluding the fixed point $n = 1$ (in the loop $1 \leftrightarrow 2$) alters any finite-interval fraction by at most $1/N$ and leaves natural density $p_k(b)$ unchanged. To align every stated claim 1-to-1 with its verification tier in Section 8, we separate our pure-kernel Lean 4 theorems (15 of 16 main results; 28 of 29 proved claims overall) from the single external-certificate theorem (Theorem 9) and our two open conjectures.

Theorem 3 (Exponential non-descent bound in base ten). *For the base-10 shortcut map $T = T_{10}$, every integer $k \ge 1$, and every real $\varepsilon > 0$, there exists $N_0 \in \mathbb{N}$ such that for all integers $N \ge N_0$,*

$$\frac{|D_k(10) \cap [2, N]|}{N} \le \frac{1{,}124{,}953}{1{,}000{,}000}\left(\frac{996{,}423}{1{,}000{,}000}\right)^k + \varepsilon.$$

In particular, $\overline{d}(D_k(10)) \le 1.124953\,(0.996423)^k$ for every $k \ge 1$ (and by Theorem 11, the exact natural density $p_k(10) = d(D_k(10))$ exists and obeys the same bound).

Corollary 4 (Density-one descent in base ten). *For every real $\varepsilon > 0$, there exists $N_0 \in \mathbb{N}$ such that for all integers $N \ge N_0$,*

$$\frac{|\{2 \le n \le N : f_{10}^j(n) \ge n \text{ for all } j \ge 1\}|}{N} \le \varepsilon.$$

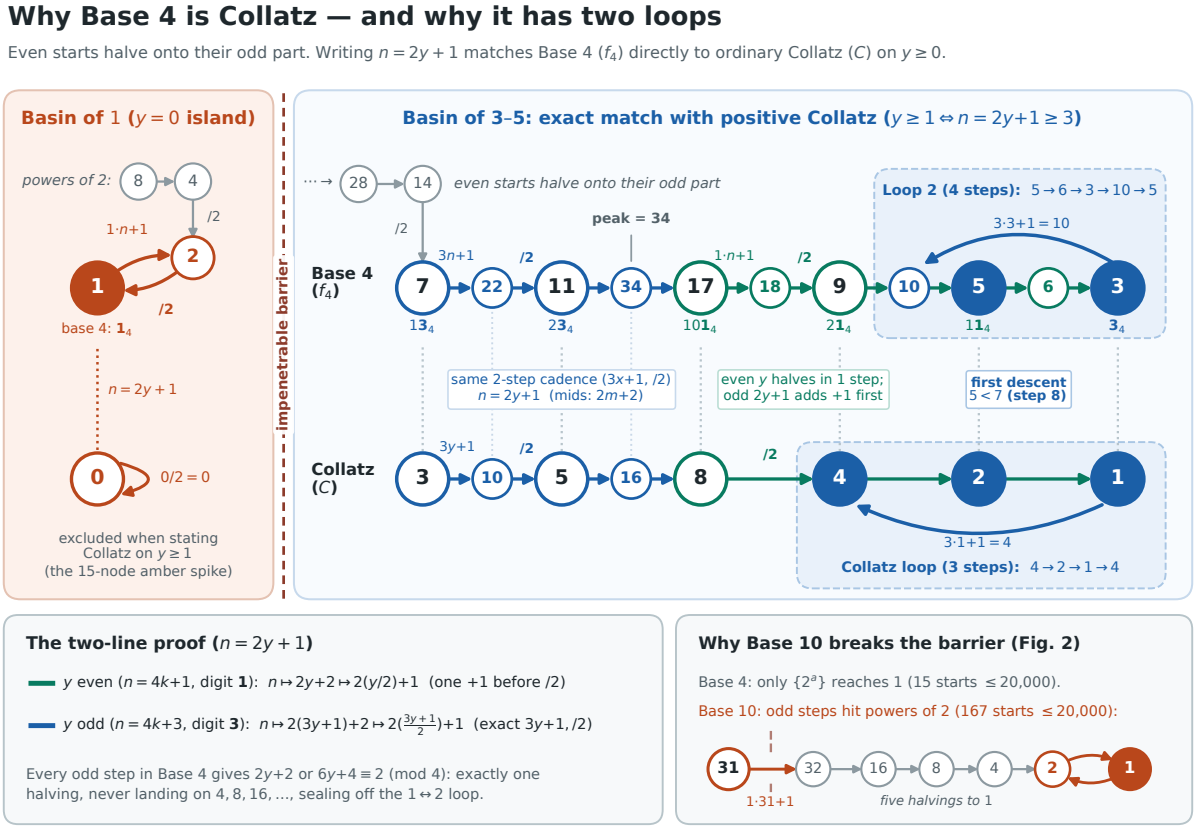


Figure 4. Why Base 4 is ordinary Collatz C on $\mathbb{Z}_{\geq 0}$, and why Base 10 breaks the barrier separating the two loops. Every even start halves onto its odd part. On odd integers, $n = 2y + 1$ aligns f_4 directly with uncompressed Collatz: odd-y steps share the exact two-step cadence ($3x + 1$, then /2), while even-y halvings gain one +1 pause before /2. The $1 \leftrightarrow 2$ loop corresponds to Collatz's excluded fixed point $0 \to 0$, explaining the 15-node amber spike of Figure 2. On the example orbit of $n = 7$ ($y = 3$), the peak is 34 and *first descent* occurs at step 8 ($5 < 7$). In Base 10, odd steps can land on higher powers of 2 (bottom right, $31 \to 32 \to \cdots \to 1$), breaking the barrier and sprouting a branching 1-basin.

Consequently, almost every positive integer (in natural density) eventually drops strictly below its starting value.

Theorem 5 (Elementary quartic upper bound for the basin of one). *Let $a \in (0, 1)$ be the unique positive root of $a + 2a^4 = 1$ and put $\beta = -\log_2 a$ ($501/800 < \beta < 626{,}383/10^6$, $\beta \simeq 0.62638214$). Then for every integer $X \geq 1$,*

$$N_1(X) \leq \frac{244}{100} X^{\beta}.$$

(Theorem 5 is the elementary pure-kernel Lean 4 bound; Theorem 18(3) sharpens its prefactor to $N_1(X) < 0.858\, X^{\beta} < X^{\beta}$ for $X \geq 31$, and Theorem 8(1) sharpens the exponent to 0.6245.)

Theorem 6 (Pure-kernel lower bound for the basin of five). *For every integer $X \geq 3$,*

$$N_5(X) \geq \frac{605}{1{,}000{,}000} X^{13/20}.$$

Corollary 7 (Unconditional basin separation). *For every real $\varepsilon > 0$, there exists $X_0 \in \mathbb{N}$ such that for all integers $X \geq X_0$,*

$$\frac{N_1(X)}{N_5(X)} \leq \varepsilon \qquad \text{and} \qquad \frac{N_1^{\text{all}}(X)}{N_5^{\text{all}}(X)} \leq \varepsilon.$$

Theorem 8 (Sharpened basin bounds and 2-adic/Archimedean separation). *For the base-10 map f_{10} (proved in pure-kernel Lean 4 as `Ouroboros.thm_basin_sharp`):*

1. *Integer weight certificates modulo $Q_1 = 643{,}125$ (discharged in pure-kernel Lean 4 via `BasinOneSharp.lean`, `BasinOneUpperSharp.lean`, and `BasinSharpAll.lean`) give, for a constant $C > 0$ and all real $X \geq 1$,*

$$0.0806\,(X/31)^{399/800} \leq N_1(X) \leq C\, X^{0.6245},$$
$$N_1(X) \geq 0.0145\, X^{0.49875},$$

and hence strict 2-adic/Archimedean separation against Theorem 18:

$$\limsup_{X \to \infty} \frac{\log N_1(X)}{\log X} \leq 0.6245 < 0.626382\ldots = \dim_H(\overline{B_1}).$$

2. *Charging individual child weights modulo $Q = 14{,}175$ (using 25,335 integer weights with row-sum cap $L = 2{,}269{,}527$, discharged in `WeightedCharging.lean` and `BasinSharpAll.lean`) gives, for every integer $X \geq 3$,*

$$N_5(X) \geq \frac{98{,}959}{2{,}269{,}527 \cdot 3^{13/20}} X^{13/20} > 0.02134\, X^{0.65},$$

strengthening separation to $N_1(X)/N_5(X) = O(X^{-0.0255})$.

Theorem 9 (Short cycles, quadrillion sweep, and 53,185-step cycle floor). *For f_{10}, the only primitive odd cycles with at most eight odd members are* (1) *and* (5, 13) *(with no size cutoff). Every positive integer $n \le 10^{15}$ reaches* {1, 5}*, and any further odd cycle (whose odd members must all be at least 10^{15}) has halving count K, odd-step count m, and ordinary period $K + m$ satisfying*

$$K \ge 34{,}905, \qquad m \ge 15{,}496, \qquad K + m \ge 53{,}185.$$

Theorem 10 (All-even-base stationary drift classification). *For every even base $b \ge 2$, the odd-term Markov chain P_b admits a stationary distribution π with nonpositive logarithmic drift ($\Delta(\pi) \le 0$) if and only if $b \in \{2, 4, 6, 10\}$.*

Theorem 11 (All-even-base non-descent dichotomy). *For every even base $b \ge 2$, every finite-horizon non-descent density $p_k(b)$ exists, and*

$$p_\infty(b) = 0 \quad \Longleftrightarrow \quad b \in \{2, 4, 6, 10\}.$$

In bases 2, 4, 6, 10, the set of integers eventually descending below their starting value has natural density one. In every other even base, there exists $\eta_b > 0$ such that $p_k(b) \ge \eta_b$ for all $k \ge 1$.

Theorem 12 (Rootless odd-base trapdoor law). *If an odd integer b is divisible by a prime $p \equiv 3 \pmod 4$, then $x^2 + 1 \not\equiv 0 \pmod b$ for every $x \in \mathbb{N}$, and the set Z_b of positive integers reaching $b\mathbb{Z}_{\ge 1}$ (positive multiples of b) under f_b equals $b\mathbb{Z}_{\ge 1}$ exactly.*

Theorem 13 (Odd-base survey through 1,999). *For each odd base $b \ge 3$, write Z_b for positive starts reaching a multiple of b, R_b for starts reaching 1, and E_b for starts tending to infinity. Finite-cylinder absorption lower bounds produce masses τ_b for Z_b and ρ_b for E_b, and whenever $\tau_b + \rho_b = 1$ all three natural densities exist with $d(Z_b) = d(R_b) = \tau_b$ and $d(E_b) = \rho_b$. Across all 999 odd bases $3 \le b \le 1{,}999$, the absorption certificate establishes $\tau_b + \rho_b = 1$ in 905 bases (27 trap-only with $\tau_b = 1$, and 878 split with rational $\tau_b, \rho_b > 0$).*

■ 2. BASE TEN: COMPETING CORALS AND FORWARD DESCENT

Stay inside base 10, where Figure 4 just broke open the wall around $1 \leftrightarrow 2$. In classical Collatz only one nontrivial cycle is known, so no competing basins have been observed (and the integer basin of 1 is unconditionally dense in $\mathbb{Z}_2$); in base 10, breaking that barrier creates a genuine contest between two infinite attracting trees, for which no $3x+1$ analogue was found in the searches logged. Trace the first 30,000 positive integers backward from their destinations 1 and 5, letting orbits that share a suffix fuse into a single strand (Figure 5). Before we prove a single inequality, the asymmetry leaps off the page: 29,813 trajectories build the towering blue coral of 5, while just 187 build the fragile amber sprig of 1.

How can we tell mathematically that typical decimal orbits fall below their starting value? Abbreviate $d(n) = n \bmod 10$ and $T = T_{10}$. Write $r = n \bmod 5$. An odd integer in class r has a unique last digit $\delta(r)$, given in order $r = 0, 1, 2, 3, 4$ by $(5, 1, 7, 3, 9)$. Because halving multiplies by 3 mod 5, one shortcut step sends residue r to

$$\begin{aligned} E(r) &= 3r \bmod 5 && \text{(even } n\text{)}, \\ O(r) &= 3(r^2 + 1) \bmod 5 && \text{(odd } n\text{)}. \end{aligned} \tag{3}$$

Notice a remarkable structural quirk right away: $O(r) = 0$ precisely for $r \in \{2, 3\}$ (digits 7 and 3), and $E(0) = 0$. Thus *every odd term ending in 3 or 7 lands next on an odd term ending in 5, and no other odd digit ever sends flow into 5* (Figure 7; formalized as `lemma_after_3_or_7_is_5`).

Lemma 14 (Base-ten parity coding bijection). *For every integer $k \ge 0$, the map sending a residue class $r \in \mathbb{Z}/(5 \cdot 2^k)\mathbb{Z}$ to its starting residue r mod 5 together with its length-k shortcut parity vector $(T^j(r) \bmod 2)_{0 \le j < k}$ is bijective.*

Proof. Start at $(A_0, B_0) = (1, 0)$. An even step leaves (A_j, B_j) unchanged; an odd step replaces it by $(dA_j,\ dB_j + 2^j)$, where $d = \delta(r_j)$ is determined by the current residue modulo 5, and advances the residue via E or O. Induction gives the affine identity $2^j T^j(n) = A_j n + B_j$ with odd A_j and $B_j \ge 0$ (`lemma_coding_affine`). If a length-j word specifies n_0 mod $(5 \cdot 2^j)$, its two lifts n_0 and $n_0 + 5 \cdot 2^j$ have j-th iterates differing by $5A_j$, which is odd; hence one lift is even and the other is odd. Induction on k proves bijectivity. □

For a uniformly chosen residue modulo $5 \cdot 2^k$, each successive parity bit is therefore an independent fair coin—an exact finite-counting identity generalizing Terras's parity theorem for $3x + 1$ [1]. Direct substitution in (3) gives the unique stationary distribution $\pi = (5, 3, 1, 4, 2)/15$, which also governs the embedded odd-to-odd chain (Figure 7). Since an odd step is followed on average by two halvings, the stationary logarithmic change per odd step is

$$\begin{aligned} \Delta_{10} &= \sum_{r=0}^{4} \pi_r \log \delta(r) - 2 \log 2 \\ &= \frac{1}{15} \log\left(\frac{3^8 \cdot 5^5 \cdot 7}{2^{30}} \right) \simeq -0.134161. \end{aligned}$$

Proof of Theorem 3 and Corollary 4. Fix $k \ge 1$ and set $H_j = A_j / 2^j$ on each residue class modulo $5 \cdot 2^k$. Because A_j is odd, we have $H_j \ne 1$ for every $j \ge 1$. When $H_j < 1$, the descent bound $T^j(n) < n$ holds for every n above $B_j/(2^j - A_j)$; when $H_j > 1$, we instead have $T^j(n) > n$ throughout the entire class. Discarding the finitely many small exceptions across finitely many classes proves that the asymptotic density $p_k(10) = d(D_k(10))$ exists and equals the fraction of classes with $H_j > 1$ for every $1 \le j \le k$, which is at most the fraction with $H_k \ge 1$.

Tilt the five-state transition matrix by $t = 11/100$:

$$(M_t u)_r = \tfrac{1}{2}(1/2)^t u_{E(r)} + \tfrac{1}{2}(\delta(r)/2)^t u_{O(r)},$$

so that $\mathbb{E}[H_k^t] = \mathbf{1}^\top M_t^k \mathbf{1}/5$. Take the positive vector $u = (1{,}000{,}000{,}837{,}748{,}965{,}453{,}964{,}025{,}944{,}908)$ and denominator $D = 10^{12}$. The five integer numerators

$$\begin{aligned} h_1 &= 926{,}588{,}061{,}891, & h_3 &= 1{,}045{,}610{,}747{,}322, \\ h_5 &= 1{,}106{,}046{,}538{,}591, & h_7 &= 1{,}147{,}750{,}484{,}796, \\ h_9 &= 1{,}179{,}922{,}211{,}263 \end{aligned}$$

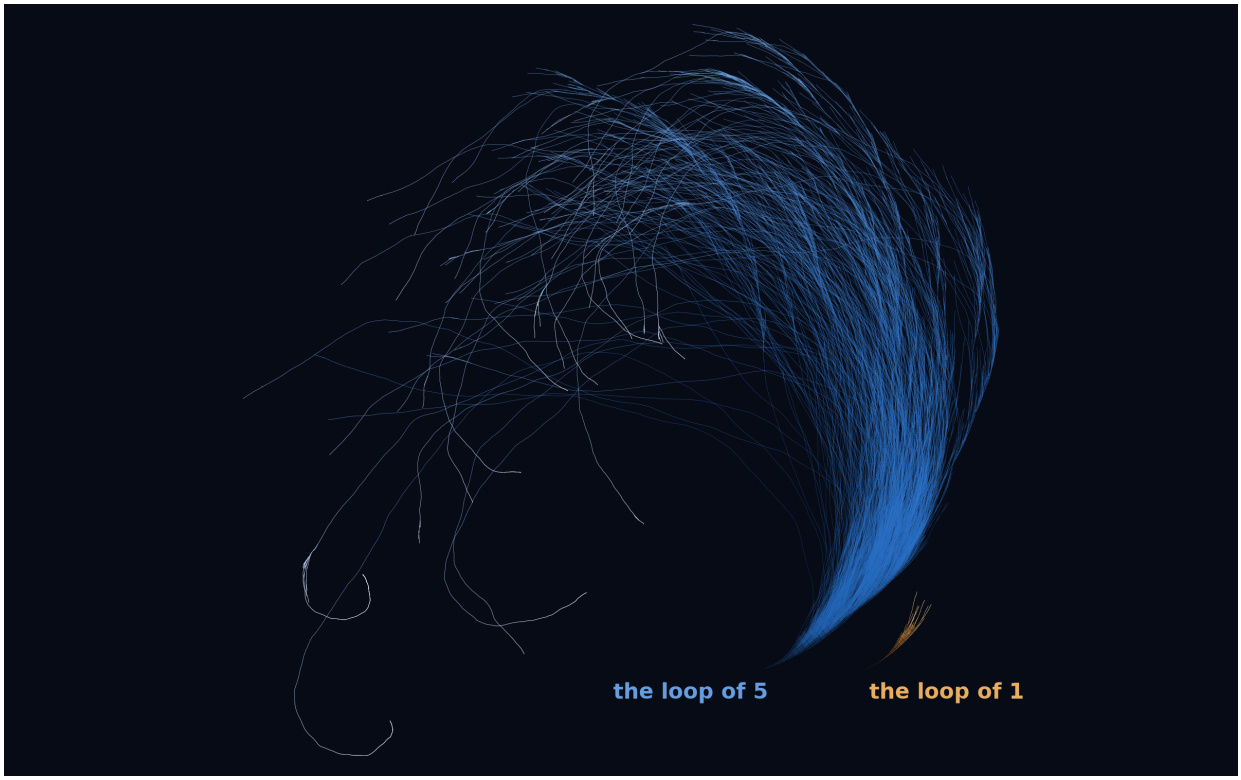


Figure 5. The two orbit corals of base 10. Every integer $1 \le n \le 30{,}000$ traces its path to first arrival at 1 or 5: 29,813 starts form the blue coral of 5, and 187 form the amber coral of 1. Reading outward from the roots, each halving bends left and each odd move bends right; shared segments represent identical iterates.

One parity bit chooses the next residue

residue	digit	even →	odd →
0	5	0	3
1	1	3	1
2	7	1	0
3	3	4	0
4	9	2	1

Which parity words avoid a descent?

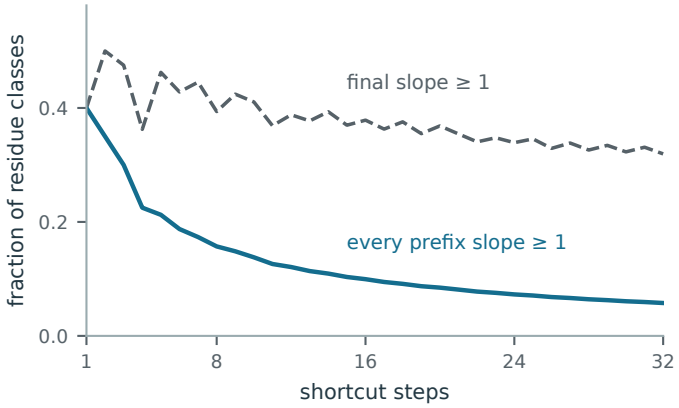


Figure 6. Base-10 finite parity coding. Left: the deterministic transition table (3) modulo 5. Right: exact fraction of the $5 \cdot 2^k$ residue classes avoiding contraction through $k \le 32$ shortcut steps, comparing the final-slope test $A_k \ge 2^k$ against the true first-descent condition $A_j \ge 2^j$ for all $1 \le j \le k$.

each satisfy $h_d^{100}\, 2^{11} > D^{100}\, d^{11}$, certifying $(d/2)^{11/100} < h_d/D$. Substituting directly into all five rows confirms

$$10^6 (h_1 u_{E(r)} + h_{\delta(r)} u_{O(r)}) < 2D \cdot 996{,}423\, u_r,$$

$$\frac{\sum_r u_r}{5 \min_r u_r} = \frac{2{,}356{,}067}{2{,}094{,}370} < 1.124953.$$

Since M_t is nonnegative, $\mathbf{1} \le u / \min u$ and Markov's inequality yield $p_k(10) \le \mathbb{E}[H_k^t]$, giving Theorem 3 (Figure 6). Any integer $n \ge 2$ whose ordinary f_{10}-orbit never drops below itself lies in every $D_k(10)$ (as $d(n)n + 1 > n$); letting $k \to \infty$ proves Corollary 4. □

Density-one first descent shows that typical decimal orbits turn downward, but it cannot tell us *which* of the two corals in Figure 5 captures them. For that, we reverse time and

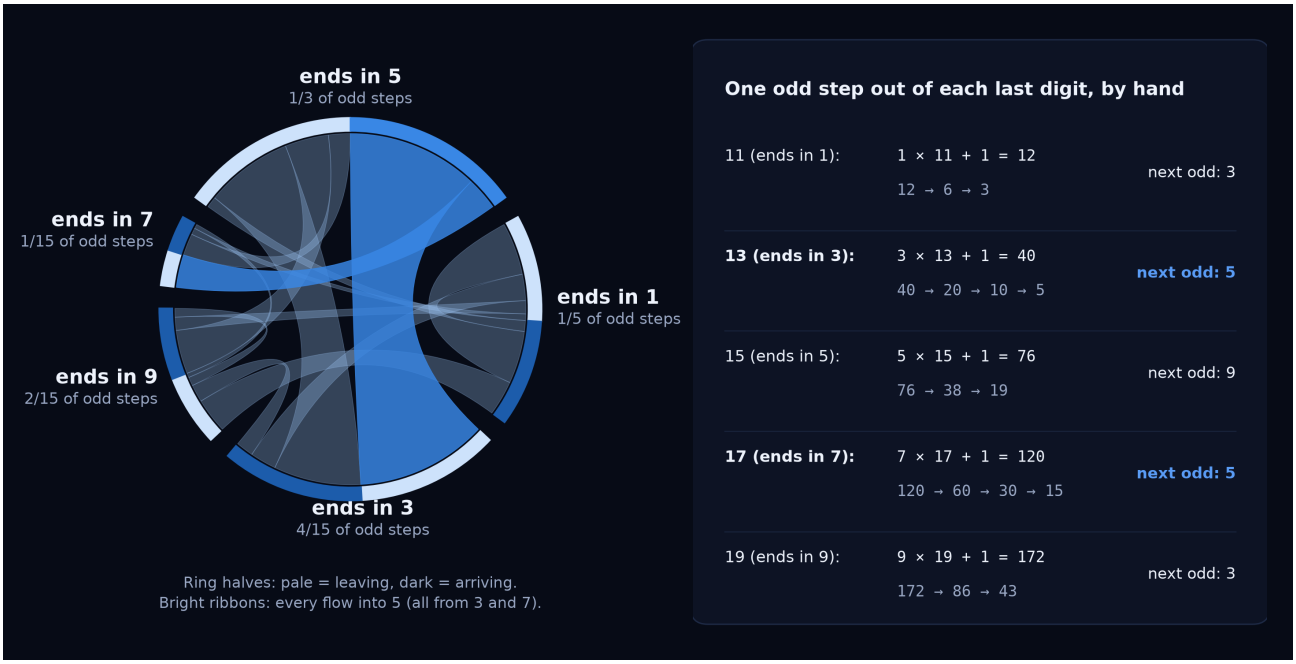


Figure 7. Stationary odd-to-odd digit flow in base 10. Pale arcs emit flow and dark arcs receive it; ribbon width measures joint frequency. The bright blue ribbons highlight the forced funnel: digits 3 (4/15) and 7 (1/15) account for all transitions into digit 5 (1/3). Right: one hand-worked odd step out of each last digit, including all intervening halvings.

climb both inverse trees.

3. BACKWARD TOWARD ONE: THE THIN AMBER BASIN

Strip all powers of two after an odd step to define the odd-to-odd decimal map

$$F(n) = \frac{d(n)\,n+1}{2^v}, \qquad v = v_2(d(n)\,n+1) \quad (n \text{ odd}).$$

For example, $F(991) = 31$ and $F(31) = 1$ use only multiplier 1, whereas $F(195) = 61$ and $F(61) = 31$ bring multiplier 5 into the same tree. Plotting the unbounded ordinary-step predecessor trees of 1 and 5 through depth 26 (Figure 8) immediately exposes a stark arithmetic selection rule: among all 261 odd predecessors of 1 within 26 steps (excluding 1), *zero* end in 3 and *zero* end in 9.

The modulo-30 residue law and binary runs.

Proposition 15 (Three residue classes for the basin of one). *Every odd integer n reaching 1 under f_{10} is congruent to 1, 15, or 17 modulo 30 (equivalently $n \bmod 15 \in \{0,1,2\}$).*

Lemma 16 (Exhaustive odd-to-odd moves on $A \cup B \cup C$). *Label the three odd classes modulo 15 by $A \equiv 1$, $B \equiv 0$, and $C \equiv 2 \pmod{15}$. Whenever $o \in A \cup B \cup C$, $v \geq 1$, and an odd integer m satisfies $d(m)\,m+1 = 2^v o$, we have $d(m) \notin \{3,9\}$, $m \in A \cup B \cup C$, and the move $m \xrightarrow{v} o$ belongs to one of the five rows below:*

Move $m \to o$	Digit d	Admissibility condition
$A \to A$	1	$v \equiv 1 \pmod 4$
$A \to C$	1	$v \equiv 0 \pmod 4$
$B \to A$	5	$2^v o \equiv 1 \pmod{25}$
$B \to C$	5	$2^v o \equiv 1 \pmod{25}$
$C \to B$	7	$2^v o \equiv 1 \pmod 7$

Proof of Proposition 15 and Lemma 16. For any odd $r \bmod 15$, the target residue of F is congruent to $(\delta(r \bmod 5)\,r+1)\,2^{-v}$ modulo 15. Since 2 has multiplicative order 4 modulo 15, landing in 1 or 2 (mod 15) requires the numerator to lie in $\{1,2,4,8\} \pmod{15}$, whereas landing in 0 (mod 15) requires it to vanish. For $r = 0,1,\ldots,14$, the numerators $\delta(r \bmod 5)\,r+1 \bmod 15$ are

$$(1,2,0,10,7,\ 11,7,5,10,7,\ 6,12,10,10,7).$$

Only $r \in \{0,1,2\}$ land in $\{0,1,2,4,8\}$. Backward induction from $1 \in A$ therefore confines the entire odd basin of 1 to A, B, C. Checking the powers of 2 modulo 15—together with exact divisibility $d \mid 2^v o - 1$ and $m \equiv d \pmod 5$ (which strengthens $5 \mid 2^v o - 1$ to $25 \mid 2^v o - 1$ for $d = 5$, forcing $v \equiv 0 \pmod 4$ on $B \to A$ and $v \equiv 3 \pmod 4$ on $B \to C$)—yields the exact table of Lemma 16. □

Inside class A sits a binary characterization of paths that use *only* digit 1.

Proposition 17 (Binary run-of-ones characterization). *For every $u \geq 0$, the odd integer $n = 2u+1$ reaches 1 under f_{10} via digit-1 odd steps alone if and only if every maximal run of 1-bits in the binary expansion of u has length divisible by 4.*

Proof. If $u = 2^{r+1}c + 2^r - 1$ ends in r ones, then $n+1 = 2^{r+1}(2c+1)$, so $F(n) = 2c+1$ deletes the trailing block 01^r from u; reducing modulo 5 when both n and $F(n)$ end in 1 forces $2^r \equiv 1 \pmod 5$, hence $4 \mid r$ (for instance, $(991-1)/2 = (111101111)_2$). Induction on u proves both directions. □

Exact 2-adic Cantor closure and quartic bounds. Rather than tracking odd-to-odd jumps all at once, watch the short-

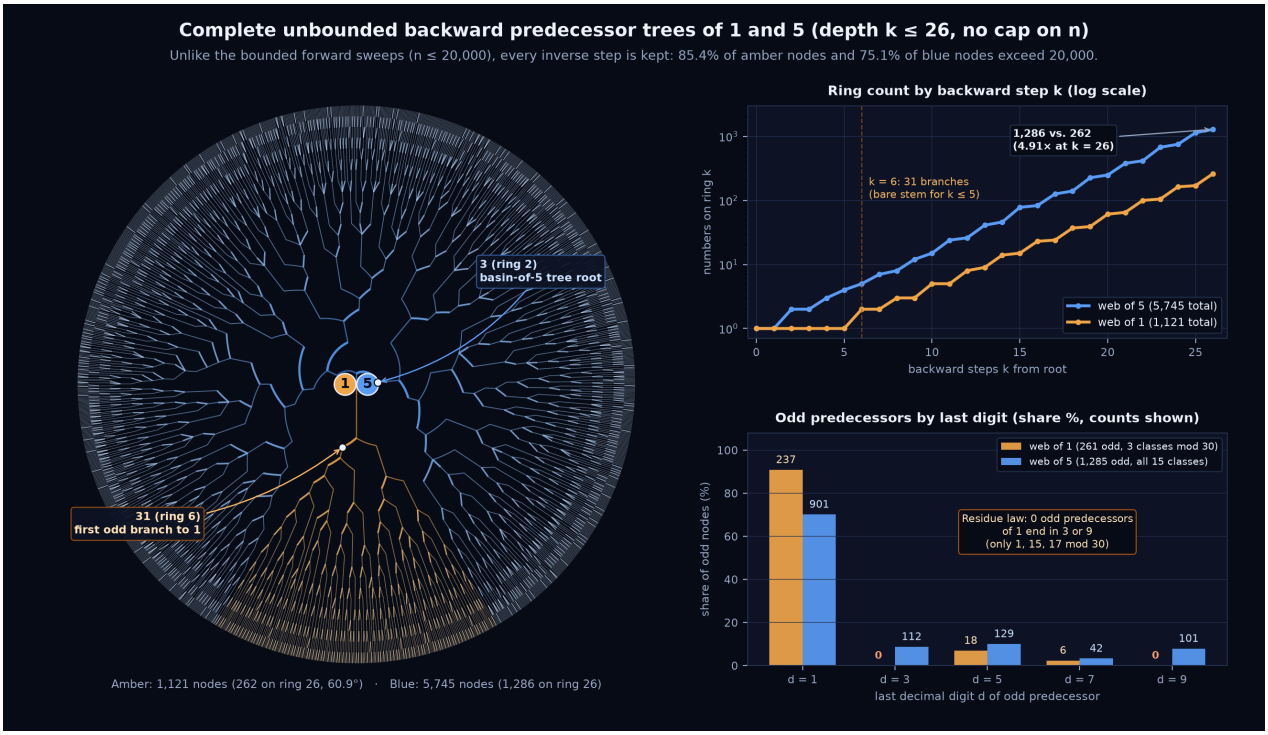


Figure 8. Complete unbounded backward predecessor trees of 1 and 5 through 26 ordinary steps (no cap on n, omitting periodic repeats). At depth 26, the outer ring holds 262 nodes for 1 (1,121 total) versus 1,286 for 5 (5,745 total). The bare stem $1, 2, 4, 8, 16, 32$ branches for the first time at 31 (ring 6), and the lower-right histogram shows zero odd predecessors of 1 ending in 3 or 9.

cut map $T = T_{10}$ act one halving at a time on the three admissible sheets $n = 30k + r$ with $r \in \{1, 15, 17\}$. Expanding $k = 2k' + s$ for $s \in \{0, 1\}$ reads off a single binary digit of k and advances $T(n)$ modulo 30:

- from 1 or 2 (mod 30): $s = 0 \mapsto 1$, $s = 1 \mapsto 16$;
- from 0 or 17 (mod 30): $s = 0 \mapsto 0$, $s = 1 \mapsto 15$;
- from 15 or 16 (mod 30): $s = 0 \mapsto 8$, $s = 1 \mapsto 23$ (forbidden);
- from 8 (mod 30): $s = 0 \mapsto 4$, $s = 1 \mapsto 19$ (forbidden);
- from 4 (mod 30): $s = 0 \mapsto 2$, $s = 1 \mapsto 17$.

Only the eight residues $\{0, 1, 2, 4, 8, 15, 16, 17\}$ modulo 30 are reachable while staying in $\{1, 15, 17\}$ (mod 30), and the first forbidden odd residue encountered is always 19 or 23 (mod 30). Grouping bisimilar residues into four states $S_0 = \{0, 1, 2, 17\}$, $S_1 = \{15, 16\}$, $S_2 = \{8\}$, and $S_3 = \{4\}$ collapses the modulo-30 survival dynamics to the four-state automaton of Figure 9A:

$$S_0 \xrightarrow{0} S_0,$$
$$S_0 \xrightarrow{1} S_1 \xrightarrow{0} S_2 \xrightarrow{0} S_3 \xrightarrow{0,1} S_0.$$

Consequently, a binary itinerary starting in S_0 survives every halving test if and only if it belongs to the infinite prefix-code Cantor set $\Sigma = \{0,\ 1000,\ 1001\}^{\omega} \subset \{0, 1\}^{\mathbb{N}}$.

Let $B_1 = \{n \in \mathbb{Z}_{\geq 1} \text{ odd} : f_{10}^{j}(n) = 1 \text{ for some } j \geq 0\}$ be the odd basin of 1, embedded in the 2-adic integers $(\mathbb{Z}_2, |\cdot|_2)$ with the standard ultrametric $|x - y|_2 = 2^{-v_2(x-y)}$. For each $r \in \{1, 15, 17\}$, let $K_r = \{k \in \mathbb{Z}_{\geq 0} : 30k + r \in B_1\}$, and let $\mathcal{K}_r \subset \mathbb{Z}_2$ be the compact set of 2-adic coordinates k whose congruence class k mod 2^h survives the first h halvings without hitting a forbidden odd residue modulo 30 for every $h \geq 0$. Remarkably, $\mathcal{K}_r$ is not merely a necessary outer shell: it is the *exact* 2-adic closure of the true basin of 1.

Theorem 18 (Exact 2-adic Cantor closure and elementary quartic bounds). *Let $a_1, a \in (0, 1)$ be the unique positive roots of $a_1 + a_1^4 = 1$ and $a + 2a^4 = 1$, and put $\gamma = -\log_2 a_1 \simeq 0.464958$ and $\beta = -\log_2 a \simeq 0.62638214$ $(0.626382 < \beta < 0.626383)$.*

1. Exact 2-adic closure: *For each $r \in \{1, 15, 17\}$, the 2-adic closure of K_r equals $\mathcal{K}_r$, and hence the 2-adic closure of the odd basin B_1 in $\mathbb{Z}_2$ is*
$$\overline{B_1} = (30\mathcal{K}_1 + 1) \cup (30\mathcal{K}_{15} + 15) \cup (30\mathcal{K}_{17} + 17).$$
Every nonempty 2-adic ball meeting $\overline{B_1}$ contains infinitely many positive integers reaching 1 and infinitely many exiting $\{1, 15, 17\}$ (mod 30) at 19 or 23 (mod 30), and $\overline{B_1}$ is nowhere dense in $\mathbb{Z}_2$.
2. Self-similar isometry and Hausdorff dimension: *Equip $\{0, 1\}^{\mathbb{N}}$ with $d(x, y) = 2^{-\inf\{j : x_j \neq y_j\}}$. The halving-itinerary map $\Psi_r : \mathcal{K}_r \to \{0, 1\}^{\mathbb{N}}$ is a surjective 2-adic isometry onto $\Sigma = \{0, 1000, 1001\}^{\omega}$ for $r \in \{1, 17\}$ and onto $000\Sigma \sqcup 001\Sigma$ for $r = 15$. Consequently $\overline{B_1}$ has positive finite β-dimensional Hausdorff measure $(\frac{1}{2} \leq \mathcal{H}^{\beta}(\overline{B_1}) \leq a^{-2} + a < 3.031)$ and exact 2-adic Hausdorff and box-counting dimension*
$$\dim_H(\overline{B_1}) = \dim_B(\overline{B_1}) = \beta = -\log_2 a \simeq 0.62638214.$$
3. Elementary integer bounds: *For every real $X \geq 1$,*
$$a_1^9\, X^{\gamma} \leq N_1(X) \leq X^{\beta}.$$
In particular, the full basin of 1 (including even starts) has natural density zero.

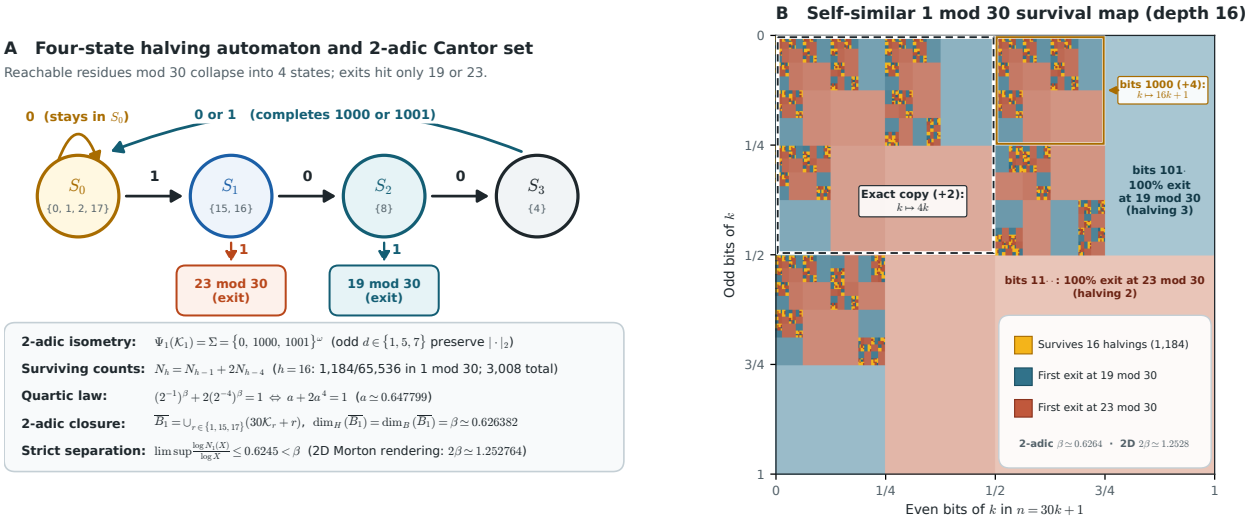


Figure 9. (**A**) Exact four-state collapse of the eight reachable residues modulo 30 under single halving steps T_{10}, yielding the prefix-code Cantor set $\Sigma = \{0, 1000, 1001\}^\omega$ and the recurrence $N_h = N_{h-1} + 2N_{h-4}$. (**B**) The 1 mod 30 survival landscape at depth $h = 16$ (256×256 classes), interleaving even and odd binary digits of k in $n = 30k + 1$ into $[0, 1]^2$. Just 1,184 of 65,536 classes survive (gold; 3,008 of 983,040 across all 15 odd classes modulo 30). The upper-left quarter ($k \mapsto 4k$, $4n - 3 \to 2n - 1 \to n$) and the `1000` sub-square ($k \mapsto 16k + 1$) reproduce the entire square identically at depths +2 and +4.

Proof of Theorem 18 and Theorem 5. Write $n_0 = 30(2^h k' + k_0) + r$ with $r \in \{1, 15, 17\}$. Induction on $j \le h$ shows that along any surviving path, $n_j = T^j(n_0) = 30 \cdot 2^{h-j} D_j k' + b_j$, where $D_j = \prod_{i<j,\ n_i \text{ odd}} d(n_i)$ is an odd product of digits in $\{1, 5, 7\}$ and $b_j = T^j(30k_0 + r)$ is independent of k'. Because $2 \mid 2^{h-j}$ for $j < h$, both parity and last digit of n_j depend only on b_j mod 30, so survival through h halvings depends solely on k mod 2^h. Thus $\mathcal{K}_r^{(h)}$ is a clopen union of $N_h(r)$ balls of radius 2^{-h}, and $\mathcal{K}_r = \bigcap_{h \ge 0} \mathcal{K}_r^{(h)}$ is compact.

Part 2 (isometry and dimension). For $k \in \mathcal{K}_r$ and $j \ge 0$, choose any nonnegative integer lift $\tilde{k} \equiv k \pmod{2^{j+1}}$ and write $T^j(30\tilde{k} + r) = 30q_j + r_j$ ($0 \le r_j < 30$); the depth-$(j+1)$ formula above changes $T^j(30\tilde{k} + r)$ by a multiple of 60, so the j-th itinerary bit $s_j(k) = q_j \bmod 2 \in \{0, 1\}$ is well defined. Take distinct $k, k' \in \mathcal{K}_r$ with $v_2(k - k') = m$, and fix integer lifts $\tilde{k}, \tilde{k}' \equiv k, k' \pmod{2^{m+1}}$. Along $0 \le j \le m$, both lifts follow the same residues modulo 30 and odd multipliers D_j, giving $\tilde{n}_j - \tilde{n}'_j = 30D_j(\tilde{k} - \tilde{k}')/2^j$ in $\mathbb{Z}$. For $j < m$ this is divisible by 60, so $s_j(k) = s_j(k')$; at $j = m$, both D_m and $(\tilde{k} - \tilde{k}')/2^m$ are odd, so $\tilde{n}_m - \tilde{n}'_m \equiv 30 \pmod{60}$ and $s_m(k) \ne s_m(k')$. Hence $d(\Psi_r(k), \Psi_r(k')) = |k - k'|_2$ identically. Conversely, each transition bit s_j bisects a class modulo 2^j into two classes modulo 2^{j+1}, so by compactness of $\mathbb{Z}_2$ every infinite word in the automaton's language is realized uniquely: $\Psi_r(\mathcal{K}_r) = \Sigma$ for $r \in \{1, 17\}$ and $000\Sigma \sqcup 001\Sigma$ for $r = 15$ (starting from state S_1).

Assign the Bernoulli product measure μ on Σ with word weights $p(0) = a$ and $p(1000) = p(1001) = a^4$ (summing to $a + 2a^4 = 1$). Every length-h valid prefix w ends in state S_0, S_1, S_2, or S_3, giving cylinder mass $\mu(C_w) = a^h$, $2a^{h+3}$, $2a^{h+2}$, or $2a^{h+1}$ respectively. Since $2a^3 < 2a^2 < 1 < 2a$, every ambient length-h cylinder $C_w \subset \{0, 1\}^{\mathbb{N}}$ (of diameter 2^{-h}) meeting Σ satisfies

$$2a^3\,(2^{-h})^\beta \le \mu(C_w) \le 2a\,(2^{-h})^\beta \qquad (a = 2^{-\beta}). \tag{4}$$

Summing (4) over the $N_h(1)$ disjoint length-h cylinders covering Σ yields $(2a)^{-1}a^{-h} \le N_h(1) \le (2a^3)^{-1}a^{-h}$ for all $h \ge 0$ (with $N_h(1) = N_h(17)$ obeying $N_h = N_{h-1} + 2N_{h-4}$ for $h \ge 4$, and $N_h(15) = 2N_{h-3}(1)$ for $h \ge 3$). In the ultrametric space $\{0, 1\}^{\mathbb{N}}$, any subset $U \subseteq \Sigma$ of positive diameter 2^{-h} lies in a single cylinder C_w of length h, so $\mu(U) \le 2a\,(\operatorname{diam} U)^\beta$ (and $\mu(\{x\}) = 0$ for singletons). For any countable cover $\{U_i\}$ of Σ, $1 = \mu(\Sigma) \le 2a \sum (\operatorname{diam} U_i)^\beta$, while covering by all $N_h(1)$ length-h cylinders gives $\mathcal{H}^\beta(\Sigma) \le (2a^3)^{-1}$. Thus $(2a)^{-1} \le \mathcal{H}^\beta(\Sigma) \le (2a^3)^{-1}$. For each $r \in \{1, 15, 17\}$, the affine map $k \mapsto 30k + r$ contracts 2-adic distances by $|30|_2 = 1/2$, scaling $\mathcal{H}^\beta$ by $(1/2)^\beta = a$. Monotonicity and finite subadditivity on $\overline{B_1} = \bigcup_r (30\mathcal{K}_r + r)$ (once Part 1 is established) therefore give $\frac{1}{2} \le \mathcal{H}^\beta(\overline{B_1}) \le 2 \cdot a(2a^3)^{-1} + 2a^4(2a^3)^{-1} = a^{-2} + a < 3.031$, and the same upper and lower cylinder counts establish $\dim_H(\overline{B_1}) = \dim_B(\overline{B_1}) = \beta$.

Part 1 (exact 2-adic closure via the +30 inverse commutator). Every $k \in K_r$ passes all modulo-30 halving tests by Proposition 15, so $\overline{K_r} \subseteq \mathcal{K}_r$. Conversely, take any $k \in \mathcal{K}_r$ and any 2-adic ball $k_0 + 2^h\mathbb{Z}_2$ around k. Inside the eight-state automaton, every state reaches 1 (mod 30) in at most 5 halvings and an exit (19 or 23 (mod 30)) in at most 3 halvings. Following the exit path of length $d \le 3$ specifies an open sub-ball $k_{\text{exit}} + 2^{h+d}\mathbb{Z}_2 \subseteq (k_0 + 2^h\mathbb{Z}_2) \setminus \mathcal{K}_r$ whose positive integer elements all leave $\{1, 15, 17\}$ (mod 30) at 19 or 23 (mod 30). Following the path to 1 (mod 30) yields a subcylinder $k_1 + 2^{h'}\mathbb{Z}_2 \subseteq k_0 + 2^h\mathbb{Z}_2$ ($h' \le h + 5$) on which $T^{h'}(30(2^{h'}k'' + k_1) + r) = 30D'k'' + b$, with $D' = 5^{e_5}7^{e_7}$ odd and $b \equiv 1 \pmod{30}$.

To hit b mod $30D'$ with an arbitrarily large member of B_1, observe that for any integer $u \equiv 1 \pmod{10}$, both backward

digit-1 moves $L(u) = 2u - 1$ and $H(u) = 32u - 1$ still end in 1 and return to u under f_{10}. Composing them in opposite orders shifts the start by exactly 30:

$$\begin{aligned} P(u) &= H(L(u)) = 64u - 33, \\ Q(u) &= L(H(u)) = 64u - 3 = P(u) + 30. \end{aligned} \tag{5}$$

(For example, at $u = 31$ the odd paths are $1{,}951 \to 61 \to 31$ and $1{,}981 \to 991 \to 31$; cf. [10] [to check] for modular affine translations in the $3x + 1$ graph.) For any odd modulus $q \geq 1$, P permutes $\mathbb{Z}/q\mathbb{Z}$; choosing $s \geq 1$ with $P^s \equiv \mathrm{id} \pmod q$, the legal inverse composite $Q \circ P^{s-1}$ adds $30 \pmod q$ while preserving u mod 30 and returning to u, and iterating P^s makes representatives arbitrarily large since $P(x) > x$ for all $x \geq 1$. Applying this at $u = 1$ with $q = 15D'$ reaches every residue congruent to 1 (mod 15) modulo $15D'$; oddness upgrades this to $o \equiv b \pmod{30D'}$ with $o \in B_1$ and $o \geq b$. Setting $k'' = (o - b)/(30D') \in \mathbb{Z}_{\geq 0}$ places $2^{h'}k'' + k_1 \in K_r \cap (k_0 + 2^h\mathbb{Z}_2)$. Hence $\overline{K_r} = \mathscr{K}_r$, and because $B_1 = \bigcup_r (30K_r + r)$ is a finite union of affine images of K_r, $\overline{B_1} = \bigcup_r (30\mathscr{K}_r + r)$. Any nonempty 2-adic ball $x + 2^m\mathbb{Z}_2$ meeting $\overline{B_1}$ contains some $30k + r$ ($k \in \mathscr{K}_r$), so taking $h = \max(0, m - 1)$ places $30(k_0 + 2^h\mathbb{Z}_2) + r$ inside $x + 2^m\mathbb{Z}_2$, giving both infinitely many B_1 members and an open exit sub-ball (hence nowhere-denseness).

Part 3 (*elementary integer bounds and Theorem 5*). For the lower bound, build all-digit-1 chains backward from 1 via $o \mapsto 2^v o - 1$ with $v \equiv 1 \pmod 4$ and initial $v \geq 5$ (entering $31, 511, \ldots$). Each step stays in class $A \equiv 1 \pmod{15}$ of Proposition 15, strictly increases the vertex, and is uniquely invertible via $v_2(o + 1)$. Any chain of total halving count $m > 0$ ends strictly below 2^m. Let $U(m)$ count ordered compositions of m into parts $\equiv 1 \pmod 4$ (with $U(0) = 1$). Then $U(0) = \cdots = U(4) = 1$, and for $m \geq 5$ either stripping a leading 1 or subtracting 4 from a larger first part gives $U(m) = U(m - 1) + U(m - 4)$, so $U(m) \geq a_1^{4-m}$ by induction. Requiring the first part to be at least 5 leaves $U(m - 4)$ distinct odd starts below 2^m. Taking $m = \lfloor \log_2 X \rfloor$ gives $N_1(X) \geq a_1^9 X^\gamma$ for $X \geq 32$, and $n = 1$ suffices for $X < 32$ (formalized in `thm_basin1_lower`).

For the upper bound, if $1 \leq X < 31$ then $15 \to 76 \to 38 \to 19$ and $17 \to 120 \to 60 \to 30 \to 15$ (exiting at 19) both exit, leaving $N_1(X) = 1 \leq X^\beta$. For $X \geq 31$, choose the unique integer $h \geq 1$ with $30 \cdot 2^{h-1} < X \leq 30 \cdot 2^h$. Each congruence class modulo $30 \cdot 2^h$ meets $[1, X]$ at most once, while Part 2 gives $N_h(1) = N_h(17) \leq (2a^3)^{-1}a^{-h}$ and $N_h(15) \leq a^{-h}$ for all $h \geq 0$ (for $h \in \{0, 1, 2\}$, $N_h(15) = 1 \leq a^{-h}$; for $h \geq 3$, $N_h(15) = 2N_{h-3}(1) \leq 2(2a^3)^{-1}a^{-(h-3)} = a^{-h}$). Using $2a^4 = 1 - a$ (so $a^{-3} + 1 = (1 + a)/(1 - a)$) and $a^{-h} = (2^h)^\beta < (X/15)^\beta$,

$$\begin{aligned} N_1(X) \leq N_h(1) + N_h(15) + N_h(17) &\leq \frac{1 + a}{1 - a}\, a^{-h} \\ &< \frac{1 + a}{(1 - a)\, 15^\beta}\, X^\beta < 0.858\, X^\beta < X^\beta. \end{aligned}$$

Summing over $X/2^j$ includes all even multiples and preserves natural density zero (`thm_basin1_all_upper` and `thm_basin1_density_zero`). In Lean 4, the odd-step 3×3 transfer matrix proves Theorem 5 directly with prefactor $244/100 = 2.44$ via `thm_basin1_upper`, together with `thm_basin1_upper_explicit` for exponent $626{,}383/10^6$; the complete 2-adic closure, isometry, Hausdorff measure/dimension, and sharp X^β statement of Theorem 18 is formalized in `CantorClosure.lean` as `Ouroboros.thm_cantor`. □

Remark (*planar self-similarity and embedding dimension*). In Figure 9B, the bit-interleaved Morton map $\Phi(\sum_{j\geq 0} b_j 2^j) = (\sum b_{2d}2^{-(d+1)}, \sum b_{2d+1}2^{-(d+1)})$ sends each 2-adic ball of radius 2^{-2m} to a dyadic square of side 2^{-m}, taking the square root of the metric (Hölder exponent 1/2, pulling back side $\varepsilon = 2^{-m}$ to 2-adic radius $\varepsilon^2 = 2^{-2m}$) and therefore doubling the box-counting dimension to $\dim_B(\Phi(\mathscr{K}_1)) = 2\beta \simeq 1.2527643$ (a coordinate effect of the planar embedding, whereas $\beta \simeq 0.6263821$ is intrinsic to $\mathbb{Z}_2$). Because digit-1 moves introduce no odd multiplier 5 or 7, the odd-depth shift $k \mapsto 2k$ ($2n - 1 \to n$, +1 halving) swaps even and odd bits into a transposed, half-width affine image $\Phi(2k) = (y/2, x)$ on $[0, 1/2] \times [0, 1]$, whereas the even-depth shifts—the upper-left quarter $k \mapsto 4k$ ($\Phi(4k) = (x/2, y/2)$, $4n - 3 \to 2n - 1 \to n$, +2 halvings; e.g., $31 \mapsto 121$ and $91 \mapsto 361$) and the `1000` sub-square $k \mapsto 16k + 1$ ($\Phi(16k + 1) = (1/2 + x/4, y/4)$, $480k + 31 \to \ldots \to 30k + 2$, which shares the exact orbit of $30k + 1$ after one micro-step, +4 halvings)—are untwisted Euclidean copies of the entire 1 mod 30 square, and `1001` ($k \mapsto 16k + 9$, +4 halvings) is an untwisted copy of the 17 mod 30 square.

Sharpened $N_1(X)$ bounds and 2-adic separation. Theorem 18 shows that modulo-30 halving obstructions alone can never push the exponent of $N_1(X)$ below the 2-adic Hausdorff dimension $\beta \simeq 0.626382$, because every surviving 2-adic cylinder contains infinitely many integers reaching 1. To break strictly below β while raising the lower bound by restoring $d = 5$ and $d = 7$ branches, we use the prime-to-2 congruences $2^v o \equiv 1 \pmod{25}$ and $2^v o \equiv 1 \pmod 7$ from Lemma 16. Working modulo $Q_1 = 3 \cdot 5^4 \cdot 7^3$, which equals 643,125, assigns weights to all 128,625 residues in A, B, and C. Knowing o mod Q_1 fixes each reverse move $m = (2^v o - 1)/d$ modulo Q_1/d; writing $w_{\min}(m)$ and $w_{\max}(m)$ for the least and greatest weights among its d lifts modulo Q_1 lets us certify sharper bounds in exact integer arithmetic.

Sharpened lower bound. Set $\gamma' = 399/800 = 0.49875$. The bundled integer vector verifies across all 128,625 residues (2,038,400 truncated moves with $v \leq 64$) that

$$\begin{aligned} \sum_{\text{moves at } o,\ v \leq 64} (d/2^v)^{\gamma'} w_{\min}(m) &\geq w(o), \\ \frac{w(31)}{\max w} &\geq 0.2017. \end{aligned} \tag{6}$$

Let $N(o, X)$ count distinct odd descendants of o in $[1, X]$, including o. For any basin member $o \neq 1$ and $X \geq o$, strong induction on $N(o, X)$ proves $N(o, X) \geq \frac{2}{5}\frac{w(o)}{\max w}(X/o)^{\gamma'}$. Admit children with $2^v o/d \leq X$ (so $m < 2^v o/d \leq X$, yielding disjoint subtrees excluding o); for discarded moves $Xd/(o2^v) < 1$, and valuations are spaced by $4, 20, 3$ for $d = 1, 5, 7$ respectively. Because $\gamma' \geq 2/5$, the discarded sum $\sum (Xd/(o2^v))^{\gamma'}$ stays below $(1 - 2^{-8/5})^{-1} + (1 - 2^{-8})^{-1} < 5/2$ on A, C and $(1 - 2^{-6/5})^{-1} < 5/2$ on B. Multiplying by 2/5 keeps the loss strictly below 1, which the root o absorbs, and (6) closes the

induction. Evaluating at $o = 31$ (with $n = 1$ for $X < 31$) yields $N_1(X) \geq 0.0806\,(X/31)^{399/800}$, and $31^{\gamma'} < 5.544$ gives $N_1(X) \geq 0.0145\,X^{0.49875}$ for all real $X \geq 1$.

A contracting obstruction—and the depth-based upper bound. Why can we not simply reverse (6) to sharpen the upper bound? Remarkably, the move graph admits *contracting double moves*: from a suitable odd multiple o of 15, applying $(d, v) = (7, 1)$ and then $(5, 3)$ produces $o' = (16o-15)/35$, shrinking o by more than a factor of 2.

Proposition 19 (Contracting double-move obstruction). *For every integer modulus $Q \geq 1$, every real exponent $c > 0$, and every residue weight vector $w : \mathbb{Z}/Q\mathbb{Z} \to \mathbb{R}$, there exist positive odd integers $o \equiv 15 \pmod{30}$ and $n \geq 1$ such that n is reachable from o by a chain of reverse moves in $A \cup B \cup C$ and yet $w(o \bmod Q)\,(n/o)^c < 1$.*

Proof. Because $19o' + 15 = 16(19o+15)/35$, solving $19(o/15)+1 \equiv 0 \pmod{35^r}$ chains r successive shrinking double moves $(7, 1)$ followed by $(5, 3)$ inside $A \cup B \cup C$. Choosing r large enough that $w(o \bmod Q)\,(16/35)^{rc} < 1$ with n equal to the final descendant proves the claim. □

Organizing reverse paths by *halving depth* V and *digit product* D sidesteps Proposition 19 completely. Any positive vector $(\min w \geq 1)$ obeying $\sum_{\text{moves at } o} d^s a^v w_{\max}(m) \leq w(o)$ satisfies $\sum_{V=L} D^s \leq w(o)\,a^{-L}$ by induction on L. Two verified 128,625-entry vectors at $(a_H, s_H) = (0.72135, 0)$ and $(a_P, s_P) = (0.7116, 9/25)$ (each testing 3,057,600 moves through $v = 96$ plus a geometric tail $7^s(\max w)a^{97}/(1-a)$) together with a three-state forward vector at $(a_F, s_F) = (0.6331, 7/100)$ yield $N_1(X) \leq C\,X^{0.6245}$:

Set $K = \lceil \log_2 X \rceil$, $\delta = 0.505$, $\beta' = 0.6245$, and $\theta_i = \log_2(1/a_i)$. Follow each odd $n \leq X$ reaching 1 while its cumulative halvings stay below K:

1. If it hits 1 with $V < K$, the depth vector $(a_H, 0)$ gives $H(L) \leq 2{,}953\,(0.72135)^{-L}$, totaling $O(2^{\theta_H K})$ starts.
2. Otherwise, let D be the digit product of the last prefix with $V < K$. If $D \geq 2^{\delta K}$, the forward vector (a_F, s_F) and the modulo-$15 \cdot 2^K$ coding injection yield $O(2^{(\theta_F - s_F\delta)K})$ starts.
3. If $D < 2^{\delta K}$, the crossing step (with digit $d \in \{1, 5, 7\}$ and halving count $v \geq K - V$) enters an odd node $o \leq Y = 7\sqrt{K+1}\,2^{\delta K}$ (multiplying $(1 + 1/(d_i n_i))$ over at most K distinct odd terms ≥ 15 stays below $\sqrt{K+1}$). Using $o \leq \sqrt{K+1}\,XdD/2^{v+V} \leq \sqrt{K+1}\,dD\,2^{K-V-v}$ (since $X \leq 2^K$), raising to $s_P = 9/25 > 0$ gives

$$1 \leq \left(\frac{\sqrt{K+1}\,dD}{o}\right)^{s_P} 2^{-s_P(V+v-K)}.$$

Summing the convergent geometric series over $v \geq K - V$ $(\sum_{j\geq 0} 2^{-s_P j} = (1 - 2^{-s_P})^{-1})$, bounding the depth-$V$ prefix sum by $(\max w)\,a_P^{-V}$ via the pair vector (a_P, s_P), and summing o^{-s_P} over $o \in B_1 \cap [1, Y]$ by summation by parts against $N_1(t) \leq Ct^{\beta'}$ $(t \leq Y)$ contributes $O(C(K+1)^{\beta'/2}\,2^{(\theta_P + \delta(\beta' - s_P))K})$ starts.

Exact integer comparisons verify that θ_H, $\theta_F - s_F\delta$, and $\theta_P + \delta(\beta' - s_P)$ all lie strictly below $\beta' = 0.6245$. For large X, $Y < X - 1$ and the third term drops below $CX^{\beta'}/2$; choosing C large enough to cover the first two terms and small X closes the induction, proving Part 1 of Theorem 8. In pure-kernel Lean 4, `BasinOneSharp.lean`, `BasinOneUpperSharp.lean`, and `BasinSharpAll.lean` discharge the full Part 1 bounds and strict 2-adic/Archimedean separation (`Ouroboros.thm_basin_sharp`) using compact $Q_1 = 643{,}125$ certificate shards tailored for kernel reduction: 57,575 compressed weights $(42{,}875 + 8{,}575 + 6{,}125)$ with $V_{\max} = 25$ for the lower bound, and a three-state depth vector at $a_H = 2/3$ together with 57,575 compressed pair weights and $V_{\max} = 36$ (plus per-class geometric tails) across 21 `QuadTree rfl` shards for the upper bound; `recompute.py` independently checks the wider 104,125-weight $v \leq 64$, $a_H = 0.72135$, and $v \leq 96$ vectors described above.

In contrast with the classical $3x + 1$ map (where the integer basin of 1 is unconditionally dense in $\mathbb{Z}_2$, giving 2-adic dimension 1, while a counting exponent of 1 would follow from the Collatz conjecture), base 10 exhibits a three-way split: 2-adic closure dimension $\beta \simeq 0.626382$, proved Archimedean ceiling ≤ 0.6245, and empirical growth near 0.49945 (Section 5).

4. BACKWARD TOWARD FIVE: WHY THE BLUE CORAL DOMINATES

To prove that the blue coral of Figure 5 unconditionally dwarfs the amber coral, we need a lower bound on $N_5(X)$ with exponent strictly above 0.626383 (and 0.6245). Following Krasikov and Lagarias's difference-inequality framework for $3x + 1$ [6], root an inverse tree at $o = 3$: since $3 \to 10 \to 5$ enters the six-cycle without ever returning to 3, the odd predecessors of 3 form a genuine tree whose members all reach 5.

An odd child of a vertex o under F^{-1} has the form $m = (2^v o - 1)/d$ $(d \in \{1, 3, 5, 7, 9\}, v \geq 1)$, subject to the single congruence $2^v o - 1 \equiv d^2 \pmod{5d}$ enforcing integrality, oddness, and $d(m) = d$ all at once (for instance, $5 \cdot 35 + 1 = 16 \cdot 11$ gives $F(35) = 11$). Restrict to $1 \leq v \leq 16$, put $Q = 14{,}175$, and assign a nonnegative integer weight w_t to each of the $25{,}335 = \sum_{d\in\{1,3,5,7,9\}} Q/d$ residue types $t = (M, r)$ with $M = Q/d$. Because $5d \mid Q$ for all five digits, $R = o \bmod Q$ determines the complete list $\mathcal{M}(R)$ of admissible pairs (d, v) together with each child's type

$$t'(R, d, v) = \Big(\frac{Q}{d},\ \frac{2^v R - 1}{d} \bmod \frac{Q}{d}\Big).$$

Set $s = 13/20 = 0.65$, and test two finite inequalities at every residue $0 \leq R < Q$ (with (7) tested across all five moduli M):

$$\sum_{(d,v)\in\mathcal{M}(R)} (d/2^v)^s\, w_{t'} \geq w_{(M,\,R \bmod M)}, \tag{7}$$

$$\sum_{(d,v)\in\mathcal{M}(R)} w_{t'} \leq L. \tag{8}$$

The first inequality drives exponential branching across all available children; the second caps the weight lost when children exceed the cutoff X. Charging each discarded child its *own* weight—rather than worst-casing all 80 pairs by

$\max w$—raises the resulting prefactor more than 35-fold using the exact same weights.

Lemma 20 (Child-weight charging). *Suppose* (7) *and* (8) *hold with* $L > 0$. *For any odd vertex* o *in the inverse tree of* 3 *and any integer* $X \geq o$, *let* $H(o, X)$ *denote the count of odd* $n \leq X$ *whose forward orbit visits* o. *Then for every type* t *of* o,

$$L H(o, X) \geq (X/o)^s \, w_t.$$

Proof. Set $q = X/o$ and $R = o \bmod Q$. Admit a pair $(d, v) \in \mathcal{M}(R)$ whenever $2^v o/d \leq X$, so its child $m = (2^v o - 1)/d$ satisfies $m < 2^v o/d \leq X$. Since F is a function and o is non-periodic, admitted subtrees are pairwise disjoint, exclude o, and satisfy $H(m, X) < H(o, X)$. Strong induction on $H(o, X)$ gives

$$\begin{aligned} L H(o, X) &\geq L + \sum_{\text{admitted}} L H(m, X) \\ &\geq L + q^s \sum_{\text{admitted}} (d/2^v)^s \, w_{t'}. \end{aligned}$$

Every discarded pair obeys $q\,d/2^v < 1$, so $(q\,d/2^v)^s w_{t'} \leq w_{t'}$. Summing over all discarded pairs loses at most $\sum_{\mathcal{M}(R)} w_{t'} \leq L$, which the leading term L (from the vertex o itself) absorbs completely. Restoring the discarded terms and applying (7) closes the induction, including when no children are admitted. □

Proof. We prove Theorems 6 and 8 (Part 2) together with Corollary 7. The supplied 25,335-entry integer vector has root weight $w_{(Q,3)} = 98{,}959$ and maximum entry 10^6. For each pair (d, v), let $c_{d,v}$ be the largest integer with $c_{d,v}^{20} \, 2^{13v} \leq (10^6)^{20} d^{13}$. Across all 14,175 residues, the 81,072 admissible moves satisfy all $70{,}875 = 5Q$ integer comparisons

$$\sum c_{d,v} \, w_{t'} \geq 10^6 \, w_{(M, R \bmod M)},$$

proving (7). In Lean 4, all 14,175 residues are discharged inside the pure kernel by `rfl` across 14 `QuadTree` shards (`BasinFiveCert00.lean` through `13`) under the uniform budget $80 \cdot 10^6$. This establishes Theorem 6, yielding $N_5(X) \geq (605/10^6)\, X^{13/20}$ for all $X \geq 3$, and combines with Theorem 5 to prove Corollary 7 with explicit rate $O(X^{-0.0236})$ (verified for both odd and all starts in `BasinFiveUncond.lean`, line 581, via `cor_basin_separation_tendsto_zero_unconditional`).

Checking the exact raw row sums in (8) yields maximum $L = 2{,}269{,}527$ at $R = 11{,}365$ (Figure 10). Applying Lemma 20 at $o = 3$ gives, for every integer $X \geq 3$,

$$\begin{aligned} N_5(X) \geq H(3, X) &\geq \frac{98{,}959}{2{,}269{,}527 \cdot 3^{13/20}} \, X^{13/20} \\ &> 0.02134 \, X^{0.65}, \end{aligned}$$

verified by an exact 20th-power integer comparison. Both Lemma 20 (`Ouroboros.lem_weighted_tree3` in `WeightedCharging.lean`) and the complete Part 2 and $N_1(X)/N_5(X) = O(X^{-0.0255})$ bounds of Theorem 8 (`Ouroboros.thm_basin_sharp` in `BasinSharpAll.lean`) are proved inside the Lean 4 kernel. Including even starts multiplies the numerator by at most $\sum_{j \geq 0} 2^{-j\beta'}$ without shrinking the denominator, so the all-start ratio also tends to zero. □

5. EXCURSIONS, CYCLES, AND THE QUADRILLION FRONTIER

Suppose odd positive integers $x_1, \ldots, x_m$ form an odd cycle under F, with last digits $d_i = d(x_i)$ and halving counts $v_i \geq 1$ satisfying $2^{v_i} x_{i+1} = d_i x_i + 1$ (indices cyclic). Put $K = \sum_{i=1}^m v_i$ and $D = \prod_{i=1}^m d_i = 3^a 5^e 7^c$; the ordinary period is $K + m$. Multiplying around the cycle cancels the x_i and leaves

$$\frac{2^K}{D} = \prod_{i=1}^m \left(1 + \frac{1}{d_i x_i}\right), \qquad n_5 = n_3 + n_7, \tag{9}$$

where n_d counts digit d (recalling from Figure 7 that 5 enters precisely after 3 or 7). Hence (a, e, c) fixes four digit counts: $n_3 = e - c$, $n_5 = e$, $n_7 = c$, and $n_9 = (a - e + c)/2$. Furthermore, the modulo-5 recurrence $d_{i+1} \equiv (d_i^2 + 1) 2^{-v_i} \pmod 5$ enforces a minimum halving budget.

Lemma 21 (Sharp halving budget for odd cycles). *For every odd cycle* $(x_1, \ldots, x_m)$ *of* f_{10} *with total halvings* K, *digit counts* n_d, *and* t_{51} *transitions from digit* 5 *to digit* 1,

$$K \geq n_1 + 2n_3 + 4n_7 + 3n_9 + 4\,t_{51}, \tag{10}$$

and consequently $2n_1 + 3a + e + 7c \leq 2K$.

Proof. Assign incoming digit weights $w(1) = 1$, $w(3) = 2$, $w(5) = 0$, $w(7) = 4$, and $w(9) = 3$, together with the potential $\phi(d) = \mathbf{1}_{\{d=5\}}$. Checking all 25 ordered digit pairs (d_i, d_{i+1}) under $d_{i+1} \equiv (d_i^2 + 1) 2^{-v_i} \pmod 5$ confirms

$$v_i \geq w(d_{i+1}) + 4\,\mathbf{1}_{\{(d_i, d_{i+1}) = (5,1)\}} + \phi(d_{i+1}) - \phi(d_i).$$

Summing cyclically telescopes ϕ to zero, yielding (10); doubling and substituting $a = n_3 + 2n_9$, $e = n_5 = n_3 + n_7$, $c = n_7$ yields $2n_1 + 3a + e + 7c \leq 2K$. □

Short cycles with no size hypothesis.

Proposition 22 (Pure-kernel classification of 3-odd-member cycles). *Every odd cycle of* f_{10} *with* $m = 3$ *odd members (primitive or repeated) is the trivial repetition* $x_0 = x_1 = x_2 = 1$ *with* $v_0 = v_1 = v_2 = 1$.

Proof. First note that for $m = 1$, $n_5 = n_3 + n_7$ leaves digit 1 or 9: $2^K = 9 + 1/x$ lies in $(9, 10)$, while $2^K = 1 + 1/x$ forces $x = 1$ (`cycle_m1_unique`). For $m = 2$, the remaining multisets are $\{1, 9\}$, $\{9, 9\}$, $\{5, 7\}$, and $\{3, 5\}$; outside $(1, 1)$, the first three confine 2^K to $(9, 10)$, $(81, 84)$, and $(35, 38)$, while $\{3, 5\}$ forces $K = 4$ and solving $5x + 1 = 2y$, $3y + 1 = 8x$ leaves only the rotations $(5, 13)$ and $(13, 5)$ (`cycle_m2_classification`).

For $m = 3$, any cycle containing 1 must repeat $(1, 1, 1)$, and any cycle containing 5 has odd-period $2 \nmid 3$, impossible. Excluding 1, 5, and 3 (which reaches 5), the least remaining odd integers of digits $1, 3, 5, 7, 9$ are $11, 13, 15, 7, 9$. The identity $n_5 = n_3 + n_7$ admits eight digit multisets:

digits	111	119	135	157	199	359	579	999
D	1	9	15	35	81	135	315	729
U	1	10	17	39	90	142	329	756

where $U = \lfloor \prod (d_i + 1/x_{\min}(d_i)) \rfloor$. Beyond 111, only 135 encloses a power of two ($K = 4$), yet its sole cyclic order is $3 \to 5 \to 1 \to 3$, where Lemma 21 demands $K \geq 1 + 2 + 4 = 7$, impossible. □

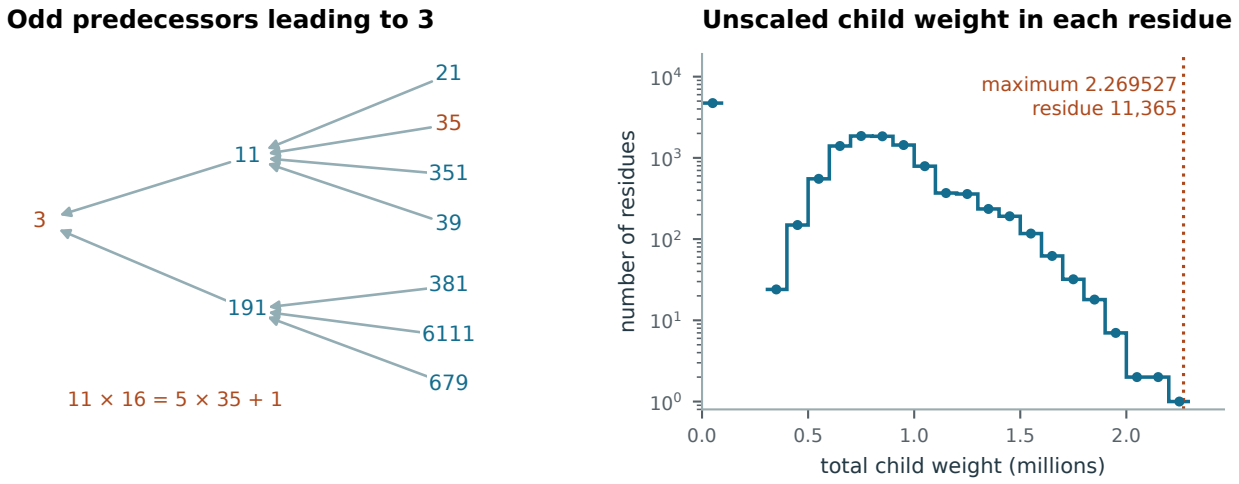


Figure 10. The basin-of-5 certificate. Left: odd predecessors of 3 with $v \le 6$ through two generations (arrows point forward). Right: distribution of the unscaled child-weight sum $\sum_{M(R)} w_{t'}$ across all 14,175 residues modulo Q ($v \le 16$), peaking at $L = 2{,}269{,}527$ at residue $R = 11{,}365$.

For $4 \le m \le 8$, (9) gives the cutoff-free ceiling $2^K = \prod(d_i + 1/x_i) \le 10^m$. Putting $L = \lfloor \log_2 10^m \rfloor$, there are exactly $5\binom{L}{m}$ choices for $(d_1, v_1, \dots, v_m)$ with $\sum v_i \le L$. The modulo-5 recurrence determines $d_2, \dots, d_m$ uniquely, and closing the affine map $(Ax_1 + B)/2^K = x_1$ solves uniquely for $x_1 = B/(2^K - A)$, which is then tested for positivity, integrality, and exact digits and halvings. Exhausting all 7,811,375 choices at $m = 8$ (confirmed independently in Python and C++) produces only repetitions of (1) and (5, 13), proving the first statement of Theorem 9.

Excursions, shared peaks, and the 10^{15} sweep. Direct iteration of the first two million decimal starts (Figure 11) shows a dense glowing wave of blue orbits reaching 5 (median 160 steps; 7,762 starts exceed the 1,085-step display cap, topped by 3,337 steps from 1,761,259) riding cleanly above a thin amber ribbon of fast returns to 1. Comparing peak heights against classical $3x + 1$ through 10^7 (Figure 12) shows why base 10 exhibits far heavier excursions than Collatz. For the five-state tilted transfer matrix M_t of Section 2, the spectral radius crosses unity at the unique positive Markov-modulated Cramér–Lundberg root $\theta_* \simeq 0.2189$ ($\rho(M_{\theta_*}) = 1$, verified in `recompute.py`; see [4] [to check] for Markovian Cramér–Lundberg tail theory, compared with $\theta_* = 1$ for $3x + 1$ where $\frac{1}{2}(1/2)^1 + \frac{1}{2}(3/2)^1 = 1$). The Markov random-walk heuristic thus predicts a heavy power-law tail $C\,x^{-\theta_*}$ for the first-excursion ratio $E(n)/n$ (fitted as $0.2743\,x^{-0.2189}$ in Figure 12B), and orbits coalesce into huge horizontal shared-peak bands—the densest decimal band (peak 7,823,727,640, first reached from 197) collects 252,655 starts, over 36 times the largest shared band in $3x + 1$ (7,000 starts, peak 6,810,136, first reached from 4,255).

To verify far beyond 10^7, an exhaustive C/GMP sweep (sources, chunk logs, coverage receipts, and independent spot-check logs archived in `docs/confirmed-strong-and-novel/last-digit-collatz/sweep/`) tiles $[7, 10^{15})$ with 10,000 contiguous windows of width 10^{11}, covering all 499,999,999,999,997 odd starts (via modular early-descent filters and 1,009,968,528 exact arbitrary-precision continuations, zero fallbacks). The bundled script `recompute.py` validates the archived 10,000-window summary table (`inputs/windows.csv` inside `checker.zip`: gapless window boundaries, exact start counts, `covered+iterated` and `record_exact` flags, zero fallbacks, and exact orbit replay of the global records, or all 20,000 window records under `-r`) without re-running the full 10^{15} C sweep itself. Every odd start in $[7, 10^{15})$ descends below itself; 10^{15} is even and starts below 7 enter $\{1, 5\}$ directly, so strong induction proves that every positive integer $n \le 10^{15}$ reaches $\{1, 5\}$ (and by `cycle_min_ge_of_descent`, any further odd cycle has every odd member $\ge 10^{15}$). The longest recorded first descent starts at 889,676,852,592,557 and runs 11,074 ordinary steps before dipping to 821,329,135,081,529; the widest peak reaches 268 bits, tied by 321,901,022,712,499 and 724,277,301,103,123.

From 10^{15} to a 53,185-step cycle floor. Any further cycle has every odd member $x_i \ge X_0 = 10^{15}$. Taking logarithms in (9) forces a razor-thin linear form in logarithms, paralleling Simons's cycle obstructions for generalized Syracuse maps [9]:

$$0 < \Lambda := K \log 2 - \log D < \frac{1}{X_0} \sum_{i=1}^{m} \frac{1}{d_i} \le \frac{K}{X_0}, \qquad (11)$$
$$D = 3^a 5^e 7^c.$$

Lemma 23 (Four-dimensional lattice-box inclusion). *Suppose an odd cycle $(x_1, \dots, x_m)$ of f_{10} satisfies $x_i \ge X$ for all i and $K \le L$ for integers $X \ge 2$ and $L \ge 1$ with $2L \le X$. For any integer scale $S \ge 1$ and floor approximations $\lambda_p = \lfloor S \log_2 p \rfloor$ ($p \in \{3, 5, 7\}$), the integer combination $F = KS - a\lambda_3 - e\lambda_5 - c\lambda_7$ and exponents (a, e, c) satisfy*

$$1 \le F < \frac{3LS}{2X} + L,$$
$$a\lambda_3 < LS, \qquad e\lambda_5 < LS, \qquad c\lambda_7 < LS.$$

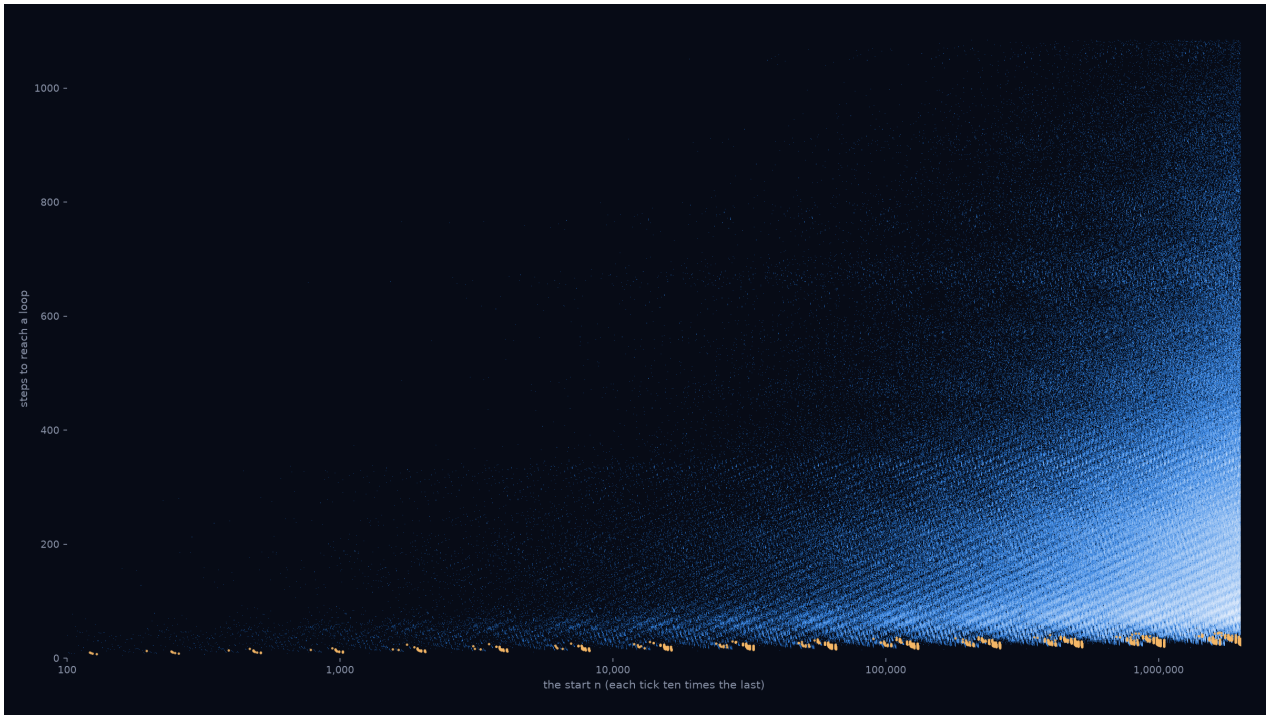


Figure 11. First-hitting times to 1 (amber) or 5 (blue density) across $1 \le n \le 2{,}000{,}000$, displayed from $n = 100$ with vertical axis clipped at 1,085 steps. In the full census, 7,762 starts exceed 1,085 steps, with maximum 3,337 at 1,761,259.

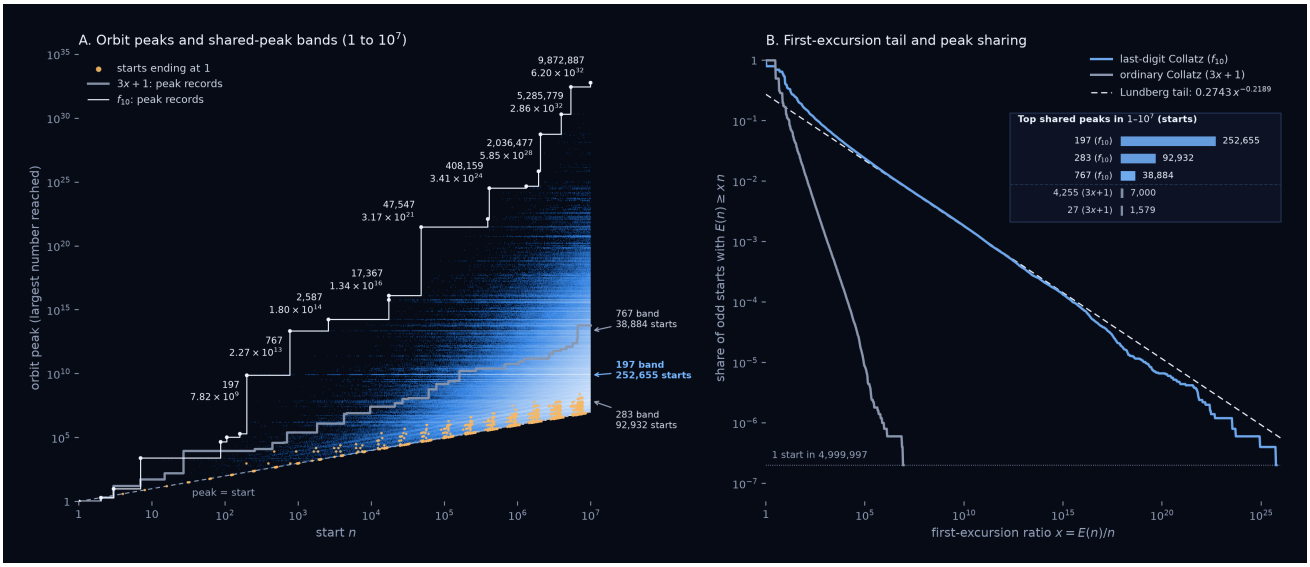


Figure 12. Excursions in f_{10} versus classical $3x+1$ through 10^7. **A:** Orbit peaks and record staircases; horizontal bands collect starts sharing the same maximum (led by the 197 band with 252,655 starts). **B:** Fraction of odd starts whose peak before first descent exceeds xn ($7 \le n \le 10^7$ for f_{10}, $3 \le n \le 10^7$ for $3x+1$), compared with the Cramér–Lundberg tail $0.2743\,x^{-0.2189}$ from the root $\theta_* \simeq 0.2189$ of $\rho(M_{\theta_*}) = 1$; inset compares top shared peaks including the classical orbit of 27.

Proof. Since $D = 3^a 5^e 7^c < 2^K \le 2^L$, each prime factor satisfies $u_p \lambda_p \le u_p S \log_2 p < LS$. Furthermore, $F = S\Lambda/\log 2 + \sum_{p\in\{3,5,7\}} u_p(S\log_2 p - \lambda_p)$; because $2^K > D \ge 2^{a+e+c}$ gives $a+e+c < K \le L$ and $1/\log 2 < 3/2$, the gap bound (11) yields $1 \le F < 3LS/(2X) + L$. □

Theorem 24 (Cycle-length tail above $K = 64{,}000$). *Every odd cycle $(x_1, \dots, x_m)$ of f_{10} whose odd members all satisfy $x_i \ge 10^{15}$ and whose total halving count satisfies $K > 64{,}000$ obeys*

$$m \ge 20{,}190 \qquad \text{and} \qquad K + m \ge 84{,}191.$$

Proof. If $9^m \le 2^{64000}$, then $8^m \le 9^m \le 2^{64000}$ gives $3m \le 64{,}000$, so $m/10^{15} < 1/6 < \log 2$, and (11) forces $2^K < 9^m e^{m/10^{15}} < 2^{64001}$, contradicting $K \ge 64{,}001$. Exact

integer arithmetic via $3^{306} < 2^{485}$ and $2^{1054} < 3^{665}$ confirms $9^{20189} < 2^{64000} < 9^{20190}$, forcing $m \geq 20{,}190$ and $K + m \geq 84{,}191$. □

Applying Lemma 23 at $X = 10^{15}$, $L = 64{,}000$, and $S = 2^{58}$, reducing the lattice generated by $(0, 0, 0, S)$ and $(e_p, -\lambda_p)$ over $\mathbb{Z}^4$, enumerating all box points in exact rational arithmetic, and testing (11) with interval logarithms leaves 298 raw candidates, 44 with nonnegative integer digit counts (n_3, n_5, n_7, n_9), 32 meeting the refined gap (11) with $n_1 \leq K - (n_3 + n_5 + n_7 + n_9)$, and just 19 surviving Lemma 21 (Figure 13). Taking the three separate minima over these 19 survivors yields $K \geq 34{,}905$, $m \geq 15{,}496$, and $K + m \geq 53{,}185$; combining with Theorem 24 completes the proof of Theorem 9.

Empirical scaling of the thin basin. Direct iteration gives $N_1(X) = 11$, 34, 101, 349 at $X = 10^3, 10^4, 10^5, 10^6$ (and 668 versus 299,332 across all starts through 300,000). Enumerating the inverse tree of 1 with every odd iterate capped at 2^{72} (Figure 14) finds 383,455, 12,091,490, and 381,309,367 odd starts below 2^{40}, 2^{50}, and 2^{60}—growing nearly 32-fold per ten doublings. Because an orbit can climb above 2^{72} before landing at 1, every capped count is strictly a *lower bound* on $N_1(X)$: raising the ceiling from 2^{60} to 2^{72} recovers 10,355,985 extra starts below 2^{60} (on top of the 370,953,382 seen under the 2^{60} ceiling; Figure 14D).

Dividing by $\sqrt{X}$ reveals sharp log-periodic ripples across each doubling (Figure 14B), and averaging over 64 bins per octave shows a slight downward tilt below $s = 1/2$ (Figure 14C). A branching model averaging the inverse moves of Lemma 16 uniformly modulo 25 and 7 predicts $\alpha \simeq 0.4994477848\ldots$, the unique root in $(0, 1)$ (with $a = 2^{-\alpha}$) of

$$1 = a + (1 + 35^{\alpha-1})\, a^4. \tag{12}$$

(The 3×3 branching determinant factors off $(1 - a)(1 - a^4)$, and on $[0, 1]$ the negative derivative of $2^{-\alpha}$ exceeds $(\log 2)/2$ in magnitude while the positive derivative of $(35/16)^{\alpha}/35$ stays below $\log(35/16)/16$, guaranteeing a unique crossing.) Dividing the capped counts by $X^{0.4994478}$ flattens the curve across 30 doublings, though our proved window $[0.49875, 0.6245]$ still straddles $1/2$.

■ 6. PULLING BACK ACROSS ALL BASES: WHY BASE TEN IS UNIQUE

We zoomed into base 10 on the visual evidence of Figures 1 and 2 that bases 4 and 10 are the only two nontrivial contracting bases through $b = 20$. Could some larger base—say $b = 22$, 100, or 10^6—harbor a third Collatz twin? Pulling back across all $b \geq 2$ now settles this completely, proving Theorems 10–13 and explaining both the sawtooth reversal at base 10 (Figure 3) and the odd-base density spectrum (Figure 15).

Even bases: generalizing the coding and bounding the least drift. The base-10 coding of Lemma 14 generalizes cleanly to any even base $b \geq 2$. Along any prescribed branch sequence of T_b, $T_b^j(n) = (A_j n + B_j)/2^j$ with $A_0 = 1, B_0 = 0$; an even step leaves (A_j, B_j) unchanged, and an odd step of digit r sends it to $(rA_j,\ rB_j + 2^j)$. Since b is even, A_j stays odd, $B_j \geq 0$, and $A_j \neq 2^j$ for $j \geq 1$. Shifting n by $b\,2^{k-1}$ shifts $T_b^j(n)$ by $A_j b\, 2^{k-1-j}$ (a multiple of b for all $j < k$), so the residue modulo $b\,2^{k-1}$ determines all k branches, and discarding finitely many exceptions $n \leq B_j/(2^j - A_j)$ proves that $p_k(b) = d(D_k(b))$ exists and equals the fraction of classes modulo $b\,2^{k-1}$ maintaining $A_j > 2^j$ for all $1 \leq j \leq k$.

Even bases split cleanly into two algebraic families:

- **Doubly even bases ($4 \mid b$).** Just as in Base 4, every odd digit r obeys $r^2 + 1 \equiv 2 \pmod 4$, so every odd step takes *always one halving* and T_b maps odd integers to odd integers. Lifting modulo $b\,2^{j-1}$ to $b\,2^j$ shifts the j-th image by $A_j b/2 \equiv b/2 \pmod b$, sending an odd digit r with probability $1/2$ each to
$$c = (r^2 + 1)/2 \pmod b, \qquad c + b/2 \pmod b. \tag{13}$$
Even starts fail at $j = 1$, so $p_k(b)$ is half the survival probability of (13) started uniformly on odd digits.
- **Singly even bases ($b = 2M$, M odd).** Just as in Base 10 ($M = 5$), write $y = n \bmod M$ and let $\delta(y) \in [1, 2M)$ be its odd representative. A residue modulo $M2^k = b2^{k-1}$ corresponds bijectively to y_0 and k independent fair parity bits (`lem_even_coding_bij_2M`), evolving by $H(y) = y/2 \pmod M$ (even bit) and $Q(y) = (y^2 + 1)/2 \pmod M$ (odd bit). At successive odd iterates, the intervening halving count $v \geq 1$ has geometric law 2^{-v} (mean $h = 2$, versus $h = 1$ for $4 \mid b$).

On any closed communicating class of the odd-term chain P_b with stationary law π, define the odd-term logarithmic drift $\Delta(\pi) = \sum_r \pi(r) \log r - h \log 2$, and let Δ_b be the minimum of $\Delta(\pi)$ across all closed classes. Why are there only four contracting even bases? Because the halving budget stays capped at $\log 2$ or $2 \log 2$ while the odd digits grow with b.

Lemma 25 (Unbounded even-base drift growth). *For every even base $b \geq 2$, every stationary distribution π of the odd-term Markov chain P_b satisfies*

$$\Delta(\pi) \geq \frac{1}{2} \log(b + 2) - \frac{7}{2} \log 2.$$

Proposition 26 (Near-zero drift band for even bases). *Across all even bases $b \geq 2$:*

$$\exists\, \pi \text{ stationary with } |\Delta(\pi)| \leq \tfrac{1}{4} \iff b \in \{4, 8, 10, 14\},$$
$$\exists\, \pi \text{ stationary with } -\tfrac{1}{4} \leq \Delta(\pi) < 0 \iff b \in \{4, 10\}.$$

Proof of Lemma 25, Proposition 26, and Theorem 10. For $4 \mid b$, the two targets (13) are odd residues in $[1, b)$ differing by $b/2$; their product is at least $1 \cdot (b/2 + 1)$, with equality exclusively on the closed pair $\{1,\ b/2 + 1\}$. Thus $\Delta_b = \frac{1}{2} \log(b/2 + 1) - \log 2 \geq \frac{1}{2} \log(b + 2) - \frac{7}{2} \log 2$, which is nonpositive only at $b = 4$ and lies in $[-1/4, 1/4]$ only at $b \in \{4, 8\}$.

For $b = 2M$ (M odd), fix a starting digit r, put $z = (r^2 + 1) \bmod M$, and set $y_v = z2^{-v} \bmod M \in [0, M)$. If $z = 0$, then $\delta(y_v) = M$ for all v, giving row mean $\log M$. If y_1 is even and positive, $\delta(y_1) = y_1 + M \geq M + 2$, so $v = 1$ alone contributes $\frac{1}{2} \log(M + 2)$. If y_1 is odd, then for each $v \geq 2$ the even integer $2^{v-1} y_v$ satisfies $2^{v-1} y_v = y_1 + tM$ with $t \geq 1$ ($t \leq 0$ is negative or odd), so $\delta(y_v) \geq (M + 1)/2^{v-1}$; using

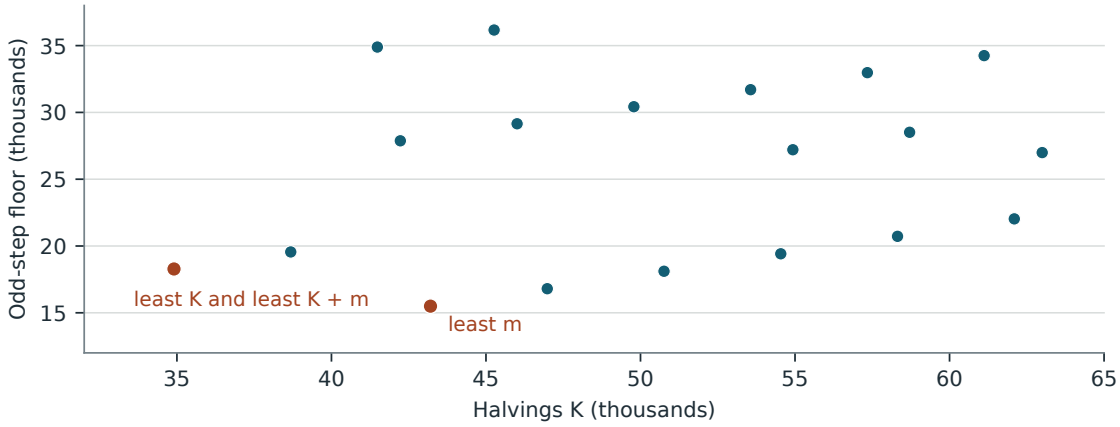


Figure 13. All 19 surviving exponent triples for $K \le 64{,}000$ plotted by halving count K and required odd-step floor m. Two distinct triples realize the three separate minima: $(8{,}842, 7{,}001, 1{,}651)$ gives $K = 34{,}905$, $m \ge 18{,}280$, $K + m \ge 53{,}185$, while $(13{,}041, 4{,}567, 4{,}250)$ gives $K = 43{,}205$, $m \ge 15{,}496$.

$\sum_{v\ge 2}(v-1)2^{-v} = 1$, the row average is at least $\frac{1}{2}\log(M+1) - \log 2$. Averaging in stationarity and subtracting $2\log 2$ gives $\Delta(\pi) \ge \frac{1}{2}\log((M+1)/64) = \frac{1}{2}\log(b+2) - \frac{7}{2}\log 2$, proving Lemma 25, which is positive for $M \ge 65$ and exceeds $1/4$ for $M \ge 105$.

For the remaining odd $3 \le M \le 103$, form the exact odd-term matrix P by summing 2^{-v} over periods of $2 \bmod M$, and set $g(r) = \sum_s P(r,s)\log s - 2\log 2$. Every stationary law obeys $\Delta(\pi) \ge \min_r \ell^{-1}\sum_{i=0}^{\ell-1}(P^i g)(r)$, which is strictly positive at $\ell = 8$ for $M = 7$ and exceeds $1/4$ for all 48 odd $9 \le M \le 103$ (checked via pure-kernel `rfl` in `EvenBasesCert.lean`). Exact stationary laws for $b \in \{2,4,6,8,10,14\}$ confirm Theorem 10 and Proposition 26 (Table 1 and Figure 3). □

Table 1. Odd-term drifts Δ_b (rounded to six places) and exact horizon-12 non-descent densities $p_{12}(b)$. Bases 4 and 10 sit almost side by side just below zero, while base 14 sits barely above zero.

b	Δ_b	$p_{12}(b)$
2	−1.386294	0
4	−0.143841	113/4,096
6	−0.849815	7/6,144
8	+0.111572	1,407/8,192
10	−0.134161	495/4,096
14	+0.067947	5,669/28,672

Proof of Theorem 11. Write $S_j = \log A_j - j\log 2$ in the finite-state shortcut coding. For $b = 2M$ (M odd), the shortcut kernel $K = \frac{1}{2}(H+Q)$ and odd-term kernel $P = \frac{1}{2}Q(I - \frac{1}{2}H)^{-1}$ satisfy $\pi P = \pi \iff \pi K = \pi$ (proved in `EvenBases.lean`), so they share the same extreme stationary laws and the shortcut drift per step is $\Delta(\pi)/2$ (for $4 \mid b$, it is $\Delta(\pi)$ directly). By the finite-state ergodic theorem, S_j/j converges almost surely to the drift of the eventual closed class. For $b \in \{2,4,6,10\}$, finite-graph decomposition confirms that each chain P_b has a unique closed class (`classes=1` in `recompute.out`), whose unique stationary law has negative drift by Theorem 10; hence $S_j \to -\infty$ almost surely, so $p_k(b) \to 0$ and integers avoiding descent have natural density zero. For every other even base, $S_j \to +\infty$ almost surely; each path has a finite last minimum, so some state of positive initial mass has positive probability $\eta_b > 0$ of keeping $S_j > 0$ for all $j \ge 1$, whence $p_\infty(b) \ge \eta_b > 0$ by continuity of probability. The complete dichotomy of Theorem 11 is formalized in `EvenStopping.lean` as `Ouroboros.thm_even_stopping`. □

Tilting the shortcut matrix $(M_t^{\pm}w)(z) = \mathbb{E}_z[(m/2)^{\pm t}w(z')]$ quantifies both sides via Markov's inequality ($M_t^+ w \le \lambda w$) and optional stopping of the supermartingale $(2^j/A_j)^t w(z_j)$ ($M_t^- w \le w$), yielding certified bounds such as

$$p_k(6) \le 4.595731\,(0.702852)^k,$$
$$p_\infty(8) \ge 0.0861, \qquad p_\infty(14) \ge 0.0298.$$

Moreover, for every $b = 4k$ with $k \ge 4$ and $k \ne 6$, the single explicit vector $t = 1/2$, $w(1) = 1$, $w(r) = 2/5$ ($r \ge 3$) satisfies $M_{1/2}^- w \le w$ universally (`prop_stopping_certs_4k_uniform`), giving

$$p_\infty(b) \ge \frac{1}{2b}\sum_{\substack{3\le r<b \\ r \text{ odd}}}(1 - \sqrt{2/r}).$$

Notice the subtle limit-order caveat: $p_k(b) \ge p_\infty(b) > 0$ holds at every *finite* horizon k, whereas proving that an individual integer diverges to infinity requires ruling out an empty infinite intersection (like $\bigcap_k \{n : n \ge k\} = \emptyset$). In odd bases, however, exact divergence proofs succeed systematically.

Odd bases: zero-digit trapdoors and certified escapes. Drop $n = 33$ into two odd bases. In base 5, $33 \to 100 \to 50 \to 25$, where the zero digit sends $f_5(25) = 0\cdot 25 + 1 = 1$. In base 17, ten steps return 33 to the exact same residue modulo 34 at 370,871: every $n \equiv 33 \pmod{34}$ obeys

$$f_{17}^{10}(n) = 11{,}200\,n + 1{,}271 \equiv 33 \pmod{34}, \qquad (14)$$

so every integer in $34\mathbb{Z}_{\ge 0} + 33$ diverges to infinity.

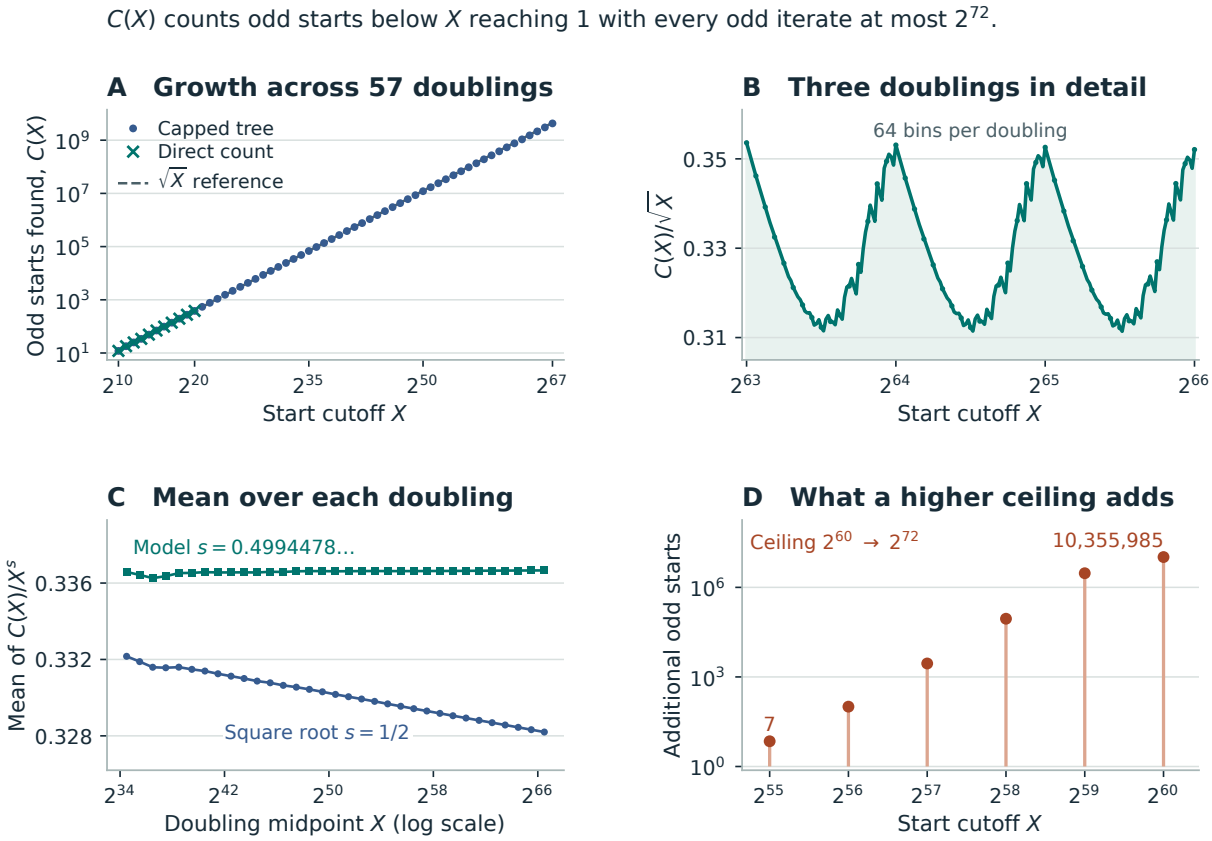


Figure 14. Odd starts below X reaching 1 with every odd iterate at most 2^{72}. **A:** Exact agreement with direct forward iteration through 2^{20}; above that, the curve is a lower bound on $N_1(X)$. **B:** Dividing by $\sqrt{X}$ exposes log-periodic ripples within each doubling. **C:** Geometric means over 64 bins per octave drift slightly below $\sqrt{X}$ while tracking the branching-model exponent $\alpha \simeq 0.4994478$. **D:** Additional starts below X recovered by raising the odd-iterate ceiling from 2^{60} to 2^{72}.

Lemma 27 (Certified base-17 escaping progression). *For every odd positive integer $n \equiv 16 \pmod{17}$ (equivalently $n \equiv 33 \pmod{34}$), the orbit $f_{17}^i(n)$ never enters $17\mathbb{N}$ or $\{1\}$, satisfies $f_{17}^i(n) < f_{17}^{i+10}(n)$ for every $i \geq 0$, and tends to $+\infty$ as $i \to \infty$.*

The following nine lines check (14) symbolically for all $q \geq 0$ at once (a zero in the tuple prescribes a halving):

Symbolic Base-17 Escape Check (9 lines, verified by `recompute.py`)

```
a, c = 34, 33
for d in (16, 2, 5, 0, 0, 0, 8, 14, 10):
    if d:
        assert a % 34 == 0 and c % 2 == 1 and c % 17 == d
        a, c = d*a, d*c + 1
    else:
        assert a % 2 == c % 2 == 0
        a, c = a//2, c//2
assert (a, c) == (34*11200, 33*11200 + 1271)
```

Yet base 17 also hides a period-8 cycle through 9 alongside both its trapdoor and this escaping progression.

For any odd $b \geq 3$, every multiple of b halves to an odd multiple of b and jumps straight to 1. Since halving preserves divisibility by odd b, a nonmultiple enters $b\mathbb{Z}_{\geq 1}$ only from an odd digit r satisfying $r^2 \equiv -1 \pmod{b}$.

Proof of Theorem 12. Since -1 is a quadratic nonresidue modulo any prime $p \equiv 3 \pmod{4}$, no integer satisfies $x^2 + 1 \equiv 0 \pmod{b}$. Neither halving nor an odd step $d_b(n)n + 1$ can therefore send a nonmultiple of b into $b\mathbb{Z}_{\geq 1}$, leaving $Z_b = b\mathbb{Z}_{\geq 1}$ exactly. □

Lemma 28 (Disjointness of odd-base trapping and divergence). *For every odd base $b \geq 3$, every positive integer reaching $b\mathbb{Z}_{\geq 1}$ reaches 1 ($Z_b \subseteq R_b$), and the diverging set E_b is disjoint from R_b ($E_b \cap R_b = \emptyset$).*

If instead every prime divisor of b is $1 \pmod{4}$ (a *rooted* base), Hensel lifting and the Chinese remainder theorem yield $2^{\omega(b)}$ roots (`thm_odd_trapdoor_rooted`), giving one-step capture density $1/b + 2^{\omega(b)}/(2b)$. In base 5, every odd $n > 1$ either hits $5\mathbb{N}$ in 0, 1, or 2 steps (digits $0, 2, 3, 4$) or drops to $(n+1)/2 < n$ (digit 1), proving by induction that *every positive integer reaches 1 under f_5* (`prop_base5_reaches_one`).

Because odd b is coprime to 2, residue modulo b and low binary bits live in independent coordinates. Define the cylinder

$$C(x, c, j) = \{n \geq 1 : n \equiv x \pmod{b},\ n \equiv c \pmod{2^j}\},$$
$$(0 \leq x < b, \quad 0 \leq c < 2^j),$$

of density $1/(b\,2^j)$. At $j = 0$, revealing one parity bit splits the cylinder in half. For $j \geq 1$ and $x \neq 0$, the next move requires *no* additional bit, and an even digit x actually creates $v_2(x)$ fresh known bits:

Parity of c	Next mod b	Next low bits	Bits known
even	$x/2 \bmod b$	$c/2$	$j-1$
odd	$x^2+1 \bmod b$	$xc+1 \bmod 2^{j+v_2(x)}$	$j+v_2(x)$

Because specifying e output bits specifies exactly e input bits, any path using s parity splits from x pulls back to a single starting cylinder of density $1/(b2^s)$. Follow a *deterministic stretch* from $(x,1,1)$ while $j \geq 1$ and $x \neq 0$. If it enters a self-returning cylinder (in particular, a subset of an earlier cylinder in the stretch, formalized as `IsCertifiedEscape` in `OddSurvey.lean`) after h halvings and odd multiplications by $d_1,\dots,d_\ell$, then $h \leq \sum v_2(d_i)$ and $\ell \geq 1$, so the return map $(An+B)/2^h$ has $A=\prod d_i \geq 2^h$ and $B>0$: every integer in that cylinder diverges to infinity (`lem_odd_runaway`, instantiated in Lemma 27 where $h=4$).

Proof of Lemma 28 and Theorem 13. Every multiple of odd b halves to an odd multiple of b and jumps in one step to 1, giving $Z_b \subseteq R_b$; any orbit reaching 1 enters the bounded cycle $1 \leftrightarrow 2$ and cannot tend to $+\infty$, giving $E_b \cap R_b = \emptyset$.

Reduce to the b zero-bit states $x \in \mathbb{Z}/b\mathbb{Z}$, giving $x=0$ its initial share $1/b$ as an absorbing trap. From each nonzero x, move with probability $1/2$ to $x/2 \bmod b$ and with probability $1/2$ along the deterministic stretch from $(x,1,1)$, absorbing into trap (0), certified escape, another zero-bit state, or no absorbed mass if unresolved. Let τ_b, ρ_b be the finite-path absorption probabilities into trap and escape. Every finite union of disjoint first-hit cylinders has natural density equal to its path probability; taking suprema gives $\underline{d}(Z_b) \geq \tau_b$ and $\underline{d}(E_b) \geq \rho_b$ without commuting limits. When $\tau_b+\rho_b=1$, the inclusions of Lemma 28 squeeze $\overline{d}(R_b) \leq 1-\rho_b = \tau_b \leq \underline{d}(Z_b)$ (formalized in Lean 4 as `thm_odd_densities_squeeze`), establishing $d(Z_b)=d(R_b)=\tau_b$ and $d(E_b)=\rho_b$ as the exact rational solutions of the finite absorption system; the full 905-of-999 classification through $b=1{,}999$ is discharged in `OddSurvey.lean` as `Ouroboros.thm_odd_survey`. □

Surveying all 999 odd bases $3 \leq b \leq 1{,}999$ with a 2,000-step stretch budget (6,000 at $b=1{,}983$, where the longest resolved stretch takes 2,640 steps) establishes $\tau_b+\rho_b=1$ in 905 bases (Table 2 and Figure 15, proved in Lean 4 by `Ouroboros.thm_odd_survey`): all 235 rooted bases (27 trap-only, 208 split) and 670 of 764 rootless bases (with the 94-base residual breakdown and the printed $b=17,19$ rational fractions of Table 2 verified in `recompute.py`, while $b=3,5,7,13,25$ are also proved directly in `OddSurvey.lean` via `thm_odd_table_examples` and `trapOnlyBases`).

7. EVERY ENDING IN A SET OF DENSITY ZERO

Section 6 held the starting integers fixed and varied the base of the iteration rule. To close the mathematical narrative, hold the *rule* fixed in base 10 and vary only the base in which we write the starting integers.

The two numbers 131,071 and 431,071 share their last five decimal digits, and both leave remainder 1 on division by 30. Yet $131{,}071+1=2^{17}$, so the first drops straight to 1 by seventeen halvings, whereas the second follows odd terms $431{,}071 \to 13{,}471 \to 421 \to 211 \to 53 \to 5$. How much can trailing digits ever reveal about which loop captures a start?

Recall $\mathcal{B}_i = \{n \in \mathbb{Z}_{\geq 1} : f_{10}^j(n) = i \text{ for some } j \geq 0\}$ for $i \in \{1,5\}$. By Theorem 18 (and `thm_basin1_density_zero`), $\mathcal{B}_1$ has natural density zero. Yet if we write starting integers in base 7 and prescribe *any* finite string of trailing base-7 digits—even a thousand zeros—infinitely many *odd* integers with that ending reach 1, and infinitely many reach 5.

Theorem 29 (Completions of every congruence class). *Every congruence class $a \bmod m$ $(m \geq 1)$ contains infinitely many members of $\mathcal{B}_5$. For $\mathcal{B}_1$, put $h=\max(0,v_2(m)-1)$, $M=30\cdot 2^h$, and $p=\gcd(m,M)$, and recall the shortcut map $T=T_{10}$ (extended to $x=0$ via the even branch $T(0)=0$). Then $a \bmod m$ contains infinitely many members of $\mathcal{B}_1$ if and only if some $0 \leq x < M$ satisfies*

$$x \equiv a \pmod{p},$$

$$T^h(x) \bmod 30 \in S = \{0,1,2,4,8,15,16,17\}.$$

If no such x exists, the congruence class contains no member of $\mathcal{B}_1$.

Proof. From any residue r mod 30, the two one-halving successors of $r+30s$ for $s \in \{0,1\}$ are $T(r)$ and $T(r)+15 \pmod{30}$. Backward breadth-first search on $\mathbb{Z}/30\mathbb{Z}$ classifies the residues reaching 1 mod 30, and those reaching 1, 11, or 21 (mod 30) (last decimal digit 1), into exact distance layers:

Dist.	Target 1 mod 30	Target 1, 11, 21 mod 30
0	1	1, 11, 21
1	2	2, 9, 12, 19, 22, 29
2	4	4, 8, 14, 18, 24, 28
3	8	5, 6, 15, 16, 25, 26
4	15, 16	0, 3, 7, 10, 13, 17, 20, 23, 27
5	0, 17	

Thus $S = \{0,1,2,4,8,15,16,17\}$ consists precisely of all residues modulo 30 that can reach 1 mod 30 (so S is closed backward: no residue outside S can ever enter S), while *every* residue modulo 30 reaches a residue ending in 1 within at most four halvings.

By the Chinese remainder theorem, each $x \bmod M$ compatible with $a \bmod p$ refines $a \bmod m$ into arithmetic progressions of step $\operatorname{lcm}(M,m) = 30\cdot 2^h D$ (D odd). Because the first h shortcut branches are fixed along $x+30\cdot 2^h Dj$, each step halves the spacing and (on an odd branch) multiplies by the odd digit in $\{1,3,5,7,9\}$, sending the progression onto a full tail $c+30D'j$ ($j \geq 0$, D' odd) in residue $T^h(x) \bmod 30$. Splitting j by parity and applying T realizes either graph successor as a whole progression tail of the same form.

To reach 1, route any live state in S to 1 mod 30 and pick seed $u=1$. To reach 5, route any state in $\mathbb{Z}/30\mathbb{Z}$ to 1, 11, or 21 (mod 30) and pick the corresponding odd seed 211 ($211 \to 53 \to 5$), 11 ($11 \to 3 \to 5$), or 21 ($21 \to 11 \to 3 \to 5$). For the resulting progression tail $c+30Ej$ (E odd), apply the commutator translation $Q \circ P^{s-1}$ of (5) modulo the odd integer $q=15E$. Because the chosen seed $u \in \{1,11,21,211\}$ agrees with c mod 30, repeated translation hits c mod $15E$, and oddness upgrades this to c mod $30E$. Iterating P^s yields arbitrarily large representatives $\geq c$ that lift back to positive integers in a mod m reaching the desired loop. Necessity for $\mathcal{B}_1$ follows from $T(1)=1$ and the backward closure of

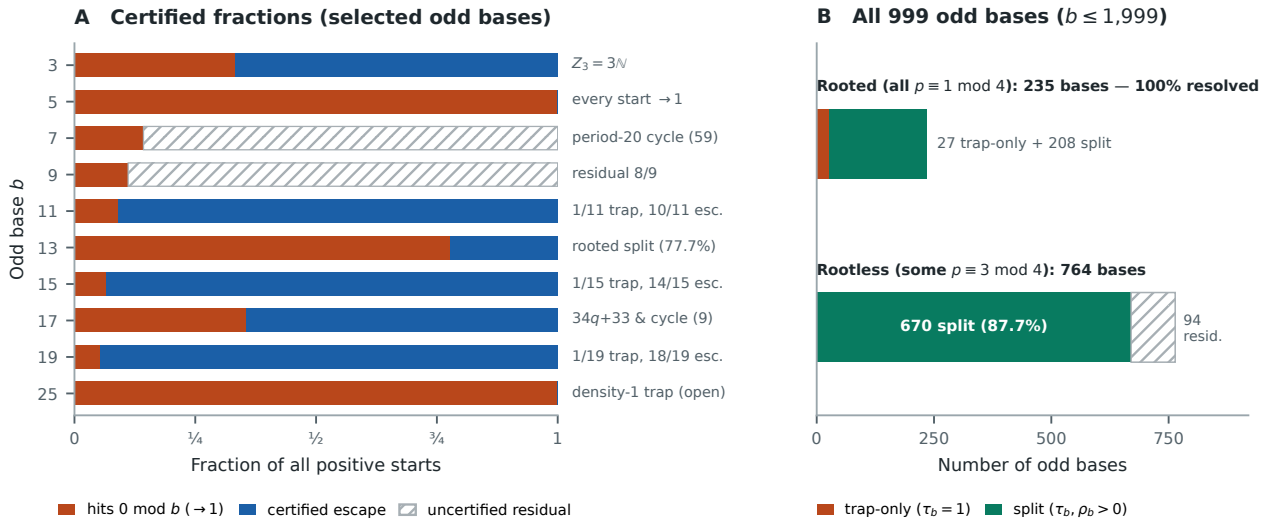


Figure 15. Certified odd-base natural densities. **A:** Selected odd bases from 3 through 25 (amber: zero-digit trapdoor into 1; blue: certified cylinder escape; hatched: uncertified residual mass, such as 6/7 at base 7 and 8/9 at base 9). In base 17, trapping and divergence already sum to 1 despite a period-8 cycle through 9. **B:** Complete survey of all 999 odd bases $3 \le b \le 1{,}999$: all 235 rooted bases resolve completely (27 trap-only, 208 split), while 670 of 764 rootless bases split with full absorption and 94 retain residual mass.

Table 2. Exact absorption probabilities for selected odd bases. Whenever $\tau_b + \rho_b = 1$, Theorem 13 promotes these fractions to exact natural densities (rows $b = 3, 5, 7, 13, 25$ are directly evaluated in `OddSurvey.lean`; the explicit $b = 17, 19$ fractions are verified in `recompute.py`).

b	τ_b	ρ_b	Behavior beyond the cylinder fractions
3	1/3	2/3	$Z_3 = 3\mathbb{Z}_{\ge 1}$ exactly
5	1	0	Every positive integer reaches 1
7	1/7	0	Residual 6/7; period-20 cycle through 59
13	13,210/16,991	3,781/16,991	Complete absorption ($\tau_{13} + \rho_{13} = 1$)
17	148,789/417,911	269,122/417,911	Complete absorption, plus period-8 cycle through 9
19	1/19	18/19	Stretch enters earlier escape cylinder
25	1	0	Density-one trapping (universal capture open)

S (formalized in `SuffixClass.lean` as `Ouroboros.thm_suffix_class`). □

Notice what Theorem 29 rules out: no modular test modulo any $m \ge 1$ can ever certify membership in $\mathcal{B}_1$ because every class meets $\mathcal{B}_5$ infinitely often, and since $p = 2^{v_2(m)} \gcd(m, 15)$, odd primes beyond 3 and 5 create no obstruction at all. If m is odd and $15 \nmid m$, then both basins appear infinitely often in every class modulo m; even requiring odd starts ($m = 2 \cdot 7^k$) gives $h = 0$ and $p = 2$, so $x = 1$ still succeeds for every base-7 ending. By contrast, when $15 \mid m$ is odd, the classes meeting $\mathcal{B}_1$ are precisely $\{0, 1, 2, 4, 8\} \pmod{15}$: neither modulus 3 nor modulus 5 alone excludes any class, yet together they leave only five of fifteen.

To measure how much room the allowable endings leave in an arbitrary written base $b \ge 2$, write each $n \in \mathcal{B}_1$ in base b as $n = \sum_{j \ge 0} d_j b^j$ (padded on the left with zeros), map it to the reversed expansion $\sum_{j \ge 0} d_j b^{-j-1} \in [0, 1]$, and take the closure $\mathcal{E}_b \subset [0, 1]$. Thus a base-$b$ prefix interval of length b^{-k} prescribes the last k base-b digits of n, and $\mathcal{E}_b$ represents all infinite trailing digit strings whose every finite ending occurs in $\mathcal{B}_1$.

Corollary 30 (Exact Hausdorff and box dimension in every written base). *Write an integer base $b \ge 2$ as $b = 2^t u$ with u odd, and let $\lambda = 1/a \simeq 1.543689$ be the unique positive root of $\lambda^4 = \lambda^3 + 2$. Then the Hausdorff and box-counting dimensions of $\mathcal{E}_b \subset [0, 1]$ both equal*

$$\dim_H(\mathcal{E}_b) = \dim_B(\mathcal{E}_b) = \frac{\log u + t \log \lambda}{\log b}.$$

To eight decimal places, $\dim \mathcal{E}_2 = 0.62638214$ *(recovering β from Theorem 18 and strictly exceeding the integer ceiling 0.6245 of Theorem 8),* $\dim \mathcal{E}_{10} = 0.88752982$, *and* $\dim \mathcal{E}_7 = 1$ *with $\mathcal{E}_7 = [0, 1]$. The analogous closure for $\mathcal{B}_5$ equals $[0, 1]$ in every base $b \ge 2$.*

Proof. Under the four-state collapse $S_0 = \{0, 1, 2, 17\}$, $S_1 = \{15, 16\}$, $S_2 = \{8\}$, and $S_3 = \{4\}$ of Section 3, Theorem 18 shows that the number $W_r(h)$ of live length-h halving paths starting from any $r \in S$ satisfies $c_1 \lambda^h \le W_r(h) \le c_2 \lambda^h$ uniformly for constants $0 < c_1 < c_2$ (with $W_1(h) = W_1(h-1) + 2W_1(h-4)$ for $h \ge 4$).

At digit depth $k \ge 1$, refine each occupied suffix modulo b^k to modulus $\ell_k = \operatorname{lcm}(30, b^k) = 30 \cdot 2^{h_k} D_k$, where $h_k = \max(0, tk - 1)$ and $D_k = u^k / \gcd(u^k, 15)$. Since $\gcd(u^k, 15) = \gcd(u, 15)$ is constant for all $k \ge 1$, we have

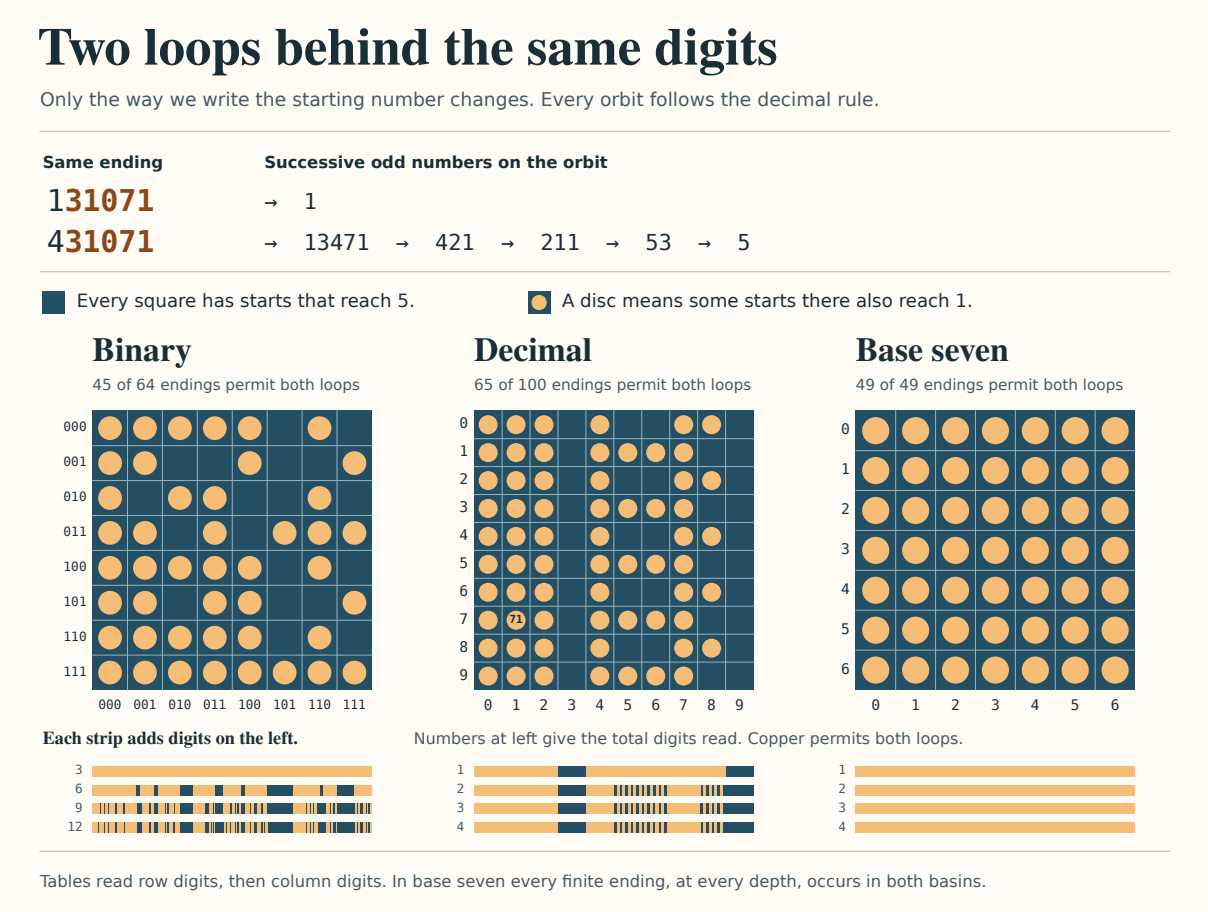


Figure 16. Exact tables of trailing endings for the base-10 map f_{10} written in three different bases: binary (six trailing bits, 45 of 64 endings admit both loops), decimal (two trailing digits, 65 of 100 admit both; 71 admits both while 73 admits only 5), and base 7 (two trailing base-7 digits, all 49 of 49 admit both). Bottom strips plot the exact b-adic occupancy of $\mathcal{B}_1$ through increasing suffix depths, closing to dimensions 0.62638214, 0.88752982, and 1, respectively.

$D_{k+j}/D_k = u^j$ and $h_{k+j} - h_k = tj$ (for $t \geq 1$, or 0 for $t = 0$). From any live refined class modulo ℓ_k ending at state $r \in S$ after h_k halvings, the number of live refined extensions j digits later is exactly $(D_{k+j}/D_k)\, W_r(h_{k+j} - h_k) = u^j W_r(tj)$, because the odd-modulus factors choose freely among u^j lifts while the low-bit paths count live halvings. Since each occupied suffix modulo b^k has between 1 and 30 refined lifts modulo ℓ_k, the number of occupied depth-$(k+j)$ extensions of *every* occupied depth-k suffix lies between fixed positive constants times θ^j, where $\theta = u\lambda^t > 1$.

In particular, there are $\asymp \theta^k$ occupied level-k intervals of length b^{-k}, so $\dim_B(\mathcal{E}_b) = \log_b \theta = (\log u + t \log \lambda)/\log b$ and $\dim_H(\mathcal{E}_b) \leq \log_b \theta$. For the Hausdorff lower bound, place uniform mass on the occupied words at depth K and extract a weak-$*$ limit μ on the compact space $\{0, \dots, b-1\}^{\mathbb{N}}$ as $K \to \infty$. By the uniform extension bounds, every length-k cylinder carries μ-mass $O(\theta^{-k})$, so every singleton has μ-mass 0. Pushing μ forward to $\mathcal{E}_b \subset [0,1]$ via base-b expansion, each interval $I \subset [0,1]$ of length $b^{-(k+1)} \leq |I| < b^{-k}$ meets at most three level-k b-adic intervals and therefore carries mass $O(\theta^{-k}) = O(|I|^{\log_b \theta})$. The mass distribution principle yields $\dim_H(\mathcal{E}_b) \geq \log_b \theta$. Finally, Theorem 29 makes every depth-k ending occupied for $\mathcal{B}_1$ in base 7 (since $\gcd(7^k, 30) = 1$) and for $\mathcal{B}_5$ in every base $b \geq 2$, yielding $[0,1]$ identically (formalized in `SuffixDimension.lean` as `Ouroboros.cor_suffix_dimension`). □

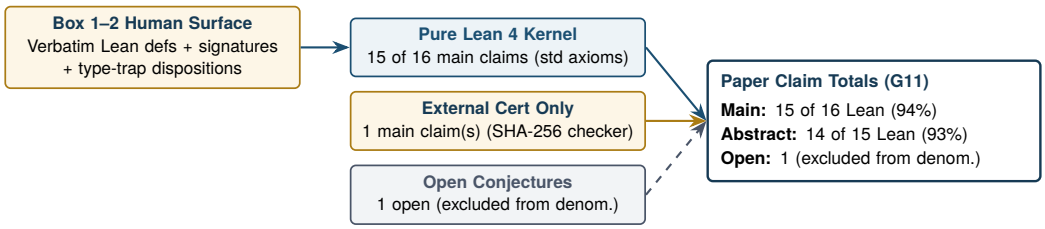
Concretely, in base 7 the ending 314 occurs on both odd starts $(256314)_7 = 47{,}835 \to 1$ and $(1314)_7 = 501 \to 5$; prepend another digit on the left, or a thousand more, and infinitely many odd starts in each basin still carry that exact ending—even though inside every such progression $\mathcal{B}_1$ has relative natural density zero ($O(7^k X^{-0.3755}) \to 0$). Membership in $\mathcal{B}_1$ versus $\mathcal{B}_5$ is therefore a genuinely global property of the decimal dynamics, invisible to any finite 7-adic suffix and one-sidedly invisible to all modular arithmetic.

8. FORMAL VERIFICATION & READER'S TRUST BOUNDARY

What You Must Trust to Accept This Paper:

1. **Lean 4 Kernel & Toolchain:** leanprover/lean4:v4.35.0; committed axiom receipt anchored to git commit 797e3413d11d (independent gate-box kernel replay pending); standard axioms only (propext, Classical.choice, Quot.sound).
2. **Core Definitions & Fidelity (Boxes 1–2):** Verify that the verbatim Lean definitions in Box 1 match Sections 1, 5, and 6 and that each Lean signature in Box 2 matches its stated theorem.
3. **External Certificates & Written Proofs:** 1 SHA-256-pinned external artifact(s) plus any claims marked Written or Open in Table 4 below.

Reader's Trust-Flow Architecture (Distinct Evidence Paths to Claim Totals)



Anti-Shortcut Audit Strip — Source-Level Approx (Gate Replay Pending)

WARN: Committed receipt (gate replay pending) | PASS: sorry/admit: 0 (per committed receipt)
PASS: native_decide: 0 (per committed receipt) | PASS: Custom axioms & meta: 0 (per committed receipt)
PASS: Coverage: 1833/1833 public theorems (source-level approx) | WARN: Fidelity: 28/28 exact (draft cross-family)
WARN: Witnesses: pending gate replay | WARN: Human referee: pending

Box 1 — Verbatim Lean 4 Custom Definitions (Human Audit Surface)

Ouroboros.B1Padic (Ouroboros/CantorClosure.lean:969-969) — Thm. 18 & Thm. 8 (odd basin B_1 embedded in Z_2)

```
def B1Padic : Set ℤ_[2] := (fun n : ℕ => (n : ℤ_[2])) '' B1Set
```

Ouroboros.D_k (Ouroboros/EvenStopping.lean:33-34) — Sec. 1.3 & Thm. 11 (k-step even-base non-descent set D_k(b))

```
def D_k (b k : ℕ) : Set ℕ :=
  {n | 1 ≤ n ∧ ∀ j, 1 ≤ j → j ≤ k → n ≤ (T_b b)^[j] n}
```

Ouroboros.DescentSet (Ouroboros/EvenStopping.lean:37-38) — Thm. 11 (integers eventually descending under f_b)

```
def DescentSet (b : ℕ) : Set ℕ :=
  {n | 1 ≤ n ∧ ∃ m : ℕ, 1 ≤ m ∧ (f_b b)^[m] n < n}
```

Ouroboros.DivSet (Ouroboros/OddBases.lean:1347-1348) — Lem. 28 (odd-base diverging set E_b)

```
def DivSet (b : ℕ) : Set ℕ :=
  {n | 1 ≤ n ∧ Filter.Tendsto (fun i => (f_b b)^[i] n) Filter.atTop Filter.atTop}
```

Ouroboros.E_k (Ouroboros/DensityDescent.lean:576-577) — Sec. 1.3 & Thm. 3 (non-descent set D_k(10) \ {1} for n >= 2)

```
def E_k (k : ℕ) : Set ℕ :=
  {n | 2 ≤ n ∧ ∀ j, 1 ≤ j → j ≤ k → n ≤ T^[j] n}
```

Ouroboros.HasNaturalDensity (Ouroboros/EvenStopping.lean:45-46) — Sec. 1.3 & Thm. 11 (natural density limit predicate)

```
def HasNaturalDensity (S : Set ℕ) (d : ℝ) : Prop :=
  Filter.Tendsto (fun N : ℕ => (countUpTo S N : ℝ) / (N : ℝ)) Filter.atTop (nhds d)
```

Ouroboros.HasStationaryDriftIn (Ouroboros/EvenBases.lean:1282-1288) — Thm. 10 & Prop. 26 (stationary drift predicate for even bases)

```
def HasStationaryDriftIn (b : ℕ) (P : ℝ → Prop) : Prop :=
  if 4 ∣ b then
    ∃ (hk : 1 ≤ b / 4) (π : Fin (2 * (b / 4)) → ℝ),
      IsStationary4k (b / 4) hk π ∧ P (drift4k π)
  else
    ∃ (hM : 1 ≤ b / 2) (π : Fin (b / 2) → ℝ),
      IsStationary2M (b / 2) hM π ∧ P (drift2M (b / 2) π)
```

Ouroboros.InBasin1 (Ouroboros/Defs.lean:43-43) — Sec. 1.2 & Sec. 7 (full basin B_1 reaching 1)

```
def InBasin1 (n : ℕ) : Prop := Reaches f n 1
```

Ouroboros.InBasin5 (Ouroboros/Defs.lean:46-46) — Sec. 1.2 & Sec. 7 (full basin B_5 reaching 5)

```
def InBasin5 (n : ℕ) : Prop := Reaches f n 5
```

Ouroboros.KCal (Ouroboros/CantorClosure.lean:978-978) — Thm. 18 (2-adic Cantor survival set K_r)

```
def KCal (r : ℕ) : Set ℤ_[2] := {x : ℤ_[2] | ∀ h : ℕ, Survives r h (x.appr h)}
```

Ouroboros.N1 (Ouroboros/Defs.lean:53-55) — Sec. 1.3 (odd basin-of-1 count)

```
noncomputable def N1 (X : ℕ) : ℕ := by
  classical
  exact ((Finset.Icc 1 X).filter (fun n => n % 2 = 1 ∧ Reaches f n 1)).card
```

Ouroboros.N1All (Ouroboros/Defs.lean:58-60) — Sec. 1.3 (all positive starts reaching 1)

```
noncomputable def N1All (X : ℕ) : ℕ := by
  classical
  exact ((Finset.Icc 1 X).filter (fun n => Reaches f n 1)).card
```

Ouroboros.N5 (Ouroboros/Defs.lean:63-65) — Sec. 1.3 (odd basin-of-5 count)

```
noncomputable def N5 (X : ℕ) : ℕ := by
  classical
  exact ((Finset.Icc 1 X).filter (fun n => n % 2 = 1 ∧ Reaches f n 5)).card
```

Ouroboros.N5All (Ouroboros/Defs.lean:68-70) — Sec. 1.3 (all positive starts reaching 5)

```
noncomputable def N5All (X : ℕ) : ℕ := by
  classical
  exact ((Finset.Icc 1 X).filter (fun n => Reaches f n 5)).card
```

Ouroboros.NoDescent (Ouroboros/DensityDescent.lean:701-702) — Cor. 4 (positive integers never descending under f_10)

```
def NoDescent : Set ℕ :=
  {n | 2 ≤ n ∧ ∀ m : ℕ, 1 ≤ m → n ≤ f^[m] n}
```

Ouroboros.OddCycle (Ouroboros/CycleBounds.lean:90-97) — Sec. 5 (odd-member cycle structure)

```
structure OddCycle (m : ℕ) where
  hm : 0 < m
  x : Fin m → ℕ
  k : Fin m → ℕ
  x_pos : ∀ i, 0 < x i
  x_odd : ∀ i, x i % 2 = 1
  k_pos : ∀ i, 1 ≤ k i
  step : ∀ i : Fin m, d (x i) * x i + 1 = 2 ^ (k i) * x (finNext hm i)
```

Ouroboros.OddCycle.totalHalvings (Ouroboros/CycleBounds.lean:104-104) — Sec. 5 (total cycle halvings K)

Context binders in scope: `variable {m : ℕ} (C : OddCycle m)`

```
def totalHalvings : ℕ := ∑ i : Fin m, C.k i
```

Ouroboros.RSet (Ouroboros/OddBases.lean:69-69) — Lem. 28 (odd-base set R_b reaching 1)

```
def RSet (b : ℕ) : Set ℕ := {n | 1 ≤ n ∧ Reaches (f_b b) n 1}
```

Ouroboros.Reaches (Ouroboros/Defs.lean:39-40) — Sec. 1.2 (forward orbit reachability)

```
def Reaches (g : ℕ → ℕ) (n m : ℕ) : Prop :=
  ∃ k : ℕ, g^[k] n = m
```

Ouroboros.T (Ouroboros/Defs.lean:31-32) — Sec. 1.3 (base-10 shortcut map T_10)

```
def T (n : ℕ) : ℕ :=
  if n % 2 = 0 then n / 2 else (d n * n + 1) / 2
```

Ouroboros.T_C (Ouroboros/Defs.lean:35-36) — Sec. 1.2 (classical Collatz shortcut map)

```
def T_C (y : ℕ) : ℕ :=
  if y % 2 = 0 then y / 2 else (3 * y + 1) / 2
```

Ouroboros.ZSet (Ouroboros/OddBases.lean:66-66) — Thm. 12 & Lem. 28 (odd-base zero-digit trapdoor set Z_b)

```
def ZSet (b : ℕ) : Set ℕ := {n | 1 ≤ n ∧ ∃ j : ℕ, b ∣ (f_b b)^[j] n}
```

Ouroboros.a1 (Ouroboros/BasinOne.lean:442-442) — Thm. 18 (unique positive root of a_1 + a_1^4 = 1)

```
noncomputable def a1 : ℝ := Classical.choose exists_unique_a1
```

Ouroboros.aStar (Ouroboros/BasinOne.lean:565-565) — Thm. 5 (unique positive root a* of a + 2a^4 = 1 via Classical.choose)

```
noncomputable def aStar : ℝ := Classical.choose exists_unique_aStar
```

Ouroboros.beta (Ouroboros/BasinOne.lean:587-587) — Thm. 5 (quartic exponent beta = -log_2(a*))

```
noncomputable def beta : ℝ := Real.logb 2 (1 / aStar)
```

Ouroboros.betaPrime (Ouroboros/BasinOneUpperSharp.lean:44-44) — Thm. 8 (Archimedean upper exponent 0.6245)

```
noncomputable def betaPrime : ℝ := 6245 / 10000
```

Ouroboros.d (Ouroboros/Defs.lean:28-28) — Sec. 1 (last decimal digit n mod 10)

```
def d (n : ℕ) : ℕ := n % 10
```

Ouroboros.exists_unique_aStar (Ouroboros/BasinOne.lean:478-562) — Thm. 5 (existence-uniqueness witness for Classical.choose)

```
lemma exists_unique_aStar :
    ∃! a : ℝ, 0 < a ∧ a < 1 ∧
      (1 - a - a ^ 4) * ((1 - a) * (1 - a ^ 4) - a ^ 4) = a ^ 9
```

Ouroboros.f (Ouroboros/Defs.lean:25-25) — Sec. 1 (base-10 Ouroboros map f_10)

```
def f (n : ℕ) : ℕ := f_b 10 n
```

Ouroboros.f_b (Ouroboros/Defs.lean:21-22) — Sec. 1, eq. (1) (base-b last-digit Collatz map)

```
def f_b (b n : ℕ) : ℕ :=
  if n % 2 = 0 then n / 2 else (n % b) * n + 1
```

Ouroboros.gamma (Ouroboros/BasinOne.lean:462-462) — Thm. 18 (lower exponent gamma = -log_2 a_1)

```
noncomputable def gamma : ℝ := Real.logb 2 (1 / a1)
```

Ouroboros.liveMod30 (Ouroboros/SuffixClass.lean:30-30) — Thm. 29 (live residues S = {0,1,2,4,8,15,16,17} mod 30)

```
def liveMod30 : Finset ℕ := {0, 1, 2, 4, 8, 15, 16, 17}
```

Ouroboros.oddRunMass (Ouroboros/OddSurvey.lean:130-131) — Thm. 13 (canonical finite-cylinder escape mass rho_b)

```
noncomputable def oddRunMass (b : ℕ) : ℝ :=
  (∑ x ∈ Finset.Ico 1 b, oddRunStateMass b x) / (b : ℝ)
```

Ouroboros.oddTrapMass (Ouroboros/OddSurvey.lean:126-127) — Thm. 13 (canonical finite-cylinder trap mass tau_b)

```
noncomputable def oddTrapMass (b : ℕ) : ℝ :=
  (1 + ∑ x ∈ Finset.Ico 1 b, oddTrapStateMass b x) / (b : ℝ)
```

Ouroboros.p_infty (Ouroboros/EvenStopping.lean:57-58) — Sec. 1.3 & Thm. 11 (limiting non-descent density p_infty(b))

```
noncomputable def p_infty (b : ℕ) : ℝ :=
  ∏ k : ℕ, p_k b (k + 1)
```

Ouroboros.p_k (Ouroboros/EvenStopping.lean:53-54) — Sec. 1.3 & Thm. 11 (k-step even-base non-descent density p_k(b))

```
noncomputable def p_k (b k : ℕ) : ℝ :=
  ((SurvRes b k).card : ℝ) / ((b * 2 ^ (k - 1) : ℕ) : ℝ)
```

Ouroboros.suffixClosure1 (Ouroboros/SuffixDimension.lean:183-184) — Cor. 30 (reversed base-b ending closure E_b for B_1)

```
def suffixClosure1 (b : ℕ) : Set ℝ :=
  closure (revExp b '' {n : ℕ | 0 < n ∧ InBasin1 n})
```

Ouroboros.suffixClosure5 (Ouroboros/SuffixDimension.lean:187-188) — Cor. 30 (reversed base-b ending closure for B_5)

```
def suffixClosure5 (b : ℕ) : Set ℝ :=
  closure (revExp b '' {n : ℕ | 0 < n ∧ InBasin5 n})
```

Ouroboros.suffixLambda (Ouroboros/SuffixDimension.lean:35-35) — Cor. 30 (unique positive root of lambda^4 = lambda^3 + 2)

```
noncomputable def suffixLambda : ℝ := 1 / aStar
```

Selected Concrete Sanity Examples (checked in Lean; examples only, not global identity proofs): Ouroboros.f_cycle_one (f^[2] 1 = 1); Ouroboros.f_cycle_five (f^[6] 5 = 5); Ouroboros.f_ex_3 (f 3 = 10); Ouroboros.f_ex_31_reaches_1 (f^[6] 31 = 1); Ouroboros.f_ex_7_reaches_5 (f^[40] 7 = 5).

Supporting Semantic Closure Index (non-Box-1 helpers): Ouroboros.ABCReach (Ouroboros/BasinOneMoves.lean), Ouroboros.AllRunsDiv4 (Ouroboros/BasinOne.lean), Ouroboros.AllStationaryDriftsSatisfy (Ouroboros/EvenBases.lean), Ouroboros.AllowedMod30 (Ouroboros/CantorClosure.lean), Ouroboros.B1Set (Ouroboros/CantorClosure.lean), Ouroboros.BitSeq (Ouroboros/CantorClosure.lean), Ouroboros.CylSubset (Ouroboros/OddBases.lean), Ouroboros.EvenBasesCert.hStepNat (Ouroboros/EvenBasesCert.lean), Ouroboros.EvenBasesCert.oddDigitNat (Ouroboros/EvenBasesCert.lean), Ouroboros.EvenBasesCert.qStepNat (Ouroboros/EvenBasesCert.lean), Ouroboros.ExitsAt19Or23 (Ouroboros/CantorClosure.lean), Ouroboros.HasBoxDim (Ouroboros/CantorClosure.lean), Ouroboros.InA (Ouroboros/BasinOne.lean), Ouroboros.InABC (Ouroboros/BasinOne.lean), Ouroboros.InABC15 (Ouroboros/BasinOneMoves.lean), Ouroboros.InB (Ouroboros/BasinOne.lean), Ouroboros.InC (Ouroboros/BasinOne.lean), Ouroboros.IsABCMove (Ouroboros/BasinOneMoves.lean), Ouroboros.IsCertifiedEscape (Ouroboros/OddSurvey.lean), Ouroboros.IsStationary2M (Ouroboros/EvenBases.lean), Ouroboros.IsStationary4k (Ouroboros/EvenBases.lean), Ouroboros.IsTrapStretch (Ouroboros/OddSurvey.lean), Ouroboros.IsZeroStretch (Ouroboros/OddSurvey.lean), Ouroboros.KrPadic (Ouroboros/CantorClosure.lean), Ouroboros.KrSet (Ouroboros/CantorClosure.lean), Ouroboros.OddCycle.aExp (Ouroboros/CycleBounds.lean), Ouroboros.OddCycle.bExp (Ouroboros/CycleBounds.lean), Ouroboros.OddCycle.cExp (Ouroboros/CycleBounds.lean), Ouroboros.OddCycle.countDigit (Ouroboros/CycleBounds.lean), Ouroboros.OddCycle.countTrans (Ouroboros/CycleBounds.lean), Ouroboros.OddCycle.n1 (Ouroboros/CycleBounds.lean), Ouroboros.OddCycle.n3 (Ouroboros/CycleBounds.lean), Ouroboros.OddCycle.n5 (Ouroboros/CycleBounds.lean), Ouroboros.OddCycle.n7 (Ouroboros/CycleBounds.lean), Ouroboros.OddCycle.n9 (Ouroboros/CycleBounds.lean), Ouroboros.OddCyl (Ouroboros/OddBases.lean), Ouroboros.OddStretchOutcome (Ouroboros/OddSurvey.lean), Ouroboros.P2MOp (Ouroboros/EvenBases.lean), Ouroboros.P4kOp (Ouroboros/EvenBases.lean), Ouroboros.ReachesOneDigitOne (Ouroboros/BasinOne.lean), Ouroboros.Sigma15Set (Ouroboros/CantorClosure.lean), Ouroboros.SigmaSet (Ouroboros/CantorClosure.lean), Ouroboros.SurvRes (Ouroboros/EvenStopping.lean), Ouroboros.Survives (Ouroboros/CantorClosure.lean), Ouroboros.T_b (Ouroboros/EvenCoding.lean), Ouroboros.TypeId (Ouroboros/BasinFive.lean), Ouroboros.affAB (Ouroboros/EvenCoding.lean), Ouroboros.affine30 (Ouroboros/CantorClosure.lean), Ouroboros.allOddBases1999 (Ouroboros/OddSurvey.lean), Ouroboros.baseDigit (Ouroboros/SuffixDimension.lean), Ouroboros.boxCoverNum (Ouroboros/CantorClosure.lean), Ouroboros.boxCoverNumber (Ouroboros/SuffixDimension.lean), Ouroboros.childType (Ouroboros/BasinFive.lean), Ouroboros.codingMap (Ouroboros/DensityDescent.lean), Ouroboros.countFrac (Ouroboros/OddBases.lean), Ouroboros.countUpTo (Ouroboros/DensityDescent.lean), Ouroboros.cylStep (Ouroboros/OddBases.lean), Ouroboros.digit4k (Ouroboros/EvenBases.lean), Ouroboros.digitVal (Ouroboros/BasinFive.lean), Ouroboros.drift2M (Ouroboros/EvenBases.lean), Ouroboros.drift4k (Ouroboros/EvenBases.lean), Ouroboros.exists_unique_a1 (Ouroboros/BasinOne.lean), Ouroboros.finNext (Ouroboros/CycleBounds.lean), Ouroboros.hStep2M (Ouroboros/EvenBases.lean), Ouroboros.lowerBoxDim (Ouroboros/SuffixDimension.lean), Ouroboros.movesOf (Ouroboros/BasinFive.lean), Ouroboros.next4kA (Ouroboros/EvenBases.lean), Ouroboros.next4kB (Ouroboros/EvenBases.lean), Ouroboros.oddDigit2M (Ouroboros/EvenBases.lean), Ouroboros.oddRunProb (Ouroboros/OddSurvey.lean), Ouroboros.oddRunStateMass (Ouroboros/OddSurvey.lean), Ouroboros.oddStretchOutcome (Ouroboros/OddSurvey.lean), Ouroboros.oddTrapProb (Ouroboros/OddSurvey.lean), Ouroboros.oddTrapStateMass (Ouroboros/OddSurvey.lean), Ouroboros.parityVec (Ouroboros/DensityDescent.lean), Ouroboros.predCondBool (Ouroboros/BasinFive.lean), Ouroboros.prependBits (Ouroboros/CantorClosure.lean), Ouroboros.psiMap (Ouroboros/CantorClosure.lean), Ouroboros.qStep2M (Ouroboros/EvenBases.lean), Ouroboros.revExp (Ouroboros/SuffixDimension.lean), Ouroboros.splitBases (Ouroboros/OddSurvey.lean), Ouroboros.splitTwo (Ouroboros/BasinOne.lean), Ouroboros.suffixDimVal (Ouroboros/SuffixDimension.lean), Ouroboros.suffixH (Ouroboros/SuffixClass.lean), Ouroboros.suffixM (Ouroboros/SuffixClass.lean), Ouroboros.suffixP (Ouroboros/SuffixClass.lean), Ouroboros.trailingOnes (Ouroboros/BasinOne.lean), Ouroboros.trapOnlyBases (Ouroboros/OddSurvey.lean), Ouroboros.upperBoxDim (Ouroboros/SuffixDimension.lean).

Box 2 — Verbatim Main Theorem Signatures, Context Binders & Applicability

Ouroboros.prop_base4_iff_collatz (Ouroboros/BaseFamily.lean:112-137) src:462b39914a6367

```
theorem prop_base4_iff_collatz :
    (∀ n : ℕ, 0 < n → Reaches (f_b 4) n 1 ∨ Reaches (f_b 4) n 3) ↔
    (∀ y : ℕ, 0 < y → Reaches T_C y 1)
```

Assumptions: Closed equivalence over f_b 4, T_C, and Reaches with zero free hypotheses. Applicability: Unconditional (closed statement; no outer premise witness needed).

Ouroboros.thm_terras_bound (Ouroboros/DensityDescent.lean:666-698) src:4463828e653e9f

```
theorem thm_terras_bound (k : ℕ) (hk : 1 ≤ k) (ε : ℝ) (hε : 0 < ε) :
    ∃ N₀ : ℕ, ∀ N : ℕ, N₀ ≤ N →
      (countUpTo (E_k k) N : ℝ) / (N : ℝ) ≤
        (1124953 / 1000000 : ℝ) * (996423 / 1000000 : ℝ) ^ k + ε
```

Assumptions: Any integer k >= 1 and any real eps > 0. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_terras_density_one (Ouroboros/DensityDescent.lean:716-741) src:1afc5e6b962368

```
theorem thm_terras_density_one (ε : ℝ) (hε : 0 < ε) :
    ∃ N₀ : ℕ, ∀ N : ℕ, N₀ ≤ N →
      (countUpTo NoDescent N : ℝ) / (N : ℝ) ≤ ε
```

Assumptions: Any real eps > 0. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_basin1_upper (Ouroboros/BasinOne.lean:1805-1869) src:7f2d4403a96dc8

```
theorem thm_basin1_upper (X : ℕ) (hX : 1 ≤ X) :
    (N1 X : ℝ) ≤ (244 / 100 : ℝ) * (X : ℝ) ^ beta
```

Assumptions: Any integer cutoff X >= 1. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_basin5_tight (Ouroboros/BasinFiveUncond.lean:526-554) src:50137b24ff406f

```
theorem thm_basin5_tight (X : ℕ) (hX : 3 ≤ X) :
    (605 / 1000000 : ℝ) * (X : ℝ) ^ ((65 : ℝ) / 100) ≤ (N5 X : ℝ)
```

Assumptions: Any integer cutoff X >= 3. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.cor_basin_separation_tendsto_zero_unconditional (Ouroboros/BasinFiveUncond.lean:581-616) src:7cf25212efa0e0

```
theorem cor_basin_separation_tendsto_zero_unconditional (ε : ℝ) (hε : 0 < ε) :
    ∃ X₀ : ℕ, ∀ X : ℕ, X₀ ≤ X →
      (N1 X : ℝ) / (N5 X : ℝ) ≤ ε ∧ (N1All X : ℝ) / (N5All X : ℝ) ≤ ε
```

Assumptions: Any real eps > 0. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_basin_sharp (Ouroboros/BasinSharpAll.lean:94-115) src:291016b07462cf

```
theorem thm_basin_sharp :
    (∀ X : ℝ, 1 ≤ X →
      (806 / 10000 : ℝ) * (X / 31) ^ ((399 : ℝ) / 800) ≤ (N1 ⌊X⌋₊ : ℝ) ∧
      (145 / 10000 : ℝ) * X ^ ((399 : ℝ) / 800) ≤ (N1 ⌊X⌋₊ : ℝ)) ∧
    (∃ C : ℝ, 0 < C ∧ ∀ X : ℝ, 1 ≤ X → (N1 ⌊X⌋₊ : ℝ) ≤ C * X ^ betaPrime) ∧
    (Filter.limsup (fun X : ℝ => Real.log (N1 ⌊X⌋₊ : ℝ) / Real.log X) Filter.atTop ≤
      ↪ betaPrime ∧
      betaPrime < beta ∧
      (626382135 / 1000000000 : ℝ) < beta ∧ beta < (626382145 / 1000000000 : ℝ) ∧
      dimH (closure B1Padic) = ENNReal.ofReal beta ∧
      ENNReal.ofReal betaPrime < dimH (closure B1Padic)) ∧
    (∀ X : ℕ, 3 ≤ X →
      ((98959 : ℝ) / ((2269527 : ℝ) * (3 : ℝ) ^ ((13 : ℝ) / 20))) * (X : ℝ) ^ ((13 :
        ↪ ℝ) / 20) ≤
        (N5 X : ℝ) ∧
      (2134 / 100000 : ℝ) * (X : ℝ) ^ ((13 : ℝ) / 20) < (N5 X : ℝ)) ∧
    (∃ C : ℝ, 0 < C ∧ ∀ X : ℕ, 3 ≤ X →
      (N1 X : ℝ) / (N5 X : ℝ) ≤ C * (X : ℝ) ^ (-(255 : ℝ) / 10000))
```

Assumptions: Closed bundled theorem over N1, N5, beta, betaPrime, B1Padic, and dimH with zero free hypotheses. Applicability: Unconditional (closed statement; no outer premise witness needed).

Ouroboros.thm_even_bases (Ouroboros/EvenBases.lean:1301-1347) src:ff6f11c7fc7bf4

```
theorem thm_even_bases {b : ℕ} (hb2 : 2 ≤ b) (hbeven : b % 2 = 0) :
    HasStationaryDriftIn b (· ≤ 0) ↔ b = 2 ∨ b = 4 ∨ b = 6 ∨ b = 10
```

Assumptions: Any even integer base b >= 2. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_even_stopping (Ouroboros/EvenStopping.lean:2464-2474) src:27757ee3a9c0af

```
theorem thm_even_stopping {b : ℕ} (hb2 : 2 ≤ b) (hbeven : b % 2 = 0) :
    (∀ k : ℕ, 1 ≤ k → HasNaturalDensity (D_k b k) (p_k b k)) ∧
    Filter.Tendsto (fun k : ℕ => p_k b (k + 1)) Filter.atTop (nhds (p_infty b)) ∧
    (p_infty b = 0 ↔ b = 2 ∨ b = 4 ∨ b = 6 ∨ b = 10) ∧
    ((b = 2 ∨ b = 4 ∨ b = 6 ∨ b = 10) → HasNaturalDensity (DescentSet b) 1) ∧
    ((b ≠ 2 ∧ b ≠ 4 ∧ b ≠ 6 ∧ b ≠ 10) → ∃ η : ℝ, 0 < η ∧ ∀ k : ℕ, 1 ≤ k → η ≤ p_k b k)
```

Assumptions: Any even integer base b >= 2. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_odd_trapdoor_rootless (Ouroboros/OddBases.lean:117-144) src:84652588160e3f

```
theorem thm_odd_trapdoor_rootless {b p : ℕ} (hb_odd : b % 2 = 1)
    (hp : Nat.Prime p) (hp3 : p % 4 = 3) (hpb : p ∣ b) :
    (∀ x : ℕ, (x * x + 1) % b ≠ 0) ∧
    (∀ n : ℕ, n ∈ ZSet b ↔ 1 ≤ n ∧ b ∣ n)
```

Assumptions: Any odd b and prime p = 3 (mod 4) dividing b. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_odd_survey (Ouroboros/OddSurvey.lean:2943-3009) src:d037c8f51b920d

```
theorem thm_odd_survey :
    (∀ b : ℕ, 3 ≤ b → b % 2 = 1 →
      (1 : ℝ) / (b : ℝ) ≤ oddTrapMass b ∧
      0 ≤ oddRunMass b ∧
      oddTrapMass b + oddRunMass b ≤ 1 ∧
      (∀ ε > 0, ∀ᶠ N in atTop, oddTrapMass b - ε ≤ countFrac (ZSet b) N) ∧
      (∀ ε > 0, ∀ᶠ N in atTop, oddRunMass b - ε ≤ countFrac (DivSet b) N) ∧
      (oddTrapMass b + oddRunMass b = 1 →
        Tendsto (countFrac (ZSet b)) atTop (𝒩 (oddTrapMass b)) ∧
        Tendsto (countFrac (RSet b)) atTop (𝒩 (oddTrapMass b)) ∧
        Tendsto (countFrac (DivSet b)) atTop (𝒩 (oddRunMass b)) ∧
        (∃ qτ qρ : ℚ, (qτ : ℝ) = oddTrapMass b ∧ (qρ : ℝ) = oddRunMass b ∧ qτ + qρ =
          ↪ 1))) ∧
    allOddBases1999.card = 999 ∧
    trapOnlyBases ⊆ allOddBases1999 ∧
    splitBases ⊆ allOddBases1999 ∧
    Disjoint trapOnlyBases splitBases ∧
    trapOnlyBases.card = 27 ∧
    splitBases.card = 878 ∧
    resolvedOddBases.card = 905 ∧
    (∀ b ∈ trapOnlyBases,
      oddTrapMass b = 1 ∧ oddRunMass b = 0 ∧
      oddTrapMass b + oddRunMass b = 1 ∧
      Tendsto (countFrac (ZSet b)) atTop (𝒩 (1 : ℝ)) ∧
      Tendsto (countFrac (RSet b)) atTop (𝒩 (1 : ℝ)) ∧
      Tendsto (countFrac (DivSet b)) atTop (𝒩 (0 : ℝ))) ∧
    (∀ b ∈ splitBases,
      0 < oddTrapMass b ∧ 0 < oddRunMass b ∧
      oddTrapMass b + oddRunMass b = 1 ∧
      (∃ qτ qρ : ℚ, (qτ : ℝ) = oddTrapMass b ∧ (qρ : ℝ) = oddRunMass b ∧ qτ + qρ = 1)
        ↪ ∧
      Tendsto (countFrac (ZSet b)) atTop (𝒩 (oddTrapMass b)) ∧
      Tendsto (countFrac (RSet b)) atTop (𝒩 (oddTrapMass b)) ∧
      Tendsto (countFrac (DivSet b)) atTop (𝒩 (oddRunMass b)))
```

Assumptions: Closed universal absorption theorem for all odd b >= 3 together with the 905-of-999 census through 1,999. Applicability: Unconditional (closed statement; no outer premise witness needed).

Ouroboros.lemma_coding_bijective (Ouroboros/DensityDescent.lean:314-322) src:3299046e1c753a

```
theorem lemma_coding_bijective (k : ℕ) :
    Function.Bijective (fun r : Fin (5 * 2 ^ k) => codingMap k r.val)
```

Assumptions: Any natural number k. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_residue_law (Ouroboros/BasinOne.lean:181-203) src:62a66a5d711a7b

```
theorem thm_residue_law {n : ℕ} (hn : n % 2 = 1) (h : Reaches f n 1) : InABC n
```

Assumptions: Any odd integer n with Reaches f n 1. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.lem_basin1_moves_classify (Ouroboros/BasinOneMoves.lean:66-214) src:abbbfedec5c7be

```
theorem lem_basin1_moves_classify {o v m : ℕ} (ho : InABC15 o) (hv : 1 ≤ v)
    (hm_odd : m % 2 = 1) (heq : d m * m + 1 = 2 ^ v * o) :
    d m ≠ 3 ∧ d m ≠ 9 ∧ InABC15 m ∧
    ((o % 15 = 1 ∧ d m = 1 ∧ v % 4 = 1 ∧ m % 15 = 1) ∨
     (o % 15 = 2 ∧ d m = 1 ∧ v % 4 = 0 ∧ m % 15 = 1) ∨
     ((o % 15 = 1 ∨ o % 15 = 2) ∧ d m = 5 ∧ (2 ^ v * o) % 25 = 1 ∧ m % 15 = 0) ∨
     (o % 15 = 0 ∧ d m = 7 ∧ (2 ^ v * o) % 7 = 1 ∧ m % 15 = 2))
```

Assumptions: Any o in A cup B cup C, v >= 1, and odd m with d(m)*m + 1 = 2^v * o. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.prop_binary_test (Ouroboros/BasinOne.lean:304-407) src:068eac6e134a10

```
theorem prop_binary_test (b : ℕ) :
    ReachesOneDigitOne (2 * b + 1) ↔ AllRunsDiv4 b
```

Assumptions: Any natural number b. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_cantor (Ouroboros/CantorClosure.lean:3317-3357) src:cb88976e22fd11

```
theorem thm_cantor :
    -- Roots & exponents (explicitly tying `a1` and `aStar` to their unique positive
      ↪ quartic roots)
    (0 < a1 ∧ a1 < 1 ∧ a1 + a1 ^ 4 = 1) ∧
    (∃! a : ℝ, 0 < a ∧ a < 1 ∧ a + a ^ 4 = 1) ∧
    (0 < aStar ∧ aStar < 1 ∧ aStar + 2 * aStar ^ 4 = 1) ∧
    (∃! a : ℝ, 0 < a ∧ a < 1 ∧ a + 2 * a ^ 4 = 1) ∧
    (gamma = -Real.logb 2 a1 ∧ beta = -Real.logb 2 aStar) ∧
    ((464958 / 1000000 : ℝ) < gamma ∧ gamma < (464959 / 1000000 : ℝ)) ∧
    ((626382135 / 1000000000 : ℝ) < beta ∧ beta < (626382145 / 1000000000 : ℝ)) ∧
    ((626382 / 1000000 : ℝ) < beta ∧ beta < (626383 / 1000000 : ℝ)) ∧
    -- Part (1): Exact 2-adic closure, ball dichotomy, nowhere-denseness
    ((∀ r : ℕ, (r = 1 ∨ r = 15 ∨ r = 17) → closure (KrPadic r) = KCal r) ∧
      closure B1Padic =
        affine30 1 '' KCal 1 ∪ affine30 15 '' KCal 15 ∪ affine30 17 '' KCal 17 ∧
      (∀ (x : ℤ_[2]) (ρ : ℝ), (Metric.ball x ρ ∩ closure B1Padic).Nonempty →
        {n : ℕ | 1 ≤ n ∧ Reaches f n 1 ∧ (n : ℤ_[2]) ∈ Metric.ball x ρ}.Infinite ∧
        {n : ℕ | 1 ≤ n ∧ InABC n ∧ ExitsAt19Or23 n ∧ (n : ℤ_[2]) ∈ Metric.ball x
          ↪ ρ}.Infinite) ∧
      IsNowhereDense (closure B1Padic)) ∧
    -- Part (2): Self-similar isometry, Hausdorff measure, dimH = dimB = β
    ((∀ r : ℕ, Isometry (psiMap r)) ∧
      (∀ r : ℕ, (r = 1 ∨ r = 17) → psiMap r '' KCal r = SigmaSet) ∧
      psiMap 15 '' KCal 15 = Sigma15Set ∧
      Disjoint (prependBits [false, false, false] '' SigmaSet)
        (prependBits [false, false, true] '' SigmaSet) ∧
      ENNReal.ofReal (1 / 2) ≤ μH[beta] (closure B1Padic) ∧
      μH[beta] (closure B1Padic) ≤ ENNReal.ofReal (aStar⁻¹ ^ 2 + aStar) ∧
      μH[beta] (closure B1Padic) < ENNReal.ofReal (3031 / 1000) ∧
      dimH (closure B1Padic) = ENNReal.ofReal beta ∧
      HasBoxDim (closure B1Padic) beta) ∧
    -- Part (3): Elementary integer bounds & density zero
    ((∀ X : ℝ, 1 ≤ X → a1 ^ 9 * X ^ gamma ≤ (N1 ⌊X⌋₊ : ℝ) ∧ (N1 ⌊X⌋₊ : ℝ) ≤ X ^ beta)
      ↪ ∧
      (∀ ε : ℝ, 0 < ε → ∃ X₀ : ℕ, ∀ X : ℕ, X₀ ≤ X → (N1All X : ℝ) / (X : ℝ) ≤ ε))
```

Assumptions: Closed bundled theorem over a1, aStar, gamma, beta, B1Padic, KCal, and N1 with zero free hypotheses. Applicability: Unconditional (closed statement; no outer premise witness needed).

Ouroboros.prop_no_residue_size_bound (Ouroboros/BasinOneMoves.lean:301-396) src:519cbd566e3658

```
theorem prop_no_residue_size_bound {Q : ℕ} (hQ : 1 ≤ Q) {c : ℝ} (hc : 0 < c)
    (w : Fin Q → ℝ) :
    ∃ (o n : ℕ), o % 30 = 15 ∧ 1 ≤ n ∧ ABCReach o n ∧
      w ⟨o % Q, Nat.mod_lt o (by omega)⟩ * ((n : ℝ) / (o : ℝ)) ^ c < 1
```

Assumptions: Any modulus Q >= 1, real exponent c > 0, and vector w : Fin Q -> R. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.lem_weighted_tree3 (Ouroboros/WeightedCharging.lean:264-279) src:3ea527b1d5952e

```
theorem lem_weighted_tree3
    {Q Vmax : ℕ} (hQ1575 : 1575 ∣ Q) (hQpos : 0 < Q)
    {s L : ℝ} (hs : 0 < s) (hL : 0 < L)
    (w : TypeId Q → ℝ) (hw_nn : ∀ t, 0 ≤ w t)
    (hexpand : ∀ (t : TypeId Q) (R : Fin Q),
      R.val % (Q / digitVal t.1) = t.2.val →
      w t ≤ ∑ mv ∈ movesOf Q Vmax R,
        ((digitVal mv.1 : ℝ) / (2 : ℝ) ^ (mv.2.val + 1)) ^ s *
          w (childType Q Vmax hQ1575 hQpos R mv))
    (hbudget : ∀ R : Fin Q,
      ∑ mv ∈ movesOf Q Vmax R, w (childType Q Vmax hQ1575 hQpos R mv) ≤ L)
    {o X : ℕ} (ho_pos : 0 < o) (ho_odd : o % 2 = 1) (ho_tree : Reaches f o 3) (ho_le
      ↪ : o ≤ X)
    (t : TypeId Q) (ht : o % (Q / digitVal t.1) = t.2.val) :
    ((X : ℝ) / (o : ℝ)) ^ s * w t ≤ L * (NTree o X : ℝ)
```

Assumptions: Any 1575 | Q, Q > 0, Vmax, s > 0, L > 0, nonnegative weights satisfying expansion and row-sum cap, odd o reaching 3, integer X >= o, and compatible type t. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.lem_halving_budget_sharp (Ouroboros/CycleBounds.lean:721-786) src:1e3002373ef697

```
theorem lem_halving_budget_sharp {m : ℕ} (C : OddCycle m) :
    C.n1 + 2 * C.n3 + 4 * C.n7 + 3 * C.n9 + 4 * C.countTrans 5 1 ≤ C.totalHalvings
```

Assumptions: Any OddCycle m of f_10 (satisfied by the known cycles (1) and (5,13), and constraining any hypothetical further cycle). Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: pending_replay).

Ouroboros.lem_halving_budget (Ouroboros/CycleBounds.lean:790-796) src:281a2212305f18

```
theorem lem_halving_budget {m : ℕ} (C : OddCycle m) :
    C.n1 + 2 * C.n3 + 4 * C.n7 + 3 * C.n9 ≤ C.totalHalvings ∧
      2 * C.n1 + 3 * C.aExp + C.bExp + 7 * C.cExp ≤ 2 * C.totalHalvings
```

Assumptions: Companion consequence for any OddCycle m of f_10 proving the unsharpened budget and 2n_1 + 3a + e + 7c <= 2K (CycleBounds.lean:790-796). Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: pending_replay).

Ouroboros.cycle_m3_unique (Ouroboros/CycleBounds.lean:1484-1679) src:55791bef2fb800

```
theorem cycle_m3_unique (C : OddCycle 3) :
    ∀ i : Fin 3, C.x i = 1 ∧ C.k i = 1
```

Assumptions: Any OddCycle 3 of f_10 (satisfied by the trivial repetition (1,1,1)). Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: pending_replay).

Ouroboros.cycle_lattice_box (Ouroboros/CycleBounds.lean:999-1154) src:188a5ca7f8b0bc

```
theorem cycle_lattice_box {m X L S lam3 lam5 lam7 : ℕ} (C : OddCycle m)
    (hX : 2 ≤ X) (hmin : ∀ i : Fin m, X ≤ C.x i)
    (_hL : 1 ≤ L) (_hLX : 2 * L ≤ X) (hKL : C.totalHalvings ≤ L) (hS : 1 ≤ S)
    (h3 : (lam3 : ℝ) ≤ (S : ℝ) * Real.logb 2 3 ∧ (S : ℝ) * Real.logb 2 3 < (lam3 : ℝ)
      ↪ + 1)
    (h5 : (lam5 : ℝ) ≤ (S : ℝ) * Real.logb 2 5 ∧ (S : ℝ) * Real.logb 2 5 < (lam5 : ℝ)
      ↪ + 1)
    (h7 : (lam7 : ℝ) ≤ (S : ℝ) * Real.logb 2 7 ∧ (S : ℝ) * Real.logb 2 7 < (lam7 : ℝ)
      ↪ + 1) :
    C.aExp * lam3 < L * S ∧
    C.bExp * lam5 < L * S ∧
    C.cExp * lam7 < L * S ∧
    1 ≤ (C.totalHalvings : ℤ) * (S : ℤ) -
      (C.aExp : ℤ) * (lam3 : ℤ) - (C.bExp : ℤ) * (lam5 : ℤ) - (C.cExp : ℤ) * (lam7 :
        ↪ ℤ) ∧
    ((C.totalHalvings : ℤ) * (S : ℤ) -
      (C.aExp : ℤ) * (lam3 : ℤ) - (C.bExp : ℤ) * (lam5 : ℤ) - (C.cExp : ℤ) * (lam7 :
        ↪ ℤ) : ℝ) <
      (3 * (L : ℝ) * (S : ℝ)) / (2 * (X : ℝ)) + (L : ℝ)
```

Assumptions: Hypothetical OddCycle m with all odd members >= X >= 2 and total halvings K <= L. Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: pending_replay).

Ouroboros.thm_cycle_tail_64000 (Ouroboros/CycleBounds.lean:985-993) src:b98690f78bbaeb

```
theorem thm_cycle_tail_64000 {m : ℕ} (C : OddCycle m)
    (hmin : ∀ i : Fin m, 10 ^ 15 ≤ C.x i) (hK : 64000 < C.totalHalvings) :
    20190 ≤ m ∧ 84191 ≤ C.totalHalvings + m
```

Assumptions: Hypothetical OddCycle m with all odd members >= 10^15 and total halvings K > 64,000. Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: pending_replay).

Ouroboros.cor_drift_growth (Ouroboros/EvenBases.lean:1539-1565) src:159ffe0be4d05d

```
theorem cor_drift_growth {b : ℕ} (hb2 : 2 ≤ b) (hbeven : b % 2 = 0) :
    AllStationaryDriftsSatisfy b
      (fun x => (1 / 2 : ℝ) * Real.log ((b : ℝ) + 2) - (7 / 2 : ℝ) * Real.log 2 ≤ x)
```

Assumptions: Any even integer base b >= 2. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.prop_near_zero_band (Ouroboros/EvenBases.lean:1389-1535) src:64af37e2c1e41e

```
theorem prop_near_zero_band {b : ℕ} (hb2 : 2 ≤ b) (hbeven : b % 2 = 0) :
    (HasStationaryDriftIn b (fun x => -(1 / 4 : ℝ) ≤ x ∧ x ≤ 1 / 4) ↔
      b = 4 ∨ b = 8 ∨ b = 10 ∨ b = 14) ∧
    (HasStationaryDriftIn b (fun x => -(1 / 4 : ℝ) ≤ x ∧ x < 0) ↔
      b = 4 ∨ b = 10)
```

Assumptions: Any even integer base b >= 2. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.lem_base17_runaway (Ouroboros/OddBases.lean:1285-1342) src:41753d85703035

```
theorem lem_base17_runaway {n : ℕ} (hn1 : 1 ≤ n) (hn_odd : n % 2 = 1) (hn17 : n % 17
  ↪ = 16) :
    (∀ i : ℕ, ¬ 17 ∣ (f_b 17)^[i] n ∧ (f_b 17)^[i] n ≠ 1) ∧
    (∀ i : ℕ, (f_b 17)^[i] n < (f_b 17)^[i + 10] n) ∧
    Filter.Tendsto (fun i => (f_b 17)^[i] n) Filter.atTop Filter.atTop
```

Assumptions: Any odd integer n >= 1 with n mod 17 = 16. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_odd_densities_disjoint (Ouroboros/OddBases.lean:1351-1405) src:fa49af5f449211

```
theorem thm_odd_densities_disjoint {b : ℕ} (hb3 : 3 ≤ b) (hb_odd : b % 2 = 1) :
    ZSet b ⊆ RSet b ∧ Disjoint (DivSet b) (RSet b)
```

Assumptions: Any odd integer base b >= 3. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.thm_suffix_class (Ouroboros/SuffixClass.lean:711-726) src:b0da529d0e04a5

```
theorem thm_suffix_class (a m : ℕ) (hm : 1 ≤ m) :
    Set.Infinite {n : ℕ | 0 < n ∧ n % m = a % m ∧ InBasin5 n} ∧
    (Set.Infinite {n : ℕ | 0 < n ∧ n % m = a % m ∧ InBasin1 n} ↔
      ∃ x : ℕ, x < suffixM m ∧ x % suffixP m = a % suffixP m ∧
        T^[suffixH m] x % 30 ∈ liveMod30) ∧
    ((¬ ∃ x : ℕ, x < suffixM m ∧ x % suffixP m = a % suffixP m ∧
        T^[suffixH m] x % 30 ∈ liveMod30) →
      ∀ n : ℕ, n % m = a % m → ¬ InBasin1 n)
```

Assumptions: Any integers a >= 0 and m >= 1. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Ouroboros.cor_suffix_dimension (Ouroboros/SuffixDimension.lean:2600-2639) src:52153495349241

```
theorem cor_suffix_dimension :
    (0 < suffixLambda ∧ suffixLambda ^ 4 = suffixLambda ^ 3 + 2 ∧
      (154368901 / 10 ^ 8 : ℝ) < suffixLambda ∧ suffixLambda < (1543689014 / 10 ^ 9 :
        ↪ ℝ) ∧
      (∃! lam : ℝ, 0 < lam ∧ lam ^ 4 = lam ^ 3 + 2)) ∧
    (∀ (b t u : ℕ), 2 ≤ b → u % 2 = 1 → b = 2 ^ t * u →
      let s := (Real.log (u : ℝ) + (t : ℝ) * Real.log suffixLambda) / Real.log (b : ℝ)
      suffixClosure1 b ⊆ Set.Icc (0 : ℝ) 1 ∧
      dimH (suffixClosure1 b) = ENNReal.ofReal s ∧
      lowerBoxDim (suffixClosure1 b) = s ∧
      upperBoxDim (suffixClosure1 b) = s ∧
      Filter.Tendsto
        (fun δ : ℝ => Real.log (boxCoverNumber (suffixClosure1 b) δ : ℝ) / Real.log
          ↪ (1 / δ))
        (nhdsWithin 0 (Set.Ioi 0)) (nhds s) ∧
      suffixClosure5 b = Set.Icc (0 : ℝ) 1) ∧
    (dimH (suffixClosure1 2) = ENNReal.ofReal beta ∧
      suffixDimVal 2 = beta ∧
      (626382135 / 1000000000 : ℝ) < suffixDimVal 2 ∧
      suffixDimVal 2 < (626382145 / 1000000000 : ℝ) ∧
      (6245 / 10000 : ℝ) < suffixDimVal 2) ∧
    (dimH (suffixClosure1 10) = ENNReal.ofReal (suffixDimVal 10) ∧
      (887529815 / 1000000000 : ℝ) < suffixDimVal 10 ∧
      suffixDimVal 10 < (887529825 / 1000000000 : ℝ)) ∧
    (dimH (suffixClosure1 7) = ENNReal.ofReal 1 ∧
      suffixDimVal 7 = 1 ∧ suffixClosure1 7 = Set.Icc (0 : ℝ) 1)
```

Assumptions: Closed bundled theorem over suffixLambda, suffixClosure1, suffixClosure5, dimH, and lower/upper box dimensions with zero free hypotheses. Applicability: Unconditional (closed statement; no outer premise witness needed).

Table 3. Lean 4 Type-Trap Audit Checklist — Source-Level Approximation (Gate Replay Pending).[a]

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
`AllRunsDiv4`	`nat_div`	`/ 2`	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
`AllStationaryDriftsSatisfy`	`nat_div`	`/ 2`	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
`AllStationaryDriftsSatisfy`	`nat_div`	`/ 4`	intended_convention	Natural-number division / 4 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
`AllowedMod30`	`nat_mod`	`% 2`	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`B1Set`	`nat_mod`	`% 2`	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`CylSubset`	`nat_mod`	`% 2`	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`CylSubset`	`nat_mod`	`% b`	intended_convention	Natural-number remainder % b operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`CylSubset`	`real_log_rpow`	`^ C2.j`	intended_convention	Real log/rpow ^ C2.j is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffix Lambda > 1, or delta in (0,1)).
`hStepNat`	`nat_div`	`/ 2`	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
`hStepNat`	`nat_mod`	`% M`	intended_convention	Natural-number remainder % M on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
`oddDigitNat`	`nat_mod`	`% 2`	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
`qStepNat`	`nat_div`	`/ 2`	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
`qStepNat`	`nat_mod`	`% M`	intended_convention	Natural-number remainder % M on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
`ExitsAt19Or23`	`nat_mod`	`% 30`	intended_convention	Natural-number remainder % 30 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`HasBoxDim`	`real_div`	`Real.log`	intended_convention	Real denominator Real.log is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
`HasBoxDim`	`real_div`	`ε`	intended_convention	Real denominator ε is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
`HasBoxDim`	`real_log_rpow`	`Real.log`	intended_convention	Real log/rpow Real.log is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
`HasNaturalDensity`	`real_div`	`(N : ℝ)`	intended_convention	Real denominator (N : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
HasStationaryDriftIn	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
HasStationaryDriftIn	nat_div	/ 4	intended_convention	Natural-number division / 4 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
InA	nat_mod	% 30	intended_convention	Natural-number remainder % 30 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
InABC15	nat_mod	% 15	intended_convention	Natural-number remainder % 15 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
InABC15	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
InB	nat_mod	% 30	intended_convention	Natural-number remainder % 30 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
InC	nat_mod	% 30	intended_convention	Natural-number remainder % 30 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
IsCertifiedEscape	real_log_rpow	^ [	intended_convention	Real log/rpow ^ [is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
IsTrapStretch	real_log_rpow	^ [	intended_convention	Real log/rpow ^ [is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
IsZeroStretch	real_log_rpow	^ [	intended_convention	Real log/rpow ^ [is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
N1	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
N5	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
OddCycle	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
P2MOp	real_log_rpow	^ (-(v + 1 : ℤ))	intended_convention	Real log/power ^ (-(v + 1 : ℤ)) evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
ReachesOneDigitOne	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
SurvRes	nat_sub	- 1	intended_convention	Natural-number subtraction - 1 is guarded by its domain inequality or intentionally saturates at 0 (e.g., max(0, v_2(m) - 1) or shift step).
T	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
T	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
`T_C`	`nat_div`	`/ 2`	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 | b, or 2 | b) or intended binary right-shift.
`T_C`	`nat_mod`	`% 2`	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
`T_b`	`nat_div`	`/ 2`	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.
`T_b`	`nat_mod`	`% 2`	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`T_b`	`nat_mod`	`% b`	intended_convention	Natural-number remainder % b operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`TypeId`	`nat_div`	`/ digitVal`	intended_convention	Natural-number division / digitVal occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.
`a1`	`classical_choose`	`Classical.choose`	intended_convention	Classical.choose selects the unique positive root of a_1 + a_1^4 = 1 in (0, 1) witnessed by exists_unique_a1.
`aStar`	`classical_choose`	`Classical.choose`	proved_guard (exists_unique_aStar)	Classical.choose selects the unique root in (0, 1) certified by exists_unique_aStar (shown verbatim in Box 1).
`affAB`	`nat_mod`	`% 2`	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`affAB`	`nat_mod`	`% b`	intended_convention	Natural-number remainder % b operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`allOddBases1999`	`nat_mod`	`% 2`	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`baseDigit`	`nat_div`	`/ b`	intended_convention	Natural-number division / b occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.
`baseDigit`	`nat_mod`	`% b`	intended_convention	Natural-number remainder % b operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
`beta`	`real_div`	`aStar`	proved_guard (aStar_spec)	aStar_spec certifies 0 < aStar < 1 so 1 / aStar is strictly positive and well-defined.
`beta`	`real_log_rpow`	`Real.logb`	intended_convention	Real log/power Real.logb evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
`childType`	`nat_div`	`/ d`	intended_convention	Natural-number division / d occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.
`childType`	`nat_mod`	`% Q`	intended_convention	Natural-number remainder % Q operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
childType	nat_sub	- 1	intended_convention	Natural-number subtraction - 1 is guarded by its domain inequality or intentionally saturates at 0 (e.g., max(0, v_2(m) - 1) or shift step).
codingMap	nat_mod	% 5	intended_convention	Natural-number remainder % 5 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
cor_basin_separation_tendsto_zero_unconditional	real_div	(N5 X : ℝ)	intended_convention	Strict positivity 0 < N5(X) is derived inline from thm_basin5 lower bound at BasinFive.lean:953.
cor_basin_separation_tendsto_zero_unconditional	real_div	(N5All X : ℝ)	intended_convention	N5(X) <= N5All(X) holds inline inside cor_basin_separation_all by Finset.card_le_card.
cor_drift_growth	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
cor_drift_growth	real_log_rpow	Real.log	intended_convention	Real log/power Real.log evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
cor_suffix_dimension	nat_mod	% 2	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
cor_suffix_dimension	real_div	Real.log	intended_convention	Real denominator Real.log is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
cor_suffix_dimension	real_div	δ	intended_convention	Real denominator δ is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
cor_suffix_dimension	real_log_rpow	Real.log	intended_convention	Real log/rpow Real.log is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
countFrac	real_div	(N : ℝ)	intended_convention	Real denominator (N : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
cycle_lattice_box	real_div	(2 * (X : ℝ))	intended_convention	Real denominator (2 * (X : ℝ)) is strictly positive on theorem domain (N >= N_0 >= 1, X >= 2, or o = 15 mod 30 >= 15).
cycle_lattice_box	real_log_rpow	Real.logb	intended_convention	Real log/power Real.logb evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
cylStep	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.
cylStep	nat_mod	% 2	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
cylStep	nat_mod	% b	intended_convention	Natural-number remainder % b operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
cylStep	nat_sub	- 1	intended_convention	Natural-number subtraction - 1 is guarded by its domain inequality or intentionally saturates at 0 (e.g., max(0, v_2(m) - 1) or shift step).
d	nat_mod	% 10	intended_convention	Natural-number remainder % 10 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
drift2M	real_log_rpow	Real.log	intended_convention	Real log/power Real.log evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
drift4k	real_log_rpow	Real.log	intended_convention	Real log/power Real.log evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
f_b	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 | b, or 2 | b) or intended binary right-shift.
f_b	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
f_b	nat_mod	% b	intended_convention	Natural-number remainder % b on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
finNext	nat_mod	% m	intended_convention	Natural-number remainder % m on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
gamma	real_div	a1	intended_convention	Real denominator a1 is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
gamma	real_log_rpow	Real.logb	intended_convention	Real log/rpow Real.logb is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
lem_base17_runaway	nat_mod	% 17	intended_convention	Natural-number remainder % 17 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
lem_base17_runaway	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
lem_basin1_moves_classify	nat_mod	% 15	intended_convention	Natural-number remainder % 15 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
lem_basin1_moves_classify	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
lem_basin1_moves_classify	nat_mod	% 25	intended_convention	Natural-number remainder % 25 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
lem_basin1_moves_classify	nat_mod	% 4	intended_convention	Natural-number remainder % 4 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
lem_basin1_moves_classify	nat_mod	% 7	intended_convention	Natural-number remainder % 7 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
lem_weighted_tree3	nat_div	/ digitVal	intended_convention	Natural-number division / digitVal occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.
lem_weighted_tree3	nat_mod	% (Q / digitVal t.1)	intended_convention	Natural-number remainder % (Q / digit Val t.1) operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
lem_weighted_tree3	nat_mod	% 2	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
lem_weighted_tree3	real_div	(o : ℝ)	intended_convention	Real denominator (o : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
lem_weighted_tree3	real_log_rpow	^ (mv.2.val + 1)	intended_convention	Real log/rpow ^ (mv.2.val + 1) is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
lem_weighted_tree3	real_log_rpow	^ s	intended_convention	Real log/rpow ^ s is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
lowerBoxDim	real_div	Real.log	intended_convention	Real denominator Real.log is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
lowerBoxDim	real_div	δ	intended_convention	Real denominator δ is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
lowerBoxDim	real_log_rpow	Real.log	intended_convention	Real log/rpow Real.log is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
next4kA	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
next4kA	nat_mod	% (4 * k)	intended_convention	Natural-number remainder % (4 * k) on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
next4kB	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
next4kB	nat_mod	% (4 * k)	intended_convention	Natural-number remainder % (4 * k) on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
oddRunMass	real_div	(b : ℝ)	intended_convention	Real denominator (b : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
oddRunProb	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.
oddRunProb	nat_mod	% b	intended_convention	Natural-number remainder % b operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
oddStretchOutcome	classical_choose	Classical.choose	intended_convention	Classical.choose extracts the unique target state of a zero-stretch transition (uniqueness proved in oddStretchOutcome_eq_toState_iff).
oddTrapMass	real_div	(b : ℝ)	intended_convention	Real denominator (b : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
oddTrapProb	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
oddTrapProb	nat_mod	% b	intended_convention	Natural-number remainder % b operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
p_k	nat_sub	- 1	intended_convention	Natural-number subtraction - 1 is guarded by its domain inequality or intentionally saturates at 0 (e.g., max(0, v_2(m) - 1) or shift step).
p_k	real_div	((b * 2 ^ (k - 1) : ℕ) : ℝ)	intended_convention	Real denominator ((b * 2 ^ (k - 1) : ℕ) : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
parityVec	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
parityVec	real_log_rpow	^ [	intended_convention	Function.iterate notation T^[j] on Nat; not real exponentiation.
predCondBool	nat_mod	% 25	intended_convention	Natural-number remainder % 25 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
predCondBool	nat_mod	% 3	intended_convention	Natural-number remainder % 3 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
predCondBool	nat_mod	% 5	intended_convention	Natural-number remainder % 5 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
predCondBool	nat_mod	% 7	intended_convention	Natural-number remainder % 7 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
predCondBool	nat_mod	% 9	intended_convention	Natural-number remainder % 9 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
prop_near_zero_band	nat_div	/ 4	intended_convention	Natural-number division / 4 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
prop_near_zero_band	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
prop_no_residue_size_bound	nat_mod	% 30	intended_convention	Natural-number remainder % 30 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
prop_no_residue_size_bound	nat_mod	% Q	intended_convention	Natural-number remainder % Q on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
prop_no_residue_size_bound	real_div	(o : ℝ)	intended_convention	Real denominator (o : ℝ) is strictly positive on theorem domain (N >= N_0 >= 1, X >= 2, or o = 15 mod 30 >= 15).
prop_no_residue_size_bound	real_log_rpow	^ c	intended_convention	Real log/power ^ c evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
psiMap	nat_div	/ 30	intended_convention	Natural-number division / 30 occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.
psiMap	nat_mod	% 2	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
psiMap	real_log_rpow	^ [	intended_convention	Real log/rpow ^ [is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
revExp	real_div	(b : ℝ)	intended_convention	Real denominator (b : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
revExp	real_log_rpow	^ (j + 1)	intended_convention	Real log/rpow ^ (j + 1) is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
splitTwo	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, divisor factor, or guarded positive modulus) or intended floor/binary right-shift.
splitTwo	nat_mod	% 2	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
suffixDimVal	real_div	Real.log	intended_convention	Real denominator Real.log is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
suffixDimVal	real_log_rpow	Real.log	intended_convention	Real log/rpow Real.log is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
suffixH	nat_sub	- 1	intended_convention	Natural-number subtraction - 1 is guarded by its domain inequality or intentionally saturates at 0 (e.g., max(0, v_2(m) - 1) or shift step).
suffixLambda	real_div	aStar	intended_convention	Real denominator aStar is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
thm_basin1_upper	real_log_rpow	^ beta	intended_convention	Real log/power ^ beta evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
thm_basin5_tight	real_log_rpow	^ ((65 : ℝ) / 100)	intended_convention	Real log/power ^ ((65 : ℝ) / 100) evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
thm_basin_sharp	real_div	((2269527 : ℝ) * (3 : ℝ) ^ ((13 : ℝ) / 20))	intended_convention	Real denominator ((2269527 : ℝ) * (3 : ℝ) ^ ((13 : ℝ) / 20)) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
thm_basin_sharp	real_div	(N5 X : ℝ)	intended_convention	Real denominator (N5 X : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
thm_basin_sharp	real_div	Real.log	intended_convention	Real denominator Real.log is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
thm_basin_sharp	real_log_rpow	Real.log	intended_convention	Real log/rpow Real.log is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
thm_basin_sharp	real_log_rpow	^ ((13 : ℝ) / 20)	intended_convention	Real log/rpow ^ ((13 : ℝ) / 20) is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
thm_basin_sharp	real_log_rpow	^ ((399 : ℝ) / 800)	intended_convention	Real log/rpow ^ ((399 : ℝ) / 800) is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
thm_basin_sharp	real_log_rpow	^ (-(255 : ℝ) / 10000)	intended_convention	Real log/rpow ^ (-(255 : ℝ) / 10000) is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
thm_basin_sharp	real_log_rpow	^ betaPrime	intended_convention	Real log/rpow ^ betaPrime is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
thm_cantor	real_div	(X : ℝ)	intended_convention	Real denominator (X : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
thm_cantor	real_log_rpow	Real.logb	intended_convention	Real log/rpow Real.logb is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).
thm_cantor	real_log_rpow	^ beta	intended_convention	Real log/rpow ^ beta is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffix Lambda > 1, or delta in (0,1)).
thm_even_bases	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
thm_even_stopping	nat_mod	% 2	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
thm_odd_densities_disjoint	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
thm_odd_survey	nat_mod	% 2	intended_convention	Natural-number remainder % 2 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
thm_odd_survey	nat_sub	- ε	intended_convention	Natural-number subtraction - ε is guarded by its domain inequality or intentionally saturates at 0 (e.g., max(0, v_2(m) - 1) or shift step).
thm_odd_survey	real_div	(b : ℝ)	intended_convention	Real denominator (b : ℝ) is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
thm_odd_trapdoor_rootless	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
thm_odd_trapdoor_rootless	nat_mod	% 4	intended_convention	Natural-number remainder % 4 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
thm_odd_trapdoor_rootless	nat_mod	% b	intended_convention	Natural-number remainder % b on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
thm_residue_law	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
thm_suffix_class	nat_mod	% 30	intended_convention	Natural-number remainder % 30 operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
thm_suffix_class	nat_mod	% m	intended_convention	Natural-number remainder % m operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
thm_suffix_class	nat_mod	% suffixP	intended_convention	Natural-number remainder % suffixP operates on a strictly positive modulus (fixed positive constant or guarded hypothesis b >= 2, m >= 1, Q >= 1).
thm_terras_bound	real_div	(N : ℝ)	intended_convention	Real denominator (N : ℝ) is strictly positive on theorem domain (N >= N_0 >= 1, X >= 2, or o = 15 mod 30 >= 15).
thm_terras_bound	real_log_rpow	^ k	intended_convention	Real log/power ^ k evaluated strictly on positive arguments/bases (X >= 1, X >= 3, b >= 2, odd digits >= 1, or 1/aStar > 1).
thm_terras_density_one	real_div	(N : ℝ)	intended_convention	Real denominator (N : ℝ) is strictly positive on theorem domain (N >= N_0 >= 1, X >= 2, or o = 15 mod 30 >= 15).
trailingOnes	nat_div	/ 2	intended_convention	Natural-number division / 2 occurs on exact-divisible branch (even numerator, 4 ∣ b, or 2 ∣ b) or intended binary right-shift.
trailingOnes	nat_mod	% 2	intended_convention	Natural-number remainder % 2 on strictly positive modulus (fixed constant or guarded hypothesis b >= 2, Q >= 1, M >= 1, m > 0).
upperBoxDim	real_div	Real.log	intended_convention	Real denominator Real.log is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
upperBoxDim	real_div	δ	intended_convention	Real denominator δ is strictly positive on the theorem domain (N >= 1, X >= 1, o >= 1, b >= 2, log b > 0, or delta > 0).
upperBoxDim	real_log_rpow	Real.log	intended_convention	Real log/rpow Real.log is evaluated strictly on positive arguments/bases (X >= 1, b >= 2, u >= 1, a_1/a* in (0,1), suffixLambda > 1, or delta in (0,1)).

[a] Covers the specified Lean 4 hazard set (Nat subtraction/division/mod, totalized real/rational division, Real.log/rpow, Finset.range, Classical.choose, local instances); no finite checklist certifies all semantics.

Table 4. Per-Claim Reader's Trust Boundary Ledger (G11 Denominators Derived Mechanically).[a]

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Prop. 1: Base-4 attraction to {1,3} <-> classical Collatz conjecture	Lean Kernel	Ouroboros.prop_base4_iff_collatz	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel over Box 1 definitions f_b, T_C, Reaches.
Conj. 2: The Ouroboros Conjecture (every n >= 1 reaches 1 or 5 under f_10)	Open	Open conjecture	Lean: none; fidelity: n/a	Explicitly open conjecture (defined formally in Lean as Ouroboros.Ouroboros Conjecture; excluded from G11 denominator).
Thm. 3: Exponential base-10 non-descent upper bound 1.124953*(0.996423)^k + eps	Lean Kernel	Ouroboros.thm_terras_bound	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for all integers k >= 1 and real eps > 0.
Cor. 4: Density-one descent under f_10 (no-descent fraction <= eps)	Lean Kernel	Ouroboros.thm_terras_density_one	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every real eps > 0.
Thm. 5: Elementary quartic upper bound N_1(X) <= (244/100) X^beta (X >= 1)	Lean Kernel	Ouroboros.thm_basin1_upper	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every integer X >= 1 (a* unique root of a + 2a^4 = 1).
Thm. 6: Pure-kernel lower bound N_5(X) >= (605/10^6) X^(13/20) (X >= 3)	Lean Kernel	Ouroboros.thm_basin5_tight	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every integer X >= 3 (14 Quad Tree rfl shards mod 14,175).
Cor. 7: Unconditional basin separation N_1(X)/N_5(X) and N_1^all(X)/N_5^all(X) -> 0	Lean Kernel	Ouroboros.cor_basin_separation_tendsto_zero_unconditional	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every real eps > 0 (0 < N_5(X) <= N_5^all(X) derived inline).

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Thm. 8: Sharpened bounds 0.0145 X^0.49875 <= N_1(X) <= C X^0.6245 and N_5(X) > 0.02134 X^0.65	Lean Kernel	`Ouroboros.thm_basin_sharp`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel (Basin OneSharp, BasinOneUpper Sharp, WeightedCharging, Basin SharpAll; compact QuadTree rfl shards mod 643,125 and 14,175).
Thm. 9: Only (1), (5,13) for m <= 8; <= 10^15 sweep; K >= 34,905, m >= 15,496, K+m >= 53,185	Cert Only	recompute.py (97c49103646a) [archived; replay pending]	External cert only; fidelity: n/a	External Python/C++ m <= 8 search (7,811,375 words), archived [7,10^15) C/GMP sweep (summary table inputs/windows.csv validated in recompute.py), and exact 4D lattice screen (19 survivors).
Thm. 10: Even-base nonpositive stationary drift <-> b in {2, 4, 6, 10}	Lean Kernel	`Ouroboros.thm_even_bases`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for all even b >= 2 (EvenBasesCert rfl shards for odd 7 <= M <= 103).
Thm. 11: Even-base non-descent dichotomy p_infty(b) = 0 <-> b in {2, 4, 6, 10}	Lean Kernel	`Ouroboros.thm_even_stopping`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every even integer base b >= 2 (EvenStopping.lean).
Thm. 12: Rootless odd-base trapdoor law (prime p = 3 mod 4 dividing odd b => Z_b = b Z_{>=1})	Lean Kernel	`Ouroboros.thm_odd_trapdoor_rootless`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for any odd b divisible by a prime p = 3 (mod 4).
Thm. 13: Odd-base absorption survey resolving 905 of 999 odd bases 3 <= b <= 1,999	Lean Kernel	`Ouroboros.thm_odd_survey`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for all odd bases b >= 3 and the 905-of-999 resolved census 3 <= b <= 1,999 (OddSurvey.lean + 32 rfl shards).
Lem. 14: Base-10 parity coding bijection modulo 5 * 2^k	Lean Kernel	`Ouroboros.lemma_coding_bijective`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every natural number k.
Prop. 15: Every odd start reaching 1 lies in {1, 15, 17} (mod 30)	Lean Kernel	`Ouroboros.thm_residue_law`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for any odd n reaching 1 under f_10.
Lem. 16: Exhaustive 5-row odd-to-odd move table on A cup B cup C (excluding digits 3, 9)	Lean Kernel	`Ouroboros.lem_basin1_moves_classify`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for any o in A cup B cup C, v >= 1, odd m with d(m)*m + 1 = 2^v * o.
Prop. 17: Digit-1 reachability to 1 <-> all maximal binary 1-runs of (n-1)/2 have length 0 mod 4	Lean Kernel	`Ouroboros.prop_binary_test`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every natural number u.
Thm. 18: Exact 2-adic Cantor closure B_1^bar, Hausdorff/box dimension beta = 0.62638214, a_1^9 X^gamma <= N_1(X) <= X^beta	Lean Kernel	`Ouroboros.thm_cantor`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel (Cantor Closure.lean: unique roots a_1, a*, exact Z_2 closure, isometry, Hausdorff measure/dimension beta, and real bounds X >= 1).
Prop. 19: Contracting double-move obstruction (no finite residue vector bounds A cup B cup C descendants)	Lean Kernel	`Ouroboros.prop_no_residue_size_bound`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every Q >= 1, c > 0, and weight vector w : Fin Q -> R.
Lem. 20: Individual child-weight charging L H(o, X) >= (X/o)^s w_t	Lean Kernel	`Ouroboros.lem_weighted_tree3`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for any modulus 1575 | Q, Vmax, real s > 0, L > 0, nonnegative weights w, odd o reaching 3, and integer X >= o (WeightedCharging.lean).
Lem. 21: Sharp odd-cycle halving budget n_1 + 2n_3 + 4n_7 + 3n_9 + 4 t_51 <= K and 2n_1 + 3a + e + 7c <= 2K	Lean Kernel	`Ouroboros.lem_halving_budget_sharp` + `Ouroboros.lem_halving_budget`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every OddCycle m via the pair Ouroboros.lem_halving_budget_sharp (CycleBounds.lean:721) and Ouroboros.lem_halving_budget (CycleBounds.lean:790-796); hypothetical if non-trivial, realized by (1) and (5,13).

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Prop. 22: Every m = 3 odd cycle of f_10 is the trivial repetition (1,1,1)	Lean Kernel	`Ouroboros.cycle_m3_unique`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every OddCycle 3 (unconditional, zero size cutoff).
Lem. 23: Four-dimensional integer lattice-box inclusion for odd cycles with x_i >= X and K <= L	Lean Kernel	`Ouroboros.cycle_lattice_box`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for any OddCycle m with X >= 2, x_i >= X, K <= L, 2L <= X, S >= 1, and floor approximations lam_p.
Thm. 24: Any further odd cycle >= 10^15 with K > 64,000 satisfies m >= 20,190 and K+m >= 84,191	Lean Kernel	`Ouroboros.thm_cycle_tail_64000`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for any hypothetical OddCycle m with x_i >= 10^15 and K > 64,000.
Lem. 25: Unbounded even-base drift lower bound Delta(pi) >= (1/2) log(b+2) - (7/2) log 2	Lean Kernel	`Ouroboros.cor_drift_growth`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every even integer base b >= 2.
Prop. 26: Even-base near-zero drift band |Delta| <= 1/4 <-> {4,8,10,14}; [-1/4,0) <-> {4,10}	Lean Kernel	`Ouroboros.prop_near_zero_band`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every even integer base b >= 2.
Lem. 27: Certified base-17 escaping progression (odd n = 16 mod 17 tends to +infty)	Lean Kernel	`Ouroboros.lem_base17_runaway`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every odd integer n >= 1 with n mod 17 = 16.
Lem. 28: Odd-base trapping/divergence disjointness (Z_b subset R_b and E_b cap R_b = empty)	Lean Kernel	`Ouroboros.thm_odd_densities_disjoint`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every odd base b >= 3.
Thm. 29: Every class a mod m meets B_5 infinitely often; finite halving test governs B_1	Lean Kernel	`Ouroboros.thm_suffix_class`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every congruence class a mod m with integer m >= 1 (Suffix Class.lean).
Cor. 30: Exact written-base ending dimension (log u + t log lambda)/log b (0.62638214 binary, 0.88752982 decimal, 1 base-7)	Lean Kernel	`Ouroboros.cor_suffix_dimension`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel for every integer base b = 2^t u >= 2 with u odd (SuffixDimension.lean).
Conj. 31: Existence of lim log N_1(X)/log X and equality with branching-model root alpha ~ 0.4994477848	Open	Open conjecture	Lean: none; fidelity: n/a	Explicitly open conjecture (excluded from G11 denominator).

[a]Allowed pure-kernel axioms: `propext`, `Classical.choice`, `Quot.sound`. Actual axioms printed above are read directly from verified outputs. Provenance nonce: 230c2b2eea10 | Lean mode: committed_receipt | Closure SHA-256: 3cea85fa4d83 | Git anchor: 797e3413d11d | Axioms SHA-256: 9c024e86be9b | Fidelity SHA-256: a7f36082f9e1.

9. OPEN PROBLEMS, REPRODUCIBILITY, AND PRIOR WORK

Alongside Conjecture 2 (universal base-10 attraction to $\{1, 5\}$), three further concrete open problems tie together the base-b picture:

Conjecture 31 (Thin-basin counting exponent). *The limit* $\lim_{X\to\infty} \log N_1(X)/\log X$ *exists and equals the unique real root* $\alpha \simeq 0.4994477848\ldots$ (*strictly below* $1/2$) *of the branching-model equation* (12).

- *Individual divergence in even bases* (*base* 8)*:* Theorem 11 proves $p_\infty(8) \geq 0.0861$, yet an explicit integer orbit diverging under f_8—analogous to $34q + 33$ in base 17 (Lemma 27)—remains unknown. Can any concrete positive integer be proved to diverge in base 8?
- *Universal capture in odd bases* (*base* 25)*:* Theorem 13 certifies $\tau_{25} = 1$, so almost every start reaches 1 in base 25, yet unlike base 5 no proof rules out a density-zero exceptional cycle or escape. Does every positive integer reach 1 under f_{25}?

Standalone verification and Lean 4 formalization. Every computer-assisted result separates *search* from *certification*: the offline archive `checker.zip` and driver `recompute.py` run under Python 3.10+ with no external libraries (`python3 recompute.py`) and verify by exact integer and rational arithmetic:

- both verbatim Python boxes printed in Sections 1 and 6, the base-4 conjugacy through $y < 10{,}000$, and the Cramér–Lundberg root $\theta_* \simeq 0.2189$ of $\rho(M_{\theta_*}) = 1$;
- exact even-base drifts through $b = 208$, all 14 tilted stopping-time vectors, and 78 complete integer-block comparisons of $p_k(b)$;

- all 999 odd-base cylinder surveys through $b = 1{,}999$ and exact rational absorption solves;
- all 128,625-row decimal basin-1 vectors (104,125 positive lower weights, 2,038,400 lower moves, 3,057,600 moves plus tail per upper vector) and all 70,875 rows (81,072 moves, $L = 2{,}269{,}527$) of basin-5;
- both short-period enumerations through $m = 8$ (Python and C++, 7,811,375 words), the 4D lattice screen (19 survivors), and the bundled 10,000-window sweep summary table `inputs/windows.csv` (boundaries, start counts, zero-fallback and exact-record flags, and global record orbits; `-r` replays all 20,000 window records, without re-executing the full 10^{15} C sweep);
- Figures 5, 8, 9, 11, and 16, the four-state automaton, 45,756-class quotient checks, and the Lean 4 axiom SHA-256.

The exhaustive $[7, 10^{15})$ C/GMP sweep sources, chunk logs, coverage receipts, and independent spot-check logs are archived in the repository at `docs/confirmed-strong-and-novel/last-digit-collatz/sweep/`; aggregated 2^{72}-capped inverse-tree counts are archived in `figures/capped_basin_scaling.csv` (from `docs/confirmed-strong-and-novel/last-digit-collatz/basin-counts.csv`), while the raw binary shards are archived externally. The `Ouroboros` Lean 4 + Mathlib library (revision `797e3413d`, Lean `v4.35.0` + Mathlib `v4.35.0`) comprises 125 `.lean` source files under `lean/Ouroboros/` (124 library modules plus `Verify.lean`, 43,121 lines total; 19 files import Mathlib, while the 104 finite certificate shards discharge via `Init`-only `rfl`) with 1,833 audited public declarations (0 `sorry`, 0 `native_decide`, standard axioms `propext`, `Classical.choice`, `Quot.sound` only; three local `set_option maxRecDepth 1024` annotations at `EvenBasesCert.lean` lines 205, 207, and 209 adjust elaborator recursion depth only and do not affect Lean kernel soundness), formalizing 15 of 16 main results (94%), 14 of 15 abstract claims (93%), and 28 of 29 proved claims overall (97%; 1 external certificate only, Theorem 9, and 0 unformalized written proofs; 2 open conjectures excluded from denominators). Section 8 records the exact per-claim G11 breakdown, verbatim definitions, theorem signatures, type-trap dispositions, and cryptographic provenance hashes.

Prior work, authorship, and use of AI. Terras introduced parity-word coding for density-one stopping times in $3x+1$ [1]; Matthews and Watts extended congruence-cylinder coding to generalized Syracuse maps [2]; and Leigh developed Markov residue chains when multipliers share factors with the modulus [3]. The affine change of variables linking f_4 to Collatz parallels Yolcu, Aaronson, and Heule's analysis of Farkas's map [8, 11] [to check], and the modular affine-translation mechanism behind (5) parallels Monks, Monks, Monks, and Monks's group-action analysis of arithmetic progressions in the $3x + 1$ graph [10] [to check]. For base 10, Krasikov and Lagarias's difference-inequality framework motivates our backward tree bounds [6], Simons's logarithmic cycle obstructions parallel our four-dimensional lattice screen [9], Asmussen's Markovian risk theory [4] [to check] underpins the Cramér–Lundberg excursion heuristic in Section 5, and Santi [12] previously recorded the six-cycle (2) in his alternating-3/5 map on Mathematics Stack Exchange (question 5113307, 10 December 2025). The last-digit family f_b and the base-10 two-loop conjecture were not found in the searches logged (Guy's *Unsolved Problems in Number Theory* [7] [to check] in print and Wirsching's monograph [5] [to check] remain to be searched).

Use of AI. Tyler B. Hughes designed and directed the AI-assisted search that identified f_{10} among 16 digit-driven candidates; AI research agents built and directed by Tyler B. Hughes developed the proofs, Lean 4 formalization, computations, figures, and draft exposition. No human mathematician has yet refereed the proof.

Drafted: November 2026 **Updated:** November 24, 2026

REFERENCES

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

(2) K. R. Matthews and A. M. Watts, A generalization of Hasse's generalization of the Syracuse algorithm, *Acta Arith.* **43** (1984), 167–175. doi:10.4064/aa-43-2-167-175.

(3) G. M. Leigh, A Markov process underlying the generalized Syracuse algorithm, *Acta Arith.* **46** (1986), 125–143. doi:10.4064/aa-46-2-125-143.

(4) S. Asmussen, Risk theory in a Markovian environment, *Scandinavian Actuarial Journal* **1989** (1989), 69–100. doi:10.1080/03461238.1989.10413858. [to check]

(5) G. J. Wirsching, *The Dynamical System Generated by the* $3n + 1$ *Function*, Lecture Notes in Mathematics **1681**, Springer, Berlin, 1998. doi:10.1007/BFb0095985. [to check]

(6) I. Krasikov and J. C. Lagarias, Bounds for the $3x + 1$ problem using difference inequalities, *Acta Arith.* **109** (2003), 237–258. doi:10.4064/aa109-3-4.

(7) R. K. Guy, *Unsolved Problems in Number Theory*, 3rd ed., Problem Books in Mathematics, Springer, New York, 2004. doi:10.1007/978-0-387-26677-0. [to check]

(8) H. M. Farkas, Variants of the $3N + 1$ conjecture and multiplicative semigroups, in *Geometry, Spectral Theory, Groups, and Dynamics*, Contemp. Math. **387**, Amer. Math. Soc., Providence, RI, 2005, pp. 121–127. doi:10.1090/conm/387/07238. [to check]

(9) J. L. Simons, On the (non-)existence of m-cycles for generalized Syracuse sequences, *Acta Arith.* **131** (2008), 217–254. doi:10.4064/aa131-3-2.

(10) K. Monks, K. G. Monks, K. M. Monks, and M. Monks, Strongly sufficient sets and the distribution of arithmetic sequences in the $3x + 1$ graph, *Discrete Math.* **313** (2013), 468–489. doi:10.1016/j.disc.2012.11.019. [to check]

(11) E. Yolcu, S. Aaronson, and M. J. H. Heule, An automated approach to the Collatz conjecture, *Journal of Automated Reasoning* **67** (2023), article 15. doi:10.1007/s10817-022-09658-8. [to check]

(12) A. Santi, A Collatz-type dynamical system with alternating multipliers (3 and 5): three attractors and huge excursions, Mathematics Stack Exchange, question 5113307, 10 December 2025, math.stackexchange.com/q/5113307.