The Exact Price of Mergeability for Capacity-Parametric Counter Replay

Victor Bona
Independent Researcher, Blumenau, Brazil
victor.bona@hotmail.com

September 2026

Abstract

A saturating counter log is easy to replay once its capacity is fixed. Retaining enough information to choose the capacity later is a different problem. We characterize this problem exactly for integer capacities 0 through N. Final-value queries from zero have 2^N observational equivalence classes and require exactly N logical bits. Requiring independently summarized chunks to merge raises the number of necessary states to $N2^N + 1$. This bound is attained by a canonical, associative summary. For $N \geq 1$, its minimum fixed-width information content is $N + \lfloor \log_2 N \rfloor + 1$ bits. The lower bound needs only correctness of answers after one binary merge; it does not assume associative merging, canonical encodings, or a particular algorithm. All semantic, reachability, counting, and lower-bound results are checked in Lean 4 without proof holes or additional axioms. The precise finite-state count and software-interface separation are candidate contributions; historical novelty remains unestablished because of substantial related work on saturation and hysteresis.

1 Problem and result

Let $w = [d_1, \ldots, d_m]$ be a finite log of signed integer updates. At capacity C, replay applies

$$\mathrm{step}(C, d, x) = \min\big(C, \max(0, x + d)\big)$$

in log order. Write $F_w(C, x)$ for the final result, with $0 \leq x \leq C$. The capacity remains constant during one replay, but is selected after summarization from $\{0, \ldots, N\}$. The upper horizon N is known to the summarizer. Event order and the event sequence are identical across these counterfactual replays.

The first interface answers $f_w(C) = F_w(C, 0)$. The second also combines independently summarized logs so that querying their combined summary returns $f_{uv}(C)$. We count logical summary states, without restricting computation time.

Theorem 1.1 (Exact price of merging)**.** *For every integer $N \geq 0$:*

1. *The zero-start response profiles have exactly 2^N equivalence classes. An online summary with N binary entries is exact and information-optimal.*
2. *The families $F_w(C, x)$, for $0 \leq x \leq C \leq N$, have exactly*

$$K_N = N2^N + 1$$

equivalence classes. A canonical summary attains this count and supports exact associative merging.

3. *Every deterministic summary supporting exact answers after one binary merge has at least K_N possible encoder outputs, even if its public queries always start from zero.*

Consequently, for $N \geq 1$ the exact minimum fixed-width information content is $N + \lfloor \log_2 N \rfloor + 1$ bits for mergeable summaries, compared with N bits without merging. At $N = 0$ both interfaces need one state, or zero bits. The proof is developed in Sections 2–4.

Horizon N	Zero-start classes	Mergeable classes K_N	Zero-start bits	Mergeable bits
1	2	3	1	2
2	4	9	2	4
4	16	65	4	7
8	256	2,049	8	12
64	2^{64}	$64 \cdot 2^{64} + 1$	64	71

Table 1: The exact logical state counts and fixed-width information requirements.

For example, $w = [+4, -2, +1]$ has final values $[0, 1, 1, 2, 3]$ at capacities 0 through 4. Adjacent differences are $[1, 0, 1, 1]$, which are the four bits of its response profile. Summing its updates would lose this information.

A motivating use is counterfactual quota or bounded-workload replay over partitioned event logs. The model applies only when each event has precisely the stated clamping semantics. It does not model admission rejection, retries, capacity changes mid-replay, or events whose existence depends on earlier decisions.

2 Binary capacity memory

Lemma 2.1 (One-bit derivative)**.** *For every log w and $C \geq 0$,*

$$f_w(0) = 0, \qquad f_w(C) \leq f_w(C+1) \leq f_w(C) + 1.$$

Proof. Couple two runs whose capacities differ by one. Their initial states both equal zero. A shared update preserves their ordering and a state difference of at most one, because each run clamps at its corresponding bounds. Induction over updates proves the claim. Thus each $b_C = f_w(C) - f_w(C-1)$ is either zero or one, and $f_w(C)$ is the sum of the first C bits. □

Lemma 2.2 (Every signature is reachable)**.** *Every N-bit string b occurs as a capacity derivative. A witnessing log has at most N updates, each with magnitude at most N. Every unbounded prefix state of that witness stays in $[0, N]$. Its response is constant above capacity N.*

Proof. Induct on N, reading the highest-capacity bit first. If that bit is zero, compile the lower bits. If it is one, first add N, then run the sign-reversed compilation of the complemented lower bits. For capacities $C \leq N$, the initial addition fills the counter and sign reversal reflects its trajectory about C. For $C \geq N$, the safe-prefix invariant makes the same reflection argument work around N without unintended saturation. The resulting response is the desired prefix sum. The proof maintains both the safe-prefix invariant and the update-count bound. □

The 2^N distinct signatures yield the zero-start lower bound. A matching online update is also verified: adding k changes the first k zero bits into ones, scanning from low capacities; subtracting k changes the first k one bits into zeros. If fewer opposite bits exist, all of them change. Existing equal bits consume no budget. The reference implementation uses lists and takes linear time in N per update; N denotes logical bits, not its heap allocation.

3 A complete normal form across capacities and initial states

A log has a shape (R, A, H), with $0 \le A, H \le R$. One interpretation uses the unbounded partial sums, including the initial zero: if their minimum and maximum are m and M and their final sum is D, then $R = M - m$, $A = -m$, and $H = D - m$. The Lean implementation computes the shape directly by the following recurrence, starting at $(0, 0, 0)$:

$$\begin{aligned} \text{update } +k: &\quad (R, A, H) \longmapsto \big(\max(R, H + k), A, H + k\big), \\ \text{update } -k: &\quad (R, A, H) \longmapsto \big(R + (k - H)^+, A + (k - H)^+, (H - k)^+\big). \end{aligned}$$

Here $t^+ = \max(0, t)$. The partial-sum interpretation is explanatory; the formal theorem uses this recurrence directly.

Lemma 3.1 (Surviving dependence on the initial state)**.** *For every legal C and x,*

$$F_w(C, x) = f_w(C) + \min\big((C - R)^+, (x - A)^+\big).$$

Moreover, $f_w(C) = H$ whenever $C \ge R$.

Proof. The empty log has shape $(0, 0, 0)$. Substitute the claimed formula into one addition or subtraction followed by saturation. An addition can extend the upper excursion; a subtraction can extend the lower excursion and shift A. The displayed recurrence is exactly what makes the resulting expression retain the same form. Induction proves the formula, the stable response above R, and the shape inequalities. □

This separates two disjoint families at horizon N:

- *Collapsed family:* $R \ge N$. Every legal initial state produces the same result. The code contains an arbitrary N-bit response signature.
- *Live family:* $R < N$. The code contains an R-bit response signature and $A \in \{0, \dots, R\}$. For $C > R$ its response signature has already stabilized, while dependence on x survives with width $C - R$.

These codes are complete by Lemma 3.1. They are also all reachable. For a live code with signature b of length R and delay A, use

$$[+(R - A), -R] \mathbin{+\!\!+} \text{compile}(b).$$

At $C \le R$ the two-update prefix resets every starting state to zero. At $C \ge R$ it leaves $\min(C - R, (x - A)^+)$. The safe-prefix property of compile(b) then adds the signature's full prefix sum without further saturation. For a collapsed code, use $[+N, -N] \mathbin{+\!\!+} \text{compile}(b)$.

Finally, the codes are distinct. At capacity N the difference $F(N, N) - F(N, 0)$ is zero for a collapsed code and $N - R$ for a live code, identifying its family and R. Zero-start queries recover

every signature bit. For a live code, a probe immediately above the delay identifies A. These arguments are formalized as semantic extensionality and a witness/encoding round trip.

Counting the disjoint families gives

$$K_N = 2^N + \sum_{R=0}^{N-1} (R+1)2^R = N2^N + 1.$$

The Lean proof counts a duplicate-free finite catalog and proves every log maps into it and every entry is realizable. The count is therefore a semantic quotient cardinality, not merely the size of a chosen data structure.

4 Why merging forces the larger state space

Suppose $\mathrm{enc}(w)$ is a summary, combine is a binary operator, and

$$\mathrm{query}\big(\mathrm{combine}(\mathrm{enc}(u), \mathrm{enc}(v)), C\big)$$

exactly returns $F_{uv}(C, 0)$. To recover $F_w(C, x)$, prepend the one-event log $[+x]$: because $x \le C$, its zero-start result is x. Therefore

$$F_w(C, x) = \mathrm{query}\big(\mathrm{combine}(\mathrm{enc}([+x]), \mathrm{enc}(w)), C\big).$$

If two logs with different full transformation families had the same encoder output, the right-hand side could not distinguish them. Thus every such encoder needs at least K_N states. This reduction assumes neither associativity nor equality between $\mathrm{combine}(\mathrm{enc}(u), \mathrm{enc}(v))$ and $\mathrm{enc}(uv)$; correctness of the merged query alone suffices.

For the upper bound, merge canonical codes by concatenating their witnessing logs and canonicalizing the result. Witness correctness and semantic uniqueness prove that this is exact, respects log concatenation, and is associative on valid codes. Merge preserves order; it is not commutative. This is a proved reference algorithm, not a constant-time merge implementation.

For $N \ge 1$, let $L = \lfloor \log_2 N \rfloor$. Then

$$2^{N+L} \le N2^N < 2^{N+L+1}.$$

Adding one puts K_N strictly above the lower power and at or below the upper power. Thus its minimum fixed-width index needs $N + L + 1$ bits, including when N is itself a power of two. The Lean theorem proves the equivalent threshold $K_N \le 2^b$ if and only if $N + \texttt{N.log2} + 1 \le b$. A finite catalog supplies the usual enumerative-index upper bound; the supplied implementation stores structured codes rather than a physically packed optimal index. This completes the proof of Theorem 1.1.

5 Formal artifact and reproducibility

The artifact uses Lean 4.24.0 and only its bundled standard library. It contains no `sorry`, `admit`, custom axioms, native-decision shortcuts, or external proof solvers. A trust-level-zero audit rechecks imported modules and reports only Lean's standard `propext`, `Classical.choice`, and `Quot.sound` axioms. Table 2 records the principal declarations.

Paper claim	Principal checked declaration
Binary profiles, constructive witnesses	`compile_spec`, `compile_delta_bound`, `signature_exact`
Optimal online zero-start summary	`optimal_summary`, `online_exact`
All-capacity normal form	`describe_spec`
Complete, reachable, distinct codes	`canonical_exact`, `witness_exact`, `Code.ext`
Exact class count and arbitrary-encoder lower bound	`full_optimality`, `full_state_lower_bound`
Exact associative merging	`canonical_append`, `merge_assoc`
Lower bound requiring only correct merged queries	`merge_query_state_lower_bound`, `merge_query_bit_lower_bound`
Attainable mergeable state count	`mergeable_optimality`

Table 2: The connection between the paper's claims and the checked Lean declarations.

Run `./verify.sh` from the artifact directory. It builds the library and executable, runs `lean --trust=0 Audit.lean`, checks proof-source hygiene, and runs deterministic executable checks. The executable checks 3,595 canonical codes for $N = 0$ through 8 against 139,947 replay queries, 716 merge pairs, 757 associativity triples, and 2,000 seeded logs against another 102,555 replay queries. These tests exercise the implementation; the universal guarantees come from the Lean proofs. Evidence logs and source hashes are retained in `evidence/`.

6 Related work and novelty

Clipped-translation summaries and parallel saturated accumulation are established by Papadantonakis et al. [1]. The fixed-capacity semigroup's plateau/ramp/plateau normal form and cardinality also appear in Zubairu et al. [2]. We do not claim those results as new.

Threshold-dependent counter memory is closely connected to classical hysteresis. Brokate, Dreßler, and Krejčí describe memory curves and reduced excursion histories [3]; Zhou et al. explicitly use binary PI-model memory states [4]. Recenter a capacity-C counter using $r = C/2$ and $s = x - r$: its update becomes $s \mapsto \min(r, \max(-r, s + d))$, the stop-operator form. This algebraic connection is our derivation and is a substantial novelty boundary. Xing studies hysteresis history semigroups and erasing relations [5].

Our candidate contribution is the exact finite-horizon quotient $K_N = N2^N + 1$, its constructive mechanization, and the sharp separation between sequential zero-start replay and independent merging under a precise software interface. The literature search did not locate this exact count or separation, but it cannot establish priority. Specialist review of hysteresis semigroups and streaming lower bounds is needed before a novelty claim or publication submission.

7 Limitations and next steps

The theorem is exact and deterministic. It does not bound approximate or probabilistically failing summaries, average-case or variable-length compression, external storage, computation time, or logs restricted to a special workload distribution. It assumes signed integer updates, a fixed capacity within each replay, and a known horizon N. The reference canonicalizer may do quadratic work in N and the merge reconstructs witness logs; no measured speedup or optimized

packed-memory implementation is claimed. A practical next step is a verified direct merge and packed rank/select representation, evaluated against repeated replay and existing saturation summaries.

Acknowledgments

The research, code, and Lean formalization were developed in an AI-assisted workflow directed by the author, using Codex (OpenAI). The formal results and executable checks are independently reproducible through the artifact described in Section 5; historical novelty remains subject to the qualifications in Section 6.

References

[1] K. Papadantonakis, N. Kapre, S. Chan, and A. DeHon, Pipelining saturated accumulation, *IEEE Transactions on Computers* (2009). doi:10.1109/TC.2008.110. Author manuscript.

[2] M. M. Zubairu, A. Umar, and M. J. Aliyu, On the combinatorial and rank properties of certain subsemigroups of full contractions of a finite chain, arXiv:2205.00438, 2022. See Theorem 2 for the fixed-chain count; the all-capacity count here concerns a different quotient.

[3] M. Brokate, K. Dreßler, and P. Krejčí, Rainflow counting and energy dissipation for hysteresis models in elastoplasticity, *European Journal of Applied Mathematics* (1996). Author manuscript.

[4] C. Zhou, C. Feng, Y. N. Aye, and W. T. Ang, A digitized representation of the modified Prandtl–Ishlinskii hysteresis model for modeling and compensating piezoelectric actuator hysteresis, *Micromachines* **12**(8) (2021), 942. doi:10.3390/mi12080942.

[5] X. Xing, A semigroup theory of rate independent hysteresis, arXiv:0707.3302, 2007.