The Delta Tree

A tombstone-free replicated list with relative ranges, isometric deletion, and an identity-arbitrated merge

Sal / FP Launchpad — KC + Claude

2026-07-14

Abstract

This note gives the complete design of the *delta tree*, a replicated list (sequence MRDT). The state has two parts with strictly separated roles. The *rendering* is a tree of live nodes carrying *parent-relative* fractional ranges (deltas): reads are a geometric traversal, inserts carve a slice of the anchor's free space, and delete is an *isometric fold* — the dead node's fraction is composed into its children's, so every survivor's displayed position is arithmetically unchanged. The *ledger* maps each identifier to its birth parent — written once, never mutated, never read outside merges. The merge arbitrates all ordering from the ledger's immutable *birth chains* (the canonical order is the RGA order of the birth tree restricted to survivors) and then re-derives the rendering; geometry is never an arbiter. The design is machine-validated: clean on the full litmus battery (L1–L25, the fooling probes, multi-epoch) except backward-run interleaving (L19) — the published RGA's own anomaly — and clean on 420 randomized version-DAG executions. Stronger: a lockstep harness asserting read-equality at every step shows the delta tree is *observationally equivalent to the published tombstoned RGA* on every scenario and every random execution tried — it admits exactly the North Star's anomalies, while storing one immutable parent id per node in place of tombstones in the sequence. Each design decision, in particular that identity arbitrates and geometry only renders, is forced by a machine-checked countermodel; §3 summarizes that campaign and §5 lays out the mechanization plan.

1 The design

State. $\sigma = (\rho, \Lambda)$. The *rendering* ρ is a finite map over live nodes,

$$\rho : t \;\mapsto\; (\,\mathrm{par},\ (lo, hi),\ \mathrm{char}\,), \qquad 0 \le lo < hi \le 1,$$

where timestamps t double as element identities and (lo, hi) is the node's range *relative to its parent's range* — the *delta*. The root sentinel 0 owns the unit interval and is never stored. A node's *absolute* range is the affine composition of the deltas along its root path. The *ledger* $\Lambda : t \mapsto$ birth parent records each node's parent at birth — written once at insert, never mutated, retained through delete. Its entries are needed only while some live node's birth chain passes through them, and pruning the unreachable rest at every merge is machine-checked sound (§4) — the ledger is a garbage-collected *heap* (roots = live nodes, pointers = birth parents), not a tombstone set. Reads never consult Λ; merges never order by ρ.

Insert. $\mathsf{ins}(x \leftarrow a)$: *carve* — the new child of anchor a takes a slice of a's free child-space above the current head. With remaining relative gap $(b, 1)$ (where b is the current head's hi, or 0), the newcomer takes $\left(b + \frac{w}{4},\, b + \frac{w}{2}\right)$, $w = 1 - b$, leaving room above for future siblings and below for

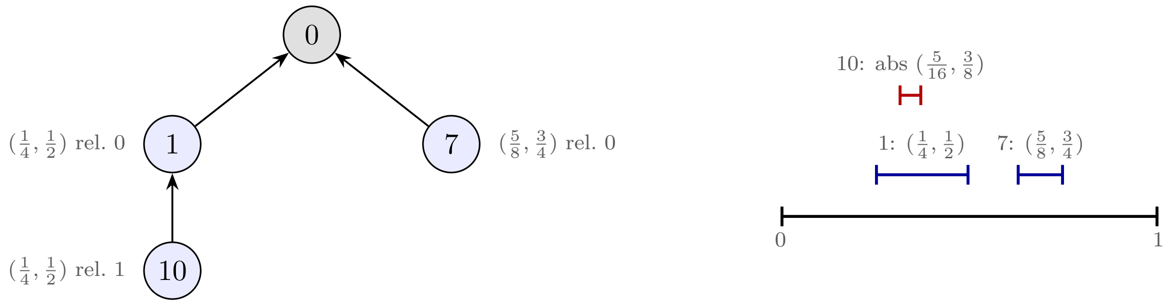

Figure 1: The rendering is a tree of deltas (left); absolute ranges are composed down the path (right): 10's absolute range is $\frac{1}{4} + \frac{1}{4} \cdot (\frac{1}{4}, \frac{1}{2}) = (\frac{5}{16}, \frac{3}{8})$. Display order: depth-first, children by descending range — newest first. The ledger for this state is $\Lambda = \{1 \mapsto 0,\ 7 \mapsto 0,\ 10 \mapsto 1\}$.

its own children. Newer siblings sit higher; the display (descending) shows them first — the RGA convention, realized geometrically instead of by timestamp comparison at read time. And one ledger write: $\Lambda[x] := a$.

Delete: the isometric fold. $\mathsf{del}(d)$: remove d's record from ρ; re-parent each child to the grandparent with the dead node's delta composed in: $c.\mathrm{frac} := d.\mathrm{frac} \circ c.\mathrm{frac}$. Every survivor's absolute position is unchanged — *delete-order preservation is an arithmetic identity*, where the flat RGA's climb re-sorts (the $[b, a, c] \to [c, b]$ anomaly that started this exploration) and the rose tree's splice needed sixteen thousand proof lines it ultimately could not have. Λ is untouched.

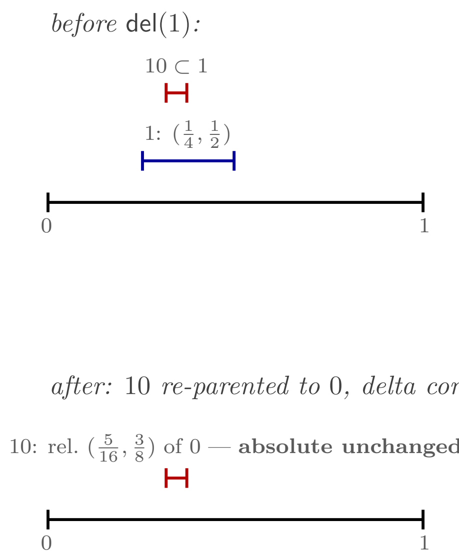

Figure 2: The isometric fold. Deleting 1 rewrites 10's delta to $(\frac{1}{4}, \frac{1}{2}) \circ (\frac{1}{4}, \frac{1}{2}) = (\frac{5}{16}, \frac{3}{8})$ relative to the root: the composition that *was* its absolute range. Nothing observable moves. Λ still records $10 \mapsto 1$ — the identity of the dead parent survives in the ledger, and §2 is why it must.

Read. Depth-first traversal of ρ, children by descending range (newest first). Purely geometric; the ledger is never consulted.

The operation payload and replay rule (per the proved RGA). Following the *why-the-path-matters* analysis: `rc` must be `Either` everywhere (any oriented insert/delete pair awakens the refuted `cond_comm_base`), so the insert carries its anchor's ancestor *id-path* plus **one** absolute slice — the composed delta at mint time. A reordered application resolves the id-path to the nearest live ancestor and re-relativizes the carried absolute, landing exactly where insert-then-delete would have put it: this fold-invariance is machine-checked as *exact state equality* (1303 reordered ins||del-anchor diamonds, `litmus/replay.py`). The placement alone is not the whole replay rule, however: a 3-op countermodel (CE1 in `replay.py`) shows that carried-slice placement, though it commutes perfectly, can converge to a sibling order contradicting the ledger's — geometry minted against one replica's sibling set is not canonical against another's. The correct rule (machine-checked clean across lockstep runs, all countermodels, and 48/48 linearizations) adds a *lazy canonical repair*: after any placement or fold, if the touched level's geometric order contradicts the ledger's birth-chain order, re-render **that level** with the sequential carve. Invariant 1 of §2 thus applies at op application too, not only at merges. At the origin the repair never fires, so generation remains exactly the local carve; concurrent same-anchor inserts commute up to $\approx$ (read- and merge-future-equal).

2 The merge

The merge is built on three invariants; each is forced by a named countermodel (§3).

1. **Arbitration from identity only.** Every order decision at a merge is computed from the ledger's immutable birth chains — never from current fractions, never by folding through mixed frames. Fractions drift: merges re-derive them, and a repair inside one sibling family perturbs numeric relations outside it. Chains never move, and a dead node's timestamp persists in its descendants' chains, so every verdict it participated in remains derivable, identically, at every replica and every time.

2. **Geometry is a rendering.** The merge re-derives all fractions to realize the canonical order. Between merges the geometry is authoritative for *display*; it is never authoritative for *arbitration*.

3. **Render/carve compatibility.** The re-render reproduces the sequential carve's geometry — oldest lowest, quarter slices, headroom above — so a post-merge state is indistinguishable from a sequentially carved one and the next local carve behaves exactly as §1 specifies. Merge output stays inside the image of the local operations.

merge(L, A, B), L the least common ancestor (the LCA discipline guarantees common past $\subseteq L$):

1. *Ledger union*: $\Lambda' = \Lambda_L \cup \Lambda_A \cup \Lambda_B$. Entries are immutable, so the union is conflict-free.

2. *Survival* (OR-set): live iff in all three or born in exactly one branch.

3. *Live parent*: each survivor re-parents to its nearest surviving ancestor, found by walking Λ' — the fold's re-parenting, computed from identity instead of frames.

4. *Canonical order*: per live parent, children sort by their **birth chains** (root paths in the immutable birth tree): compare per level, newest timestamp first; an ancestor's chain is a prefix of its descendant's and sorts first. Equivalently: *the canonical order is the RGA order of the birth tree, restricted to survivors.*

5. *Render*: re-derive all fractions realizing the canonical order by replaying the sequential carve — children bottom-to-top, oldest first, each taking $(b+\frac{w}{4}, b+\frac{w}{2})$, $w = 1-b$ (invariant 3). Recurse.

Worked example: concurrent twins. L empty; A: $\mathsf{ins}(10 \leftarrow 0)$, B: $\mathsf{ins}(6 \leftarrow 0)$ — identical carves, identical slices $(\frac{1}{4}, \frac{1}{2})$. Merge: chains are $\langle 10\rangle$ and $\langle 6\rangle$; canonical order puts 10 (newer) first. Render, oldest first: $6 \mapsto (\frac{1}{4}, \frac{1}{2})$, then 10 above it with $w = \frac{1}{2}$: $10 \mapsto (\frac{5}{8}, \frac{3}{4})$. Display $[10, 6]$ — exactly the state a replica that saw 6 then 10 would have carved sequentially (invariant 3).

Worked example: a verdict surviving its arbiter's death. The scenario that kills every geometry-arbitrating alternative (litmus L25), five nodes:

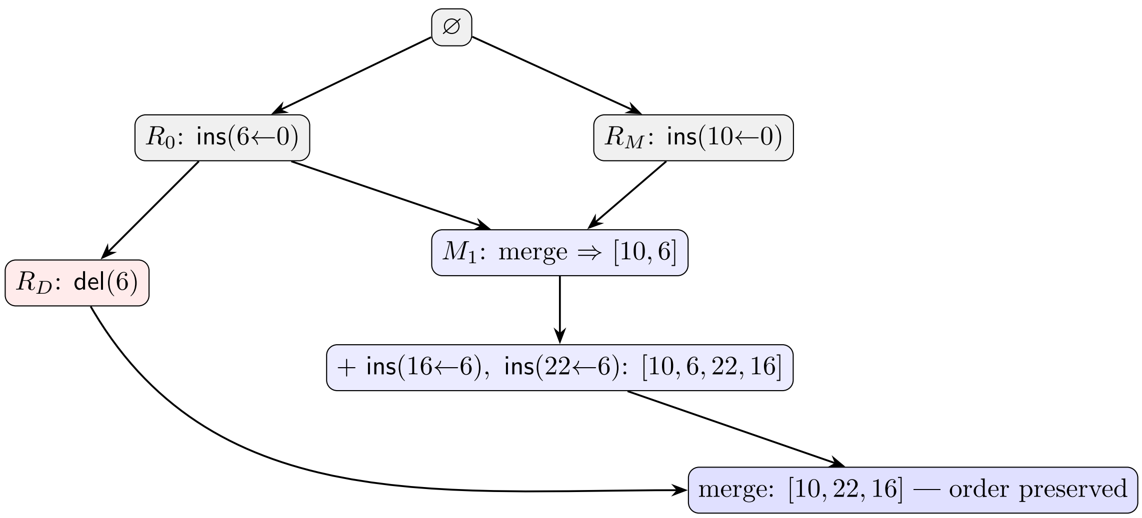

Figure 3: The M_1 merge ordered 10 before 6 (timestamps: $10 > 6$); typing 22 under 6 co-displayed 10 before 22. The deleter branch R_D knew only $\{6\}$. At the final merge 6 is dead — yet the verdict survives.

At the final merge the survivors are $\{10, 22, 16\}$, all with live parent 0. Their chains: $\langle 10\rangle$ for the root-born 10; $\langle 6, 22\rangle$ and $\langle 6, 16\rangle$ for 6's heirs — the chains run *through* the dead 6. Comparing 10 with 22 at the first level compares 10 with 6: the dead node's timestamp, persisting in its heirs' chains, keeps deciding, and 10 wins exactly as M_1 ruled. Within 6's heirs, $22 > 16$ puts 22 first. Display: $[10, 22, 16]$ — both previously co-displayed pairs preserved. Any merge that instead orders by folded geometry lets 22 re-litigate the slot with its *own* (newer) timestamp and win a contest its parent had lost: the display flips to $[22, 10, 16]$.

3 Design rationale: why identity arbitrates

Every choice above is the survivor of a machine-checked refutation campaign over the "geometric key" design space (sixteen designs, countermodels L17–L25 in `whiteboard/litmus/`; full record in its `README` and this file's git history). In brief: eager scalar keys die at the ceiling escapes (L17/L18); name-free carved ranges die at tie inheritance (L20); re-ranging at merge dies at three-branch convergence (L22); and a delta tree whose merge arbitrates from geometry while keeping the state *strictly* dead-free — no dead identifier anywhere — dies at random-DAG scale by two mechanisms: *fold-verdict erasure* (a dead node's timestamp is load-bearing for repair verdicts it participated in, and erasing it lets heirs re-litigate: the scenario of fig. 3) and *repair non-locality* (re-slotting inside one overlap family perturbs numeric relations outside it, so timestamp-decided ties get re-decided by position at causally disjoint merges). Eight variants of that family produced eight countermodels.

The campaign's summary is a triangle: of {strictly dead-free, pairwise display stability, topology convergence}, every pairwise combination has a machine-checked witness and the triple has only failed attempts — order verdicts must be re-derivable, identically, everywhere, and only immutable data naming *all* participants, including the dead ones, guarantees it. The delta tree accepts the minimum toll the triangle demands: one immutable parent id per node, at live-reachable scope, never consulted by reads. The birth chains are precisely the one-sided immutable path key (`path-2`'s after-side), stored as a parent pointer per node instead of a materialized path; the delta tree is that specification's runtime — $O(1)$ isometric deletes and geometric reads between merges. *Identity for deciding, geometry for reading.*

4 Validation

The litmus battery. The suite (`whiteboard/litmus/`, tests with provenance) ran the implementation (`DeltaTreeV3` in `litmus/delta_tree.py`). "RGA$_\dagger$" = the published, tombstoned RGA; "RGA$_{tf}$" = the repo's *proved* tombstone-free RGA.

test	delta tree	RGA$_\dagger$	RGA$_{tf}$	what it detects
L1 delete-reorder	✓	✓	×	sequential splice re-sort
L2 splice fooling pair	✓	✓	×	delete erases the side bit
L3a/b deep delete-runs	✓	✓	✓	transitive testimony loss
L4 criss-cross split	✓	✓	✓	the rose tree's refutation
L7 concurrent runs (g)	✓	✓	✓	forward interleaving
L17/L18 ceiling escapes	✓	✓	✓/×	eager-scalar staleness
L19 backward runs (h)	×	×	×	prepend interleaving
L20 tie inheritance	✓	✓	×	nameless carving
L21/L22/L23 topology	✓	✓	✓	re-ranging / frame leaks
L24 frame mixing	✓	✓	✓	cross-frame siblings
L25 fold-verdict inheritance	✓	✓	×	fig. 3's scenario
L5/L8–L13/L16 slot family	✓	✓	× (L8/9/13)	w-slot / puncture family
M1 two epochs	✓	✓	✓	key exhaustion across merges
L14/L15 fooling probes	✓†	✓†	× both	bounded-state impossibility
DAG PBT	✓(120/120, 300/300)	✓	× (80/120)	random branches and merges

† the delta tree and RGA$_\dagger$ *escape* the impossibility probes the way the spine designs do: their retained data (the ledger; the tombstones) distinguishes the probes' two worlds, so the impossibility's bounded-state hypothesis does not apply.

The execution model. Randomized version-DAG executions (`litmus/pbt.py`): n replicas; per round each replica either issues honest operations against its own read or merges with a peer — legal only when some recorded version's event set equals the two heads' intersection (the LCA discipline); final convergence rounds drive all replicas to the full event set along different merge paths. Checks: **FLIP** (global pairwise display stability: across every read of every replica, no pair is ever shown in both orders), **CONV** (equal event sets ⇒ equal reads), **LIVE/DUP** (survival correctness). Clean at 120/120 (4 replicas, 8 rounds) and 300/300 (6 replicas, 12 rounds), plus fresh-seed re-runs.

Ledger retention, state size, and precision (`litmus/measure.py`). Three machine-checked facts fix the cost model. *(a) Pruning is referential, not causal.* Per-replica reachability pruning — at every merge, drop Λ-entries on no live node's birth chain — is **sound**: lockstep-clean against unpruned across 800 DAG executions plus 210 churn executions (185,228 ops, 61,561 merges), with a deliberately-broken control confirming the harness bites. The invariant is inductive: every live node's full birth chain lies in Λ; at a merge, each survivor is live in some input, and that input's own closure re-supplies exactly the chain entries the survivor needs (entries are write-once, so unions never conflict). Conversely, **causal stability alone is unsound**: a six-step countermodel ($\mathsf{ins}(1 \leftarrow 0)$; $\mathsf{ins}(2 \leftarrow 1)$; $\mathsf{del}(1)$; full sync — the delete now stable everywhere; drop $\Lambda[1]$; $\mathsf{ins}(3 \leftarrow 2)$; merge) breaks the chain walk: survivor 2 is live and its chain passes through the stably deleted 1. The ledger inverts the classical tombstone-GC picture: what licenses forgetting is unreachability, not stability. *(b) Size.* The unpruned ledger equals the tombstoned RGA's state *exactly* (both = #inserts); the pruned ledger is $O(\text{live})$: at round 300 of the churn regime, 9 live + 17 ledger entries vs. 906 for $\text{RGA}_\dagger$ ($\approx 35\times$, growing linearly). *(c) Precision.* Post-merge, the maximum denominator obeys an exact law — bits = max sibling-set width + 2, verified on 3,241/3,241 merges — because the re-render replays quarter-carves: merges reset fold-accumulated precision, leaving a structural, not historical, floor (churn: 14 bits across 1,803 ops, vs. 76 bits merge-free).

The anomaly contract, relative to the North Star. What anomalies does the delta tree admit, compared to the published (tombstoned) RGA? The table shows column-for-column agreement; the true statement is stronger. A lockstep harness (`PairedV3RGA` in `delta_tree.py`) runs both designs on identical histories and asserts *read equality at every apply, every merge, and every read*: no divergence on any battery scenario — the two designs' L19 failures are the *identical* interleaving — nor on 420 randomized DAG executions. **Empirically, the delta tree *is* the published RGA**, with the tombstones compacted out of the display structure into the birth ledger: the canonical order (RGA order of the birth tree restricted to survivors) is precisely what the tombstoned read computes when it skips tombstones. The admitted anomalies are therefore exactly the North Star's: backward-run (prepend) interleaving — h, litmus L19, the classic RGA weakness that motivated Fugue — and nothing else; strong-list (e) and oracle fidelity (f) hold on every scenario tested and are unproven in general for both designs equally.

5 Mechanization plan

The two-part state suggests the proof structure directly.

- **Chains as the specification.** The canonical order is a deterministic function of the event set (the birth tree is immutable; survival is an OR-set), so convergence is determinism, and pairwise display stability reduces to monotonicity of the canonical order under event-set growth — statements about immutable data, in the style of the proved RGA's global-key re-sort.
- **The rendering as a refinement.** The geometric layer owes exactly invariants 2 and 3: the merge's render realizes the canonical order, and its output is in the image of the local operations. Local steps preserve the correspondence (the isometric fold is position-invariant by construction). The width+2 precision law (§4) says the fractions are bounded by structure, not history — so the mechanization can treat them as *derived* data (the render of the finite sibling order), never as stored truth.

- **Garbage collection as heap reachability.** The pruning obligation is referential — live nodes are roots, birth parents are pointers — so it phrases naturally as separation-logic-style reachability over the id heap, *not* as a causal-stability side-condition (which is machine-refuted, §4).

- **The conjectured headline theorem.** $\mathrm{read}_{\mathrm{delta}} = \mathrm{read}_{\mathrm{RGA}_\dagger}$ on all histories — the delta tree as a *compaction* of the published RGA, inheriting the North Star's specification wholesale. The lockstep harness is this theorem's test double.

- **The op layer.** `do_` = carried-slice placement + lazy canonical repair of the touched level (§1), written with the repair as the first-class case (empirically it is the common path in replays). Its invariant is level-local — *every sibling level canonical with respect to the ledger, and contained* — with three lemma-shaped obligations: placement+repair preserves it; fold+repair preserves it; any invariant state reads exactly the canonical order of (live ids, ledger). Replay ≈ merge is then immediate, since the merge renders that same order.

- **Framework fit.** A Conditioned MRDT with rc = Either everywhere and the id-path payload of §1, reusing the proved RGA's conditioning scaffolding (honest delivery; born accuracy + applicable delivery). Two scope facts from the simulation sketch (`litmus/equiv-proof-sketch.md`): the equivalence theorem is a statement of the *MRDT model* (under op-based causal delivery with tombstone-displayed anchors the two designs genuinely separate), and no-resurrection at merges rests on the honest-LCA / reachability conditioning — exactly the framework's existing machinery.

Provenance. Every table cell and number in this note is reproduced by `whiteboard/litmus/`: `litmus.py` (the battery), `pbt.py` (the execution model), `delta_tree.py` (`DeltaTreeV3`, the refuted geometry-arbitrating variants kept as regression artifacts, the lockstep equivalence harness, and the self-test runner). The refutation campaign that fixed this design's invariants is recorded in `whiteboard/litmus/README.md` and in this file's git history.