OCTA Research

Technical Textbook Series

Memory Arithmetic

Numbers, History, and the Mathematics of Irreversibility

Chapter 1 Standalone Packet (Final)

Why Memory Must Enter Mathematics

Expanded Math-Book Edition · Build v1.9

Standalone compile: title + preface + notation + chapter + exercises + worked diagrams + solutions

---

Thesis Statement (Operational)

Classical arithmetic is complete on *values* yet incomplete on *provenance*. Memory Arithmetic upgrades the number type from "a value" to a structured object

$$m = \langle \text{value} \mid \text{history} \rangle,$$

so that computation, audit, measurement, and learning can be treated with correct identity semantics. Composition introduces *new events* (intrinsic irreversibility), while abstraction is an *explicit operator* (forgetting) with budget and cost semantics.

Read Preface and Notation, then Chapter 1 sequentially. Each exercise is immediately followed by a worked example/diag

Contents

Preface

This is Chapter 1 of the **OCTA Research** textbook *Memory Arithmetic*. The aim is not stylistic novelty. The aim is *type-correctness* for mathematics as it is used in real systems.

Key Idea

Core claim. A number used in the real world is not merely a value; it is a value produced by a process. Processes have identity, constraints, irreversible events, and compressible summaries. Therefore, the number type must include provenance as structure.

This chapter is written as a self-contained teaching unit:

- definitions and axioms,
- canonical model (so the theory is not "hand-wavy"),
- diagrams that show how the objects behave,
- a complete exercise set: **Exercise** → **Worked Example/Diagram** → **Solution**.

OCTA Research Note

Layout discipline. Margin notes have been removed entirely for this packet because they frequently cause left-bleed on certain compilers and templates. All side commentary is embedded as boxed notes.

Notation

OCTA Research Note

Notation is intentionally redundant; this file must compile and teach as a standalone chapter.

- **Value space.** $\mathbb{V}$ is a commutative monoid $(\mathbb{V}, +, 0_V)$; often a commutative semiring $(\mathbb{V}, +, \cdot, 0_V, 1_V)$.
- **History space.** $\mathbb{H}$ is a class of finite provenance objects (trees or DAGs), with isomorphism relation $\cong$.
- **Memory numbers.** $\mathbb{M} := \mathbb{V} \times \mathbb{H}$. A memory number is $m = \langle v \mid h \rangle$ with $v \in \mathbb{V}$, $h \in \mathbb{H}$.
- **Projections.** $\mathrm{val} : \mathbb{M} \to \mathbb{V}$ and $\mathrm{hist} : \mathbb{M} \to \mathbb{H}$ are projections: $\mathrm{val}(\langle v \mid h \rangle) = v$, $\mathrm{hist}(\langle v \mid h \rangle) = h$.
- **Operations.** $\oplus, \otimes : \mathbb{M} \times \mathbb{M} \to \mathbb{M}$ (memory-addition and memory-multiplication). They are *value-correct* and *history-expansive*.
- **Mass.** $\mu : \mathbb{M} \to \mathbb{R}_{\geq 0}$ measures stored provenance. Canonical choice: internal operation nodes in $\mathrm{hist}(m)$.
- **Entropy functional.** $\mathcal{S}(m) := \log(1 + \mu(m))$ (baseline; later chapters refine).
- **Forgetting.** $\mathcal{F}_k : \mathbb{M} \to \mathbb{M}$ is abstraction/forgetting with budget $k \in \mathbb{N}$. Forgetting is value-preserving and mass-capping.
- **Equalities.**
  - Value equality: $m \equiv_v n \iff \mathrm{val}(m) = \mathrm{val}(n)$.
  - Structural equality: $m \equiv_s n \iff \mathrm{hist}(m) \cong \mathrm{hist}(n)$.
  - Experiential equality: $m \equiv_e n$ (informal in Ch. 1) when histories share a nontrivial causal core; formalized later via embeddings/intersections.
- **Genesis.** $\mathbf{1} = \langle 1_V \mid \mathbf{G} \rangle$ is the primitive "atom" with genesis label $\mathbf{G}$.
- **Summary leaf.** $\bullet$ is used in worked diagrams to represent a summarized/forgotten subtree.

Chapter 1

Why Memory Must Enter Mathematics

1.1 Orientation: arithmetic as used vs arithmetic as idealized

Classical arithmetic is a value calculus. It treats numbers as memoryless objects. But systems that *use* numbers—measurement, computation, training, audit, security—do not interact with numbers in that way. They interact with *procedures* that generate values under constraints.

Key Idea

Upgrade the type, not the commentary. If provenance is essential, it cannot be an optional annotation. It must live inside the object type:

$$m = \langle v \mid h \rangle.$$

The projection $m \mapsto v$ recovers classical arithmetic, but the full object carries identity.

1.2 Three concrete failures of value-only arithmetic

1.2.1 Failure I: value conflates identity

Two different procedures can yield the same value. In practice they can be non-interchangeable:

- same test accuracy, different training data lineage;
- same calibration number, different standards chain;
- same ledger balance, different transaction provenance.

1.2.2 Failure II: reversibility is assumed where none exists

Value-only arithmetic suggests one can "undo" constructions freely (cancellation as identity). But provenance grows with each composition event; undoing would require removing history, which is a *different primitive*.

1.2.3 Failure III: abstraction is treated informally

In real systems, we intentionally discard information (compression, summarization, hashing, checkpointing). Value-only arithmetic has no operator for "forgetting with budget."

1.3 Design principles

We fix a discipline that later chapters refine.

Key Idea

Design principles (Chapter 1).

(P1) **Projection correctness:** values behave classically under projection.

(P2) **Provenance structure:** histories participate in algebra (not mere metadata).

(P3) **Intrinsic irreversibility:** composition introduces a new event.

(P4) **Explicit abstraction:** forgetting is an operator with parameters/budgets.

(P5) **Layered equality:** there is not one equality but several.

1.4 Formal objects

1.4.1 Value space

Definition 1.1 (Value space)**.** *A value space is a commutative monoid* $(\mathbb{V}, +, 0_V)$*. When multiplication is used, assume a commutative semiring* $(\mathbb{V}, +, \cdot, 0_V, 1_V)$*.*

1.4.2 History space

Definition 1.2 (History space)**.** *A history space* $\mathbb{H}$ *is a class of finite provenance objects equipped with:*

(H1) an isomorphism relation $\cong$*,*

(H2) a size functional $|h|$*,*

(H3) join constructors J_+ *and* $J_\times$ *that form a new event node from two histories.*

1.4.3 Memory numbers

Definition 1.3 (Memory number)**.** *A memory number is* $m = \langle v \mid h \rangle \in \mathbb{M} := \mathbb{V} \times \mathbb{H}$ *with projections* $\mathrm{val}(m) = v$ *and* $\mathrm{hist}(m) = h$*.*

1.5 Axioms (Chapter 1 baseline)

1.5.1 Genesis

Definition 1.4 (Genesis atom)**.** *Let* $\mathbf{1} := \langle 1_V \mid \mathbf{G} \rangle$ *be the primitive unit with genesis label* $\mathbf{G}$*.*

1.5.2 Value correctness

Definition 1.5 (Value-correct operations)**.** *Define* $\oplus, \otimes : \mathbb{M} \times \mathbb{M} \to \mathbb{M}$ *such that:*

$$\mathrm{val}(m \oplus n) = \mathrm{val}(m) + \mathrm{val}(n), \qquad \mathrm{val}(m \otimes n) = \mathrm{val}(m) \cdot \mathrm{val}(n).$$

1.5.3 History expansion

Definition 1.6 (History-expansive operations)**.** *Require join constructors* $J_+, J_\times$ *with:*

$$\mathrm{hist}(m \oplus n) = J_+(\mathrm{hist}(m), \mathrm{hist}(n)), \qquad \mathrm{hist}(m \otimes n) = J_\times(\mathrm{hist}(m), \mathrm{hist}(n)).$$

1.5.4 Mass and strict growth

Definition 1.7 (Mass)**.** *Fix* $\mu(m) := |\,\mathrm{hist}(m)|$.

Definition 1.8 (Strict growth (irreversibility axiom))**.** *For all* $m, n \in \mathbb{M}$,

$$\mu(m \oplus n) > \max(\mu(m), \mu(n)), \qquad \mu(m \otimes n) > \max(\mu(m), \mu(n)).$$

1.5.5 Forgetting

Definition 1.9 (Forgetting operators)**.** *A family* $\{\mathcal{F}_k\}_{k \in \mathbb{N}}$ *satisfies:*

$$\mathrm{val}(\mathcal{F}_k(m)) = \mathrm{val}(m), \quad \mathcal{F}_k(\mathcal{F}_k(m)) = \mathcal{F}_k(m), \quad \mu(\mathcal{F}_k(m)) \le k.$$

Warning / Pitfall

Pitfall. Forgetting is not "normalization." Forgetting discards identity information; normalization seeks a canonical representative within an identity class. Normalization appears later once the equivalence relations are fully stabilized.

1.6 Canonical model (ordered history trees)

To ensure this chapter is not abstract-only, we give a concrete model you can compute with immediately.

Definition 1.10 (Canonical ordered-tree model)**.** *Let* $\mathbb{H}$ *be finite rooted ordered labeled trees with labels in* $\{\mathbf{G}, +, \times\}$*. Let* $\cong$ *preserve root, labels, and left-to-right child order.*

Define:

$$\mathbf{1} = \langle 1 \mid \mathbf{G} \rangle, \quad m_1 \oplus m_2 = \langle v_1 + v_2 \mid +(h_1, h_2) \rangle, \quad m_1 \otimes m_2 = \langle v_1 v_2 \mid \times(h_1, h_2) \rangle.$$

Let $|h|$ *count internal nodes labeled* $+$ *or* $\times$ *(genesis leaves do not contribute).*

Proposition 1.11 (Mass recursion in canonical model)**.**

$$\mu(m \oplus n) = \mu(m) + \mu(n) + 1, \qquad \mu(m \otimes n) = \mu(m) + \mu(n) + 1.$$

1.7 Layered equality (introduced here, used throughout)

Definition 1.12 (Value equality)**.**

$$m \equiv_v n \iff \mathrm{val}(m) = \mathrm{val}(n).$$

Definition 1.13 (Structural equality)**.**

$$m \equiv_s n \iff \mathrm{hist}(m) \cong \mathrm{hist}(n).$$

Definition 1.14 (Experiential equality (Chapter 1 preview))**.** *$m \equiv_e n$ if their histories share a nontrivial causal core. In Chapter 1 this is intuition; later chapters define it via embeddings/intersections and "shared ancestry" operators.*

1.8 Core diagrams (fully constrained and centered)

1.8.1 Memory number factorization

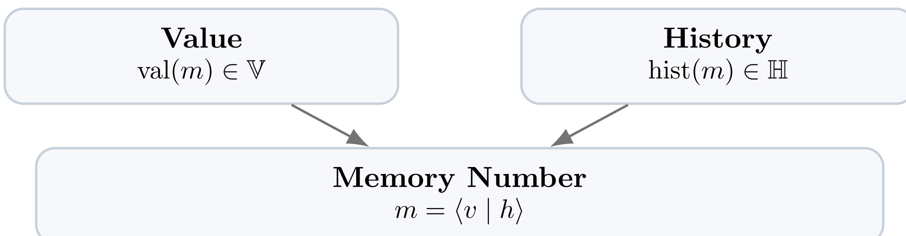


Figure 1.1: A memory number is a paired object: value + provenance.

1.8.2 Strict-growth arrow (intrinsic irreversibility)

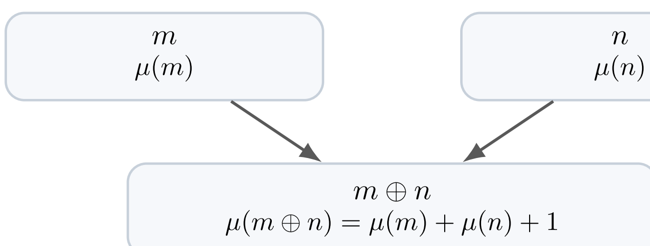


Figure 1.2: Each composition introduces a new event: irreversibility is intrinsic.

1.8.3 Quotient collapse to classical arithmetic

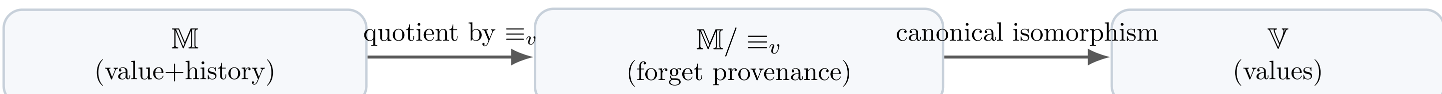


Figure 1.3: Classical arithmetic is recovered by quotienting out provenance.

1.9 Entropy functional (Chapter 1 baseline)

Definition 1.15 (Memory entropy)**.** *Define*

$$\mathcal{S}(m) := \log(1 + \mu(m)).$$

Proposition 1.16 (Entropy strictly increases under composition (canonical model))**.** *If μ follows the recursion $\mu(m \oplus n) = \mu(m) + \mu(n) + 1$, then*

$$\mathcal{S}(m \oplus n) > \max(\mathcal{S}(m), \mathcal{S}(n)).$$

OCTA Research Note

This is the minimal "Second Law" form needed in Chapter 1. Later chapters replace μ by description-length or MDL-based quantities and add physical semantics (Landauer-style costs).

1.10 Exercises (each with worked diagrams and solutions)

Exercise

Exercise 1 (Quotient recovery). Define $\sim$ on $\mathbb{M}$ by $m \sim n \iff m \equiv_v n$. Assume val is surjective onto $\mathbb{V}$.

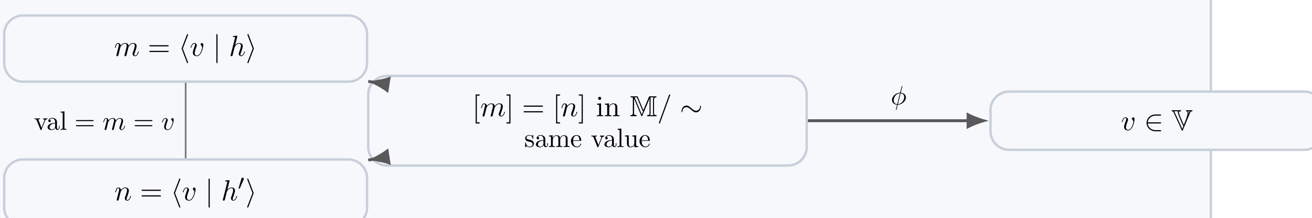
(a) Prove $\sim$ is an equivalence relation.

(b) Define $\phi : \mathbb{M}/\sim \to \mathbb{V}$ by $\phi([m]) = \mathrm{val}(m)$ and prove ϕ is a bijection.

(c) Define $[m] \oplus [n] := [m \oplus n]$ and prove it is well-defined.

Worked Example / Guided Work

Worked diagram.



Solution

Solution. (a) Reflexive: $\mathrm{val}(m) = \mathrm{val}(m)$. Symmetric and transitive follow from equality in $\mathbb{V}$. (b) Well-defined: if $[m] = [m']$ then $\mathrm{val}(m) = \mathrm{val}(m')$. Injective: $\phi([m]) = \phi([n]) \Rightarrow \mathrm{val}(m) = \mathrm{val}(n) \Rightarrow [m] = [n]$. Surjective: val surjective gives m with $\mathrm{val}(m) = v$ for any

$v \in \mathbb{V}$. (c) If $m \equiv_v m'$ and $n \equiv_v n'$, then

$$\mathrm{val}(m \oplus n) = \mathrm{val}(m) + \mathrm{val}(n) = \mathrm{val}(m') + \mathrm{val}(n') = \mathrm{val}(m' \oplus n'),$$

so $m \oplus n \equiv_v m' \oplus n'$ and the operation is well-defined.

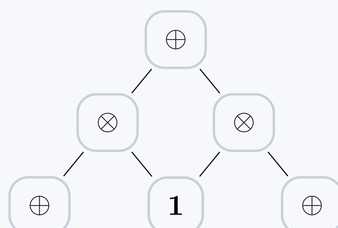
Exercise

Exercise 2 (Mass equals operation count). In the ordered-tree canonical model, prove $\mu(t)$ equals the number of operation symbols in term t built from $\mathbf{1}$ using $\oplus, \otimes$. Compute $\mu(t)$ for:

$$t = ((\mathbf{1} \oplus \mathbf{1}) \otimes \mathbf{1}) \oplus (\mathbf{1} \otimes (\mathbf{1} \oplus \mathbf{1})).$$

Worked Example / Guided Work

Worked tree (operation skeleton).



Count operation nodes: root $\oplus$ (1), two $\otimes$ (2,3), two $\oplus$ (4,5). Hence $\mu(t) = 5$.

Solution

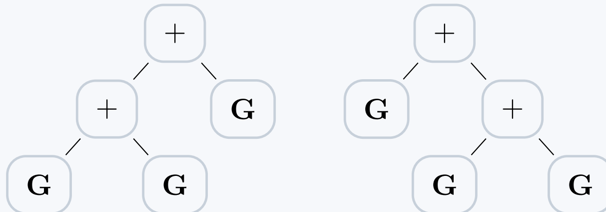
Solution. Induct on the term: base $\mathbf{1}$ has 0 ops and $\mu(\mathbf{1}) = 0$. If $t = t_1 \oplus t_2$ then $\mu(t) = \mu(t_1) + \mu(t_2) + 1$, matching op-count recursion, similarly for $\otimes$. Thus op-count equals $\mu(t)$ for all terms; for the given t, $\mu(t) = 5$.

Exercise

Exercise 3 (Associativity fracture, drawn). Let $a = (\mathbf{1} \oplus \mathbf{1}) \oplus \mathbf{1}$ and $b = \mathbf{1} \oplus (\mathbf{1} \oplus \mathbf{1})$. Prove $a \equiv_v b$ but $a \not\equiv_s b$.

Worked Example / Guided Work

Worked histories.



Ordered-tree isomorphisms preserve left/right child positions, so these are not isomorphic.

Solution

Solution. Value: $(1+1)+1=3$ and $1+(1+1)=3$, so $a \equiv_v b$. Structure: the root's left child label differs ($+$ vs $\mathbf{G}$) under ordered isomorphisms, hence $a \not\equiv_s b$.

Exercise

Exercise 4 (Structural noncommutativity witness). Let $m = \mathbf{1}$ and $n = \mathbf{1} \oplus \mathbf{1}$. Show $m \oplus n \equiv_v n \oplus m$ but $m \oplus n \not\equiv_s n \oplus m$ by drawing histories.

Worked Example / Guided Work

Worked rewriting to Exercise 3 forms.

$$m \oplus n = \mathbf{1} \oplus (\mathbf{1} \oplus \mathbf{1}), \qquad n \oplus m = (\mathbf{1} \oplus \mathbf{1}) \oplus \mathbf{1}.$$

These are exactly the two trees from Exercise 3; hence value-equal but structurally distinct.

Solution

Solution. Value equality holds by commutativity in $\mathbb{V}$: $1+(1+1) = (1+1)+1$. Structural inequality holds because ordered-tree child positions differ, so no isomorphism exists.

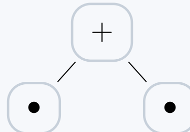
Exercise

Exercise 5 (Forgetting collapses identity). Using depth truncation $\mathcal{F}_1$, show there exist $m \not\equiv_s n$ with $\mathcal{F}_1(m) \equiv_s \mathcal{F}_1(n)$. Use $m = (\mathbf{1} \oplus \mathbf{1}) \oplus \mathbf{1}$ and $n = \mathbf{1} \oplus (\mathbf{1} \oplus \mathbf{1})$.

Worked Example / Guided Work

Worked truncation (depth 1). Depth-1 truncation keeps only root and its children; below becomes $\bullet$. Both histories reduce to:

$$+(\bullet, \bullet).$$



Solution

Solution. $m \not\equiv_s n$ from Exercise 3. After depth-1 truncation both become the same skeleton $+(\bullet, \bullet)$, hence $\mathcal{F}_1(m) \equiv_s \mathcal{F}_1(n)$.

Exercise

Exercise 6 (No structural cancellation in canonical model). Assume the canonical mass recursion $\mu(m \oplus n) = \mu(m) + \mu(n) + 1$ and that structural equality implies equal mass. Show there is no operation $\ominus$ on $\mathbb{M}$ such that for all m, n,

$$(m \oplus n) \ominus n \equiv_s m.$$

Worked Example / Guided Work

Worked mass contradiction. If such an $\ominus$ existed, then

$$(m \oplus n) \ominus n \equiv_s m \quad \Rightarrow \quad \mu((m \oplus n) \ominus n) = \mu(m).$$

But $(m \oplus n)$ has mass $\mu(m) + \mu(n) + 1$, and any attempt to "remove n" must delete event nodes. Deleting event nodes is exactly what forgetting does, not a value-style inverse.

Solution

Solution. Suppose $\ominus$ exists with $(m \oplus n) \ominus n \equiv_s m$ for all m, n. Take any n with $\mu(n) \geq 0$ and any m. Then $\mu(m \oplus n) = \mu(m) + \mu(n) + 1 > \mu(m)$. If $(m \oplus n) \ominus n \equiv_s m$, then masses must match, so

$$\mu((m \oplus n) \ominus n) = \mu(m).$$

But transforming $m \oplus n$ to an object structurally equal to m would require removing at least the top join event (and typically the entire n branch), which contradicts the strict growth discipline unless an explicit forgetting primitive is invoked. Hence no such $\ominus$ exists.

Exercise

Exercise 7 (Entropy monotonicity from strict growth). Let $\mathcal{S}(m) = \log(1 + \mu(m))$. Assume strict growth $\mu(m \oplus n) > \max(\mu(m), \mu(n))$. Prove:

$$\mathcal{S}(m \oplus n) > \max(\mathcal{S}(m), \mathcal{S}(n)).$$

Worked Example / Guided Work

Worked inequality chain. Strict growth gives $\mu(m \oplus n) > \mu(m)$ and $\mu(m \oplus n) > \mu(n)$. Since $\log(1 + x)$ is strictly increasing on $x \geq 0$,

$$\log(1 + \mu(m \oplus n)) > \log(1 + \mu(m)) \quad \text{and} \quad \log(1 + \mu(m \oplus n)) > \log(1 + \mu(n)).$$

Solution

Solution. Apply monotonicity of $\log(1 + x)$ to both strict inequalities from strict growth; the result follows immediately.

Chapter Summary

- The number type is upgraded: $m = \langle v \mid h \rangle$ with projections $\mathrm{val}, \mathrm{hist}$.
- Operations are *value-correct* but *history-expansive*; irreversibility is internal via strict growth.
- Forgetting is an explicit operator: value-preserving, mass-capping, idempotent.
- Equality splits into layers: $\equiv_v$ (value), $\equiv_s$ (structure), $\equiv_e$ (experience/causal core).
- A concrete canonical model (ordered trees) is given so everything is computable immediately.
- Exercises fully demonstrate quotient recovery, structural fractures, forgetting collapse, no cancellation, and entropy monotonicity.

Roadmap (Where This Goes Next)

Immediate next chapter (Chapter 2) should:

- formalize multiple forgetting families (depth, budget, MDL, stochastic) and compare them,
- build a history metric d_H and induced memory metric d,
- define experiential equality $\equiv_e$ rigorously via embeddings/intersections,
- introduce normalization vs forgetting (canonical forms) as separate mechanisms,
- expand problem sets to include DAG histories and distributivity witness diagrams.