Memory Arithmetic

Numbers, History, and the Mathematics of Irreversibility

A Foundational Textbook (Expanded Edition)

OCTA Research · Version 1.0

OCTA Research Thesis

Classical arithmetic is complete on values yet incomplete on provenance. Memory Arithmetic upgrades mathematics by making history structural: layered equality, intrinsic irreversibility, explicit forgetting, geometric/thermodynamic semantics, and mechanizable foundations.

Contents

Preface

Classical arithmetic treats numbers as memoryless. Computation and physics do not. This text builds a coherent discipline—*Memory Arithmetic*—where history is not metadata but structure.

- layered equality (value vs identity),
- intrinsic irreversibility,
- explicit operators of forgetting and abstraction,
- geometric, thermodynamic, and computational semantics,
- formalizability inside proof assistants.

OCTA Research Note

This build is *chapter-integrated*. Prior "Volume I/II/III" material is promoted into full chapters, and all their exercises are placed in the correct core locations (geometry, physics/time, formalization).

Notation

- V : value space (commutative monoid or semiring).
- H : history space (finite rooted ordered DAGs/trees with operation labels).
- $\mathbb{M}$: memory-number space.
- $\mathrm{val} : \mathbb{M} \to V$, $\mathrm{hist} : \mathbb{M} \to H$.
- $\mu(m)$: memory mass.
- $\mathcal{S}(m)$: memory entropy.
- $m_1 \oplus m_2$: memory-addition; $m_1 \otimes m_2$: memory-multiplication.
- $\mathcal{F}_k(m)$: forgetting operator (truncate/abstract history).
- History tools: $\mathrm{Join}, \mathrm{Prune}, \mathrm{Edit}$.
- Equalities: $\equiv_v$ (value), $\equiv_s$ (structural), $\equiv_e$ (experiential).

Part I

Foundations

Chapter 1

Why Memory Must Enter Mathematics

Mathematics traditionally treats numbers as eternal, context-free objects. Yet every real process that computes, measures, or observes numbers is *historical*.

1.1 Three Failures of Classical Arithmetic

CTA
ue $\neq$ identity.

1. It erases the path by which a value is obtained.
2. It assumes reversibility where none exists.
3. It conflates equality of value with equality of identity.

1.2 Design Principles

- **Projection**: value remains classical.
- **Provenance**: identity carries history.
- **Irreversibility**: history grows under composition.
- **Forgetting**: abstraction is an operator.

1.3 Diagram: Memory Number Factorization

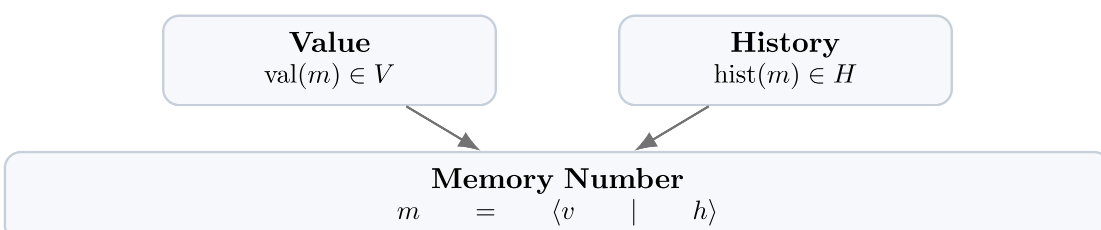

Figure 1.1: A memory number splits into value and provenance.

Chapter Summary

Memory Arithmetic exists because real computation is not value-only. We will formalize (i) layered equality, (ii) intrinsic irreversibility, (iii) explicit abstraction.

Exercises (This Chapter)

Exercise 1.1. *Give a computation (training, measurement, audit) where value-only arithmetic loses meaning. State what a history object must encode.*

Exercise 1.2. *Provide two procedures producing the same numerical result where you would not accept them as the same outcome. State which identity property you used.*

Selected Solutions

Solution. *In ML, identical accuracy can arise from disjoint datasets or unstable training trajectories; history must encode provenance (data, transforms, steps, constraints).*

Chapter 2

Primitive Objects and Intuition

2.1 What Is a Memory Number?

Definition 2.1 (Memory Number). *A* memory number *is an ordered pair*

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

where $v \in V$ is a value and $h \in H$ is a finite history object encoding how v arose.

Remark 2.2. *Two memory numbers may have the same value yet be fundamentally different objects.*

2.2 Toy Example: Associativity Fractures Structurally

$$(\mathbf{1} \oplus \mathbf{1}) \oplus \mathbf{1} \quad \text{and} \quad \mathbf{1} \oplus (\mathbf{1} \oplus \mathbf{1})$$

have equal value but different identity.

OCTA Research Note

Disable FAST mode to render paired left-nested vs right-nested history trees.

Chapter Summary

A memory number is not a value with annotations; it is a value *paired with* a provenance object that participates in algebra.

Exercises (This Chapter)

Exercise 2.1. *Construct explicit m, n such that $m \equiv_v n$ but $m \not\equiv_s n$.*

Exercise 2.2. *In the canonical tree model where mass counts internal nodes, compute $\mu(m)$ and $\mu(n)$ for your example.*

Selected Solutions

Solution. *Take* $m = ((\mathbf{1} \oplus \mathbf{1}) \oplus \mathbf{1})$ *and* $n = (\mathbf{1} \oplus (\mathbf{1} \oplus \mathbf{1}))$*. Then* $\mathrm{val}(m) = \mathrm{val}(n) = 3 1_V$ *but ordered-tree histories are non-isomorphic, so* $m \not\equiv_s n$*.*

Chapter 3

Formal Foundations

3.1 Value Space

Definition 3.1 (Value Space (Monoid))**.** *The value space* $(V, \oplus_V, 0_V)$ *is a commutative monoid: for all* $a, b, c \in V$*,*

$$a \oplus_V b = b \oplus_V a, \qquad (a \oplus_V b) \oplus_V c = a \oplus_V (b \oplus_V c), \qquad a \oplus_V 0_V = a.$$

When multiplication is needed, we assume $(V, \oplus_V, \otimes_V, 0_V, 1_V)$ *is a commutative semiring. In the canonical model,* $V = \mathbb{R}$ *with standard* $+$ *and* $\cdot$*.*

3.2 History Space

Definition 3.2 (History Space)**.** H *is a class of finite rooted ordered DAGs (or trees) whose nodes are labeled by operation symbols (e.g.* $\{G, +, \times, \delta\}$*). Histories are considered up to an isomorphism relation* $\cong$*.*

3.3 The Universe of Memory Numbers

$$\mathbb{M} := V \times H.$$

3.4 Diagram: Value Quotient Collapses Identity

$$\boxed{\begin{array}{c}\mathbb{M}\\ \text{(value + history)}\end{array}} \xrightarrow{\text{quotient by } \equiv_v} \boxed{\begin{array}{c}V\\ \text{(values only)}\end{array}}$$

Figure 3.1: Classical arithmetic is the value quotient of Memory Arithmetic.

Chapter Summary

We build $\mathbb{M}$ as a product space with a projection val and a provenance component hist, then define operations that are value-correct but history-expansive.

Exercises (This Chapter)

Exercise 3.1. *Give two realizations of H (trees vs DAGs). For each, state one advantage and one complication.*

Exercise 3.2. *Define $\cong$ for ordered labeled trees. What must be preserved?*

Chapter 4

Core Axioms of Memory Arithmetic

4.1 Signature

Primitives: $\mathrm{val}, \mathrm{hist}, \oplus, \otimes, \mathcal{F}_k, \mu$.

Axiom 4.1 (Projection).

$$\mathrm{val} : \mathbb{M} \to V, \quad \mathrm{hist} : \mathbb{M} \to H.$$

Axiom 4.2 (Genesis).

$$\mathbf{1} = \langle 1_V \mid G \rangle.$$

Axiom 4.3 (Value of Memory-Addition).

$$\mathrm{val}(m_1 \oplus m_2) = \mathrm{val}(m_1) \oplus_V \mathrm{val}(m_2).$$

Axiom 4.4 (History of Memory-Addition).

$$\mathrm{hist}(m_1 \oplus m_2) = \mathrm{Join}(\mathrm{hist}(m_1), \mathrm{hist}(m_2); m_1, m_2).$$

Axiom 4.5 (Memory Mass).

$$\mu(m) = |\,\mathrm{hist}(m)|.$$

Axiom 4.6 (Strict Growth).

$$\mu(m_1 \oplus m_2) > \max(\mu(m_1), \mu(m_2)).$$

4.2 Multiplication Axioms

Axiom 4.7 (Value of Memory-Multiplication).

$$\mathrm{val}(m_1 \otimes m_2) = \mathrm{val}(m_1) \otimes_V \mathrm{val}(m_2).$$

Axiom 4.8 (History of Memory-Multiplication).

$$\mathrm{hist}(m_1 \otimes m_2) = \mathrm{Join}(\mathrm{hist}(m_1), \mathrm{hist}(m_2); m_1, m_2) \ \textit{with label} \ \times .$$

Theorem 4.9 (Value-Level Distributivity).

$$m_1 \otimes (m_2 \oplus m_3) \ \equiv_v \ (m_1 \otimes m_2) \oplus (m_1 \otimes m_3).$$

4.3 Diagram: Distributivity Witness (Two Histories)

$\boxed{\times(a,\ +(b,c))}$ $\boxed{+\big(\times(a,b),\ \times(a,c)\big)}$

Figure 4.1: Value-equal distributivity; structurally distinct histories.

Exercises (This Chapter)

Exercise 4.1. *Assume μ counts internal nodes. Prove:*

$$\mu(m_1 \oplus \cdots \oplus m_k) \geq (k-1) + \max_i \mu(m_i).$$

Chapter 5

Equality Is No Longer Singular

5.1 Three Layers of Equality

Definition 5.1 (Value Equality).

$$m_1 \equiv_v m_2 \iff \mathrm{val}(m_1) = \mathrm{val}(m_2).$$

Definition 5.2 (Structural Equality).

$$m_1 \equiv_s m_2 \iff \mathrm{hist}(m_1) \cong \mathrm{hist}(m_2).$$

Definition 5.3 (Experiential Equality). *$m_1 \equiv_e m_2$ if their histories share a nontrivial common ancestral substructure.*

5.2 Congruence

Theorem 5.4 ($\equiv_v$ is a Congruence). *If $m_1 \equiv_v m_1'$ and $m_2 \equiv_v m_2'$, then*

$$(m_1 \oplus m_2) \equiv_v (m_1' \oplus m_2') \quad \text{and} \quad (m_1 \otimes m_2) \equiv_v (m_1' \otimes m_2').$$

Exercises (This Chapter)

Exercise 5.1. *Prove that $\equiv_v$ is an equivalence relation and a congruence for $\oplus$ and $\otimes$.*

Chapter 6

Noncommutativity and Nonassociativity

Theorem 6.1 (Structural Noncommutativity)**.**

$$m_1 \oplus m_2 \not\equiv_s m_2 \oplus m_1 \quad \textit{in general.}$$

Theorem 6.2 (Structural Nonassociativity)**.**

$$(m_1 \oplus m_2) \oplus m_3 \not\equiv_s m_1 \oplus (m_2 \oplus m_3) \quad \textit{in general.}$$

6.1 Diagram: Order Matters

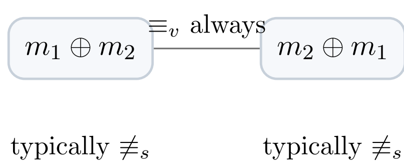

Figure 6.1: Value commutativity can coexist with structural noncommutativity.

Exercises (This Chapter)

Exercise 6.1. *Show that $m \oplus n \equiv_v n \oplus m$ always holds, but $m \oplus n \equiv_s n \oplus m$ fails generically. Give a sufficient condition for structural commutativity.*

Chapter 7

Forgetting, Decay, and Approximation

Definition 7.1 (Forgetting).

$$\mathcal{F}_k(m) = \langle \mathrm{val}(m) \mid \mathrm{Prune}_k(\mathrm{hist}(m)) \rangle.$$

Axiom 7.2 (Forgetting Preserves Value).

$$\mathrm{val}(\mathcal{F}_k(m)) = \mathrm{val}(m).$$

Axiom 7.3 (Forgetting Is Idempotent).

$$\mathcal{F}_k(\mathcal{F}_k(m)) = \mathcal{F}_k(m).$$

Axiom 7.4 (Forgetting Caps Mass).

$$\mu(\mathcal{F}_k(m)) \leq k.$$

Definition 7.5 (Decay).

$$m \ominus_d \lambda := \mathcal{F}_{\lfloor \lambda\, \mu(m) \rfloor}(m), \quad \lambda \in [0, 1].$$

7.1 Concrete Forgetting Operators

TA
th, budget,
L, stochastic.

7.1.1 Scheme A: Depth Truncation

Definition 7.6 (Depth Truncation Prune). $\mathrm{Prune}_k^{\mathrm{depth}}(h)$ *truncates below depth k and replaces cut subtrees by summary leaves.*

7.1.2 Scheme B: Budget Pruning

Definition 7.7 (Budget Prune). *Given budget B, choose h' obtained by summarizing subtrees so that $C(h') \leq B$ while maximizing retained structure.*

7.1.3 Scheme C: MDL / Description-Length Pruning

Definition 7.8 (MDL Prune). *Given cap L, choose h' with $\ell(h') \leq L$ minimizing information loss under the encoding.*

7.1.4 Scheme D: Stochastic Pruning

Definition 7.9 (Stochastic Prune Kernel). *Retain node u with probability p_u and summarize otherwise; this induces a distribution over pruned histories.*

7.2 Diagram: Four Prune Families as a Single Abstraction Map

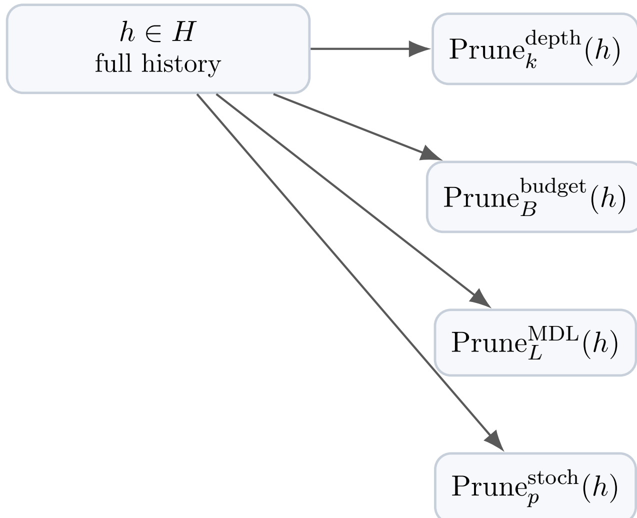

Figure 7.1: Forgetting schemes are different constraints on identity compression.

Exercises (This Chapter)

Exercise 7.1. *Define a continuous family $\mathcal{F}_s$ satisfying $\mathcal{F}_s \circ \mathcal{F}_t = \mathcal{F}_{\min(s,t)}$. Give an explicit construction under MDL pruning by setting the description-length cap $L = s$.*

Exercise 7.2. *For depth pruning $\mathcal{F}_k$, prove monotonicity:*

$$k \leq \ell \implies \mu(\mathcal{F}_k(m)) \leq \mu(\mathcal{F}_\ell(m)).$$

Exercise 7.3. *Show that forgetting is idempotent and value-preserving but not injective. Construct $m \not\equiv_s n$ such that $\mathcal{F}_2(m) \equiv_s \mathcal{F}_2(n)$.*

Exercise 7.4. *Under depth truncation, compute $\mathcal{F}_1(m)$ and $\mathcal{F}_2(m)$ for*

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

Describe the retained skeletons.

Chapter 8

Irreversibility Theorems

Theorem 8.1 (No Cancellation (Structural))**.** *There exists no binary operator $\ominus$ on $\mathbb{M}$ such that for all m, n,*

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

Theorem 8.2 (Strong Irreversibility)**.** *Let $\mathcal{O}$ be any term built from $\oplus, \otimes$ (and constants) but not $\mathcal{F}_k$. Then mass is nondecreasing in each input and strictly increases whenever a join event is introduced.*

8.1 Diagram: Irreversibility as a Strict Mass Arrow

$$\boxed{m} \xrightarrow{\text{compose}} \boxed{m \oplus n}$$

$$\mu(m \oplus n) > \max(\mu(m), \mu(n))$$

Figure 8.1: Strict Growth makes time internal: a theorem, not a rule.

Exercises (This Chapter)

Exercise 8.1. *Model time reversal as an operator R that attempts to invert $\oplus$ structurally. Prove R cannot exist without using $\mathcal{F}$.*

Exercise 8.2. *Prove that no finite sequence of $\oplus$ and $\otimes$ operations can reduce μ.*

Chapter 9

Entropy and the Arrow of Time

Definition 9.1 (Memory Entropy)**.**

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

Theorem 9.2 (Second Law of Arithmetic)**.**

$$\mathcal{S}(m_1 \oplus m_2) > \max(\mathcal{S}(m_1), \mathcal{S}(m_2)).$$

9.1 Thermodynamics Insert

If a history description length $\ell(h)$ is available and forgetting reduces description length by $\Delta\ell$, then

$$Q \geq k_B T \ln 2 \cdot \Delta\ell.$$

9.2 Diagram: Entropy Growth Under Composition

$$\begin{tikzcd} \mathcal{S}(m_1) \arrow[dr] & & \mathcal{S}(m_2) \arrow[dl] \\ & \mathcal{S}(m_1 \oplus m_2) & \end{tikzcd}$$

strictly larger than both inputs

Figure 9.1: Arithmetic Second Law: entropy strictly increases under composition.

Exercises (This Chapter)

Exercise 9.1. *Prove the Second Law using Strict Growth and monotonicity of* $\log(1+x)$*.*

Exercise 9.2. *Assume description length $\ell(h)$ and define erased bits $\Delta\ell$ under pruning. Derive the Landauer-style lower bound on heat Q for forgetting.*

Exercise 9.3. *Show how entropy monotonicity implies a no-cycle theorem for histories under $\oplus$ in the absence of forgetting.*

Exercise 9.4. *Assume MDL pruning. Interpret $\mathcal{S}(m)$ as a proxy for Kolmogorov complexity of history. Discuss limitations.*

Chapter 10

Metric Geometry of Memory

10.1 History Distance

Definition 10.1 (History Edit Distance)**.** *Let* $d_H : H \times H \to \mathbb{R}_{\geq 0}$ *be a metric (or pseudometric), e.g.*

$$d_H(h_1, h_2) := \mathrm{Edit}(h_1, h_2).$$

10.2 Memory Metric

Definition 10.2 (Memory Metric)**.** *Fix* $\lambda \geq 0$ *and define*

$$d(m_1, m_2) = |\,\mathrm{val}(m_1) - \mathrm{val}(m_2)| + \lambda\, d_H(\mathrm{hist}(m_1), \mathrm{hist}(m_2)).$$

Theorem 10.3. *If* d_H *is a metric, then* $(\mathbb{M}, d)$ *is a metric space.*

10.3 Forgetting Profiles

Fix loss $\mathcal{L} : \mathbb{M} \to \mathbb{R}_{\geq 0}$. Define

$$\Phi_m(k) := \mathcal{L}(\mathcal{F}_k(m)), \qquad \Delta\Phi_m(k) := \Phi_m(k+1) - \Phi_m(k).$$

10.4 Operational Curvature

Curvature is detected when two reduction routes at the same budget yield structurally inequivalent results.

10.5 Diagram: Two Reduction Routes (Curvature Witness Pattern)

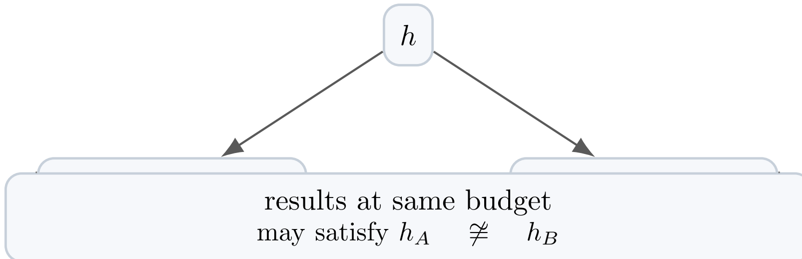

Figure 10.1: Forgetting curvature: non-unique abstraction outcomes at equal budget.

Exercises (This Chapter)

Exercise 10.1. *Prove that if d_H is a metric, then d is a metric.*

Exercise 10.2. *Define $\Phi_m(s) = \mathcal{L}(\mathcal{F}_s(m))$ for a continuous budget s. Show Φ_m is monotone nonincreasing if $\mathcal{L}$ measures retained structure.*

Exercise 10.3. *Construct an explicit example where two reduction routes at the same budget yield non-isomorphic results. Explain what "curvature" means operationally in your example.*

Exercise 10.4. *Show that $\mathcal{F}_k$ is 1-Lipschitz in the history component under depth truncation for a suitable edit metric.*

Chapter 11

Quotients and Classical Recovery

Definition 11.1 (Value Quotient). *Define* $\mathbb{M}_V = \mathbb{M}/\equiv_v$.

Theorem 11.2 (Classical Recovery). *If* val *is surjective onto* V*, then* $\mathbb{M}_V \cong V$ *as commutative monoids (or semirings).*

Exercises (This Chapter)

Exercise 11.1. *Define a quotient type by value equality and show it is isomorphic to the base value type.*

Chapter 12

Algorithms and Computation

Definition 12.1 (Normal Form (One Canonicalization))**.** *A memory number is in* normal form *if no two sibling sub-histories are isomorphic and histories are ordered by a fixed canonical ordering on isomorphism classes.*

Theorem 12.2 (Hardness (Informal but Accurate))**.** *Canonical normal forms for general DAG histories subsume graph isomorphism-like subproblems and NP-hard edit alignment tasks.*

12.1 Diagram: Canonicalization Pipeline

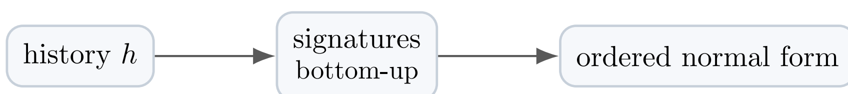

Figure 12.1: Tree canonicalization is feasible; DAG canonicalization is harder.

Exercises (This Chapter)

Exercise 12.1. *Define a canonical ordering on rooted labeled trees and show it yields a deterministic normal form for trees (not DAGs). What breaks for DAGs?*

Exercise 12.2. *Assume H is a tree space (not DAG). Argue why tree isomorphism is decidable in polynomial time. Discuss what changes for general DAG histories.*

Exercise 12.3. *Give a pair of histories where computing d_H (edit distance) differs sharply from computing isomorphism. Explain why identity and distance are different tasks.*

Chapter 13

Games, Dynamics, and Play

Example 13.1 (Irreducible Ten)**.** *From ten genesis atoms, construct m with $\mathrm{val}(m) = 101_V$ maximizing $\mu(\mathcal{F}_3(m))$.*

Example 13.2 (Arithmetic Trauma)**.** *Compare forgetting profiles of a deep chain versus a balanced fold at equal value.*

Exercises (This Chapter)

Exercise 13.1. *Let m be built from n genesis atoms using only $\oplus$. Give tight bounds on the minimal possible depth and maximal possible depth of $\mathrm{hist}(m)$.*

Exercise 13.2. *Assume depth pruning and loss $\mathcal{L}(m) = \mathrm{depth}(\mathrm{hist}(m))$. Compute $\Phi_m(k)$ for a chain and a balanced history at value 2^n and compare asymptotics.*

Chapter 14

Interpretations and Implications

Memory Arithmetic formalizes:

- irreversible computation (history as intrinsic trace),
- provenance-aware systems (auditability as algebra),
- thermodynamic cost of erasure (forgetting as a primitive),
- learning and memory (capacity vs generalization as controlled forgetting).

Exercises (This Chapter)

Exercise 14.1. *Construct two memory numbers with equal value but different memory mass by choosing a mass functional that emphasizes depth.*

Part II

Advanced Structures and Extensions

Chapter 15

Algebraic Structure of Memory Arithmetic

Theorem 15.1. $(\mathbb{M}, \oplus, \otimes)$ *is generally noncommutative and nonassociative structurally; its value-quotient is a commutative semiring isomorphic to* V.

Theorem 15.2. $\mathbb{M}$ *cannot be a ring under any extension preserving Strict Growth (additive inverses would require structural cancellation).*

Exercises (This Chapter)

Exercise 15.1. *Define a structural quotient identifying histories up to permutation of children at* $+$ *nodes. Show* $\oplus$ *becomes structurally commutative. What information is lost?*

Exercise 15.2. *Provide asymptotic bounds (or growth arguments) for the number of distinct structural identities realizing value* n *in the tree model.*

Chapter 16

Category-Theoretic Formulation

Definition 16.1. *Define a category* **Mem***: objects are* $m \in \mathbb{M}$*; morphism* $m \to n$ *exists iff* $\mathrm{hist}(m)$ *embeds as a rooted subgraph of* $\mathrm{hist}(n)$*.*

Theorem 16.2. **Mem** *is thin (a preorder category).*

16.1 Diagram: Thin Category View (Embedding Preorder)

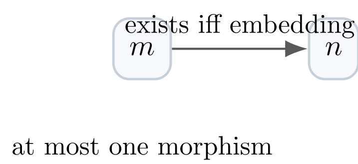

Figure 16.1: **Mem** is thin: morphisms are existence statements.

Exercises (This Chapter)

Exercise 16.1. *Prove that* $\preceq$ *defined by history embedding is a preorder. When is it a partial order?*

Exercise 16.2. *Show that* **Mem** *is thin. Give an example where two different embeddings exist but collapse to one morphism.*

Chapter 17

Stability, Fragility, and Phase Transitions

Definition 17.1 (Fragility Index).

$$\mathfrak{F}(m) := \max_{k \geq 0} \big(\Phi_m(k+1) - \Phi_m(k)\big).$$

Theorem 17.2 (Balanced Histories Reduce Fragility (Structural Statement)). *Under depth-based pruning, balanced histories have lower worst-case fragility than chains for broad loss families aligned with subtree summarization.*

17.1 Diagram: Chain vs Balanced Depth Exposure

chain depth $\Theta(n)$	balanced depth $\Theta(\log n)$
high truncation loss	lower truncation loss

Figure 17.1: Depth pruning punishes chains earlier than balanced forms.

Exercises (This Chapter)

Exercise 17.1. *Show that if histories are trees and pruning is depth truncation, balancing reduces the maximum loss of internal nodes at a fixed depth threshold.*

Exercise 17.2. *Define a geodesic as minimizing expected fragility under stochastic pruning. Propose a proof strategy that balanced trees are near-geodesic for large n.*

Chapter 18

Memory Calculus and Rewrite Dynamics

18.1 Correct "Derivatives": Differentiate Functionals

We differentiate functionals of memory numbers, not memory numbers themselves.

18.2 History Gradient Flows

Let $\mathcal{J}(h)$ be a cost on histories (mass + fragility penalty). A history flow is a sequence $h_{t+1} = \mathcal{T}(h_t)$ that reduces $\mathcal{J}$ while preserving value.

Exercises (This Chapter)

Exercise 18.1. *Construct $F(m) = \alpha\, \mu(m) + \beta\, \mathfrak{F}(m)$ and interpret as an energy. Show forgetting decreases F for suitable parameter regimes.*

Exercise 18.2. *Define a rewrite system that rotates tree structure (AVL-like) and prove value invariance. Interpret this as a "memory gradient step."*

Exercise 18.3. *Discuss how DAG histories complicate geodesic notions and propose a relaxation (tree unfolding or quotient).*

Chapter 19

Memory Physics: Causality, Proper Time, and Free Energy

19.1 Causality as History Inclusion

Define causal order:

$$m_1 \prec m_2 \iff \mathrm{hist}(m_1) \text{ embeds into } \mathrm{hist}(m_2).$$

19.2 Proper Memory Time

$$\tau(m) = \mu(m).$$

19.3 Free Energy (Interpretive)

$$F(m) = \mathrm{val}(m) - T\mathcal{S}(m).$$

19.4 Diagram: Memory Twin Paradox Pattern

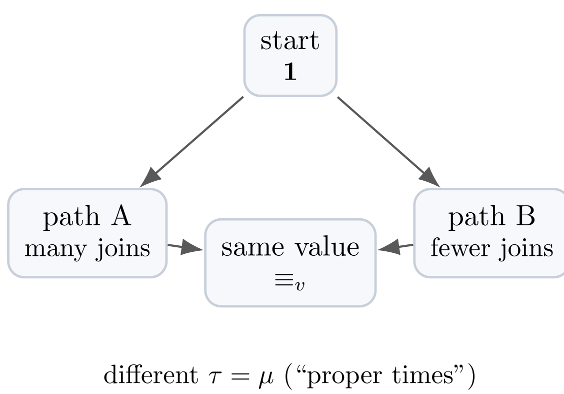

Figure 19.1: Same value, different mass: "memory twin paradox" analogue.

Exercises (This Chapter)

Exercise 19.1. *Define causal order by embedding. Prove transitivity and show how forgetting can violate antisymmetry.*

Exercise 19.2. *Compare two constructions with equal value and interpret* $\Delta\tau$ *as a proper-memory-time difference.*

Exercise 19.3. *Show that at fixed value, decreasing mass decreases free energy* $F(m) = \mathrm{val}(m) - T\mathcal{S}(m)$.

Exercise 19.4. *Explain stochastic pruning as coupling with a thermal bath via a Markov kernel on* $\mathbb{M}$.

Exercise 19.5. *Construct an explicit "memory twin paradox" example: two paths yield the same value but different mass. Explain carefully.*

Exercise 19.6. *Propose a simulation experiment relating mass and fragility over random expression trees; state hypotheses.*

Chapter 20

Formal Verification and Proof Assistants

20.1 Lean/Coq-Style Structure Sketch

```
-- PSEUDO-LEAN (STRUCTURE SKETCH)

structure History := (repr : Nat) -- placeholder; replace with trees/DAGs

structure Mem :=
(val : Real)
(hist : History)

def mu (m : Mem) : Real := (m.hist.repr : Real)

axiom join_add : History -> History -> Mem -> Mem -> History

def madd (m1 m2 : Mem) : Mem :=
{ val := m1.val + m2.val,
  hist := join_add m1.hist m2.hist m1 m2 }

axiom strict_growth : forall m1 m2, mu (madd m1 m2) > max (mu m1) (mu m2)

def forget (k : Nat) (m : Mem) : Mem :=
{ val := m.val, hist := { repr := min m.hist.repr k } }
```

20.2 Diagram: Formalization Layers

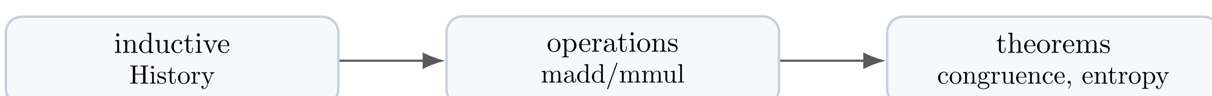

Figure 20.1: Mechanization pipeline: types, operations, theorems.

Exercises (This Chapter)

Exercise 20.1. *Define an inductive type* `History` *for rooted ordered labeled trees. Implement* `joinAdd` *and* `joinMul`*.*

Exercise 20.2. *Prove that value equality is a congruence for* `madd` *and* `mmul`*.*

Exercise 20.3. *Formalize entropy* `S m := log (1 + mu m)` *and prove monotonicity from strict growth.*

Exercise 20.4. *Formalize forgetting as truncation and prove idempotence.*

Exercise 20.5. *State and prove a machine-checkable no-cancellation theorem: no* `sub` *satisfies* `sub (madd m n) n` *is structurally equal to* `m`*.*

Exercise 20.6. *Extract executable code for* `madd`*,* `forget`*, and* `mu`*. Demonstrate with test cases.*

Exercise 20.7. *Implement a canonical-form function for tree histories and prove isomorphic trees share canonical forms.*

Exercise 20.8. *Define a quotient type by value equality and show it is isomorphic to the base value type.*

Exercise 20.9. *Implement stochastic pruning monadically and show it defines a distribution over histories.*

Exercise 20.10. *Prove that forgetting is the only primitive that can reduce mass, given strict growth (no $\mathcal{F}$ in the language).*

Chapter 21

Canonical Forms, Normalization, and Structural Compression

21.1 Motivation

In classical arithmetic, normalization is trivial: numbers reduce to canonical numerals. In Memory Arithmetic, normalization is a nontrivial structural act.

CTA
malization $\neq$
etting.

Normalization seeks a *canonical representative* within a structural equivalence class, while forgetting intentionally destroys identity information.

Definition 21.1 (Structural Normal Form)**.** *A memory number $m = \langle v \mid h \rangle$ is in* structural normal form *if:*

1. *h is minimal with respect to a fixed canonical ordering,*
2. *no two sibling subhistories are isomorphic,*
3. *operation nodes are ordered lexicographically by subtree signature.*

21.2 Tree Signatures

Definition 21.2 (History Signature)**.** *Let $\sigma : H \to \Sigma^*$ be defined recursively:*

$$\sigma(h) = \begin{cases} \texttt{G} & h = genesis, \\ \texttt{+}(\sigma(h_1), \sigma(h_2)) & h = +(h_1, h_2), \\ \texttt{x}(\sigma(h_1), \sigma(h_2)) & h = \times(h_1, h_2), \end{cases}$$

with children ordered lexicographically.

Theorem 21.3 (Canonicalization Correctness)**.** *Two tree histories h_1, h_2 are isomorphic if and only if $\sigma(h_1) = \sigma(h_2)$.*

Proof. By induction on tree height. Leaf case is trivial. Inductive step follows from ordered-child canonicalization. $\square$

21.3 Normalization Algorithm

```
normalize(node):
  children = [normalize(c) for c in children(node)]
  children = unique(children)
  children = sort_by_signature(children)
  return Node(label(node), children)
```

Theorem 21.4 (Termination)**.** *Normalization terminates for all finite tree histories.*

Proof. Recursive descent on finite trees strictly decreases height. □

Exercises (This Chapter)

Exercise 21.1. *Implement normalization for depth-pruned histories and show idempotence.*

Exercise 21.2. *Construct two distinct histories that normalize to the same canonical form. Explain the lost information.*

Chapter 22

Structural Complexity Classes

22.1 Counting Structural Identities

CTA
e grows
arly; identity
vs
binatorially.

Definition 22.1 (Structural Realization Count)**.** *Let $N(n)$ be the number of distinct structural identities (up to $\equiv_s$) realizing value $n1_V$ using only $\oplus$ and $\mathbf{1}$.*

Theorem 22.2 (Catalan Lower Bound)**.**

$$N(n) \geq C_{n-1},$$

where C_k is the kth Catalan number.

Proof. Binary tree shapes with n leaves embed injectively into histories. □

Theorem 22.3 (Superpolynomial Growth)**.** *$N(n)$ grows superpolynomially in n.*

22.2 Structural Complexity Classes

Definition 22.4. *Define:*

- MemP*: histories whose canonical form computable in polynomial time,*
- MemGI*: histories with graph-isomorphism–level complexity,*
- MemNP*: histories requiring combinatorial edit alignment.*

Remark 22.5. *Tree-only Memory Arithmetic lives in* MemP*. DAG histories escalate complexity sharply.*

Exercises (This Chapter)

Exercise 22.1. *Prove that tree normalization is in* P*.*

Exercise 22.2. *Give an example where DAG normalization encodes a known hard problem.*

Chapter 23

Probabilistic Memory Arithmetic

23.1 Stochastic Histories

Definition 23.1 (Random Memory Number). *A* probabilistic memory number *is a distribution*

$$\mathbb{P}(m) = \mathbb{P}(\langle v \mid h \rangle)$$

over histories with fixed value v.

23.2 Expected Mass

Definition 23.2.

$$\mathbb{E}[\mu] = \sum_{h} \mu(\langle v \mid h \rangle)\, \mathbb{P}(h).$$

Theorem 23.3 (Expected Growth). *For independent m_1, m_2,*

$$\mathbb{E}[\mu(m_1 \oplus m_2)] > \max(\mathbb{E}[\mu(m_1)], \mathbb{E}[\mu(m_2)]).$$

23.3 Stochastic Forgetting

Definition 23.4 (Forgetful Kernel). *A Markov kernel $\mathcal{K}_k(h' \mid h)$ defines stochastic pruning:*

$$\mathcal{F}_k(m) = \langle \mathrm{val}(m) \mid h' \sim \mathcal{K}_k(\cdot \mid \mathrm{hist}(m)) \rangle.$$

Theorem 23.5 (Entropy Decrease in Expectation).

$$\mathbb{E}[\mathcal{S}(\mathcal{F}_k(m))] \leq \mathcal{S}(m).$$

Exercises (This Chapter)

Exercise 23.1. *Define a Bernoulli pruning kernel and compute expected retained depth.*

Exercise 23.2. *Show that stochastic forgetting induces a contraction in expected edit distance.*

Chapter 24

Memory Arithmetic as a Rewrite System

24.1 Rewrite Rules

Definition 24.1 (Rewrite Rule)**.** *A rewrite rule is a value-preserving transformation*

$$h \;\Rightarrow\; h' \quad \textit{with} \quad \mathrm{val}(h) = \mathrm{val}(h').$$

Example 24.2. *Tree rotation preserves value but alters structure:*

$$+(a, +(b, c)) \Rightarrow +(+(a, b), c).$$

24.2 Confluence and Nonconfluence

Definition 24.3. *A rewrite system is* confluent *if all reduction paths lead to a unique normal form.*

Theorem 24.4 (Nonconfluence Without Forgetting)**.** *Memory Arithmetic rewrite systems are generally nonconfluent.*

Proof. Distinct structural reductions may preserve value while producing inequivalent histories. □

Exercises (This Chapter)

Exercise 24.1. *Construct two rewrite sequences from the same history that yield non-isomorphic results.*

Exercise 24.2. *Identify a restricted rewrite subset that is confluent.*

Chapter 25

Memory Arithmetic and Information Theory

25.1 History as Code

TA
ories are
ix codes.

Definition 25.1. *A history h induces a codeword $\sigma(h)$. The description length is $\ell(h) = |\sigma(h)|$.*

25.2 Compression Bounds

Theorem 25.2. *For any lossless compressor C,*

$$\ell(C(h)) \geq K(h),$$

where $K(h)$ is Kolmogorov complexity.

Theorem 25.3 (Optimal Forgetting)**.** *MDL pruning approximates minimum-description-length abstraction.*

Exercises (This Chapter)

Exercise 25.1. *Estimate $\ell(h)$ for balanced vs chain histories.*

Exercise 25.2. *Relate forgetting depth k to compression ratio.*

Chapter 26

Learning, Memory, and Generalization

26.1 Learning as History Growth

Definition 26.1. *Learning is a sequence $\{m_t\}$ with strictly increasing $\mu(m_t)$ under value constraints.*

26.2 Overfitting as Excess Memory

Definition 26.2 (Overfitting)**.** *A memory number overfits when $\mu(m)$ grows without corresponding value improvement.*

Theorem 26.3. *Forgetting reduces overfitting by projecting onto lower-mass representatives.*

Exercises (This Chapter)

Exercise 26.1. *Model SGD as incremental history growth.*

Exercise 26.2. *Show how early stopping corresponds to implicit forgetting.*

Chapter 27

Philosophical and Ontological Consequences

27.1 Identity Beyond Equality

CTA
eness has
rs.

Memory Arithmetic formalizes distinctions long treated informally: origin, path, effort, and causality.

27.2 Irreversibility as Fundamental

Time is not imposed on arithmetic; it emerges.

Exercises (This Chapter)

Exercise 27.1. *Compare Memory Arithmetic identity with Leibniz's identity of indiscernibles.*

Chapter 28

Open Problems and Research Directions

1. Classification of minimal-mass realizations.
2. Complete complexity taxonomy of DAG histories.
3. Continuous-time limits of forgetting operators.
4. Physical realization costs of structural mass.
5. Memory Arithmetic–native programming languages.

OCTA Research Note

This chapter is intentionally open-ended. The theory is designed to grow.

Appendix A

Canonical Model (Concrete Realization)

Let H be finite rooted ordered labeled trees with labels in $\{G, +, \times\}$. Let $\cong$ be tree isomorphism preserving labels and left-to-right order.
Define:

- $\mathbf{1} = \langle 1 \mid G \rangle$.
- $m_1 \oplus m_2 := \langle \mathrm{val}(m_1) + \mathrm{val}(m_2) \mid +(h_1, h_2) \rangle$.
- $m_1 \otimes m_2 := \langle \mathrm{val}(m_1) \cdot \mathrm{val}(m_2) \mid \times(h_1, h_2) \rangle$.
- $|h|$: number of internal nodes (or total nodes, fixed choice).
- Prune_k: truncate to depth k and replace cut subtrees with summary leaves.

Theorem A.1. *This model satisfies the core axioms (with $V = \mathbb{R}$).*

Appendix B

Consistency

Theorem B.1. *Memory Arithmetic is consistent relative to the consistency of the underlying value theory (e.g. $\mathbb{R}$).*

Proof. Exhibit the canonical model above. $\square$

Index (Placeholder)

A full index can be generated later once terminology stabilizes.