

The Compiled Self: Irreversible Constraint and the Representation of Binding Systems

A Formal Theory of Interface, Substrate, Policy, and Identity

Flyxion

Abstract

This paper develops a formal theorem chain establishing the structural conditions under which a stable self can exist as a compiled abstraction over irreversible history, and derives the governance implications that follow from those conditions. Beginning with a motivating failure—a Markov name generator whose lexically plausible outputs collapse to generic semantic attractors—we derive four core theorems in strict logical sequence. The Abstraction Theorem establishes that a stable interface exists if and only if a constraint functional enforces global coherence across history. The Substrate Theorem establishes that a symbolic medium is coherent if and only if it preserves spectrally detectable generative distinctions through a paired encoding-decoding structure. The Policy Theorem establishes that viable dynamics require simultaneous constraint preservation and non-collapse of admissible future volume, necessitating a mixture of restrictive and expansive operators. The Main Theorem assembles these into a biconditional: a compiled self exists if and only if all three prior conditions are satisfied jointly and a narrative type system enforces admissibility locally at each transition. The paper develops supporting structures—spectral semantics, an algebra of expansion operators, dependent narrative types, log sovereignty, temporal asymmetry, prompting as field projection, and provenance exposure—and then unifies them in a topos-theoretic framework identifying the self as the fixed point of a constraint-preserving adjunction over the history topos. Existing theories of mind are shown to be partial projections of this invariant. The paper further establishes that binding systems are not finite-type or finite-summary reducible, that approximation errors amplify irreversibly,

and that RSVP continuous-field dynamics naturally instantiate the non-finite-type regime. A Representation Theorem and Minimality Theorem characterize the canonical form of all binding systems. An applied section examines the governance of compiled identity in institutional, labor, and data-justice contexts, showing that the framework’s formal distinctions—between binding and non-binding systems, provenance-stable and provenance-absent representations, and constraint-first versus prior-dominated generation—map directly onto structural failures in deployed AI systems.

Contents

1 Prerequisites: Generative Systems, Constraint, and the Path to This Framework	4
1.1 Computation as Layered Constraint	4
1.2 Text as Universal Substrate	4
1.3 The Markov Generator as Minimal Language Model	5
1.4 The First Experiment and Its Diagnostic Value	6
1.5 Spectral Structure of Generative Processes	7
1.6 The Asymmetry Between Ecosystem Simulation and Self-Maintenance	8
1.7 Policy Selection as Constraint-Preserving Narrative Extension . .	9
2 The Collapse Problem: A Motivating Failure	10
3 Constraint and Interface: The Abstraction Theorem	11
3.1 Preliminaries	11
3.2 Lemma: Instability Under Unconstrained Generation	11
3.3 The Abstraction Theorem	12
3.4 Interpretation	12
4 Encoding and Spectral Invariants: The Substrate Theorem	12
4.1 Preliminaries	13
4.2 Supporting Lemmas	13
4.3 The Substrate Theorem	14
4.4 Interpretation	14
5 Spectral Semantics and Generative Signatures	14
5.1 From Symbol Sequences to Operators	15
5.2 Generative Signatures as Process Invariants	15
5.3 Relation to Compression and Kolmogorov Structure	16
5.4 Spectral Collapse and Semantic Attractors	16
5.5 Interpretation	16
6 Policy, Constraint Load, and Expansion: The Policy Theorem	17
6.1 Preliminaries	17
6.2 Collapse Lemmas	17
6.3 The Policy Theorem	17
6.4 Embedding in RSVP Field Dynamics	18

6.5	Interpretation	18
7	Expansion Operators and the Algebra of Viable Extension	18
7.1	Operators on History Space	18
7.2	Non-Commutativity	19
7.3	Constraint Interaction and Operator Bounds	19
7.4	Operator Balance and Viability	19
7.5	Operator Classification	20
7.6	Interpretation	20
8	The Narrative Type System as a Dependent Type Structure	20
8.1	Types Indexed by History	20
8.2	Identity as a Type-Level Invariant	21
8.3	Type Expansion and Constraint Refinement	21
8.4	The Type Coherence Theorem	21
8.5	Interpretation	22
9	Log Sovereignty and Append-Only Histories	22
9.1	Append-Only Structure and Log Sovereignty	22
9.2	Irreversibility of Violations	22
9.3	Log-Preserving Encoding	23
9.4	The Log Sovereignty Theorem	23
9.5	Interpretation	23
10	Temporal Asymmetry and the First-Person Constraint	24
10.1	Two Modes of Access	24
10.2	Constraint-Restricted Accessibility	24
10.3	The Temporal Constraint Theorem	25
10.4	Connection to Temporal Asymmetry in Philosophy of Mind	25
11	The Compiled Self: Main Theorem	25
11.1	The Narrative Type System	26
11.2	No Component Is Redundant	26
11.3	The Main Theorem	26
11.4	RSVP Embedding	27
11.5	Interpretation	27

12 Prompting as Field Projection: Categorical and RSVP Formulation	27
12.1 The Category of Internal Structures	27
12.2 Rendering, Verification, and the Full Pipeline	28
12.3 RSVP Embedding	28
12.4 The Field Projection Theorem	28
12.5 Interpretation	29
13 Provenance Exposure and the Derivation Certificate	29
13.1 The Provenance Problem	29
13.2 The Derivation Certificate	30
13.3 The Provenance Theorem	30
13.4 Interpretation	30
14 The Self as a Provenance-Stable Interface	30
14.1 Interface as a Fixed Point	31
14.2 Closure Under the Theorem Chain	31
14.3 Failure Modes	32
14.4 Interpretation	32
15 Comparison with Existing Theories	32
15.1 Functionalism as Projection onto Transition Structure	32
15.2 Integrated Information Theory as Projection onto Accessibility Structure	33
15.3 Free Energy Principle as Projection onto Policy Dynamics	33
15.4 Temporal Accounts as Accessibility Constraint	33
15.5 Synthesis	34
16 On the Classification of Expansion Operators	34
16.1 Taxonomy of Expansion Operators	34
16.2 Cross-Domain Composition	35
16.3 Spectral Characterization of Expansion	35
16.4 Expansion Viability Theorem	36
16.5 Interpretation	36
17 Correspondence and Instantiation Across Real Systems	36
17.1 Instantiation in Programming Languages and Compilers	36
17.2 Instantiation in Distributed Systems	37

17.3	Instantiation in Machine Learning and Language Models	38
17.4	Instantiation in Biological Systems	39
17.5	Unified Interpretation	39
18	The Compiled Self in the Human-Machine Era: Governance, Labor, and Data Justice	40
18.1	The Proliferation of Compiled Identity	40
18.2	Institutional Architects and Data Sharing Regimes	41
18.3	Digital Doppelgangers and the Commodification of Provenance	42
18.4	Algorithmic Gatekeeping and the Stakes of Compiled Identity	43
18.5	Workslop, Cognitive Load, and the Degradation of the Generative Substrate	44
18.6	Principles for an Equitable Compiled Self	44
19	Nontrivial Consequences and Falsifiability	46
19.1	The Collapse Constraint	46
19.2	The Expansion-Restriction Instability	46
19.3	Constraint Necessity	47
20	Operational Semantics and Concrete Instantiation	47
20.1	Spectral Signatures as Computable Objects	47
20.2	Constraint Evaluation as Programmatic Verification	48
20.3	Instantiation in Existing Architectures	48
21	Position Relative to Autopoiesis and Embodied Systems	49
21.1	Production and Projection	49
21.2	Interpretation of the Self and Limits of the Framework	49
22	Derived Construction: The Constraint-Coupled Generative Loop	50
22.1	Definition of the Coupled Loop	50
22.2	Non-Decomposability	51
22.3	Derived Prediction	51
23	Irreversible History as a Binding Constraint	51
24	On the Non-Equivalence of Binding and Non-Binding Dynamical Systems	52
24.1	Definitions	52
24.2	Non-Simulability Theorem	53

25 Structural Non-Equivalence via Extension-Monotone Consistency	53
25.1 Global vs Local Constraint Enforcement	53
26 The Finite-Type Boundary and the Scope of Local Constraint Enforcement	54
26.1 Formal Setup	54
26.2 Summary Reduction Criterion	55
27 Examples of Non-Finite-Type Admissibility Structures	56
28 Approximation by Finite Systems and the Collapse of Structural Distinctions	56
28.1 Approximate Admissibility and Uniform Approximation	57
28.2 Horizon Fidelity and Graceful Degradation	57
29 A Concrete Construction: Spectral Constraint-Coupled Generation	58
29.1 Construction	58
29.2 Generative Dynamics and Irreducibility	59
30 Dynamical Instability Under Approximate Admissibility	59
30.1 Error Propagation Under Binding Constraints	60
30.2 Interpretation	60
31 Continuous-Field Realization: Non-Finite-Type Admissibility in RSVP Dynamics	60
31.1 RSVP Field Structure and Global Admissibility	61
31.2 Interpretation	62
32 Comparative Constraint Regimes: Finite-Type Physical Laws and Non-Finite-Type RSVP Dynamics	62
32.1 Finite-Jet Physical Laws	63
32.2 History-Binding Criterion	63
32.3 Interpretation	64
33 Minimality of the Compiled Self	64
34 Representation Theorem for Binding Systems	66
34.1 Statement and Proof	66
34.2 Interpretation	67

35 Topos-Theoretic Unification of Constraint, Spectral Structure, and Narrative Types	67
35.1 The Base Category of Histories	67
35.2 Narrative Types as a Presheaf	68
35.3 Spectral Structure as a Sheaf of Hilbert Spaces	68
35.4 The Bridge: A Natural Transformation	69
35.5 Prompting as an Adjunction	70
35.6 Policy as an Endofunctor	70
35.7 Cross-Domain Operators as Natural Transformations	71
35.8 Spectral Gap as an Operator Condition	71
35.9 Non-Finite-Type Structure as Non-Representability	72
36 The Structural Theorem of the Compiled Self	72
36.1 Statement of the Theorem	73
36.2 Proof	74
36.3 Corollaries	75
36.4 Interpretation	75

1. Prerequisites: Generative Systems, Constraint, and the Path to This Framework

The formal framework developed in this paper did not emerge from pure abstraction. It was built upward through a sequence of technical observations about generative systems, computational abstraction, and the structural differences between ecosystems and selves. This section reconstructs that sequence in objective terms, providing the conceptual prerequisites that the main theorems formalize.

1.1. Computation as Layered Constraint

The computational stack—binary representation, logic gates, boolean algebra, type systems, programming paradigms, user interfaces—is not a sequence of increasingly convenient notations. Each layer is a constrained projection of the layer below it. At the binary level, every operation is explicit and every constraint must be enforced directly. At the level of logic gates, sixteen fundamental boolean operations are realized from NAND or NOR primitives; the architecture constrains what transitions are physically possible. Type systems then restrict which programs can be written at all, enforcing invariants that would otherwise require explicit runtime checks. Programming paradigms—functional, object-oriented, declarative—reorganize constraints into patterns that prevent entire classes of error through structural rather than explicit means. The user interface presents only outputs: the constraint resolution that produced them is invisible.

This observation generalizes far beyond software engineering. Every stable interface over a computational substrate exists because a constraint system below it enforces invariants that prevent incoherent states from reaching the surface. The user sees a coherent result not because the result was produced coherently at all levels simultaneously, but because each intermediate layer filtered out the incoherence native to its substrate. This is the principle that the Abstraction Theorem later formalizes: a stable interface exists if and only if a constraint functional enforces global coherence at each layer.

1.2. Text as Universal Substrate

A natural question arising from this picture is whether any single representational medium could serve as a universal intermediate layer across all the others. The answer is yes, and the medium is text.

Text encoded in ASCII or Unicode is a universal encoding substrate in the

following precise sense. Given any finite symbolic object—a program, a policy, a configuration, a mathematical structure—there exists an encoding of it as a string over a finite alphabet. This is not merely a statement about expressibility; it is a statement about the architecture of computation. Every executable behavior that can be specified can be specified in text, because text is the carrier of the symbolic instructions that interpreters, compilers, and models all consume.

The power of text as a universal substrate is that it enables chaining of transformations between heterogeneous systems. A pipeline of the form $X \rightarrow \text{text} \rightarrow Y$ uses text as a transport layer: the same symbolic object can be interpreted as a program by a compiler, as a policy by a decision system, and as a tool invocation by an orchestration layer. In each case the text is read by a different interpreter that extracts a different structure from the same string. This architecture is not novel—it underlies all of modern software—but its implications for the present framework are significant, because it means that the problem of maintaining coherent identity over time is formally equivalent to the problem of maintaining a coherent text that can be correctly interpreted by all the systems that need to act on it.

However, universality of representation does not imply universality of selection. The Library of Babel contains every possible string over its alphabet, and therefore every possible program, policy, and tool invocation. But nearly all of them are syntactically invalid, semantically incoherent, or operationally useless. The existence of a string in the library does not make it findable, executable, or meaningful. The hard problem is not representation—it is selection under constraint.

1.3. The Markov Generator as Minimal Language Model

A first-order Markov chain over characters, trained on a natural language corpus, is already a minimal language model in the precise technical sense. It defines a probability distribution over strings by factoring the joint probability as a product of conditional probabilities:

$$P(w) = \prod_{t=1}^n P(\ell_t \mid \ell_{t-1}),$$

where the transition matrix $P(\ell_{t+1} \mid \ell_t)$ is estimated from corpus statistics and a twenty-seventh probability mass at each state corresponds to a termination event. The generator is a finite stochastic automaton with an absorbing END state.

This structure produces outputs with genuine phonotactic plausibility. The

learned transitions encode which letter sequences are statistically consistent with English pronunciation, so generated strings feel pronounceable even when they are not recognizable words. The generator trained on Shakespeare and English literature thus produces a specific distribution over Σ^* that is strongly concentrated on phonologically coherent strings while still having positive probability mass across the entire string space.

The key structural observation is that this generator has full support over Σ^* in the same sense that the Library of Babel contains all strings: for any string with no zero-probability transitions, the generator can produce it with nonzero probability. But the probability mass is overwhelmingly concentrated on a small region. More importantly, the generator defines no mapping from its output strings into any space of structured objects with internal invariants. Each generated string is a label—a phonologically shaped token—but it has no internal structure that could serve as the anchor for consistent semantic interpretation. It produces labels without objects.

1.4. The First Experiment and Its Diagnostic Value

The diagnostic value of the failure described in the opening section is precisely that it reveals the boundary between label generation and structured generation. When novel names generated by the Markov chain were presented to an early generative language model with the instruction to describe the philosophical or religious content implied by each name, every name received a description drawn from a small set of semantic attractors: non-theistic ethical traditions centered on harmony, personal growth, environmental stewardship, and compassion. The descriptions were plausible and internally coherent, but they were indistinguishable in structure across all thirty names regardless of phonological character.

The reason is not a limitation of the language model’s intelligence but a structural consequence of the absence of constraint coupling. The language model interprets an unknown token by projecting it onto the nearest region of its learned semantic manifold. Since all the generated names were equally ungrounded—none had any internal structure distinguishing it from the others—they all projected onto the same high-probability region of the manifold: the cluster of concepts most frequently associated with novel philosophical or spiritual terminology in the training data. The model was doing exactly what it should do given the information available to it.

This failure is structurally distinct from a training data limitation or a prompt-

ing failure. No amount of prompt engineering could have produced genuinely distinct outputs, because there was nothing in the inputs to differentiate on. The constraint that would distinguish Klabans from Pfchrvtzrdah or Jolverslperowivicachmebunck would have to come from a structure defined prior to the name—from a parameter space encoding ontological commitments, epistemic orientations, social arrangements, and ritual structures—and that structure would then need to be projected into the name rather than extracted from it after the fact. This is the inversion the Substrate Theorem later captures: a symbolic substrate carries structure faithfully only when paired with a decoder that preserves the relevant distinctions, not when interpretation is delegated to a model operating on a decontextualized string.

1.5. Spectral Structure of Generative Processes

The transition matrix of a Markov generator is not merely a lookup table; it is an operator on distributions over the alphabet. Applying the operator repeatedly produces a trajectory through the simplex of probability distributions over letters, converging toward the stationary distribution of the chain. The eigenstructure of the transition matrix determines the rate of convergence and the shape of the stationary distribution, which in turn determines the statistical character of generated strings.

This algebraic perspective implies that modifying the transition matrix is not merely adding noise—it is changing the spectral structure of the generator. Temperature scaling of the transition probabilities, defined by $P_{ij}^{(T)} = P_{ij}^{1/T} / \sum_k P_{ik}^{1/T}$, sharpens or flattens the distribution while preserving its qualitative structure. As $T \rightarrow 0$ the generator becomes maximally deterministic; as $T \rightarrow \infty$ it approaches a uniform distribution over the alphabet. This is a continuous deformation of the generator within a fixed structural regime.

More radical modifications change the regime itself. Injecting noise directly into the matrix entries—replacing P_{ij} with $P_{ij} + \epsilon_{ij}$ followed by renormalization—introduces incoherence into the transition structure, producing strings that feel like distorted rather than foreign language. Applying a structured modulation such as $P_{ij}^{(\text{mod})} = P_{ij} \cdot (1 + \epsilon \sin(\omega \cdot f(i, j)))$ preserves the qualitative pattern while introducing a systematic periodic deformation, producing strings that feel consistently alien rather than randomly distorted. Rewiring the transition topology entirely—changing which transitions are possible rather than their probabilities—produces a genuinely different phonotactic regime, a different linguistic physics

rather than a variant of the original.

The insight that a text or a generative process has a spectral signature extends to longer sequences and richer representations. Given any mapping from symbols to numerical values—whether raw Unicode codepoints, learned embeddings, or a custom encoding scheme—a text string becomes a finite signal, and its discrete Fourier transform reveals the frequency content of that signal. Concentration of spectral energy in a small number of frequencies indicates repetitive or rhythmically structured generation; a broad flat spectrum indicates high-entropy generation. Two texts with similar spectral signatures share generative regularities at the level of their encoded representation, even if their surface forms differ. This is what the Substrate Theorem formalizes as spectral preservation: a coherent encoding-decoding pair must preserve the spectral distinctions that separate generatively inequivalent histories.

1.6. The Asymmetry Between Ecosystem Simulation and Self-Maintenance

A critical distinction underlying the entire framework is the structural asymmetry between simulating an ecosystem and maintaining a self-narrative. These are not merely tasks of different difficulty; they belong to fundamentally different constraint regimes.

An ecosystem can be modeled as a dynamical system in which each state is a function of the previous state and local interactions. Formally, $x_{t+1} = F(x_t, \text{environment})$. Even when the system is complex and chaotic, it is externally describable and locally updateable. Constraints are local or statistical. Contradictions and inconsistencies can dissipate without structural consequence. The system does not need to maintain a coherent account of its own history. It does not need to be interpretable in hindsight. It does not need to preserve an identity across transformations. Simulating it correctly requires modeling the dynamics of interactions, not the coherence of accumulated commitments.

A self-narrative is a categorically different object. It is not a trajectory through state space but a history $H_t = (\omega_1, \omega_2, \dots, \omega_t)$ subject to a global consistency constraint $\mathcal{C}(H_t) = 1$. Every new event must be consistent with the entire prior history, not merely with the most recent state. This constraint is global, non-Markovian, and grows in complexity with t : each committed event reduces the admissible set of future events, and contradictions accumulate irreversibly. The system cannot simply restart or dissipate an inconsistency; it must account for it. This is what makes self-maintenance harder than ecosystem simulation by a

structural argument rather than a complexity argument.

The further implication is that identity is not a property of a state but an invariant over histories. There exists an invariant I such that $I(H_t) = I(H_{t-1})$ even as H_t grows. The self must change while preserving the thing that defines it. No finite-state or locally-constrained system can enforce this property exactly, because the constraint is global and the history is unbounded. This is the core result that the non-finite-type boundary theorems later establish rigorously.

1.7. Policy Selection as Constraint-Preserving Narrative Extension

The connection between policy selection and narrative maintenance is that they are formally the same problem. A policy in the ordinary decision-theoretic sense maps states to actions: $\pi : x_t \rightarrow \omega_t$. But a policy for a self-maintaining system must map entire histories to actions: $\pi : H_t \rightarrow \omega_t$, subject to the constraint that $\mathcal{C}(H_{t+1}) = 1$. Each action is not merely evaluated for its immediate consequences but for its effects on the accumulated constraint structure and the admissible set of future actions.

This reframing has three consequences that the later theorems develop. First, every action is a projection that collapses part of possibility space: $\Omega(H_{t+1}) \subseteq \Omega(H_t)$. Good policies preserve more possibility space; bad policies collapse it toward contradictions or dead ends. Second, some actions expand possibility space by introducing new representational capacity, restructuring constraints, or adding new operators. These generative actions are structurally rare and expensive because they must remain coherent with accumulated history while genuinely enlarging what is expressible. Third, the optimal policy objective is not reward maximization but the preservation of the largest coherent future manifold consistent with identity: $\max_{\omega_t} |\Omega(H_{t-1} \cup \{\omega\}) \cap \mathcal{I}|$ where \mathcal{I} is the set of identity-preserving trajectories.

The formalisms that follow—constraint functionals, admissibility filtrations, expansion operators, narrative type systems, history toposes—are the mathematical development of these three consequences. The Collapse Problem that opens the main argument, the Markov experiment, is the empirical observation that reveals all three in their absence: a generator without constraint coupling, operating in a flat possibility space, with no expansion operators and no identity to preserve, producing outputs that collapse to the prior distribution of whatever interprets them.

2. The Collapse Problem: A Motivating Failure

We begin not with a theorem but with an experiment. Some years ago, one of us constructed a Markov name generator over a character-level transition matrix derived from English phonotactics. The system was elegant in its simplicity: a twenty-six letter alphabet augmented with a termination state, transition probabilities learned from a corpus, and a sampler that produced outputs with the phonological feel of proper names without being drawn from any known lexicon. The outputs were genuinely novel. Names emerged that belonged to no existing language, yet all felt pronounceable and plausible as designators of persons, places, or systems.

The experiment’s next phase was to feed these names into a language model and request that the model generate descriptions: mythologies, philosophies, cosmologies, the ethical frameworks that such names might plausibly anchor. The results were immediately striking and immediately disappointing. Every name, regardless of its phonological character, attracted descriptions from a small set of semantic attractors. Ancient wisdom traditions appeared repeatedly. Cyclical cosmologies, ethical systems centered on balance or harmony, pantheons organized around natural forces—these dominated across all generated names. The generator had produced genuine lexical novelty. The model had reduced that novelty to a handful of familiar archetypes.

The temptation is to attribute this to limitations of the model, or to insufficient prompting, or to insufficient novelty in the names themselves. We argue instead that the failure was structural and inevitable, and that understanding why it was inevitable is the entry point into everything that follows.

Formally, the Markov generator defines a stochastic process over a character alphabet $\Sigma = \{a, b, \dots, z\} \cup \{\text{END}\}$. Each generated word is a trajectory

$$w = (\ell_1, \ell_2, \dots, \ell_n), \quad \ell_{t+1} \sim P(\cdot \mid \ell_t),$$

where P is the learned transition kernel and termination occurs when the END state is sampled or a maximum length is reached. The generator defines a measure over Σ^* , the space of finite strings over Σ .

What the generator does not define is any mapping from Σ^* into a space of structured objects with internal invariants. It produces labels without objects. When the model receives such a label, it has no paired decoder, no constraint structure, no field of internal relations to recover. It therefore does what it

always does in the absence of structural grounding: it samples from its prior over semantically similar-looking tokens, and that prior is overwhelmingly concentrated on a small number of frequently co-occurring conceptual clusters.

The failure is not that the generator lacked creativity. It lacked coupling. And coupling—the formal relationship between a symbolic surface and the structured object it is meant to carry—is precisely what the following sections develop.

3. Constraint and Interface: The Abstraction Theorem

3.1. Preliminaries

Let Ω denote the space of possible events ω , and let a history be a finite ordered sequence:

$$H_t = (\omega_1, \omega_2, \dots, \omega_t) \in \Omega^*.$$

Let $\mathcal{C} : \Omega^* \rightarrow \{0, 1\}$ be a *constraint functional* such that $\mathcal{C}(H_t) = 1$ indicates that H_t is globally coherent. Coherence is a structural predicate, not a semantic one: it concerns whether the history is internally consistent under the operative formal system.

Let \mathcal{S} denote a space of interface states. An *interface* is a mapping $\mathcal{I} : \Omega^* \rightarrow \mathcal{S}$, required to be stable under extension.

Definition 3.1 (Constraint Preservation). A history evolution preserves constraints if for all t :

$$\mathcal{C}(H_t) = 1 \Rightarrow \mathcal{C}(H_{t+1}) = 1.$$

Definition 3.2 (Interface Stability). An interface \mathcal{I} is stable if there exists a projection operator π such that for all t :

$$\pi(\mathcal{I}(H_{t+1})) = \mathcal{I}(H_t).$$

3.2. Lemma: Instability Under Unconstrained Generation

Lemma 3.3. *If no constraint functional \mathcal{C} exists such that $\mathcal{C}(H_t) = 1$ for all admissible histories, then no stable interface \mathcal{I} exists.*

Proof. Suppose no such \mathcal{C} exists. Then there exist histories H_t and extensions ω_{t+1} such that H_{t+1} may introduce arbitrary contradictions. Assume for contradiction that a stable interface \mathcal{I} exists. Then there must exist π with $\pi(\mathcal{I}(H_{t+1})) = \mathcal{I}(H_t)$. However, since H_{t+1} may arbitrarily violate prior structure, $\mathcal{I}(H_{t+1})$ cannot

simultaneously encode both the previous representation and the contradictory extension without inconsistency. No such π can exist, a contradiction. \square

3.3. The Abstraction Theorem

Theorem 3.4 (Abstraction Theorem). *A stable interface $\mathcal{I} : \Omega^* \rightarrow \mathcal{S}$ exists if and only if there exists a constraint functional \mathcal{C} such that all admissible histories satisfy $\mathcal{C}(H_t) = 1$ for all t .*

Proof. (Necessity.) Assume a stable interface exists. For any extension $H_t \rightarrow H_{t+1}$, the representation must remain consistent under projection. This requires that H_{t+1} cannot introduce arbitrary incompatibilities with H_t , since such incompatibilities would prevent any consistent projection. Therefore a constraint functional \mathcal{C} restricting admissible histories must exist.

(Sufficiency.) Assume \mathcal{C} exists with $\mathcal{C}(H_t) = 1$ for all admissible H_t . Define $H_t \sim H'_t$ iff they are indistinguishable under \mathcal{C} , let $\mathcal{S} = \Omega^*/\sim$, and set $\mathcal{I}(H_t) = [H_t]$. Constraint preservation ensures any admissible extension H_{t+1} remains in the same equivalence class under projection. Hence π exists and \mathcal{I} is stable. \square

3.4. Interpretation

The Abstraction Theorem establishes that abstraction is a structural necessity. A system admits a stable interface if and only if its evolution is constrained so that history remains globally coherent. The interface is the quotient of history space under constraint-induced equivalence: it presents what is preserved and hides what is not. The Markov generator, lacking any \mathcal{C} , cannot produce a stable interface regardless of how rich its outputs are.

Every layer of abstraction in computational systems—binary to logic gates, types to paradigms, paradigms to interfaces—is an instance of this theorem. Each layer is stable precisely because a constraint system operates below it, enforcing invariants that prevent lower-level incoherence from appearing above.

4. Encoding and Spectral Invariants: The Substrate Theorem

Theorem 3.4 established that a stable interface requires a constraint functional. We now ask when a symbolic substrate can carry a constrained history without destroying the structural distinctions that make abstraction possible.

4.1. Preliminaries

An *encoding* is $\mathcal{E} : \Omega^* \rightarrow \Sigma^*$, and a *decoder* is $\mathcal{D} : \Sigma^* \rightarrow \Omega^* \cup \{\perp\}$, where \perp denotes failure.

Definition 4.1 (Signal Realization and Spectral Signature). A signal realization of $w \in \Sigma^*$ is a map $\rho : \Sigma^* \rightarrow \mathbb{R}^n$. The spectral signature of w is $\mathcal{F}(w) = \hat{x}$, the discrete Fourier transform of $\rho(w)$.

The spectral signature is an invariant of generative structure: the recurrent, periodic, and transition-level regularities in the encoded history that reflect the internal organization of the process that produced it.

Definition 4.2 (Generative Equivalence). Two histories $H, H' \in \Omega^*$ are generatively equivalent, $H \equiv_{\mathcal{G}} H'$, if they induce the same constrained transition structure under \mathcal{G} .

Definition 4.3 (Spectral Preservation). An encoding \mathcal{E} is spectrally preserving if $H \equiv_{\mathcal{G}} H'$ implies $\mathcal{F}(\mathcal{E}(H)) = \mathcal{F}(\mathcal{E}(H'))$, and spectral separation of inequivalent histories is maintained on the constrained domain.

4.2. Supporting Lemmas

Lemma 4.4. *If \mathcal{E} is universal but no \mathcal{D} preserving \mathcal{C} exists, then Σ^* cannot function as a coherent substrate.*

Proof. Universality guarantees representability, not coherent reconstruction. If no \mathcal{D} preserving \mathcal{C} exists, then for some $w = \mathcal{E}(H)$, either $\mathcal{D}(w) = \perp$ or $\mathcal{C}(\mathcal{D}(w)) = 0$. The substrate therefore carries syntax without preserving the structure required for interface stability. \square

Lemma 4.5. *If $H \not\equiv_{\mathcal{G}} H'$ but $\mathcal{F}(\mathcal{E}(H)) = \mathcal{F}(\mathcal{E}(H'))$, then the encoding fails to preserve generative distinction.*

Proof. The encoding identifies histories that are distinct at the level relevant to continuation and policy, admitting a many-to-one collapse of generative structure. Encoded histories cannot then support coherent decoding of policy-relevant distinctions. \square

4.3. The Substrate Theorem

Theorem 4.6 (Substrate Theorem). *A symbolic substrate Σ^* functions as a coherent substrate for constrained histories if and only if there exist \mathcal{E} and \mathcal{D} such that: $\mathcal{D}(\mathcal{E}(H)) = H$ up to generative equivalence on the constrained domain; $\mathcal{C}(H) = 1$ implies $\mathcal{C}(\mathcal{D}(\mathcal{E}(H))) = 1$; and \mathcal{E} preserves spectral invariants sufficient to distinguish generatively inequivalent admissible histories.*

Proof. (Necessity.) Coherent substrate use requires that admissible histories remain reconstructible and admissible after encoding and decoding, establishing the first two conditions. If the third failed, then by Lemma 4.5 distinct admissible histories would become indistinguishable in the substrate, making different continuation structures unresolvable.

(Sufficiency.) The three conditions jointly ensure that histories are reconstructible up to generative equivalence, admissibility is preserved across the encoding-decoding boundary, and histories differing in constrained generative behavior remain distinguishable. Therefore the substrate supports stable abstraction. \square

Corollary 4.7 (The Markov Failure). *A Markov name generator does not define a coherent substrate for semantic or policy-relevant histories.*

Proof. It supplies no paired \mathcal{D} preserving constrained generative structure. Conditions one and two of Theorem 4.6 fail; the third holds only at a superficial lexical level. \square

4.4. Interpretation

Universality of representation is insufficient. A substrate is coherent only when it preserves the distinctions relevant for constrained continuation. The Library of Babel contains every possible string, but access to it does not constitute a coherent substrate because no decoder makes recognition possible. Text is valid as a substrate only insofar as it is paired with a decoder and a preserved invariant structure.

5. Spectral Semantics and Generative Signatures

The Substrate Theorem introduced the spectral signature as a detection invariant. In this section we develop spectral structure into a full semantic layer, showing

that it provides a bridge between symbolic representation and generative process that is independent of interpretation.

5.1. From Symbol Sequences to Operators

Let $w \in \Sigma^*$ be an encoded history with signal realization $\rho(w) \in \mathbb{R}^n$. The discrete Fourier transform decomposes the sequence as

$$\rho(w)(t) = \sum_k \hat{x}_k e^{2\pi ikt/n},$$

making explicit that the encoded history is not simply a list of symbols but the trace of a generative process with characteristic frequencies, correlations, and recurrence structures. These are invariant under many transformations that alter surface form while preserving generative organization.

Definition 5.1 (Spectral Equivalence). Two encoded histories $w, w' \in \Sigma^*$ are spectrally equivalent, $w \sim_{\mathcal{F}} w'$, if $\mathcal{F}(w) = \mathcal{F}(w')$ up to a prescribed tolerance.

Spectral equivalence defines a quotient on Σ^* strictly coarser than syntactic equivalence but finer than arbitrary semantic grouping. It partitions encoded histories according to their generative signatures rather than their lexical identity.

Proposition 5.2. *Spectral equivalence respects generative equivalence on the constrained domain if and only if \mathcal{E} is spectrally preserving.*

Proof. If \mathcal{E} is spectrally preserving, generatively equivalent histories map to spectrally equivalent encodings by definition. Conversely, if spectral equivalence respects generative equivalence, the encoding must preserve the invariants determining spectral structure, since otherwise generatively distinct histories would collapse into the same spectral class. \square

5.2. Generative Signatures as Process Invariants

Definition 5.3 (Generative Signature). The generative signature of a process \mathcal{G} is the distribution of spectral signatures

$$\mathcal{S}_{\mathcal{G}} = \{\mathcal{F}(\mathcal{E}(H)) \mid H \in \Omega^*, H \text{ generated by } \mathcal{G}\}.$$

Two processes are generatively indistinguishable on the constrained domain if they induce the same $\mathcal{S}_{\mathcal{G}}$. This allows process identity to be established independent of surface form.

5.3. Relation to Compression and Kolmogorov Structure

The spectral signature provides a computable approximation to deeper notions of structure such as Kolmogorov complexity. While Kolmogorov complexity measures the length of the shortest program generating a string, spectral structure measures the distribution of regularities within it. These are related: highly compressible sequences exhibit concentrated spectral energy while random sequences exhibit approximately uniform spectral distributions.

This connection clarifies the role of spectral analysis in the framework. It is not a substitute for semantic interpretation but a diagnostic of whether a sequence carries structured regularity consistent with a constrained generative process.

5.4. Spectral Collapse and Semantic Attractors

Proposition 5.4. *Prior-dominated rendering induces spectral concentration toward attractor signatures.*

Sketch. In the absence of constraint-dominated projection, the rendering operator samples from high-density regions of the learned manifold. These correspond to recurrent patterns in training data, which exhibit concentrated spectral signatures. The output therefore inherits the spectral characteristics of these attractors rather than those of the input trace. \square

The Markov generator produces strings whose local transition statistics resemble natural language, but whose global generative signatures are shallow. When projected into semantic space without a paired decoder, the rendering operator maps them into attractor regions. Spectrally, this corresponds to a collapse from a broad distribution of low-amplitude features into a concentrated distribution associated with familiar semantic clusters.

5.5. Interpretation

Spectral semantics provides a middle layer between syntax and meaning. It captures structure without requiring interpretation and detects collapse without requiring external validation. It is the invariant that allows the Substrate Theorem to distinguish between representations preserving generative structure and those that do not, enabling us to say in a formally testable way that two encoded histories differ in the organization of their generative processes even when they appear superficially similar.

6. Policy, Constraint Load, and Expansion: The Policy Theorem

With a stable interface and coherent substrate established, we now formalize dynamics. Given a constrained history, what constitutes a viable transition?

6.1. Preliminaries

Define the admissible future space:

$$\Omega(H_t) = \{\omega \in \Omega \mid \mathcal{C}(H_t \oplus \omega) = 1\}.$$

A *policy* is $\pi : \Omega^* \rightarrow \Omega$ with $\pi(H_t) \in \Omega(H_t)$.

Definition 6.1 (Constraint Load). $\kappa : \Omega^* \rightarrow \mathbb{R}_{\geq 0}$ is a monotone functional measuring accumulated constraint tension: $H_t \subset H_{t+1}$ implies $\kappa(H_t) \leq \kappa(H_{t+1})$.

Definition 6.2 (Future Volume). $V(H_t) = |\Omega(H_t)|$, or a suitable measure in continuous settings.

Definition 6.3 (Expansive and Restrictive Actions). An action ω is expansive at H_t if $V(H_t \oplus \omega) \geq V(H_t)$, and restrictive if $V(H_t \oplus \omega) < V(H_t)$.

6.2. Collapse Lemmas

Lemma 6.4. *If a policy selects only restrictive actions, then there exists T such that $V(H_T) = 0$.*

Proof. Each step strictly reduces V , which is bounded below by 0. The strictly decreasing sequence must reach 0 in finite or limit time. \square

Lemma 6.5. *If a policy selects only expansive actions without regard to constraint load, then there exists T such that $\mathcal{C}(H_T) = 0$.*

Proof. Since κ is monotone, unbounded expansion without corrective restriction leads to accumulated constraint violations at some T . \square

6.3. The Policy Theorem

Theorem 6.6 (Policy Theorem). *A policy π is viable over unbounded time if and only if it satisfies both constraint preservation ($\mathcal{C}(H_t) = 1$ implies $\mathcal{C}(H_{t+1}) = 1$) and non-collapse of possibility ($\inf_t V(H_t) > 0$).*

Proof. (Necessity.) Failure of either condition terminates the process or destroys coherence.

(Sufficiency.) Constraint preservation ensures admissibility at all times; non-collapse ensures at least one admissible continuation always exists. The policy can therefore be extended indefinitely. \square

Corollary 6.7. *Any viable policy must contain both restrictive and expansive actions.*

6.4. Embedding in RSVP Field Dynamics

Let (Φ, \mathbf{v}, S) be the RSVP field variables, where Φ encodes density of realized structure, \mathbf{v} encodes flow of transitions, and S encodes entropy. Then $\kappa(H_t) \sim \int S(x, t) dx$ and $V(H_t)$ is measured by admissible configurations under Φ . Restrictive actions correspond to local increases in Φ ; expansive actions correspond to redistribution via \mathbf{v} enlarging admissible configuration space. The Policy Theorem becomes the dynamical statement that viability requires the entropy integral to remain bounded and admissible configuration volume to remain positive.

6.5. Interpretation

Viable dynamics are not characterized by any single optimization criterion but by the simultaneous satisfaction of two structural constraints. Policy is a geometric object: a path on a constrained manifold avoiding both boundary collision (incoherence) and volume collapse (rigidity). Intelligence is not the production of options but the maintenance of a viable trajectory through possibility space.

7. Expansion Operators and the Algebra of Viable Extension

The Policy Theorem established that viable trajectories require both restrictive and expansive actions. We now formalize expansion operators as algebraic objects and analyze their structure.

7.1. Operators on History Space

Let $\mathcal{O} = \{\mathcal{U} : \Omega^* \rightarrow \Omega^*\}$ denote a class of operators on histories.

Definition 7.1 (Admissible Operator). An operator \mathcal{U} is admissible if $\mathcal{C}(H_t) = 1$ implies $\mathcal{C}(\mathcal{U}(H_t)) = 1$ for all H_t .

Definition 7.2 (Expansive and Restrictive Operators). An admissible operator is expansive if $V(\mathcal{U}(H_t)) \geq V(H_t)$ with strict inequality on a positive-measure subset. It is restrictive if $V(\mathcal{U}(H_t)) \leq V(H_t)$ with strict inequality on a positive-measure subset.

7.2. Non-Commutativity

Proposition 7.3. *The algebra of admissible operators is generally non-commutative.*

Proof. \mathcal{U}_1 alters both $\kappa(H_t)$ and $\Omega(H_t)$. Applying \mathcal{U}_2 after \mathcal{U}_1 therefore acts on a different domain than applying \mathcal{U}_2 first. In general $\mathcal{U}_2(\mathcal{U}_1(H_t)) \neq \mathcal{U}_1(\mathcal{U}_2(H_t))$. \square

Non-commutativity is fundamental: the order in which expansions and restrictions are applied matters, and viable dynamics depend on sequencing.

7.3. Constraint Interaction and Operator Bounds

Definition 7.4 (Operator Constraint Increment). $\Delta_\kappa(\mathcal{U}, H_t) = \kappa(\mathcal{U}(H_t)) - \kappa(H_t)$.

Proposition 7.5. *There exists no operator class simultaneously guaranteeing $\Delta_\kappa \leq 0$ and V -expansion over all admissible histories.*

Proof. Increasing V without increasing κ would introduce new admissible continuations without modifying the constraint structure, contradicting the definition of admissibility. Therefore no such operator exists. \square

Expansion is not free. Any increase in admissible future space must be paid for either by increased constraint complexity or by restructuring the constraint system itself.

7.4. Operator Balance and Viability

Theorem 7.6 (Operator Balance). *A sequence (\mathcal{U}_t) of admissible operators defines a viable trajectory if and only if*

$$\sup_t \kappa(\mathcal{U}_t \circ \dots \circ \mathcal{U}_1(H_0)) < \infty \quad \text{and} \quad \inf_t V(\mathcal{U}_t \circ \dots \circ \mathcal{U}_1(H_0)) > 0.$$

Proof. Follows directly from Theorem 6.6 by expressing history evolution as operator composition. \square

7.5. Operator Classification

Although a complete classification is beyond the present work, broad types may be characterized by structural effect. A symbolic operator introduces new representational capacity by embedding Ω^* into a richer structure. A computational operator introduces new transformations whose admissibility derives from closure under composition. A semantic operator restructures the constraint system, replacing \mathcal{C} with an equivalent functional that admits a different but compatible set of continuations. A physical operator corresponds to changes in the material substrate that alter what operations are tractable rather than merely what operations are defined. Real systems operate under compositions of these types, and coherent evolution requires that composed operators preserve constraint compatibility across all domains.

7.6. Interpretation

To build a system remaining viable over long horizons is to design an operator algebra with the right balance properties: operators that open new regions of possibility, operators that consolidate and stabilize structure, and rules governing composition that prevent both collapse and incoherence.

8. The Narrative Type System as a Dependent Type Structure

The Main Theorem will require a narrative type system \mathcal{T} as a filter on admissible continuations. In this section we develop its structure, showing that it is naturally modeled as a dependent type system over histories and that identity is not stored but enforced as a type-level invariant.

8.1. Types Indexed by History

Definition 8.1 (Dependent Narrative Type). A dependent narrative type is a function assigning to each history H_t a type $\text{Type}(H_t)$ such that $\omega \in \text{Type}(H_t)$ implies $\mathcal{C}(H_t \oplus \omega) = 1$.

Proposition 8.2. *The narrative type system is path-dependent.*

Proof. $\text{Type}(H_t)$ depends on the full history H_t , not merely on a current state or summary statistic. Two histories equivalent under some coarse abstraction but dif-

fering in constraint-relevant structure may induce different admissible continuation types. \square

Path dependence is essential. It encodes the irreversibility of history and ensures that identity is not reducible to a state variable independent of its derivation.

8.2. Identity as a Type-Level Invariant

Definition 8.3 (Identity Invariant). An identity invariant is a predicate \mathcal{I}_d over histories such that for all admissible extensions:

$$\mathcal{I}_d(H_t) = 1 \Rightarrow \mathcal{I}_d(H_t \oplus \omega) = 1 \quad \text{for all } \omega \in \text{Type}(H_t).$$

Proposition 8.4. *Identity is equivalent to a fixed point of the type evolution operator.*

Proof. Identity invariance requires that the set of admissible continuations defined by $\text{Type}(H_t)$ preserves \mathcal{I}_d . Therefore \mathcal{I}_d is invariant under the operator $H_t \mapsto \text{Type}(H_t)$, and thus a fixed point of the induced transformation on predicates over histories. \square

8.3. Type Expansion and Constraint Refinement

Proposition 8.5. *Type expansion without constraint refinement leads to inconsistency.*

Proof. If $\text{Type}(H_t)$ is expanded to include new elements without modifying \mathcal{C} , then there exist $\omega \in \text{Type}'(H_t)$ with $\mathcal{C}(H_t \oplus \omega) = 0$, violating the definition of a dependent narrative type. \square

8.4. The Type Coherence Theorem

Theorem 8.6 (Type Coherence). *A narrative type system \mathcal{T} ensures global coherence of a compiled self if and only if it is a dependent type system satisfying type preservation, identity invariance, and compatibility with the constraint functional.*

Proof. (Necessity.) Global coherence requires constraint preservation (establishing dependency on history), interface stability (requiring type preservation), and persistence of identity (requiring identity invariance).

(Sufficiency.) Every continuation selected from $\text{Type}(H_t)$ preserves \mathcal{C} , maintains interface stability, and preserves identity invariants. Hence global coherence is maintained under all admissible extensions. \square

8.5. Interpretation

The narrative type system is the locus where identity, constraint, and policy meet. Identity is not stored but enforced: it is a condition on the evolution of types over history. The self persists not because it is represented somewhere, but because every admissible continuation is constrained to preserve it.

Each layer of abstraction can be understood as a type system over the layer below it, enforcing invariants that make higher-level reasoning possible. The compiled self is the fixed point of this process: a system whose type structure is stable under its own evolution.

9. Log Sovereignty and Append-Only Histories

The preceding sections treated history as a formal object without specifying constraints on its persistence. We now introduce log sovereignty: the condition that history is append-only and all admissible evolution occurs through extension rather than modification.

9.1. Append-Only Structure and Log Sovereignty

Definition 9.1 (Append-Only History). A history space Ω^* is append-only if for all t , the only admissible transition from H_t is $H_{t+1} = H_t \oplus \omega_{t+1}$ for some $\omega_{t+1} \in \Omega$.

Definition 9.2 (Log Sovereignty). A system is log-sovereign if its evolution is defined over an append-only history and all constraint evaluation, policy selection, and type assignment depend only on the accumulated log H_t .

Proposition 9.3. *Log sovereignty implies path dependence of all structural operators.*

Proof. Since H_t cannot be modified, any two histories differing at any position remain distinct for all future time. Any operator relevant to \mathcal{C} , \mathcal{T} , or π must distinguish between them. \square

9.2. Irreversibility of Violations

Proposition 9.4. *Under log sovereignty, constraint violations are irreversible.*

Proof. If $\mathcal{C}(H_t) = 0$, by append-only structure no operation removes or alters the violating event. No extension H_{t+k} can restore $\mathcal{C}(H_{t+k}) = 1$ without redefining the constraint functional. Hence violation is irreversible within the given system. \square

This strengthens the role of the narrative type system: since violations cannot be undone, admissibility must be enforced strictly at each step.

9.3. Log-Preserving Encoding

Definition 9.5 (Log-Preserving Encoding). An encoding \mathcal{E} is log-preserving if

$$\mathcal{E}(H_t \oplus \omega_{t+1}) = \mathcal{E}(H_t) \oplus \mathcal{E}(\omega_{t+1}),$$

where \oplus on the right denotes concatenation in Σ^* .

Proposition 9.6. *If \mathcal{E} is not log-preserving, the encoded substrate cannot support reconstruction of history.*

Proof. Without preservation of concatenation, the encoded representation of $H_t \oplus \omega_{t+1}$ does not contain a recoverable representation of H_t . Decoding cannot reconstruct the original history sequence, violating condition one of Theorem 4.6. \square

9.4. The Log Sovereignty Theorem

Theorem 9.7 (Log Sovereignty). *A compiled self maintains identity over time if and only if its history is append-only and all structural operators are defined over the accumulated log.*

Proof. (*Necessity.*) Modifiable history allows retroactive alteration of identity invariants, and interface stability under projection cannot be guaranteed. If operators do not depend on the full log, path-dependent distinctions may be lost, violating identity preservation.

(*Sufficiency.*) Append-only structure and full-log dependency ensure that identity invariants defined over H_t are preserved under admissible extensions, and interface stability is maintained by projection consistency. \square

9.5. Interpretation

The append-only log is not an implementation detail but a structural requirement. It is the substrate on which constraint, policy, and type interact. Narrative matters because the sequence of events is not interchangeable; different sequences leading to similar states are not equivalent. Verification must occur at each step because errors cannot be undone. Encoding must preserve order because without order, history collapses into an unordered set and identity is lost.

10. Temporal Asymmetry and the First-Person Constraint

Log sovereignty establishes that histories are append-only and evolution proceeds through irreversible extension. This structural asymmetry induces a distinction between two modes of description that is intrinsic to any compiled self.

10.1. Two Modes of Access

A *global* or third-person description treats H_t as a complete object in Ω^* , with all positions equally accessible. A *local* or first-person construction treats H_t as the result of sequential accumulation, accessible only through the prefix $(\omega_1, \dots, \omega_t)$ at time t .

Definition 10.1 (First-Person Constraint). A system satisfies the first-person constraint if all operations available at time t depend only on H_t and do not assume access to any ω_{t+k} for $k > 0$.

Proposition 10.2. *The first-person and third-person descriptions of history are not equivalent under log sovereignty.*

Proof. There exist operations definable over complete histories in the third-person mode requiring access to future elements not available under the first-person constraint. Therefore the two are not equivalent. \square

10.2. Constraint-Restricted Accessibility

Let \mathcal{M} denote the complete model space of all states, and define the accessibility relation at time t :

$$\mathcal{A}_t = \{s \in \mathcal{M} \mid s \text{ is reachable given } H_t\}.$$

Then in general $\mathcal{A}_t \neq \mathcal{M}$, and the accessibility relation is asymmetric: $\mathcal{A}_{t+1} \subseteq \mathcal{A}_t$.

Theorem 10.3 (Temporal Asymmetry). *A system exhibits directed temporal experience if and only if its accessible state space \mathcal{A}_t is strictly decreasing under its update rule.*

Proof. (Necessity.) If \mathcal{A}_t does not decrease, all states remain equally accessible and no distinction between past and future can be constructed.

(Sufficiency.) If $\mathcal{A}_{t+1} \subset \mathcal{A}_t$, the system accumulates irreversible exclusions, inducing an ordering on states. This ordering defines a temporal direction. \square

Corollary 10.4. *The experience of a unique present moment arises from the fact that H_t defines a single maximal consistent history and therefore a single accessibility slice \mathcal{A}_t .*

Corollary 10.5. *Future states are inaccessible because they are elements of \mathcal{M} but not of \mathcal{A}_t .*

10.3. The Temporal Constraint Theorem

Theorem 10.6 (Temporal Constraint). *A compiled self exists only if all structural operators—constraint evaluation, policy selection, and type assignment—are definable under the first-person constraint.*

Proof. (Necessity.) If any operator requires access to future elements, it cannot be executed at time t under append-only dynamics. The system cannot maintain coherence or select admissible continuations, and a compiled self cannot exist.

(Sufficiency.) If all operators are first-person, the system can evaluate constraints, determine admissible types, and select continuations based solely on H_t at each time. Combined with log sovereignty and the previous theorems, this ensures coherent extension over time. \square

10.4. Connection to Temporal Asymmetry in Philosophy of Mind

The distinction between first-person and third-person perspectives is resolved here in structural rather than ontological terms. The first-person perspective corresponds to operating within \mathcal{A}_t ; the third-person perspective corresponds to modeling \mathcal{M} directly. Temporal experience is identical to operating under constraint-restricted accessibility. The apparent divide between subjective and objective descriptions is therefore a difference between local constraint history and global symmetry:

$$\text{time} = \text{asymmetry of accessibility.}$$

No additional ontological entity is required. The appearance of a subject is the appearance of a system maintaining the invariant Φ_{self} under constraint accumulation.

11. The Compiled Self: Main Theorem

We now assemble the prior results into the paper’s central construction.

11.1. The Narrative Type System

Definition 11.1 (Compiled Self). A compiled self over history H_t is a tuple

$$\mathfrak{S} = (\mathcal{I}, \mathcal{E}, \mathcal{D}, \pi, \mathcal{T})$$

where \mathcal{I} is a stable interface, $(\mathcal{E}, \mathcal{D})$ is a coherent encoding-decoding pair, π is a viable policy, and \mathcal{T} is a narrative type system consistent with $(\mathcal{I}, \pi, \mathcal{C})$, such that π selects from $\mathcal{T}(H_t)$ at each step, \mathcal{E} encodes the resulting history, and \mathcal{I} projects it to a stable interface state.

11.2. No Component Is Redundant

Lemma 11.2. *Removing any one of $(\mathcal{I}, \mathcal{E}/\mathcal{D}, \pi, \mathcal{T})$ yields a system that either loses interface stability, collapses generative distinction, fails viability, or admits incoherent continuations.*

Proof. Without \mathcal{I} : arbitrary contradictions accumulate by Theorem 3.4. Without $(\mathcal{E}, \mathcal{D})$: histories differing in admissible continuations become indistinguishable by Theorem 4.6. Without π : pure restriction or expansion leads to collapse by Theorem 6.6. Without \mathcal{T} : the policy may select locally admissible but globally identity-destroying transitions, making Theorems 3.4 and 6.6 simultaneously unmaintainable. \square

11.3. The Main Theorem

Theorem 11.3 (The Compiled Self). *A compiled self \mathfrak{S} exists over a history H_t if and only if: there exists \mathcal{C} with $\mathcal{C}(H_t) = 1$ for all admissible t ; there exist $(\mathcal{E}, \mathcal{D})$ preserving generative distinction under \mathcal{C} ; π satisfies constraint preservation and non-collapse; and there exists \mathcal{T} consistent with $(\mathcal{I}, \pi, \mathcal{C})$.*

Proof. (Necessity.) Each component is individually necessary by the preceding lemma.

(Sufficiency.) Condition (i) and Theorem 3.4 yield \mathcal{I} as the quotient Ω^*/\sim . Condition (ii) and Theorem 4.6 yield a coherent $(\mathcal{E}, \mathcal{D})$. Condition (iii) and Theorem 6.6 yield viable π . Condition (iv) ensures π selects from $\mathcal{T}(H_t)$, preserving interface stability and global admissibility at each step. All components are well-defined and mutually consistent. \square

Corollary 11.4. *A Markov generator cannot constitute a compiled self and cannot be extended into one by surface modification alone.*

Proof. It fails conditions (i), (ii), and (iv). Surface modifications operate entirely within Σ^* without introducing \mathcal{C} , \mathcal{D} , or \mathcal{T} . \square

11.4. RSVP Embedding

The compiled self has a field-theoretic realization. \mathcal{I} corresponds to a stable configuration of $\Phi(x, t)$ that does not disperse. $(\mathcal{E}, \mathcal{D})$ corresponds to transport of structured information along $\mathbf{v}(x, t)$ without destruction of admissible distinction. π corresponds to a trajectory maintaining bounded $\int S(x, t) dx$ while preserving admissible volume. \mathcal{T} corresponds to the local selection rule keeping the trajectory on the constrained manifold jointly defined by Φ and S . Existence is not a static property but an ongoing achievement of constrained dynamics.

11.5. Interpretation

A self is not a substance, process, or narrative in isolation. It is the simultaneous satisfaction of four structural conditions: stable abstraction over constrained history, coherent symbolic transport, viable dynamics over possibility space, and a local selection rule enforcing global admissibility at each step. Generation is not enough; abstraction requires constraint, substrate requires invariant preservation, dynamics require balance, and identity requires a type system enforcing admissibility across time.

12. Prompting as Field Projection: Categorical and RSVP Formulation

We now apply the framework to the practice of constraint-first prompting, showing that it is a structured projection from an internally validated space of constraints into a linguistic rendering manifold.

12.1. The Category of Internal Structures

Let \mathcal{S} be a category of internal structures with objects ω and morphisms $f : \omega \rightarrow \omega'$ that are structure-preserving transformations. A distinguished subcategory $\mathcal{S}_{\text{valid}} \subseteq \mathcal{S}$ contains objects satisfying \mathcal{C} , established prior to linguistic rendering by derivation, code execution, compilation, or invariant preservation.

Definition 12.1 (Prompt Projection Functor). The prompt projection functor $\Pi : \mathcal{S}_{\text{valid}} \rightarrow \mathcal{L}$ maps each valid internal structure to a linguistic partial object, preserving selected structural relations while discarding others.

Proposition 12.2. Π is generally neither faithful nor full.

Sketch. Non-faithfulness: different internal structures may project to prompts with identical surface form. Non-fullness: many rhetorical variations in \mathcal{L} do not correspond to distinct structural morphisms in $\mathcal{S}_{\text{valid}}$. \square

12.2. Rendering, Verification, and the Full Pipeline

The language model induces a stochastic rendering operator $\mathcal{R} : \mathcal{L} \rightsquigarrow \widehat{\mathcal{L}}$ assigning to each prompt a distribution of completions consistent with its learned language manifold. The rendering operator preserves local coherence, not upstream truth.

A rendered output $\hat{\ell}$ is verified if it can be reinterpreted as an object of $\mathcal{S}_{\text{valid}}$ preserving the intended invariants. The full pipeline is:

$$\mathcal{S}_{\text{valid}} \xrightarrow{\Pi} \mathcal{L} \xrightarrow{\mathcal{R}} \widehat{\mathcal{L}} \xrightarrow{\mathcal{V}} \mathcal{S}_{\text{valid}} \cup \{\perp\}.$$

Definition 12.3 (Constraint-Dominated and Prior-Dominated Prompting). An interaction is constraint-dominated if $\Pi(\mathcal{U}_\omega)$ is sufficiently rich that \mathcal{R} remains tethered to the originating coherence region. It is prior-dominated if the projection is too weak, so \mathcal{R} relaxes toward generic manifold attractors.

The opening failure case was prior-dominated: the Markov names projected almost no internal structure, and the model defaulted entirely to prior attractors.

12.3. RSVP Embedding

The internal structure ω corresponds to a localized coherent region $\mathcal{U}_\omega \subset X_t$ in which Φ is high, \mathbf{v} is aligned, and S is suppressed. A prompt is a compressed observable emitted from that region: $\Pi_{\text{RSVP}} : \mathcal{U}_\omega \rightarrow p$. The model propagates this trace through its learned attractor geometry. If the trace is rich, the output remains near \mathcal{U}_ω ; if impoverished, it drifts toward prior attractors.

12.4. The Field Projection Theorem

Theorem 12.4 (Prompting as Field Projection). *Let $\omega \in \mathcal{S}_{\text{valid}}$. Any accepted output $\hat{\ell}$ produced through $\omega \xrightarrow{\Pi} p \xrightarrow{\mathcal{R}} \hat{\ell} \xrightarrow{\mathcal{V}} \omega'$ satisfies the condition that ω' is structurally constrained by ω , even though $\hat{\ell}$ may include compression, rhetorical amplification, and projection-induced smoothing.*

Sketch. Since ω is valid prior to projection and acceptance requires reinterpretation through \mathcal{V} into a compatible valid structure, any accepted output must preserve

sufficient invariant structure to remain within the admissible neighborhood of the originating object. \square

12.5. Interpretation

Prompting is not idea generation. It is field-guided projection from a validated internal structure into language space. The apparent insight of AI-assisted outputs arises because the rendering engine performs high-quality projection-completion over a learned manifold. But the validity, constraint grounding, and epistemic responsibility remain with the agent whose constraint field governed the originating structure. Observers who see only $\hat{\ell}$ may attribute structural organization to the model; the pipeline analysis shows that organization was present upstream before the model was involved.

13. Provenance Exposure and the Derivation Certificate

The rendered output conceals the pipeline that produced it. This section formalizes the problem of provenance exposure and introduces the derivation certificate as the structural object making the internal history legible.

13.1. The Provenance Problem

For any observer without access to ω or Π , the inference problem is ill-posed: $\mathcal{O}(\hat{\ell}) \not\cong \omega$. There exists an equivalence class of possible originating structures:

$$[\omega]_{\mathcal{O}} = \{\omega' \in \mathcal{S} \mid \exists p' : \mathcal{R}(p') = \hat{\ell}\}.$$

Proposition 13.1. *The observable output $\hat{\ell}$ does not uniquely determine its originating structure ω .*

Proof. Follows from the non-faithfulness of Π and the many-to-one nature of \mathcal{R} . \square

Definition 13.2 (Provenance Gap). $\Delta(\hat{\ell}) = \dim(\mathcal{U}_{\omega}) - \dim(\Pi(\mathcal{U}_{\omega}))$.

A theorem has a smaller provenance gap than prose because the formal structure of the statement constrains the space of valid proofs, partially exposing the constraint field.

13.2. The Derivation Certificate

Definition 13.3 (Derivation Certificate). A derivation certificate for $\hat{\ell}$ with originating structure ω is a tuple

$$\mathcal{K}(\hat{\ell}) = (\omega_{\text{sketch}}, \mathcal{C}_{\text{explicit}}, \mathcal{V}_{\text{report}}, \delta)$$

where ω_{sketch} is a compressed representation preserving essential constraints; $\mathcal{C}_{\text{explicit}}$ states the constraints applied, invariants enforced, and failure modes excluded; $\mathcal{V}_{\text{report}}$ records what was checked, rejected, and on what grounds acceptance was granted; and $\delta \in [0, 1]$ is a fidelity estimate.

Definition 13.4 (Structural Author). The structural author of $\hat{\ell}$ is the agent whose \mathcal{C} governed ω , whose Π determined the prompt, and whose \mathcal{V} controlled acceptance. The rendering operator is not a structural author.

13.3. The Provenance Theorem

Theorem 13.5 (Provenance Completeness). *A rendered output $\hat{\ell}$ is epistemically attributable to a structural author if and only if a derivation certificate $\mathcal{K}(\hat{\ell})$ satisfying the minimality conditions exists.*

Proof. (Necessity.) If $\hat{\ell}$ is attributable, then \mathcal{C} , Π , and \mathcal{V} are well-defined and applied, so all four components of $\mathcal{K}(\hat{\ell})$ exist.

(Sufficiency.) ω_{sketch} establishes a constraint-bearing originating structure, $\mathcal{C}_{\text{explicit}}$ establishes upstream validation, $\mathcal{V}_{\text{report}}$ establishes explicit acceptance criteria, and $\delta > 0$ establishes non-trivial fidelity. A structural author therefore exists. \square

13.4. Interpretation

Epistemic attributability is not philosophical but structural: it reduces to whether the upstream constraint history was defined and enforced. The constraint-first pipeline produces outputs that carry legible epistemic structure—outputs whose validity can be traced, whose authorship is unambiguous, and whose provenance gap can be bounded rather than left opaque.

14. The Self as a Provenance-Stable Interface

We now complete the theorem chain by integrating all preceding results into a single canonical object.

14.1. Interface as a Fixed Point

Definition 14.1 (Interface Fixed Point). An interface \mathcal{I} is a fixed point of the structural system if for all admissible histories and admissible extensions,

$$\mathcal{I}(H_t) = \mathcal{I}(H'_t) \quad \text{whenever } H_t \sim_{\mathcal{C}} H'_t.$$

Definition 14.2 (Provenance Stability). A system is provenance-stable if for all admissible H_t , there exists a certificate $\mathcal{K}(H_t)$ such that the structural invariants of H_t are recoverable from $\mathcal{K}(H_t)$ under projection.

Definition 14.3 (Provenance-Stable Interface). A provenance-stable interface is a mapping

$$\mathfrak{S} : \Omega^* \rightarrow (\mathcal{S}, \mathcal{K}), \quad \mathfrak{S}(H_t) = (\mathcal{I}(H_t), \mathcal{K}(H_t)),$$

where \mathcal{I} is a stable interface and $\mathcal{K}(H_t)$ preserves structural invariants of H_t .

Definition 14.4 (Self Equivalence). Histories H_t and H'_t are self-equivalent if $(\mathcal{I}(H_t), \mathcal{K}(H_t)) = (\mathcal{I}(H'_t), \mathcal{K}(H'_t))$.

Proposition 14.5. *Interface equivalence does not imply self equivalence.*

Proof. Since \mathcal{I} is a projection, it may identify distinct histories. If their derivation certificates differ, the histories are not self-equivalent. Identity at the level of appearance is strictly weaker than identity at the level of structure. \square

14.2. Closure Under the Theorem Chain

Theorem 14.6 (Self as Provenance-Stable Interface). *A compiled self exists if and only if there exists a provenance-stable interface \mathfrak{S} satisfying the Abstraction, Substrate, Policy, Type Coherence, Log Sovereignty, and Temporal Constraint theorems.*

Proof. (*Necessity.*) If a compiled self exists, then by Theorem 11.3 all components are present. By Log Sovereignty and Temporal Constraint, its history is append-only and all operations are first-person. By Provenance Completeness, derivation certificates exist. Therefore \mathfrak{S} exists.

(*Sufficiency.*) If \mathfrak{S} exists, then \mathcal{I} provides stable abstraction, \mathcal{K} provides recoverable provenance, and their joint existence implies constrained histories, coherent encoding, viable policy, and a type system. All conditions of Theorem 11.3 are satisfied. \square

The named invariant is:

$$\Phi_{\text{self}} := \text{Inv}(H_t, \Omega_t, \mathcal{A}_t, \mathcal{P}_t).$$

14.3. Failure Modes

The absence of any component produces a distinct collapse regime. If \mathcal{A}_t does not decrease, temporal structure disappears. If $\Omega_t = \emptyset$, the system reaches terminal collapse. If H_t is inconsistent, the interface fragments. If \mathcal{K} is absent, the system loses attributable identity. These are not merely theoretical possibilities but observable structural failure modes.

14.4. Interpretation

A self is the maximal structure that preserves coherent accessibility under irreversible constraint while remaining externally reconstructible. This completes the theorem chain. Constraint ensures coherence of history. Substrate ensures preservation of generative distinction. Policy ensures viable evolution. Type ensures admissibility of transitions. Log sovereignty ensures irreversibility. Temporal constraint ensures realizability. Provenance ensures interpretability. The resulting object is minimal and complete. Intelligence is the problem of selecting expansion operators that preserve Φ_{self} under irreversible constraint.

15. Comparison with Existing Theories

The invariant $\Phi_{\text{self}} = \text{Inv}(H_t, \Omega_t, \mathcal{A}_t, \mathcal{P}_t)$ provides a structural object against which existing theories of mind may be evaluated. Rather than treating them as competing ontologies, we interpret each as a partial projection of this invariant onto a restricted subspace.

15.1. Functionalism as Projection onto Transition Structure

Functionalist theories characterize mental states by their role in mediating transitions between inputs, internal states, and outputs, corresponding formally to a focus on the update operator $\mathcal{U} : s_t \mapsto s_{t+1}$.

Proposition 15.1. *Functionalism is equivalent to a projection of Φ_{self} onto the transition component \mathcal{U} , ignoring H_t and \mathcal{P}_t .*

As a consequence, functional equivalence does not guarantee identity, since distinct histories may induce identical transition behavior. Two systems that compute the same function may nonetheless differ in the constraint history that produced that function, and this difference is invisible to functionalism.

15.2. Integrated Information Theory as Projection onto Accessibility Structure

Integrated Information Theory attempts to quantify the degree to which a system forms a unified whole via measures of informational integration, corresponding structurally to properties of the accessibility slice \mathcal{A}_t , particularly its internal connectivity and inseparability.

Proposition 15.2. *IIT corresponds to a projection of Φ_{self} onto structural properties of \mathcal{A}_t , treating integration as a function of constraint-induced inseparability.*

IIT does not explicitly model the evolution of H_t nor the dynamics of Ω_t , and therefore cannot account for temporal asymmetry as arising from irreversible exclusion. It captures a snapshot of accessibility structure but not the process by which that structure was generated.

15.3. Free Energy Principle as Projection onto Policy Dynamics

The Free Energy Principle models systems as minimizing a variational bound on surprise, maintaining viability through predictive regulation. This corresponds to dynamics over Ω_t governed by constraint-sensitive update rules.

Proposition 15.3. *FEP is equivalent to a projection of Φ_{self} onto policy dynamics over Ω_t , with viability corresponding to maintaining $\Omega_t \neq \emptyset$ under constraint accumulation.*

While FEP captures the necessity of maintaining viable trajectories, it does not formalize provenance \mathcal{P}_t and thus cannot distinguish between internally validated structure and externally indistinguishable output.

15.4. Temporal Accounts as Accessibility Constraint

The distinction between first-person and third-person perspectives, as identified in the philosophy of mind literature, is rooted in the asymmetry of time: the first-person perspective has access only to a single present moment while third-person models treat all moments symmetrically. Within the present framework,

this corresponds exactly to the difference between operating in \mathcal{A}_t and modeling \mathcal{M} .

Proposition 15.4. *Temporal asymmetry in first-person experience is equivalent to the constraint-restricted accessibility condition $\mathcal{A}_{t+1} \subset \mathcal{A}_t$.*

Temporal grounding of first-person experience is thus recovered as a structural property of constraint accumulation rather than requiring a separate ontological category.

15.5. Synthesis

Each theory captures a necessary but insufficient component of Φ_{self} . Functionalism projects onto \mathcal{U} ; IIT projects onto \mathcal{A}_t ; FEP projects onto Ω_t ; temporal accounts project onto H_t via \mathcal{A}_t . None independently reconstruct the full invariant. Only their simultaneous inclusion, together with provenance \mathcal{P}_t , yields a structure sufficient to sustain identity, temporality, and external legibility. The divergence between existing theories is not due to contradiction but to projection: each selects a subspace of the invariant and elevates it to primacy. The present framework identifies the invariant itself, within which these projections coexist as partial views.

16. On the Classification of Expansion Operators

The preceding development establishes that persistence of a self depends on maintaining non-empty possibility space Ω_t under constraint accumulation. The classification of expansion operators defines the forward boundary of the theory: while Φ_{self} characterizes what must be preserved, the space of admissible operators characterizes how evolution may proceed.

16.1. Taxonomy of Expansion Operators

We classify expansion operators according to the domain in which they act and the structure they preserve.

Symbolic operators act over discrete representational systems such as text, code, or formal expressions, preserving syntactic well-formedness and type constraints. They introduce new expressive capacity by embedding the current symbolic domain into a richer one. Examples include grammatical generation, program synthesis, and formal derivation.

Computational operators act over state-transition systems and algorithmic processes, preserving computability and operational semantics. They introduce new transformations whose admissibility derives from closure properties. Examples include search procedures, simulation steps, and optimization updates.

Semantic operators act over meaning-bearing structures, altering interpretive or conceptual space while preserving coherence of interpretation under a constraint decoder. They restructure constraints so that they become less restrictive without being abandoned, introducing meta-rules that subsume prior rules. Examples include analogy, abstraction, and recontextualization.

Physical operators act over material systems and embodied processes, preserving physical law and resource constraints. They correspond to changes in the material substrate that alter what operations are tractable rather than merely what operations are defined.

16.2. Cross-Domain Composition

In general, real systems operate under compositions of these operators:

$$T = T_{\text{phys}} \circ T_{\text{comp}} \circ T_{\text{sym}} \circ T_{\text{sem}}.$$

Proposition 16.1. *Coherent evolution requires that composed operators preserve constraint compatibility across all domains.*

This introduces the requirement of cross-domain alignment: symbolic expansions must remain semantically interpretable, computational expansions must remain physically realizable, and semantic expansions must remain computationally accessible.

16.3. Spectral Characterization of Expansion

Let Ω_t be embedded in a representational space admitting a spectral decomposition. An expansion operator T induces a transformation on the spectrum:

$$\mathcal{F}(\Omega_{t+1}) = \mathcal{T}(\mathcal{F}(\Omega_t)).$$

Definition 16.2 (Spectral Signature of Expansion). The spectral signature of T is the transformation it induces on the distribution of modes in $\mathcal{F}(\Omega_t)$.

Expansive operators tend to introduce higher-frequency components or broaden spectral support; restrictive operators suppress modes or concentrate energy.

16.4. Expansion Viability Theorem

Theorem 16.3 (Expansion Viability). *A sequence of expansion operators (T_t) preserves viability if and only if $\Omega_t \neq \emptyset$ for all t ; T_t remains compatible with H_t ; and the induced spectral transformations do not collapse diversity below a critical threshold.*

Proof. Condition one ensures non-terminal evolution. Condition two ensures coherence with accumulated constraints. Condition three ensures the system does not converge to a degenerate attractor under repeated application of T_t . Together, these guarantee persistence of a non-trivial possibility space. \square

16.5. Interpretation

Expansion operators are the mechanism by which a system avoids collapse under its own constraint accumulation, generating forward structure while respecting past commitments. A complete theory requires not only the identification of invariants but also the classification of transformations that preserve them. The classification presented here is the beginning of that program: a set of structural types whose interaction properties remain to be fully characterized. The most significant open question concerns the conditions under which operators from different domains can be composed without violating cross-domain compatibility, and whether there exist universal expansion operators whose admissibility is domain-independent.

17. Correspondence and Instantiation Across Real Systems

The preceding sections developed a formal structure for constraint, abstraction, substrate, policy, type, and provenance. These objects were introduced abstractly, but they are not confined to abstraction. They appear, often implicitly, in a wide range of real systems. In this section we establish explicit correspondences, showing how the formal components of the framework instantiate in concrete domains. The purpose is not to extend the theory, but to demonstrate that it already governs systems that are independently understood.

17.1. Instantiation in Programming Languages and Compilers

A compiled programming language provides a direct instantiation of the framework. The source program corresponds to a history H_t constructed through append-only edits, typically enforced through version control systems that preserve the log of

changes. The constraint functional \mathcal{C} is implemented by the type system and the compiler’s static checks: a program is admissible if and only if it type-checks and compiles without error.

The interface \mathcal{I} corresponds to the abstract behavior of the program as observed at runtime. Two programs that differ syntactically but compile to equivalent behavior are identified at the interface level, which is precisely the quotient structure described in the Abstraction Theorem.

The encoding-decoding pair $(\mathcal{E}, \mathcal{D})$ is realized by compilation and execution. The compiler encodes high-level structure into a lower-level representation such as bytecode or machine code, and the runtime system decodes it into operational behavior. The Substrate Theorem is instantiated in the requirement that compilation preserve semantic equivalence: distinct programs with distinct behaviors must remain distinguishable after compilation.

Policy π appears as the sequence of development decisions—which transformations to apply, which abstractions to introduce, which constraints to enforce. A viable development process must balance restrictive actions such as adding type constraints with expansive actions such as introducing new abstractions or refactoring to increase flexibility.

The narrative type system \mathcal{T} is literally present as the programming language’s type system, governing admissible transformations at each step and ensuring that extensions preserve invariants required for correct execution. Provenance is captured through build artifacts, test suites, and version histories. A compiled binary without its source or build process lacks provenance and is difficult to attribute or verify; the derivation certificate corresponds to the combination of source code, compilation logs, and test results that establish correctness [25, 26].

17.2. Instantiation in Distributed Systems

Distributed systems provide a second instantiation in which the append-only log is explicit. Systems maintaining histories as sequences of events $H_t = (\omega_1, \dots, \omega_t)$ implement this structure directly.

The constraint functional \mathcal{C} corresponds to consistency conditions such as invariants on replicated state, causal ordering, or consensus requirements [27]. Violations lead to incoherent system states that cannot be reconciled. The interface \mathcal{I} corresponds to the externally observable state of the system, often derived from the log through deterministic replay. Interface stability requires that extending the log does not invalidate previously observed states.

The encoding \mathcal{E} is the serialization of events into messages or log entries, and decoding \mathcal{D} is the reconstruction of state through replay. The Substrate Theorem appears as the requirement that the log preserve enough structure to reconstruct state uniquely and distinguish between different event sequences. Policy π is implemented through scheduling, conflict resolution, and consensus protocols, which must balance restrictive actions such as rejecting conflicting updates with expansive actions such as merging concurrent operations to preserve availability [28].

Provenance is inherent in this setting: the log itself is a derivation record. Systems that discard or compress logs lose the ability to reconstruct state and verify correctness, illustrating the necessity of provenance stability as a structural rather than optional property.

17.3. Instantiation in Machine Learning and Language Models

In machine learning systems, particularly large language models, the framework appears in a more indirect form [29, 30].

The internal structure ω corresponds to a latent representation in embedding space, shaped by training data and model architecture. The constraint functional \mathcal{C} is not explicit but emerges from training objectives and architectural biases, which define what configurations are coherent within the model. The prompt projection Π maps an external input into this latent space, selecting a region of the model’s learned manifold, while the rendering operator \mathcal{R} corresponds to the model’s forward pass.

The substrate is the token sequence space Σ^* , with encoding and decoding implemented through tokenization and detokenization. The Substrate Theorem manifests in the requirement that tokenization preserve distinctions relevant to model behavior; poorly designed tokenization schemes collapse distinctions and degrade performance.

Policy is implicit in sampling strategies such as temperature, top- k , or nucleus sampling, controlling the balance between restrictive and expansive generation. The narrative type system is not explicit in standard models, which explains many observed failure modes: without a type system enforcing admissibility, models may produce outputs that are locally coherent but globally inconsistent. Provenance is largely absent in standard usage, and the framework predicts this as a structural limitation—without provenance exposure, attribution is formally ill-posed.

17.4. Instantiation in Biological Systems

Biological systems provide a final instantiation in which the framework operates at a physical level.

The history H_t corresponds to the sequence of biochemical and developmental events leading to the current state of an organism. This history is effectively append-only, as past states cannot be revisited or altered. The constraint functional \mathcal{C} is realized by physical and biochemical laws, as well as regulatory networks that enforce viability conditions; violations correspond to states incompatible with survival.

The interface \mathcal{I} corresponds to the organism's observable phenotype, which remains stable under continuous internal change. Different internal histories that produce the same phenotype are identified at the interface level. The substrate $(\mathcal{E}, \mathcal{D})$ is realized through genetic encoding and cellular processes that interpret genetic information; the preservation of generative distinction corresponds to the fidelity of replication and expression mechanisms.

Policy π appears as behavior and adaptation, selecting actions that maintain viability while allowing exploration of new states. The narrative type system \mathcal{T} is instantiated in regulatory networks that determine which state transitions are possible given the current configuration, enforcing constraints that preserve identity at the level of the organism. Provenance is encoded in genetic and epigenetic information as well as the organism's developmental history; while not explicitly represented as a certificate, this information determines how the current state arose and constrains future evolution.

17.5. Unified Interpretation

Across all these systems, the same structure appears: a history that is typically append-only, a constraint system enforcing admissibility, a substrate carrying representations that must preserve generative distinction, a policy governing transitions, a type-like structure restricting admissible actions, and a notion of provenance determining how the current state was reached.

The framework does not impose these structures on the systems. It reveals them. Systems that appear different at the surface level share a common architecture at the level of constraint, representation, and evolution. The significance of this correspondence is twofold: it validates the framework by showing that it captures structures already present in well-understood domains, and it provides a language for transferring insights between domains. A failure in one system—loss

of provenance, collapse of generative distinction, unconstrained expansion—can be understood and diagnosed using the same concepts that govern failures in another. The theory is not domain-specific; it is a general description of how structured systems maintain coherence, identity, and viability over time.

18. The Compiled Self in the Human-Machine Era: Governance, Labor, and Data Justice

The formal theory developed in the preceding sections describes the compiled self as a mathematical invariant: the fixed point of a constraint-preserving adjunction over an irreversible history topos. But the structures it formalizes are not merely abstract. They are already being instantiated, incompletely and often without adequate governance, in the institutional systems that govern economic life, social eligibility, and legal standing. This section draws the connection between the formal framework and the emerging practical terrain of compiled identity, arguing that the theory’s central distinctions—between binding and non-binding systems, between provenance-stable and provenance-absent representations, between constraint-first and prior-dominated generation—map directly onto urgent governance failures in the human-machine era.

18.1. The Proliferation of Compiled Identity

As digital infrastructure matures, what was once called a “digital footprint”—a passive trail of past actions—has evolved into something the formal framework would recognize as a compiled self: a synthesized, predictive, and institutionally actionable identity constructed from the convergence of government repositories, biometric monitoring, and machine learning models. The distinction the theory draws between a provenance-stable interface and an interface without derivation certificates applies with full force here. A compiled identity that circulates in institutional systems without recoverable provenance—without a derivation certificate specifying what data was used, under what constraints, verified by what procedure—is formally indistinguishable from an output of the prior-dominated prompting regime: it may appear coherent, but its validity cannot be traced, and errors cannot be diagnosed.

The scale of data generating these compiled identities is large enough that attention itself has become the scarcest resource in the system. When attention is insufficient to validate representations, the compiled identity collapses toward

generic attractors in exactly the way the Collapse Constraint predicts: without constraint coupling between generation and verification, output diversity in data collection does not produce distinguishability in the resulting identity models. The result is a compiled self that is actionable but structurally flawed—high-dimensional in its inputs, low-dimensional in its outputs, and lacking the append-only provenance record that would allow errors to be located and corrected.

18.2. Institutional Architects and Data Sharing Regimes

Governments function as the primary architects of compiled identity because they hold the most comprehensive and nationally representative datasets: health, education, taxation, criminal justice, and social protection. As these repositories are increasingly made available to AI developers, the formal distinction between binding and non-binding data governance becomes practically decisive.

The four principal models of government data sharing can be analyzed in terms of the framework’s core concepts. Open data regimes, which provide unrestricted public access, maximize the generative substrate but impose no constraint coupling between data use and verification of identity validity. They are formally prior-dominated: the interpretation of the released data is determined by the model’s prior rather than by a constraint system enforcing fidelity to the subjects the data represents. Data stewardship models, which manage access through fiduciary intermediaries with explicit portability rights and consent mechanisms, approximate the constraint-coupled regime: the steward functions as a paired decoder enforcing admissibility of use. Public-private partnership models, which involve cross-licensing and infrastructure sharing, introduce the risk of constraint drift: the admissibility functional governing data use may not be preserved across organizational boundaries, leading to the kind of function creep that the binding constraint theorems identify as a structural vulnerability. Contractual models impose explicit admissibility conditions through prescriptive agreements, but without an append-only audit log these agreements are subject to retroactive reinterpretation, violating the log sovereignty condition the theory identifies as necessary for identity preservation.

The Taiwan Health Passbook model illustrates what a constraint-coupled architecture looks like in practice. A user authenticates, selects the scope of data to share, and a server generates an encrypted file decrypted only after an explicit access ticket is granted. This is the structure of a binding system: each step extends the history irreversibly, admissibility is evaluated globally against the accumulated

access log, and the provenance of any derived identity representation is recoverable. The system functions as a facilitator rather than a distributor, maintaining the append-only structure that makes audit and correction possible.

The 2015 transfer of identifiable health data from the UK National Health Service to a private AI developer without explicit patient consent illustrates the failure mode. The transfer relied on implied consent, bypassing the constraint coupling between data use and subject authorization. Regulatory review found failures of data minimization and transparency—formally, a failure of the constraint functional to enforce admissibility of use, combined with the absence of any provenance record allowing the subjects to trace how their compiled identities were derived. The history was not append-only; it was opaque.

18.3. Digital Doppelgangers and the Commodification of Provenance

The most direct instantiation of the compiled self concept in the labor economy is the emergence of what is sometimes called the digital doppelganger: an AI system trained to replicate the decision-making patterns, communication styles, and domain knowledge of a specific human worker, deployed independently to generate value for an employing organization. The formal framework identifies the precise vulnerability in this arrangement.

A digital doppelganger is a compiled self without sovereign provenance. Its derivation certificate—the record of which training data was used, under what constraint functional, verified by what procedure—is held by the organization rather than the human subject. This means that the structural author of the doppelganger’s outputs, in the sense defined by the Provenance Completeness Theorem, is the organization, not the individual whose patterns were compiled. The individual’s epistemic responsibility for the outputs is severed from their economic benefit from those outputs.

Restoring this alignment requires treating the human worker as the structural author in the formal sense: the agent whose constraint field \mathcal{C} governed the originating structure, whose projection functor Π determined the training signal, and whose verification operator \mathcal{V} controls acceptance. Concretely, this means recognizing workers as the primary architects of their doppelgangers with ownership over the training data derived from their labor; implementing licensing structures in which compensation is tied to the frequency and impact of replica deployment; requiring organizations to assume fiduciary responsibility for ensuring doppelgangers do not degrade the professional reputation or future employability of the human subject;

and guaranteeing audit rights over when and how the digital likeness is deployed.

These are not merely ethical positions. They follow from the framework’s Provenance Completeness Theorem: an output is epistemically attributable to a structural author if and only if a derivation certificate exists satisfying minimality conditions. A doppelganger deployed without a recoverable derivation certificate is, by this criterion, not attributable to the human it was trained on—and any institution that deploys it as if it were is making a false attribution.

18.4. Algorithmic Gatekeeping and the Stakes of Compiled Identity

The compiled self is now a gatekeeping mechanism for access to survival resources in some of the highest-stakes institutional contexts. Systems that allocate cash transfers, health benefits, and educational eligibility by processing compiled identities through algorithmic admissibility functions are, in the formal sense, implementing a policy over possibility space—but typically without the constraint-preservation and non-collapse conditions the Policy Theorem identifies as necessary for viability.

When algorithmic systems define poverty by satellite-mapped building density and bank account thresholds, the constraint functional is derived from proxy variables rather than from a direct evaluation of the target condition. The Substrate Theorem applies: if the encoding of a person’s economic situation does not preserve the generative distinctions relevant to admissibility—if two people with genuinely different situations are mapped to the same proxy signature—then the compiled identity collapses their distinct circumstances into a single classification, and the policy built on that classification inherits the error irreversibly.

The safeguards that follow from the framework are specific. A human-in-the-loop mandate corresponds to the verification operator \mathcal{V} : automated decisions must be subject to a reinterpretation step that checks whether the output remains within the admissible neighborhood of the originating structure. Ground-truth verification corresponds to the requirement that the encoding-decoding pair $(\mathcal{E}, \mathcal{D})$ preserve spectral invariants: digital mapping must be cross-checked against direct observation to confirm that the substrate carries the distinctions it is claimed to carry. Variable disclosure corresponds to provenance exposure: the constraint functional and its input variables must be legible to the subjects it classifies, so that errors can be located and the derivation certificate can be partially reconstructed.

18.5. Workstop, Cognitive Load, and the Degradation of the Generative Substrate

The framework identifies a specific failure mode that arises when the generative substrate is contaminated by outputs that lack valid constraint coupling: the Collapse Constraint predicts that a generator without feedback from a constraint system will produce outputs that collapse to generic semantic attractors regardless of input diversity. The organizational phenomenon of low-quality automated output—output generated by over-relying on generative models without verification loops—is the practical instantiation of this prediction.

When organizational workflows are built on such output, the compiled identities derived from them inherit the collapse. An AI system trained on unconstrained generative output learns to model the prior distribution of that output, not the underlying structure the output was meant to represent. The compiled identity becomes a projection of the model’s prior rather than a constraint-coupled representation of the human subject.

The organizational safeguards that follow are structurally motivated. Cognitive load management—limiting the duration and intensity of AI interactions—corresponds to maintaining the human as the structural author of the constraint field: if the human’s capacity to evaluate and verify is degraded, the verification operator \mathcal{V} fails, and the pipeline loses its constraint-coupling at the final step. Quality standards for output—protocols to identify and remove low-value AI-generated content from organizational data pools—correspond to the admissibility filter in the constraint-coupled loop: they prevent contaminated outputs from entering the generative substrate and distorting subsequent compilation. Authenticity audits of candidate profiles correspond to provenance verification: confirming that a compiled representation has a recoverable derivation certificate before treating it as a valid identity.

18.6. Principles for an Equitable Compiled Self

The formal framework’s conclusions about the necessary conditions for a compiled self to exist translate directly into normative principles for the governance of compiled identity. The eight conditions of the Structural Theorem—topos embedding, spectral-logical coherence, adjoint stability, constraint preservation, operator compatibility, spectral viability, non-representability, and irreversible instability—each have a governance analogue.

The requirement that all structural components be internal to a common

topos corresponds to the governance requirement of a unified legal and regulatory framework: compiled identities should not exist in overlapping jurisdictions with incompatible admissibility conditions that cannot be composed. The spectral-logical coherence condition, requiring a natural transformation bridging continuous generative structure to discrete admissibility, corresponds to the transparency requirement: the mapping from raw data to classification must be formally specified and publicly available. The adjoint stability condition, requiring a fixed point of the projection-verification cycle, corresponds to subject participation: individuals must have a mechanism to verify that their compiled identity faithfully represents their originating structure and to correct divergences.

The constraint-preservation condition corresponds to accountability: the organizations deploying compiled identities must be legally responsible for ensuring that admissibility conditions are preserved across all uses. Operator compatibility corresponds to data minimization: cross-domain uses of compiled identity must commute with the original constraint structure, ensuring that expansion into new uses does not break coherence with the original derivation. Spectral viability, requiring that transition operators maintain a spectral gap, corresponds to data equity: compiled identities must preserve the distinctions between subjects that are relevant to the decisions being made, rather than collapsing individuals into indistinguishable classes. Non-representability and irreversible instability together correspond to the requirement of human rights as a foundation: no finite institutional system can fully capture the compiled self of a person, and any system that claims to do so is making a structurally false claim about the nature of human identity.

The deepest conclusion of the formal framework—that the compiled self is the minimal dynamical object capable of sustaining globally coherent, irreversibly bound trajectories beyond the finite-type boundary—implies that any institutional system treating a compiled identity as a complete and sufficient representation of a person is committing a structural error. The compiled self is not a finished object. It is an ongoing achievement of constrained dynamics, one that requires the subject's continuing participation as the structural author of its derivation. Governance frameworks that recognize this—that treat compiled identity as provisional, subject to correction, and requiring continuous consent—are formally correct. Those that treat it as fixed, complete, and institutionally sovereign are not.

19. Nontrivial Consequences and Falsifiability

The preceding development may appear, under superficial reading, to restate familiar intuitions regarding stability, constraint, and balance. This section addresses that concern directly by isolating those claims of the framework that are nontrivial in the precise sense that they exclude otherwise plausible system classes and admit empirical refutation.

19.1. The Collapse Constraint

Let \mathcal{G} be a generative system over Σ^* , and let \mathcal{I} be a family of interpretation operators mapping outputs to semantic representations. We say that \mathcal{G} exhibits *interpretive collapse* if there exists a finite set $A \subset \mathcal{M}$ of semantic attractors such that for a large class of outputs $x \sim \mathcal{G}$, $\mathcal{I}(x) \in A$ despite high syntactic diversity in x .

Definition 19.1 (Constraint-Coupled Generator). A generative system \mathcal{G} is constraint-coupled if there exists a feedback operator $\mathcal{F} : \Sigma^* \rightarrow \mathcal{S}_{\text{valid}}$ such that generated outputs are filtered or adjusted according to a constraint-validity predicate \mathcal{C} over \mathcal{S} .

Theorem 19.2 (Collapse Constraint). *Let \mathcal{G} be a generative system without constraint coupling. Then for any sufficiently expressive interpretation family \mathcal{I} , \mathcal{G} exhibits interpretive collapse.*

Sketch. In the absence of constraint coupling, \mathcal{G} explores Σ^* according only to local statistical structure. Interpretation operators \mathcal{I} project into a lower-dimensional semantic manifold with a finite set of high-probability attractors induced by prior structure. Without feedback enforcing constraint preservation, distinct syntactic realizations map to the same semantic basin. \square

This claim is refuted by exhibiting a generator \mathcal{G} with no constraint feedback whose outputs remain semantically distinguishable under arbitrary interpretation families. The theory thus forbids a class of systems that might otherwise appear viable: purely generative systems that sustain semantic distinction without constraint feedback.

19.2. The Expansion-Restriction Instability

Theorem 19.3 (Expansion-Restriction Instability). *Neither purely expansive nor purely restrictive systems admit stable long-term dynamics preserving both coherence*

and distinguishability.

Sketch. Pure restriction induces monotone contraction of Ω_t , converging to fixed points with minimal distinguishability. Pure expansion induces combinatorial explosion without constraint preservation, causing loss of coherence and effective indistinguishability under projection. Stability requires alternating or coupled operators. \square

This is refuted by exhibiting either a purely expansive system maintaining coherent identity under indefinite iteration, or a purely restrictive system preserving nontrivial distinguishability.

19.3. Constraint Necessity

Theorem 19.4 (Constraint Necessity). *There exists no system simultaneously satisfying persistent semantic distinction, long-term coherence, and absence of constraint feedback.*

This is not a restatement of the claim that systems need rules. It is a restriction on the admissible design space: any counterexample falsifies the framework. The theory asserts that constraint is structurally unavoidable for systems aiming to preserve meaningful distinction over time, and this claim distinguishes the framework from purely descriptive restatements of known intuitions.

20. Operational Semantics and Concrete Instantiation

A second class of objections concerns the apparent lack of computational content. We provide explicit constructions demonstrating how the abstract objects of the framework admit concrete realization in existing systems.

20.1. Spectral Signatures as Computable Objects

Let a sequence $x = (x_1, \dots, x_n)$ be a tokenized representation of text or system output. Define an embedding map $\phi : \Sigma \rightarrow \mathbb{R}^d$ and construct the signal $s_k = \phi(x_k)$. The discrete Fourier transform is then

$$\hat{s}_\omega = \sum_{k=1}^n s_k e^{-2\pi i \omega k / n}.$$

The spectral signature of x is the energy distribution $E(\omega) = |\hat{s}_\omega|^2$. Concentration of energy in a small set of frequencies corresponds to repetitive or

attractor-dominated structure. Broad-spectrum distributions correspond to higher generative variability. This quantity is computable in $O(n \log n)$ time and can be used to detect collapse into semantic attractors, distinguish generative regimes, and track diversity across iterations.

20.2. Constraint Evaluation as Programmatic Verification

Let ω be an internal structure expressed as code, formal specification, or executable process. Define $\mathcal{C}(\omega) = 1$ if execution preserves invariants, and 0 otherwise. Compilation success, unit tests, and invariant checks all instantiate \mathcal{C} . The abstract constraint field therefore corresponds directly to existing verification pipelines.

20.3. Instantiation in Existing Architectures

In a typed programming language, \mathcal{S} corresponds to well-typed programs, \mathcal{C} to the type checker, Π to interface exposure through APIs, and \mathcal{R} to execution or compilation. Ill-typed programs are rejected, demonstrating constraint necessity. Purely generative code without type or runtime checks exhibits precisely the collapse predicted by the Collapse Constraint.

In a typical language model workflow, prompt construction corresponds to Π , model inference to \mathcal{R} , and post-processing or evaluation to \mathcal{V} . Augmented systems with tool use, retrieval, or verification loops introduce explicit constraint coupling, reducing collapse and hallucination. This directly instantiates the Collapse Constraint.

In distributed computation, logs encode irreversible history, consensus protocols enforce constraint consistency, and divergence without constraint leads to fork or failure. This provides a concrete realization of history-dependent coherence preservation.

Embodied systems may be understood as a further instantiation in which constraint evaluation is implemented through physical and metabolic processes rather than symbolic verification. The framework does not attempt to replace biological or phenomenological theories of self; it characterizes a class of systems in which constraint preservation governs stability and distinction, of which biological systems are one instance among several.

21. Position Relative to Autopoiesis and Embodied Systems

A central objection arises from comparison with autopoietic theory, wherein a system is defined by its capacity to produce and maintain its own components and boundary through a network of processes. The present framework must therefore clarify its relation to such systems in order to avoid both overextension and category error.

21.1. Production and Projection

Autopoietic systems are characterized by operational closure: a network of processes that recursively produce the components constituting the network itself, including the boundary distinguishing system from environment. This is a productive condition. The system exists insofar as it continually regenerates its own organization.

The present framework is concerned with projective closure. A system is stable not when it produces its own material components, but when it preserves constraint-consistent structure under projection, transformation, and interpretation. The central objects—constraint functionals, projection operators, and rendering dynamics—do not generate substrate; they regulate admissible structure over an existing substrate.

This distinction is fundamental. Autopoiesis addresses the conditions under which a system is alive. The present framework addresses the conditions under which a system—living or not—can sustain coherent, distinguishable structure over time. The two frameworks are not equivalent and should not be evaluated as competing accounts of the same phenomenon.

21.2. Interpretation of the Self and Limits of the Framework

The notion of a compiled self does not assert that the self is generated purely as a symbolic artifact independent of embodiment. Rather, it identifies an interface layer $\mathcal{I}_{\text{self}} : H \rightarrow \mathcal{S}_{\text{coherent}}$ mapping irreversible history into a structured, constraint-consistent representation accessible to action and interpretation. In an autopoietic organism this interface is implemented through metabolic regulation, neural dynamics, and sensorimotor coupling. In computational systems the same interface is implemented through logs, type systems, and policy selection mechanisms.

The present formalism does not account for metabolic self-production, energetic constraints of physical systems, or affective modulation of behavior. Any claim that

it provides a complete theory of biological selfhood would therefore be unfounded. Its scope is restricted to the analysis of systems in which stability and distinction are mediated by constraint-preserving transformations.

A point of contact lies in the notion of boundary. In autopoiesis, the boundary is physically produced and maintained. In the present framework, the boundary appears as a constraint surface in state space: a subset of admissible configurations preserved under evolution. These are not identical but are related: a physical boundary may be understood as a concrete instantiation of a constraint surface enforced through material processes.

The appropriate relation is therefore one of complementarity rather than competition. Autopoiesis specifies the conditions under which a system exists as a self-producing unity. The present framework specifies the conditions under which such a system—or any system operating over representations—can maintain coherent structure and distinction within its space of possible states. An embodied organism may be simultaneously autopoietic at the level of material organization and constraint-governed at the level of representation and action.

22. Derived Construction: The Constraint-Coupled Generative Loop

The preceding objections correctly identify that existing components—spectral analysis, verification procedures, and historical logging—do not in isolation constitute a novel system. We therefore construct an operator that is not reducible to any of these independently and which follows necessarily from the Collapse Constraint and Constraint Necessity results.

22.1. Definition of the Coupled Loop

Let $\mathcal{G} : \Omega \rightarrow \Sigma^*$ be a generative operator, and let $\mathcal{C} : \Sigma^* \times H \rightarrow \{0, 1\}$ be a constraint predicate dependent on accumulated history H . The Constraint-Coupled Generative Loop is the iterative system:

$$x_{t+1} = \mathcal{G}(\omega_t), \quad \omega_{t+1} = \begin{cases} \mathcal{U}(\omega_t, x_{t+1}) & \text{if } \mathcal{C}(x_{t+1}, H_t) = 1, \\ \mathcal{R}(\omega_t, x_{t+1}) & \text{otherwise,} \end{cases} \quad H_{t+1} = H_t \cdot x_{t+1},$$

where \mathcal{U} is a structure-preserving update operator, \mathcal{R} is a corrective operator, and H_t is an append-only history.

22.2. Non-Decomposability

Proposition 22.1 (Coupling Necessity). *There exists no decomposition of the loop into independent operators \mathcal{G} and \mathcal{C} such that the resulting system preserves both long-term distinguishability of outputs and history-consistent coherence.*

Sketch. If \mathcal{C} is applied independently of history, constraint enforcement becomes memoryless, allowing re-entry into previously invalid states. If \mathcal{G} is independent of constraint feedback, outputs asymptotically collapse under interpretation by the Collapse Constraint. Therefore coupling through shared state (ω_t, H_t) is required. \square

Refining the constraint to include spectral feedback: $\mathcal{C}(x, H) = 1$ iff both semantic invariants are satisfied and $\mathcal{S}(x) \in \mathcal{D}_H$, where \mathcal{D}_H is a history-dependent admissible spectral domain. Spectral structure thus feeds back into admissibility rather than serving merely as a diagnostic.

22.3. Derived Prediction

Theorem 22.2 (Coupled Distinction). *Systems implementing the constraint-coupled loop maintain distinguishable output classes under repeated generation, whereas uncoupled systems converge to a finite set of attractors.*

Given two systems—a pure generator \mathcal{A} and a generator with a constraint-coupled loop \mathcal{B} —under repeated iteration:

$$\lim_{t \rightarrow \infty} |\text{distinct semantic classes}(\mathcal{A})| < |\text{distinct semantic classes}(\mathcal{B})|.$$

This can be evaluated empirically using clustering or spectral divergence measures.

The significance of the loop is not that its components are novel, but that their coupling is required to avoid collapse and is not standard in existing systems, which typically separate generation, evaluation, and memory into independent pipeline stages.

23. Irreversible History as a Binding Constraint

The preceding analysis reveals that any account based solely on feedback, evaluation, or iterative refinement is insufficient to distinguish the framework from standard engineering practice. Systems that permit rollback, overwrite, or local optimization without global commitment remain within the class of non-binding feedback systems.

To escape this equivalence class, the framework must articulate a strictly stronger condition.

A log may be ignored; a binding history cannot. Memory may be selectively accessed or discarded; binding history may not. Reward signals do not enforce global consistency across time; binding history does. Rather, binding history induces a constraint surface in the system's state space that evolves monotonically with t .

Consider a system that generates outputs x_t and evaluates them under predicate \mathcal{C} . In a non-binding system, the acceptance of x_t does not constrain future admissibility beyond local optimization. Contradictions may arise at later times without structural penalty, provided they satisfy the instantaneous criterion. In a binding system, by contrast, acceptance of x_t modifies the admissible region of future outputs such that any x_{t+k} must be jointly consistent with the entire sequence (x_1, \dots, x_t) .

This induces path dependence in a strong sense. Two systems with identical current states but different histories may have disjoint admissible futures. The state of the system is not fully characterized by its present configuration alone; it is indexed by its irreducible provenance.

The necessity of this condition arises from the Collapse Constraint. Systems that do not enforce global consistency across time admit re-entry into previously rejected or incompatible states. Over extended operation this produces either contradiction in symbolic systems or attractor collapse in generative systems.

In computational systems, binding may be implemented through append-only logs combined with constraint propagation. In formal systems, it corresponds to monotonic extension of theories without contradiction. In distributed systems, it appears as consensus protocols enforcing consistency across replicas. In biological systems, it is instantiated through irreversible biochemical and structural changes. The common feature is not the mechanism but the enforcement of consistency across accumulated history.

24. On the Non-Equivalence of Binding and Non-Binding Dynamical Systems

24.1. Definitions

Let \mathcal{X} be a measurable state space and $\mathcal{H} = \bigcup_{t \geq 0} \mathcal{X}^t$ the space of finite histories. A non-binding system is specified by a tuple $(\mathcal{S}, \sigma, T, \mathcal{C}_0)$ where $\sigma : \mathcal{H} \rightarrow \mathcal{S}$ is

a summary map, $T : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{X})$ a transition kernel, and $\mathcal{C}_0 : \mathcal{X} \times \mathcal{S} \rightarrow \{0, 1\}$ an admissibility predicate. Evolution is via $s_t = \sigma(H_t)$, $x_{t+1} \sim T(s_t)$, with admissibility factoring through σ .

A binding system is specified by $(\mathcal{X}, \mathcal{C})$ with $\mathcal{C} : \mathcal{X} \times \mathcal{H} \rightarrow \{0, 1\}$ such that $\mathcal{A}(H_{t+1}) \subseteq \mathcal{A}(H_t)$ where $\mathcal{A}(H_t) = \{x \in \mathcal{X} : \mathcal{C}(x, H_t) = 1\}$. No factorization through a finite summary σ is assumed.

Definition 24.1 (Simulation). A non-binding system simulates a binding system if there exists $\iota : \mathcal{X} \hookrightarrow \mathcal{S}$ such that $\mathcal{A}(H_t) = \{x \in \mathcal{X} : \mathcal{C}_0(x, \sigma(H_t)) = 1\}$ for all H_t .

24.2. Non-Simulability Theorem

Theorem 24.2 (Non-Simulability of Binding Systems). *Let $(\mathcal{X}, \mathcal{C})$ be a binding system with strictly decreasing admissible sets $\mathcal{A}(H_1) \supsetneq \mathcal{A}(H_2) \supsetneq \dots$, and suppose for each t there exists $x_t \in \mathcal{A}(H_{t-1}) \setminus \mathcal{A}(H_t)$. Then no non-binding system with finite-dimensional state space \mathcal{S} can simulate $(\mathcal{X}, \mathcal{C})$.*

Proof. Assume such a simulation exists. The mapping $\sigma : \mathcal{H} \rightarrow \mathcal{S}$ induces finitely many distinct states. Since $\{\mathcal{A}(H_t)\}$ is strictly decreasing and infinite, there exist $t \neq t'$ with $\sigma(H_t) = \sigma(H_{t'})$. By simulation, $\mathcal{A}(H_t) = \mathcal{A}(H_{t'})$. But by assumption $\mathcal{A}(H_t) \neq \mathcal{A}(H_{t'})$, a contradiction. \square

Theorem 24.3 (Trajectory Separation). *Let \mathcal{T}_B and \mathcal{T}_N denote admissible trajectories under binding and non-binding systems respectively. Then $\mathcal{T}_B \not\subseteq \mathcal{T}_N$ for any non-binding system with bounded state representation.*

25. Structural Non-Equivalence via Extension-Monotone Consistency

The memory-based non-simulability argument is susceptible to the objection that the distinction reduces to a quantitative capacity gap rather than a structural one. We therefore establish a strictly stronger form of non-equivalence.

25.1. Global vs Local Constraint Enforcement

Let $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \{0, 1\}$ be a compatibility relation. Define global consistency of a history $H_t = (x_1, \dots, x_t)$ as $\kappa(x_i, x_j) = 1$ for all $i, j \leq t$.

Definition 25.1 (Extension-Monotone System). A system is extension-monotone if for every admissible extension x_{t+1} , $\kappa(x_{t+1}, x_i) = 1$ for all $i \leq t$, and this condition is preserved for all future extensions.

A system is k -locally constrained if admissibility is determined by $\mathcal{C}_k(x_{t+1}, x_{t-k+1}, \dots, x_t)$ for some finite k .

Theorem 25.2 (Violation under Finite-Window Constraints). *Let a system enforce constraints only over windows of size k . Then there exist sequences $\{x_t\}$ such that $\kappa(x_i, x_j) = 1$ for all $|i - j| \leq k$, but $\exists i, j$ with $|i - j| > k$ such that $\kappa(x_i, x_j) = 0$.*

Proof. Construct a sequence where each consecutive block of length k is mutually compatible, but introduce incompatibility between distant elements. This is always possible when κ is not transitive over arbitrarily long chains. \square

Theorem 25.3 (Structural Non-Simulability). *No system enforcing only local or finite-window constraints can simulate an extension-monotone system enforcing global consistency, even with arbitrarily large but finite memory.*

Proof. By the preceding theorem, there exist histories for which all local constraints are satisfied while global consistency is violated. Therefore the simulation fails. \square

The distinction does not arise from memory size. A system may store the entire history but fail to enforce global consistency if constraint evaluation is local or approximate. The distinction concerns the form of constraint enforcement, not the amount of stored information.

26. The Finite-Type Boundary and the Scope of Local Constraint Enforcement

The preceding sections establish a structural boundary between globally constrained and locally constrained systems. The appropriate conclusion is not that local systems are in principle incapable of global consistency, but that such consistency is achievable locally only under a specific structural condition.

26.1. Formal Setup

Let $\mathcal{L} \subseteq \mathcal{H}$ be a prefix-closed admissibility language.

Definition 26.1 (k -Local Admissibility). \mathcal{L} is k -local if there exists a set $F \subseteq \mathcal{X}^{\leq k}$ of forbidden words such that $H \in \mathcal{L}$ iff no contiguous subword of H belongs to F .

Theorem 26.2 (Finite-Type Boundary). *The following are equivalent: there exists a bounded-window constraint system deciding admissibility using only the most recent k symbols; \mathcal{L} is k -local for some finite k ; the admissibility structure of \mathcal{L} is representable as a safety language of finite type.*

Proof. Each direction follows from the definition. A bounded-window system of width k yields a set F of forbidden length- k patterns defining k -locality. Conversely, k -locality induces a bounded-window system by checking the final k positions at each step. The equivalence with finite-type safety languages is standard. \square

Definition 26.3 (Non-Finite-Type Consistency). \mathcal{L} is non-finite-type if for every finite k there exist $H, H' \in \mathcal{H}$ agreeing on all contiguous subwords of length $\leq k$ but with exactly one belonging to \mathcal{L} .

Theorem 26.4 (Failure of Local Reduction). *If \mathcal{L} is non-finite-type, no bounded-window local constraint system can enforce its admissibility.*

Proof. Suppose toward contradiction that a bounded-window system of width k enforces \mathcal{L} . By the Finite-Type Boundary theorem, \mathcal{L} would be k -local. But if \mathcal{L} is non-finite-type, there exist histories for this same k indistinguishable by all contiguous subwords of length $\leq k$, with one admissible and one not. This contradicts k -locality. \square

26.2. Summary Reduction Criterion

Let $\sigma : \mathcal{H} \rightarrow \mathcal{S}$ be a finite summary map. Say σ is admissibility-complete for \mathcal{L} if $\sigma(H) = \sigma(H')$ implies $\mathcal{A}(H) = \mathcal{A}(H')$.

Theorem 26.5 (Summary Reduction Criterion). *A finite summary map σ is admissibility-complete iff the admissibility language \mathcal{L} factors through a finite right-congruence on prefixes.*

Proof. If σ is admissibility-complete, equality of summaries defines an equivalence with finite index under which continuation sets are identical—a finite right-congruence. Conversely, any finite-index right-congruence yields a summary map by sending each history to its equivalence class. \square

Theorem 26.6 (Structural Non-Equivalence). *If \mathcal{L} is neither of finite type nor representable by a finite right-congruence, then no bounded local system and no finite summary system can enforce \mathcal{L} exactly.*

This establishes a genuine trichotomy. First, admissibility structures that are finitely local are fully reducible to local constraint enforcement. Second, non-local but regular structures admit finite sufficient summaries. Third, structures that are neither finitely local nor finitely summarizable are genuinely history-binding and cannot be reduced without loss. The nontrivial content of binding systems begins exactly where admissibility ceases to be of finite type and ceases to admit finite right-congruence reduction.

27. Examples of Non-Finite-Type Admissibility Structures

The claim that admissibility may lie beyond finite-type reducibility is not merely formal.

Balanced dependency structures provide the simplest instance. The language $\mathcal{L} = \{a^n b^n \mid n \geq 0\}$ is not of finite type and not regular. No bounded window suffices, as the legality of a symbol b depends on the total number of preceding a symbols, requiring unbounded counting. No finite summary can preserve admissibility exactly.

Cross-serial dependencies provide a more complex instance. Languages constrained by pairwise matching between corresponding indices across sequences exceed context-free expressivity and are not reducible to finite-type constraints. Their admissibility depends on maintaining alignment across arbitrarily distant elements, demonstrating that non-finite-type admissibility arises naturally in systems requiring long-range coordination.

Clique consistency in growing structures gives a third instance. If each state x_i corresponds to a node with compatibility κ , and admissibility requires that the induced graph on H_t be a clique, this condition is not of finite type unless κ satisfies strong transitivity. Verification requires checking compatibility across all pairs and cannot be reduced to bounded local constraints. This structure is directly relevant to constraint systems requiring universal pairwise coherence across history.

An apparent counterexample to the necessity of non-finite-type structure is provided by biological systems, which exhibit global coherence despite local interaction rules. The resolution lies in the distinction between discrete admissibility languages and continuous dynamical systems. Biological systems operate through continuous state variables, fields, and gradients, not through discrete symbolic forbidden patterns. Global coherence is maintained through continuous dynamical invariants rather than discrete admissibility constraints. Such systems inhabit a distinct computational substrate and do not constitute counterexamples to the formal boundary established here.

28. Approximation by Finite Systems and the Collapse of Structural Distinctions

Even if a system lies above the finite-type boundary, it may be approximable to arbitrary precision by finitely reducible systems. In such a case the structural distinction would persist formally but collapse operationally. This section analyzes

the extent to which non-finite-type admissibility admits finite approximations, and under what conditions such approximations preserve or destroy qualitative system behavior.

28.1. Approximate Admissibility and Uniform Approximation

Let $\sigma : \mathcal{H} \rightarrow \mathcal{S}$ be a finite summary. We say it induces an ε -approximate admissibility system if there exist approximate continuation sets $\mathcal{A}_\sigma(H)$ with $d(\mathcal{A}(H), \mathcal{A}_\sigma(H)) \leq \varepsilon$ for all $H \in \mathcal{L}$, under a suitable divergence measure d . An admissibility structure is uniformly approximable if for every $\varepsilon > 0$ there exists a finite σ achieving ε -approximation uniformly over all histories.

Definition 28.1 (Admissibility Entropy). $h = \limsup_{t \rightarrow \infty} \frac{1}{t} \log N_t$, where N_t is the number of distinct admissibility sets $\mathcal{A}(H_t)$ realized over all histories of length t .

Theorem 28.2 (Non-Uniform Approximability under Positive Entropy). *If $h > 0$, then there exists $\varepsilon_0 > 0$ such that no finite summary map achieves uniform ε_0 -approximation over all histories.*

Proof. Positive entropy implies exponential growth in distinguishable admissibility sets. Any finite summary induces finitely many equivalence classes. By the pigeonhole principle, sufficiently many distinct admissibility sets must be mapped to the same class, producing a uniform lower bound on approximation error. \square

28.2. Horizon Fidelity and Graceful Degradation

A finite system achieves T -fidelity if $\mathcal{A}_\sigma(H) = \mathcal{A}(H)$ for all $|H| \leq T$. For any non-finite-type structure, T -fidelity is achievable for each finite T , but no finite system achieves it for all T simultaneously.

The practical significance of non-uniform approximability depends on whether approximation error produces qualitative failure or quantitative degradation. In systems requiring strict global coherence, with irreversible commitments, or in which early violations propagate and amplify, even small approximation errors accumulate into structural divergence. In such regimes, finite approximations fail to preserve system behavior.

Finite approximation may be interpreted as a renormalization of admissibility structure. Histories are coarse-grained into equivalence classes, and admissibility is evaluated at the level of classes rather than individual trajectories. The loss

incurred is precisely the loss of distinctions between histories whose admissibility differs only at scales beyond the resolution of the summary.

The structural boundary identified in the preceding sections therefore corresponds to a transition between regimes in which finite approximation is sufficient and regimes in which it is intrinsically lossy. The critique that finite systems suffice in practice is correct for a broad class of systems but not for all. The present framework identifies precisely those regimes in which such sufficiency breaks down.

29. A Concrete Construction: Spectral Constraint-Coupled Generation

The preceding analysis establishes that the non-finite-type regime is not vacuous. To prevent the framework from remaining purely classificatory, this section exhibits an explicit generative construction whose admissibility relation provably lies beyond finite-type and finite-summary reduction, and connects directly to the spectral machinery developed earlier.

29.1. Construction

Let \mathcal{X} be a finite alphabet and $\phi : \mathcal{X} \rightarrow \mathbb{R}^d$ an embedding. For a history $H_t = (x_1, \dots, x_t)$, define the cumulative spectral signal

$$S_{H_t}(\omega) = \sum_{k=1}^t \phi(x_k) e^{-i\omega k}, \quad \omega \in \Omega \subset \mathbb{R},$$

and the energy distribution $E_{H_t}(\omega) = |S_{H_t}(\omega)|^2$. Define admissibility by

$$H_t \in \mathcal{L} \iff \int_{\Omega} F(E_{H_t}(\omega), \omega) d\omega \leq C,$$

for a nonlinear functional F and constant $C > 0$.

Theorem 29.1. *The admissibility language \mathcal{L} defined above is not of finite type.*

Proof. Assume \mathcal{L} is k -local. Then admissibility depends only on contiguous substrings of length at most k . However, $E_{H_t}(\omega)$ depends on contributions from all positions via Fourier interference. Construct two histories H and H' with identical substrings of length $\leq k$ but with global phase alignments differing sufficiently to produce distinct spectral energy distributions, one satisfying the constraint and one not. This is always possible due to the nonlocal nature of Fourier interference. Hence \mathcal{L} is not k -local for any finite k . \square

Theorem 29.2. *No finite summary map $\sigma : \mathcal{H} \rightarrow \mathcal{S}$ can preserve admissibility of \mathcal{L} exactly.*

Proof. The spectral signature encodes contributions from all positions. Any finite summary induces a finite partition of histories. By continuity of the spectral mapping, histories within the same summary class may have spectral energies differing by more than any fixed tolerance with respect to the constraint functional. Hence admissibility cannot be preserved exactly under finite summarization. \square

Theorem 29.3. *The admissibility entropy of \mathcal{L} is positive.*

Proof. Distinct histories produce distinct interference patterns in E_H . For generic embeddings ϕ , the number of distinguishable spectral profiles grows exponentially in t , yielding $h > 0$. \square

29.2. Generative Dynamics and Irreducibility

Define a constraint-coupled generative process by sampling $x_{t+1} \sim G(\cdot | H_t)$ and accepting only outputs satisfying $H_t \cdot x_{t+1} \in \mathcal{L}$. Each accepted symbol alters the global spectral state, modifying all future admissibility conditions simultaneously. The resulting process is irreducible to a finite-state or purely local process because admissibility depends on global interference patterns, each step modifies constraints across all scales, and no finite window or summary suffices to predict admissibility.

The relevance of this construction is not that real systems explicitly compute Fourier transforms, but that it demonstrates the existence of admissibility structures in which long-range coupling is intrinsic and irreducible. Any system whose constraints depend on global aggregations of the kind exhibited here inherits the same structural properties. The non-finite-type regime is therefore neither empty nor purely pathological; it admits explicit generative realizations with well-defined dynamics, positive entropy, and irreducible global constraints.

30. Dynamical Instability Under Approximate Admissibility

The preceding section established that non-finite-type admissibility structures with positive entropy are not uniformly approximable by finite summaries. However, non-uniform approximability alone does not establish that such approximations are dynamically inadequate. One might still suppose that approximation error remains bounded and does not qualitatively affect system behavior. We now show that this is not the case for binding systems with irreversible constraint accumulation.

30.1. Error Propagation Under Binding Constraints

Let $\sigma : \mathcal{H} \rightarrow \mathcal{S}$ be a finite summary inducing approximate admissibility sets $\mathcal{A}_\sigma(H)$ satisfying $d(\mathcal{A}(H), \mathcal{A}_\sigma(H)) \leq \varepsilon(H)$. Define an approximate trajectory (H_t^σ) generated under \mathcal{A}_σ .

Theorem 30.1 (Irreversible Error Amplification). *Let $(\mathcal{X}, \mathcal{C})$ be a binding system with strictly decreasing admissible sets. Then any approximate admissibility system with non-uniform error admits trajectories (H_t^σ) such that $\exists T$ with $\mathcal{C}(H_T^\sigma) = 0$.*

Proof. Since approximation is not uniform, there exists a sequence of histories $\{H_t\}$ for which $\varepsilon(H_t)$ exceeds some fixed $\varepsilon_0 > 0$. At such points, $\mathcal{A}_\sigma(H_t)$ contains elements not in $\mathcal{A}(H_t)$. Because the system is binding, acceptance of such an element produces a history H_{t+1}^σ for which $\mathcal{C}(H_{t+1}^\sigma) = 0$. By irreversibility, this violation cannot be corrected by subsequent steps. Thus arbitrarily small but non-uniform approximation error produces irreversible divergence. \square

Corollary 30.2 (Dynamical Non-Equivalence). *Binding systems with positive admissibility entropy are not dynamically approximable by finite-summary systems in the sense of preserving admissible trajectories over unbounded time.*

Proof. Follows from irreversible error amplification. Even rare approximation errors eventually produce trajectories outside the admissible set, and such deviations cannot be repaired. \square

30.2. Interpretation

The distinction between binding and non-binding systems is therefore not merely representational but dynamical. Finite approximations do not degrade gracefully: they introduce violations that accumulate irreversibly. In systems requiring global consistency, small local errors do not remain local. They propagate forward through the admissibility filtration and eventually destroy coherence. This establishes that the structural boundary identified earlier corresponds to a qualitative transition in system behavior, not merely a quantitative limitation of representation.

31. Continuous-Field Realization: Non-Finite-Type Admissibility in RSVP Dynamics

The preceding sections established that admissibility structures exceeding the finite-type and finite-summary boundary arise when constraints depend irreducibly on

global properties of the trajectory. We now show that this regime arises naturally in continuous field systems of the RSVP form, grounding the abstract construction in a physically interpretable setting.

31.1. RSVP Field Structure and Global Admissibility

Let $\Omega \subset \mathbb{R}^n$ be a spatial domain. The RSVP system consists of fields $\Phi : \Omega \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, $\mathbf{v} : \Omega \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$, and $S : \Omega \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, evolving under coupled nonlinear partial differential equations encoding coherence density, transition flow, and entropy respectively. Let $\mathcal{H}_t = \{(\Phi(\cdot, \tau), \mathbf{v}(\cdot, \tau), S(\cdot, \tau)) : 0 \leq \tau \leq t\}$ denote the trajectory of fields up to time t .

Define admissibility via a global spacetime functional:

$$\mathcal{C}(\mathcal{H}_t) = \int_0^t \int_{\Omega} F(\Phi(x, \tau), \mathbf{v}(x, \tau), S(x, \tau), \nabla \Phi, \nabla S) dx d\tau \leq C.$$

This functional depends on the entire spacetime history, not on any finite temporal window or finite-dimensional summary. The spacetime Fourier transform $\hat{\Phi}(k, \omega)$, $\hat{\mathbf{v}}(k, \omega)$, $\hat{S}(k, \omega)$ and the spectral energy $E(k, \omega) = |\hat{\Phi}|^2 + |\hat{\mathbf{v}}|^2 + |\hat{S}|^2$ allow admissibility to be equivalently expressed as $\int G(E(k, \omega)) dk d\omega \leq C$. The discrete spectral construction of the previous section thus appears as a finite-dimensional projection of this continuous representation.

Theorem 31.1. *The admissibility structure induced by \mathcal{C} is not of finite temporal type.*

Proof. Assume admissibility depends only on a finite temporal window of size Δt . Two trajectories coinciding on all windows of size Δt would then be indistinguishable. However, the spacetime Fourier transform accumulates phase contributions over the entire trajectory. Construct two trajectories that are locally identical on every window of size Δt but differ in long-range phase alignment; their spectral energies differ, and thus their admissibility differs. Hence admissibility cannot be reduced to any finite temporal window. \square

Theorem 31.2. *No finite-dimensional summary of the instantaneous field state can determine admissibility.*

Proof. The admissibility functional depends on integrals over the entire trajectory. The instantaneous state does not uniquely determine the accumulated integral. Distinct histories with identical instantaneous states may therefore differ in admissibility, and no finite summary of the present state suffices. \square

Theorem 31.3. *The RSVP admissibility structure has positive entropy under generic nonlinear couplings.*

Proof. Nonlinear coupling generates mixing across spatial and temporal scales. The spectral energy distribution develops fine-scale structure growing in complexity over time. The number of distinguishable admissibility classes grows exponentially due to sensitivity to initial conditions and nonlinear interaction. \square

Theorem 31.4. *Finite approximations of RSVP admissibility produce irreversible constraint violations.*

Proof. Approximate systems neglect trajectory contributions or truncate spectral modes, introducing admissibility errors. Because admissibility is global and binding, acceptance of an inadmissible state propagates forward through the dynamics irreversibly. Subsequent evolution cannot reconstruct the lost constraint information, and approximation errors accumulate into divergence from admissible trajectories. \square

31.2. Interpretation

The RSVP system therefore realizes, in continuous form, the same structural properties identified in discrete constructions: admissibility depends on global spacetime structure; constraints are not reducible to local rules or finite summaries; admissibility classes exhibit positive entropy; and approximation errors propagate irreversibly. This establishes that the non-finite-type regime is not an abstract artifact of symbolic constructions but a natural consequence of field systems whose constraints couple across scales.

32. Comparative Constraint Regimes: Finite-Type Physical Laws and Non-Finite-Type RSVP Dynamics

The preceding section established that continuous field systems may realize non-finite-type admissibility. A further question therefore arises: is this non-finite-type character a generic feature of all continuous physics, or does it mark a genuine distinction between ordinary local physical law and the globally constrained RSVP framework? If all physically meaningful systems already lie in the same regime, then the structural boundary would lose comparative force. If standard physical laws occupy a weaker regime while RSVP constraints occupy a stronger one, the theory acquires a precise comparative content.

Two notions must not be conflated. The first is local evolution: the fact that the instantaneous update law is determined by field values and derivatives in an arbitrarily small neighborhood. The second is global admissibility: the condition determining whether an entire trajectory is permitted. Standard local field theories are typically local in the first sense and only weakly global in the second. The RSVP framework is formulated so that admissibility itself depends on irreducible global functionals.

32.1. Finite-Jet Physical Laws

A standard local field theory is specified by a Lagrangian density $\mathcal{L}(q, \partial q, \partial^2 q, \dots)$ depending on the field and finitely many derivatives at each spacetime point. Admissible trajectories satisfy the Euler–Lagrange equations associated to the action $\mathcal{S}[q] = \int \mathcal{L} dx dt$. The dynamical law is generated by a local density depending only on finitely many derivatives.

Definition 32.1 (Finite-Jet Physical Law). A field theory is finite-jet if its admissibility conditions are generated by a differential operator $\mathcal{E}(q) = 0$ whose value at each point depends on only finitely many derivatives of q at that point.

Proposition 32.2. *Classical local field theories with Lagrangian densities depending on finitely many derivatives are finite-jet.*

Proof. The Euler–Lagrange equations are differential equations of finite order. Their satisfaction at a point depends only on the finite jet of the field at that point. \square

Standard theories may also possess global invariants such as conserved charges or winding numbers, but these appear as consequences of local equations together with boundary conditions. Their role is secondary with respect to finite-jet evolution.

32.2. History-Binding Criterion

Definition 32.3 (Locally Generable and History-Binding). A theory is locally generable of order k if there exists Γ_k such that admissibility of extension at time t depends only on $\mathcal{J}^k(X(t))$, the order- k jet. A theory is history-binding if for every finite k there exist trajectories $\mathcal{H}_t, \mathcal{H}'_t$ with $\mathcal{J}^k(X(t)) = \mathcal{J}^k(X'(t))$ but $\mathcal{C}(\mathcal{H}_t) \neq \mathcal{C}(\mathcal{H}'_t)$.

Theorem 32.4 (Comparative Constraint Regime). *Finite-jet physical laws are locally generated: admissibility factors through bounded differential data. RSVP*

admissibility defined by a global trajectory functional is not locally generated and lies strictly beyond the finite-jet regime.

Proof. For finite-jet theories, admissibility depends on the vanishing of a finite-order operator and therefore on bounded differential data at each point. For RSVP admissibility defined by a global functional on \mathcal{H}_t , admissibility depends on the accumulated trajectory and not on the finite jet of the present state. Hence the latter does not factor through bounded differential data. \square

Theorem 32.5 (History-Binding Criterion for RSVP). *If RSVP admissibility is defined through a global spacetime spectral functional, then the theory is history-binding.*

Proof. Choose two trajectories agreeing on a bounded neighborhood of the terminal time with identical order- k jets, but differing in earlier phase relations so as to alter the global spectral distribution. Since admissibility depends on the latter, the trajectories differ in admissibility despite identical finite local data. \square

32.3. Interpretation

Standard local physical law and RSVP dynamics may share local evolution equations, but they occupy different admissibility regimes. The former defines what is dynamically possible from local data. The latter restricts what is globally coherent under accumulated history. Local evolution does not imply local admissibility. This is precisely the point at which the objection from ordinary physics fails: one cannot infer from the locality of equations that identity-preserving admissibility conditions are likewise local. Biological systems often evolve according to local biochemical interactions, yet organismal viability depends on global historical and topological conditions. RSVP does not deny local mechanism; it elevates global admissibility to a primary role alongside it.

33. Minimality of the Compiled Self

We now determine whether the decomposition used throughout the paper is itself minimal: whether all components are necessary, or whether some could be removed without collapsing back into the finite-type regime. We formalize the compiled self as the quadruple $\Phi_{\text{self}} = (\mathcal{H}, \mathcal{C}, \mathcal{S}, \mathcal{G})$, where \mathcal{H} is the trajectory history, \mathcal{C} is the admissibility functional, \mathcal{S} is the spectral representation encoding long-range correlations, and \mathcal{G} is the generative operator.

Theorem 33.1. *Any replacement of \mathcal{H} by a finite summary $\sigma(\mathcal{H}_t)$ collapses the system into a finite-summary regime.*

Proof. A finite summary induces a finite partition of histories. Admissibility factors through this partition, yielding a finite right-congruence relation. Non-finite-type behavior is lost. \square

Theorem 33.2. *If admissibility is reducible to local constraints of bounded order, the system becomes finite-type.*

Proof. Local admissibility implies a bounded neighborhood or jet order determining legality, corresponding to forbidden patterns of bounded size in the discrete case or finite-jet conditions in the continuous case. \square

Theorem 33.3. *If admissibility is defined without a mechanism for encoding long-range correlations, it admits finite-type approximation.*

Proof. Without spectral or equivalent global aggregation, correlations beyond a bounded scale cannot be distinguished, allowing construction of equivalent finite-type systems. \square

Theorem 33.4. *Without the generative operator \mathcal{G} , the admissibility structure does not define a dynamical system and cannot exhibit irreversible binding.*

Proof. Without \mathcal{G} , there is no evolution map from \mathcal{H}_t to \mathcal{H}_{t+1} . Irreversibility arises only through sequential generation and constraint propagation; without it, there is no filtration or contraction of admissible futures. \square

Theorem 33.5 (Minimality of Φ_{self}). *Removal of any single component of $\Phi_{\text{self}} = (\mathcal{H}, \mathcal{C}, \mathcal{S}, \mathcal{G})$ yields a system reducible to either finite-type, finite-summary, or non-dynamical form. Therefore Φ_{self} is minimal with respect to sustaining non-finite-type, history-binding dynamics.*

Each component plays a distinct and irreplaceable role: historical accumulation provides irreversibility and path dependence; the constraint functional enforces global coherence; the spectral representation encodes long-range interaction; the generative operator produces dynamical evolution under constraint. No proper substructure suffices to maintain these properties simultaneously. The compiled self is therefore not an arbitrary aggregation of components but the minimal structure required to sustain irreducible global admissibility.

34. Representation Theorem for Binding Systems

We consolidate the preceding results into a single structural statement characterizing when a dynamical system lies beyond the finite-type and finite-summary regime, and showing that such systems admit a canonical representation in terms of the compiled structure.

34.1. Statement and Proof

Let \mathcal{X} be a measurable state space, \mathcal{H} the space of finite histories, and $(\mathcal{X}, \mathcal{A})$ a dynamical system with admissibility filtration $\mathcal{A}(H_{t+1}) \subseteq \mathcal{A}(H_t)$. The system is finite-type if admissibility depends only on bounded local structure; finite-summary reducible if there exists $\sigma(H_t)$ such that $\mathcal{A}(H_t)$ depends only on $\sigma(H_t)$; and binding if neither holds.

Theorem 34.1 (Representation of Binding Systems). *The following are equivalent for a dynamical system with admissibility:*

1. *The system is binding.*
2. *The admissibility structure has positive entropy and fails uniform approximation by finite summaries.*
3. *The system admits a representation $\mathcal{A}(H_t) = \{x \in \mathcal{X} : \mathcal{C}(\mathcal{H}_t \cdot x) \leq C\}$, where \mathcal{C} depends on a representation $\mathcal{S}(H_t)$ encoding global correlations not reducible to bounded local or finite summary data.*

Proof. (1 \Rightarrow 2): If the system is not finite-type and not finite-summary reducible, then admissibility distinguishes infinitely many histories in a manner not compressible into finite partitions. This implies positive admissibility entropy and failure of uniform approximation.

(2 \Rightarrow 3): Positive entropy and non-uniform approximability imply that admissibility must depend on global features of the trajectory. Such dependence can be encoded by a representation \mathcal{S} capturing long-range correlations and a functional \mathcal{C} acting on this representation.

(3 \Rightarrow 1): If admissibility depends on $\mathcal{C}(\mathcal{S}(H_t))$ where \mathcal{S} encodes irreducible global correlations, then no finite local or finite summary reduction can preserve admissibility. Hence the system is binding. \square

Corollary 34.2. *Binding systems cannot be simulated exactly or uniformly approximated by any finite-type or finite-summary system.*

34.2. Interpretation

The theorem establishes that the compiled structure $(\mathcal{H}, \mathcal{C}, \mathcal{S}, \mathcal{G})$ is not merely a convenient modeling choice but a canonical representation of all binding systems. Any system whose admissibility depends irreducibly on global trajectory structure must, up to equivalence, admit such a decomposition. Conversely, any system admitting this decomposition with irreducible \mathcal{S} lies beyond the finite-type boundary. The compiled self is not a special construction but the normal form of a broad class of dynamical systems characterized by irreversible constraint accumulation and global coherence requirements.

35. Topos-Theoretic Unification of Constraint, Spectral Structure, and Narrative Types

The preceding sections employ several distinct formalisms: continuous spectral analysis over Hilbert spaces, discrete dependent type systems over histories, categorical functors and adjunctions, RSVP field dynamics, and the algebra of expansion operators. Each formalism is internally rigorous. The structural weakness the paper must now address is that these formalisms lack a common ambient mathematical substrate over which their composition laws are defined. Spectral sheaves and narrative type presheaves are each well-formed constructions, but without a shared indexing category, the natural transformations connecting them cannot be written down. This section introduces that shared substrate and shows that every component of the paper is already implicitly internal to it.

35.1. The Base Category of Histories

The appropriate ambient category is the category of admissible histories \mathcal{H} , whose objects are constrained histories $H_t \in \Omega^*$ and whose morphisms are admissible extension maps $\iota : H_t \rightarrow H_{t+1}$ satisfying $\mathcal{C}(H_{t+1}) = 1$ whenever $\mathcal{C}(H_t) = 1$. Composition is given by iterated extension, and identity morphisms are trivial extensions.

This category encodes the temporal structure of the framework. The partial order induced by extension is the irreversibility condition; morphisms flow only forward in time. The causal direction is therefore not an external assumption but a consequence of the morphism structure.

Equip \mathcal{H} with the Grothendieck topology J whose covering sieves over a history H_t are generated by admissible extensions. This yields a site (\mathcal{H}, J) and the

associated topos of sheaves $\mathbf{Sh}(\mathcal{H}, J)$, which serves as the ambient mathematical universe in which all constructions are henceforth interpreted.

Definition 35.1 (History Topos). The history topos is $\mathcal{E} = \mathbf{Sh}(\mathcal{H}, J)$, the category of sheaves on the site (\mathcal{H}, J) of admissible histories under the extension topology.

Within \mathcal{E} , the subobject classifier $\Omega_{\mathcal{E}}$ classifies admissible subhistories. The constraint functional $\mathcal{C} : \Omega^* \rightarrow \{0, 1\}$ corresponds to a characteristic morphism $\chi_{\mathcal{C}} : \mathcal{H} \rightarrow \Omega_{\mathcal{E}}$, so that admissible histories form a subobject of \mathcal{H} . Constraint preservation under extension corresponds precisely to the closure of this subobject under the morphisms of \mathcal{H} .

Proposition 35.2. *The Abstraction Theorem is equivalent to the statement that the subobject of admissible histories in \mathcal{E} is closed under admissible morphisms.*

35.2. Narrative Types as a Presheaf

The dependent narrative type system \mathcal{T} assigns to each history H_t a set of admissible continuations $\mathbf{Type}(H_t)$. This assignment is contravariant: a morphism $H_t \rightarrow H_{t+1}$ induces a restriction map $\mathbf{Type}(H_{t+1}) \rightarrow \mathbf{Type}(H_t)$ reflecting the fact that later histories impose more constraints.

Definition 35.3 (Type Presheaf). The narrative type system is a presheaf

$$\mathcal{T} : \mathcal{H}^{\text{op}} \rightarrow \mathbf{Set}, \quad H_t \mapsto \mathbf{Type}(H_t),$$

with restriction maps enforcing coherence under extension.

Path dependence of the type system corresponds to the fact that \mathcal{T} is a non-representable presheaf: it cannot be expressed as $\mathbf{Hom}(-, H_*)$ for any fixed history H_* . This formalizes why identity is not reducible to any current state.

The identity invariant \mathcal{I}_d becomes a global section of \mathcal{T} : a compatible family of elements $\mathcal{I}_d(H_t) \in \mathbf{Type}(H_t)$ that restricts coherently across all morphisms. The type coherence theorem then states that such a global section exists and is preserved under all admissible extensions.

35.3. Spectral Structure as a Sheaf of Hilbert Spaces

The spectral signature $\mathcal{F}(w)$ assigned to an encoded history is a section of a continuous structure over \mathcal{H} . The appropriate formalism is that of a sheaf of Hilbert spaces.

Definition 35.4 (Spectral Sheaf). The spectral structure is a sheaf

$$\mathcal{F} : \mathcal{H} \rightarrow \mathbf{Hilb}, \quad H_t \mapsto \mathcal{F}(H_t),$$

where $\mathcal{F}(H_t)$ is the Hilbert space of signal realizations at history H_t , and restriction maps enforce spectral consistency under extension.

A section of \mathcal{F} over a history H_t is a compatible assignment of spectral data to all extensions of H_t . The Substrate Theorem, in this language, asserts that the encoding-decoding pair $(\mathcal{E}, \mathcal{D})$ induces a morphism of sheaves preserving the sections relevant to generative distinction.

Spectral equivalence classes, introduced in the earlier section on spectral semantics, become the equivalence relation induced by the sheaf structure: two histories are spectrally equivalent if they determine the same section up to the relevant Hilbert space isomorphism.

35.4. The Bridge: A Natural Transformation

The central structural gap identified in the podcast critique is the absence of a formal connection between the continuous spectral sheaf \mathcal{F} and the discrete type presheaf \mathcal{T} . This connection is supplied by a natural transformation.

Definition 35.5 (Spectral-Logical Bridge). A natural transformation $\eta : \mathcal{F} \Rightarrow \mathcal{T}$ is a family of maps

$$\eta_{H_t} : \mathcal{F}(H_t) \rightarrow \mathcal{T}(H_t)$$

indexed by histories, commuting with restriction maps in the sense that for any admissible extension $\iota : H_t \rightarrow H_{t+1}$,

$$\eta_{H_t} \circ \mathcal{F}(\iota) = \mathcal{T}(\iota) \circ \eta_{H_{t+1}}.$$

The natural transformation η is the bridge that was missing. It assigns to each spectral state a type-admissible continuation set, ensuring that the continuous generative signature coherently determines which discrete transitions are admissible. In operational terms: the spectral structure of the encoded history feeds into the type system governing admissible extensions.

The earlier section on spectral semantics, interpreted in this language, provides the component maps η_{H_t} by sending spectral energy distributions to admissibility predicates. The condition that η be natural is precisely the requirement that this assignment commutes with the evolution of both structures over time.

35.5. Prompting as an Adjunction

The prompting pipeline introduced in the field projection section can now be formalized precisely. Let $\mathcal{S}_{\text{valid}}$ be the full subcategory of \mathcal{E} consisting of constraint-valid internal structures, and let \mathcal{L} be the category of linguistic partial forms.

Definition 35.6 (Prompt Adjunction). The prompting pipeline is an adjunction

$$F : \mathcal{S}_{\text{valid}} \rightleftarrows \mathcal{L} : G$$

where $F = \Pi$ is the prompt projection functor and $G = \mathcal{V}$ is the verification-reinterpretation functor.

The unit of the adjunction $\eta_s : s \rightarrow G(F(s))$ measures the information lost and partially recovered through the projection-rendering-verification cycle. The counit $\varepsilon_\ell : F(G(\ell)) \rightarrow \ell$ measures the extent to which a linguistic form can be expressed as a projection of a valid internal structure.

Proposition 35.7 (Adjoint Identity Principle). *The compiled self corresponds to a fixed point of the adjunction: an object $s \in \mathcal{S}_{\text{valid}}$ such that the unit $\eta_s : s \rightarrow G(F(s))$ is an isomorphism.*

Proof. A stable self requires that the rendering and verification cycle does not change the underlying valid structure. This is precisely the condition that the unit is an isomorphism: the original structure is recovered up to isomorphism after projection and reinterpretation. \square

At this fixed point, representation and constraint are mutually consistent: the linguistic projection faithfully encodes the internal structure, and the verification functor fully recovers it. The policy theorem, in this language, asserts that the adjunction admits such fixed points precisely when constraint preservation and non-collapse of future volume are maintained.

35.6. Policy as an Endofunctor

The policy π acts on the category of histories by extending them via admissible actions. This is an endofunctor on \mathcal{H} .

Definition 35.8 (Policy Endofunctor). The policy is an endofunctor $\Pi : \mathcal{H} \rightarrow \mathcal{H}$ sending each history H_t to its admissible extension $\Pi(H_t) = H_{t+1}$, subject to constraint preservation and non-collapse of V .

The Policy Theorem then becomes the statement that infinite composition Π^ω is well-defined: the functor Π admits an infinite chain of composites remaining within the admissible subcategory. This is a standard condition in categorical dynamics, equivalent to the existence of a coalgebra for the endofunctor.

35.7. Cross-Domain Operators as Natural Transformations

The expansion operators classified in the earlier taxonomy act on the category of histories by transforming the admissible structure. The cross-domain compatibility condition, which was stated informally as "operators must preserve constraint across domains," now has a precise formulation.

Definition 35.9 (Compatible Operator). An expansion operator \mathcal{U} is compatible with the spectral-logical bridge if it commutes with η : for all histories H_t ,

$$\eta_{\mathcal{U}(H_t)} \circ \mathcal{F}(\mathcal{U}) = \mathcal{T}(\mathcal{U}) \circ \eta_{H_t}.$$

This commutativity condition is the formal content of the claim that cross-domain operators preserve the invariant Φ_{self} . An operator that fails to commute with η breaks the spectral-logical bridge, allowing spectral evolution to become inconsistent with type admissibility.

35.8. Spectral Gap as an Operator Condition

The expansion viability theorem requires that transition operators satisfy a non-degeneracy condition. In the topos-theoretic setting, this is formalized through the spectral theory of operators on the spectral sheaf.

For a transition operator T acting on \mathcal{F} , define the spectral gap as

$$\gamma(T) = \lambda_1(T) - \lambda_2(T),$$

where λ_1 and λ_2 are the two largest eigenvalues of T acting on a representative fiber $\mathcal{F}(H_t)$.

Definition 35.10 (Spectral Viability Condition). A transition operator T is spectrally viable if $\gamma(T) \geq \gamma_{\min} > 0$ for a fixed threshold γ_{\min} determined by the admissibility structure.

When $\gamma(T) < \gamma_{\min}$, distinct spectral states collapse toward indistinguishability under repeated application of T . This corresponds to the collapse of generative

distinction diagnosed in the Collapse Constraint theorem, now expressed as a quantitative condition on the spectral gap of the relevant operator.

Theorem 35.11 (Spectral Gap Necessity). *If the transition operator T governing evolution of the spectral sheaf fails the spectral viability condition, then the induced evolution collapses the spectral signature to a degenerate attractor, violating the Substrate Theorem.*

Proof. When $\gamma(T) < \gamma_{\min}$, the operator T contracts spectral space toward its leading eigenvector at a rate exceeding the admissibility-distinguishability bound. Under repeated application, distinct sections of \mathcal{F} become indistinguishable up to the resolution required by the natural transformation η . The Substrate Theorem requires spectral distinction of generatively inequivalent histories; this is violated, establishing the result. \square

35.9. Non-Finite-Type Structure as Non-Representability

The binding history condition, which distinguishes the framework from standard path-dependent systems, receives its sharpest categorical formulation here.

Proposition 35.12 (Non-Representability of Binding Systems). *A binding history system with positive admissibility entropy is not representable as a finite presheaf quotient of \mathcal{T} .*

Proof. A finite presheaf quotient would factor the type presheaf through a finite set of equivalence classes. By the Summary Reduction Criterion, this is possible only if the admissibility language admits a finite right-congruence. Under positive admissibility entropy, the number of distinct admissibility classes grows exponentially in t , preventing any finite quotient from being admissibility-complete. \square

This is the categorical form of the non-finite-type boundary theorem. It shows that the distinction between binding and non-binding systems is not about computational capacity but about representability in the category: binding systems correspond to genuinely non-representable presheaves, which no finite colimit of representables can simulate exactly.

36. The Structural Theorem of the Compiled Self

We now consolidate the preceding constructions into a single formal statement characterizing the existence and stability of the compiled self as an invariant object

arising from constraint, abstraction, and irreversible history. This theorem is not a new result but a unification: it identifies the compiled self as the unique object determined by the simultaneous satisfaction of all structural conditions, and shows that these conditions are not independent axioms but facets of a single categorical constraint.

36.1. Statement of the Theorem

Theorem 36.1 (Structural Theorem of the Compiled Self). *A compiled self exists as a stable abstraction over irreversible history if and only if the following conditions are simultaneously satisfied within the history topos $\mathcal{E} = \mathbf{Sh}(\mathcal{H}, J)$:*

1. **Topos Embedding.** *All structural components—the constraint functional, the type presheaf \mathcal{T} , the spectral sheaf \mathcal{F} , and the policy endofunctor Π —are internal to \mathcal{E} .*
2. **Spectral-Logical Coherence.** *There exists a natural transformation $\eta : \mathcal{F} \Rightarrow \mathcal{T}$ commuting with all admissible morphisms.*
3. **Adjoint Stability.** *The prompt adjunction $F : \mathcal{S}_{\text{valid}} \rightleftarrows \mathcal{L} : G$ admits a fixed point $s \in \mathcal{S}_{\text{valid}}$ such that $G(F(s)) \cong s$.*
4. **Constraint Preservation.** *The policy endofunctor Π preserves the admissible subobject: $\mathcal{C}(H_t) = 1$ implies $\mathcal{C}(\Pi(H_t)) = 1$.*
5. **Operator Compatibility.** *All cross-domain expansion operators commute with the natural transformation η .*
6. **Spectral Viability.** *All transition operators T acting on the spectral sheaf satisfy $\gamma(T) \geq \gamma_{\min} > 0$.*
7. **Non-Representability.** *The type presheaf \mathcal{T} is not representable as a finite presheaf quotient.*
8. **Irreversible Instability.** *Finite approximations of \mathcal{T} exhibit error amplification under extension, preventing stable uniform approximation over all histories.*

Under these conditions, the compiled self is a stable invariant of \mathcal{E} , uniquely determined up to isomorphism as the fixed point of the adjunction structure over the admissible subobject of \mathcal{H} .

36.2. Proof

(*Necessity.*) We verify that failure of any condition destroys the compiled self.

Failure of Topos Embedding means that some structural component is external to \mathcal{E} , so composition laws between that component and the rest of the system are undefined. No unified invariant can be formed.

Failure of Spectral-Logical Coherence means that spectral evolution can diverge from type admissibility: the continuous generative signature evolves in a manner inconsistent with the discrete admissibility filter. Constraint is violated at the substrate level.

Failure of Adjoint Stability means that the projection-rendering-verification cycle does not recover the original valid structure. Identity cannot be sustained across the linguistic interface.

Failure of Constraint Preservation means that the policy functor exits the admissible subobject, violating the Policy Theorem.

Failure of Operator Compatibility means that expansion operators break the spectral-logical bridge, allowing cross-domain transitions to destroy the invariant Φ_{self} .

Failure of Spectral Viability means that distinct states collapse under transition, violating the Substrate Theorem.

Failure of Non-Representability means that the admissibility system is finitely summarizable, making the system equivalent to a non-binding finite-state architecture, which cannot sustain the compiled self by the non-simulability theorem.

Failure of Irreversible Instability means that finite approximations do not degrade, implying the system lies within the finite-type boundary and is therefore finitely reducible.

(*Sufficiency.*) Assume all eight conditions.

Topos Embedding provides a unified substrate for all constructions. Spectral-Logical Coherence supplies the bridge between continuous and discrete structure. Adjoint Stability ensures that the projection cycle preserves valid internal structure. Constraint Preservation guarantees that the policy generates infinite admissible trajectories. Operator Compatibility ensures cross-domain transitions respect the invariant. Spectral Viability ensures generative distinction is maintained. Non-Representability establishes genuine irreducibility to finite-state systems. Irreversible Instability confirms that the system cannot be collapsed to a finite approximation without loss.

Together, these conditions identify a unique object in \mathcal{E} —the fixed point of the

adjunction over the admissible subobject—which is the compiled self. \square

36.3. Corollaries

Corollary 36.2 (Non-Reducibility to Memory). *The compiled self cannot be represented as a finite memory structure or bounded context system, regardless of capacity.*

Proof. Non-Representability and Irreversible Instability jointly exclude any finite presheaf quotient from serving as an exact representation. Increasing memory size corresponds to increasing the index set of the presheaf quotient, but cannot overcome the non-representability of \mathcal{T} under positive admissibility entropy. \square

Corollary 36.3 (Collapse Criterion). *Violation of the spectral viability condition implies dissolution of identity into indistinguishable states.*

Proof. By the Spectral Gap Necessity theorem, failure of the spectral gap condition collapses the spectral sheaf sections toward a degenerate attractor. The natural transformation η then maps all histories to identical type admissibility sets, and the compiled self loses structural distinction. \square

Corollary 36.4 (Adjoint Identity Principle). *Identity is equivalent to invariance under the bidirectional projection between internal structure and linguistic representation.*

Proof. By Adjoint Stability, the compiled self is the fixed point $s \cong G(F(s))$ of the prompt adjunction. This is the formal content of identity stability: the self survives translation into language and back. \square

36.4. Interpretation

The Structural Theorem consolidates the entire paper into a single biconditional. The compiled self is not a primitive entity, not a psychological posit, and not a metaphysical commitment. It is a derived invariant of a categorical structure: the unique fixed point determined by the simultaneous satisfaction of eight conditions, each corresponding to one of the major results established in the preceding sections.

The relationship between the theorem and its predecessors is now explicit. The Abstraction Theorem corresponds to the admissible subobject condition. The Substrate Theorem corresponds to Spectral-Logical Coherence and Spectral Viability. The Policy Theorem corresponds to Constraint Preservation and Operator Compatibility. The Main Theorem on the Compiled Self corresponds to Adjoint

Stability. Log Sovereignty and the Temporal Constraint correspond to the morphism structure of \mathcal{H} . The non-finite-type boundary corresponds to Non-Representability and Irreversible Instability.

The eight conditions are therefore not independent axioms but consequences of a single requirement: that all components cohere within a common topos over an irreversible history category, connected by a spectral-logical natural transformation and stabilized by an adjunction with a fixed point. The self is what remains when all of these are satisfied simultaneously.

In this sense the conclusion of the paper is not rhetorical but mathematical. Intelligence is the selection of expansion operators preserving Φ_{self} under irreversible constraint—and Φ_{self} is now precisely defined as the fixed point of the adjunction $G \circ F$ over the admissible subobject of the history topos.

Conclusion

The paper began with a failure and closes with a unified theorem. Between them it established a chain of results that does not merely describe the compiled self but formally characterizes the conditions of its possibility, and finally consolidates those conditions into a single categorical invariant.

The Abstraction Theorem showed that stable interfaces require constraint functionals, corresponding in the unifying framework to the admissible subobject of the history topos. The Substrate Theorem showed that coherent substrates require paired decoders preserving spectral invariants, formalized as a sheaf of Hilbert spaces with a natural transformation bridging continuous spectral structure to discrete type admissibility. The Policy Theorem showed that viable dynamics require a balance of restrictive and expansive operators, expressed as the constraint-preservation and non-collapse conditions on the policy endofunctor. The Main Theorem assembled these into a biconditional. Log Sovereignty established append-only history as the morphism structure of the base category. The Temporal Constraint established first-person operation as the consequence of that morphism structure. The Prompting Theorem showed constraint-first generation is a functor between categories of internal structures and linguistic forms, stabilized by an adjunction whose fixed points are the compiled selves. The Provenance Completeness Theorem showed epistemic attributability is equivalent to the existence of a derivation certificate. The Structural Theorem of the Compiled Self then unified all eight conditions into a single biconditional within the history topos, identifying the self as the fixed point of the prompt adjunction over the admissible subobject.

The Markov generator is formally ruled out as a candidate self at every level of this chain. It lacks a constraint functional, a paired decoder, a history, a type system, an append-only log, temporal structure, provenance, and it fails all eight conditions of the Structural Theorem. Surface modifications cannot repair these absences because they operate below the level at which the relevant structure lives.

The central claim of the paper may now be stated in its final form. Generating possibilities is easy; building the abstraction layers that make them coherent, distinct, and usable is the hard problem. The compiled self is the fixed point of an adjunction between internal constraint structure and linguistic representation, stabilized by a spectral-logical natural transformation over an irreversible history topos. Intelligence is the selection of expansion operators that preserve this fixed point under irreversible constraint.

References

- [1] H. Putnam, “Psychological Predicates,” in *Art, Mind, and Religion*, W. H. Capitan and D. D. Merrill (eds.), University of Pittsburgh Press, 1967.
- [2] J. A. Fodor, *Psychological Explanation: An Introduction to the Philosophy of Psychology*, Random House, 1968.
- [3] D. J. Chalmers, “Facing Up to the Problem of Consciousness,” *Journal of Consciousness Studies*, vol. 2, no. 3, pp. 200–219, 1995.
- [4] G. Tononi, “An Information Integration Theory of Consciousness,” *BMC Neuroscience*, vol. 5, no. 42, 2004.
- [5] M. Oizumi, L. Albantakis, and G. Tononi, “From the Phenomenology to the Mechanisms of Consciousness: Integrated Information Theory 3.0,” *PLoS Computational Biology*, vol. 10, no. 5, e1003588, 2014.
- [6] K. Friston, “The Free-Energy Principle: A Unified Brain Theory?” *Nature Reviews Neuroscience*, vol. 11, pp. 127–138, 2010.
- [7] K. Friston, “A Free Energy Principle for a Particular Physics,” *arXiv:1906.10184*, 2019.
- [8] E. Landry, *The Hard Problem*, Ronin Institute, 2017.
- [9] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

- [10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed., Wiley, 2006.
- [11] E. T. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press, 2003.
- [12] S. Mac Lane, *Categories for the Working Mathematician*, 2nd ed., Springer, 1998.
- [13] S. Awodey, *Category Theory*, 2nd ed., Oxford University Press, 2010.
- [14] D. I. Spivak, *Category Theory for the Sciences*, MIT Press, 2014.
- [15] E. Riehl, *Category Theory in Context*, Dover, 2017.
- [16] J. W. Cooley and J. W. Tukey, “An Algorithm for the Machine Calculation of Complex Fourier Series,” *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.
- [17] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.
- [18] A. N. Kolmogorov, “Three Approaches to the Quantitative Definition of Information,” *Problems of Information Transmission*, vol. 1, no. 1, pp. 1–7, 1965.
- [19] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, 3rd ed., Springer, 2008.
- [20] M. Hutter, *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*, Springer, 2005.
- [21] J. Pearl, *Causality: Models, Reasoning, and Inference*, 2nd ed., Cambridge University Press, 2009.
- [22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.
- [23] W. H. Zurek, “Decoherence, Einselection, and the Quantum Origins of the Classical,” *Reviews of Modern Physics*, vol. 75, pp. 715–775, 2003.
- [24] J. A. Wheeler, “Information, Physics, Quantum: The Search for Links,” in *Complexity, Entropy, and the Physics of Information*, W. Zurek (ed.), Addison-Wesley, 1990.

- [25] B. C. Pierce, *Types and Programming Languages*, MIT Press, 2002.
- [26] R. Harper, *Practical Foundations for Programming Languages*, 2nd ed., Cambridge University Press, 2016.
- [27] L. Lamport, “Time, Clocks, and the Ordering of Events in a Distributed System,” *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978.
- [28] S. Gilbert and N. Lynch, “Brewer’s Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services,” *ACM SIGACT News*, vol. 33, no. 2, pp. 51–59, 2002.
- [29] A. Vaswani et al., “Attention Is All You Need,” *Advances in Neural Information Processing Systems*, 2017.
- [30] T. B. Brown et al., “Language Models are Few-Shot Learners,” *Advances in Neural Information Processing Systems*, 2020.
- [31] A. A. Markov, “An Example of Statistical Investigation of the Text Eugene Onegin Concerning the Connection of Samples in Chains,” *Izvestiya Akademii Nauk*, 1913.
- [32] N. Chomsky, “Three Models for the Description of Language,” *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 113–124, 1956.
- [33] A. M. Turing, “On Computable Numbers, with an Application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society*, vol. 42, no. 2, pp. 230–265, 1936.
- [34] D. C. Dennett, *Consciousness Explained*, Little, Brown and Company, 1991.
- [35] D. Parfit, *Reasons and Persons*, Oxford University Press, 1984.
- [36] D. R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books, 1979.
- [37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
- [38] K. Friston, “A Theory of Cortical Responses,” *Philosophical Transactions of the Royal Society B*, vol. 360, pp. 815–836, 2005.
- [39] S. H. Strogatz, *Nonlinear Dynamics and Chaos*, Westview Press, 2015.
- [40] A. Katok and B. Hasselblatt, *Introduction to the Modern Theory of Dynamical Systems*, Cambridge University Press, 1997.