Protocol Suite

Constraint, Irreversibility, and Structural Inference

Abstract

This work introduces a unified mathematical framework for reasoning about systems whose identity is determined not solely by their present configuration but by the admissible trajectories through which they evolve. The framework formalizes state as a structured tuple consisting of a possibility space, a constraint family, an induced admissible subset, an append-only history trace, and a projection to observable state. Within this setting, irreversibility is taken as a primitive rather than an anomaly, and reconstruction is distinguished sharply from reversal.

A central result is that observational equivalence does not imply structural equivalence when the projection from underlying state to observables is non-injective. This induces a general indistinguishability phenomenon and motivates the explicit retention of provenance. The framework further introduces entropy-like measures of residual ambiguity, semigroup-based evolution operators, and a constraint-aware formulation of merge as a structure-preserving optimization problem.

The resulting system provides a rigorous basis for analyzing inference, simulation, and integration across domains, replacing object-centric descriptions with a trajectory-centric ontology in which constraint, admissibility, and history determine the space of meaningful transformations.

1 Introduction

Contemporary formalisms in mathematics, physics, and computation frequently adopt a state-centric ontology in which a system is identified with a point in a configuration space. Within such frameworks, dynamics are expressed as transformations on states, and identity is implicitly treated as extensional, determined entirely by present configuration. While effective in domains where reversibility or full observability may be assumed, this perspective becomes insufficient when systems exhibit information loss, projection, or path dependence.

In many practical and theoretical settings, the observable state of a system is a compressed representation of a richer underlying structure. Distinct generative processes may converge to indistinguishable outputs, and the internal evolution that produced a given configuration may not be recoverable from that configuration alone. This phenomenon is not exceptional but pervasive. It arises in physical systems with dissipative dynamics, in computational systems that discard intermediate state, and in semantic systems where multiple derivations yield identical expressions.

The central aim of this work is to provide a formal framework in which these features are not treated as anomalies but as fundamental. The framework replaces the identification of systems with static states by an identification with constrained trajectories through a structured space of possibilities. Rather than asking what a system *is* at a given moment, the framework asks what configurations remain admissible under a given constraint family and how those admissible configurations evolve under irreversible operations.

A key distinction is drawn between the underlying state of a system and its observable projection. Let a mapping

$$\pi : \mathcal{S} \to \mathcal{X}$$

represent the compression from detailed state space $\mathcal{S}$ to observable space $\mathcal{X}$. When π fails to be injective, the observable state does not uniquely determine the underlying state. Consequently, any reasoning process that operates solely at the level of $\mathcal{X}$ risks conflating structurally distinct configurations.

This observation leads to a reorientation of several core notions. Identity is no longer purely extensional but may depend on historical trace. Evolution is not assumed to be invertible, and thus reconstruction must be treated as an inference problem rather than as the application of an inverse operator. Equivalence must be refined to distinguish observational indistinguishability from structural equivalence.

The framework developed here formalizes these ideas by introducing a structured state consisting of a possibility space, a constraint family, an induced admissible subset, and an append-only history. Dynamics act on this structure through updates to constraints, admissible sets, and history, rather than solely through pointwise state transitions. This formulation naturally accommodates ambiguity, underdetermination, and path dependence.

Beyond providing a descriptive language, the framework yields operational consequences.

It enables the definition of entropy-like measures of residual ambiguity, clarifies the limits of inference under projection, and supports a principled approach to integration tasks such as merge, where the goal is not to combine representations but to preserve compatible structure. It also admits mappings to continuous field descriptions, offering a bridge between discrete and continuous models of system evolution.

The subsequent sections develop these ideas in a precise manner. The formal state representation is introduced, followed by a treatment of constraints and admissibility. Irreversibility and history are then examined, leading to a discussion of projection and indistinguishability. The framework is subsequently applied to the problem of semantic merge and extended through field and categorical interpretations. The overall result is a unified perspective in which constraint, admissibility, and trajectory replace static state as the primary carriers of meaning.

2 The Protocol State

The fundamental object of the framework is a structured state that separates possibility, constraint, admissibility, history, and observation. At time t, define the state

$$Y_t = (\Omega_t, C_t, A_t, H_t, X_t, I_t),$$

where each component plays a distinct and non-interchangeable role.

The set Ω_t is the possibility space. It represents the collection of all configurations that are under consideration prior to the enforcement of constraints. The specification of Ω_t may be explicit, as in a finite combinatorial space, or implicit, as in a functional or geometric domain. In either case, it serves as the ambient domain over which admissibility is defined.

The family C_t is the active constraint set. Each element of C_t restricts the space of allowable configurations, and the total constraint family determines the structure of feasible evolution. Constraints may be expressed as predicates, equations, inequalities, typing rules, compatibility conditions, or admissibility relations. The framework does not privilege a particular representation, but it requires that constraints be made explicit rather than absorbed into narrative description.

The admissible set $A_t \subseteq \Omega_t$ is defined through a constraint operator

$$A_t = \mathrm{Adm}(\Omega_t, C_t).$$

This operator encodes the rule by which constraints act on the possibility space. The admissible set is the locus of configurations that satisfy all active constraints. In many situations, A_t is the primary object of interest, as it determines what remains structurally possible at time t.

The history H_t is an append-only trace of events. Each element of H_t records a transformation, observation, or constraint update that has occurred during the evolution

of the system. Formally, if e_{t+1} denotes the event at the next step, then

$$H_{t+1} = H_t \oplus e_{t+1},$$

where $\oplus$ denotes concatenation without deletion. The absence of a general inverse for this operation reflects the irreversibility built into the framework.

The observable state X_t is defined as a projection

$$X_t = \pi(\Omega_t, C_t, H_t),$$

where π is a compression map. The purpose of X_t is to represent what is externally visible or retained in a reduced representation. In general, π is not injective, and therefore multiple underlying states may correspond to the same observable.

The collection I_t represents invariants or monitored quantities. Each element of I_t is a function or relation that is expected to remain stable under admissible evolution. These invariants provide a mechanism for detecting structural drift and for enforcing consistency across transformations.

This decomposition clarifies several distinctions that are often obscured. The possibility space Ω_t should not be confused with the admissible set A_t, and the observable X_t should not be identified with the full state Y_t. Similarly, the history H_t is not reducible to the current admissible configuration, even when the latter is known precisely.

The structure also admits natural generalization. One may enrich the state with additional components such as metadata, confidence weights, or multi-scale representations, but the core tuple provides a minimal schema that captures the essential features of constrained, irreversible evolution.

Finally, it is important to emphasize that Y_t is not intended as a static container but as a dynamically evolving object. Each component participates in the evolution of the system, and the interactions among these components determine the admissible trajectories through the possibility space. The subsequent sections formalize the mechanisms by which these components evolve and interact.

3 Constraints and Admissibility

The structure of admissibility is determined by the constraint family C_t acting on the possibility space Ω_t. This interaction is formalized through an admissibility operator

$$\mathrm{Adm} : \mathcal{P}(\Omega) \times \mathcal{C} \to \mathcal{P}(\Omega),$$

which assigns to each pair (Ω_t, C_t) a subset $A_t \subseteq \Omega_t$. The admissible set is therefore not intrinsic to the possibility space alone, but emerges from the conjunction of possibilities and constraints.

In many applications, it is useful to represent constraints through a defect functional

$$\Delta_t : \Omega_t \to \mathbb{R}_{\geq 0},$$

which quantifies the degree to which a configuration violates the constraint family. In this representation, exact admissibility is characterized by the zero set

$$A_t = \{x \in \Omega_t \mid \Delta_t(x) = 0\}.$$

More generally, one may consider an approximate admissible region

$$A_t^{\varepsilon} = \{x \in \Omega_t \mid \Delta_t(x) \leq \varepsilon\},$$

which allows for tolerance in the enforcement of constraints. This formulation unifies feasibility conditions, optimization criteria, and probabilistic weighting schemes within a single mathematical structure.

The constraint family C_t may be decomposed into subclasses according to their role. A common distinction is between constraints that define feasibility and those that define preference. Let

$$C_t = C_t^{\text{hard}} \cup C_t^{\text{soft}},$$

where hard constraints determine membership in A_t, and soft constraints contribute to the structure of Δ_t without necessarily excluding configurations outright. This distinction is particularly relevant in systems where trade-offs must be managed rather than eliminated.

The geometry of the admissible set A_t plays a central role in the analysis of the system. Depending on the nature of C_t, the set A_t may be discrete, continuous, convex, disconnected, or highly structured. The shape of A_t determines not only which configurations are possible but also how the system can evolve. For example, a convex admissible region permits continuous interpolation between admissible states, while a disconnected region imposes combinatorial transitions.

An important feature of this framework is that constraints are treated as first-class objects rather than as implicit background assumptions. This requires that constraints be expressed in a form that can be analyzed, composed, and modified. When constraints are left implicit, it becomes difficult to determine whether a configuration is inadmissible due to structural incompatibility or due to an unarticulated assumption.

Constraint failure is therefore described in structural terms. Given a configuration $x \in \Omega_t$ such that $x \notin A_t$, the failure is not merely that x is "incorrect," but that there exists at least one constraint $c \in C_t$ such that $c(x)$ is violated. A proper analysis of failure identifies the specific constraint, the manner of violation, and the minimal modification required to restore admissibility.

The interaction between constraints may also produce emergent structure. Constraints may be redundant, independent, or mutually incompatible. Redundant constraints do not

reduce the admissible set beyond what is already enforced, while incompatible constraints may render A_t empty. In such cases, the system is overconstrained, and resolution requires modification of the constraint family itself.

Finally, the evolution of the system may involve changes to C_t. New constraints may be introduced, existing constraints may be relaxed, or priorities among constraints may be altered. Such changes induce corresponding transformations in A_t, and therefore in the trajectory of the system. The explicit representation of C_t allows these transformations to be tracked and analyzed, rather than absorbed into opaque state transitions.

In summary, constraints define the structure of possibility, admissibility defines the region of feasibility, and the interaction between them determines the space of meaningful evolution. The next section examines how this structure behaves under irreversible dynamics and the implications for history and reconstruction.

4 Irreversibility and History

The framework adopts irreversibility as a primitive structural feature rather than as a special case. The evolution of history is governed by an append-only operation

$$H_{t+1} = H_t \oplus e_{t+1},$$

where e_{t+1} denotes the event occurring at time $t+1$, and $\oplus$ denotes concatenation without deletion or modification. This operation defines a semigroup structure on histories, but in general does not admit an inverse.

The absence of a general inverse has immediate consequences. In particular, there is no operator R such that

$$R(H_t \oplus e_{t+1}) = H_t$$

for arbitrary histories and events. Thus, evolution is not assumed to be reversible, and the notion of reconstructing prior states must be treated independently of the notion of reversing a dynamic.

To formalize this distinction, consider the full state space

$$\mathcal{S}_t = \Omega_t \times \mathcal{H}_t,$$

where $\mathcal{H}_t$ denotes the set of all admissible histories up to time t. A transition operator

$$\mathcal{T}_t : \mathcal{S}_t \to \mathcal{S}_{t+1}$$

acts by updating both the possibility space and the history. Even if $\mathcal{T}_t$ is injective on its domain, the presence of projection in the observable layer implies that information may be lost at the level of X_t.

Let

$$\pi : \mathcal{S}_t \to \mathcal{X}_t$$

be the observable projection. If π is not injective, then distinct elements of $\mathcal{S}_t$ may map to the same observable state. In this case, reconstruction of prior states from X_t alone is ill-posed, as the preimage

$$\pi^{-1}(X_t)$$

contains multiple candidates.

This leads to a fundamental distinction between three operations. The first is replay, which reconstructs the present state from the retained history H_t by reapplying the sequence of events. The second is reconstruction, which attempts to infer a plausible history or prior state from incomplete information. The third is reversal, which would require an inverse operator for the dynamics. The framework treats replay as well-defined when history is retained, reconstruction as an inference problem subject to ambiguity, and reversal as generally unavailable.

The role of history in determining identity becomes clear in this setting. Two states Y_t and Y_t' may share the same observable component $X_t = X_t'$ while differing in their histories $H_t \neq H_t'$. If the semantics of the system depend on provenance, then these states must be regarded as distinct, even though they are observationally equivalent.

This observation motivates a refinement of equivalence relations. Define observational equivalence by

$$Y_t \sim_{\text{obs}} Y_t' \iff X_t = X_t'.$$

Define historical equivalence by

$$Y_t \sim_{\text{hist}} Y_t' \iff (X_t = X_t' \text{ and } H_t \equiv H_t' \text{ up to a specified equivalence on histories}).$$

In general, $\sim_{\text{hist}}$ is strictly finer than $\sim_{\text{obs}}$.

Irreversibility also affects the structure of admissible trajectories. Let a trajectory be a sequence

$$(Y_0, Y_1, \ldots, Y_t)$$

such that each step satisfies the admissibility conditions induced by the evolving constraint family. Because history accumulates, the admissibility of future states may depend not only on the current configuration but also on the path taken to reach it. This introduces path dependence into the system, which cannot be captured by state-only models.

From a design perspective, the explicit treatment of history leads to several principles. Systems that require provenance-sensitive reasoning must retain sufficient information in H_t to distinguish relevant trajectories. Compression mechanisms that discard history must be recognized as lossy and must be accompanied by an analysis of the resulting ambiguity. When reconstruction is required, it should be framed as a constrained inference problem over admissible histories rather than as an attempt to invert a dynamic.

In summary, irreversibility and history transform the nature of state evolution. They require a shift from reversible transformations on static configurations to semigroup actions

on structured states with accumulated trace. This shift is essential for accurately modeling systems in which information loss, projection, and path dependence are intrinsic.

5 Ambiguity, Defect, and Entropy-Like Structure

The admissible set $A_t \subseteq \Omega_t$ encodes not only feasibility but also the residual ambiguity of the system. When A_t contains more than one element, the system is underdetermined with respect to the active constraint family. This underdetermination may arise from incomplete information, non-injective projections, or intrinsic degeneracy of the constraint structure.

To quantify this ambiguity, let μ be a measure on Ω_t, defined appropriately for the domain under consideration. The effective ambiguity of the system at time t is then given by

$$S_{\text{eff}}(t) = \log \mu(A_t),$$

provided that $\mu(A_t)$ is finite and nonzero. This quantity measures the "volume" of the admissible region and serves as an entropy-like functional over the possibility space.

It is essential to emphasize that S_{eff} is not thermodynamic entropy in the classical sense, although formal analogies may arise. Rather, it is a structural measure of how many configurations remain compatible with the current constraints. A decrease in S_{eff} corresponds to refinement or information gain, while an increase corresponds to relaxation or loss of constraint.

The defect functional introduced previously,

$$\Delta_t : \Omega_t \to \mathbb{R}_{\geq 0},$$

provides a complementary perspective. While S_{eff} measures the size of the admissible region, Δ_t measures the degree of violation for inadmissible configurations. In optimization settings, one often selects configurations that minimize Δ_t, thereby approximating admissibility when exact feasibility is unattainable.

The interaction between Δ_t and S_{eff} reveals important structural features. A constraint update that sharply reduces S_{eff} typically corresponds to the introduction of a strong constraint, which may also increase the average defect over Ω_t. Conversely, relaxing constraints may increase S_{eff} while reducing the defect of previously inadmissible configurations.

An additional quantity of interest is invariant drift. Let $I_t = (I_t^{(1)}, \ldots, I_t^{(k)})$ be a collection of invariant monitors, where each $I_t^{(j)}$ is a function on the state. The drift between successive steps may be defined as

$$D_I(t) = \sum_{j=1}^{k} d_j(I_{t+1}^{(j)}, I_t^{(j)}),$$

where each d_j is a suitable metric on the range of $I_t^{(j)}$. This quantity measures the extent

to which invariants are preserved or violated during evolution.

The concept of ambiguity is also closely tied to projection. Let

$$\pi : (\Omega_t, C_t, H_t) \to X_t$$

be the observable map. The fiber

$$\pi^{-1}(X_t)$$

consists of all underlying states consistent with the observed summary. The size or measure of this fiber provides a notion of provenance ambiguity. Even if A_t is small, the projection may collapse multiple admissible states into a single observable, thereby increasing ambiguity at the level of X_t.

This leads to an important distinction between internal ambiguity, measured by S_{eff}, and observational ambiguity, induced by the projection π. A system may be internally well-constrained but observationally ambiguous if the projection discards relevant structure.

From an operational perspective, tracking ambiguity and defect provides a means of guiding system evolution. Constraint refinement can be directed toward reducing S_{eff}, while repair strategies can be evaluated by their effect on Δ_t. Invariant drift serves as a diagnostic for unintended structural changes, particularly in integration or merge tasks.

In summary, ambiguity and defect are dual aspects of admissibility. The admissible set determines what is possible, the defect functional measures deviation from feasibility, and the entropy-like measure S_{eff} quantifies how much uncertainty remains. Together, these quantities provide a quantitative language for reasoning about constraint-driven systems.

6 Dynamics and Evolution Operators

The evolution of the system is governed by operators acting on the structured state

$$Y_t = (\Omega_t, C_t, A_t, H_t, X_t, I_t).$$

Unlike classical formulations in which dynamics act solely on a state variable, the present framework requires simultaneous evolution of possibility, constraint, admissibility, and history.

A general transition is represented as

$$Y_{t+1} = \mathcal{T}_t(Y_t, u_t),$$

where u_t denotes an external input, intervention, observation, or internally generated operation. The operator $\mathcal{T}_t$ is not assumed to be time-invariant, nor is it assumed to admit an inverse.

Expanding component-wise, the transition may be decomposed into a family of update operators:

$$\Omega_{t+1} = U_\Omega(\Omega_t, C_t, u_t),$$

$$C_{t+1} = U_C(C_t, u_t),$$

$$A_{t+1} = \mathrm{Adm}(\Omega_{t+1}, C_{t+1}),$$

$$H_{t+1} = H_t \oplus e_{t+1},$$

$$X_{t+1} = \pi(\Omega_{t+1}, C_{t+1}, H_{t+1}),$$

$$I_{t+1} = U_I(I_t, Y_t, u_t).$$

Each component update has a distinct interpretation. The operator U_Ω modifies the possibility space, either by expanding it to incorporate new candidates or by restricting it based on newly available information. The operator U_C modifies the constraint family, introducing, removing, or reweighting constraints. The admissible set is then recomputed as the image of the admissibility operator applied to the updated possibility space and constraint family.

The history update is strictly append-only, reflecting the irreversibility of the process. The observable state is recomputed via the projection map, which may compress the updated structure. The invariant update operator U_I monitors changes in quantities expected to remain stable, allowing detection of structural drift.

The composition of transitions over multiple steps yields a sequence

$$Y_0 \xrightarrow{\mathcal{T}_0} Y_1 \xrightarrow{\mathcal{T}_1} \cdots \xrightarrow{\mathcal{T}_{t-1}} Y_t,$$

which defines a trajectory through the space of structured states. Because history accumulates and constraints may evolve, the admissibility of future states depends on the entire trajectory, not solely on the current configuration.

From an algebraic perspective, the collection of transition operators forms a semigroup under composition. Let $\mathcal{T}_{t,s}$ denote the composition of transitions from time t to time s, with $t < s$. Then

$$\mathcal{T}_{t,s} = \mathcal{T}_{s-1} \circ \cdots \circ \mathcal{T}_t,$$

and the composition law satisfies associativity. However, there is no requirement that each $\mathcal{T}_t$ be invertible, and therefore the structure is not, in general, a group.

The semigroup nature of evolution has significant implications. In particular, the absence of inverses implies that trajectories cannot be uniquely retraced, and that the mapping from initial states to final states may be many-to-one. This reinforces the distinction between replay, which uses retained history to reconstruct a trajectory, and reversal, which would require invertibility.

The dependence of U_Ω and U_C on u_t allows the framework to accommodate a wide range of processes. Observations may refine the possibility space by eliminating inconsistent configurations. Interventions may alter the constraint family by introducing new requirements. Internal operations may restructure both Ω_t and C_t to improve admissibility or reduce ambiguity.

The interaction between updates may also produce nontrivial effects. For example, a constraint update that tightens C_t may render previously admissible configurations inadmissible, thereby reducing A_t and decreasing S_{eff}. Conversely, an expansion of Ω_t may introduce new admissible configurations, increasing ambiguity. The dynamics of the system are therefore governed by the interplay between expansion, restriction, and constraint evolution.

In continuous settings, one may represent the evolution of components such as Ω_t or derived fields by differential equations. In discrete settings, update rules may be expressed algorithmically. The framework is agnostic to the specific form of the dynamics, provided that the separation between possibility, constraint, admissibility, and history is maintained.

In summary, the evolution operator acts not on a single state variable but on a structured tuple whose components interact under constraint-driven dynamics. The semigroup structure of transitions, combined with append-only history, provides a general model of irreversible evolution that accommodates both discrete and continuous systems.

7 Projection and Indistinguishability

A central structural feature of the framework is the presence of a projection from detailed state to observable state. Let

$$\pi : (\Omega_t, C_t, H_t) \to X_t$$

be the mapping that produces the observable representation. This projection may discard information, either by design or by necessity, and is therefore not assumed to be injective.

When π fails to be injective, there exist distinct underlying states that map to the same observable:

$$(\Omega_t, C_t, H_t) \neq (\Omega'_t, C'_t, H'_t) \quad \text{while} \quad X_t = X'_t.$$

The preimage of an observable state,

$$\pi^{-1}(X_t),$$

is referred to as the fiber over X_t. The structure and size of this fiber determine the degree of observational ambiguity.

This leads to a fundamental indistinguishability phenomenon. Consider an observer whose access to the system is restricted to X_t. If all observable quantities are functions of X_t alone, then any two underlying states in the same fiber are indistinguishable to that observer. Formally, let f be any observable function satisfying

$$f = \tilde{f} \circ \pi$$

for some function $\tilde{f}$ on the observable space. Then

$$\pi(Y_t) = \pi(Y_t') \implies f(Y_t) = f(Y_t').$$

This motivates a distinction between observational equivalence and structural equivalence. Two states are observationally equivalent if they yield the same observable output, but they are structurally equivalent only if they coincide in all components relevant to the system's semantics, including history and constraint structure.

The following theorem formalizes the indistinguishability phenomenon.

Theorem 7.1 (Indistinguishability Under Projection)**.** *Let $\mathcal{S}_t$ denote the space of detailed states and $\pi : \mathcal{S}_t \to \mathcal{X}_t$ a projection map. Suppose that all observable functions $f : \mathcal{S}_t \to \mathbb{R}$ factor through π, so that $f = \tilde{f} \circ \pi$ for some $\tilde{f}$.*

Then for any $Y_t, Y_t' \in \mathcal{S}_t$ such that

$$\pi(Y_t) = \pi(Y_t'),$$

it follows that

$$f(Y_t) = f(Y_t')$$

for all observable functions f. Consequently, no observer restricted to $\mathcal{X}_t$ can distinguish Y_t from Y_t'.

Proof. By assumption, every observable function f factors as $f = \tilde{f} \circ \pi$. Therefore,

$$f(Y_t) = \tilde{f}(\pi(Y_t)), \quad f(Y_t') = \tilde{f}(\pi(Y_t')).$$

If $\pi(Y_t) = \pi(Y_t')$, then

$$\tilde{f}(\pi(Y_t)) = \tilde{f}(\pi(Y_t')),$$

which implies

$$f(Y_t) = f(Y_t').$$

This holds for all observable functions f, establishing indistinguishability. □

This result has several implications. First, it shows that indistinguishability is not a matter of insufficient measurement but a structural property of the projection. Second, it implies that any attempt to reconstruct underlying state from observables alone must contend with the multiplicity of the fiber. Third, it clarifies that equality at the level of observables does not imply equality at the level of admissible trajectories.

The size or measure of the fiber $\pi^{-1}(X_t)$ provides a quantitative notion of observational ambiguity. In particular, even if the admissible set A_t is small, the projection may collapse multiple admissible states into a single observable, thereby increasing ambiguity at the level of X_t.

From a practical perspective, this analysis motivates the retention of additional structure when provenance matters. If the projection discards information that is relevant for future inference or decision-making, then the resulting ambiguity cannot be resolved without external input. Conversely, if the projection is sufficiently rich to preserve the distinctions of interest, then observational equivalence may coincide with structural equivalence for the purposes at hand.

In summary, projection introduces a layer of abstraction that enables compact representation but at the cost of potential ambiguity. The indistinguishability theorem formalizes this trade-off and underscores the necessity of distinguishing between observable and structural notions of equality.

8 Semantic Merge and Structural Integration

The integration of independently evolved artifacts presents a fundamental challenge when structure, constraints, and history are taken as primary. Traditional merge procedures operate on representations, typically treating artifacts as sequences of symbols and resolving conflicts through local heuristics. Such methods implicitly assume that semantic compatibility follows from syntactic compatibility. Within the present framework, this assumption is generally invalid.

Let A and B be two artifacts derived from a common predecessor P. Each artifact induces a structured state, including a possibility space, a constraint family, an admissible region, and a history trace. The task of merge is to construct a new artifact M that preserves as much compatible structure as possible while resolving or exposing incompatibilities.

To formalize this, associate to each artifact the following collections. Let $K(A)$ denote the set of claims or assertions encoded by A. Let $I(A)$ denote the set of invariants preserved by A. Let $O(A)$ denote the set of operators, transformations, or interfaces defined by A. Let $H(A)$ denote the history or provenance trace.

A semantic merge seeks an artifact M such that the induced structures satisfy compatibility conditions with respect to both A and B. In general, no artifact will satisfy all such conditions simultaneously, and therefore merge must be formulated as an optimization or selection problem.

Define a cost functional

$$J(M) = \lambda_1 D_K(M; A, B) + \lambda_2 D_I(M; A, B) + \lambda_3 D_O(M; A, B) + \lambda_4 D_H(M; A, B),$$

where each term measures a different form of deviation. The quantity D_K measures the loss or contradiction of claims. The quantity D_I measures invariant drift, capturing the extent to which preserved properties are altered. The quantity D_O measures mismatch in operators or interfaces. The quantity D_H measures the loss or distortion of provenance.

The weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ encode priorities among these objectives. Different applications may assign different relative importance to claim preservation, invariant stability,

interface compatibility, and provenance retention.

A merge is considered successful when a minimizer M^* of J satisfies acceptable thresholds for each component. However, it is essential to recognize that a minimizer need not correspond to a coherent artifact in the absence of compatibility. In such cases, the correct outcome is not forced integration but explicit representation of the obstruction.

To analyze compatibility, it is useful to distinguish several classes of conflict. Conflicts may arise from differences in notation or naming, which are often resolvable by relabeling. Conflicts may arise from differences in ordering or implementation, which may be reconciled through refactoring. More fundamentally, conflicts may arise from incompatible constraints or invariants, in which case no single artifact can satisfy both without modification.

Let A_t and B_t denote the admissible sets induced by the two artifacts. A necessary condition for a conflict-free merge is that the intersection

$$A_t^{(A)} \cap A_t^{(B)}$$

be nonempty. If this intersection is empty, the combined constraint family is inconsistent, and the system is overconstrained. In such cases, resolution requires modification of the constraint sets rather than mere reconciliation of representations.

The role of history is particularly significant in merge. Two artifacts may produce identical observable outputs while differing in their histories. If provenance is relevant, then a merge that discards historical distinctions introduces ambiguity. The cost term D_H captures this effect and penalizes merges that collapse distinct histories without justification.

The process of merge may also be viewed through the lens of admissible trajectories. Each artifact corresponds to a trajectory through its respective state space. A merged artifact must correspond to a trajectory that is admissible under a combined or reconciled constraint family. When such a trajectory does not exist, the obstruction should be made explicit rather than obscured.

In some cases, it is advantageous to represent the result of merge not as a single artifact but as a structured pair or family, together with a mapping that identifies shared structure and records differences. This approach preserves information about incompatibilities and allows for deferred resolution.

From a broader perspective, semantic merge is an instance of structural integration. It requires aligning possibility spaces, reconciling constraint families, preserving invariants, and managing history. The framework developed here provides a language for expressing these requirements and for analyzing the conditions under which integration is possible.

In summary, merge is not the combination of texts but the reconciliation of constrained structures. Its success depends on the compatibility of admissible sets, the preservation of invariants, and the careful treatment of provenance. When these conditions are not met, the correct outcome is not forced unification but explicit recognition of structural incompatibility.

9 Field Representations and Continuous Limits

While the framework is formulated in terms of discrete structures such as possibility spaces and constraint families, it admits a natural extension to continuous representations. In many systems, particularly those with large or infinite possibility spaces, it is advantageous to represent structure through fields defined over a domain.

Let $\mathcal{D}$ be a spatial, abstract, or parameter domain. A field representation associates to each point $x \in \mathcal{D}$ a collection of quantities that encode aspects of the system. A particularly useful representation is given by a triple

$$(\Phi(x,t), \mathbf{v}(x,t), S(x,t)),$$

where each component has a structural interpretation.

The scalar field Φ represents density, amplitude, coherence, or intensity. It may be interpreted as a measure of concentration of admissible configurations or as a weighting over the possibility space. The vector field $\mathbf{v}$ represents flow, transport, or directional tendency, encoding how structure moves or propagates through the domain. The scalar field S represents defect or entropy-like structure, capturing dispersion, ambiguity, or deviation from admissibility.

The evolution of these fields may be described by coupled differential equations of the form

$$\partial_t \Phi = D_\Phi \nabla^2 \Phi - \nabla \cdot (\Phi \mathbf{v}) + R_\Phi(\Phi, \mathbf{v}, S),$$

$$\partial_t \mathbf{v} = D_v \nabla^2 \mathbf{v} + R_v(\Phi, \mathbf{v}, S),$$

$$\partial_t S = D_S \nabla^2 S + P_S(\Phi, \mathbf{v}, S) - Q_S(\Phi, \mathbf{v}, S),$$

where D_Φ, D_v, D_S are diffusion coefficients and R_Φ, R_v, P_S, Q_S are reaction terms that encode interactions among the fields.

This system is not intended as a specific physical law but as a structural template. The diffusion terms represent smoothing or spreading of quantities across the domain. The transport term $-\nabla \cdot (\Phi \mathbf{v})$ represents advection, corresponding to the movement of structure along the vector field. The reaction terms encode local transformations, including generation and dissipation of defect.

The connection between the field representation and the discrete framework can be made explicit. The scalar field Φ may be viewed as a continuous analogue of a distribution over the admissible set A_t. The defect field S corresponds to a spatially distributed version of the defect functional Δ_t. The vector field $\mathbf{v}$ encodes directional bias in the evolution of the system, analogous to update operators acting on the possibility space.

In discrete settings, similar structures may be defined on graphs or lattices. Let $G = (V, E)$ be a graph representing relationships among configurations or regions of the

possibility space. One may define functions

$$\Phi : V \to \mathbb{R}, \quad \mathbf{v} : E \to \mathbb{R}, \quad S : V \to \mathbb{R},$$

with update rules that mimic diffusion, transport, and reaction. Such discrete field models provide computationally tractable approximations of continuous dynamics.

The field perspective is particularly useful when the system exhibits large-scale structure or when local interactions give rise to global behavior. It allows one to analyze patterns such as clustering, dispersion, and flow without enumerating individual configurations. It also provides a bridge to established techniques in partial differential equations, statistical mechanics, and dynamical systems.

However, the use of field representations must be justified by the structure of the system. Not all systems admit a meaningful embedding into a continuous domain, and forcing such an embedding may obscure rather than clarify the underlying dynamics. The framework therefore treats the field representation as an optional mapping rather than a mandatory component.

When applicable, the field representation provides insight into the interplay between constraint and evolution. Regions of high Φ correspond to concentrations of admissible structure, while regions of high S indicate ambiguity or defect. The vector field $\mathbf{v}$ directs the flow of structure, and the coupled dynamics describe how admissibility and defect evolve over time.

In summary, the field representation offers a continuous analogue of the discrete framework, capturing the evolution of structure through diffusion, transport, and reaction. It extends the applicability of the framework to systems where large-scale or continuous descriptions are more natural, while preserving the core principles of constraint, admissibility, and irreversibility.

10 A Category-Theoretic Perspective

The structural features of the framework admit a natural, though deliberately restrained, categorical interpretation. The purpose of this perspective is not to rephrase all constructions in categorical language, but to identify when structure-preserving relationships can be formalized as morphisms and when integration tasks resemble universal constructions.

Let $\mathcal{A}$ denote a category whose objects are structured states $Y = (\Omega, C, A, H, X, I)$ satisfying the admissibility conditions described previously. A morphism

$$f : Y \to Y'$$

is required to preserve the relevant structure of the state. The precise definition of preservation depends on the application, but generally includes the following properties.

First, the morphism must relate possibility spaces in a manner compatible with admissibility. If $f_\Omega : \Omega \to \Omega'$ is the induced map on possibility spaces, then it should

satisfy

$$f_\Omega(A) \subseteq A',$$

ensuring that admissible configurations are mapped to admissible configurations.

Second, the morphism must respect constraints. If C and C' are constraint families, then the action of f should map constraints in C to constraints in C' in a way that preserves their effect on admissibility. This may be expressed through commutation conditions involving the admissibility operator.

Third, the morphism must account for history. In general, a map on histories

$$f_H : H \to H'$$

should preserve the append-only structure, meaning that concatenation is respected. In many cases, this requirement restricts morphisms to those that embed or extend histories rather than collapse them.

Fourth, the morphism must be compatible with the observable projection. If π and π' are the projection maps for Y and Y', respectively, then one typically requires that

$$\pi' \circ f = g \circ \pi$$

for some map g on observable states. This ensures that the morphism does not introduce inconsistency between underlying structure and observable representation.

Under these conditions, morphisms represent structure-preserving transformations between systems. Composition of morphisms corresponds to successive transformations, and the identity morphism corresponds to the trivial transformation that preserves all components.

Within this categorical setting, integration tasks such as merge can be interpreted in terms of universal constructions. Suppose two objects A and B share a common substructure P, represented by morphisms

$$P \to A, \quad P \to B.$$

A merge, when it exists, may be viewed as a colimit of this diagram, producing an object M together with morphisms

$$A \to M, \quad B \to M$$

that satisfy the appropriate universal property.

However, the existence of such a colimit is not guaranteed. If the structures of A and B are incompatible, then no object M can satisfy all required conditions. In this case, the failure of the colimit reflects a genuine obstruction in the underlying structure, rather than a deficiency of representation.

The categorical perspective also clarifies the role of invariants. An invariant may be viewed as a property that is preserved under all morphisms in a specified class. This aligns

with the earlier notion of invariant monitors, providing a formal mechanism for identifying which aspects of a system remain stable under admissible transformations.

It is important to emphasize that the categorical formulation is a tool for expressing structure, not an end in itself. The introduction of objects, morphisms, and universal properties is justified only when it corresponds to genuine relationships among the components of the system. When such relationships are absent or unclear, the categorical language should not be imposed.

In summary, the category-theoretic perspective provides a framework for reasoning about structure-preserving transformations and integration. It formalizes the notion of compatibility and obstruction and connects merge operations to universal constructions. At the same time, it reinforces the principle that formal language must be grounded in explicit structure, ensuring that abstraction does not outpace meaning.

11 Applications and Practical Consequences

The abstract structure developed in the preceding sections yields concrete consequences across domains in which constraint, ambiguity, and irreversibility play central roles. The framework provides a unified language for analyzing inference, system design, and integration, and it clarifies limitations that arise when observable state is treated as sufficient.

In inference problems, the distinction between admissible sets and observable projections becomes decisive. Given an observation X_t, the task of inference is to characterize the fiber

$$\pi^{-1}(X_t) \cap \mathcal{A}_t,$$

where $\mathcal{A}_t \subseteq \mathcal{S}_t$ denotes the admissible region in the full state space. When this intersection contains multiple elements, inference is inherently ambiguous. The framework therefore shifts emphasis from identifying a single solution to characterizing the admissible set of solutions and quantifying its structure through measures such as S_{eff}. This perspective aligns naturally with inverse problems, Bayesian inference, and constraint satisfaction, while making explicit the role of projection-induced degeneracy.

In system design, the explicit separation of state components leads to improved handling of provenance and irreversibility. Systems that retain only the observable state X_t may be efficient in storage but are unable to answer questions that depend on history. By maintaining an append-only trace H_t, one enables replay, audit, and provenance-sensitive reasoning. The framework thus provides a principled basis for the design of logging systems, version control mechanisms, and stateful computational processes.

In integration tasks, particularly those involving multiple evolving artifacts, the notion of semantic merge replaces purely syntactic combination. The requirement that admissible sets intersect and that invariants be preserved imposes structural constraints on integration. When these constraints are incompatible, the framework prescribes explicit representation

of the conflict rather than forced resolution. This approach is applicable to software integration, knowledge synthesis, and collaborative modeling, where hidden inconsistencies can lead to subtle failures.

The framework also informs the analysis of model equivalence. Two models that produce identical observable outputs may differ in their underlying admissible structures or histories. Such models are observationally equivalent but structurally distinct. Recognizing this distinction is crucial in contexts where internal consistency, interpretability, or provenance matters, such as scientific modeling or machine learning.

In simulation, the explicit representation of constraints and admissibility allows for the construction of models that respect structural invariants by design. Rather than relying solely on numerical stability or empirical validation, one can enforce admissibility conditions at each step of the simulation. This reduces the risk of drift into inadmissible regions and provides a clearer interpretation of simulation results.

The field representation introduced earlier extends these applications to systems with continuous structure. In such cases, the scalar and vector fields provide a compact description of admissible density, flow, and defect. This enables the study of large-scale behavior without enumerating individual configurations, and it connects the framework to established methods in partial differential equations and dynamical systems.

Finally, the framework has implications for the design of reasoning systems, including automated and semi-automated agents. By structuring reasoning in terms of possibility spaces, constraint families, admissible sets, and history, one can design systems that are robust to ambiguity and explicit about their assumptions. Such systems can track the evolution of constraints, identify underdetermined regions, and avoid conflating observational equivalence with structural identity.

In summary, the practical consequences of the framework arise from its insistence on explicit structure. By distinguishing possibility, constraint, admissibility, history, and observation, the framework provides tools for managing ambiguity, preserving provenance, and ensuring structural consistency. These tools apply broadly across domains, offering a coherent approach to problems that are otherwise treated in isolation.

12 Conclusion

A general framework has been developed in which systems are modeled as constrained evolutions of possibility rather than as isolated states. The central object, a structured state consisting of a possibility space, a constraint family, an admissible subset, an append-only history, and an observable projection, provides a unified representation in which ambiguity, irreversibility, and provenance are explicit.

Several key principles emerge from this formulation. First, admissibility is primary. The structure of what is possible at a given moment is determined by the interaction between the possibility space and the constraint family, and this structure governs all subsequent evolution. Second, irreversibility is intrinsic. The accumulation of history

through append-only operations implies that reconstruction cannot be reduced to inversion, and that provenance must be treated as a fundamental component of identity. Third, projection introduces ambiguity. Observable states are generally compressions of richer underlying structures, and equality at the level of observables does not imply structural equivalence.

These principles lead to a reorganization of several standard concepts. Inference becomes the characterization of admissible sets rather than the identification of single solutions. Integration becomes the reconciliation of constrained structures rather than the combination of representations. Dynamics become semigroup actions on structured states rather than reversible transformations on points.

The framework also provides quantitative tools, including entropy-like measures of residual ambiguity and defect functionals that quantify deviation from admissibility. These tools enable the analysis of constraint refinement, invariant preservation, and the trade-offs involved in system evolution. The introduction of field representations extends the framework to continuous settings, while the category-theoretic perspective clarifies the conditions under which structure-preserving transformations and integrations are possible.

Beyond its formal contributions, the framework has practical implications. It informs the design of systems that must manage ambiguity and provenance, supports more reliable integration of independently developed artifacts, and provides a disciplined approach to reasoning under incomplete information. By making explicit the distinctions that are often implicit or ignored, it reduces the risk of conflating structurally distinct configurations and of overlooking the consequences of information loss.

The broader significance of the framework lies in its shift of emphasis. Rather than treating structure as an attribute of objects, it treats structure as arising from constraints on trajectories. This shift aligns with a wide range of phenomena in which path dependence, information loss, and underdetermination are central. It suggests that a more faithful account of such systems requires attention not only to what is observed, but to what remains admissible, what has been discarded, and how the present state has been reached.

Future work may extend the framework in several directions. One avenue is the development of more refined measures of ambiguity and defect, particularly in high-dimensional or non-Euclidean spaces. Another is the formalization of reconstruction procedures under partial observability, including conditions for identifiability. A further direction is the integration of probabilistic structure, allowing the framework to accommodate uncertainty in both constraints and observations. Finally, the categorical perspective may be developed to capture more general forms of composition and interaction, provided that such developments remain grounded in explicit structural relationships.

In closing, the framework presented here offers a coherent approach to systems in which constraint, admissibility, and history are fundamental. By treating these elements as primary, it provides both a descriptive language and an operational methodology for analyzing and constructing complex systems across domains.

Appendices

A Appendix A: Realization in the `protocol` Directory

The abstract framework developed in the main text is instantiated concretely in the project directory `protocol`. The directory structure reflects a direct decomposition of the formal state

$$Y_t = (\Omega_t, C_t, A_t, H_t, X_t, I_t)$$

into operational components. Each subdirectory corresponds not to an implementation artifact in the conventional sense, but to a structural role within the framework.

The `core` directory contains the primary definitions governing admissibility, irreversibility, and state structure. The files `ontology.md`, `constraints.md`, and `state-model.md` correspond directly to the formal definitions of Ω_t, C_t, and A_t, while `irreversibility.md` encodes the append-only structure of H_t. The file `metrics.md` provides operational forms of the defect functional Δ_t and the ambiguity measure S_{eff}.

The `workflows` directory implements transition operators $\mathcal{T}_t$ in a procedural form. Each workflow corresponds to a structured transformation of the state. For example, `constraint-analysis.md` operationalizes the computation of $A_t = \mathrm{Adm}(\Omega_t, C_t)$, while `trajectory-reconstruction.md` addresses inference over the fiber $\pi^{-1}(X_t)$. These workflows act as explicit realizations of the update operators described in the dynamics section.

The `agents` directory provides specialized morphisms acting on structured states. Each agent encodes a restricted transformation that preserves specific components of the state. The `constraint-auditor` enforces consistency of C_t and A_t, the `interpreter` constructs mappings from raw artifacts into structured representations, the `merge-specialist` operates on pairs of states to produce candidate merges, and the `proof-assembler` transforms informal claims into formal structures. These agents can be understood as partial endomorphisms on the category of structured states.

The `commands` directory defines standardized projections from the internal state to structured outputs. Each command corresponds to a particular observable map π composed with a formatting constraint. For example, `constraint-check.md` produces a decomposition of Ω_t, C_t, and A_t, while `history-audit.md` explicitly analyzes the dependence of X_t on H_t. These commands ensure that outputs remain aligned with the internal structure rather than collapsing into unstructured text.

The `mappings` directory implements auxiliary representations of the state. The file `history.md` elaborates the role of H_t and its equivalence classes, `merge.md` formalizes the cost functional $J(M)$, and `rsvp.md` provides the field-theoretic mapping to $(\Phi, \mathbf{v}, S)$. The file `category.md` encodes the categorical interpretation of morphisms and integration. These mappings serve as alternative coordinate systems on the same underlying structure.

The `schemas` directory provides machine-readable encodings of the state and event structures. The file `state.schema.json` corresponds directly to the tuple Y_t, while `event.schema.json` encodes elements of H_t. The merge report schema formalizes the

output of semantic merge as a structured object aligned with the cost functional J. These schemas ensure that the abstract definitions can be instantiated in computational systems without loss of structure.

The `examples` directory provides minimal realizations of the framework. The file `minimal-state.json` instantiates a concrete Y_t, while `minimal-merge-report.json` demonstrates the decomposition of merge outcomes into structural components. The worked note illustrates the indistinguishability phenomenon by exhibiting two implementations with identical X_t but distinct H_t.

Finally, the `adapters` directory provides thin layers that map the framework into specific execution environments. These adapters do not redefine the protocol but rather provide entry points through which external systems can operate within the structured framework.

The directory as a whole constitutes a realization of the abstract framework in which each component has a direct correspondence to a formal object. The separation of concerns enforced by the directory structure mirrors the separation of components in the state Y_t, ensuring that implementation does not collapse distinct structural roles.

B Appendix B: Structural Completeness and Remaining Gaps

While the `protocol` directory provides a faithful realization of the framework, several aspects remain incomplete from a mathematical and operational perspective. These gaps are not deficiencies but indicators of directions in which the framework may be extended.

One such gap concerns the explicit characterization of the admissibility operator Adm. In the current implementation, admissibility is specified procedurally within workflows rather than as a fully formal operator with well-defined algebraic properties. A more complete treatment would define classes of admissibility operators with known properties, such as monotonicity, closure, or stability under composition.

A second gap concerns the formal structure of the possibility space Ω_t. In practice, Ω_t is often implicit, represented by a set of candidate interpretations or configurations. A more rigorous implementation would specify Ω_t as a measurable or topological space, enabling precise definitions of $\mu(A_t)$ and S_{eff}.

A third gap arises in the treatment of history equivalence. While the framework distinguishes between observational and historical equivalence, the implementation does not yet provide a general mechanism for defining equivalence classes on H_t. Such a mechanism would require a formal language for events and a notion of equivalence that respects the semantics of the system.

The merge functional $J(M)$ is currently specified at a conceptual level, with the components D_K, D_I, D_O, D_H defined informally. A complete implementation would require explicit metrics for each component, along with algorithms for computing or approximating the minimizer M^*. This is particularly challenging in high-dimensional or combinatorial spaces.

The field representation provides a powerful mapping, but its connection to discrete implementations remains only partially specified. In particular, the correspondence between discrete updates in workflows and continuous dynamics in the field equations has not been fully formalized. Establishing this correspondence would require a theory of discretization and scaling limits.

The categorical perspective is also incomplete. While the notion of morphisms preserving structure is introduced, the category $\mathcal{A}$ has not been fully characterized. Questions regarding limits, colimits, and functorial relationships between different representations remain open. A more developed categorical treatment could provide deeper insight into the composition and integration of systems.

Finally, the treatment of uncertainty remains largely structural rather than probabilistic. While ambiguity is quantified through S_{eff}, the framework does not yet incorporate probabilistic weights or measures on Ω_t in a fully integrated manner. Extending the framework to include probability distributions over admissible sets would connect it more directly to statistical inference and decision theory.

Despite these gaps, the current implementation captures the essential features of the framework. It enforces the separation of possibility, constraint, admissibility, history, and observation, and it provides operational procedures that respect this structure. The remaining work consists in refining these components, formalizing their interactions, and extending the framework to cover broader classes of systems.

In this sense, the `protocol` directory should be understood not as a finished system but as a scaffold. It encodes a set of principles that are already operational, while leaving open the possibility of further mathematical development and practical refinement.

References

[1] S. Mac Lane, *Categories for the Working Mathematician*, 2nd ed., Springer, 1998.

[2] S. Awodey, *Category Theory*, 2nd ed., Oxford University Press, 2010.

[3] T. Jacobson, "Thermodynamics of Spacetime: The Einstein Equation of State," *Physical Review Letters*, vol. 75, no. 7, pp. 1260–1263, 1995.

[4] E. Verlinde, "On the Origin of Gravity and the Laws of Newton," *Journal of High Energy Physics*, 2011.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed., Wiley, 2006.

[6] E. T. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press, 2003.

[7] W. Rudin, *Functional Analysis*, 2nd ed., McGraw-Hill, 1991.

[8] L. C. Evans, *Partial Differential Equations*, 2nd ed., American Mathematical Society, 2010.

[9] K.-J. Engel and R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations*, Springer, 2000.

[10] A. N. Kolmogorov, "Foundations of the Theory of Probability," Chelsea Publishing, 1956.

[11] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[12] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, 1936.

[13] 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.

[14] R. Milner, *Communicating and Mobile Systems: The Pi-Calculus*, Cambridge University Press, 1999.

[15] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice Hall, 1985.

[16] D. I. Spivak, *Category Theory for the Sciences*, MIT Press, 2014.

[17] J. Lurie, *Higher Topos Theory*, Princeton University Press, 2009.

[18] R. Hartshorne, *Algebraic Geometry*, Springer, 1977.

[19] M. Gromov, *Metric Structures for Riemannian and Non-Riemannian Spaces*, Birkhäuser, 1999.

[20] E. Bishop and D. Bridges, *Constructive Analysis*, Springer, 1985.