Procedural Justice in Heterogeneous Conversion: A Note on Noncommutative Ordering

Wanhong HUANG
huangwanhong@serendip.ngo

Abstract

Heterogeneous conversion procedures can involve recognition, classification, attribution, valuation, authorization, conversion, settlement, contestation, and revision. These operations need not be procedurally interchangeable. An earlier operation can alter the relational configuration on which a later operation acts, change which later operations are admissible, or modify the institutional meaning of their outputs. Recognition before conversion can therefore generate a different history from recognition after conversion; revision before final settlement can differ from revision after settlement; and in some cases one ordering is procedurally available while the reverse ordering is not.

This discussion note develops a limited formal language for this order-sensitive dimension of procedural justice. It represents conversion-relevant operations as partially defined, state-dependent transformations and distinguishes three structures: mutual compositions with different terminal configurations, distinct procedural paths that can share a terminal configuration, and one-way composability in which one ordered sequence is unavailable. The term noncommutativity is reserved for the first of these compositional cases; partial composability captures the accessibility asymmetry of the third. The analysis is algebra-inspired but does not assume a linear operator algebra or a physical quantum interpretation. Its purpose is to clarify procedural structure rather than derive a substantive justice ranking from ordering alone.

The paper examines recognition, classification, valuation, conversion, settlement, and revision, with particular attention to irreversibility and retrospective recognition. It then shows how procedural theories can evaluate the resulting histories through criteria such as standing, informational conditions, participation, review, and repair. The argument remains deliberately narrow: order-sensitive procedural composition identifies one possible locus of procedural significance within heterogeneous conversion and leaves broader questions of distributive justice, epistemic justice, commensurability, power, and substantive conversion norms to the wider conversion-justice literature.

Keywords: procedural justice; heterogeneous conversion; noncommutative ordering; partial composability; recognition

Discussion Paper Note

This manuscript isolates one limited issue within the broader problem of justice in heterogeneous value conversion: the procedural significance of order-sensitive operations. It does not propose a general theory of conversion justice, a complete theory of procedural justice, or a substantive criterion for ranking conversion outcomes. The discussion asks how recognition, classification, valuation, conversion, settlement, and revision can generate different relational histories when their order changes.

The paper uses a noncommutative-algebra-inspired language in a restricted sense. Strict compositional noncommutativity refers here to mutually executable ordered compositions with different terminal configurations. The paper separately treats path divergence and one-way composability, which should not be collapsed into that strict notion. No conventional algebraic commutator is introduced because the required linear structure is not assumed. The mathematical language is therefore representational rather than a claim that social or legal procedures instantiate quantum-mechanical operator algebra.

The companion papers, *A Preliminary Formalism for Heterogeneous Value Conversion in the Generative Relational Framework* and *Heterogeneous Value Conversion and Justice: A Generative Relational Discussion*, provide the broader formal and normative context. The present note reproduces only the concepts required for the narrow ordering problem.

Responsible Use and Rights Reservation

Notices

Status. This manuscript is a focused discussion note on one procedural dimension of heterogeneous value conversion. It proposes a compositional language for order-sensitive operations and partial composability. It does not offer a complete theory of conversion justice, procedural justice, or social ontology, and it does not infer a justice ranking directly from structural order sensitivity.

Statement on the use of language models. The exploratory discussions and preparation of this paper involved OpenAI's ChatGPT. ChatGPT supported exploratory dialogue, formal reconstruction, argumentative criticism, source discovery followed by verification, and drafting in LaTeX. The author selected the research questions, directed and approved the theoretical commitments and epistemic status of the claims, and bears sole responsibility for the manuscript, including its arguments, conclusions, and errors. Authorship credit remains with the human author.

Contents

1 Introduction

This paper isolates a narrow procedural problem within heterogeneous value conversion: the normative significance of the order in which conversion-relevant operations occur. A conversion process may contain recognition, classification, attribution, valuation, authorization, negotiation, transfer, settlement, contestation, and revision. A simplified account can represent these operations as acting on a fixed object in an interchangeable sequence. The present discussion considers cases in which an earlier operation changes the relational configuration on which a later operation acts. In such cases, reversing the order can produce a different conversion history even when a coarse description records similar parties, similar value forms, or a similar final transfer.

The motivating example is deliberately simple. Consider a value-bearing contribution whose significance becomes institutionally recognizable either before or after a conversion. Recognition before conversion can alter bargaining position, evidentiary standing, available classifications, admissible claims, or the terms under which conversion occurs. Recognition after conversion acts on a configuration in which rights, possession, obligations, expectations, or settlement status may already have changed. The two sequences therefore need not be procedurally equivalent. The difference does not establish by itself that one sequence is just. It identifies an ordering difference whose normative relevance must be supplied by a procedural account.

The paper develops this point through a compositional language. Let Ω denote a space of relational configurations and let Φ_i denote partially defined transformations on that space. A conversion-relevant operation is represented as a partial map whose domain depends on the current configuration. Two operations Φ_i and Φ_j are terminally order-sensitive at a configuration ω when both compositions are defined there and Equation 1 holds.

$$\Phi_j(\Phi_i(\omega)) \neq \Phi_i(\Phi_j(\omega)) \,. \tag{1}$$

A distinct accessibility asymmetry arises when one ordering is admissible while the reverse ordering is undefined because the first operation creates, removes, or modifies the conditions required by the second. The paper refers to this structure as *partial composability*. Both forms are relevant to procedure because institutional processes can establish standing, create rights, close claims, alter evidence, or change available avenues of revision.

The use of *noncommutative ordering* in the title refers to the strict mutual-composability case in which both ordered compositions are executable and yield different terminal configurations. The paper uses *order-sensitive procedure* as the wider umbrella for that case, path divergence with a shared terminal configuration, and one-way composability. The argument does not require a vector space, Hilbert space, or a subtraction operation between transformations. Accordingly, the paper does not introduce the algebraic commutator $[\widehat{A}, \widehat{B}]$. Such an object would require additional algebraic structure specifying how transformations can be added or subtracted. This distinction keeps the mathematical claim aligned with the compositional structure actually used in the paper.

The relational dependence of later operations strengthens the ordering problem. If recognition transforms ω_0 into ω_R, conversion after recognition is a transformation whose effective domain and meaning are evaluated at ω_R. Reversing the sequence first produces a conversion-transformed configuration ω_X, after which recognition occurs under different conditions. Equation 2 represents the resulting terminal configurations schematically.

$$\begin{aligned} \omega_{RX} &= \Phi_X^{[\omega_R]}(\omega_R), & \qquad \omega_R &= \Phi_R^{[\omega_0]}(\omega_0), \\ \omega_{XR} &= \Phi_R^{[\omega_X]}(\omega_X), & \qquad \omega_X &= \Phi_X^{[\omega_0]}(\omega_0). \end{aligned} \tag{2}$$

The superscripts in Equation 2 indicate the relational configuration relative to which each concrete transformation is instantiated. Order sensitivity therefore concerns more than exchanging two fixed functions. The first operation can alter the configuration, the availability of later operations, and the institutional significance of their outputs.

This structure is particularly relevant when conversion contains changes that are irreversible or costly to reverse relative to the modeled intervention repertoire. Transfer of ownership, final settlement, environmental transformation, disclosure of information, loss of an opportunity, or dissipation of a resource can make later recognition or revision unable to reconstruct selected features of the history that would have resulted from an earlier intervention. Procedural analysis therefore requires attention to the position of recognition, information, consent, settlement, and revision within the conversion history rather than only to the final observable allocation.

The paper maintains a deliberately limited normative ambition. Strict compositional noncommutativity does not determine a justice ordering. Different procedural theories can assign different significance to standing, informed participation, equality of hearing, consent, review, finality, or repair. The present formal language identifies where order-sensitive divergence or accessibility asymmetry occurs and provides common structural objects on which those theories can operate. It therefore treats order-sensitive composition, including strict noncommutativity as one case, as a diagnostic structure for procedural inquiry rather than a substantive justice criterion.

The discussion is also narrower than the broader problem of heterogeneous conversion justice. Questions of ontology, commensurability, attribution, distributive incidence, epistemic justice, power, and conversion nomos remain relevant, but they enter this paper only where they affect procedural ordering. A conversion may be unjust for reasons unrelated to order sensitivity, and a strictly noncommutative ordering may be normatively acceptable under a particular theory. This scope discipline permits the paper to focus on one small but analytically distinct feature of conversion procedure.

The remainder of the paper proceeds in stages. Section 2 locates procedural ordering within heterogeneous conversion. Section 3 introduces partial maps, composition, partial composability, and state-dependent transformation. Section 4 specifies the reduced Generative Relational representation used in the note. Section 5 examines recognition, classification, valuation, conversion, settlement, and revision as order-sensitive operations. Section 6 considers irreversibility and retrospective recognition. Section 7 connects the structural analysis to procedural-justice questions without selecting a universal criterion. Section 8 develops illustrative conversion cases, and Section 9 states the formal and normative limits of the proposal before the conclusion.

2 Heterogeneous Conversion and Procedural Structure

This section establishes the procedural vocabulary used throughout the paper before introducing partial maps and composition in Section 3. Its role is to separate three levels that are easily conflated in discussions of heterogeneous conversion: the substantive objects being converted, the sequence of operations through which conversion becomes institutionally possible, and the procedural questions that arise from that sequence. The discussion is descriptive and structural at this stage. It identifies the operations whose ordering may later acquire normative significance without assigning a justice ranking to any ordering.

2.1 Conversion Histories and Procedural Order

A heterogeneous conversion episode can contain more procedural structure than is visible in an expression such as $A \rightsquigarrow B$. The visible expression records two manifested value forms and a relation of conversion between them. A procedure may additionally determine which entity is recognized as the bearer of A, which classification is applied to that entity, which evidence is admitted, how a valuation is constructed, who has standing to negotiate, when a transfer becomes effective, and whether later contestation can reopen the settlement. The present note treats these operations as analytically separable even when a practical institution performs several of them together.

For the limited purpose of recording order before a full state-transition semantics is introduced, Definition 2.1 defines a finite alphabet of operation types. The list is illustrative rather than exhaustive.

Definition 2.1 (Procedural operation alphabet)**.** Let $\mathcal{L}_P$ denote a set of labels for conversion-relevant procedural operations. The working alphabet used in this paper is stated in Equation 3.

$$\mathcal{L}_P = \{R, K, A, V, U, N, X, S, Q\}. \tag{3}$$

The labels denote, respectively, recognition, classification, attribution, valuation, authorization, negotiation, conversion or transfer, settlement, and contestation or revision. A label identifies an operation type only; it does not yet specify a transformation on a state space.

Definition 2.1 deliberately separates the broad expression *conversion procedure* from the narrower operation X, which denotes the transfer or conversion act within that procedure. The distinction prevents the whole procedure from being collapsed into the transfer event that is most visible at the end of the process.

A procedural ordering can now be represented without yet assuming that each label is realized by a total function. Definition 2.2 introduces the corresponding ordered record.

Definition 2.2 (Procedural trace)**.** A procedural trace is a nonempty finite word over $\mathcal{L}_P$, written as Equation 4.

$$\mathbf{p} = (p_1, p_2, \ldots, p_n), \qquad p_k \in \mathcal{L}_P, \qquad n \geq 1. \tag{4}$$

The trace records the order in which operation types occur. It does not, by itself, determine the relational configurations through which the procedure passes.

The distinction in Definition 2.2 is important for the derivation developed later. A sequence of labels such as (R, V, X, S) tells the reader that recognition precedes valuation, transfer, and settlement. It does not tell the reader what recognition changes, whether valuation is admissible before recognition, or whether settlement removes a later avenue of revision. Those questions require an operational semantics, supplied in Section 3 by partially defined transformations on a relational configuration space.

Two traces may contain the same operation types while differing only in their order. Equation 5 gives the simplest representation of this relation.

$$\mathbf{p} = (p_1, \ldots, p_i, p_j, \ldots, p_n), \qquad \mathbf{p}' = (p_1, \ldots, p_j, p_i, \ldots, p_n). \tag{5}$$

Equation 5 states only that $\mathbf{p}$ and $\mathbf{p}'$ are related by an adjacent permutation. Whether the two orderings generate the same realized conversion history is a separate question. The distinction between permutation of labels and equality of realized histories prevents a logical leap from combinatorial order to procedural equivalence.

Remark 2.3 (Label order and realized history)**.** A procedural trace is a syntactic object. A realized conversion history requires a relational configuration, a realization of each operation label as a state-dependent transformation, and satisfaction of the domain conditions required for sequential composition. Section 3 introduces those additional objects.

Remark 2.3 also clarifies why compositional noncommutativity is not assumed at the level of $\mathcal{L}_P$. A word over an alphabet has an order, but strict noncommutativity becomes analytically meaningful only after operation labels are represented by transformations whose mutually executable compositions can be compared.

2.2 Recognition, Classification, Valuation, and Settlement

The operation types in Definition 2.1 play different procedural roles. Table 1 states the working interpretation used in the paper. These interpretations are deliberately broad enough to cover legal, economic, administrative, and organizational conversion processes while remaining specific enough to support later formalization.

Table 1 does not impose a canonical sequence. Some procedures classify before explicit recognition, some combine recognition and classification in one administrative act, and some valuation procedures

Table 1: Working roles of conversion-relevant procedural operations.

Label	Operation type	Procedural role	Typical state-sensitive feature
R	Recognition	Records or establishes that an entity, claim, contribution, or relation has standing or institutional visibility within the modeled procedure.	Recognition can create standing, admissible claims, or evidentiary relevance for later operations.
K	Classification	Assigns an entity, claim, contribution, or relation to a category used by the modeled procedure.	Classification can alter applicable rules, comparators, and evidentiary burdens.
A	Attribution	Records or assigns contribution, responsibility, entitlement, or authorship to one or more identified entities within the modeled procedure.	Attribution can change who may receive, contest, or redirect converted value.
V	Valuation	Produces or records a valuation, ranking, interval, or other evaluative representation used by the modeled conversion procedure.	The valuation can depend on earlier recognition, classification, evidence, and institutional context.
U	Authorization	Records or establishes that a contemplated operation is procedurally permitted or institutionally effective within the modeled setting.	Authorization can create or remove the domain in which later operations are available.
N	Negotiation	Provides a modeled channel through which relevant parties can propose, contest, or revise conversion terms.	Bargaining positions and available claims can depend on prior recognition and information.
X	Conversion / transfer	Represents the conversion-relevant transfer, substitution, compensation, or transformation selected for analysis.	Conversion can change possession, rights, obligations, control, or later evidentiary position.
S	Settlement	Records or produces a modeled condition under which the conversion is treated as provisionally or finally effective within the institution.	Settlement can close claims, discharge obligations, or raise the cost of later revision.
Q	Contestation / revision	Provides a modeled route through which a previous procedural result can be challenged, reconsidered, repaired, or modified.	Revision may depend on whether settlement is reversible and whether standing or evidence remains available.

occur repeatedly as new evidence enters. The table supplies operation types whose possible ordering can be studied. It therefore functions as a vocabulary rather than a universal procedural template.

Recognition deserves particular attention because later operations can presuppose a recognized entity or claim. If a worker's contribution is institutionally invisible at the time of negotiation, a later valuation may be performed over a different object than the valuation that would have followed prior recognition. Similar dependence can arise when environmental effects become legally cognizable only after a transfer, or when a historical contribution is acknowledged after rights have already been settled. The formal point is that recognition can modify the conditions under which later operations are defined.

Classification is analytically distinct from recognition. Within the working vocabulary, recognition concerns whether something enters the modeled procedure, whereas classification concerns the category under which it enters. A recognized activity classified as employment may be governed by different comparison rules, rights, and settlement procedures than the same activity classified as an independent service. The difference can therefore propagate into valuation and conversion even when the underlying material activity remains similar.

Valuation is likewise broader than the assignment of a single scalar price. A valuation operation may produce a monetary amount, a range, a rank, a categorical status, a compensation schedule, or another representation that supports conversion. The note therefore uses V for the procedural act of valuation and reserves substantive questions about the adequacy of a particular metric for broader theories of conversion justice.

Settlement marks another important boundary in the working vocabulary. A transfer can occur before every issue has been closed, whereas a settlement operation records or changes the institutional status of the transfer within the relevant procedure. The distinction is central to retrospective recognition. A late recognition event that occurs before settlement can enter an open procedure; the same recognition after settlement may require an additional revision mechanism. Section 6 develops this difference in relation to reversibility and repair.

These operations can also be grouped by the procedural function they primarily serve. Equation 6 defines a working partition used only for exposition.

$$\mathcal{L}_P = \mathcal{L}_{\text{id}} \cup \mathcal{L}_{\text{eval}} \cup \mathcal{L}_{\text{act}} \cup \mathcal{L}_{\text{rev}}, \tag{6}$$

where $\mathcal{L}_{\text{id}} = \{R, K, A\}$ concerns procedural identification and attribution, $\mathcal{L}_{\text{eval}} = \{V, U, N\}$ concerns evaluation and authorization of terms, $\mathcal{L}_{\text{act}} = \{X, S\}$ concerns conversion and settlement, and $\mathcal{L}_{\text{rev}} = \{Q\}$ concerns contestation and revision. Equation 6 is a bookkeeping device rather than an ontological claim. A practical operation can realize more than one function.

2.3 Procedural Scope of the Present Note

The present note studies a restricted question: how ordering among conversion-relevant operations can alter the procedural history that becomes available for normative evaluation. This scope requires three distinctions.

First, procedural structure is distinct from substantive conversion value. Two procedures may implement the same visible conversion ratio while differing in recognition, information, participation, settlement, or revision. Conversely, two procedures may share the same ordering and produce different substantive terms. Order sensitivity therefore supplies one axis of procedural analysis rather than a complete account of conversion justice.

Second, structural divergence is distinct from a justice verdict. If two orderings generate different histories, a procedural theory must still state why that difference matters. One theory may emphasize equal standing and meaningful participation, another may emphasize informed consent, another may emphasize reviewability or protection against irreversible disadvantage. The formalism developed here provides a common history on which such criteria can operate without selecting one criterion in advance.

Third, procedural justice in this paper is narrower than the full set of justice questions raised by heterogeneous conversion. Ontological identification, epistemic injustice, distributive incidence, domination, commensurability, environmental standing, and conversion nomos can all affect whether a conversion is just. They enter the present analysis only when they change the ordering, availability, or consequences of procedural operations.

Definition 2.4 states a preliminary criterion for treating an ordering difference as procedurally relevant. Section 7 later refines the mutually realizable case through an explicit interpretive frame.

Definition 2.4 (Ordering relevance)**.** Let two conversion procedures differ only in the ordering of one or more corresponding operation types. The ordering difference is *procedurally relevant* within the present note when either (i) the change of order changes whether a corresponding trace is realizable in a way that a specified procedural theory treats as evaluatively salient, or (ii) both traces are realizable and their realized histories differ in at least one theory-salient feature, including standing, admitted information, participation, authorization, settlement status, reviewability, or feasible repair.

Definition 2.4 intentionally contains a theory-relative clause. A structural difference becomes a justice-relevant difference only through an evaluative theory that identifies the affected feature as normatively significant. This preserves the central separation between structural order sensitivity and procedural justice.

The paper therefore follows the analytical sequence summarized in Table 2. Section 3 supplies the compositional mathematics needed for the second and third rows, while Sections 7 and 8 address the final evaluative step.

The separation in Table 2 prevents three recurring logical shortcuts. A difference in labels does not by itself establish a difference in realized histories. A difference in realized histories does not by itself establish procedural injustice. A procedural injustice, where established by a particular theory, does not by itself exhaust the broader justice of the heterogeneous conversion. The subsequent sections preserve these distinctions while adding the formal structure needed to analyze them rigorously.

Table 2: Analytical separation between procedural description and normative evaluation.

Analytical layer	Main question	Output used by the next layer
Procedural vocabulary	Which operation types occur and in what recorded order?	Procedural trace **p**.
Operational realization	Which state-dependent transformations realize those operation types?	A composable or partially composable sequence of transformations.
Ordering comparison	Do alternative orders generate the same history, different histories, or one-sided accessibility?	A structural divergence between realized histories or an accessibility asymmetry when one candidate history is unrealizable.
Normative interpretation	Which divergent features matter under a specified procedural theory?	A theory-relative procedural evaluation.

3 Composition and Order Sensitivity

This section supplies the mathematical structure required to turn the procedural vocabulary of Section 2 into realized conversion histories. Its role is deliberately limited: it defines partial transformations, composition, order sensitivity, partial composability, and a state-dependent realization of procedural operations. The objective is to make later claims about noncommutative ordering type-consistent and derivable from explicit assumptions. The construction remains set-theoretic and compositional. No linear structure, inner product, commutator, or quantum-mechanical interpretation is assumed.

3.1 Partial Maps and Sequential Transformation

The formalism begins from a configuration space containing the relational states on which conversion-relevant operations can act. For the present discussion, Assumption 3.1 treats this space as a set. This choice avoids unnecessary foundational complications while leaving class-valued or categorical generalizations open.

Assumption 3.1 (Configuration space)**.** Let Ω be a nonempty set of relational configurations. At the chosen level of resolution, each $\omega \in \Omega$ is taken to contain the information on which the modeled operation schemas and domain conditions depend. This state-sufficiency condition is representational: if two real situations differ in a history or relation that affects later procedure, the model must distinguish them in Ω or else that dependence lies outside the formalism.

A procedural operation need not be available at every configuration. Recognition may require a submitted claim, settlement may require a valid transfer, and revision may require standing or a review mechanism. Partial maps therefore provide the basic transformation type.

Definition 3.2 (Partial transformation)**.** A *partial transformation* on Ω is a function $\Phi : D_\Phi \to \Omega$ whose domain D_Φ is a subset of Ω. The notation $\Phi : \Omega \rightharpoonup \Omega$ records that the map may be undefined outside D_Φ.

Definition 3.2 makes the domain part of the operation's procedural meaning. Two operations with the same output formula on overlapping states can still represent different procedural structures when their admissible domains differ. The domain therefore carries information about standing, authorization, preconditions, and institutional availability.

Sequential procedure is represented by ordinary composition of partial maps, with the domain of the composite determined by both operations. Equation 7 defines the domain of the composite $\Psi \circ \Phi$.

$$D_{\Psi\circ\Phi} = \{\omega \in D_\Phi \mid \Phi(\omega) \in D_\Psi\}\,. \tag{7}$$

For every $\omega \in D_{\Psi\circ\Phi}$, the composite is defined by Equation 8.

$$(\Psi \circ \Phi)(\omega) = \Psi(\Phi(\omega))\,. \tag{8}$$

Equations 7 and 8 separate two questions that can be collapsed in informal procedural reasoning: whether a sequence is available and, conditional on its availability, which configuration it produces.

The present paper models finite sequential traces. If the wider event structure is only partially ordered, a trace should be understood as a selected chain or linearization of the operations whose relative order is under comparison. Concurrent or incomparable events require an extension beyond the sequential composition used here; Section 9.4 returns to this limitation.

A procedural trace from Definition 2.2 becomes a realized history only after each label is associated with a partial transformation and an admissible initial configuration is supplied. The realized history retains both the operation-type labels and the configurations they generate, because different operation-type traces can in principle produce the same state path. Definition 3.3 states this lift explicitly.

Definition 3.3 (Operational realization of a procedural trace)**.** Let $\mathbf{p} = (p_1, \ldots, p_n)$ be a procedural trace with $p_k \in \mathcal{L}_P$. An *operational realization* of $\mathbf{p}$ is a sequence of partial transformations $(\Phi_{p_1}, \ldots, \Phi_{p_n})$ together with an initial configuration $\omega_0 \in \Omega$ for which the recursive sequence in Equation 9 is defined.

$$\omega_k = \Phi_{p_k}(\omega_{k-1}), \qquad k = 1, \ldots, n. \tag{9}$$

The resulting labeled object $H_{\mathbf{p}}(\omega_0) = (\mathbf{p}; \omega_0, \omega_1, \ldots, \omega_n)$ is the *realized procedural history* generated by that operational realization. Its state path is $(\omega_0, \ldots, \omega_n)$, while $\mathbf{p}$ records the operation-type trace.

Definition 3.3 makes the dependence on intermediate states explicit. An operation that is admissible at ω_0 need not remain admissible at ω_1, and an operation unavailable at ω_0 can become available after an earlier transformation.

The label granularity is itself a modeling choice. If two institutionally distinct implementations share the same broad operation type, generate the same configuration transition, and nevertheless matter procedurally, the object $H_{\mathbf{p}}(\omega_0)$ will not distinguish them unless the operation labels or history features are refined. The present paper therefore treats $\mathcal{L}_P$ as a working alphabet rather than a claim that its labels exhaust every procedurally relevant implementation difference.

The composite transformation associated with an operational realization is stated in Equation 10.

$$\Gamma_{\mathbf{p}} = \Phi_{p_n} \circ \cdots \circ \Phi_{p_2} \circ \Phi_{p_1}. \tag{10}$$

Proposition 3.4 records the equivalence between recursive realizability and membership in the domain of the composite.

Proposition 3.4 (History-composite definedness)**.** *For the operational realization in Definition 3.3, the realized history $H_{\mathbf{p}}(\omega_0)$ exists if and only if $\omega_0 \in D_{\Gamma_{\mathbf{p}}}$. When it exists, its terminal configuration satisfies $\omega_n = \Gamma_{\mathbf{p}}(\omega_0)$.*

Proof. The claim follows by induction on the length of the trace. For $n = 1$, realizability is exactly the condition $\omega_0 \in D_{\Phi_{p_1}}$. Assume the statement holds for a trace of length $n - 1$. The length-n history exists precisely when the first $n - 1$ operations generate ω_{n-1} and $\omega_{n-1} \in D_{\Phi_{p_n}}$. By the induction hypothesis and Equation 7, this condition is equivalent to $\omega_0 \in D_{\Phi_{p_n} \circ \cdots \circ \Phi_{p_1}}$. Equation 8 then gives the terminal equality. $\square$

Proposition 3.4 is modest but important. It ensures that later comparisons between alternative orderings concern well-defined procedural histories rather than symbolic permutations of labels.

3.2 Compositional Noncommutativity

Order sensitivity becomes mathematically substantive when two operation orders are both available but produce different results. The present note uses *compositional noncommutativity* for this property. The term refers to composition of partial transformations and does not presuppose a linear operator algebra.

Definition 3.5 (Local commutativity)**.** Let $\Phi_i, \Phi_j : \Omega \rightharpoonup \Omega$, and let $\omega \in D_{\Phi_j \circ \Phi_i} \cap D_{\Phi_i \circ \Phi_j}$. The pair (Φ_i, Φ_j) is *locally commutative at* ω when Equation 11 holds.

$$(\Phi_j \circ \Phi_i)(\omega) = (\Phi_i \circ \Phi_j)(\omega). \tag{11}$$

It is *locally noncommutative at* ω when both compositions are defined and their outputs are unequal.

The state-specific definition is preferable to a global equality requirement because procedural operations are partial. Two maps may commute on one subset of configurations and diverge on another. A recognition step may be procedurally inert when all relevant claims are already acknowledged, yet materially alter later conversion when recognition changes standing, evidence, or available terms.

Definition 3.6 extends the local notion to the domain where both orderings can be compared.

Definition 3.6 (Terminally order-sensitive pair)**.** Let $M_{ij} = D_{\Phi_j \circ \Phi_i} \cap D_{\Phi_i \circ \Phi_j}$ denote the mutual-composability domain. The pair (Φ_i, Φ_j) is *terminally order-sensitive* when there exists at least one $\omega \in M_{ij}$ at which the pair is locally noncommutative.

This definition yields a direct derivation of the introductory condition in Equation 1. For any $\omega \in M_{ij}$, the two terminal configurations are defined by Equation 12.

$$\omega_{ij} = \Phi_j(\Phi_i(\omega)), \qquad \omega_{ji} = \Phi_i(\Phi_j(\omega)). \tag{12}$$

Terminal order sensitivity at ω is therefore the condition $\omega_{ij} \neq \omega_{ji}$. The inequality is a statement about terminal relational configurations. It does not yet specify which difference between them is procedurally salient or normatively preferable.

The intermediate states can matter even when the terminal states coincide. Definition 3.7 therefore separates terminal equality from history equality.

Definition 3.7 (History-sensitive comparison)**.** For two realized two-step histories from the same initial configuration ω, let $H_{ij}(\omega) = ((i,j); \omega, \Phi_i(\omega), \omega_{ij})$ and $H_{ji}(\omega) = ((j,i); \omega, \Phi_j(\omega), \omega_{ji})$. The histories are *terminally equivalent* when $\omega_{ij} = \omega_{ji}$. They are *configuration-path equivalent* when both their intermediate and terminal configurations agree, as stated in Equation 13.

$$\Phi_i(\omega) = \Phi_j(\omega) \quad \text{and} \quad \omega_{ij} = \omega_{ji}. \tag{13}$$

When terminal equivalence holds while Equation 13 fails, the two orderings reach the same terminal configuration through different intermediate configurations.

Definition 3.7 keeps the structural comparison independent of a normative theory. A procedural theory can later apply a coarser or richer equivalence relation to the realized histories, but that additional evaluative structure is not needed to establish whether the modeled configuration paths differ. A recognition event that occurs before negotiation can therefore remain procedurally relevant even when later compensation restores the same monetary allocation.

Figure 1 visualizes the two composable histories. The diagram is purely structural: it records the two routes through configuration space and does not assign a justice ranking to either route.

The algebra-inspired terminology becomes stronger only after additional algebraic structure is supplied. If a later representation embeds suitable transformations into a linear operator space, then an algebraic commutator can be defined. The present note does not require that step. Composition and definedness already capture the procedural phenomenon under study.

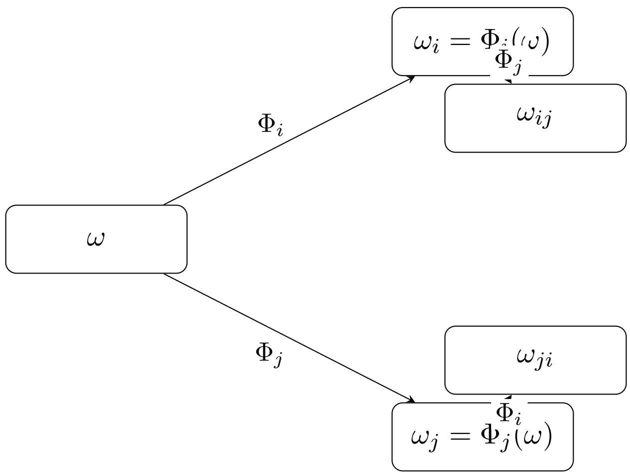

Figure 1: Two alternative orderings from a common initial configuration. Both routes are shown only when the corresponding compositions are defined. Strict compositional noncommutativity occurs when the terminal configurations differ; configuration-path divergence can also remain when the terminal configurations coincide.

3.3 Partial Composability

Noncommutativity compares two orderings that are both defined. Procedural systems can also exhibit a different asymmetry: one ordering can be performed while the reverse ordering is unavailable. Recognition can create standing required for negotiation; authorization can make settlement available; settlement can close a revision route. The domain structure of partial maps represents these cases directly.

Definition 3.8 (One-way composability)**.** Let $\Phi_i, \Phi_j : \Omega \rightharpoonup \Omega$. The ordering $i \to j$ is *one-way composable relative to $j \to i$ at ω* when Equation 14 holds.

$$\omega \in D_{\Phi_j \circ \Phi_i}, \qquad \omega \notin D_{\Phi_i \circ \Phi_j}. \tag{14}$$

Definition 3.8 and Equation 14 describe an accessibility asymmetry rather than an output inequality because the reverse history has no realized terminal configuration in the model. The asymmetry concerns procedural accessibility rather than comparative outcome alone.

The four elementary possibilities for a pair of operations at an initial configuration are summarized in Table 3. The table distinguishes mutual composability, one-way composability, and mutual unavailability before any normative interpretation is added.

A useful derived relation records one-way accessibility. Equation 15 defines a local precedence-dependency symbol for the present note.

$$\Phi_i \prec_\omega^{\text{dep}} \Phi_j \quad \Longleftrightarrow \quad \omega \in D_{\Phi_j \circ \Phi_i} \text{ and } \omega \notin D_{\Phi_i \circ \Phi_j}. \tag{15}$$

The symbol $\prec_\omega^{\text{dep}}$ is deliberately local. It does not define a global partial order over procedural operations. In particular, transitivity and antisymmetry are not assumed. A state change produced by a third operation can alter the availability relation between the first two operations. The symbol records only that, at the specified initial configuration, executing Φ_i before Φ_j is procedurally accessible while the reverse two-step sequence is not.

Partial composability therefore adds a second structural question to procedural analysis. Beyond asking whether alternative orders generate different results, the analysis must ask whether the institution makes both alternatives available in the first place. This distinction becomes important in later sections when standing, authorization, settlement finality, and access to revision are treated as procedural features.

Table 3: Pairwise composability cases for two procedural operations at an initial configuration ω.

Case	$\Phi_j \circ \Phi_i$	$\Phi_i \circ \Phi_j$	Structural interpretation
Mutual, terminally equal	Defined	Defined	Both orders are available and have equal terminal configurations at ω; their intermediate paths can still differ.
Mutual, terminally divergent	Defined	Defined	Both orders are available and produce different terminal configurations.
One-way	Defined	Undefined	The first operation creates or preserves conditions required for the second, while the reverse sequence is unavailable.
Mutual unavailability	Undefined	Undefined	Neither two-step order is available from the current configuration.

3.4 State-Dependent Operations

The partial-map model already allows an operation's output and availability to depend on the current configuration. Some procedures require an additional distinction: the concrete transformation implementing an operation type can itself be selected or parameterized by the configuration in which the operation occurs. Valuation rules can depend on classification, settlement rules can depend on the rights created by transfer, and recognition procedures can depend on the institutional status produced by earlier events. The present subsection formalizes this stronger form of state dependence while preserving type consistency.

Let $\mathrm{PEnd}(\Omega)$ denote the set of partial endomorphisms of Ω, meaning all partial transformations $\Phi : \Omega \rightharpoonup \Omega$. Definition 3.9 distinguishes an operation type from its state-instantiated transformation.

Definition 3.9 (State-dependent operation schema)**.** For an operation type $p \in \mathcal{L}_P$, a *state-dependent operation schema* is a map $\mathfrak{F}_p : \Omega \to \mathrm{PEnd}(\Omega)$. For a configuration ω, the selected partial transformation is denoted by Equation 16.

$$\Phi_p^{[\omega]} = \mathfrak{F}_p(\omega). \tag{16}$$

The operation is executable at ω only when $\omega \in D_{\Phi_p^{[\omega]}}$.

Definition 3.9 allows the current configuration to determine which concrete transformation realizes the procedural label. The actual state update can therefore be collected into a single induced partial map. Definition 3.10 introduces this map.

Definition 3.10 (Induced state update)**.** Given a state-dependent operation schema $\mathfrak{F}_p$, define the induced partial update $U_p : \Omega \rightharpoonup \Omega$ by Equation 17.

$$U_p(\omega) = \Phi_p^{[\omega]}(\omega), \qquad D_{U_p} = \left\{ \omega \in \Omega \;\middle|\; \omega \in D_{\Phi_p^{[\omega]}} \right\}. \tag{17}$$

The induced update restores an ordinary partial-map type while retaining state-dependent semantics internally. When a procedural trace is realized through state-dependent schemas, the later sections use the induced updates U_{p_k} as the partial transformations in the operational-realization construction of Definition 3.3; the state-instantiated transformations $\Phi_{p_k}^{[\omega]}$ remain the internal selections that define those induced maps. Sequential state-dependent execution can therefore be represented rigorously by composition of the induced updates. For two operation types i and j, Equation 18 expands the composition at an initial configuration ω_0.

$$(U_j \circ U_i)(\omega_0) = \Phi_j^{[\omega_i]}(\omega_i)\,, \qquad \omega_i = \Phi_i^{[\omega_0]}(\omega_0)\,. \tag{18}$$

Equation 18 gives the precise form of the schematic notation used in Equation 2. The later transformation is selected from the intermediate configuration generated by the earlier operation. Reversing the operation order can select a different pair of state-instantiated transformations, even before the terminal configurations are compared.

For recognition R and conversion X, the two expanded outcomes are stated in Equation 19.

$$\begin{aligned} \omega_{RX} &= \Phi_X^{[\omega_R]}(\omega_R), \qquad & \omega_R &= \Phi_R^{[\omega_0]}(\omega_0), \\ \omega_{XR} &= \Phi_R^{[\omega_X]}(\omega_X), \qquad & \omega_X &= \Phi_X^{[\omega_0]}(\omega_0). \end{aligned} \tag{19}$$

Equation 19 makes the central mechanism explicit. Recognition can change the configuration from which conversion is instantiated; conversion can change the configuration from which recognition is instantiated. The two later operations therefore need not be the same concrete transformation with their order merely exchanged.

Remark 3.11 (Relation to operator-algebra language)**.** Remark 3.11 records the algebraic scope of the construction. The objects U_p in Definition 3.10 are partial transformations under composition. Their order can be noncommutative in the compositional sense of Definition 3.5. The present construction does not supply vector-space addition, scalar multiplication, adjoints, norms, or a multiplication operator beyond composition. An algebraic commutator therefore remains an optional later representation rather than part of the current derivation.

The formal hierarchy developed in this section is summarized in Table 4. Each level has a distinct type and analytical role, which prevents procedural labels, transformation schemas, realized histories, and normative evaluations from being conflated.

Table 4 completes the formal bridge promised in Table 2. Section 4 can now specify the reduced Generative Relational content of the configurations in Ω, while Sections 5 and 6 can analyze concrete procedural orderings without introducing undefined transformation types or assuming that structural divergence already constitutes injustice.

Table 4: Formal hierarchy used for order-sensitive conversion procedures.

Object	Type	Analytical role
Operation label p	Element of $\mathcal{L}_P$	Names a procedural function such as recognition, valuation, or settlement.
Operation schema $\mathfrak{F}_p$	$\Omega \to \mathrm{PEnd}(\Omega)$	Selects a concrete partial transformation relative to the current configuration.
Instantiated transformation $\Phi_p^{[\omega]}$	Partial endomorphism of Ω	Represents the concrete realization selected at configuration ω.
Induced update U_p	$\Omega \rightharpoonup \Omega$	Executes the state-dependent operation while retaining an ordinary partial-map type.
Procedural trace $\mathbf{p}$	Finite sequence in $\mathcal{L}_P$	Records syntactic operation order.
Realized history $H_{\mathbf{p}}(\omega_0)$	Labeled finite path over $\mathcal{L}_P$ and Ω	Records both the operation-type trace and the configurations generated by an admissible realization.
Composite history map $\Gamma_{\mathbf{p}}$	$\Omega \rightharpoonup \Omega$	Maps admissible initial configurations to terminal configurations for a fixed operational realization.

4 Relational Representation of Conversion Procedure

The formal machinery in Section 3 treats a conversion procedure as a sequence of partially defined transformations on an abstract configuration space Ω. The present section supplies the reduced relational semantics used for that abstraction. Its role is deliberately limited: it specifies what information a configuration must retain for recognition, conversion, settlement, and revision to become state-dependent operations, while leaving the richer ontology of heterogeneous value conversion to the companion formalism paper. The construction separates epistemic readout from state transformation and distinguishes transfer from settlement so that later ordering claims can be stated without type ambiguity.

4.1 Relational Configurations

A procedural state must retain enough relational information to determine which operations are available and how their concrete realizations are selected. For the present note, that information is grouped into three structured components: accumulated relational history, the currently active relational boundary, and the effective conversion nomos. This grouping is a representational convention rather than a proof that three components are sufficient for every application. Definition 4.1 introduces the reduced working representation.

Definition 4.1 (Reduced relational configuration)**.** A *reduced relational configuration* is an element $\omega \in \Omega$ represented by the ordered triple in Equation 20.

$$\omega = \big(\mathcal{H}_\omega, \mathcal{B}_\omega, \mathcal{N}_\omega\big). \tag{20}$$

Here $\mathcal{H}_\omega$ denotes the accumulated relational history represented in the model, $\mathcal{B}_\omega$ denotes the relational structures treated as active at the present procedural boundary, and $\mathcal{N}_\omega$ denotes the effective nomos that conditions admissibility, recognition, settlement, and revision in that configuration.

Equation 20 is a working state representation rather than a claim that social reality has exactly three ontological components. Each entry can itself be a structured object, and an application must enrich or refine the state whenever omitted distinctions affect later domains or operation schemas. The three entries serve distinct procedural functions. $\mathcal{H}_\omega$ records the historical structures from which a later operation may draw evidence or standing; $\mathcal{B}_\omega$ records the relations that remain presently operative; and $\mathcal{N}_\omega$ records the institutional-normative conditions under which an operation can count as effective. Additional application-specific variables may be incorporated into ω when required.

The distinction between history and active boundary matters for procedure. A settled contract can remain part of $\mathcal{H}_\omega$ after its ordinary obligations cease to be active in $\mathcal{B}_\omega$. A previously unrecognized contribution can remain represented in historical materials even when it lacks current standing under $\mathcal{N}_\omega$. Later recognition can therefore alter the active and normative components without requiring the model to erase the earlier procedural history.

For any component of Equation 20, an ordinary readout can be represented by a projection-like map. Equation 21 defines the three elementary readouts used below.

$$\mathrm{hist}(\omega) = \mathcal{H}_\omega, \qquad \mathrm{act}(\omega) = \mathcal{B}_\omega, \qquad \mathrm{nom}(\omega) = \mathcal{N}_\omega. \tag{21}$$

Equation 21 introduces readout maps only. They do not themselves transform ω. This distinction becomes important in Subsection 4.2, where epistemic observation and performative recognition are assigned different types.

Table 5 summarizes the semantic content assigned to Ω in the present note. The representation remains intentionally coarse. Its purpose is to make state-dependent procedural ordering intelligible without reproducing the full higher-order event formalism of the broader conversion framework.

4.2 Observation and Recognition

Procedural analysis requires a sharp distinction between obtaining information about a configuration and changing the configuration by recognizing an entity, claim, status, or contribution. An observation can produce a readout while leaving the modeled procedural state unchanged. Recognition can use such a readout and then alter standing, classification access, admissible claims, or other active relations. Treating both operations as the same map would obscure the mechanism through which recognition changes later conversion possibilities.

Let X_c denote an observation space associated with an observational context c. Definition 4.2 introduces a typed observation map.

Definition 4.2 (Procedural observation map)**.** A *procedural observation map* in context c is a map of the form stated in Equation 22.

$$\pi^{(c)} : \Omega \longrightarrow X_c. \tag{22}$$

Its output $x = \pi^{(c)}(\omega)$ is an epistemic or institutional readout of ω; the map does not, by definition, specify a new configuration in Ω.

Recognition is represented separately through the induced update U_R defined by the recognition schema $\mathfrak{F}_R$ in Definition 3.10. When recognition is executable, Equation 23 defines the recognized configuration.

$$\omega_R = U_R(\omega) = \Phi_R^{[\omega]}(\omega). \tag{23}$$

Equation 23 allows recognition to modify $\mathcal{B}_\omega$, $\mathcal{N}_\omega$, or both, and it also permits a recognition event to be recorded in $\mathcal{H}_{\omega_R}$. The precise update depends on the application. A legal recognition can create standing; an attributional recognition can make a contribution institutionally visible; a classification decision can open a valuation procedure that was previously unavailable.

Where a single institutional act both returns a readout and changes the configuration, the two effects can be packaged without conflating their types. Definition 4.3 introduces a recognition instrument.

Definition 4.3 (Recognition instrument)**.** Given an observation map $\pi_R^{(c)} : \Omega \to X_R^{(c)}$ and a recognition update $U_R : \Omega \rightharpoonup \Omega$, define the partial recognition instrument $\mathcal{M}_R^{(c)}$ by Equation 24.

$$\mathcal{M}_R^{(c)} : \Omega \rightharpoonup X_R^{(c)} \times \Omega, \qquad \mathcal{M}_R^{(c)}(\omega) = \big(\pi_R^{(c)}(\omega), U_R(\omega)\big). \tag{24}$$

The domain of $\mathcal{M}_R^{(c)}$ is the domain of U_R.

Table 5: Reduced relational components used in the procedural representation.

Component	Working interpretation	Procedural role
$\mathcal{H}_\omega$	Accumulated relational history represented in the model	Supplies historical relations, records, prior operations, and potential evidentiary or attributional material.
$\mathcal{B}_\omega$	Presently active relational boundary	Records operative rights, obligations, statuses, holdings, dependencies, and other relations relevant to immediate procedure.
$\mathcal{N}_\omega$	Effective conversion nomos	Conditions admissibility, standing, institutional recognition, settlement, finality, and revision.

Definition 4.3 records the epistemic and performative outputs of one recognition procedure while preserving their different codomains. The product type $X_R^{(c)} \times \Omega$ is therefore intentional: $\mathcal{M}_R^{(c)}$ is an instrument, whereas U_R remains the state-changing operation that participates in procedural composition.

The separation also clarifies one source of state-dependent order sensitivity. Suppose recognition changes a component on which conversion depends. Then the conversion schema $\mathfrak{F}_X$ can select a different transformation at ω_R than at ω. Proposition 4.4 states this implication at the level required in this note.

Proposition 4.4 (Recognition-dependent conversion instantiation)**.** *Let $\omega \in D_{U_R}$, let $\omega_R = U_R(\omega)$, and let $\mathfrak{F}_X : \Omega \to \mathrm{PEnd}(\Omega)$ be the conversion schema. If Equation 25 holds, then conversion after recognition is realized by a different concrete partial transformation from conversion instantiated at the initial configuration.*

$$\mathfrak{F}_X(\omega_R) \neq \mathfrak{F}_X(\omega). \tag{25}$$

Equivalently, $\Phi_X^{[\omega_R]} \neq \Phi_X^{[\omega]}$.

Proof. By Definition 3.9, $\Phi_X^{[\omega_R]} = \mathfrak{F}_X(\omega_R)$ and $\Phi_X^{[\omega]} = \mathfrak{F}_X(\omega)$. Equation 25 therefore implies $\Phi_X^{[\omega_R]} \neq \Phi_X^{[\omega]}$. □

Proposition 4.4 is formally elementary, but it locates the mechanism that later sections evaluate procedurally. Recognition can affect conversion by changing the configuration from which conversion is instantiated, even before terminal outcomes are compared.

4.3 Conversion and Settlement

Conversion and settlement occupy different positions in the procedural architecture. The operation type X represents the transfer, exchange, or other conversion act through which rights, holdings, claims, or recognized value relations change. The operation type S represents institutional settlement: the process through which a conversion acquires a specified degree of completion, discharge, recording, or finality. Keeping the two operations separate allows the analysis to represent conversions that have occurred but remain unsettled, as well as settlements that close some avenues of later revision.

For a configuration $\omega \in D_{U_X}$, the conversion update is stated in Equation 26.

$$\omega_X = U_X(\omega) = \Phi_X^{[\omega]}(\omega). \tag{26}$$

The active boundary $\mathcal{B}_{\omega_X}$ can differ from $\mathcal{B}_\omega$ because conversion may alter possession, contractual position, access, claims, obligations, or another application-specific relation. The historical component can also record the conversion event. Equation 26 does not require settlement to occur simultaneously.

For a converted configuration $\omega_X \in D_{U_S}$, settlement produces the configuration in Equation 27.

$$\omega_{XS} = U_S(\omega_X) = \Phi_S^{[\omega_X]}(\omega_X). \tag{27}$$

Settlement can alter the effective nomos, the active status of obligations, and the accessibility of later review. To represent this effect without assigning a substantive theory of finality, let $\mathcal{Q}(\omega)$ denote the set of contestation or revision procedures institutionally available at configuration ω. Definition 4.5 introduces this set-valued readout.

Definition 4.5 (Review-access set)**.** For a relational configuration ω, let $\mathcal{Q}(\omega)$ denote the set of revision or contestation operations institutionally available from ω. The readout is represented by Equation 28.

$$\mathcal{Q} : \Omega \longrightarrow 2^{\mathcal{L}_Q}, \tag{28}$$

where $\mathcal{L}_Q$ is an application-specific set of concrete revision-procedure identifiers and $2^{\mathcal{L}_Q}$ is its power set. These identifiers refine the broad operation type Q in $\mathcal{L}_P$; they are used to distinguish multiple institutional review routes rather than to enlarge the procedural type alphabet.

Definition 4.5 does not assume that settlement always decreases review access. Some settlement regimes create appeal or review rights at the same moment that they close ordinary negotiation. The definition only makes the availability structure explicit so that a later section can compare $\mathcal{Q}(\omega_X)$ with $\mathcal{Q}(\omega_{XS})$.

A similar distinction applies to visible conversion output. An observer may record a coarse transfer through a readout $\pi_X^{(c)}(\omega_X)$, while the state ω_X retains a broader set of relational consequences. The procedural analysis therefore treats the observed conversion outcome and the relational configuration generated by conversion as analytically distinct objects.

4.4 Contestation and Revision

Contestation and revision are grouped under the operation label Q in Definition 2.1. Their procedural role is to make an earlier classification, valuation, authorization, conversion, or settlement available for challenge, reconsideration, repair, or replacement. The present section models only accessibility and state transformation; Section 6 later distinguishes reversal, repair, and retrospective recognition in greater detail.

A revision update is an induced partial transformation $U_Q : \Omega \rightharpoonup \Omega$. Its execution at ω is stated in Equation 29.

$$\omega_Q = U_Q(\omega) = \Phi_Q^{[\omega]}(\omega), \qquad \omega \in D_{U_Q}. \tag{29}$$

The domain condition in Equation 29 is substantive for ordering analysis. A procedure can make review available only after recognition creates standing, or it can make review unavailable after settlement closes a claim. The general partial-map framework already accommodates both cases; the relational representation explains which components of ω can encode the relevant access conditions.

To connect review access with the executable update, Assumption 4.6 states a minimal consistency requirement for applications that explicitly use $\mathcal{Q}(\omega)$.

Assumption 4.6 (Review-access consistency)**.** Suppose a concrete revision procedure $q \in \mathcal{L}_Q$ is represented by a partial update $U_q : \Omega \rightharpoonup \Omega$. For configurations in the modeled institutional domain, Equation 30 holds.

$$q \in \mathcal{Q}(\omega) \quad \Longleftrightarrow \quad \omega \in D_{U_q}. \tag{30}$$

Assumption 4.6 is a bookkeeping condition connecting an institutional-access readout to the domain of its corresponding partial transformation. It can be relaxed in applications that distinguish nominal rights from practically executable procedures. The assumption is sufficient for the derivations in Sections 5 and 6.

The procedural consequences of the four operations emphasized in this section are summarized in Table 6. The table distinguishes readout, relational transformation, and accessibility effects so that later order comparisons do not rely on an unspecified notion of "procedure."

Table 6: Relational semantics assigned to the principal operations used in later order comparisons.

Operation	Primary object	Possible configuration change	Ordering relevance
Observation $\pi^{(c)}$	Readout in X_c	None by definition	Supplies information used by a later state-changing procedure.
Recognition U_R	Standing, status, or institutional legibility	May change $\mathcal{B}_\omega$, $\mathcal{N}_\omega$, and recorded history	Can change which conversion, valuation, or review transformation is instantiated or admissible.
Conversion U_X	Transfer or transformation of recognized relations	May change holdings, claims, obligations, access, and historical record	Can alter the state from which recognition, settlement, or revision subsequently occurs.
Settlement U_S	Completion, discharge, recording, or finality	May change active obligations, nomos, and review access	Can preserve, create, narrow, or close later revision paths.
Revision U_Q	Reconsideration, repair, or replacement of prior procedural results	May reopen or modify active relations and later nomos	Its domain can depend on recognition, settlement status, and earlier procedural history.

Table 6 completes the reduced relational semantics used for the rest of the paper. The configuration space Ω now has enough internal structure to distinguish historical record, active relations, and effective nomos; observation is separated from performative recognition; conversion is separated from settlement; and revision has an explicit accessibility condition. Section 5 can therefore analyze concrete ordering divergences using the compositional machinery of Section 3 without introducing additional undefined state variables.

5 Ordering Divergence in Conversion

The compositional machinery in Section 3 distinguishes symbolic order, admissible composition, state-dependent operation schemas, and realized procedural histories. The relational semantics in Section 4 then specifies how recognition, conversion, settlement, and revision can modify the configuration on which later operations act. The present section combines those two layers. Its role is to derive several concrete forms of ordering divergence without attaching a justice verdict to them. The analysis proceeds through recognition–conversion order, classification–valuation order, conversion–revision order, and the distinction between visible endpoint similarity and historical equivalence.

5.1 Recognition and Conversion

Recognition and conversion provide the clearest case in which the first operation can alter both the admissibility and the concrete realization of the second. Let $U_R, U_X : \Omega \rightharpoonup \Omega$ denote the induced recognition and conversion updates defined through the state-dependent schemas of Definition 3.9. For an initial configuration ω_0, the two candidate two-step histories are defined in Equation 31 whenever the required domain conditions are satisfied.

$$\begin{aligned} H_{RX}(\omega_0) &= \big((R, X); \omega_0, U_R(\omega_0), (U_X \circ U_R)(\omega_0)\big), \\ H_{XR}(\omega_0) &= \big((X, R); \omega_0, U_X(\omega_0), (U_R \circ U_X)(\omega_0)\big). \end{aligned} \tag{31}$$

The configurations at which both histories can be compared form the mutual-composability domain in Equation 32.

$$M_{RX} = D_{U_X \circ U_R} \cap D_{U_R \circ U_X}. \tag{32}$$

Equation 32 separates mutual comparison from the distinct accessibility asymmetry introduced in Definition 3.8. Within M_{RX}, recognition and conversion are terminally order-sensitive at ω_0 when Equation 33 holds.

$$(U_X \circ U_R)(\omega_0) \neq (U_R \circ U_X)(\omega_0). \tag{33}$$

Terminal divergence can arise through several mechanisms. Recognition can alter standing, admissible evidence, applicable nomos, or the active relational boundary. Those changes can modify the conversion transformation selected by the conversion schema. Definition 5.1 isolates this mechanism from terminal divergence itself.

Definition 5.1 (Recognition-sensitive conversion schema)**.** Let $\omega_0 \in D_{U_R}$, and write $\omega_R = U_R(\omega_0)$. The conversion schema $\mathfrak{F}_X$ is *recognition-sensitive at* ω_0 when Equation 34 holds.

$$\mathfrak{F}_X(\omega_R) \neq \mathfrak{F}_X(\omega_0). \tag{34}$$

Definition 5.1 concerns operator selection. It does not by itself imply Equation 33: two distinct partial transformations can agree on the particular arguments at which they are executed. The distinction prevents a logical leap from schema difference to outcome difference.

Recognition can also change whether conversion is executable at all. Proposition 5.2 derives the corresponding one-way accessibility result.

Proposition 5.2 (Recognition-enabled conversion)**.** *Let $\omega_0 \in D_{U_R}$, let $\omega_R = U_R(\omega_0)$, and suppose the conditions in Equation 35 hold.*

$$\omega_0 \notin D_{U_X}, \qquad \omega_R \in D_{U_X}. \tag{35}$$

Then $\omega_0 \in D_{U_X \circ U_R}$ and $\omega_0 \notin D_{U_R \circ U_X}$. Hence $U_R \prec^{\mathrm{dep}}_{\omega_0} U_X$ in the sense of Equation 15.

Proof. Because $\omega_0 \in D_{U_R}$ and $U_R(\omega_0) = \omega_R \in D_{U_X}$, Equation 7 gives $\omega_0 \in D_{U_X \circ U_R}$. The condition $\omega_0 \notin D_{U_X}$ prevents the reverse composite from being initiated, so $\omega_0 \notin D_{U_R \circ U_X}$. Equation 15 then yields the stated dependency relation. □

Proposition 5.2 captures a structurally important procedural case: recognition does more than modify conversion terms; it creates the standing, status, or institutional legibility required for conversion to occur. The proposition remains descriptive. Whether such precedence is procedurally desirable depends on the evaluative theory introduced later in Section 7.

Figure 2 summarizes the distinction between mutual divergence and recognition-enabled one-way accessibility.

5.2 Classification and Valuation

Classification and valuation illustrate a different form of state dependence. Recognition determines whether an entity enters a procedure, while classification can determine the category under which valuation is performed. A valuation can therefore depend on a classification-generated configuration even when both operation orders are procedurally available.

Let $U_K, U_V : \Omega \rightharpoonup \Omega$ denote the induced classification and valuation updates. Because a valuation operation can produce a scalar, interval, ranking, category, or other representation, the present note does not identify valuation with a numerical function. Instead, let Y_V be an application-specific valuation-output space and let Equation 36 define a passive readout from configurations in which a valuation record is available.

$$\vartheta_V : \Omega_V \longrightarrow Y_V, \qquad \Omega_V \subseteq \Omega. \tag{36}$$

The subset Ω_V in Equation 36 contains configurations for which the modeled valuation result is institutionally legible. The readout does not create the valuation; that role belongs to the state-changing update U_V.

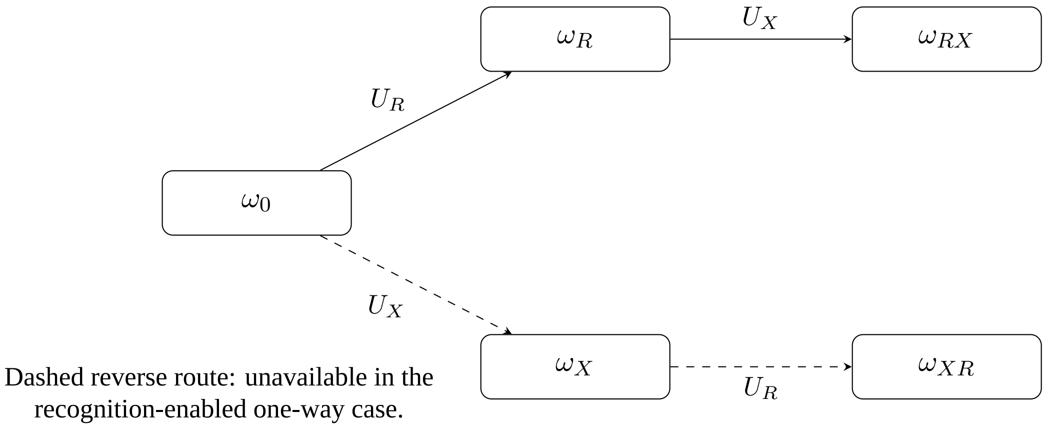

Figure 2: Recognition–conversion ordering. Mutual composability permits comparison of ω_{RX} and ω_{XR}. The dashed reverse route marks the recognition-enabled one-way case in which conversion cannot begin directly from ω_0; in mutually composable cases that route is executable and is compared with the recognition-first route.

For an initial configuration ω_0 on which both two-step orders are defined and whose terminal states lie in Ω_V, define the two valuation readouts in Equation 37.

$$\begin{aligned} y_{KV}(\omega_0) &= \vartheta_V((U_V \circ U_K)(\omega_0)) \, , \\ y_{VK}(\omega_0) &= \vartheta_V((U_K \circ U_V)(\omega_0)) \, . \end{aligned} \tag{37}$$

Definition 5.3 isolates order sensitivity at the level of the valuation representation.

Definition 5.3 (Classification-sensitive valuation)**.** Classification and valuation are *valuation-divergent at* ω_0 when both readouts in Equation 37 are defined and Equation 38 holds.

$$y_{KV}(\omega_0) \neq y_{VK}(\omega_0). \tag{38}$$

Definition 5.3 permits order sensitivity to be detected at the evaluative interface even when a later coarse representation suppresses the underlying state difference. The definition also avoids assuming that Y_V carries arithmetic structure.

A common mechanism is schema sensitivity. If $\omega_K = U_K(\omega_0)$, then classification can change the concrete valuation transformation according to Equation 39.

$$\mathfrak{F}_V(\omega_K) \neq \mathfrak{F}_V(\omega_0). \tag{39}$$

As in Definition 5.1, Equation 39 is a mechanism-level statement and does not guarantee Equation 38 for every initial configuration. It identifies a route through which category assignment can alter the valuation procedure applied later.

The same formal structure can represent the reverse influence. A valuation completed before classification can itself alter the active evidence, recorded status, or institutional incentives used by a later classification operation. Hence the relevant comparison concerns two histories, not an assumption that classification always causally dominates valuation.

5.3 Conversion and Revision

Conversion and revision introduce accessibility questions that extend beyond terminal outcome comparison. A revision process may be available before conversion, unavailable afterward, or newly created by the conversion or settlement regime. Section 4.4 introduced the review-access set $\mathcal{Q}(\omega)$. The present subsection uses that object to derive order-sensitive review access.

Fix a revision procedure label $q \in \mathcal{L}_Q$ with induced partial update U_q, and retain Assumption 4.6. Let U_X denote conversion. Definition 5.4 identifies a configuration at which conversion removes access to the selected revision procedure.

Definition 5.4 (Conversion-induced review loss)**.** Let $\omega_0 \in D_{U_X}$ and write $\omega_X = U_X(\omega_0)$. Conversion produces *review loss for q at ω_0* when Equation 40 holds.

$$q \in \mathcal{Q}(\omega_0), \qquad q \notin \mathcal{Q}(\omega_X). \tag{40}$$

Under review-access consistency, Equation 40 means that revision is executable at the initial state but unavailable after conversion. One further condition is required to conclude one-way composability, because revision performed first must still leave conversion executable. Proposition 5.5 states the complete result.

Proposition 5.5 (Revision-before-conversion accessibility)**.** *Suppose conversion produces review loss for q at ω_0, and suppose the revised configuration $\omega_q = U_q(\omega_0)$ satisfies $\omega_q \in D_{U_X}$. Then the concrete update sequence that applies U_q before U_X is executable from ω_0, while the reverse two-step sequence is not executable from ω_0.*

Proof. Equation 40 and Assumption 4.6 imply $\omega_0 \in D_{U_q}$ and $\omega_X \notin D_{U_q}$. The additional condition $\omega_q \in D_{U_X}$ gives $\omega_0 \in D_{U_X \circ U_q}$ by Equation 7. Since $\omega_X \notin D_{U_q}$, the same equation gives $\omega_0 \notin D_{U_q \circ U_X}$. Thus the concrete revision-first sequence is executable and the conversion-first sequence is not. This conclusion is a statement about the partial updates U_q and U_X; the broad type label Q in $\mathcal{L}_P$ remains the procedural-trace label for revision. $\square$

The reverse pattern is also possible. A conversion can create a review right that did not exist beforehand, for example by generating a transaction record, appellate status, or statutory review trigger. The partial-map framework therefore treats review loss and review creation symmetrically as changes in the domain of U_q. No general monotonicity of review access is assumed.

Settlement can strengthen this ordering effect. If settlement U_S follows conversion and changes the review-access set, the relevant comparison can involve concrete three-step update sequences corresponding at the type level to (X, S, Q), (X, Q, S), and (Q, X, S), with Q instantiated by the selected procedure q. The present subsection records the two-operation mechanism; Section 6 develops the additional distinction between reopening, repair, and irreversible relational change after settlement.

5.4 Endpoint Similarity and Historical Difference

Procedural comparison requires a final distinction between what an institution visibly records and the fuller relational history that produced that record. Two conversion procedures can appear equivalent under a coarse endpoint readout while differing in terminal configuration, intermediate configuration, or both. This subsection introduces a hierarchy of equivalence relations that makes those possibilities explicit.

Let $\mathscr{H}(\Omega)$ denote the set of finite labeled realized histories generated by the model. A history label sequence may use the broad operation types in $\mathcal{L}_P$ or a finer set of concrete update identifiers when the analysis needs to distinguish implementations beneath a shared operation type. Thus a generic history can be written $H = (\boldsymbol{\lambda}; \omega_0, \ldots, \omega_n)$, where $\boldsymbol{\lambda}$ records the labels retained at the chosen resolution; the histories $H_{\mathbf{p}}(\omega_0)$ from Definition 3.3 are the special case $\boldsymbol{\lambda} = \mathbf{p}$. Let $\mathrm{term}(H) = \omega_n$ denote the terminal configuration and let $\mathrm{states}(H) = (\omega_0, \ldots, \omega_n)$ denote the state path. Let $r_{\mathrm{out}} : \Omega \to Y_{\mathrm{out}}$ be a passive readout of the visible conversion result, where Y_{out} can contain prices, transferred quantities, settlement statuses, or other application-specific outputs.

Definition 5.6 (Observed endpoint equivalence)**.** Two realized histories $H, H' \in \mathscr{H}(\Omega)$ are *observationally endpoint-equivalent* under r_{out} when Equation 41 holds.

$$r_{\mathrm{out}}(\mathrm{term}(H)) = r_{\mathrm{out}}\big(\mathrm{term}(H')\big)\,. \tag{41}$$

Definition 5.6 makes observed endpoint equivalence weaker than equality of terminal relational configurations. To represent historically relevant information between the initial and terminal states, let $\mathcal{R} = \{r_1, \ldots, r_m\}$ be a finite nonempty family of passive readouts $r_\ell : \Omega \to Y_\ell$. Definition 5.7 records these readouts along the full history.

Definition 5.7 (History signature)**.** For $H = (\boldsymbol{\lambda}; \omega_0, \ldots, \omega_n) \in \mathscr{H}(\Omega)$, the $\mathcal{R}$-state signature of H is the finite sequence stated in Equation 42. The signature records selected state readouts and does not, by itself, encode the retained operation or update labels in $\boldsymbol{\lambda}$.

$$\mathrm{Sig}_{\mathcal{R}}(H) = \Big((r_1(\omega_k), \ldots, r_m(\omega_k))\Big)_{k=0}^{n}. \tag{42}$$

The signature in Definition 5.7 is intentionally theory-neutral. An application can choose readouts for standing, admitted information, active obligations, review access, settlement status, or another procedural feature. A later theory of procedural justice can then specify which components are normatively salient.

Definition 5.8 introduces the corresponding equivalence relation.

Definition 5.8 (Signature equivalence)**.** Two histories $H, H' \in \mathscr{H}(\Omega)$ are $\mathcal{R}$*-signature equivalent* when Equation 43 holds.

$$\mathrm{Sig}_{\mathcal{R}}(H) = \mathrm{Sig}_{\mathcal{R}}(H'). \tag{43}$$

The comparison levels now admit an implication hierarchy. Proposition 5.9 states the hierarchy for histories of equal length when the readout family $\mathcal{R}$ contains r_{out} evaluated at the terminal state.

Proposition 5.9 (Hierarchy of state-signature equivalence)**.** *Let $H = (\mathbf{p}; \omega_0, \ldots, \omega_n)$ and $H' = (\mathbf{p}'; \omega'_0, \ldots, \omega'_n)$ be realized histories with state paths of equal length. If $\omega_k = \omega'_k$ for every k, then H and H' are $\mathcal{R}$-signature equivalent. If they are $\mathcal{R}$-signature equivalent and $r_{\text{out}} \in \mathcal{R}$, then they are observationally endpoint-equivalent under r_{out}. The converse implications need not hold.*

Proof. Equality of every configuration gives equality of every readout value at every step, so Equation 43 follows directly from Equation 42. If the signatures are equal and $r_{\text{out}} \in \mathcal{R}$, their terminal components have equal r_{out} values, which is Equation 41. The converses can fail when the chosen readouts are many-to-one or omit intermediate state differences; those properties make distinct state paths observationally or signature-equivalent without requiring that every such pair collapse. □

Proposition 5.9 identifies the formal location of a common inferential mistake. Equality of a visible endpoint does not establish equality of terminal relational configurations, and equality of terminal configurations does not reconstruct the route or operation trace through which those configurations were reached. A conversion procedure can therefore preserve the visible settlement while altering standing, information, review access, or another history-dependent feature.

Figure 3 illustrates this distinction. The two histories terminate in configurations with the same visible outcome readout even though their intermediate relational histories differ.

The ordering mechanisms developed in this section are summarized in Table 7. Each mechanism is defined structurally before its possible normative significance is considered.

Table 7 completes the structural derivation promised in Section 2. Recognition, classification, conversion, and revision can generate ordering divergence through distinct mechanisms: schema change, domain change, output change, and coarse-graining of historically different trajectories. Section 6 next examines the special case in which later recognition encounters relational changes that cannot be fully undone by simply reversing the original operation sequence.

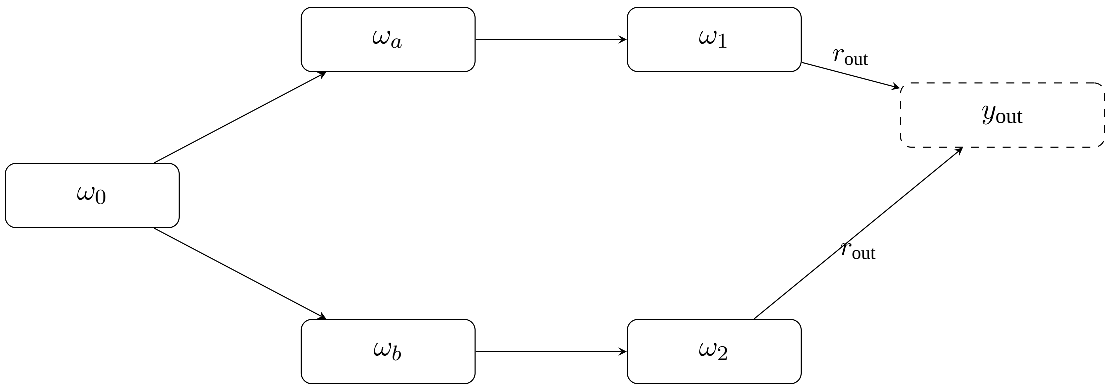

Figure 3: Observed endpoint equivalence with historical divergence. Distinct intermediate and terminal relational configurations can map to the same visible conversion outcome under a coarse readout r_{out}.

Table 7: Structural mechanisms of ordering divergence examined in Section 5.

Mechanism	Formal indicator	Procedural meaning
Recognition-sensitive conversion	$\mathfrak{F}_X(U_R\omega) \neq \mathfrak{F}_X(\omega)$	Recognition changes the concrete conversion transformation selected afterward.
Recognition-enabled conversion	$\omega \notin D_{U_X}, U_R(\omega) \in D_{U_X}$	Recognition creates conversion accessibility.
Classification-sensitive valuation	$y_{KV}(\omega) \neq y_{VK}(\omega)$	Classification and valuation order changes the valuation representation.
Conversion-induced review loss	$q \in \mathcal{Q}(\omega)$, $q \notin \mathcal{Q}(U_X\omega)$	Conversion removes access to a selected revision procedure.
Observed endpoint equivalence	Equal r_{out} values with distinct histories	Coarse outcome equality masks relational or procedural history differences.

6 Irreversibility and Retrospective Recognition

6.1 Irreversible Relational Change

Section 5 established that procedural order can change the transformation selected at a later step, the domain on which a later step is executable, and the relational history that remains after an apparently similar endpoint. The present subsection isolates a further structural question: after an order-sensitive transformation has occurred, which earlier configurations or procedurally relevant features remain reachable from the configuration that actually resulted? Its role is to define irreversibility at the level of modeled accessibility before Sections 6.2 and 6.3 introduce delayed recognition and institutional repair. The construction proceeds through an admissible update repertoire, exact reachability, and readout-relative recovery. The resulting notions are model-relative and do not establish a metaphysical thesis about the irreversibility of history.

The partial transformations developed in Section 3 already contain the ingredients required for a reachability analysis. What must be added is an explicit specification of which future transformations count as available for the comparison. Let $\mathcal{U} \subseteq \mathrm{PEnd}(\Omega)$ be a chosen family of partial updates. Depending on the application, $\mathcal{U}$ can contain recognition, revision, settlement, reopening, compensatory, or other institutionally admissible updates. The family is an analytical input: changing the legal regime, institutional horizon, available resources, or modeled intervention set can change $\mathcal{U}$, and therefore can change the resulting reachability relation.

Definition 6.1 collects finite admissible compositions of updates from $\mathcal{U}$. The identity map is included so that every configuration is reachable from itself through the zero-step path.

Definition 6.1 (Generated update family)**.** For a chosen update repertoire $\mathcal{U} \subseteq \mathrm{PEnd}(\Omega)$, let $\mathrm{Comp}(\mathcal{U})$

denote the family of finite composites stated in Equation 44.

$$\mathrm{Comp}(\mathcal{U}) = \{\mathrm{id}_\Omega\} \cup \{V_n \circ \cdots \circ V_1 \mid n \geq 1,\ V_k \in \mathcal{U}\}\,. \tag{44}$$

Each nonidentity composite is interpreted with the partial-composition domain defined in Equation 7.

Definition 6.1 records possible finite update sequences without asserting that every symbolic sequence is executable from every configuration. Domain membership remains decisive. Definition 6.2 therefore defines reachability from a particular configuration by retaining only composites that are actually defined there.

Definition 6.2 (Update-reachable set)**.** For $\omega \in \Omega$ and an update repertoire $\mathcal{U}$, the $\mathcal{U}$-reachable set from ω is the set in Equation 45.

$$\mathrm{Reach}_{\mathcal{U}}(\omega) = \{\Psi(\omega) \mid \Psi \in \mathrm{Comp}(\mathcal{U}),\ \omega \in D_\Psi\}\,. \tag{45}$$

The reachable set in Definition 6.2 is relative both to the modeled configuration space and to the chosen update repertoire. It is therefore narrower than a claim about everything that could happen in the social world. For example, a model that includes ordinary administrative review but excludes legislative change can classify a configuration as inaccessible even though a later change of law could create a new route. Conversely, adding a formally imaginable transformation to $\mathcal{U}$ without institutional grounds would exaggerate practical recoverability. Reachability is meaningful only when the repertoire corresponds to the intervention horizon under analysis.

The set in Equation 45 induces a directed accessibility relation. Definition 6.3 introduces notation for this relation.

Definition 6.3 (Update accessibility)**.** For $\omega, \omega' \in \Omega$, write $\omega \rightsquigarrow_{\mathcal{U}} \omega'$ when the condition in Equation 46 holds.

$$\omega \rightsquigarrow_{\mathcal{U}} \omega' \quad \Longleftrightarrow \quad \omega' \in \mathrm{Reach}_{\mathcal{U}}(\omega). \tag{46}$$

Proposition 6.4 records two elementary properties that become useful when discussing reversal and repair.

Proposition 6.4 (Reflexive and transitive update accessibility)**.** *The relation $\rightsquigarrow_{\mathcal{U}}$ in Definition 6.3 is reflexive and transitive on Ω. It need not be symmetric or antisymmetric.*

Proof. Reflexivity follows because $\mathrm{id}_\Omega \in \mathrm{Comp}(\mathcal{U})$ and is defined at every $\omega \in \Omega$. For transitivity, suppose $\omega \rightsquigarrow_{\mathcal{U}} \omega'$ through a composite Ψ and $\omega' \rightsquigarrow_{\mathcal{U}} \omega''$ through a composite Θ. The domain conditions imply that $\Theta \circ \Psi$ is defined at ω, and closure of finite composition gives $\Theta \circ \Psi \in \mathrm{Comp}(\mathcal{U})$. Hence $\omega \rightsquigarrow_{\mathcal{U}} \omega''$. Symmetry can fail when the reverse path is unavailable, while antisymmetry can fail when two distinct configurations are mutually reachable. □

Proposition 6.4 gives a precise sense in which procedural accessibility can have direction. A conversion, settlement, or recognition event can move the system into a region from which some earlier configuration is no longer reachable under the modeled future repertoire. Definition 6.5 uses this asymmetry to define exact structural irreversibility.

Definition 6.5 (Exact structural irreversibility)**.** Let ω_1 be a later configuration reached from a reference configuration ω_0 through an already-realized procedural history, and let $\mathcal{U}$ be the chosen repertoire of updates available for subsequent intervention. The realized change is *exactly irreversible relative to* $\mathcal{U}$ when Equation 47 holds.

$$\omega_0 \notin \mathrm{Reach}_{\mathcal{U}}(\omega_1). \tag{47}$$

Definition 6.5 concerns exact equality of modeled configurations. Under the reduced representation in Equation 20, exact return therefore requires restoration of the accumulated history $\mathcal{H}_\omega$, the active relational boundary $\mathcal{B}_\omega$, and the effective nomos $\mathcal{N}_\omega$ simultaneously. This requirement is intentionally demanding. Under a historical-persistence convention—the modeling choice that realized events remain represented in the historical component—the record of an earlier event is retained rather than overwritten by later updates. If a later update preserves the record of an intervening conversion, settlement, contestation, or repair, the resulting configuration can differ from the earlier configuration even when present rights, obligations, holdings, or institutional statuses have been restored. Exact structural irreversibility can therefore coexist with substantial present-oriented recovery.

This point also clarifies the status of that historical-persistence convention. It supplies a traceable modeling representation in which later transformations do not erase realized event history. If $\mathcal{H}_\omega$ is implemented as a strictly append-only record and every update in $\mathcal{U}$ preserves that record, then exact return to a pre-intervention configuration after a nontrivial event is excluded largely by the state representation itself. Exact irreversibility in that implementation should therefore be read as a deliberately strong bookkeeping benchmark rather than as an independently discovered social law. The convention does not establish that physical or social reality possesses a universal arrow of time, and Definition 6.5 does not rely on such a thesis. A different model could quotient historical records more coarsely or omit them from its state description, thereby changing the equality criterion on Ω.

Procedural analysis can require a weaker comparison than exact equality. An institution can care about whether standing has been restored, whether an obligation has been reopened, whether a transfer has been reversed, or whether a review right is again available, while still preserving a record that the earlier event occurred. The passive readout family $\mathcal{R} = \{r_1, \ldots, r_m\}$ introduced in Definition 5.7 can be reused for this purpose. Definition 6.6 applies the family to individual configurations rather than to complete histories.

Definition 6.6 (Configuration readout equivalence)**.** Let $\mathcal{R} = \{r_1, \ldots, r_m\}$, with $m \geq 1$, be a finite family of passive readouts on Ω. Two configurations $\omega, \omega' \in \Omega$ are *$\mathcal{R}$-equivalent*, written $\omega \sim_{\mathcal{R}} \omega'$, when Equation 48 holds.

$$\omega \sim_{\mathcal{R}} \omega' \quad \Longleftrightarrow \quad r_\ell(\omega) = r_\ell(\omega') \quad \text{for every } \ell = 1, \ldots, m. \tag{48}$$

Definition 6.6 does not require the readouts to be numerical or to share a common codomain. The family can include application-specific readouts of standing, possession, obligations, recognized attribution, settlement status, effective nomos, or review access. The relation is intentionally indexed by $\mathcal{R}$: two configurations that are equivalent under a monetary outcome readout can remain distinguishable under readouts of recognition status or review access.

For a reference configuration ω_0, let $[\omega_0]_{\mathcal{R}}$ denote its equivalence class under Definition 6.6. Definition 6.7 then distinguishes readout-relative recovery from exact return.

Definition 6.7 (Readout-relative recovery)**.** Let ω_0 be a reference configuration and ω_1 a later configuration. The reference is *recoverable from ω_1 relative to $(\mathcal{U}, \mathcal{R})$* when Equation 49 holds.

$$\mathrm{Reach}_{\mathcal{U}}(\omega_1) \cap [\omega_0]_{\mathcal{R}} \neq \varnothing. \tag{49}$$

If the intersection is empty, the reference is *$\mathcal{R}$-irrecoverable relative to $\mathcal{U}$*.

Definition 6.7 makes the recovery criterion explicit. A later configuration can be exactly different from ω_0 while entering the same equivalence class under procedurally selected readouts. A reversed transfer, for example, can restore a holdings readout while leaving the historical record, transaction costs, informational effects, or institutional interpretation changed. The formalism records this possibility without deciding whether the restored readouts are normatively sufficient.

Proposition 6.8 states the logical relation between exact return and readout-relative recovery.

Proposition 6.8 (Hierarchy of exact and readout-relative recovery)**.** *If $\omega_0 \in \mathrm{Reach}_{\mathcal{U}}(\omega_1)$, then ω_0 is recoverable from ω_1 relative to $(\mathcal{U}, \mathcal{R})$ for every readout family $\mathcal{R}$. Consequently, $\mathcal{R}$-irrecoverability implies failure of exact return. The converse implication need not hold.*

Proof. Every configuration is equivalent to itself under Equation 48. Hence $\omega_0 \in [\omega_0]_{\mathcal{R}}$. If $\omega_0 \in \text{Reach}_{\mathcal{U}}(\omega_1)$, then ω_0 belongs to the intersection in Equation 49, so the intersection is nonempty. The contrapositive gives the second statement. The converse can fail when $\omega_0 \notin \text{Reach}_{\mathcal{U}}(\omega_1)$ but some distinct reachable configuration $\omega^* \in \text{Reach}_{\mathcal{U}}(\omega_1)$ satisfies $\omega^* \sim_{\mathcal{R}} \omega_0$. $\square$

Proposition 6.8 prevents a common conflation. The failure of literal reversal does not establish the failure of every procedurally relevant form of recovery. At the same time, recovery under a deliberately coarse readout does not establish restoration of the fuller relational configuration. The conclusion always depends on which features the readout family retains.

Some interventions restore only part of a selected readout family. To represent this intermediate case without yet assigning a justice criterion, Definition 6.9 records which readouts of a reference configuration are matched by a reachable configuration.

Definition 6.9 (Restoration profile)**.** For a readout family $\mathcal{R} = \{r_1, \ldots, r_m\}$, a reference configuration ω_0, and a comparison configuration ω^*, the restoration profile is the index set in Equation 50.

$$\text{Rest}_{\mathcal{R}}(\omega^*; \omega_0) = \{\ell \in \{1, \ldots, m\} \mid r_\ell(\omega^*) = r_\ell(\omega_0)\}\,. \tag{50}$$

Definition 6.9 is descriptive and, by itself, makes no reachability claim. When a restoration profile is used to describe recovery from a current configuration, the comparison configuration must separately be shown to belong to the relevant reachable set. A larger restoration profile is not automatically more just, because different procedural theories can assign different importance to standing, participation, review, attribution, compensation, finality, or other dimensions. The profile instead supplies a typed object that Section 6.3 can use when distinguishing reopening and repair from exact reversal.

Table 8 summarizes the three comparison levels developed in this subsection. Their separation is important because the same post-conversion intervention can fail at one level while succeeding at another.

Table 8 also clarifies why irreversibility is best treated as a qualified structural statement. An earlier configuration can be exactly inaccessible under a given update repertoire while selected present relations remain recoverable. A different repertoire can later enlarge the reachable set, and a different readout family can change the equivalence criterion. Structural irreversibility therefore records a relation among a reference configuration, a later configuration, an admissible update repertoire, and a chosen level of state discrimination.

Caution 6.10 (Scope of structural irreversibility)**.** Definition 6.5 and Definition 6.7 support claims about accessibility within a specified model. They do not by themselves establish that an institutional loss is permanent, that compensation is impossible, that a historical event can be erased, or that a particular response is just. Those stronger conclusions require additional empirical and normative premises.

Caution 6.10 sets the boundary for the next step. The relevant question for retrospective recognition is now precise. When conversion occurs before recognition, the later recognition update begins from a configuration already changed by conversion. Section 6.2 can therefore ask whether the recognition-first reference configuration, or selected procedural features of that reference, remains reachable from the configuration produced by delayed recognition.

Table 8: Comparison levels for model-relative irreversibility and recovery.

Comparison level	Formal criterion	Interpretation
Exact return	$\omega_0 \in \text{Reach}_{\mathcal{U}}(\omega_1)$	The earlier modeled configuration, including every state component represented in Ω, is reachable again.
Readout-relative recovery	$\text{Reach}_{\mathcal{U}}(\omega_1) \cap [\omega_0]_{\mathcal{R}} \neq \varnothing$	A reachable configuration matches the reference on every readout selected in $\mathcal{R}$.
Partial restoration	$\omega^* \in \text{Reach}_{\mathcal{U}}(\omega_1)$ and $\varnothing \neq \text{Rest}_{\mathcal{R}}(\omega^*; \omega_0) \subsetneq \{1, \ldots, m\}$	A reachable configuration restores selected modeled features while others remain different.

6.2 Delayed Recognition

The reachability distinctions in Section 6.1 become operationally significant when recognition occurs after a conversion that could have been preceded by recognition. The present subsection has three roles. It identifies the configuration on which delayed recognition is actually executed, compares early- and delayed-recognition histories without treating the recognition-first history as an automatic normative ideal, and isolates the effect of settlement on the direct availability of retrospective recognition. The analysis remains structural: the justice significance of recognition timing is reserved for Section 7.

Retrospective recognition contains two temporal aspects that should remain distinct. Its *referential content* can concern an earlier contribution, status, event, claim, or relation represented in the accumulated history. Its *performative execution* occurs at the configuration that exists when the recognition procedure is carried out. Under the historical-persistence convention just described, an earlier event can remain represented in $\mathcal{H}_\omega$ and therefore remain available as an object of later institutional interpretation. The state-changing recognition update nevertheless acts on the present configuration in Ω. This distinction prevents a retrospective decision from being modeled as though an operation were literally inserted into an earlier realized state.

Consider an initial configuration $\omega_0 \in M_{RX}$, where both recognition–conversion and conversion–recognition orders are executable. The intermediate conversion configuration and the two terminal comparison configurations are collected in Equation 51.

$$
\begin{aligned}
\omega_X &= U_X(\omega_0), \\
\omega_{RX} &= (U_X \circ U_R)(\omega_0), \\
\omega_{XR} &= (U_R \circ U_X)(\omega_0) = U_R(\omega_X).
\end{aligned}
\tag{51}
$$

Equation 51 gives ω_{RX} and ω_{XR} different analytical roles. The configuration ω_{RX} is the terminal state of the recognition-first comparison history $H_{RX}(\omega_0)$. The configuration ω_{XR} is the terminal state produced when conversion occurs first and recognition is performed retrospectively. Calling ω_{RX} a *reference configuration* in the following analysis means only that it supplies a comparison target. Additional normative premises are required before it can be treated as a preferred or just outcome.

The state-dependent semantics from Section 3 also apply to the recognition operation itself. Early recognition is instantiated from ω_0, whereas delayed recognition is instantiated from ω_X. The two concrete recognition transformations are shown in Equation 52.

$$
\Phi_R^{[\omega_0]} = \mathfrak{F}_R(\omega_0), \qquad \Phi_R^{[\omega_X]} = \mathfrak{F}_R(\omega_X). \tag{52}
$$

Equation 52 shows why delayed recognition cannot be assumed to use the earlier recognition transformation unchanged after conversion. Conversion can alter active relations, evidentiary records, institutional categories, possession, obligations, or the effective nomos. Any of these changes can alter the recognition transformation selected by $\mathfrak{F}_R$. Definition 6.11 records this symmetric counterpart of recognition-sensitive conversion.

Definition 6.11 (Conversion-sensitive recognition schema)**.** Let $\omega_0 \in D_{U_X}$, write $\omega_X = U_X(\omega_0)$, and suppose recognition is defined at both ω_0 and ω_X. The recognition schema $\mathfrak{F}_R$ is *conversion-sensitive at* ω_0 when Equation 53 holds.

$$\mathfrak{F}_R(\omega_X) \neq \mathfrak{F}_R(\omega_0). \tag{53}$$

Definition 6.11 concerns the concrete transformation selected for recognition. It does not imply that every observable result of recognition differs across the two timings. Distinct partial transformations can coincide on selected readouts, and different histories can later enter the same readout-equivalence class. The definition therefore identifies a source of procedural divergence while leaving its empirical magnitude and normative significance open.

The distinction becomes especially important when recognition has generative effects. Recognition performed at ω_0 can alter the configuration from which conversion is subsequently instantiated. Delayed recognition performed at ω_X enters after that conversion instantiation has already occurred. A later recognition event can change present standing, attribution, claim status, or future accessibility, yet those changes do not automatically reconstruct the conversion transformation that would have been selected from $U_R(\omega_0)$. In the present model, the already-realized conversion remains part of the history on which subsequent operations act.

This observation can be stated without assuming exact irreversibility. Let $\mathcal{U}_{\text{post}}$ denote the repertoire of updates institutionally available after delayed recognition. Equation 54 states the condition under which the recognition-first terminal configuration remains exactly reachable from the delayed-recognition terminal configuration.

$$\omega_{RX} \in \text{Reach}_{\mathcal{U}_{\text{post}}}(\omega_{XR}). \tag{54}$$

When Equation 54 holds, a finite sequence of admissible later updates can reach the same terminal configuration generated by the recognition-first comparison path. The condition is demanding because equality is evaluated in the full modeled configuration space. A weaker comparison uses the readout family $\mathcal{R}$. Equation 55 gives the corresponding readout-relative condition.

$$\text{Reach}_{\mathcal{U}_{\text{post}}}(\omega_{XR}) \cap [\omega_{RX}]_{\mathcal{R}} \neq \varnothing. \tag{55}$$

Equation 55 permits retrospective recognition and later intervention to restore selected procedurally relevant features even when exact convergence in Equation 54 fails. For example, a procedure can restore recognized authorship, standing, a claim entitlement, or review access while preserving a different transaction history. The choice of $\mathcal{R}$ determines which dimensions count as matched; the equation therefore supplies a comparison criterion rather than a sufficient condition for justice.

Corollary 6.12 follows directly from Proposition 6.8 and records the relation between the two conditions.

Corollary 6.12 (Delayed-recognition recovery hierarchy)**.** *If Equation 54 holds, then Equation 55 holds for every readout family $\mathcal{R}$. Failure of Equation 55 therefore implies failure of exact convergence to ω_{RX}. Readout-relative convergence can occur while exact convergence fails.*

Proof. Apply Proposition 6.8 with reference configuration ω_{RX}, later configuration ω_{XR}, and update repertoire $\mathcal{U}_{\text{post}}$. □

Corollary 6.12 gives a precise interpretation of a common retrospective situation. Delayed recognition can be consequential even when it cannot make the realized history equivalent to the recognition-first history. Its consequence can lie in changing the present configuration and enlarging the set of future histories from that configuration. This forward-generative effect is analytically separate from historical equivalence.

The comparison can be sharpened further by using the restoration profile in Definition 6.9. For a reachable post-recognition configuration $\omega^* \in \text{Reach}_{\mathcal{U}_{\text{post}}}(\omega_{XR})$, Equation 56 applies the profile to the recognition-first reference.

$$\mathrm{Rest}_{\mathcal{R}}(\omega^*;\omega_{RX}) = \{\ell \mid r_\ell(\omega^*) = r_\ell(\omega_{RX})\}\,. \tag{56}$$

Equation 56 allows a delayed-recognition procedure to be described dimension by dimension. A later configuration might match the recognition-first reference with respect to standing and attribution while differing with respect to possession, compensation, review history, or settlement status. Such a profile is more informative than a single binary statement that recognition was either "effective" or "ineffective." Its normative interpretation depends on which dimensions a procedural theory regards as salient and on the burdens created during the period of nonrecognition.

Recognition timing also interacts with settlement. Section 4.3 separated conversion X from settlement S precisely because a conversion can occur before institutional finality has been assigned. Suppose conversion is followed by settlement. The post-settlement configuration is defined in Equation 57.

$$\omega_{XS} = (U_S \circ U_X)(\omega_0). \tag{57}$$

Direct retrospective recognition after settlement requires $\omega_{XS} \in D_{U_R}$. Settlement-conditioned recognition closure occurs when recognition remains directly executable after conversion but ceases to be directly executable after settlement, as stated in Equation 58.

$$\omega_X \in D_{U_R}, \qquad \omega_{XS} \notin D_{U_R}. \tag{58}$$

Proposition 6.13 records the resulting difference in direct procedural accessibility.

Proposition 6.13 (Settlement-conditioned recognition closure)**.** *Let $\omega_0 \in D_{U_X}$, let $\omega_X = U_X(\omega_0)$, and suppose settlement is defined at ω_X. If Equation 58 holds, then the delayed-recognition trace (X, R) is directly realizable from ω_0, while the three-step trace (X, S, R) is not directly realizable through the same recognition update U_R.*

Proof. The condition $\omega_X \in D_{U_R}$ implies $\omega_0 \in D_{U_R \circ U_X}$ by Equation 7, so (X, R) is realizable. Settlement produces ω_{XS}, and $\omega_{XS} \notin D_{U_R}$ prevents the final recognition step in (X, S, R) from being executed through U_R. Hence that three-step trace is not directly realizable through the same recognition procedure. □

Proposition 6.13 does not imply that recognition after settlement is institutionally impossible. It says that the recognition operation modeled by U_R is unavailable directly from ω_{XS}. A reopening, appeal, annulment, legislative change, compensatory procedure, or another update can enlarge the operative repertoire or move the system to a configuration from which recognition becomes available again. Those possibilities require an additional procedural step and are therefore structurally different from direct delayed recognition before settlement.

Table 9 summarizes the distinctions developed in this subsection. The table separates the time at which recognition is performed from the historical object to which it can refer, the transformation selected at that time, and the form of recovery that can remain available afterward.

Table 9 locates retrospective recognition within the broader order-sensitive-composition argument. Delayed recognition is a new event executed from a changed configuration; its content can concern an earlier event, and its effects can be substantial, while its history remains distinguishable from recognition that preceded conversion. Settlement can add a further domain boundary by changing whether recognition is directly executable at all. The next subsection develops the additional transformations through which a closed or divergent procedure can be reopened, repaired, or normatively rectified.

Table 9: Structural distinctions in delayed and retrospective recognition.

Analytical dimension	Formal object	Structural interpretation
Execution configuration	ω_0 or ω_X	Recognition is executed on the configuration present when the procedure occurs.
Recognition instantiation	$\Phi_R^{[\omega_0]}$ or $\Phi_R^{[\omega_X]}$	Conversion can change the concrete recognition transformation selected by the schema.
Comparison reference	ω_{RX}	The recognition-first terminal configuration supplies a comparison target without receiving an automatic normative priority.
Exact convergence	Equation 54	Later admissible updates reproduce the full modeled recognition-first terminal configuration.
Readout-relative convergence	Equation 55	Later updates reproduce the selected procedural readouts while other state components can remain different.
Settlement-conditioned access	Equation 58	Settlement can remove direct access to the recognition update that remained available immediately after conversion.

6.3 Reopening, Repair, and Rectification

The preceding subsections separated exact return from readout-relative recovery and showed that delayed recognition can encounter a settlement-conditioned loss of direct accessibility. The present subsection develops the corresponding response structure. Its role is to distinguish three response concepts that can be compressed into a single language of correction. Reopening concerns the restoration of procedural access to a review or revision operation. Repair concerns admissible future histories that recover selected relational or procedural features from the configuration that actually exists. Rectification is a normative interpretation of a response history under an explicitly supplied theory of justice. The analysis proceeds in that order so that accessibility and recoverability can be characterized before normative adequacy is assessed.

Reopening is most naturally expressed through the review-access set introduced in Definition 4.5. Let $q \in \mathcal{L}_Q$ denote a concrete review or revision procedure, and let ω_c be a configuration at which that procedure is unavailable. A reopening need not itself revise the disputed classification, conversion, settlement, or recognition relation. Its minimal structural role is to transform the current configuration into one from which the relevant review procedure becomes institutionally available.

Definition 6.14 (Review reopening)**.** Let $q \in \mathcal{L}_Q$, let $\omega_c \in \Omega$, and let $\mathcal{U}_{\text{open}} \subseteq \text{PEnd}(\Omega)$ be a repertoire of updates capable of altering review access. A composite $\Psi_{\text{open}} \in \text{Comp}(\mathcal{U}_{\text{open}})$ is a *reopening path for q from ω_c* when Equation 59 holds.

$$q \notin \mathcal{Q}(\omega_c), \qquad \omega_c \in D_{\Psi_{\text{open}}}, \qquad q \in \mathcal{Q}\big(\Psi_{\text{open}}(\omega_c)\big)\,. \tag{59}$$

Definition 6.14 is deliberately narrower than substantive revision. It classifies a path as reopening when the path changes the accessibility structure surrounding q. A reopening can be generated by an appeal gateway, annulment of finality, recognition of standing, authorization to contest, a jurisdictional reassignment, or another modeled update. The definition does not require the disputed result itself to change during the reopening path.

Under Assumption 4.6, reopening has an immediate domain consequence. Proposition 6.15 records this relation.

Proposition 6.15 (Reopening restores modeled domain access)**.** *Suppose Assumption 4.6 applies to a review procedure q represented by the partial update U_q. If Ψ_{open} is a reopening path for q from ω_c, then* *the terminal reopening configuration belongs to the domain of U_q, as stated in Equation 60.*

$$\Psi_{\text{open}}(\omega_c) \in D_{U_q}. \tag{60}$$

Proof. Equation 59 gives $q \in \mathcal{Q}(\Psi_{\text{open}}(\omega_c))$. Assumption 4.6 identifies membership of q in the review-access set with membership of the corresponding configuration in D_{U_q}. Equation 60 follows. □

Proposition 6.15 does not imply that the subsequent review will succeed, that it will restore a reference configuration, or that reopening itself is normatively sufficient. It establishes only that a procedure previously outside the modeled access set has become executable. This distinction is useful after settlement-conditioned closure in Proposition 6.13: an additional path can restore access to review before any substantive remedial transformation occurs.

Repair requires a richer object because the path taken after reopening can itself remain procedurally significant. A terminal-state set alone records where an intervention ends while suppressing intermediate recognition, delay, burden, contestation, or settlement events. The paper therefore extends the realized-history construction from Section 3 to future update sequences generated by a chosen intervention repertoire.

Definition 6.16 (Future update history)**.** Let $\mathcal{U} \subseteq \text{PEnd}(\Omega)$ and let $\omega \in \Omega$. The family $\mathfrak{H}_{\mathcal{U}}(\omega)$ consists of every finite labeled future history generated from ω by an admissible sequence of updates from $\mathcal{U}$, together with the zero-step history. For $n \geq 1$, its elements have the form stated in Equation 61; for $n = 0$, the update tuple is empty and the history consists only of the initial configuration ω.

$$\begin{aligned} &H = \big((V_1, \dots, V_n); (\omega_0, \omega_1, \dots, \omega_n)\big), \quad \omega_0 = \omega, \\ &\omega_k = V_k(\omega_{k-1}), \qquad V_k \in \mathcal{U}, \quad \omega_{k-1} \in D_{V_k}, \quad k = 1, \dots, n. \end{aligned} \tag{61}$$

For $H \in \mathfrak{H}_{\mathcal{U}}(\omega)$, let $\tau(H)$ denote its terminal configuration. These future update histories are included in the general labeled-history space $\mathscr{H}(\Omega)$ by using the concrete update identities $(V_1, \dots, V_n)$ as the retained label sequence. Retaining that sequence ensures that two remedial paths that pass through the same configurations by different institutional operations need not collapse into the same history.

Definition 6.16 preserves the same partial-composition discipline used throughout the paper. Every adjacent update must be executable from the configuration actually generated by the preceding updates. The terminal configurations of histories in $\mathfrak{H}_{\mathcal{U}}(\omega)$ coincide with the reachable set in Equation 45, but the history family retains information that the reachable set intentionally quotients away.

A repair target can now be specified through selected components of the restoration profile. Let $\mathcal{R} = \{r_1, \dots, r_m\}$, with $m \geq 1$, be a readout family, let ω_{ref} be a reference configuration, and let $J \subseteq \{1, \dots, m\}$ be a nonempty set of target readout indices. The set J records which selected features are to be recovered for the structural comparison. It does not state that those features are morally sufficient.

Definition 6.17 (Repair corridor)**.** For a current configuration ω, update repertoire $\mathcal{U}$, reference configuration ω_{ref}, readout family $\mathcal{R}$, and nonempty target index set J, the *repair corridor* is the history family in Equation 62.

$$\text{Rep}_{\mathcal{U},\mathcal{R},J}(\omega; \omega_{\text{ref}}) = \{H \in \mathfrak{H}_{\mathcal{U}}(\omega) \mid J \subseteq \text{Rest}_{\mathcal{R}}(\tau(H); \omega_{\text{ref}})\}. \tag{62}$$

Definition 6.17 treats repair as a family of admissible histories rather than a unique corrective act. A corridor can contain paths that reopen review and then revise a settlement, paths that preserve settlement while restoring attribution or standing, paths that add compensation, and paths that alter future access while leaving some historical or distributive differences in place. The definition requires only the selected structural target encoded by J. A normative theory can later distinguish among histories inside the same corridor.

The history-sensitive formulation also prevents a second form of endpoint compression. Two repair histories can terminate at configurations with the same restoration profile while differing in duration, sequence, burdens borne during the process, opportunities for participation, or the institutional acknowledgment recorded along the way. Those differences remain represented in H even when $\tau(H)$ is identical. This property is consistent with the broader distinction between terminal equivalence and configuration-path equivalence in Definition 3.7.

Definition 6.17 nests the recovery notions developed in Section 6.1. Proposition 6.18 states the relation precisely.

Proposition 6.18 (Repair-corridor recovery relation)**.** *Let $J = \{1, \ldots, m\}$. The repair corridor in Definition 6.17 is nonempty if and only if the reference configuration ω_{ref} is recoverable from ω relative to $(\mathcal{U}, \mathcal{R})$ in the sense of Definition 6.7.*

Proof. If the corridor is nonempty, there exists $H \in \mathfrak{H}_{\mathcal{U}}(\omega)$ whose terminal configuration $\tau(H)$ matches every readout in $\mathcal{R}$ with the reference configuration. Hence $\tau(H) \in \text{Reach}_{\mathcal{U}}(\omega) \cap [\omega_{\text{ref}}]_{\mathcal{R}}$, so readout-relative recovery holds. Conversely, if readout-relative recovery holds, some $\omega^* \in \text{Reach}_{\mathcal{U}}(\omega) \cap [\omega_{\text{ref}}]_{\mathcal{R}}$ is the terminal configuration of at least one admissible future update history from ω. That history has the full restoration profile and therefore belongs to the repair corridor with $J = \{1, \ldots, m\}$. $\square$

Proposition 6.18 shows that the corridor construction extends rather than replaces the readout-relative recovery criterion. When J is a proper subset, a corridor can remain nonempty even though complete readout-relative recovery fails. This is the structurally relevant case for partial repair after irreversible change. It also leaves open the possibility that exact return fails under Equation 47 while a substantial repair corridor remains available.

The choice of J should remain explicit. If J contains only recognition status, a corridor can classify a history as structurally reparative even when compensation, participation, review access, or another salient relation remains unmatched. If J contains review access and standing but excludes monetary allocation, the same history receives a different structural description. The formalism therefore exposes the target selection rather than hiding it inside the word "repair." Section 7 will supply theory-relative interpretations of such selections.

Reopening and repair can now be related without being identified. A reopening path changes the accessibility of at least one review procedure. A repair history reaches a configuration satisfying selected restoration targets. Some repair histories contain a reopening path as an initial segment; others operate through mechanisms that do not require reopening. Conversely, a reopening path can terminate before any selected relational feature of the reference configuration has been restored. The distinction is therefore both compositional and substantive.

Rectification introduces a further layer. Within the present paper, the term denotes a normative status assigned to a response history under an explicitly specified evaluative framework. A structurally available repair corridor can contain several histories that satisfy the same selected readouts while differing in participation, burden allocation, acknowledgment, compensation, finality, or later contestability. Whether one of those histories counts as adequate rectification cannot be derived from reachability, restoration-profile size, or structural order sensitivity alone.

This separation also constrains comparisons between a repair history and the earlier reference history. The reference configuration ω_{ref} can serve as a counterfactual or procedural benchmark without becoming an automatic target of justice. In some applications, later information, changed circumstances, third-party reliance, ecological change, or newly recognized relations can make literal restoration undesirable or infeasible even when some form of remedial response remains warranted. The corridor construction accommodates this possibility because J can identify selected recoverable relations while the realized repair history continues from the actual current configuration.

Table 10 summarizes the conceptual separation established in this subsection. The table places exact reversal alongside the three response concepts so that their different formal and normative roles remain visible.

Table 10 makes the main result of Section 6 explicit. Procedural irreversibility need not collapse the analysis into a binary choice between complete reversal and permanent loss. A closed review route can sometimes be reopened; selected relations can sometimes be repaired from the configuration that

Table 10: Structural roles of reversal, reopening, repair, and rectification.

Concept	Primary formal object	Minimal structural condition	Interpretive status
Exact reversal	$\mathrm{Reach}_{\mathcal{U}}(\omega)$	The full reference configuration is reachable.	Strong state-level recovery; historical equality still depends on the state representation.
Reopening	$\mathcal{Q}(\omega)$ and a reopening path	A previously unavailable review procedure becomes available.	Restores procedural access without determining the review result.
Repair	$\mathrm{Rep}_{\mathcal{U},\mathcal{R},J}(\omega;\omega_{\mathrm{ref}})$	At least one admissible future history restores the selected target readouts.	Structural recovery relative to an explicit target profile; can remain partial and path-sensitive.
Rectification	Theory-relative evaluation of a response history	Additional normative conditions supplied by an evaluative framework are satisfied.	Normative status; it is not entailed by reachability or restoration alone.

actually exists; and the adequacy of such a response requires a separate normative judgment. The structural formalism identifies the available paths, their ordering conditions, and the features they recover. Section 7 turns to the additional question of how procedural theories can assign significance to standing, admissibility, information, participation, revision, and review within those paths.

7 Procedural Justice Interpretations

7.1 Ordering as a Procedural Condition

The preceding sections established a structural account of order-sensitive conversion without assigning a justice ranking to the histories thereby generated. The present section introduces the additional interpretive layer required for procedural evaluation. Its role is to identify when a difference in ordering becomes relevant under a specified procedural theory, while preserving the distinction between structural divergence and normative assessment. The discussion begins with a theory-relative comparison framework and then uses that framework to locate ordering among other procedural conditions. Subsequent subsections apply the same architecture to standing and admissibility, information and participation, and revision and review.

Procedural justice has long treated the organization of decision procedures as a subject of evaluation distinct from the distributive content of their outcomes. Thibaut and Walker's early social-psychological work examined how alternative conflict-resolution procedures affect perceptions of procedural justice, and later work by Lind and Tyler synthesized a broader research programme on reactions to decision procedures across legal, political, and organizational settings [3, 2]. Tyler's study of legal authority further connected perceived procedural fairness with legitimacy and voluntary deference [4]. In political philosophy, Ceva develops a proceduralist account of interactive justice organized around fair hearing and procedural equality under value conflict [1]. These traditions differ in object, method, and normative ambition. The present paper therefore uses them as sources of procedural questions rather than treating them as one unified theory.

The formal results developed earlier add a specific complication to such questions. A conversion procedure may contain operations whose ordering changes the terminal configuration, the intermediate history, the admissibility of later operations, or the concrete transformation selected by a state-dependent operation schema. Under these conditions, procedural criteria cannot always be assessed by asking only whether an operation occurred somewhere in the realized history. The position of the operation within that history can affect what the operation was able to do and which relations were already fixed when it occurred.

A simple example is enough to expose the issue. Suppose a procedure contains recognition, conversion, and review. Recognition can occur before conversion, when it can alter the conversion instantiation under Proposition 4.4; it can occur after conversion, when it is executed from the changed configuration analyzed in Section 6.2; or it can become accessible only after an additional reopening path. A statement that "recognition occurred" therefore compresses several histories whose procedural significance can differ. The same reasoning applies to participation before or after valuation, disclosure before or after authorization, and review before or after settlement.

The paper requires a comparison structure that can represent those differences without forcing them into a single numerical score. Let Θ denote a procedural interpretive frame. The frame specifies which features of realized histories matter for the theory under consideration and how those features can be compared. Configuration readouts and their history signatures remain useful building blocks, but some relevant objects—such as an access trajectory, an influence window, an operation trace, or a duration—are functions of a history as a whole. Definition 7.1 therefore takes history-level feature maps as the general type.

Definition 7.1 (Procedural interpretive frame)**.** A *procedural interpretive frame* is a pair $\Theta = (\mathcal{E}_\Theta, \succeq_\Theta)$, where $\mathcal{E}_\Theta = \{e_1, \ldots, e_m\}$, with $m \geq 1$, is a finite family of theory-relevant history feature maps $e_\ell : \mathscr{H}(\Omega) \to Z_\ell$, and $\succeq_\Theta$ is a preorder on the product feature space $\prod_{\ell=1}^{m} Z_\ell$. The feature tuple is defined in Equation 63.

$$\mathrm{Feat}_\Theta(H) = \big(e_1(H), \ldots, e_m(H)\big). \tag{63}$$

The preorder is not required to be complete, so two feature tuples can remain incomparable.

Definition 7.1 separates feature selection from evaluation. A configuration readout family $\mathcal{R}$ induces the history feature $H \mapsto \mathrm{Sig}_{\mathcal{R}}(H)$, so the state-signature construction in Definition 5.7 is a special case. Other feature maps can encode operation labels, standing or admissibility trajectories, timing, influence windows, review scope, reopening burden, or another history-level relation. The preorder expresses the frame's comparative judgments over the resulting tuples. A preorder imposes reflexivity and transitivity while leaving room for ties and incomparability. No cardinal utility, additive aggregation rule, or complete ranking is required.

Two realized histories can now be procedurally equivalent relative to a frame even when they remain structurally different. Definition 7.2 makes this relation explicit.

Definition 7.2 (Frame-relative procedural equivalence)**.** For a procedural interpretive frame Θ, two realized histories $H, H' \in \mathscr{H}(\Omega)$ are *procedurally equivalent under* Θ when Equation 64 holds.

$$\mathrm{Feat}_\Theta(H) \succeq_\Theta \mathrm{Feat}_\Theta(H') \quad \text{and} \quad \mathrm{Feat}_\Theta(H') \succeq_\Theta \mathrm{Feat}_\Theta(H). \tag{64}$$

The relation is denoted by $H \sim_\Theta H'$.

Frame-relative equivalence is deliberately coarser than identity of labeled histories. A frame can treat two histories as procedurally equivalent when their structural differences fall outside the theory's selected criteria or when the theory regards different feature tuples as normatively tied. Conversely, histories with the same visible settlement can remain procedurally distinct when the selected features record differences in operation trace, standing, information, participation, review access, or another procedural relation.

The strict part of the preorder is useful when a frame does make a directional comparison. For feature tuples s and s', write $s \succ_\Theta s'$ when $s \succeq_\Theta s'$ and the reverse comparison does not hold. Incomparability remains a separate possibility. The notation therefore permits three outcomes for a pair of histories: equivalence, directional comparison, and unresolved comparison under the chosen frame.

Ordering becomes procedurally material when changing the order changes the frame-relative status of the realized histories. Definition 7.3 states this condition without selecting a preferred ordering in advance.

Definition 7.3 (Procedurally material ordering)**.** Let $H_{ij}(\omega)$ and $H_{ji}(\omega)$ be two realized histories generated by alternative orderings of the same two modeled operations from a common initial configuration, whenever both histories are defined. The ordering is *procedurally material under* Θ *at* ω when Equation 65 holds.

$$H_{ij}(\omega) \not\sim_{\Theta} H_{ji}(\omega). \tag{65}$$

Definition 7.3 gives "ordering matters" a theory-relative meaning for mutually realizable orderings. Materiality includes both directional comparison and incomparability. A frame can therefore register an ordering difference as procedurally significant even when it lacks a basis for declaring one history superior. One-way composability is handled separately: because the reverse ordering has no realized history, its procedural significance is evaluated through accessibility, standing, admissibility, or other history features rather than through $H_{ij} \sim_{\Theta} H_{ji}$.

The distinction between structural order sensitivity and procedural materiality is essential. Definition 3.6 asks whether two mutually executable compositions can generate different terminal relational configurations. Definition 7.3 asks whether the resulting histories remain nonequivalent under a supplied procedural frame. Proposition 7.4 records their logical separation.

Proposition 7.4 (Structural and procedural separation)**.** *Structural terminal order sensitivity alone does not entail procedurally material ordering for every procedural interpretive frame* Θ*. Procedurally material ordering can also occur between histories that are terminally equivalent.*

Proof. For the first statement, let two mutually executable orderings produce different terminal configurations while Feat_{Θ} maps their realized histories to feature tuples that are equivalent under $\succeq_{\Theta}$. Equation 64 then holds, so Equation 65 fails despite terminal order sensitivity. For the second statement, let two histories be terminally equivalent while differing at an intermediate configuration or in their operation traces. If $\mathcal{E}_{\Theta}$ records a theory-relevant difference and the resulting feature tuples are nonequivalent under $\succeq_{\Theta}$, Equation 65 holds even though the terminal configurations coincide. □

Proposition 7.4 establishes the limited normative role of strict compositional noncommutativity in this paper. Such noncommutativity identifies one structural source of possible procedural divergence; path divergence and one-way composability supply additional sources within the wider order-sensitive analysis. The interpretive frame determines whether the divergence has procedural significance and, where the frame supplies a directional comparison, how that significance is evaluated. A justice conclusion therefore requires additional normative premises concerning the features included in $\mathcal{E}_{\Theta}$ and the comparisons represented by $\succeq_{\Theta}$.

This separation also clarifies the relation between ordering and outcome quality. A procedurally preferable history under some frame can terminate in an outcome that another substantive theory evaluates unfavorably. A substantively attractive outcome can likewise be produced by a history that a procedural frame evaluates poorly. The present paper addresses the procedural dimension only. Its formalism can represent interactions between procedural and substantive evaluation, although it supplies no general rule for resolving conflicts between them.

Ordering can enter procedural evaluation through at least four structurally distinct channels. First, an earlier operation can change the *content* of a later operation by changing the state from which its schema is instantiated. Recognition-sensitive conversion in Definition 5.1 is an example. Second, an earlier operation can change the *domain* of a later operation, producing one-way composability or loss of review access. Proposition 6.13 provides a settlement-conditioned example. Third, ordering can change the *history* while leaving a coarse terminal readout unchanged, as shown by the equivalence hierarchy in Proposition 5.9. Fourth, an ordering difference can alter which reference features remain reachable through later intervention, as represented by the repair corridor in Definition 6.17. These channels correspond respectively to transformation sensitivity, accessibility sensitivity, path sensitivity, and recovery sensitivity.

Table 11 summarizes these layers. The table also separates the formal question asked at each level from the additional procedural question introduced by the interpretive frame.

Table 11: Structural channels through which ordering can become procedurally material.

Channel	Structural question	Procedural question under Θ	Representative object
Transformation sensitivity	Does an earlier operation change the later transformation selected by a state-dependent schema?	Does the changed transformation alter a theory-relevant history feature or feature tuple?	$\Phi_p^{[\omega]} = \mathfrak{F}_p(\omega)$
Accessibility sensitivity	Does an earlier operation create or remove the domain of a later operation?	Does the resulting availability, exclusion, or delay affect a criterion selected by the frame?	$D_{\Phi_j \circ \Phi_i}$, $\mathcal{Q}(\omega)$
Path sensitivity	Can different intermediate histories lead to the same coarse endpoint?	Does the frame distinguish the intermediate operation trace, standing, information, participation, or review relations?	$\mathrm{Feat}_\Theta(H)$
Recovery sensitivity	After divergence, which reference features remain reachable through admissible future histories?	Which forms of recovery count as relevant or sufficient under the frame?	$\mathrm{Rep}_{\mathcal{U},\mathcal{R},J}$

Table 11 also shows why the temporal location of a formally identical label can matter. A hearing, disclosure, recognition, or review event can retain the same abstract procedural name while having different effects because its execution configuration has changed. The paper therefore treats procedural timing primarily as position within the modeled sequential trace rather than as clock time alone. Where that trace is extracted from a broader partially ordered event structure, it represents only the chain or linearization relevant to the comparison at hand. A calendar timestamp can remain empirically important, especially for deadlines and delay, while concurrency and incomparable events require additional semantics beyond the present sequential model.

The distinction is especially important for participation. Procedural-justice research has often examined opportunities for voice or process control as components of how procedures are experienced and evaluated [3, 2]. Within the present formalism, the analytical contribution is narrower. A participation opportunity can be represented as occurring at a point where it can still affect classification, valuation, authorization, or conversion, or at a later point where those operations have already been instantiated or settled. The formal model does not infer that earlier participation is invariably superior. It identifies the changed opportunity structure that a procedural theory may then evaluate.

The same caution applies to fair hearing and procedural equality. Ceva's proceduralist account gives these ideas a substantive normative interpretation in the context of political value conflict [1]. The present paper does not import that account as the unique criterion for heterogeneous conversion. It uses fair hearing as one example of a feature that can be represented by a theory-relative frame. Another theory can select neutrality, consistency, accuracy, publicity, accountability, participation, contestability, or a different set of procedural concerns. Definition 7.1 is designed to preserve those differences rather than aggregate them automatically.

A further consequence concerns retrospective procedures. Section 6.2 showed that recognition performed later can refer to an earlier contribution or status while still acting on the configuration present at the time of execution. Section 6.3 then showed that reopening and repair can expand the future corridor without reproducing the earlier history. Under a procedural frame, these later responses can therefore have substantial value while remaining distinguishable from ex ante access. The frame can credit acknowledgment, restored standing, compensation, renewed contestability, or other recovered relations while also recording burdens created by delay or closure.

This structure avoids two opposite reductions. The first reduction treats any order-sensitive pair as procedurally unjust merely because the operations fail to commute. Proposition 7.4 blocks that inference. The second reduction treats order as irrelevant whenever the same visible endpoint is eventually restored. The history-signature construction and repair-corridor formalism preserve procedural differences that can survive endpoint convergence. A procedural judgment must therefore identify both the structural difference and the normative reason for attending to it.

The next three subsections instantiate this general architecture. Section 7.2 examines how standing and admissibility determine who or what can enter a conversion procedure and at which stage. Section 7.3 examines informational conditions and meaningful participation, including the difference between formal opportunity and practical influence. Section 7.4 examines revision and review, with particular attention to finality, reopening, and the recoverability structures developed in Section 6.

7.2 Standing and Admissibility

The general framework in Subsection 7.1 treats accessibility sensitivity as one channel through which ordering can become procedurally material. The present subsection gives that channel a more precise interpretation. Its role is to distinguish standing, authorization, and operation-specific admissibility, and then to show how their ordering can alter which procedural histories are realizable. The analysis proceeds from configuration readouts to partial-map domains and finally to history-level access profiles. This sequence preserves a central distinction: standing is a relational or institutional status represented within a configuration, whereas admissibility is the definedness of a particular operation at that configuration.

Standing is treated as context-sensitive in the present model of heterogeneous conversion. A person, organization, contribution, claim, ecological relation, or historical interest can be recognized for one procedural purpose while remaining outside another. The present note therefore avoids a universal binary variable for standing. For a context c, Definition 7.5 introduces a passive standing readout whose codomain can be chosen to fit the application.

Definition 7.5 (Standing readout)**.** For a procedural context c, a *standing readout* is a passive map of the form stated in Equation 66.

$$\mathrm{st}^{(c)} : \Omega \longrightarrow S_c. \tag{66}$$

The output $\mathrm{st}^{(c)}(\omega)$ records the modeled standing status relevant to context c. The codomain S_c can contain binary, categorical, set-valued, or otherwise structured standing descriptions.

Definition 7.5 follows the observation discipline established in Definition 4.2. The readout reports a feature of the current configuration and leaves the configuration unchanged. A recognition update U_R, by contrast, can modify $\mathcal{B}_\omega$, $\mathcal{N}_\omega$, or both, and can thereby change the standing subsequently reported by $\mathrm{st}^{(c)}$. This separation allows the manuscript to represent a transition in standing without treating the act of reading that status as the act that creates it.

Admissibility concerns a different question. For an operation type p with induced update $U_p : \Omega \rightharpoonup \Omega$, the relevant structural fact is whether the current configuration belongs to D_{U_p}. Definition 7.6 packages this domain information into a readout that can later be incorporated into a procedural interpretive frame.

Definition 7.6 (Operation-specific admissibility indicator)**.** For an induced update $U_p : \Omega \rightharpoonup \Omega$, define the operation-specific admissibility indicator $a_p : \Omega \to \{0, 1\}$ by Equation 67.

$$a_p(\omega) = \begin{cases} 1, & \omega \in D_{U_p}, \\ 0, & \omega \notin D_{U_p}. \end{cases} \tag{67}$$

Equation 67 converts the definedness condition from Definition 3.2 into a passive configuration readout. It adds no new institutional criterion. Every reason that an application places inside the domain of U_p remains available: standing, authorization, jurisdiction, evidentiary sufficiency, deadlines, settlement status, capacity, or another modeled condition. The indicator therefore records executability while leaving the source of that executability explicit in the underlying model.

The distinction between Definitions 7.5 and 7.6 prevents standing from being identified mechanically with procedural access. Two configurations can report the same standing status while differing in the admissibility of a particular operation because authorization, timing, evidence, or settlement differs. Conversely, an institution can create an exceptional access route for a particular operation while leaving a broader standing classification unchanged. A procedural theory can attend to either difference, or to both, through the history features selected in $\mathcal{E}_\Theta$.

Authorization supplies a related but separate layer. The operation label U in Definition 2.1 denotes an authorization event that can modify the configuration and thereby change the domain of a later operation. Authorization can depend on standing, and recognition can alter authorization conditions, while the concepts remain analytically separable. A claimant can possess standing to participate in a process while a proposed transfer remains unauthorized; an authorized institutional action can likewise proceed under a rule whose relevant standing relation concerns an office, representative, or recognized interest rather than the ultimate beneficiary. The partial-domain formalism accommodates these combinations without assigning one canonical hierarchy among them.

For ordering analysis, the important question is how admissibility changes after an earlier operation. Definition 7.7 identifies domain-opening and domain-closing updates relative to a target procedure.

Definition 7.7 (Domain-opening and domain-closing update)**.** Let $U_g, U_p : \Omega \rightharpoonup \Omega$, and suppose $\omega \in D_{U_g}$. The update U_g is *domain-opening for U_p at ω* when Equation 68 holds.

$$a_p(\omega) = 0, \qquad a_p\big(U_g(\omega)\big) = 1. \tag{68}$$

The update U_g is *domain-closing for U_p at ω* when Equation 69 holds.

$$a_p(\omega) = 1, \qquad a_p\big(U_g(\omega)\big) = 0. \tag{69}$$

Definition 7.7 is deliberately local. The same operation type can open a domain in one configuration, preserve it in another, and close it in a third because the concrete transformation and institutional conditions are state-dependent. Recognition can be domain-opening when it creates the standing required for negotiation or review. Authorization can be domain-opening when it makes transfer executable. Settlement can be domain-closing when it removes an ordinary revision route. Reopening from Definition 6.14 is a specialized multi-step construction that can restore access after such closure.

The opening case has an immediate compositional consequence. Proposition 7.8 connects Definition 7.7 to the one-way composability relation introduced earlier.

Proposition 7.8 (Domain opening induces local precedence dependency)**.** *Let $U_g, U_p : \Omega \rightharpoonup \Omega$. If U_g is domain-opening for U_p at ω, then the ordering $g \to p$ is one-way composable relative to $p \to g$ at ω. Equivalently, Equation 70 holds.*

$$U_g \prec_\omega^{\mathrm{dep}} U_p. \tag{70}$$

Proof. Domain opening requires $\omega \in D_{U_g}$, $a_p(\omega) = 0$, and $a_p(U_g(\omega)) = 1$. By Definition 7.6, the second condition gives $\omega \notin D_{U_p}$, while the third gives $U_g(\omega) \in D_{U_p}$. Hence $\omega \in D_{U_p \circ U_g}$. The reverse composite $U_g \circ U_p$ is undefined at ω because U_p is undefined there. Definition 3.8 and Equation 15 therefore yield Equation 70. □

Proposition 7.8 formalizes one route through which recognition can become procedurally consequential. Suppose U_R changes the standing reported by $\mathrm{st}^{(c)}$ and is domain-opening for negotiation, valuation, review, or another target operation. The resulting ordering has a local precedence dependency: the target procedure becomes realizable after recognition while the reverse two-step history is unavailable from the same initial configuration. The formalism records the change in standing and the change in admissibility separately. An application that attributes the domain change specifically to standing should additionally encode or establish that institutional dependency rather than infer causation from co-variation alone.

The same discipline applies when recognition occurs after a consequential operation. Delayed recognition from Subsection 6.2 can establish standing in the current configuration while leaving earlier operation domains historically inaccessible. A person or claim can therefore possess present standing even though a valuation, negotiation, authorization, or conversion stage that would have been available under earlier recognition has already passed. Section 6.3 showed that subsequent reopening and repair can expand the future corridor. Those later transformations can restore selected access conditions while preserving a distinct realized history.

A history-level readout is needed to represent this timing directly. Let $H = (\mathbf{p}; \omega_0, \ldots, \omega_n)$ be a realized history. Equation 71 defines paired standing and admissibility trajectories along its state path.

$$\begin{aligned} \mathrm{St}^{(c)}(H) &= \big(\mathrm{st}^{(c)}(\omega_0), \ldots, \mathrm{st}^{(c)}(\omega_n)\big), \\ \mathrm{Adm}_p(H) &= \big(a_p(\omega_0), \ldots, a_p(\omega_n)\big). \end{aligned} \tag{71}$$

Equation 71 preserves when standing and operation-specific access appear, disappear, or remain stable along the realized path. Either trajectory can be included among the history features in $\mathcal{E}_\Theta$. A frame can therefore distinguish histories that share the same terminal standing and terminal admissibility while reaching those terminal conditions through different periods of exclusion, delay, or access.

This history-sensitive representation is important when an operation has influence only before another event. Consider a negotiation operation N whose domain becomes available after recognition. If recognition occurs before valuation or conversion, the resulting admissibility trajectory can include a negotiation opportunity while relevant terms remain open. If recognition occurs after settlement, a later reopening path can make contestation or renegotiation executable, yet the resulting history contains a different access trajectory. The framework describes that difference without assuming that every early opportunity is superior to every retrospective remedy. The procedural frame supplies the criteria for evaluating the timing, burdens, institutional reasons, and available forms of recovery.

Table 12 shows why a single statement that a participant "had standing" can be too coarse for order-sensitive conversion. The status can be present while a particular procedure remains unavailable, and a later status change can open a procedure only after another operation has already altered the configuration. Conversely, a domain restriction can be procedurally justified under a supplied theory because the operation requires jurisdiction, evidentiary sufficiency, protection of other parties, or another admissibility condition. The formalism therefore exposes the gate and its timing while leaving justification to the interpretive frame.

This separation also clarifies equality claims. Equal terminal standing does not imply equal procedural histories when $\mathrm{St}^{(c)}(H)$ differs across the relevant period. Equal standing trajectories do not imply equal access when $\mathrm{Adm}_p(H)$ differs. Equal access to one operation does not imply equal access to every operation in the conversion sequence. A frame concerned with procedural equality must consequently specify the status, operation, stage, and comparison relation that it evaluates. Definition 7.1 provides the place for those choices without embedding them into the structural model.

The distinction carries over to repair. A repair corridor can restore standing, reopen a review domain, or create a new compensatory route while leaving the earlier admissibility trajectory unchanged as historical fact. A procedural frame can value the restored relation and still register the burdens associated with earlier exclusion or delayed access. This possibility is one reason the repair corridor in Definition 6.17 is defined over realized future histories rather than terminal states alone.

Standing and admissibility therefore supply a first substantive interpretation of accessibility sensitivity. Partial domains identify which operations can occur; standing readouts identify one class of relational statuses that can help condition those domains; recognition and authorization can change the configuration from which later domains are evaluated; and history-level trajectories record when access existed. The next subsection extends the same order-sensitive analysis to information and meaningful participation, where formal access to a procedure can remain distinct from the informational conditions under which participation can affect the subsequent conversion history.

Table 12: Distinct layers of standing and procedural admissibility in the ordering analysis.

Layer	Formal representation	Structural question	Possible procedural significance under Θ
Standing status	$\mathrm{st}^{(c)}(\omega)$	Which status, claim, relation, or eligibility is currently recognized in context c?	Equality of standing, acknowledgment, entitlement to be heard, or another theory-selected relation.
Authorization	Operation type U and its induced update	Has an institutional act changed whether a contemplated later operation is permitted or effective?	Control over permission, institutional competence, reason-giving, or accountability for access decisions.
Operation admissibility	$a_p(\omega)$ and D_{U_p}	Is the specific target operation executable from the current configuration?	Access, exclusion, delay, or differential procedural opportunity.
Access trajectory	$\mathrm{Adm}_p(H)$	At which points in the realized history is the operation executable?	Timing of access relative to valuation, conversion, settlement, or other consequential events.
Standing trajectory	$\mathrm{St}^{(c)}(H)$	How does modeled standing change along the realized history?	Persistence, delayed recognition, temporary exclusion, or retrospective acknowledgment.

7.3 Information and Meaningful Participation

Subsection 7.2 separated a participant's standing from the admissibility of a particular operation. The present subsection adds two further dimensions: the information available when participation occurs and the set of later operations that participation can still affect. Its role is to distinguish formal access to a participatory channel from participation situated within an information and influence structure. The analysis proceeds through information readouts, participation repertoires, and history-level influence windows. The construction remains structural. It identifies when participation can enter an order-sensitive conversion process and which later operations remain responsive to it, while leaving the normative adequacy of information, voice, influence, and burden to the procedural interpretive frame introduced in Definition 7.1.

Procedural-justice research has long distinguished control over presenting one's case from control over the decision itself, and later research has treated voice and participation as important features of how procedures are perceived and evaluated [3, 2]. The present formalism uses that literature only to motivate a narrower analytical question. A participant can possess an opportunity to speak or submit material while the information required to understand the conversion is incomplete, while the relevant classification or valuation has already been fixed, or while the only remaining target is a later review procedure. A statement that participation "occurred" therefore does not by itself identify what information accompanied it or which subsequent transformations remained open to influence.

Information is represented first as a passive readout. The codomain is left application-specific because information relevant to a wage negotiation, environmental valuation, attribution dispute, or historical claim can differ substantially in type and granularity. Definition 7.9 introduces this object.

Definition 7.9 (Procedural information readout)**.** For an informational context c, a *procedural information readout* is a passive map of the form stated in Equation 72.

$$\mathrm{inf}^{(c)} : \Omega \longrightarrow I_c. \tag{72}$$

The output $\mathrm{inf}^{(c)}(\omega)$ records the information state selected for analysis in context c.

Definition 7.9 does not define informational sufficiency. The codomain I_c can encode disclosed records, uncertainty ranges, reasons, model outputs, valuation assumptions, translated materials, or another context-specific informational state. Whether one state is adequate, intelligible, timely, or fairly distributed is a judgment supplied through the readouts and preorder of Θ. This separation prevents the structural model from embedding a universal informational standard.

The history-level analogue records when information becomes available. For a realized history $H = (\mathbf{p}; \omega_0, \dots, \omega_n)$, Equation 73 defines the information trajectory along its state path.

$$\mathrm{Inf}^{(c)}(H) = \big(\mathrm{inf}^{(c)}(\omega_0), \dots, \mathrm{inf}^{(c)}(\omega_n)\big). \tag{73}$$

Equation 73 allows two histories with the same terminal disclosure to remain distinguishable when the disclosure occurred at different procedural positions. The readout $\mathrm{inf}^{(c)}$ is passive, so it does not itself alter valuation or any other operation. When disclosure, evidence submission, or another state-changing update changes the information represented in ω, information available before valuation can form part of the configuration from which valuation is instantiated; comparable disclosure after conversion or settlement can instead bear on acknowledgment, review, or repair. The ordering difference therefore concerns the underlying state-changing informational event, while $\mathrm{Inf}^{(c)}(H)$ records its informational consequence.

Participation requires an equally careful representation. The procedural alphabet in Definition 2.1 already contains operations through which participation can occur, including negotiation and contestation. The paper therefore does not add a universal participation primitive. For a context c, let $\mathcal{U}^{(c)}_{\mathrm{part}} \subseteq \mathrm{PEnd}(\Omega)$ denote a selected repertoire of partial updates through which participants can submit reasons, proposals, evidence, objections, or other procedurally recognized contributions. Depending on the application, the repertoire can contain negotiation updates, review submissions, recognition requests, or other existing institutional operations.

The structural significance of participation depends partly on what later operation remains responsive to those contributions. Let $t \in \mathcal{L}_P$ denote a target operation type. The target can be classification, valuation, authorization, conversion, settlement, revision, or another modeled operation. Definition 7.10 packages three features of the target: whether the induced update is currently admissible, which concrete transformation its state-dependent schema selects, and which immediate configuration the target produces when executed.

Definition 7.10 (Target-response descriptor)**.** For an operation type $t \in \mathcal{L}_P$, let $\perp_t$ be a fixed symbol denoting current unavailability of the target operation. Define the target-response descriptor χ_t by Equation 74. When the target is executable, the descriptor records both the selected concrete transformation and the immediate configuration produced by executing that target from the current state.

$$\chi_t(\omega) = \begin{cases} (0, \perp_t, \perp_t), & \omega \notin D_{U_t}, \\ (1, \mathfrak{F}_t(\omega), U_t(\omega)), & \omega \in D_{U_t}. \end{cases} \tag{74}$$

Equation 74 records the concrete target transformation and its immediate realized output only when the target is currently executable. Two configurations at which the target is unavailable therefore receive the same unavailable descriptor rather than being classified as participation-sensitive merely because the total schema $\mathfrak{F}_t$ selects different partial maps off the executable domain. A participatory intervention can matter by changing target admissibility, changing the transformation selected on an admissible state, changing the target's realized output under an otherwise unchanged transformation, or some combination of these. Equality of descriptors is intentionally strong; an application can replace χ_t with a coarser feature when only selected aspects of target response are empirically observable or normatively relevant.

Definition 7.11 uses the descriptor to identify configurations in which different admissible participatory paths can still make a structural difference to the target operation.

Definition 7.11 (Participation-sensitive configuration)**.** Let $\mathcal{U}^{(c)}_{\mathrm{part}}$ be a participation repertoire and let $t \in \mathcal{L}_P$ be a target operation type. A configuration ω is *participation-sensitive for* (c, t) when there exist two composites $\Psi_1, \Psi_2 \in \mathrm{Comp}(\mathcal{U}^{(c)}_{\text{part}})$, both defined at ω, such that Equation 75 holds.

$$\chi_t(\Psi_1(\omega)) \neq \chi_t(\Psi_2(\omega)) \,. \tag{75}$$

Definition 7.11 expresses potential influence rather than guaranteed success. The definition requires only that different admissible participatory histories can lead to different target-response descriptors. It does not require the participant to obtain a preferred outcome, and it does not imply that every available contribution must be accepted. The criterion is therefore compatible with procedures in which participants can affect the informational or justificatory basis of a decision while the institution retains decision authority.

The definition also distinguishes participation from symbolic presence. A hearing or negotiation can be formally available while all relevant target operations have already become insensitive to the available participatory repertoire. Such participation can still have acknowledgment, expressive, evidentiary, or retrospective significance under a procedural frame. The structural point is narrower: its capacity to alter the specified target operation differs from participation occurring while that target remains responsive.

The timing of this capacity can be represented along a realized history. Definition 7.12 introduces the corresponding influence window.

Definition 7.12 (Participation influence window)**.** Let $H = (\mathbf{p}; \omega_0, \ldots, \omega_n)$ be a realized history. The *participation influence window* for context c and target operation t is the index set in Equation 76.

$$\mathcal{W}_{c,t}(H) = \{i \in \{0, \ldots, n\} \mid \omega_i \text{ is participation-sensitive for } (c,t)\} \,. \tag{76}$$

Definition 7.12 converts the intuitive language of participation "before the decision is fixed" into a history-relative structural object. The window can be empty, contiguous over a range of history indices, intermittent, or reopened after an earlier closure. A recognition update can create standing and thereby open a participation channel; a valuation or settlement update can narrow the set of target transformations still responsive to that channel; a later review procedure can create a new influence window directed at revision or repair rather than at the original conversion operation.

The information trajectory in Equation 73 and the influence window in Equation 76 should be interpreted together. A history can contain a broad influence window but limited information during that period. Another can contain extensive disclosure only after the original influence window has closed. A third can reopen participation after settlement with improved information but a different target operation. These histories can have the same terminal disclosure and the same formal presence of participation while differing substantially in their order-sensitive procedural structure.

A procedural frame can represent those differences without adopting a universal definition of meaningful participation. For example, $\mathcal{E}_\Theta$ can include feature maps returning $\mathrm{Inf}^{(c)}(H)$, the standing trajectory from Equation 71, and $\mathcal{W}_{c,t}(H)$. The preorder $\succeq_\Theta$ can then assess timeliness, intelligibility, opportunity, influence, burden, equality, or other theory-relevant dimensions. A different frame can assign different significance to the same structural history. Meaningfulness is therefore an interpretive conclusion grounded in explicit features rather than a primitive label generated by the formalism.

This construction clarifies several ordering patterns. Participation before classification can alter which category becomes the input to valuation. Participation after classification but before valuation can leave classification fixed while still changing valuation inputs. Participation after conversion can shift its primary target toward recognition, settlement, review, or repair. Participation after a reopening path can become consequential again, while the reopened window remains historically distinct from the earlier opportunity. None of these orderings is assigned a universal ranking by the model. Their procedural significance depends on the target operation, the information available, the available alternatives, and the evaluative frame.

The distinction is also relevant to retrospective recognition. A participant whose contribution becomes recognized only after conversion can receive standing and gain access to a new participatory channel. The resulting influence window may concern revision or repair rather than the conversion operation that would have been sensitive to participation earlier. The later opportunity can therefore be substantial without becoming structurally identical to ex ante participation. This is the participation analogue of the delayed-recognition distinction developed in Subsection 6.2.

Table 13 makes the structural separation explicit. Formal opportunity, information, and influence need not coincide. A procedure can provide access with limited information, disclosure after the main target has become insensitive, or a later reopened opportunity aimed at a different target. The procedural interpretive frame determines which of these differences count as salient and how they are compared.

Information and participation therefore extend the ordering analysis beyond simple access. Standing can open a procedure, information can shape the configuration from which participation occurs, and participation can alter a later operation only within an application-specific influence structure. Once a conversion or settlement narrows that structure, subsequent participation may operate through revision and review. The next subsection examines those mechanisms directly and distinguishes immediate review, reopenable review, and closure relative to a specified future repertoire.

Table 13: Information and participation dimensions in an order-sensitive conversion history.

Dimension	Formal representation	Structural question	Theory-relative interpretation
Information state	$\inf^{(c)}(\omega)$	What selected information is available at the current configuration?	Adequacy, intelligibility, disclosure, uncertainty, or informational equality.
Information trajectory	$\mathrm{Inf}^{(c)}(H)$	When does the selected information become available along the realized history?	Timeliness, delay, sequencing of reasons and evidence, or burden generated by late disclosure.
Participation repertoire	$\mathcal{U}^{(c)}_{\text{part}}$	Which institutionally recognized contributions can alter the configuration?	Voice, hearing, negotiation, representation, evidentiary contribution, or another selected mode.
Target responsiveness	$\chi_t(\omega)$	Can admissible participatory paths change target admissibility, target instantiation, or the immediate target result?	Potential influence, decision openness, responsiveness, or the limits of participant control.
Influence window	$\mathcal{W}_{c,t}(H)$	At which history positions does participatory variation remain structurally consequential for target t?	Timing of meaningful opportunity, closure, reopening, or retrospective participation.

7.4 Revision and Review

The preceding subsection identified how participation can lose or regain influence as an order-sensitive procedure develops. The present subsection turns to the institutional mechanisms through which an earlier classification, valuation, authorization, conversion, or settlement can be reconsidered. Its role is to distinguish review access, review scope, reopening, and repertoire-relative closure. The analysis combines the review-access set $\mathcal{Q}(\omega)$ from Definition 4.5 with the reachability construction from Section 6.1. This combination permits a more precise discussion of finality and contestability while preserving the paper's theory-relative approach to their normative evaluation.

Review access is already represented by $\mathcal{Q}(\omega)$, and Assumption 4.6 connects membership in that set to the domain of the corresponding review update. Direct accessibility, however, is only one possible status. A procedure can be unavailable at the current configuration while remaining reachable after an appeal gateway, recognition event, jurisdictional reassignment, or another reopening update. A second procedure can be unavailable both directly and through every update in the repertoire chosen for the analysis. The reachability framework makes those cases distinguishable.

Let $q \in \mathcal{L}_Q$ be a concrete review procedure and let $\mathcal{U} \subseteq \mathrm{PEnd}(\Omega)$ be a selected future update repertoire. Definition 7.13 collects the configurations reachable from ω at which q is available.

Definition 7.13 (Review-reach set)**.** The *review-reach set* for q relative to $\mathcal{U}$ is defined by Equation 77.

$$\mathrm{RevReach}_{q,\mathcal{U}}(\omega) = \left\{\omega' \in \mathrm{Reach}_{\mathcal{U}}(\omega) \;\middle|\; q \in \mathcal{Q}(\omega')\right\}. \tag{77}$$

Definition 7.13 is repertoire-relative in the same sense as the irreversibility definitions in Section 6.1. Expanding the modeled repertoire can turn a closed review route into a reopenable one; restricting the repertoire can remove a path that exists institutionally but lies outside the analysis. The construction therefore describes review accessibility under stated assumptions rather than an absolute property of the institution.

Three structurally different access statuses follow. Definition 7.14 names them.

Definition 7.14 (Review-access status)**.** For $q \in \mathcal{L}_Q$, configuration ω, and repertoire $\mathcal{U}$, the review procedure has:

1. *direct access* when $q \in \mathcal{Q}(\omega)$;
2. *mediated access* when $q \notin \mathcal{Q}(\omega)$ and $\mathrm{RevReach}_{q,\mathcal{U}}(\omega) \neq \varnothing$;
3. *repertoire-relative closure* when $\mathrm{RevReach}_{q,\mathcal{U}}(\omega) = \varnothing$.

Direct and mediated access can impose different procedural burdens even when both eventually lead to the same review forum. Mediated access requires an additional history before q becomes executable. That history can involve delay, additional proof, standing recognition, cost, authorization, or another institutional transformation. The structural distinction is therefore retained even when a procedural frame ultimately regards the two routes as equivalent.

Definition 7.14 also gives the reopening concept from Definition 6.14 a reachability characterization. Proposition 7.15 states the relation for a repertoire chosen specifically for reopening.

Proposition 7.15 (Mediated access and reopening)**.** *Let $q \notin \mathcal{Q}(\omega)$, and let $\mathcal{U}_{\text{open}}$ be a repertoire of updates capable of altering review access. The procedure q has mediated access relative to $\mathcal{U}_{\text{open}}$ if and only if at least one reopening path for q from ω exists in the sense of Definition 6.14.*

Proof. If q has mediated access, Definition 7.14 and Equation 77 give a reachable configuration ω' with $q \in \mathcal{Q}(\omega')$. By the definition of $\mathrm{Reach}_{\mathcal{U}_{\text{open}}}(\omega)$, there exists a composite $\Psi_{\text{open}} \in \mathrm{Comp}(\mathcal{U}_{\text{open}})$ defined at ω such that $\Psi_{\text{open}}(\omega) = \omega'$. Together with $q \notin \mathcal{Q}(\omega)$, these conditions satisfy Equation 59. Conversely, a reopening path produces a reachable configuration at which q belongs to the review-access set, so Equation 77 is nonempty and mediated access follows. □

Proposition 7.15 separates two questions that can be merged in ordinary descriptions of appeal. The first is whether review is presently available. The second is whether a route exists through which it can become available. A procedure can therefore remain indirectly contestable even when the contested configuration is not itself inside the immediate domain of the review update.

Access alone still does not determine what review can revisit. A review procedure can be competent to reconsider classification while leaving conversion final, or it can reopen valuation without disturbing attribution. Another can authorize a broader remedial transformation. Definition 7.16 represents this second structural dimension.

Definition 7.16 (Review-scope readout)**.** For a review procedure $q \in \mathcal{L}_Q$, let the *review-scope readout* be the map in Equation 78.

$$\mathrm{scope}_q : \Omega \longrightarrow 2^{\mathcal{L}_P}. \tag{78}$$

The set $\mathrm{scope}_q(\omega)$ contains the modeled operation types whose realized effects or institutional determinations are within the competence of q to reconsider at configuration ω.

Definition 7.16 is a coarse institutional representation. Because the readout is defined on all modeled configurations, $\mathrm{scope}_q(\omega)$ should be interpreted as the modeled jurisdictional or remedial scope associated with q at ω, whether or not q is directly accessible there; its operative use still depends on access or reopening. Concrete applications can instead make scope partial if latent scope has no institutional meaning. Concrete applications can also refine its codomain to distinguish standards of review, available remedies, evidentiary limits, or the portions of an earlier determination that remain revisable. The present representation imposes no functional dependence between review access and review scope. It can therefore represent a procedure that remains directly available while its scope narrows after settlement, a reopening path that restores access while preserving restricted remedial competence, or a new review route with broader scope than a route that previously closed. Whether such combinations occur in a particular institution is an empirical question.

History-level review can now be represented through both access and scope trajectories. For a realized history $H = (\mathbf{p}; \omega_0, \ldots, \omega_n)$, Equation 79 defines these objects along its state path.

$$
\begin{aligned}
\mathrm{QTraj}(H) &= \big(\mathcal{Q}(\omega_0), \ldots, \mathcal{Q}(\omega_n)\big), \\
\mathrm{Sc}_q(H) &= \big(\mathrm{scope}_q(\omega_0), \ldots, \mathrm{scope}_q(\omega_n)\big).
\end{aligned}
\tag{79}
$$

Equation 79 allows a procedural frame to distinguish histories that end with the same currently available review route while differing in earlier closure, reopening burden, or remedial scope. It also allows review to be evaluated together with standing, information, and participation rather than as an isolated terminal entitlement.

The resulting structure sharpens the language of finality. Repertoire-relative closure in Definition 7.14 is a structural condition: under the specified future update repertoire, no reachable configuration makes q available. Institutional or normative finality can be broader or narrower than this condition. A regime can treat a determination as final for ordinary purposes while retaining extraordinary reopening mechanisms. Another can preserve continuing review access while assigning strong reliance interests to an existing settlement. The formalism therefore uses closure, access, and scope as descriptive objects and leaves the justification of finality to Θ.

The same restraint applies to contestability. More review access does not automatically imply a procedurally superior arrangement. Repeated review can impose delay, cost, uncertainty, strategic burden, or instability on other participants. Conversely, early closure can prevent correction of misclassification, exclusion, or newly available evidence. A procedural theory can assign weight to finality, reliance, accuracy, equality, participation, non-domination, institutional capacity, or other considerations. Definition 7.1 permits those considerations to remain explicit and potentially incomparable rather than encoding a universal preference for maximal openness.

The repair framework from Section 6.3 supplies a final distinction. Review can be one route into a repair corridor, but review and repair are not coextensive. A reopening path can restore access to q while the subsequent review leaves all selected repair targets unmatched. A compensatory or recognition update can enter a repair corridor without using formal review. A review decision can also alter the set of future repair histories by changing standing, scope, authorization, or settlement status. These possibilities reinforce the paper's compositional approach: contestation is one operation inside a larger future history rather than an automatic synonym for correction.

A procedurally material ordering can therefore arise at several review-related locations. Review before settlement can operate on an unsettled configuration and possess one scope. Review after settlement can require mediated access and possess another scope. Reopening after delayed recognition can make a previously unavailable claim reviewable while preserving a different historical path from ex ante access. Two cases can reach the same visible settlement after review while differing in $\mathrm{QTraj}(H)$, $\mathrm{Sc}_q(H)$, participation history, and the repair corridor that remained available at each stage.

Table 14 summarizes the distinction between current access, reopenability, closure, scope, and repair. These dimensions can vary independently. Their separation helps prevent finality from being identified mechanically with the absence of immediate review and prevents contestability from being identified mechanically with successful repair.

Section 7 can now be summarized in one structural sequence. Ordering becomes procedurally material only under a supplied interpretive frame. Standing and admissibility determine which actors, claims, and operations can enter a procedure and when. Information and participation determine what selected information accompanies participation and which later operations remain responsive to participant-generated variation. Revision and review determine whether earlier determinations are directly contestable, reopenable through additional histories, or closed relative to a stated future repertoire, and they identify the scope of what an available review can revisit. Reopening and repair then describe possible responses to earlier closure or divergence without converting those structural possibilities into a universal justice ranking.

The resulting framework is deliberately plural. A procedural theory can emphasize standing, equality of hearing, information, responsiveness, review, finality, repair, or another criterion; it can combine

Table 14: Review, closure, and repair dimensions in the procedural ordering framework.

Dimension	Formal representation	Structural question	Interpretive status
Direct review access	$q \in \mathcal{Q}(\omega)$	Is the review procedure executable from the current configuration?	Immediate contestability; normative adequacy depends on the frame.
Mediated review access	$\mathrm{RevReach}_{q,\mathcal{U}}(\omega) \neq \varnothing$ with no current access	Can an admissible future path reopen the procedure?	Contestability with an additional procedural burden or gateway.
Repertoire-relative closure	$\mathrm{RevReach}_{q,\mathcal{U}}(\omega) = \varnothing$	Does the modeled future repertoire contain any path to review access?	Structural closure under stated assumptions; it does not itself establish justified finality.
Review scope	$\mathrm{scope}_q(\omega)$	Which earlier operation types can the available review procedure reconsider?	Breadth of contestation, remedial competence, and limits of review.
Repair relation	$\mathrm{Rep}_{\mathcal{U},\mathcal{R},J}(\omega; \omega_{\mathrm{ref}})$	Which admissible future histories restore selected relational or procedural targets?	Structural recovery; rectificatory adequacy remains theory-relative.

several dimensions or leave some histories incomparable. The order-sensitive formalism contributes a diagnostic question to each theory: does changing the order alter the status, access, information, influence, review path, or recoverability that the theory regards as procedurally relevant? Section 8 applies this question to a small set of heterogeneous conversion cases without treating the examples as exhaustive empirical models.

8 Illustrative Conversion Cases

The preceding sections developed the paper's structural and interpretive machinery in abstract form. The present section uses four stylized conversion cases to show how the same machinery can organize practical reasoning without turning the examples into empirical case studies or complete normative analyses. Each subsection isolates a limited ordering problem, identifies the structural channel through which the order can matter, and indicates how a procedural interpretive frame could evaluate the resulting histories. The examples are deliberately schematic. They do not assume that the named institutions operate identically across jurisdictions, and they do not infer a justice verdict from the formal divergence alone.

The cases also serve a second purpose. Papers on heterogeneous value conversion can easily move from an abstract sequence such as recognition–valuation–conversion to a substantive conclusion about wages, compensation, intellectual contribution, or historical claims without making the intervening assumptions explicit. The present illustrations instead keep the operations visible. Recognition, classification, attribution, valuation, authorization, negotiation, transfer, settlement, and revision retain the meanings stated in Table 1. The examples then ask what becomes available, unavailable, or differently instantiated when the order changes.

Across all four cases, three interpretive cautions remain in force. First, a comparison history functions as a reference configuration rather than an automatically preferred history. Second, a later response can be valuable without reproducing the earlier path. Third, restoration of one selected dimension does not entail restoration of every relation affected by the conversion. These cautions are especially important in environmental and historical settings, where the objects entering the procedure can exceed the dimensions represented by a monetary or administrative settlement.

8.1 Labor Contribution and Compensation

The labor case illustrates how recognition, classification, attribution, negotiation, and settlement can interact before compensation becomes institutionally stabilized. The example concerns a stylized work relation in which a contribution is initially present in the relational history but its procedural status is not yet fixed. The relevant issue is not the substantive level of compensation. The issue is whether the contribution is recognized and classified at a stage when those operations can still influence attribution, valuation, negotiation, and the conversion that produces compensation.

Consider two procedural traces that contain comparable identification, evaluation, and settlement operations but place recognition at different positions. Equation 80 records the contrast.

$$\mathbf{p}_L^{\text{early}} = (R, K, A, V, N, U, X, S), \qquad \mathbf{p}_L^{\text{late}} = (K, A, V, N, U, X, S, R). \tag{80}$$

Equation 80 is schematic rather than canonical. A real institution can combine several operations, repeat valuation, or place authorization elsewhere. The comparison isolates one feature: in the early-recognition trace, recognition occurs before classification, attribution, valuation, negotiation, and settlement; in the late-recognition trace, recognition occurs after the compensation procedure has already been settled.

The first structural question concerns admissibility. Suppose recognition creates standing for a negotiation channel or changes the class of claims that can be entered into valuation. In the language of Definition 7.7, the recognition update can be domain-opening for negotiation or for a contribution-specific valuation procedure. Recognition then induces a local precedence dependency of the type established in Proposition 7.8. The late-recognition trace need not contain an equivalent negotiation opportunity because the relevant negotiation update may already have been executed under a different standing or classification state.

The second question concerns state-dependent instantiation. Even when valuation and conversion remain admissible in both traces, the concrete transformations selected by their schemas can differ. Recognition or attribution before valuation can change the configuration from which $\mathfrak{F}_V$ is evaluated. Recognition before conversion can likewise change the conversion instantiation under the mechanism analyzed in Section 5. The formal point is limited: an operation called "valuation" or "compensation" can be present in both histories while acting on different identified entities, claim structures, or attributed contributions.

Participation adds a further distinction. If negotiation belongs to the participation repertoire $\mathcal{U}_{\text{part}}^{(L)}$, the early history can contain a participation influence window for valuation or conversion after recognition has created standing. In the late history, a negotiation-like opportunity offered only after settlement can instead target review, acknowledgment, or repair. The two opportunities can therefore be institutionally meaningful while having different target-response descriptors in the sense of Definition 7.10. A terminal record that merely states that the contributor was eventually heard would suppress this difference.

Settlement makes the temporal distinction sharper. Suppose the late recognition event occurs at a configuration of the form ω_{XS}, where settlement has already closed the direct recognition-sensitive compensation route described in Subsection 6.2. If the conditions of Proposition 6.13 hold, recognition can remain referentially directed toward the earlier contribution while no longer entering the original compensation procedure directly. A later response then requires some additional path, potentially including reopening and review.

The repair framework prevents the analysis from collapsing into an all-or-nothing question. A later review can restore selected dimensions such as formal attribution, current compensation, eligibility for a future benefit, or access to a revised classification. Another dimension, such as participation in the original negotiation or the opportunity to use compensation at an earlier time, may remain historically different. The restoration profile from Definition 6.9 can record these differences without declaring one repair package sufficient.

A procedural interpretive frame can therefore evaluate several distinct features of the two histories: the timing of standing, the classification under which the contribution was valued, the information available during negotiation, the existence of an influence window before settlement, the burden of reopening, and the dimensions recovered through repair. Nothing in the compositional formalism determines how these features should be weighted. The example shows why a wage or compensation endpoint can be procedurally incomplete as a representation of the history that produced it.

8.2 Environmental Compensation

Environmental compensation provides a second ordering problem because the set of relations recognized as affected can itself change during the compensation process. The stylized case concerns a procedure that assigns some compensatory response to an environmental loss or alteration. The analysis does not assume that ecological, cultural, livelihood, or historical values are fully commensurable with a monetary measure. The narrower question is how recognition of affected relations interacts with valuation, conversion, settlement, and later repair.

In this subsection, the broad recognition label R denotes recognition directed toward an affected relation that is procedurally absent or weakly represented at the beginning of the comparison. The operation can concern a community relation, an ecological dependency, a use relation, a historical connection, or another feature admitted by the application. Equation 81 contrasts recognition before valuation with recognition after settlement.

$$\mathbf{p}_E^{\text{recognized}} = (R, K, V, N, U, X, S), \qquad \mathbf{p}_E^{\text{retrospective}} = (K, V, N, U, X, S, R, Q). \tag{81}$$

The additional revision operation in the second trace of Equation 81 is intentional. Once recognition occurs after settlement, the response problem is no longer a simple permutation of the original compensation sequence. The later history contains an additional institutional task: determining whether and how the settled procedure can be reconsidered.

Recognition before valuation can alter the object that valuation represents. If the affected-relation recognition modifies $\mathcal{B}_\omega$ or the effective conversion nomos $\mathcal{N}_\omega$, the valuation schema can select a different transformation from the one instantiated before that relation entered the procedure. This need not mean that the newly recognized relation receives a scalar price. The valuation output can be multidimensional, categorical, interval-valued, or linked to a non-monetary response. The paper's formalism requires only that the recognition event can change the configuration from which the later operation is selected.

Recognition after settlement raises a different problem. A monetary settlement can remain valid under its original rules while a later procedure comes to recognize an affected relation that those rules did not previously represent. The delayed recognition is performed on the later configuration, in accordance with the execution distinction developed in Subsection 6.2. It can change present standing, present information, or the scope of review without being historically identical to recognition before valuation.

The environmental example also makes exact return a particularly poor default model of remedy. Some changes represented in the configuration can be repertoire-relative irreversible under Definition 6.5. Even when physical restoration is feasible, the complete relational configuration can include elapsed time, lost opportunities, changed dependencies, third-party adjustments, or a settlement history that remains part of the modeled past. The relevant institutional question can therefore concern reachable repair rather than literal reconstruction of a prior configuration.

A repair corridor is useful because different response paths can restore different selected properties. One path can restore access to an environmental resource. Another can provide monetary compensation. A third can restore procedural standing, monitoring rights, or a review channel. A fourth can combine several responses while leaving other losses outside the selected target family. Let J_E denote the readout indices selected for a particular environmental repair inquiry. A nonempty corridor $\mathrm{Rep}_{\mathcal{U},\mathcal{R},J_E}(\omega;\omega_{\text{ref}})$ establishes the existence of at least one admissible future history restoring those selected readouts. It does not establish equivalence with the reference history on unselected dimensions.

This distinction also protects the analysis against an implicit commensurability assumption. Monetary compensation can be one conversion operation inside a broader repair history while the procedural frame separately records ecological access, recognition, participation, monitoring, cultural relation, or another application-specific dimension. The framework therefore permits monetary and non-monetary responses to coexist without forcing them into a common scalar metric.

The normative comparison remains open. A procedural theory can attend to early recognition of affected relations, information available before valuation, representation during negotiation, reviewability after settlement, burdens of proof during reopening, or the path by which a repair package is generated. A substantive environmental theory can add criteria concerning ecological value, intergenerational effects, nonhuman interests, or limits on permissible conversion. Those additional commitments exceed the present paper's order-sensitive procedural analysis.

8.3 Knowledge Attribution and Transfer

Knowledge-related conversion illustrates ordering between attribution, recognition, transfer, and later control. The stylized case concerns a contribution that can be associated with symbolic credit, contractual entitlement, licensing authority, institutional status, monetary return, or several of these dimensions. The paper does not assume that these value forms are interchangeable. Its concern is the procedure through which attribution and recognition become connected to a later transfer or settlement.

The comparison begins with two traces in which attribution is positioned on opposite sides of transfer. Equation 82 records the simplified structures.

$$\mathbf{p}_K^{\text{prior}} = (R, A, N, U, X, S), \qquad \mathbf{p}_K^{\text{later}} = (N, U, X, S, R, A, Q). \tag{82}$$

In the first trace of Equation 82, recognition and attribution are part of the configuration from which negotiation, authorization, and transfer proceed. In the second, transfer and settlement occur before later recognition and attribution generate a revision problem. The traces are deliberately agnostic about the legal form of the transfer; X can represent a modeled transfer of rights, control, payment entitlement, licensing position, or another conversion-relevant relation.

Prior attribution can matter through both standing and transformation sensitivity. If attribution associates the contribution with a particular entity for the purposes of negotiation, it can open a negotiation domain or alter the standing trajectory defined in Equation 71. It can also modify the concrete transfer transformation selected by $\mathfrak{F}_X$. A transfer performed after attribution can therefore differ from one performed when the same contribution remains unattributed or differently attributed, even if both procedures eventually produce a public statement of recognition.

Later attribution demonstrates why endpoint credit is an incomplete readout. Suppose a settled transfer is followed by recognition that publicly associates the contribution with the previously unrecognized contributor. The later recognition can change symbolic standing and future opportunity. It can also open a review procedure. Yet the earlier transfer may already have changed control, bargaining position, revenues, dissemination paths, or third-party relations. A coarse readout that records only current attribution can classify the histories as equivalent while a richer history signature distinguishes them.

Review scope becomes especially important in this case. The availability of a revision procedure does not imply that every component of the settled relation can be reconsidered. The review-scope readout scope_q from Definition 7.16 can distinguish a procedure authorized to revise attribution from one also authorized to revise payment, transfer terms, control, or subsequent licensing. Two histories can therefore have equal current recognition and equal formal review access while differing in the remedial competence of the available review body.

The repair corridor again admits partial restoration. A later history can restore attribution while leaving the transfer unchanged. Another can restore attribution and provide compensation while leaving control with the transferee. A further history can revise control prospectively without reproducing the bargaining position that would have existed before transfer. These are structurally different repairs even when several of them satisfy a selected subset of readouts.

The example also clarifies the paper's restraint concerning authorship and contribution. The formalism does not determine who should count as an author, inventor, owner, contributor, or beneficiary. Those categories require domain-specific legal, institutional, epistemic, and normative criteria. The ordering analysis begins after an application specifies how recognition and attribution operations act on the configuration. It then asks how their position relative to negotiation, transfer, settlement, and review changes the realized procedural history.

Under a procedural interpretive frame, relevant readouts can therefore include attribution status, timing of standing, access to negotiation, information available before authorization, transfer scope, review competence, burden of reopening, and the dimensions restored through later repair. The case shows how symbolic recognition can be both substantively important and structurally insufficient to erase the procedural consequences of an earlier transfer.

8.4 Historical Recognition and Retrospective Claims

Historical recognition provides the most direct application of the delayed-recognition machinery developed in Section 6. The stylized case begins with an earlier event, contribution, loss, or relation that remains represented in the accumulated history $\mathcal{H}_\omega$ but does not obtain the procedural recognition needed to affect a contemporaneous conversion or settlement. Recognition arrives later, potentially after the original institution, parties, evidentiary environment, or conversion structure has changed.

The comparison is represented schematically by an early-recognition trace and a candidate extended retrospective trace. Equation 83 records these syntactic orderings; whether the final review step in the retrospective trace is realizable remains conditional on the access structure analyzed below.

$$\mathbf{p}_H^{\text{early}} = (R, A, V, N, U, X, S), \qquad \mathbf{p}_H^{\text{retrospective}} = (V, N, U, X, S, R, Q). \tag{83}$$

Equation 83 does not describe literal insertion of recognition into the past. The retrospective recognition event occurs after settlement and refers to an earlier object retained in the modeled history. Its execution configuration is the later configuration, as emphasized in Subsection 6.2. The resulting update can therefore create present standing or present review access while remaining structurally different from the recognition operation that could have occurred before valuation and settlement.

The first question is whether direct review exists. If the relevant procedure q belongs to $\mathcal{Q}(\omega_R)$ after retrospective recognition, the claim can enter review directly under Assumption 4.6. If direct review remains unavailable but $\mathrm{RevReach}_{q,\mathcal{U}}(\omega_R)$ is nonempty, the procedure is mediated or reopenable relative to the chosen repertoire. If that review-reach set is empty, the route is closed relative to the modeled repertoire. These categories separate present institutional accessibility from the substantive merits of the historical claim.

The second question concerns evidentiary and informational asymmetry. Later recognition can occur with information unavailable during the original procedure, while other records or participants can have disappeared. The information trajectory from Equation 73 can represent this asymmetry without presuming that later information is complete or superior in every respect. A procedural frame can then attend to the burden created by missing records, the treatment of testimony, the timing of disclosure, or the relation between present evidence and an earlier decision.

The third question concerns the target of repair. A retrospective procedure can seek acknowledgment, status, monetary compensation, restoration of access, revision of an institutional record, renewed decision authority, or another response. These targets need not coincide. The restoration profile can therefore record a vector of recoveries rather than forcing retrospective response into a binary success condition. Exact historical return can remain unavailable while several significant forms of present repair are reachable.

Historical distance can also change the identity of the actors affected by repair. The parties present when revision becomes possible can differ from those involved in the original conversion. Institutions can have changed mandates; rights and obligations can have moved through subsequent transactions; current beneficiaries can have entered the relational structure after the original event. The formalism represents these developments as part of the current configuration from which repair must proceed. It does not supply a substantive theory for allocating burdens across changed parties.

This feature reinforces the distinction between a reference history and a normative ideal. An early-recognition history can help identify which procedural opportunities were absent in the realized history, but literal reproduction of that history may be infeasible or normatively inappropriate under later conditions. The repair corridor instead asks which future histories remain available from the current configuration and which selected relations they restore. Rectification remains the theory-relative assessment of whether a particular reachable response is adequate.

The historical case therefore combines nearly every structural distinction developed in the paper: retrospective reference with present execution, state-dependent recognition, settlement-conditioned closure, reopening, review scope, information trajectories, and path-sensitive repair. Its purpose is not to resolve historical claims in general. It shows how an order-sensitive compositional description can keep separate several questions that can be compressed into the single phrase "later recognition."

Table 15 shows that the examples differ in substantive object while sharing a common structural pattern. An earlier identification or recognition event can change standing, admissibility, information, participation, or the state-dependent transformation selected later in the procedure. Settlement can narrow the available future repertoire. A later recognition or attribution event can then create new standing and new response paths while acting on a configuration already changed by the earlier conversion history.

The examples also demonstrate why the paper does not treat order sensitivity, including strict compositional noncommutativity, as an autonomous justice criterion. In each case, structural divergence identifies a question for normative analysis: which standing relation matters, which information should have been available, which participation opportunity should have existed, which settlement deserves finality, which review scope is appropriate, and which repair dimensions are relevant. Different procedural theories can answer those questions differently while using the same structural description of the realized history.

The cases therefore complete the paper's movement from formal ordering to procedural interpretation. Section 9 now states the limits of that movement explicitly. It separates the diagnostic role of order-sensitive composition from substantive normative judgment, clarifies the algebra-inspired status of the representation, locates this note within the broader heterogeneous-conversion programme, and identifies what would be required for empirical institutional application.

Table 15: Order-sensitive structures across the four illustrative conversion cases.

Case	Focal ordering	Main structural channel	Later response and interpretive caution
Labor contribution	Recognition / classification relative to valuation, negotiation, and settlement	Domain opening, standing trajectory, participation influence, state-dependent valuation or conversion	Reopening, reclassification, compensation, or acknowledgment can restore selected dimensions; equal current compensation need not reconstruct earlier negotiation opportunity.
Environmental compensation	Recognition of affected relations relative to valuation and settlement	State-dependent valuation, retrospective recognition, repertoire-relative irreversibility	Review, restoration, compensation, monitoring, or renewed standing can coexist; monetary recovery need not exhaust ecological or relational dimensions.
Knowledge attribution	Attribution / recognition relative to transfer and settlement	Standing, transfer instantiation, endpoint coarse-graining, review scope	Attribution, compensation, or prospective control revision can be meaningful while earlier transfer consequences remain part of the history.
Historical recognition	Recognition relative to earlier conversion and settlement	Delayed recognition, review reachability, information asymmetry, repair corridor	Reopening, acknowledgment, compensation, or record revision remain current interventions; the early-recognition history is a comparison object rather than an automatic remedial blueprint.

9 Scope and Limitations

The preceding sections have developed a deliberately narrow account of order-sensitive procedure in heterogeneous conversion. The present section identifies the boundaries within which that account should be interpreted, separates the paper's structural results from the normative judgments that may be built upon them, and specifies the additional work required for empirical use. The discussion proceeds in four parts. Subsection 9.1 clarifies the relation between compositional diagnosis and procedural evaluation. Subsection 9.2 states the mathematical status of the algebra-inspired language. Subsection 9.3 locates the paper within the broader conversion-justice programme developed in the companion papers. Subsection 9.4 identifies the objects that an empirical or institutional application would need to operationalize. The common aim is to preserve the explanatory usefulness of the formalism while keeping its epistemic status explicit.

9.1 Structural Analysis and Normative Judgment

The principal structural result of the paper is that procedural operations can be order-sensitive in several distinct ways. Earlier operations can alter the configuration on which later operations act, alter the concrete transformation selected by a state-dependent schema, create or remove access to a later operation, or change the set of future histories through which review and repair remain possible. These possibilities are represented through composition, partial domains, state-dependent operation schemas, reachability, review access, and history-sensitive readouts. The resulting structure identifies where two procedural traces diverge and which dimensions of the divergence remain visible under a chosen description.

A structural divergence acquires normative significance only through an interpretive frame of the kind introduced in Definition 7.1. The frame determines which history features matter for the theory under consideration and how the resulting feature tuples are compared, including the possibility that some histories remain incomparable. The relation between the two analytical levels is therefore directional. Structural analysis supplies distinctions that a normative theory can evaluate; the normative theory supplies the reasons for regarding some of those distinctions as procedurally important.

This separation applies even in cases where the structural difference appears intuitively salient. Recognition before conversion can create standing or alter valuation. Participation before settlement can retain an influence window that later participation lacks. Review before closure can operate directly while later review requires reopening. Each statement identifies a change in the procedural history. A justice judgment requires additional premises concerning standing, participation, equality, legitimacy, reliance, finality, repair, institutional burden, or another relevant consideration. The formalism leaves those premises visible rather than embedding them inside a structural definition of order sensitivity.

The same point applies to apparently symmetric histories. Suppose two orderings are compositionally different but a procedural frame selects readouts under which their histories are equivalent. Definition 7.3 then does not classify the ordering difference as procedurally material under that frame. Conversely, histories with the same terminal readout can remain procedurally distinct when the frame includes access trajectories, information conditions, participation windows, review scope, or repair burden. Endpoint agreement and procedural equivalence therefore remain separate analytical questions.

The framework also leaves room for justified sequencing. An order-sensitive rule can be procedurally defensible when the earlier operation establishes competence, verifies evidence, protects reliance, stabilizes expectations, allocates scarce review capacity, or performs another function accepted by the relevant normative and institutional account. One-way composability therefore identifies a dependency relation rather than a defect. Domain closure after settlement can likewise represent an institution's finality structure while remaining open to evaluation concerning the reasons, exceptions, and remedial routes associated with that closure.

Normative pluralism introduces a further limitation. Different procedural theories can select different readout families, compare the same signatures differently, or assign significance to dimensions that the present reduced configuration omits. A theory emphasizing equal standing may focus on the trajectories introduced in Section 7; another may emphasize opportunities for influence, quality of justification, freedom from domination, or the burdens imposed by repeated contestation. The framework can host these differences when they can be represented through explicit readouts and comparison relations, while the substantive justification of the theories remains external to the compositional machinery.

The formalism also leaves some procedural phenomena only partially represented. Reasons, narratives, emotions, trust, symbolic meaning, institutional culture, and interpersonal recognition can influence how a procedure is experienced and judged even when they are difficult to express through the current state variables. Their omission from a reduced model does not indicate irrelevance. It identifies a boundary of the representation. An application that treats one of these phenomena as material would need to enrich the configuration space, the labeled-history representation, the history features, or some combination of them.

A related granularity limit concerns operation identity. The main realized-history object records broad operation-type labels and the state path. Two institutionally different implementations that share one label and generate the same modeled transition can therefore collapse into the same history unless the alphabet, configuration, or history-feature family is refined. This is a representation choice rather than an equivalence claim about the underlying procedures.

For these reasons, the paper's strongest conclusion is diagnostic. Order-sensitive composition can reveal procedural distinctions that become invisible when a conversion is represented only by its terminal allocation, price, settlement, or administrative status. The diagnostic result is useful because it identifies candidate locations for procedural evaluation. The evaluative conclusion remains dependent on a stated theory, an institutional context, and the evidence available for the case under examination.

The propositions proved in this paper should be read in the same conditional manner. Most are internal consequences of stipulated definitions, domain assumptions, state representations, and update repertoires. Their proofs establish consistency relations within the formal model; they do not by themselves establish empirical regularities, causal mechanisms, frequencies, or institutional laws. An empirical application must separately justify the mapping from observed practices to configurations, operation domains, state-dependent schemas, history features, and admissible future updates before the corresponding formal result can be used descriptively.

9.2 Algebra-Inspired Representation

The paper uses algebra-inspired language because composition provides a compact way to represent procedural ordering. The basic mathematical objects are partial transformations on a configuration space, together with state-dependent schemas that select concrete transformations from the current configuration. This structure is sufficient for the paper's principal questions: whether two operations can be composed in both orders, whether the two compositions reach different configurations, whether one order changes the later transformation itself, and whether one sequence is admissible while its reverse is unavailable.

The term *noncommutative* is reserved here for the mutual-composability case in which the two ordered composites are defined at the same initial configuration and yield different terminal configurations. The wider analysis concerns *order-sensitive composition*. For partial transformations, order sensitivity can also appear as path divergence with a shared terminal configuration or as an asymmetry in the domains of the two composites. One-way composability therefore belongs to the paper as a distinct accessibility case rather than as strict noncommutativity.

This compositional analysis does not require a linear operator space, vector-space addition, scalar multiplication, an inner product, or a physical observable algebra. In particular, the paper has not assumed that an expression formed by subtracting two composites is well defined or analytically useful. A commutator of the familiar algebraic form would require additional structure specifying what the difference between transformations means, the space in which that difference lives, and how its magnitude or sign should be interpreted. Those assumptions can be introduced in a later model when an application supplies a justified linear or operator-algebraic representation.

The same restraint applies to the paper's relation to quantum field theoretic and spin-foam-inspired language used in the wider GR programme. The current paper itself uses sequential traces and partial transformations. A broader event model can place those traces inside a partially ordered event structure, and a 2-complex or related construction can be a convenient representation when branching, concurrency, local updates, or multiple interacting conversion sites matter. Those extensions are modeling choices rather than premises of the present derivations. The procedural results established here do not depend on a claim that social institutions instantiate a quantum field theory or that their operators possess the physical meaning of creation, annihilation, or measurement operators.

The state-dependent schema construction deserves similar qualification. The map $\mathfrak{F}_p$ represents the idea that an operation type such as recognition, valuation, conversion, or review can be concretely instantiated differently in different configurations. This captures institutional dependence on standing, categories, evidence, rules, prior settlements, and other relational conditions. The construction is deliberately abstract. It does not by itself specify an algorithm that an institution uses to generate the transformation, and it does not entail that a unique transformation can always be identified empirically. It also relies on the model-relative state-sufficiency convention in Assumption 3.1: if two institutionally different situations are mapped to the same ω even though the omitted difference affects later procedure, the induced-update representation has hidden that dependence rather than explained it.

The reachability constructions in Section 6 are likewise relative to a chosen repertoire of admissible future updates. Exact irreversibility means that a reference configuration is unreachable under that repertoire from the current configuration. It does not establish a metaphysical direction of time, an ontological impossibility of reversal, or an absolute limit across every conceivable institutional redesign. Enlarging the repertoire can change the reachable set. Refining the state description can also change whether two configurations count as equal. Irreversibility is therefore a property of a specified model, configuration description, and intervention repertoire.

Coarse readouts introduce another representational dependence. Two histories can appear equivalent under a selected family of readouts while differing under a finer description. The distinction is intentional: institutions routinely observe and record only some dimensions of their own histories. The formalism can represent the resulting coarse-graining, but the choice of readouts determines which differences remain detectable. A substantive analysis should therefore justify why its selected readouts capture the dimensions relevant to the institutional and normative question.

The mathematical language should consequently be read as a disciplined representation of procedural structure. Its value lies in making composition, domain restrictions, state dependence, accessibility, and history-sensitive comparison explicit. Further algebraic development is possible, including enriched categories of procedures, operator algebras on suitable function spaces, probabilistic transition structures, or field-like models over distributed event complexes. Each extension would add assumptions and should be justified by the problem it helps to analyze rather than by formal generality alone.

9.3 Relation to Broader Conversion Justice

The paper addresses one procedural subproblem inside a broader research programme on heterogeneous value conversion. Its object is the ordering of conversion-relevant operations and the procedural consequences that can follow when those operations are partially defined and state dependent. The framework therefore concentrates on how a conversion history is organized. It does not attempt to replace the wider questions concerning what is being converted, how the relevant entities are constituted, which forms of equivalence are admissible, how power shapes the conversion interface, or how the resulting distributions should be judged.

The first companion paper develops the wider formal vocabulary. It treats relational configurations, partial-order event structure, observation and entity identification, generative transformations, conversion histories, conversion nomos, and theory-relative normative evaluation. The present paper inherits these resources in reduced form. Its procedural alphabet and configuration representation are intentionally smaller because the analytical task is narrower: to isolate order sensitivity and partial composability without reproducing the entire formal system.

The second companion paper develops the broader justice discussion surrounding heterogeneous conversion. Its scope includes procedural, distributive, relational, epistemic, and institutional dimensions; common-equivalent conversion; conversion nomos and norm formation; governance of conversion interfaces; and the possibility that different justice theories select different entities and histories for evaluation. The present paper takes one strand of that programme and develops it more sharply. In particular, Sections 7 and 8 show how ordering can change standing, admissibility, information, participation, review, and recoverability while leaving the substantive justice interpretation theory relative.

Several major conversion-justice questions therefore remain outside the present paper's principal argument. One concerns commensurability: whether heterogeneous values can or should be expressed through a common measure, and which relations disappear when such a measure is imposed. Another concerns distribution: who receives resources, recognition, risks, burdens, or opportunities after conversion. A third concerns epistemic structure: whose categories, observations, evidence, and interpretive standards become operative. A fourth concerns power: who can create, stabilize, revise, or exit the conversion rules. A fifth concerns normogenesis: how repeated practices and institutional feedback can generate or transform the conversion nomos itself.

These dimensions can interact with procedural ordering. For example, a classification scheme can encode an epistemic hierarchy before valuation begins; a conversion rule can distribute bargaining power asymmetrically; repeated settlement practices can stabilize a norm that changes the domain of future claims. The present paper can represent some consequences of these processes once they appear as configuration changes, domain restrictions, or state-dependent transformations. Their causal explanation and normative evaluation can require conceptual resources beyond the order-sensitive formalism.

The boundary is especially important for value transfer. A procedurally well-ordered sequence can still produce a substantively objectionable conversion under a supplied theory of justice. A procedurally burdensome sequence can, under some circumstances, protect other values that the theory regards as important. Similarly, successful repair of selected procedural dimensions does not entail distributive restoration, epistemic recognition, or recovery of every relation affected by the original conversion. The repair corridor in Definition 6.17 was designed to preserve this selectivity by requiring the target dimensions to be stated explicitly.

The relation among the three papers is therefore modular. The broader formalism supplies the relational and conversion architecture. The broader justice paper supplies a plural normative landscape. The present paper supplies a focused account of ordering, partial composability, delayed recognition, review, and repair within that landscape. This modularity also keeps later extensions possible: an empirical study can adopt the ordering framework without accepting every formal device in the wider programme, while a normative study can use the broader justice account without treating order sensitivity as its central analytical object.

Table 16 summarizes the division of analytical labor. The boundaries should be interpreted as interfaces rather than sealed compartments. Ordering can influence distribution, epistemic standing, and power, while those dimensions can determine which orderings become admissible. The purpose of the separation is methodological clarity: each conclusion should be supported by the level of analysis that actually warrants it.

9.4 Empirical and Institutional Extensions

An empirical application would require a translation from the paper's abstract objects into observable or reconstructible institutional features. The formalism is compatible with several research designs, including qualitative process tracing, archival reconstruction, comparative institutional analysis, administrative-data analysis, computational simulation, and mixed-method studies. The appropriate design depends on the conversion process and on whether the research question concerns realized histories, counterfactual orderings, access conditions, institutional rules, or repair possibilities.

The first operational task is to specify the configuration. A researcher must decide which features of ω are required to distinguish the procedural states relevant to the application. These features may include recognized parties, claim categories, attributed contributions, valuation records, authorizations, settlement status, active rules, review channels, and other institutional relations. The state description should be detailed enough to support the intended comparison while remaining empirically recoverable. A description that omits standing or review status can make an important ordering difference invisible; an excessively detailed description can become impossible to estimate or validate.

The second task is to identify operation types and their domains. For each relevant procedural operation, the empirical model needs criteria for determining when the operation is available and what institutional action counts as its execution. Domain membership can sometimes be reconstructed from formal rules, jurisdictional conditions, filing requirements, or administrative records. In other settings, effective access depends on informal practice, discretionary gatekeeping, information, cost, or social relations. The distinction between formal and effective admissibility may therefore require separate readouts rather than a single binary domain indicator.

The third task concerns state-dependent schemas. An empirical researcher would need evidence that the same nominal operation is instantiated differently under different prior configurations. This can be examined by comparing decision rules, available remedies, valuation bases, evidentiary standards, bargaining sets, or other features before and after a relevant earlier operation. The objective is to identify a change in the later operation's concrete response, rather than infer state dependence merely from a different final outcome. Where the institutional rule is stochastic or discretionary, the schema can be represented probabilistically or set-valuedly rather than as a deterministic transformation.

The fourth task is temporal and historical reconstruction. The present paper uses sequential traces, not a general event-poset calculus. When the institutional history contains concurrency or only partial ordering, empirical work must justify the chain or linearization used for the ordering comparison, or extend the model with an event graph or another partial-order representation. Timestamps, durations, deadlines, and waiting periods can be attached wherever they are available, but clock order should not be substituted automatically for the precedence relations relevant to admissibility and state change.

The fifth task is to operationalize the readouts used for procedural interpretation. Standing, information, participation, review access, review scope, and repair targets need domain-specific indicators. Some indicators can be directly observed, such as the existence of a formal appeal route or the date at which a claim category becomes recognized. Others require interpretive coding, such as whether a participant

Table 16: Analytical scope of the present paper within the broader heterogeneous-conversion programme.

Analytical dimension	Present paper	Broader programme	Boundary requiring additional work
Procedural ordering	Composition, partial domains, state-dependent operation schemas, precedence dependency, delayed recognition, review, and repair	Embedded within more general conversion histories	Institutional justification of particular procedural sequences
Entity and relation constitution	Reduced configuration and readout structure	Relational ontology, entity identification, observation, and higher-order relations	Domain-specific ontology and empirical identification rules
Normative evaluation	Theory-relative procedural interpretive frame	Procedural, distributive, relational, epistemic, and other justice perspectives	Substantive justification and comparison of normative theories
Conversion nomos and power	Represented when they alter states, domains, schemas, or future access	Formation, stabilization, revision, and governance of conversion rules	Causal and historical analysis of institutions and power relations
Commensurability and distribution	Treated only when they affect procedural structure	Common-equivalent conversion, heterogeneous value, distributive consequences	Measurement theory, substantive valuation, and distributive assessment
Empirical application	Structural objects specified abstractly	Potential interface with qualitative, quantitative, historical, and computational inquiry	Operational definitions, data, identification strategy, uncertainty, and validation

possessed information sufficient to understand a valuation basis or whether a hearing occurred while a target operation remained responsive. Coding rules should therefore be separated from the normative frame that later evaluates the coded history.

Repair analysis requires an additional counterfactual layer. A repair corridor concerns admissible future histories from the current configuration relative to specified target readouts. Empirical work must consequently identify the intervention repertoire regarded as institutionally available, the costs and constraints associated with those interventions, and the evidence supporting claims about reachability. Administrative rules can sometimes establish that a review or reopening route exists. Claims about whether a route can restore a selected target may require historical analogues, institutional expertise, simulation, or explicit counterfactual assumptions.

Uncertainty should be represented rather than suppressed. The initial configuration can be partially observed, the sequence of events can be disputed, domain conditions can be ambiguous, and the response of a state-dependent schema can be uncertain. Several extensions are possible: sets of admissible configurations, probability distributions over transformations, interval-valued readouts, competing event reconstructions, or sensitivity analysis over alternative repertoires. The present deterministic notation can serve as a baseline while these extensions make epistemic uncertainty explicit.

Identification of causal effects presents a separate challenge. Observing that recognition occurred before conversion in one case and after conversion in another does not by itself establish that ordering caused the outcome difference. Case selection, institutional heterogeneity, strategic behavior, and unobserved prior conditions can jointly affect both the order and the result. Empirical studies seeking causal conclusions therefore need an identification strategy appropriate to their data and institutional setting. The structural formalism can specify the hypothesized pathway but does not substitute for causal identification.

Institutional design applications introduce yet another level of analysis. Designers can use the framework to ask where standing should be established, which operations should remain revisable, which settlement effects should be reversible or reopenable, and which information or participation channels should occur before particular target operations become insensitive. These questions can be represented through domains, influence windows, review access, scope, and repair corridors. Design recommendations additionally require evidence about administrative cost, strategic use, delay, error, participant burden, reliance, enforcement, and the normative objectives of the institution.

Table 17 identifies a practical translation required before the formal objects can support an empirical claim. The table also indicates why an operationalized model should preserve uncertainty. A domain boundary inferred from written rules can differ from effective access in practice; an apparently identical operation can contain discretionary variation; and a reachable repair path can depend on institutional capacities that change over time.

The most suitable empirical extension will therefore be problem specific. A qualitative reconstruction may be the strongest method for a small number of institutionally rich histories. Administrative data may support large-scale analysis of sequence, delay, and review access. Simulation can explore the consequences of alternative sequencing rules when the institutional mechanisms are sufficiently specified. Historical work can reconstruct how conversion nomos and admissibility conditions themselves evolved. These methods can be combined, but the paper does not assume that formalization supplies a privileged route to evidence.

The limitations identified in this section define the intended use of the framework. It is a structural language for making ordering, accessibility, state dependence, closure, and recovery visible. It becomes a normative argument only when combined with an articulated theory of procedural significance, and it becomes an empirical model only when its configurations, operations, domains, readouts, and future repertoires are operationalized against evidence. Maintaining these distinctions allows the formalism to remain extensible while keeping the warrant for each conclusion explicit.

Table 17: Operationalization tasks for empirical and institutional applications.

Formal object	Empirical task	Illustrative evidence	Principal uncertainty
Configuration ω	Specify the institutional state variables relevant to ordering	Case files, administrative records, contracts, rules, interviews, archival sources	Missing relations, disputed status, coarse state description
Operation domain D_{U_p}	Determine when a procedure is formally and effectively available	Jurisdiction rules, eligibility criteria, filing records, access data, observed practice	Gap between formal entitlement and effective access
Schema $\mathfrak{F}_p$	Determine whether prior configurations change the later operation instantiated	Decision rules, remedy sets, valuation bases, bargaining sets, comparative cases	Discretion, stochastic response, hidden decision criteria
History and sequencing	Reconstruct the relevant precedence relations and justify any linearization of concurrent or partially ordered events	Timestamps, correspondence, minutes, event logs, testimony, process records	Missing dates, simultaneous events, contested chronology, linearization dependence
Procedural readouts	Code standing, information, influence, review access, scope, and related dimensions	Institutional documents, participant records, interviews, coded decisions	Measurement validity and coder interpretation
Reachability and repair	Specify feasible future interventions and restoration targets	Review rules, reopening procedures, precedent cases, simulation, expert elicitation	Counterfactual feasibility, future institutional change, intervention cost

10 Conclusion

This paper has examined a narrow procedural question within heterogeneous value conversion: how the order of recognition, classification, attribution, valuation, authorization, negotiation, conversion, settlement, and revision can alter the history through which conversion becomes institutionally effective. Its objective has been structural rather than comprehensive. The analysis has isolated ordering as one source of procedural differentiation while leaving broader questions of commensurability, distributive consequences, epistemic structure, power, and substantive conversion justice to the wider framework developed in the companion papers.

The formal argument began by representing conversion-relevant operations as partially defined transformations on a relational configuration space. This representation makes two sources of order sensitivity visible. First, the output of one operation can change the configuration on which a later operation acts. Second, the earlier operation can change the concrete later transformation selected by a state-dependent operation schema. The resulting difference therefore need not be limited to alternative outputs from fixed maps. In some histories, the earlier event changes the later institutional operation itself by altering standing, classification, evidence, authorization, settlement status, or another component of the relational configuration.

The compositional analysis in Sections 3–5 distinguishes several procedural structures that can otherwise be compressed into a generic claim that "order matters." Two operations can both remain admissible in either order while producing divergent histories. A distinct accessibility asymmetry appears when an earlier operation opens or closes the domain of a later one, generating local precedence dependency and one-way composability. These cases differ analytically. Divergent mutual composability concerns alternative realized transformations or outputs under two available sequences. One-way composability concerns the procedural accessibility of the sequences themselves. The distinction is important because a process can constrain procedural possibility before any comparison of terminal outcomes is made.

The analysis of recognition in Section 5 illustrates this structure. Recognition before conversion can change standing, admissible claims, bargaining position, classifications, or the conversion transformation instantiated in the subsequent configuration. Recognition after conversion is executed on a configuration already altered by conversion and possibly by settlement. Retrospective recognition can therefore be consequential while remaining historically distinct from recognition that occurred earlier. Its referential content may concern a past contribution, status, or event, while its performative execution belongs to the present configuration in which institutional relations, entitlements, expectations, or review possibilities may already have changed.

Section 6 extended this point through reachability. Irreversibility was defined relative to a specified configuration description and repertoire of admissible future updates. Exact return to an earlier configuration is the strongest recovery condition. Coarser recovery can remain possible when selected readouts are restored despite continued historical difference. Partial restoration can remain possible when only a specified subset of procedural or relational dimensions is recovered. These distinctions make room for reopening and repair after delayed recognition without treating later intervention as literal reconstruction of an unrealized past.

The repair-corridor construction adds a history-sensitive dimension to this result. A remedial process is represented through admissible future histories whose terminal configurations recover explicitly selected features relative to a reference configuration. Two interventions can therefore arrive at the same selected endpoint readouts while differing in delay, participation, acknowledgment, intermediate burden, review opportunities, or settlement history. The construction preserves the possibility that repair is substantial even where exact reversal is unavailable, and it also preserves the possibility that endpoint restoration leaves procedurally important historical differences unresolved. Rectification remains a further normative judgment concerning whether the available response is adequate under a supplied theory of justice.

Section 7 translated these structural results into a theory-relative procedural language. The procedural interpretive frame introduced there selects history-level features that a given account treats as normatively relevant and supplies a possibly incomplete comparison relation among the resulting feature tuples. Within this framework, order-sensitive composition is diagnostically useful because it identifies locations where alternative procedural histories can diverge or where one ordering becomes unavailable. Its normative significance depends on what the interpretive frame regards as relevant. Structural order sensitivity can be invisible under a coarse procedural readout, while histories with similar terminal configurations can remain procedurally distinct when standing, information, participation, review, or repair paths differ.

The resulting procedural analysis separates several dimensions that can be collapsed in institutional description. Standing concerns a modeled status or relation. Admissibility concerns whether a particular operation can be executed in the current configuration. Information concerns the epistemic conditions under which procedural involvement occurs. Participation concerns access to relevant procedural operations and the extent to which intervention can still alter a later target. Review concerns both availability and scope: a review channel can remain formally open while losing competence to reconsider a classification, valuation, conversion, settlement, or another earlier operation. These distinctions help identify how ordering can affect procedural opportunity before, during, and after conversion.

The stylized cases in Section 8 show how the same formal distinctions can travel across substantively different domains. In labor contribution and compensation, early recognition can affect standing and the terms under which compensation is generated. In environmental compensation, recognition of affected relations before valuation can produce a different conversion history from later recognition after settlement. In knowledge attribution and transfer, attribution before transfer can alter the transformation through which rights or benefits are assigned, while later attribution may operate mainly through acknowledgment, review, or repair. In historical recognition and retrospective claims, delayed recognition can interact with settlement, evidentiary structure, review access, and changed institutional conditions. These examples function as analytical illustrations. Domain-specific conclusions require empirical and legal specification beyond the formal structure developed here.

Section 9 identified the corresponding limits. The mathematical language employed in this paper is a disciplined representation of partial composition, state dependence, accessibility, and history-sensitive comparison. It does not require a linear operator algebra, and it does not derive a physical or social ontology from quantum-field-theoretic or spin-foam-inspired terminology used elsewhere in the broader GR programme. Reachability depends on the chosen intervention repertoire. Equivalence depends on the chosen configuration description and readouts. State-dependent operation schemas remain abstract until an empirical application identifies how institutional rules, evidence, categories, and discretionary practices select concrete transformations.

These limitations also define the next empirical task. An institutional application would need to identify the events and relations represented in the configuration space, specify the domains of relevant operations, reconstruct the relevant sequencing relations among realized events and justify any linearization used when the empirical event structure is only partially ordered, determine which transformations are state dependent, and justify the procedural readouts used for comparison. It would then need evidence concerning actual access, information, participation, review, settlement, and reopening conditions. Qualitative, archival, administrative-data, computational, historical, and mixed-method designs can each contribute to this reconstruction. The formalism provides a common structural vocabulary for organizing such inquiry; it does not substitute for the evidence through which an institutional history is established.

The paper's contribution is therefore a focused qualification of how procedural justice can be studied in heterogeneous conversion. Conversion procedure can depend on more than the presence of recognized procedural elements. Their ordering can change the configuration in which later operations become available, the transformations that those operations instantiate, and the possibilities that remain for revision or repair. Some orderings are mutually accessible yet divergent. Others exhibit one-way accessibility. Settlement or related closure can make later recognition dependent on reopening, and subsequent repair can recover selected relations while preserving historical difference.

Within the limits of the present framework, these structures provide a way to identify procedural questions that endpoint comparison alone can obscure. A conversion history can retain normatively relevant differences even when a coarse terminal description records similar transfers, statuses, or settlements. Conversely, the existence of structural order sensitivity supplies no universal verdict on which sequence should govern. Procedural judgment requires an explicit account of standing, information, participation, review, repair, and any other features selected by the relevant normative theory. Order-sensitive composition, with strict noncommutativity treated as one specific mutual-composability case, thereby serves as a diagnostic language for tracing how procedural possibility is generated, constrained, and transformed across heterogeneous conversion histories.

References

[1] Emanuela Ceva. *Interactive Justice: A Proceduralist Approach to Value Conflict in Politics*. Routledge, New York, 2016.

[2] E. Allan Lind and Tom R. Tyler. *The Social Psychology of Procedural Justice*. Springer, New York, 1988. doi:10.1007/978-1-4899-2115-4.

[3] John W. Thibaut and Laurens Walker. *Procedural Justice: A Psychological Analysis*. L. Erlbaum Associates, 1975.

[4] Tom R. Tyler. *Why People Obey the Law*. Yale University Press, 1990.