UIM Theorem Frontier

Technical Companion and User Guide

What information survives a mathematical representation?

An executable exposition of feature fibers, exact obstructions, bounded certification, and recovery.

Kevin L. Brown
Independent Researcher

August 2026

Architecture specification: 1.2
State specification: UTF-1.0
Prototype: https://creationunified.com/uim-theorem_frontier

Abstract

UIM Theorem Frontier is an executable mathematical exposition for a precise question: when a mathematical object is replaced by a representation, what information survives, and which properties can still be determined from that representation alone?

The central criterion is the feature-fiber equivalence recorded as UIM Theorem 6.5. For a feature map e and target predicate P, the property factors through the representation exactly when P is constant on every fiber of e. The result itself is elementary; the contribution of Theorem Frontier is to make representation choice, exact fiber computation, adversarial obstruction search, bounded certification, recovery testing, and provenance part of one reproducible scholarly object.

$$\boxed{P = p \circ e \quad \Longleftrightarrow \quad \forall X, Y:\ e(X) = e(Y) \Rightarrow P(X) = P(Y)}$$

A **fiber** is the set of source objects that receive the same representation value.

The current prototype uses finite simple undirected graphs on labeled vertices as an exact computational domain. The reader can change the finite scope, representation coordinates, and target property; the browser then recomputes the mathematical state and reports the strongest conclusion justified by that state. The system is not an automated theorem prover and does not promote finite computation into an unrestricted theorem without a separate mathematical bridge.

The mathematical problem

Mathematics often replaces a structure by a reduced description: a graph by a degree sequence, a matrix by a spectrum, or a dynamical system by selected orbit statistics. Such representations can be useful precisely because they discard information. The logical question is whether they discarded information required by the theorem or property under investigation.

Let D be a class of source objects and let

$$e : \mathrm{Ob}(D) \longrightarrow Z \tag{1}$$

be a representation. Let

$$P : \mathrm{Ob}(D) \longrightarrow A \tag{2}$$

be the target property. We ask whether there exists a rule

$$p : Z \longrightarrow A \tag{3}$$

such that

$$P = p \circ e. \tag{4}$$

Visually, the desired factorization is

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[rr, bend right, "P"'] & e(X) \arrow[r, "p"] & P(X) \end{tikzcd}$$

Theorem 6.5 in *Unified Informational Mathematics* gives the exact criterion: such a p exists if and only if P is constant on every fiber of e. If two source objects collapse to the same representation while the target property distinguishes them, the representation cannot determine that property on any class containing both objects.

> **Novelty boundary.** The fiber-constancy equivalence is elementary. Theorem Frontier does not claim to invent that fact. Its contribution is the executable scholarly architecture that lets a reader choose the representation and predicate, compute the fibers, generate exact witnesses, and see the strongest conclusion that remains justified.

Current source domain: finite labeled graphs

The prototype deliberately begins with one exact domain rather than many partially specified ones. For a fixed positive integer n, let

$$\mathcal{G}_n = \{\text{simple undirected graphs on the labeled vertex set } \{1, \dots, n\}\}. \tag{5}$$

Literal graph equality means equality of edge sets on this fixed labeled carrier. There are

$$|\mathcal{G}_n| = 2^{\binom{n}{2}} \tag{6}$$

such graphs. Thus $|\mathcal{G}_6| = 32{,}768$, which is small enough for exact browser enumeration in the contest prototype.

The current representation builder can combine the following registered coordinates.

ID	Mathematical feature	Definition / role
G-F01	$v(G)$	Number of vertices.
G-F02	$m(G)$	Number of edges.
G-F03	$\mathrm{degseq}(G)$	Sorted vertex-degree sequence.
G-F04	$t(G)$	Number of unordered vertex triples spanning a triangle.
G-F05	$c(G)$	Number of connected components.
G-F06	$a(G)$	Exact labeled adjacency bitstring; positive-control exact representation.

The target predicate registry currently includes connectedness, bipartiteness, triangle presence, Eulerian status under the frozen project convention, and tree status. The ordinary WordPress administration layer may control presentation and which demonstrations are exposed, but it does not redefine these scientific objects.

Canonical obstruction: degree sequence versus connectedness

The primary demonstration asks whether the sorted degree sequence contains enough information to determine connectedness. Set

$$e(G) = \mathrm{degseq}(G), \qquad P_{\mathrm{conn}}(G) = \begin{cases} 1, & G \text{ connected}, \\ 0, & G \text{ disconnected}. \end{cases} \tag{7}$$

The exact adversarial search finds the first obstruction at five vertices. The witness pair is

$$X = P_5, \qquad Y = C_3 \sqcup K_2. \tag{8}$$

Both graphs have the same sorted degree sequence:

$$e(P_5) = (1, 1, 2, 2, 2), \tag{9}$$
$$e(C_3 \sqcup K_2) = (1, 1, 2, 2, 2). \tag{10}$$

But their target-property values differ:

$$P_{\text{conn}}(P_5) = 1, \qquad P_{\text{conn}}(C_3 \sqcup K_2) = 0. \tag{11}$$

Therefore

$$e(P_5) = e(C_3 \sqcup K_2) \quad \text{while} \quad P_{\text{conn}}(P_5) \neq P_{\text{conn}}(C_3 \sqcup K_2), \tag{12}$$

and hence

$$\boxed{P_{\text{conn}} \neq p \circ e \quad \text{for every } p \text{ on any domain containing both graphs.}} \tag{13}$$

Interpretation. The conclusion is not that degree sequence is a weak predictor of connectedness. It is stronger: degree sequence cannot determine connectedness on any graph class containing this witness pair. One exact heterogeneous fiber is sufficient.

The reference search independently verified that no degree-sequence/connectedness obstruction occurs for smaller vertex counts $n \leq 4$, so this is minimal in the searched hierarchy by vertex count.

Feature Test mode

Feature Test asks the property-specific question

$$\boxed{\text{Can the selected representation determine the selected predicate?}} \tag{14}$$

The reader controls the exact finite universe D_U, the representation e, and the predicate P. The deterministic kernel then executes

$$(D_U, e, P) \longrightarrow \text{enumerate } D_U \longrightarrow \text{compute } e(X), P(X) \longrightarrow \text{form fibers} \longrightarrow \text{classify the frontier.} \tag{15}$$

Two operations are central:

- **Compute representation fibers** traverses the complete declared finite universe and constructs the full fiber partition.
- **Find smallest obstruction** searches the declared hierarchy for the first exact heterogeneous fiber and may stop once a decisive counterexample is found.

A positive bounded result requires complete traversal of the declared universe. An interrupted computation cannot be promoted to exhaustive factorization.

Theorem Frontier classifications

The system reports logical status rather than a numerical confidence score.

Status	Meaning	Permitted conclusion
TF-0	Invalid or incompatible experiment specification.	No mathematical conclusion.
TF-1	Exact heterogeneous fiber found.	Non-factorization on every class containing the witness.
TF-2	Complete finite enumeration; every fiber homogeneous.	Exact factorization theorem on the declared bounded universe only.
TF-3	Separate registered theorem establishes factorization on the declared general class.	General conclusion only within that theorem's hypotheses and scope.

The classifications are different logical kinds of result. They are not percentages of certainty.

The finite-to-general boundary

The most important asymmetry is

$$\boxed{\text{one exact collision} \Longrightarrow \text{general factorization fails on every domain containing it}} \tag{16}$$

whereas

$$\boxed{\text{no collision in a completely enumerated } D_U \Longrightarrow \text{factorization proved on } D_U} \tag{17}$$

but, without an additional theorem,

$$\text{factorization on } D_U \not\Longrightarrow \text{factorization on an unrestricted larger class.} \tag{18}$$

Why this matters. Increasing the finite search from thousands to billions of examples can strengthen a finite certificate, but it does not change the quantifier. A general theorem requires a mathematical bridge, not merely more finite cases.

This boundary is enforced by the conclusion engine. It is not left as a discretionary warning in explanatory prose.

Feature factorization and exact recovery are different

Feature Test asks whether a particular property survives a representation. Exact Recovery asks whether the original mathematical object can be reconstructed.

$$\boxed{X} \xrightarrow{e} \boxed{e(X)} \xrightarrow{p} \boxed{P(X)}$$

predicate-specific factorization

$$\boxed{X} \xrightarrow{E} \boxed{E(X)} \xrightarrow{R} \boxed{R(E(X))}$$

structural recovery: $R(E(X)) \simeq X$ (or $= X$ in the exact certificate)

A compressed feature may determine one predicate without retaining the entire graph. Conversely, exact labeled adjacency information permits exact recovery of the current finite labeled graph object because the edge set can be read back from the fixed bit ordering.

This distinction parallels the stronger UIM recovery calculus. UIM Theorem 6.1 shows that a natural recovery isomorphism forces faithfulness and reflects isomorphisms; Theorem 6.4 permits transfer only in an explicitly declared recoverable language. UIM Theorem 6.7 gives a full, faithful, exactly recoverable realization for finite directed graphs. The current prototype uses a simpler finite simple undirected labeled subdomain and verifies exact recovery directly through its adjacency representation.

Target leakage

A representation can directly contain the answer to the chosen predicate. For example, connected-component count determines connectedness because

$$P_{\text{conn}}(G) = 1 \iff c(G) = 1 \tag{19}$$

under the declared nonempty graph convention. Such a factorization is mathematically valid but epistemically unsurprising. The interface therefore flags target-revealing features rather than presenting them as discovered compression.

Fiber Explorer and Property Spectrum

Fiber Explorer exposes the actual partition induced by e. For each representation value, the reader can inspect the graphs in the corresponding fiber, the fiber size, the selected feature values, and the target-predicate values. A heterogeneous fiber is the proof object for TF-1.

Property Spectrum fixes the representation while varying the registered target predicate. The same feature map may preserve one property and destroy another. The output is therefore not a universal "quality score" for the representation; it is a set of property-specific factorization results.

Reproducibility and the Experiment ID

An executable exposition needs a stable way to identify the exact mathematical experiment under discussion. A scholarly state contains the mathematical scope, representation, predicate, state-specification version, domain pack, enumerator version, and scientific registry versions. Canonical serialization produces a deterministic SHA-256 fingerprint.

> **Experiment ID.** A unique fingerprint identifying one exact mathematical experiment for reproducibility and citation. The ID is provenance, not proof.

In plugin release v0.1.2 the public field may still be labeled **State ID**; this guide uses **Experiment ID** as the clearer user-facing term intended for a later presentation update.

A valid citable state must be fully version-pinned. If the mathematical controls change after a state has been recorded, the previous state becomes stale and should not continue to identify the modified experiment.

Deterministic core and validation boundary

The core mathematical result is not produced by free-form AI text. The trusted computational layer includes the graph enumerator, feature implementations, predicate implementations, canonical representation serializer, fiber engine, frontier classifier, recovery logic, and theorem-registry matching.

The project validation discipline distinguishes three states:

$$\boxed{\text{artifact exists} \neq \text{test performed} \neq \text{test passed.}} \tag{20}$$

The frozen architecture therefore requires independent replay rather than allowing the browser implementation to certify itself by assertion. The reference and browser kernels were designed to reproduce the same mathematical states independently, with exact witness and state-hash tests.

How to use the prototype

For a first session, use the canonical connectedness demonstration.

1. Open `https://creationunified.com/uim-theorem_frontier`.
2. Choose **Feature Test**.
3. Select the preset **Degree sequence** $\to$ **connectedness**.
4. Use cumulative scope through $n = 5$.
5. Run **Find smallest obstruction**.
6. Inspect the generated P_5 and $C_3 \sqcup K_2$ witness and confirm the shared degree sequence and differing connectedness values.
7. Read the **Strongest justified claim** and the generalization boundary.
8. Switch to **Exact adjacency** $\to$ **connectedness** and recompute. Observe how the fibers refine.
9. Open **Recovery Test** and run the exact recovery certificate to distinguish property factorization from reconstruction of the full labeled graph.
10. Record the Experiment ID / State ID only after the desired mathematical state has been computed and version-pinned.

What the instrument does not claim

UIM Theorem Frontier does not claim that UIM invented the elementary feature-fiber equivalence. It does not infer unrestricted theorems from finite enumeration. It does not equate preservation of one predicate with recovery of an entire structure. It is not a general-purpose proof assistant, and it does not claim progress on famous open problems merely because their proof architecture can be analyzed through related preservation principles.

The broader UIM monograph also distinguishes formal mathematical coherence from empirical or physical confirmation. Theorem Frontier is a mathematical communication and verification instrument; it does not by itself establish the empirical truth of the upstream physical ontology.

Why this is an executable exposition

The mathematical state is

$$S = (D_U, e, P). \tag{21}$$

When the reader changes the domain, representation, or target property, the object of investigation changes. The browser then recomputes the consequences:

$$\boxed{D_U \longrightarrow e\text{-fibers} \longrightarrow P\text{-values} \longrightarrow \text{witness or certificate} \longrightarrow \text{strongest justified claim}}. \tag{22}$$

The interaction is therefore epistemic rather than decorative. It supports the scholarly loop

$$\boxed{\text{Understand} \longleftrightarrow \text{Experiment} \longleftrightarrow \text{Verify.}} \tag{23}$$

A conventional paper can explain what information a representation may lose. Theorem Frontier lets the reader alter what is preserved, compute what collapses, inspect the exact obstruction, and observe where the theorem stops.

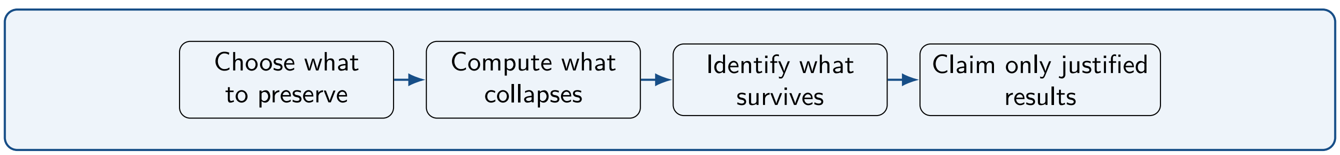
Extension path

The graph domain is a proof of concept, not the intended endpoint. Additional domain packs can be added only when their source objects, equality/equivalence rules, feature maps, predicates, recovery maps, algorithms, and theorem provenance are formally frozen and independently validated.

A natural later extension is deterministic transition systems. UIM Theorem 6.10 proves a full, faithful, exactly recoverable representation for countable deterministic transition systems with intertwining morphisms and shows preservation of finite iterates, orbits, and reachability statements. Importantly, exact representation does not thereby decide unbounded halting questions; the same proof-boundary discipline remains necessary.

Conclusion

Theorem Frontier begins with a modest but exact mathematical rule: if two objects receive the same representation while a target property distinguishes them, the representation cannot determine that property. From that rule follows an executable discipline.



The objective is not to make mathematics look interactive. It is to make the boundary of mathematical knowledge inspectable, reproducible, and interactive.

Primary sources

[1] Kevin L. Brown, *Unified Informational Mathematics: UIPO Foundations, Formal Semantics, Analytic Realization, and Recovery Theorems*, August 2026. Primary source for UIM Theorems 6.1, 6.4, 6.5, 6.7, and 6.10 and the representation-and-recovery calculus.

[2] Kevin L. Brown, *Unified Informational Mathematics and Ten Landmark Problems: An Adversarial Preservation-and-Recovery Stress Test with Formal Proof Development*, August 2026. Secondary source for proof/computation separation and bounded-versus-unrestricted proof discipline.

[3] Kevin L. Brown, *UIM Theorem Frontier - Mathematical and Functional Specification*, v1.2, August 2026. Architecture source for the graph-domain implementation, Theorem Frontier classifications, deterministic runtime, reproducibility architecture, and product non-claims.