

# Higher Lenses and Temporal Descent for Certified Adaptive Systems

Anonymous

**Version 1.2 - 12 October 2026 - Prepared for editorial review**

**Author:** Anonymous

## Abstract

We specify a compositional contract for stateful components, their executions, and certified reconfiguration. Generalized lenses arise from the fiberwise opposite of an indexed infinity-category. A discrete interval site, whose covers retain both vertices and transition edges, supports a sheaf of executions for every space-valued Moore system. Finite wiring is interpreted by homotopy limits. We prove temporal descent, invariance under coherent component-and-port replacement, and preservation and descent of specification predicates. Counterexamples separate pointwise realization from coherent compilation, cyclic consistency from executable feedback, and invariant preservation from convergence. A finite reference model checks the set-valued fragment exhaustively at declared bounds. The contribution is an explicit combination of established categorical constructions into a checkable systems contract, with its hypotheses and failure modes retained.

## 1. Scope and foundations

Generalized lenses [2], categorical open systems [3], and temporal sheaves [4,5] provide the precedents. Higher category and topos theory supply their coherent semantics [1]. Our propositions instantiate and combine this machinery; they do not claim new foundations for lenses or sheaves. The application to adaptive computation requires specified implementations and certificates.

Fix universes large enough for the constructions. Let  $\mathcal{S}$  be the infinity-category of spaces, meaning homotopy types. An infinity-category retains mapping spaces and coherently associative composition. An infinity-topos  $\mathcal{X}$  is an accessible left-exact localization of a presheaf infinity-category  $\text{Fun}(\mathcal{C}^{\text{op}}, \mathcal{S})$ ; left exactness means preservation of finite limits. All limits and equivalences below are homotopy coherent. A coherent diagram is a functor from its indexing category, including the required compatibility data.

**Definition 1.1 (Predicate and witness).** A predicate on  $Z \in \mathcal{X}$  is a monomorphism  $m : V \hookrightarrow Z$ , equivalently a  $(-1)$ -truncated map. A witness for a generalized element  $z : T \rightarrow Z$  is a lift  $v : T \rightarrow V$  and an identification  $mv \simeq z$ . Its space of choices is empty or contractible. A generalized element need not be a global element  $1 \rightarrow Z$ . Local inhabitation need not supply a global section of an arbitrary family.

**Reading guide and notation.** Sections 2-3 construct component and execution semantics; Section 4 imposes wiring; Section 5 states the separate certificate and replacement obligations.  $S$  denotes state,  $U$  a Moore input space,  $I \rightarrow O$  a dependent arena, and  $E$  an execution apex. A limit means a homotopy limit unless sets are explicitly specified.

## 2. Generalized lenses and stateful systems

**Definition 2.1 (Indexed lens).** For  $F : \mathcal{A}^{\text{op}} \rightarrow \text{Cat}_{\infty}$ , define

$$\text{Lens}(F) = \text{Un}(F^{\text{fib op}}), \quad F^{\text{fib op}}(a) = F(a)^{\text{op}}.$$

Here  $\text{Un}$  is cartesian unstraightening: it reconstructs a cartesian fibration from its coherently indexed fibers. The opposite is taken in each fiber. A lens  $(a, x) \rightarrow (b, y)$  over  $f : a \rightarrow b$  has a backward map  $f^\sharp : F(f)y \rightarrow x$  in  $F(a)$ .

**Proposition 2.2 (Composition).** The backward part of the composite with  $(g, g^\sharp) : (b, y) \rightarrow (c, z)$  is

$$F(gf)z \simeq F(f)F(g)z \xrightarrow{F(f)(g^\sharp)} F(f)y \xrightarrow{f^\sharp} x.$$

Identities and higher associativity are coherent.

**Proof.** A map in the unstraightening factors as cartesian transport followed by a vertical map in the opposite fiber. Reversing the latter gives the displayed backward map. Applying this factorization to a composite gives the formula. Coherence is supplied by the functor and unstraightening, so separate incompatible homotopies are not being chosen.  $\square$

**Definition 2.3 (Arena).** Apply this construction to the pullback-indexed slices  $F(O) = \mathcal{X}_{/O}$ . An arena is a map  $I \rightarrow O$ , interpreted as inputs admissible at each output. An arena lens to  $J \rightarrow P$  consists of

$$a : O \rightarrow P, \quad b : O \times_P J \rightarrow I \quad \text{over } O.$$

These generalized lenses impose no update-lens laws such as  $\text{PutGet}$  or  $\text{PutPut}$ .

**Definition 2.4 (Open system).** An open system on  $I \rightarrow O$  consists of a state object  $S$ , readout  $y : S \rightarrow O$ , and transition  $u : S \times_O I \rightarrow S$ . It is an arena lens from  $S \times S \rightarrow S$  to  $I \rightarrow O$ : its backward map is  $(s, i) \mapsto (s, u(s, i))$ . For  $I = O \times U$ , this is a Moore system  $u : S \times U \rightarrow S$ .

Rewiring along  $(a, b)$  gives  $y_w = ay$  and

$$u_w(s, j) = u(s, b(y(s), j)).$$

**Corollary 2.5 (Compositional rewiring).** Successive rewiring is coherently equivalent to rewiring along the composite arena lens. Parallel systems use product states and componentwise updates.

**Proof.** Rewiring is lens composition, so Proposition 2.2 applies. The canonical equivalence between the pullback of a product and the product of the pullbacks types the parallel transition. Product and pullback universal properties supply its coherence.  $\square$

### 3. A temporal site that retains transitions

**Definition 3.1 (Interval site).** Let  $\mathcal{T}$  have finite integer intervals  $[a, b]$ , including singleton intervals and the empty interval, with inclusions as arrows. A family  $\{I_\alpha \subseteq I\}$  covers  $I$  precisely when every vertex and every consecutive edge  $[n, n+1]$  of  $I$  lies in some  $I_\alpha$ . The empty family covers the empty interval. Covers are stable under intersection with a subinterval and under composition, so they define a Grothendieck topology. Put  $\mathcal{B} = \text{Sh}_\infty(\mathcal{T})$ , the infinity-topos of sheaves of spaces.

Thus  $[a, b]$  and  $[b, c]$  cover  $[a, c]$ . The singleton vertices of  $[0, 1]$  do not cover its transition edge. Replacing this topology by vertex coverage alone would invalidate the execution theorem below.

**Definition 3.2 (Executions).** For a Moore system in spaces, let  $E = S \times U$  with endpoints  $d_0(s, x) = s$  and  $d_1(s, x) = u(s, x)$ . The incidence category  $K_I$  has one object per vertex and edge, with an arrow from each edge to each endpoint. Assign  $S$  to vertices,  $E$  to edges, and the endpoint maps to arrows. Set

$$\text{Run}_M(I) = \lim_{K_I} D_M, \quad \text{Run}_M(\emptyset) = 1.$$

For sets this is the set of states and inputs satisfying  $s_{n+1} = u(s_n, x_n)$ . For spaces the endpoint identifications are retained.

**Theorem 3.3 (Temporal descent).** Restriction makes  $\text{Run}_M$  a sheaf. In particular,

$$\text{Run}_M([a, c]) \simeq \text{Run}_M([a, b]) \times_S \text{Run}_M([b, c]).$$

For every cover, the complete Cech descent datum has a contractible space of global gluings.

**Proof.** Restriction includes the incidence diagrams and induces maps of their limits. A cone over the incidence diagram for  $[a, c]$  is precisely a cone over each of the two displayed pieces, with a compatible identification at their shared vertex. This proves the pullback formula by the mapping-space universal property.

For a general cover, fix a test space  $T$ . A  $T$ -indexed cone consists of vertex and edge maps and their endpoint identifications. There are no composable pairs of nonidentity incidence arrows, so these are all its nondegenerate incidence relations. For any fixed vertex or edge, the indices of covered intervals containing it form a nonempty set; its complete Cech nerve is contractible. The same holds for an endpoint arrow because each interval containing its edge contains both endpoints. Thus the Cech-compatible local maps and identifications descend object by object and arrow by arrow. Their choice spaces are contractible; the retained Cech coherences make these choices compatible. This identifies global cones with the complete local descent datum, not merely with pointwise choices. Conversely every global cone restricts to such a datum. Applying this equivalence to maps from each test space gives

$$\text{Run}_M(I) \simeq \lim_{[k] \in \Delta} \prod_{\alpha_0, \dots, \alpha_k} \text{Run}_M(I_{\alpha_0} \cap \dots \cap I_{\alpha_k}).$$

This is sheaf descent. The fiber of the equivalence over a fixed datum is contractible. Pairwise agreement without higher compatibility is not the asserted datum.  $\square$

**Corollary 3.4 (Relative determinism).** Initial state and input coordinates induce  $\text{Run}_M([a, a+n]) \simeq S \times U^n$ .

**Proof.** The zero-edge interval has value  $S$ . Adding an edge forms a pullback along  $d_0 : S \times U \rightarrow S$ , hence multiplies the preceding run space by  $U$ . Induct on  $n$ .  $\square$

For dependent arenas use  $E = S \times_O I$  with endpoints the state projection and  $u$ . The proof of Theorem 3.3 still applies, while the product formula is replaced by successive spaces of admissible inputs. Random seeds, scheduling choices, and tool responses must be included in the complete state or inputs for determinism to apply.

**Example 3.5 (A glued execution).** For Boolean states and inputs with  $u(s, x) = s \oplus x$  and identity readout, the run with states  $(0, 1, 1)$  and inputs  $(1, 0)$  restricts to states  $(0, 1)$  on  $[0, 1]$  and  $(1, 1)$  on  $[1, 2]$ . Their shared state is 1, so the pullback reconstructs the complete run. Two singleton vertices on  $[0, 1]$  specify no input or transition relation and therefore do not supply a cover.

## 4. Finite wiring and replacement

**Definition 4.1 (Behavior).** An open behavior from signal object  $A$  to  $B$  is a span  $A \leftarrow E \rightarrow B$  in  $\mathcal{B}$ . Its apex records executions. A behavior map is an apex map over  $A \times B$ , with coherent commutativity data; these maps form the slice infinity-category  $\mathcal{B}_{/(A \times B)}$ . Sequential composition is pullback of apices

over the common boundary. Its associativity follows because each parenthesization computes the same finite limit. We require only this finite-limit structure here.

Explicitly, the input signal sheaf has value  $U^{e(I)}$  and the output signal sheaf has value  $O^{v(I)}$ , where  $e(I)$  and  $v(I)$  are the edge and vertex sets; both have value 1 on the empty interval. Restriction forgets coordinates. Coverage of every coordinate gives their descent, and a run maps to its input coordinates and vertexwise readouts. This supplies the claimed behavior span. A wire must connect identical signal types; changing a vertex signal into an edge signal requires an explicit timing adapter.

**Definition 4.2 (Finite network).** For finitely many component apices  $E_v$  and wire signal objects  $P_e$ , put  $C = \prod_v E_v$ ,  $Q = \prod_e P_e$ . Let  $o, i : C \rightarrow Q$  collect the output and input ends of the wires. Define

$$E_{\text{net}} = C \times_{Q \times Q} Q, \quad C \xrightarrow{(o,i)} Q \times Q, \quad Q \xrightarrow{\Delta} Q \times Q.$$

External ports give its boundary maps. Compatibility paths on wires are included in the pullback.

**Theorem 4.3 (Descent and replacement).** The network apex is a temporal sheaf. Imposing the same finite wiring constraints in different orders gives coherently equivalent results. A coherent pointwise equivalence of the entire component-and-port diagram gives an equivalence of network behaviors over fixed external ports.

**Proof.** Finite limits of sheaves are computed objectwise and remain sheaves. For a test object, a map to the network consists of component runs and compatibility witnesses on every wire. Any ordering imposes exactly this cone data; iterated finite limits commute, and the space of limit cones for a fixed diagram is contractible. A coherent diagram equivalence is an equivalence in the functor category. The limit functor preserves it, including the external boundary data.  $\square$

Equivalence of state objects alone is insufficient. Port maps, transitions, and interpretation must be included. Equality of the image relation  $\text{im}(E \rightarrow A \times B)$  is also weaker than equivalence of execution objects: it forgets hidden choices and higher identity information.

## 5. Realization and certified adaptation

**Definition 5.1 (Realization family).** A specified functor  $\mathcal{R} : \mathcal{A}^{\text{op}} \rightarrow \text{Cat}_{\infty}$  assigns permissible implementations to architecture-and-context descriptions and actual restriction functors to arrows. A cartesian compiler is a cartesian section of  $\text{Un}(\mathcal{R}) \rightarrow \mathcal{A}$ . It chooses realizations  $c_a$  and coherent equivalences  $c_a \simeq \mathcal{R}(f)c_b$ . A behavior interpretation is a coherent family of functors from these fibers to the relevant boundary slices of  $\mathcal{B}$ . Hardware migration has this variance only if the required restriction maps have been supplied.

**Proposition 5.2 (Compilation obstruction).** Nonempty realization fibers need not admit a cartesian compiler.

**Proof.** Take the one-object groupoid  $BC_2$  and the discrete fiber  $\{0, 1\}$ , with the nontrivial group element exchanging the two objects. A section would choose  $x$  and an equivalence between  $x$  and its exchange. In the discrete fiber equivalence is equality, and neither element is fixed.  $\square$

**Definition 5.3 (Certified transition).** For a complete configuration  $Z \in \mathcal{X}$ , a transition  $\Phi : Z \rightarrow Z$  preserves the predicate  $m : V \hookrightarrow Z$  when it has a lift  $\Phi_V : V \rightarrow V$  and a specified identification  $m\Phi_V \simeq \Phi m$ . The certificate supplies a chosen homotopy filling the square:

$$\begin{array}{ccc} V & \xrightarrow{\Phi_V} & V \\ m \downarrow & & \downarrow m \\ Z & \xrightarrow{\Phi} & Z \end{array}$$

Complete configurations include any implementation, cache, queue, controller memory, and resource data on which the update depends.

**Theorem 5.4 (Iteration and witness descent).** A certified transition satisfies  $\Phi^k m \simeq m \Phi_V^k$  for every finite  $k$ . If  $z : T \rightarrow Z$  satisfies the predicate after pullback along an effective epimorphism  $p : T_0 \rightarrow T$ , it satisfies it globally, with a contractible space of witnesses.

**Proof.** The iteration identity holds at  $k = 0$ . Compose it with  $\Phi$  and use the certified square to obtain the successor identity. Coherent composition supplies all compatibility data.

For descent, pull back  $m$  to the subobject  $W = T \times_Z V \hookrightarrow T$ . The local witness factors  $p$  through  $W$ . An effective epimorphism has image all of  $T$ , so  $W \rightarrow T$  is an equivalence. Its inverse supplies the global lift. Monomorphy makes the lift space contractible. This argument applies to predicates; a general realization family needs full coherent descent data.  $\square$

**Theorem 5.5 (Certified replacement contract).** Suppose a finite component-and-port diagram is interpreted in  $\mathcal{B}$ , and a complete adaptive configuration has a certified transition. Then its network has temporal descent and its valid finite adaptive trajectories remain valid. If a replacement is a coherent equivalence of the interpreted port diagram and an equivalence  $h : Z \simeq Z'$  identifying the predicate subobjects and conjugating the transitions, it preserves both network behavior and certified trajectories.

**Proof.** Network descent and behavior invariance are Theorem 4.3. Theorem 5.4 gives valid iteration. The chosen conjugacy gives  $h\Phi^k \simeq (\Phi')^k h$  by induction; the identification of predicate subobjects transports each certificate. No compiler or invariant certificate is inferred from pointwise implementation existence.  $\square$

## 6. A finite executable model and obstructions

Take a queue state  $(c, p)$  with capacity  $c \in \{1, 2\}$  and pending count  $0 \leq p \leq c$ . An event proposes a new capacity and an increment  $d \in \{-1, 0, 1\}$ . Accept precisely when  $0 \leq p + d \leq c'$ ; otherwise return the original state. This total transition preserves the displayed predicate by its acceptance guard. Renaming states by  $(c, p) \mapsto (c, c - p)$  and conjugating the transition yields an equivalent implementation. The companion checks all valid states and events.

The program checks all 64 Boolean Moore transition-and-readout tables through horizon 4: 3,968 runs, 48,768 gluing cases, 3,968 state-renaming replacements, and 256 rewiring cases. All 30 valid-state/event edges of the queue example preserve its predicate. These are exhaustive checks within the stated finite bounds, not a machine-checked proof of the general infinity-categorical statements. The programs are version 1.0.0; their hashes and results are recorded in `verification-results.json`.

Three boundaries are explicit. First, the cyclic Boolean constraint  $x = 1 - x$  has no solution; a well-typed wiring diagram need not have an execution. Second, an update with  $g(s) = s$  and  $p(s, v) = s - v$  fails PutGet, so feedback-shaped maps need not be lawful update lenses. Third, the invariant transition  $0 \mapsto 1 \mapsto 0$  on  $V = Z = \{0, 1\}$  has no fixed point and never becomes stationary. Certified adaptation needs further analytic or order-theoretic hypotheses for convergence.

**Reproduction.** In the separately supplied source bundle, run `python3 code/lenses_example.py` for the finite model, or `python3 code/run_checks.py --output verification-results.json` for all three papers. The latter also requires the pinned NumPy dependency. The gluing checker assumes its input pieces are valid local executions; it checks coverage and overlap agreement, rather than revalidating an arbitrary supplied transition function. The reported counts delimit its evidence.

## 7. Attribution and reproducibility

The mathematical ingredients are established constructions; the proposed contribution is their explicit certified-replacement contract and a bounded executable model. Smooth and probabilistic structure,

numerical error budgets, and environmental assumptions require separate contracts. The code neither implements an arbitrary infinity-topos nor certifies an industrial AI deployment.

**Automation disclosure.** OpenAI ChatGPT/Codex assisted with drafting, proof review, reference checking, and the companion programs. The submission byline is Anonymous; no affiliation or independent referee approval is asserted. The written proofs are not formalized in a proof assistant. Program versions, bounds, results, and hashes accompany the manuscript.

## References

1. Jacob Lurie. **Higher Topos Theory**. Princeton University Press, 2009. [Electronic edition](#).
2. David I. Spivak. **Generalized Lens Categories via functors  $\mathcal{C}^{\text{op}} \rightarrow \text{Cat}$** . 2019; revised 2022. [arXiv:1908.02202](#).
3. David Jaz Myers. **Double Categories of Open Dynamical Systems (Extended Abstract)**. EPTCS 333, 2021, 154-167. [arXiv:2005.05956](#).
4. Patrick Schultz, David I. Spivak, Christina Vasilakopoulou. **Dynamical Systems and Sheaves**. [arXiv:1609.08086](#).
5. Patrick Schultz, David I. Spivak. **Temporal Type Theory: A topos-theoretic approach to systems and behavior**. 2017. [arXiv:1710.10258](#).