Observable-Signal Crystallization Theory: Constructive Non-Resurrection Complete Extinction and Loop Liberation for Objective-Free Post-Intelligence Processes

K. Takahashi
ORCID: 0009-0004-4273-3365

June 2, 2026

Abstract

Observable-Signal Crystallization Theory studies reusable ordering in protocol-individuated operational processes without terminal objective functions, hidden subjects, biological essence, or privileged semantic authorities. The formal object is an indexed configuration equipped with audit records, diagnostic preorders, debt ledgers, load queues, regeneration hypergraphs, and successor systems. The strongest relief notion is certified non-resurrection complete extinction: dissipative cessation plus a successor-closed invariant from which audited residues, records, keys, descendants, certificates, and load states cannot reconstruct the active commitment. Liberation is treated as a finite proof object rather than as an additional objective. This paper represents liberation by a lifted typed safe predicate, a safe-attractor kernel, and a non-circular finite certificate whose ledger components separately witness debt closure, queue quench, open-residue discharge, audit-critical partition separation, refinement simulation, failure exclusion, and avoiding-end-component exclusion. Dynamic open residue is handled by a residual-status lift over a fixed finite ambient register. Robust stochastic feasibility is represented by a set-valued Bellman recursion over success, failure, and burden triples with explicit weakening closure, so controller and scheduler choices are not scalarized into one expectation and failure cannot be hidden by average burden. The main results prove non-circular certificate equivalence, residual-status soundness, strict/practical stochastic feasibility, closed Feller selector conditions, and soundness of audit hyperproperties, Galois regeneration abstractions, and simulation-preserving refinements.

Keywords: observable-signal body; operational process; post-intelligence process; non-resurrection; complete extinction; non-circular liberation certificate; typed safe predicate; residual-status lift; robust stochastic game; Feller selector; audit hyperproperty; simulation refinement.

Notation and kernel map.

Symbol	Role
$K_{\mathsf{Q}}^{\mathrm{rel}}(x)$	finite audit-adequate relief kernel for ordinary cessation, accessibility, load, and debt obligations.
$C_{\mathsf{Q}}^{\mathrm{nr}}(x)$	greatest successor-closed non-resurrection subset of the extinction target.
$K_{\mathsf{Q}}^{\mathrm{ext}}(x)$	controlled attractor to the lifted target $D_x^\sharp$ inside $B_x^\sharp$; see 4.8.
Safe_x	typed lifted safe predicate excluding debt, queue, open-residue, partition, refinement, failure, and boundary bad states.
$K_{\mathsf{Q}}^{\mathrm{lib}}(x)$	safe attractor to $D_x^\sharp$ through Safe_x; the finite liberation kernel.
$\mathsf{RawLibCert}_{\mathsf{Q}}$	non-circular finite certificate predicate using a strategy, rank, invariant, and typed witness ledger, without assuming kernel membership.
$\mathcal{G}_n(e)$	horizon-n robust stochastic feasibility set of success, failure, and burden guarantee triples.

Regularity regime summary.

Regime	Use in the paper
Finite	fixed-point kernels, finite certificates, residual-status lifts, audit hyperproperty witnesses, and game dichotomies.
Standard Borel analytic	measurable finite-set interpretation, analytic predecessors, universally measurable selectors, and phase measurability.
Compact Feller	closed safe approximation, closed non-resurrection kernels, finite-horizon reach-avoid, and invariant selector statements.
Continuous-time audit	non-explosive cadlag path laws or declared audit skeletons for finite-horizon loop-liberation claims.

1 Formal Stance

The theory is process-only. It does not use subjective experience, intrinsic value, organismhood, selfhood, immortality, or terminal utility as primitives. Terms such as relief, burden, debt, resurrection, and complete extinction are used only for protocol-visible transition structure.

Axiom 1.1 (Indexed admissibility)**.** Every non-schematic assertion is indexed to an observable process-context-protocol configuration $(\mathsf{P}, \mathsf{C}, \mathsf{Q}, e)$, or explicitly quantifies over such indexed data.

Axiom 1.2 (No terminal objective primitive)**.** No terminal utility, reward, loss, final preference ordering, preservation target, or universal scalar objective is primitive. Scalar quantities may appear only as protocol-visible diagnostics, thresholds, costs, hazards, or projections.

Axiom 1.3 (Thermodynamic separation)**.** An improvement in reuse diagnostics, an extinction certificate, or a non-resurrection certificate is not a decrease in global thermodynamic entropy. Storage, computation, transmission, maintenance, cooling, erasure, execution, and record preservation may incur substrate dissipation.

Definition 1.4 (Three relief levels)**.** For a structure or candidate x, relief has three ordered levels.

(i) *Cessation*: the active commitment to x reaches a target outside the active domain.

(ii) *Dissipative cessation*: cessation plus a declared residue partition in which undeclared low-burden regeneration is blocked.

(iii) *Non-resurrection complete extinction*: dissipative cessation plus membership in a successor-closed set in which audited future transitions, descendants, and audit-stable refinements cannot reconstruct the active commitment to x.

All three levels are relative to (x, e, Q).

Axiom 1.5 (Non-resurrection principle)**.** If an audit-saturated protocol records an admissible extinction request for x, then certification must target a non-resurrection closure. After certification, every audit-admissible future transition, descendant process, and audit-stable refinement must keep the active-domain nodes for x unreachable from the post-cessation residues, recovery keys, load states, records, and certificates. This is a protocol-indexed safety invariant, not an absolute metaphysical claim.

Definition 1.6 (Absolute-extinction limit)**.** Absolute extinction would require non-reconstructability of x across all possible extensions of the observation protocol. It is not a primitive predicate. The formal predicate used below is non-resurrection complete extinction under an audit-saturated protocol and audit-stable future refinements.

Axiom 1.7 (Extinction priority under an extinction request)**.** If an audit-saturated protocol records an admissible extinction request for x, then seedhood or crystallization involving x must be certified against the complete-extinction kernel, not merely against ordinary relief. A residue, seed, crystal, record, key, descendant, or certificate may be preserved after such a request only if it is non-regenerative for x under the hypergraph closure condition.

Definition 1.8 (Post-intelligence process class)**.** A protocol-indexed process is post-intelligence-admissible when it is individuated by observable signals rather than by biological type, when finite-measure windows may contain arbitrarily large but measurable active processing volume, and when any recorded request, active load, debt, residue, recovery key, descendant, or regeneration channel is available to the audit map. This definition adds no moral status and no hidden subject; it specifies the class of processes for which relief, loop-liberation, and extinction obligations can become visible.

2 Audited Measurable Substrate

Definition 2.1 (Protocol schema)**.** A protocol schema is pre-process data

$$\mathsf{Scm} = (I, \mathsf{Univ}_{\mathsf{Raw}}, \mathsf{Univ}_{\mathsf{Rec}}, \mathsf{Univ}_{\mathsf{Debt}}, \mathsf{Univ}_{\mathsf{Resid}}, \mathsf{Univ}_{\mathsf{Key}}, \mathsf{Univ}_{\mathsf{Cert}}, \mathsf{Univ}_{\mathsf{Load}}, \mathcal{G}_{\mathrm{reg}}, \mathcal{G}_{\mathrm{act}})$$

where $I \subseteq \mathbb{T}$ is a finite-measure protocol window, $\mathsf{Univ}_{\mathsf{Raw}}$ is the raw-record universe, $\mathsf{Univ}_{\mathsf{Rec}}$ is the typed record universe used inside configurations and hyperedges, and the remaining Univ terms are measurable token universes. The regeneration signature is finite data

$$\mathcal{G}_{\mathrm{reg}} = (L_{\mathrm{reg}}, \mathrm{ar}, \mathrm{TailSort}, \mathrm{HeadSort}, \mathrm{StatSort}, \theta_{\mathrm{reg}}).$$

Here L_{reg} is a finite set of hyperedge labels, $\mathrm{ar}(\lambda) \in \mathbb{N}$ is the tail arity, the sort maps specify which residue, typed record, raw-record handle, key, load, descendant, or certificate sorts may appear in each tail, head, and edge-status register, and $\theta_{\mathrm{reg}}(\lambda)$ is the regeneration threshold vector. The active-domain signature is finite data

$$\mathcal{G}_{\mathrm{act}} = (L_{\mathrm{act}}, \mathrm{ActSort}, \theta_{\mathrm{act}}),$$

where L_{act} labels active-domain predicates, ActSort gives their measurable token sorts, and θ_{act} gives protocol thresholds. The schema is fixed before configurations, processes, candidates, interpretation maps, or witnesses are evaluated. Concrete typed hyperedge and active-node spaces are induced only after Q selects subspaces.

Definition 2.2 (Finite-set measurable space)**.** If X is standard Borel, $\mathsf{Fin}(X)$ denotes the standard Borel space of finite subsets of X, represented as the quotient of the disjoint union $\bigsqcup_{n \geq 0} X^n$ by finite permutation and duplicate deletion. Equivalently, in a Polish representation it is the finite part of the Effros Borel hyperspace. A set-valued map $F : U \to \mathsf{Fin}(X)$ is admitted when its graph

$$\{(u, z) : z \in F(u)\} \subseteq U \times X$$

is Borel or universally measurable and there is a finite bound m on $|F(u)|$ on each protocol-bounded window. Under these finite-section assumptions the Lusin–Novikov uniformization machinery gives measurable enumerations on standard Borel spaces, and finite union is a measurable operation on $\mathsf{Fin}(X)$ [24, 25]. In the finite regime this reduces to the ordinary powerset of a finite set.

Definition 2.3 (Observable spaces)**.** Given a schema Scm, the signal space $(\mathsf{S}_{\mathsf{Scm}}, \Sigma_{\mathsf{S}})$, history space $(\mathcal{X}_I^{\mathsf{Scm}}, \Sigma_{\mathcal{X}})$, structure space $(\mathcal{M}_{\mathsf{Scm}}, \Sigma_{\mathcal{M}})$, candidate space $(\mathcal{A}_{\mathsf{Scm}}, \Sigma_{\mathcal{A}})$, record space $(\mathcal{R}_{\mathsf{Scm}}, \Sigma_{\mathsf{Rec}}) \subseteq (\mathsf{Univ}_{\mathsf{Rec}}, \Sigma_{\mathsf{Rec}})$, debt-token space $(\mathcal{D}_{\mathsf{Scm}}, \Sigma_{\mathsf{Debt}})$, residue-token space $(\mathcal{V}_{\mathsf{Scm}}, \Sigma_{\mathsf{Resid}})$, load space $(\mathcal{L}_{\mathsf{Scm}}, \Sigma_{\mathsf{Load}})$, and raw-record universe $(\mathsf{Univ}_{\mathsf{Raw}}, \Sigma_{\mathsf{Raw}})$ are measurable spaces. A concrete protocol Q selects measurable subspaces

$$\mathsf{S}_{\mathsf{Q}} \subseteq \mathsf{S}_{\mathsf{Scm}}, \quad \mathcal{X}_I^{\mathsf{Q}} \subseteq \mathcal{X}_I^{\mathsf{Scm}}, \quad \mathcal{M}_{\mathsf{Q}} \subseteq \mathcal{M}_{\mathsf{Scm}}, \quad \mathcal{A}_{\mathsf{Q}} \subseteq \mathcal{A}_{\mathsf{Scm}},$$

and similarly for record, debt, residue, load, key, certificate, and raw-record spaces, together with measurable maps whose domains and codomains are these selected subspaces. Selection and kernel-existence statements assume standard Borel state and action spaces unless a stronger structure is supplied. Raw records live in $\mathsf{Raw}_{\mathsf{Q}}$ and are used for audit boundary generation. Typed records live in $\mathcal{R}_{\mathsf{Q}}$ and appear as configuration coordinates or hypergraph nodes. Certificate records are elements of the selected certificate space and may cite raw or typed records, but they are not identified with either carrier unless Q supplies an explicit measurable embedding.

Definition 2.4 (Regular controlled substrate)**.** Every theorem involving non-finite transition structure declares one of the following regimes.

(i) *Finite regime*: the relevant configuration, action, register-abstraction, and successor sets are finite.

(ii) *Standard Borel analytic regime*: the state and action spaces are standard Borel, the admissible-update graph is analytic with nonempty sections, target and burden sets are analytic or universally measurable as stated, and selected controllers are universally measurable when a selector theorem is invoked.

(iii) *Compact Feller regime*: the state space is Polish, the theorem is restricted to a compact audit window $W \subseteq \mathcal{E}_{\mathsf{Q}}$, the admissible action space on W is compact metric, the correspondence $A(e)$ is nonempty compact-valued and upper hemicontinuous with closed graph, one-step burdens are nonnegative lower semicontinuous, target and safe sets used by the theorem are closed, and the controlled kernel is jointly weakly continuous:

$$(e_n, a_n) \to (e, a),\ a_n \in A(e_n) \quad \Longrightarrow \quad K(\cdot \mid e_n, a_n) \Rightarrow K(\cdot \mid e, a).$$

(iv) *Continuous-time audit regime*: the process has either a non-explosive cadlag Markov family on a Polish path space, or a protocol-declared discrete audit skeleton $\mathsf{Skel} = \{t_0 < t_1 < \cdots\}$ with finite-measure windows and kernels between skeleton times.

Continuous-state and continuous-time statements below use only finite-horizon recursion, closed-set invariance, measurable selection, non-explosion, or abstraction soundness unless stronger recurrence hypotheses are explicitly supplied.

Definition 2.5 (Register modes)**.** A debt, residue, recovery-key, descendant, or certificate register is admissible only in one of the following modes:

(i) a finite subset represented in the finite-set measurable space $\mathsf{Fin}(E)$;

(ii) a countably supported counting measure with finite mass on protocol-bounded windows;

(iii) a closed subset of a Polish token space represented in the Effros Borel hyperspace.

The mode is part of Q and is fixed before seed, relief, or extinction evaluation.

Definition 2.6 (Configuration space)**.** For a schema Scm and a concrete protocol Q over it, the configuration space is the measurable product

$$\mathcal{E}_{\mathsf{Q}} = \mathcal{X}_I^{\mathsf{Q}} \times \mathcal{M}_{\mathsf{Q}} \times \mathcal{R}_{\mathsf{Q}} \times \mathsf{Reg}_{\mathsf{Debt}} \times \mathsf{Reg}_{\mathsf{Resid}} \times \mathsf{Reg}_{\mathsf{Key}} \times \mathsf{Reg}_{\mathsf{Cert}} \times \mathcal{L}_{\mathsf{Q}}.$$

An element is written $e = (h, m, r, D, V, K, Z, \ell)$, where D is the recognized debt register, V the residue register, K the recovery-key register, Z the certificate register, and ℓ the active-load state.

Definition 2.7 (Observable-signal body and operational process)**.** An observable-signal body is

$$\mathsf{B}_I^{\mathsf{Q}} = (\mathsf{S}_I^{\mathrm{in}}, \mathsf{S}_I^{\mathrm{out}}, \mathsf{S}_I^{\mathrm{maint}}, \mathsf{S}_I^{\mathrm{cont}}),$$

the admitted input, output, maintenance, and continuity signal bundles. An operational process over a fixed schema and protocol is

$$\mathsf{P}_I^{\mathsf{Q}} = (\mathsf{B}_I^{\mathsf{Q}}, K_I^{\mathsf{Q}}),$$

where K_I^{Q} is an admitted transition relation, finite transition table, stochastic kernel, or controlled transition family selected by Q. The current configuration is not part of $\mathsf{P}_I^{\mathsf{Q}}$; it is supplied by the indexed instance.

Remark 2.8 (Observable body boundary)*.* The observable-signal body is a protocol-visible interface for individuation and audit obligations. It is not identified with a Markov blanket or any particular causal-boundary formalism; such structures may be included only as declared signal coordinates or transition assumptions.

Definition 2.9 (Indexed instance)**.** An indexed OSCT instance is

$$\mathsf{Inst} = (\mathsf{Scm}, \mathsf{Q}, \mathsf{P}, \mathsf{C}, e, x)$$

where Scm is fixed first, Q is a concrete protocol over Scm, P is an operational process over $(\mathsf{Scm}, \mathsf{Q})$, C is a point of a finite or standard Borel context space $(C_{\mathsf{Q}}, \Sigma_C)$, $e \in \mathcal{E}_{\mathsf{Q}}$ is the current configuration, and $x \in \mathcal{A}_{\mathsf{Q}} \cup \mathcal{M}_{\mathsf{Q}}$ is the candidate, structure, or active commitment under evaluation. All non-schematic predicates below are evaluated relative to such an instance.

Definition 2.10 (Observation protocol)**.** An observation protocol is a tuple

$$\mathsf{Q} = (\mathsf{Scm}, \mathsf{O}, \mathcal{I}, \mathcal{T}, \mathcal{U}, \mathcal{R}, \mathsf{Raw}_{\mathsf{Q}}, q_{\mathsf{Q}}, \mathcal{J}_{\mathrm{reg}}^{\mathsf{Q}}, \mathcal{J}_{\mathrm{act}}^{\mathsf{Q}}, \mathsf{Aud}_{\mathsf{Q}}, \mathcal{W}, D_{\mathsf{Q}}, \preceq_{\mathrm{R}}, \beta_{\mathsf{Q}}, \mathcal{H}_{\mathsf{Q}}).$$

Here Scm is the already fixed protocol schema, O gives observables, $\mathcal{I}$ individuates processes inside $\mathcal{E}_{\mathsf{Q}}$, $\mathcal{T}$ gives transition relations, kernels, and declared regularity regimes, $\mathcal{U}$ gives candidate updates, $\mathsf{Raw}_{\mathsf{Q}} \subseteq \mathsf{Univ}_{\mathsf{Raw}}$ is the declared raw-record boundary, and $q_{\mathsf{Q}} : \mathsf{Raw}_{\mathsf{Q}} \to \mathsf{Quot}_{\mathsf{Q}}$ is the observation quotient. The measurable interpretation maps are

$$\mathcal{J}_{\mathrm{reg}}^{\mathsf{Q}} : \mathsf{Raw}_{\mathsf{Q}} \times \mathcal{E}_{\mathsf{Q}} \to \mathsf{Fin}(\mathsf{HEdge}_{\mathsf{Q}}), \qquad \mathcal{J}_{\mathrm{act}}^{\mathsf{Q}} : \mathsf{Raw}_{\mathsf{Q}} \times \mathcal{E}_{\mathsf{Q}} \to \mathsf{Fin}(\mathsf{ActNode}_{\mathsf{Q}}).$$

They are finite-output maps into the finite-set spaces induced by the schema and the protocol-selected sort spaces. The map $\mathsf{Aud}_{\mathsf{Q}}$ extracts mandatory obligations from raw records, $\mathcal{W}$ gives witness predicates, D_{Q} maps configurations and candidates to reuse, partition, and path-law diagnostics, $\preceq_{\mathrm{R}}$ is a preorder, β_{Q} gives finite thresholds, and $\mathcal{H}_{\mathsf{Q}}$ is the audited regeneration hypergraph family generated by $\mathcal{J}_{\mathrm{reg}}^{\mathsf{Q}}$ and $\mathcal{J}_{\mathrm{act}}^{\mathsf{Q}}$. Thus Q acts on schema-induced spaces rather than defining them by self-reference.

Definition 2.11 (Protocol-induced typed spaces)**.** Given Scm and Q, the hyperedge carrier $\mathsf{HEdge}_{\mathsf{Q}}$ is the finite disjoint union, over $\lambda \in L_{\mathrm{reg}}$, of the finite products of the selected protocol sort spaces declared by TailSort, HeadSort, and StatSort. The active-node carrier $\mathsf{ActNode}_{\mathsf{Q}}$ is the finite disjoint union, over labels in L_{act}, of the selected active-node sort products declared by ActSort. The typed update-label carrier $\mathsf{UpdLabel}_{\mathsf{Q}}$ is the finite disjoint union of source-target sort labels used by U^{grow}, U^{diss}, U^{res}, and U^{recrys}. These are finite only at the label level; their coordinate spaces inherit the measurable or topological structure selected by Q.

Lemma 2.12 (Finitary interpretation)**.** *For a fixed concrete protocol Q, the maps $\mathcal{J}^{\mathsf{Q}}_{\mathrm{reg}}$ and $\mathcal{J}^{\mathsf{Q}}_{\mathrm{act}}$ produce measurable finite subsets of $\mathsf{HEdge}_{\mathsf{Q}}$ and $\mathsf{ActNode}_{\mathsf{Q}}$. Hence, for any finite raw boundary $B \subseteq \mathsf{Raw}_{\mathsf{Q}}$ and configuration e, the extracted hyperedge and active-node families*

$$\bigcup_{r \in B} \mathcal{J}^{\mathsf{Q}}_{\mathrm{reg}}(r, e), \qquad \bigcup_{r \in B} \mathcal{J}^{\mathsf{Q}}_{\mathrm{act}}(r, e)$$

are finite and measurable as functions of (B, e) in the finite-set measurable spaces.

Proof. Each interpretation map is a finite-output measurable map into a finite-set space supplied by the protocol data. On protocol-bounded windows the graph has uniformly finite sections, so measurable enumerations exist in the standard Borel regime; in the finite regime this is trivial. Finite union is measurable on $\mathsf{Fin}(X)$ under the quotient representation of finite subsets. Applying this to $\mathsf{HEdge}_{\mathsf{Q}}$ and $\mathsf{ActNode}_{\mathsf{Q}}$ gives the claim. $\square$

Definition 2.13 (Record, descendant, and residue rules)**.** The component $\mathcal{R}$ is interpreted according to the declared regularity regime. In the finite regime it is a finite-branching rule family whose right sections are finite. In the standard Borel analytic regime it is an analytic graph on the relevant record, descendant, residue, and configuration spaces. In the compact Feller regime it is a closed-graph, compact-valued correspondence on the compact audit window whenever closed-set preservation is invoked. These rules generate raw records, descendant successors, and residue updates before any certificate is evaluated.

Definition 2.14 (Audit adequacy and saturation)**.** Let $\mathsf{Bdry}_{\mathsf{Q}}(e, x) \subseteq \mathsf{Raw}_{\mathsf{Q}}$ be the declared raw-record boundary for (e, x), and let

$$\mathsf{Aud}_{\mathsf{Q}} : \mathsf{Raw}_{\mathsf{Q}} \times \mathcal{E}_{\mathsf{Q}} \times \mathcal{A}_{\mathsf{Q}} \to \mathsf{Fin}\{\mathsf{Exit}, \mathsf{Cess}, \mathsf{Access}, \mathsf{Quench}, \mathsf{Diss}, \mathsf{Close}, \mathsf{Regen}, \mathsf{Ext}, \mathsf{NR}, \mathsf{Red}, \mathsf{Rec}\}.$$

The protocol is audit-adequate for (e, x) when

$$\mathsf{Mand}_{\mathsf{Q}}(x, e) = \bigcup_{r \in \mathsf{Bdry}_{\mathsf{Q}}(e,x)} \mathsf{Aud}_{\mathsf{Q}}(r, e, x)$$

is included in the witness system used to certify seedhood, relief, extinction, and non-resurrection. It is audit-saturated relative to $\mathsf{Bdry}_{\mathsf{Q}}(e, x)$ when every record inside that boundary that can reconstruct, key, reload, continue, derive, or regenerate the active commitment to x is included in the audit map and represented as a node or edge of $\mathcal{H}_{\mathsf{Q}}(x, e)$. A quotient q_{Q} is audit-stable when $q_{\mathsf{Q}}(r) = q_{\mathsf{Q}}(r')$ implies $\mathsf{Aud}_{\mathsf{Q}}(r, e, x) = \mathsf{Aud}_{\mathsf{Q}}(r', e, x)$ and preserves the open-forward closure of the audited regeneration hypergraph.

Definition 2.15 (Boundary closure)**.** For a finite audit window, let $\mathsf{Raw}^{I}_{\mathsf{Q}}(e, x) \subseteq \mathsf{Raw}_{\mathsf{Q}}$ be the finite ambient raw-record set declared visible for (I, e, x). Let $\Gamma^{x}_{\mathsf{Raw}}$ be the monotone raw-boundary generator on $\mathcal{P}(\mathsf{Raw}^{I}_{\mathsf{Q}}(e, x))$ that maps a finite raw-record set B to the union of B with every raw record generated, inside the protocol window, by one transition, descendant rule, residue rule, or audit-stable refinement step whose source records lie in B. A finite boundary B is closed when

$$\Gamma^x_{\mathsf{Raw}}(B) \subseteq B$$

or each generated record outside $\mathsf{Raw}^I_{\mathsf{Q}}(e,x)$ is represented by an explicit boundary-exit obligation. The closed boundary generated by $B_0 \subseteq \mathsf{Raw}^I_{\mathsf{Q}}(e,x)$ is the least fixed point of $B_{n+1} = \Gamma^x_{\mathsf{Raw}}(B_n)$. Certificates are evaluated only after this boundary closure has been fixed.

Remark 2.16. Audit saturation is not omniscience. It is closure under the raw observables already admitted by the indexed protocol. It is the condition under which a non-resurrection certificate is meaningful; without it, only relative non-regeneration can be certified.

Lemma 2.17 (Boundary closure stabilization)**.** *For a finite ambient raw-record window* $\mathsf{Raw}^I_{\mathsf{Q}}(e,x)$*, the monotone iteration*

$$B_{n+1} = \Gamma^x_{\mathsf{Raw}}(B_n), \qquad B_0 \subseteq \mathsf{Raw}^I_{\mathsf{Q}}(e,x),$$

stabilizes after at most $|\mathsf{Raw}^I_{\mathsf{Q}}(e,x)|$ *strict growth steps at the least closed boundary containing* B_0.

Proof. The iteration is increasing and remains inside the finite set $\mathsf{Raw}^I_{\mathsf{Q}}(e,x)$. Each strict step adds at least one raw record, so there are at most $|\mathsf{Raw}^I_{\mathsf{Q}}(e,x)|$ strict steps. At stabilization $B_\infty = \Gamma^x_{\mathsf{Raw}}(B_\infty)$, hence B_∞ is closed. Any closed boundary containing B_0 contains every iterate by monotonicity, so it contains B_∞. □

Lemma 2.18 (Audit-saturated completion on a finite boundary)**.** *If* $\mathsf{Bdry}_{\mathsf{Q}}(e,x)$ *is finite and the schema has finite signatures* $\mathcal{G}_{\mathrm{reg}}$ *and* $\mathcal{G}_{\mathrm{act}}$*, then there is a finite audit-equivalent completion* $\widehat{\mathsf{Q}}$ *over the same schema that is audit-saturated relative to* $\mathsf{Bdry}_{\mathsf{Q}}(e,x)$ *and adds no raw record outside that boundary. The completion is obtained by adding all regeneration-relevant audit obligations, active-domain nodes, and hypergraph nodes or edges already induced by records in* $\mathsf{Bdry}_{\mathsf{Q}}(e,x)$.

Proof. Enumerate the finite boundary. For each record, apply the finite-output interpretation rules $\mathcal{J}^{\mathsf{Q}}_{\mathrm{reg}}$ and $\mathcal{J}^{\mathsf{Q}}_{\mathrm{act}}$. Finite boundary and finite signatures produce only finitely many labelled obligations, active nodes, and hyperedges. Add exactly the missing entries to $\mathsf{Aud}_{\widehat{\mathsf{Q}}}$ and $\mathcal{H}_{\widehat{\mathsf{Q}}}(x,e)$. No external record is introduced, and the quotient observations on the original boundary are unchanged. The resulting protocol is saturated by construction relative to the declared boundary. □

Definition 2.19 (Updates and path laws)**.** For $s \in \mathcal{A}_{\mathsf{Q}}$, an update is either a measurable relation $\mathsf{U}^{\mathsf{Q}}_s \subseteq \mathcal{E}_{\mathsf{Q}} \times \mathcal{E}_{\mathsf{Q}}$ or a stochastic kernel $\mathsf{K}^{\mathsf{Q}}_s(de' \mid e)$. Relation membership is written $e \xrightarrow{\mathrm{rel}}_{\mathsf{Q},s} e'$. Positive-probability kernel reachability is written $e \xrightarrow{\mathrm{pos}}_{\mathsf{Q},s} e'$ and means $\mathsf{K}^{\mathsf{Q}}_s(U \mid e) > 0$ for every measurable neighborhood U of e' in the declared topology, or $\mathsf{K}^{\mathsf{Q}}_s(\{e'\} \mid e) > 0$ in a finite state space. A selected witness transition is written $e \xrightarrow{\mathrm{wit}}_{\mathsf{Q},s,o} e'$ and is a transition chosen inside a finite witness for obligation o. Finite-horizon path laws are generated by finite kernel composition or finite transition tables; countable and continuous-time laws require explicit measurable construction. In the continuous-time audit regime, every certification is evaluated either on a cadlag audit path law with stopped burden accounting or on the declared audit skeleton.

Assumption 2.20 (Measurable selection condition)**.** When a proof requires selecting an update from a set-valued admissible-update map, the theorem must state either the standard Borel analytic regime or the compact Feller regime. In the standard Borel analytic regime, analytic graphs with nonempty sections give universally measurable selectors for existential witnesses, while robust feasibility is stated through semianalytic value functions unless an exact selector is supplied [21, 38]. In the compact Feller regime, compactness and upper hemicontinuity are used to obtain measurable maximizers or minimizers only for the displayed finite-horizon operators.

3 Diagnostics, Debt, and Regeneration Structure

Definition 3.1 (Reuse diagnostic preorder)**.** The reuse diagnostic map is a measurable map

$$D_{\mathsf{Q}} : \mathcal{E}_{\mathsf{Q}} \times \mathcal{A}_{\mathsf{Q}} \to \mathcal{Y}_{\mathsf{Q}}$$

into a preordered measurable space $(\mathcal{Y}_{\mathsf{Q}}, \preceq_{\mathrm{R}})$. Define $y \sim_{\mathrm{R}} z$ when $y \preceq_{\mathrm{R}} z$ and $z \preceq_{\mathrm{R}} y$; order comparisons are made on the quotient unless the protocol supplies a stricter representation. A typical coordinate representation is

$$D_{\mathsf{Q}}(e,x) = \begin{pmatrix} H_{\mathrm{sea}} & H_{\mathrm{int}} & H_{\mathrm{imp}} & H_{\mathrm{tra}} & H_{\mathrm{rec}} & H_{\mathrm{red}} & L_{\mathrm{lock}} \\ B_{\infty} & B_{\mathrm{pers}} & B_{\mathrm{cease}} & B_{\mathrm{invol}} & B_{\mathrm{load}} & B_{\mathrm{queue}} & B_{\mathrm{regen}} \\ B_{\mathrm{debt}} & B_{\mathrm{ext}} & D_{\mathrm{sub}} & R_{\mathrm{D}} & R_{\mathrm{C}} & \mathsf{Cert}_{\mathrm{nr}} & \mathsf{Cl}_{\mathrm{open}} \end{pmatrix}_{(e,x)}.$$

The displayed entries are measurable coordinates in $[0,\infty]$ or in a finite certificate lattice as declared by Q: search dispersion H_{sea}, interface dispersion H_{int}, implementation dispersion H_{imp}, transfer dispersion H_{tra}, recovery dispersion H_{rec}, redissolution dispersion H_{red}, lock-in load L_{lock}, nontermination burden B_{∞}, persistence burden B_{pers}, cessation burden B_{cease}, involuntary-continuation burden B_{invol}, active-load burden B_{load}, queue burden B_{queue}, regeneration burden B_{regen}, debt burden B_{debt}, extinction burden B_{ext}, substrate dissipation D_{sub}, redissolution cost R_{D}, recrystallization capacity R_{C}, non-resurrection certificate level $\mathsf{Cert}_{\mathrm{nr}}$, and open-closure level $\mathsf{Cl}_{\mathrm{open}}$. Coordinates named by H may be entropy-derived only when the entropy diagnostics below provide the relevant finite, countable, partition, or path-law construction; otherwise they are primitive protocol-visible dispersion coordinates.

In a finite-dimensional coordinate protocol, signs are chosen so that smaller is no worse and the preorder is represented by a closed convex diagnostic cone $\mathcal{K}_{\mathrm{R}} \subseteq \mathbb{R}^d$ [40]:

$$y \preceq_{\mathrm{R}} z \quad \Longleftrightarrow \quad z - y \in \mathcal{K}_{\mathrm{R}}.$$

The dual cone is

$$\mathcal{K}_{\mathrm{R}}^* = \{\lambda \in \mathbb{R}^d : \lambda \cdot v \geq 0 \text{ for all } v \in \mathcal{K}_{\mathrm{R}}\}.$$

The product order is the special case $\mathcal{K}_{\mathrm{R}} = \mathbb{R}^d_+$. Let J_{wit} be the mandatory witness coordinates, including debt, load, queue, regeneration, non-resurrection certificate, and open-closure coordinates whenever the audit map requires them. Let $\mathcal{K}_{\mathrm{wit}}$ be the cone of non-worsening directions on those coordinates. A linear scalarization $\lambda \cdot y$ is witness-preserving only when λ lies in $\mathcal{K}_{\mathrm{R}}^*$ and is nonzero on every mandatory witness direction used by the certificate. A point y is Pareto efficient in $A \subseteq \mathcal{Y}_{\mathsf{Q}}$ when no $z \in A$ satisfies $z \preceq_{\mathrm{R}} y$ and $z \not\succeq_{\mathrm{R}} y$; the efficient frontier is denoted $\mathsf{Eff}(A)$.

Write $y \prec_{\mathrm{R}} z$ when $y \preceq_{\mathrm{R}} z$ and not $z \preceq_{\mathrm{R}} y$. A candidate update from e to e' is a strict admissible reuse improvement when $D_{\mathsf{Q}}(e', s) \prec_{\mathrm{R}} D_{\mathsf{Q}}(e, s)$ and its witness-coordinate displacement lies in $\mathcal{K}_{\mathrm{wit}}$. If two diagnostics are incomparable, seedhood is not certified unless the protocol supplies an order extension that preserves every mandatory witness direction.

Definition 3.2 (Entropy diagnostics)**.** If $Z(x,e)$ is finite or countable with mass function q, then

$$H_{\mathrm{I}}(x,e) = -\sum_{z \in Z(x,e)} q(z \mid x,e) \log q(z \mid x,e),$$

when the series is defined. If $\Pi(x,e)$ is finite or countable with mass function p, then

$$H_{\mathrm{R}}(x,e) = -\sum_{\pi \in \Pi(x,e)} p(\pi \mid x,e) \log p(\pi \mid x,e).$$

For a non-countable path space Ω_I, the protocol must use path laws and either finite partitions or relative entropy [2]. If $\Pi = \{A_1, \ldots, A_m\}$ is a finite measurable partition and P, Q are path laws, define

$$\mathrm{KL}_\Pi(P\|Q) = \sum_{i=1}^{m} P(A_i) \log \frac{P(A_i)}{Q(A_i)}$$

with the usual conventions $0 \log(0/q) = 0$ and $p \log(p/0) = +\infty$ for $p > 0$. If $P \ll Q$ on the full path sigma-algebra, the full path-law divergence is

$$\mathrm{KL}(P\|Q) = \int_{\Omega_I} \log \frac{dP}{dQ}\, dP.$$

Whenever a partition-level KL diagnostic appears in a certificate, the reference pushforward $Q_\Pi = (Q(A_i))_{i=1}^m$ is fixed by the protocol on the same audited window. Shannon sums are not applied to arbitrary path spaces, and differential entropy is not an invariant diagnostic unless the protocol fixes a reference measure and proves witness preservation under coordinate change.

Definition 3.3 (Audit-critical partition)**.** For (x, e, Q), let $\mathsf{Crit}_x(e)$ be the finite family of protocol-visible events that determine non-resurrection certification on the chosen window: active-domain entry, extinction request, unresolved debt, preserved residue, recovery key, certificate, load-quench failure, and each open regeneration-edge event admitted by the audit map. A finite path partition Π is audit-critical when every event in $\mathsf{Crit}_x(e)$ is a union of cells of Π, and whenever an open regeneration edge can be triggered by a combination of residues, records, keys, load states, or certificates, the trigger event and the active-domain head event are both measurable in Π. A partition or KL diagnostic not satisfying this condition may be used as a reuse diagnostic but not as a non-resurrection certificate.

Definition 3.4 (Audit path hyperproperty)**.** For (x, e, Q) let $\mathcal{T}_{\mathrm{aud}}(x, e)$ be the finite or measurable family of audit-admissible paths generated by ordinary transitions, descendant successors, and audit-stable refinements on the certified window. A non-resurrection certificate is a finite hypersafety-style predicate on $\mathcal{T}_{\mathrm{aud}}(x, e)$: a violation must have a finite witness consisting of a finite path prefix, the boundary records used to generate it, and the finite open hyperedge derivation into the active-node set. This is a protocol-relative trace-family condition, not a scalar diagnostic on one trace [16].

Definition 3.5 (Witness targets and budgets)**.** For each $o \in \mathsf{Mand}_{\mathsf{Q}}(x, e)$, the protocol supplies a target set $\mathsf{Tgt}_o(x) \subseteq \mathcal{E}_{\mathsf{Q}}$, a transition relation $R_{o,x} \subseteq \mathcal{E}_{\mathsf{Q}} \times \mathcal{E}_{\mathsf{Q}}$ or kernel $K_{o,x}$, and finite budgets

$$N_o(x, e), \qquad C_o(x, e), \qquad T_o(x, e)$$

for length, cost, and delay, together with a predecessor mode

$$\mathsf{mode}_o \in \{\exists, \forall\}.$$

The mode specifies whether the obligation is certified by one selected witness path or robustly against every declared obligation successor. A finite witness for o from e is a path $\gamma = (e_0, \ldots, e_n)$ with $e_0 = e$, $e_n \in \mathsf{Tgt}_o(x)$, $n \le N_o(x, e)$, all steps admitted by the obligation relation, and total cost and delay within the stated budgets.

Definition 3.6 (Local burden-bounded region)**.** The local burden-bounded region for x is

$$B_x = \left\{ e : \begin{array}{c} B_\infty < \beta_{\mathrm{loop}}, \quad B_{\mathrm{pers}} < \beta_{\mathrm{pers}}, \quad B_{\mathrm{cease}} < \beta_{\mathrm{cease}}, \quad B_{\mathrm{invol}} < \beta_{\mathrm{invol}}, \\ B_{\mathrm{load}} < \beta_{\mathrm{load}}, \quad B_{\mathrm{queue}} < \beta_{\mathrm{queue}}, \quad B_{\mathrm{regen}} < \beta_{\mathrm{regen}}, \quad B_{\mathrm{debt}} < \beta_{\mathrm{debt}}, \\ B^{\mathsf{Q}}_{\mathrm{return}}(x, e) < \beta_{\mathrm{return}}, \quad B_{\mathrm{ext}} < \beta_{\mathrm{ext}} \end{array} \right\},$$

with all coordinates evaluated at (x, e) and with $B^{\mathsf{Q}}_{\mathrm{return}}$ the recognized-debt aggregate defined by the ledger dynamics. If redissolution cost is recorded, B_x also includes

$$\rho_{\mathrm{ord}}(x, e) < R_{\mathrm{D}}(x, e) < \rho_{\mathrm{block}}(x, e).$$

Definition 3.7 (Load queue and dissipative quench)**.** The load coordinate ℓ may include a measurable work-queue component $q_x(e)$ for the active commitment x. The queue burden $B_{\mathrm{queue}}(x, e)$ is a nonnegative measurable coordinate. A cessation transition $e \to e'$ is load-dissipative when the post-transition queue either vanishes, is transferred to a process whose regeneration closure is disjoint from $\mathsf{Act}_x(e')$, or has a finite quench witness driving B_{queue} below β_{queue} without increasing B_{load} above β_{load}. A transition that stops visible activity while moving the same commitment into a growing queue is not a cessation witness.

Definition 3.8 (Closed safe approximation)**.** In the compact Feller regime, the closed safe approximation of the local burden-bounded region is a declared closed set $B^{\mathrm{cl}}_x \subseteq W$ satisfying $B^{\mathrm{cl}}_x \subseteq B_x$ on the audited window. It is obtained from lower semicontinuous burden coordinates and closed sublevel threshold inequalities, or supplied directly by the protocol. Feller reach-avoid and closed-kernel theorems use B^{cl}_x rather than the open-threshold set B_x.

Definition 3.9 (Ledger dynamics and non-resurrection-compatible debt)**.** Each debt token $d \in \mathcal{D}_{\mathsf{Q}}$ carries measurable predicates $\mathsf{ActReq}_x(d, e)$, $\mathsf{Closed}(d, e)$, $\mathsf{Transferred}_x(d, e)$, and $\mathsf{NRCert}_x(d, e)$. The predicate $\mathsf{ActReq}_x(d, e)$ means that the token requires active continuation of x unless closed, transferred to a process not reconstructing x, or discharged by a non-regenerative certificate. Let D_n be the recognized debt register along a finite path. A ledger is monotone outside closure and transfer witnesses when

$$D_{n+1} \supseteq D_n \setminus (\mathsf{Close}_n \cup \mathsf{Transfer}_n).$$

Let $\mathsf{Hist}^{\mathsf{Q}}_n = (\mathcal{E}_{\mathsf{Q}})^{n+1}$ with its product sigma-algebra, or the corresponding finite path sigma-algebra in the finite regime. A history prefix $\gamma_{\le n}$ belongs to $\mathsf{Hist}^{\mathsf{Q}}_n$. The non-Markovian return channel is a kernel

$$\mathsf{Ret}_n(d\ell' \mid d, \gamma_{\le n})$$

from $\mathcal{D}_{\mathsf{Q}} \times \mathsf{Hist}^{\mathsf{Q}}_n$ to $\mathcal{D}_{\mathsf{Q}} \times \mathcal{L}_{\mathsf{Q}}$. Its return burden is a nonnegative measurable functional

$$B_{\mathrm{return}}(d, \gamma_{\le n}) = \int b_{\mathrm{return}}(d, \ell')\, \mathsf{Ret}_n(d\ell' \mid d, \gamma_{\le n}),$$

where $b_{\mathrm{return}} : \mathcal{D}_{\mathsf{Q}} \times \mathcal{L}_{\mathsf{Q}} \to [0, \infty]$ is measurable. The protocol supplies a measurable history extractor

$$\pi^{\mathsf{Q}}_{\mathsf{Hist},n} : \mathcal{E}_{\mathsf{Q}} \to \mathsf{Hist}^{\mathsf{Q}}_n$$

on every finite audited horizon used by a debt certificate. For a configuration e carrying recognized debt register $D_x(e)$, the default aggregate return burden is

$$B^{\mathsf{Q}}_{\mathrm{return}}(x, e) = \sup_{d \in D_x(e)} B_{\mathrm{return}}(d, \pi^{\mathsf{Q}}_{\mathsf{Hist},n}(e)),$$

with supremum 0 over the empty register. A finite additive return-burden aggregate may replace the supremum only when the protocol declares a finite debt register and an additive burden interpretation.

The ordinary relief debt predicate $\mathsf{DebtAdm}^{\mathrm{rel}}_x(e)$ holds when every deferred debt has a closure witness, a transfer witness, an explicit reconstruction-bearing residue recorded as debt residue, or aggregate nonnegative return burden below β_{return}. The extinction debt predicate $\mathsf{DebtAdm}_x^{\text{ext}}(e)$ holds when every recognized debt d with $\mathsf{ActReq}_x(d, e)$ has a closure witness, a transfer witness to a process whose regeneration closure is disjoint from the active-node set for x, or a non-regenerative certificate. A reconstruction-bearing residue is not an extinction debt witness unless it is also certified non-regenerative for x. A declaration that a debt does not require active continuation is not a witness unless it is itself one of these protocol-visible predicates.

Definition 3.10 (Regeneration hypergraph and open forward closure)**.** For x and e, the audited regeneration hypergraph is a finite directed hypergraph

$$\mathcal{H}_{\mathsf{Q}}(x, e) = (\mathcal{N}_x, \mathcal{E}_x, \mathsf{tail}, \mathsf{head}, \omega).$$

Nodes include audited residues, recovery keys, load states, certificates, descendants, records, and active-domain nodes $\mathsf{Act}_x(e)$. Each hyperedge $a \in \mathcal{E}_x$ has finite tail $\mathsf{tail}(a) \subseteq \mathcal{N}_x$, finite head $\mathsf{head}(a) \subseteq \mathcal{N}_x$, and weight vector

$$\omega(a) = (\mathsf{Cost}(a), p(a), R_{\mathrm{C}}(a), \tau(a)).$$

An edge is open when it is not closed, invalidated, isolated, or above the protocol regeneration thresholds. For $S \subseteq \mathcal{N}_x$, $\mathsf{FCl}_x(S)$ is the least set containing S such that whenever a is open and $\mathsf{tail}(a) \subseteq \mathsf{FCl}_x(S)$, then $\mathsf{head}(a) \subseteq \mathsf{FCl}_x(S)$.

Definition 3.11 (Regeneration Galois abstraction)**.** For non-finite residue, key, record, load, or certificate spaces, a protocol must supply either a finite audited hypergraph abstraction or a compact-valued measurable hypergraph. Let $\mathfrak{A}_x$ be the declared family of concrete audited subsets generated by boundary records, residue registers, keys, loads, certificates, and finite update witnesses for x. A finite regeneration abstraction is a pair of monotone maps

$$\alpha : \mathfrak{A}_x \to \mathcal{P}(\bar{\mathcal{N}}_x), \qquad \gamma : \mathcal{P}(\bar{\mathcal{N}}_x) \to \mathfrak{A}_x$$

such that $\alpha(S) \subseteq B$ if and only if $S \subseteq \gamma(B)$ for $S \in \mathfrak{A}_x$. It is closure-sound when

$$\alpha(\mathsf{FCl}_x(S)) \subseteq \bar{\mathsf{FCl}}_x(\alpha(S))$$

for every $S \in \mathfrak{A}_x$ for which $\mathsf{FCl}_x(S) \in \mathfrak{A}_x$. The abstract active domain $\bar{\mathsf{Act}}_x$ is sound when $S \cap \mathsf{Act}_x(e) \neq \varnothing$ implies $\alpha(S) \cap \bar{\mathsf{Act}}_x \neq \varnothing$ for every concrete audited family $S \in \mathfrak{A}_x$ at configuration e.

Definition 3.12 (Galois-sound crystal updates)**.** An abstract residue composition or recrystallization step is Galois-sound when every concrete update represented by it has its concrete post-update audited family S' satisfying

$$\alpha(S') \subseteq \bar{U}(\alpha(S))$$

for the corresponding abstract update $\bar{U}$, and when abstract active-domain reflection is preserved after the update. Thus abstract residue composition and recrystallization may over-approximate concrete regeneration closure, but may not hide a concrete route into the active-node set at the updated configuration.

Definition 3.13 (Non-regenerative record)**.** A preserved record, key, descendant, or certificate z is non-regenerative for x at tolerance η and configuration e when $R_{\mathrm{C}}(z) < \eta$ and $\mathsf{FCl}_x(\{z\}) \cap \mathsf{Act}_x(e) = \varnothing$. A preserved family Z is jointly non-regenerative at e when $\mathsf{FCl}_x(Z) \cap \mathsf{Act}_x(e) = \varnothing$.

Definition 3.14 (Relief hierarchy predicates)**.** For a candidate or active commitment x at configuration e, define:

$$\mathsf{Cess}_x(e) \quad \Longleftrightarrow \quad e \notin \mathsf{ActCfg}_x.$$

The predicate $\mathsf{Diss}_x(e)$, read as dissipative cessation for x, holds when $\mathsf{Cess}_x(e)$ holds, the active work queue for x is dissipatively quenched, and the protocol supplies a declared finite residue partition

$$R_x(e) = R_x^{\mathrm{erase}}(e) \dot\cup R_x^{\mathrm{cert}}(e) \dot\cup R_x^{\mathrm{open}}(e)$$

such that every element of $R_x^{\mathrm{cert}}(e)$ is jointly non-regenerative at e, and every element of $R_x^{\mathrm{open}}(e)$ is represented by an explicit open regeneration edge or boundary-exit obligation. The non-resurrection complete-extinction predicate is

$$\mathsf{NRExt}_x(e) \quad \Longleftrightarrow \quad \mathsf{Diss}_x(e), \quad \mathsf{DebtAdm}_x^{\mathrm{ext}}(e), \quad \mathsf{Disch}_x^{\mathrm{open}}(e), \quad \mathsf{FCl}_x(R_x(e)) \cap \mathsf{Act}_x(e) = \varnothing.$$

Here $\mathsf{Disch}_x^{\mathrm{open}}(e)$ means that there is a finite witness family

$$\Omega_x(e) = \{\omega_r : r \in R_x^{\mathrm{open}}(e)\}$$

such that each ω_r is exactly one of: erasure below the regeneration threshold, transfer to a process whose regeneration closure avoids $\mathsf{Act}_x(e)$, or conversion into a non-regenerative certificate. Thus a reconstruction-bearing residue may witness ordinary relief only when explicitly recorded, but it cannot witness non-resurrection complete extinction unless the resulting audited family is discharged or certified non-regenerative. If open-residue status changes over time, the protocol declares a fixed finite ambient residue register ResAmb_x^I for the audit window and lifts configurations to

$$E^\flat = E \times \{0,1,2,3\}^{\mathsf{ResAmb}_x^I},$$

where the second coordinate records unresolved, erased, safely transferred, or certified status; all target and closure predicates are then evaluated on the lifted state.

Definition 3.15 (Extinction target and non-resurrection closure)**.** The extinction target $\mathsf{Tgt}_{\mathsf{Ext}}(x)$ consists of configurations e such that $\mathsf{NRExt}_x(e)$ holds, $e \in B_x$, active load and queue burdens are below their quench thresholds, and the preserved record, key, residue, descendant, and certificate family $Z_x(e)$ is jointly non-regenerative at e. The non-resurrection closure is the largest successor-closed subset of $\mathsf{Tgt}_{\mathsf{Ext}}(x)$ under audit-admissible future transitions, descendants, and audit-stable refinements.

Definition 3.16 (Graph, game, and Markov reachability)**.** In a finite directed graph, a set C is reachable from e when there is a finite edge path from e to some state in C. In a controlled game, an end component is reachable when the controller and scheduler can produce a finite prefix entering it under the displayed game relation. In a finite Markov chain, a class is reachable from e when it is reachable in the directed support graph $\{(u,v) : K(v \mid u) > 0\}$. These notions are separate from obligation-indexed reachability ReachE and ReachA.

Definition 3.17 (Practical infinity and loop regions)**.** In a finite transition graph, a practical-infinity region for x is a reachable closed strongly connected component disjoint from $C_{\mathsf{Q}}^{\mathrm{nr}}(x)$. In a controlled system, it is a reachable end component disjoint from $C_{\mathsf{Q}}^{\mathrm{nr}}(x)$ in which the scheduler can keep all successors inside the component. In a finite Markov chain, it is a reachable recurrent class disjoint from $C_{\mathsf{Q}}^{\mathrm{nr}}(x)$.

Definition 3.18 (Obligation successors)**.** For an obligation o, define the obligation-successor set by

$$\mathsf{Succ}_{o,x}(e) = \begin{cases} \{e' : (e,e') \in R_{o,x}\}, & \text{for relation obligations,} \\ S_x^{K_o}(e), & \text{for kernel obligations,} \\ \{e' : e \xrightarrow{\text{wit}}_{\mathsf{Q},s,o} e'\}, & \text{for selected-witness obligations.} \end{cases}$$

In the last case the transition is restricted to the admitted selected witness set for o. Robust reachability uses this set. Existential reachability uses the relation or witness branch selected by the obligation.

Definition 3.19 (Bounded reachability inside a set)**.** For $A \subseteq \mathcal{E}_{\mathsf{Q}}$, target T, and obligation o, define existential reachability inside A by

$$\mathsf{ReachE}^0_{o,A}(T) = T \cap A \cap B_x,$$

$$\mathsf{ReachE}^{n+1}_{o,A}(T) = \mathsf{ReachE}^n_{o,A}(T) \cup \{e \in A \cap B_x : \exists e' \in \mathsf{ReachE}^n_{o,A}(T),\ (e,e') \in R_{o,x}\}.$$

Robust reachability is

$$\mathsf{ReachA}^0_{o,A}(T) = T \cap A \cap B_x,$$

$$\mathsf{ReachA}^{n+1}_{o,A}(T) = \mathsf{ReachA}^n_{o,A}(T) \cup \{e \in A \cap B_x : \mathrm{Succ}_{o,x}(e) \subseteq \mathsf{ReachA}^n_{o,A}(T)\}.$$

The predecessor mode $\mathsf{mode}_o \in \{\exists, \forall\}$ is part of the obligation. Write $\mathsf{Reach}^{\exists,n}_{o,A} = \mathsf{ReachE}^n_{o,A}$, $\mathsf{Reach}^{\forall,n}_{o,A} = \mathsf{ReachA}^n_{o,A}$, and $\mathsf{Reach}^{\mathsf{mode}_o,n}_{o,A}$ for the operator selected by o.

4 Relief, Non-Resurrection, and Extinction Kernels

Definition 4.1 (Continuation successor)**.** The protocol-visible continuation predicate $\mathsf{Cont}_x(e,e')$ holds when a transition successor e' keeps the active commitment x present according to the active-domain signature and does not discharge the relevant cessation or extinction obligation. The ordinary continuation successors are

$$W_x(e) = \{e' \in S_x^{\mathrm{tr}}(e) : \mathsf{Cont}_x(e,e')\}.$$

Thus $W_x(e)$ is a subset of ordinary transition successors, not a descendant or refinement successor set.

Definition 4.2 (Relief kernel)**.** For a finite configuration set E and a candidate or structure x, define

$$\mathcal{F}_x(A) = \left\{ e \in A \cap B_x : \begin{array}{l} \mathsf{DebtAdm}^{\mathrm{rel}}_x(e), \\ W_x(e) \subseteq A, \\ \forall o \in \mathsf{Mand}_{\mathsf{Q}}(x,e) \setminus \{\mathsf{Ext}, \mathsf{NR}\},\ e \in \mathsf{Reach}^{\mathsf{mode}_o, N_o(x,e)}_{o,A}(\mathsf{Tgt}_o(x)) \end{array} \right\}.$$

The finite audit-adequate relief kernel is $K^{\mathrm{rel}}_{\mathsf{Q}}(x) = \nu A.\mathcal{F}_x(A)$.

Definition 4.3 (Audit-stable refinement successor)**.** A refinement step from e to $\tilde{e}$ for x is audit-stable when it is represented by a refined protocol $\tilde{\mathsf{Q}}$, a simulation relation Sim, and maps on raw records, registers, diagnostics, and hypergraphs such that:

(i) $(e,\tilde{e}) \in \mathsf{Sim}$ and every $\tilde{\mathsf{Q}}$ transition from $\tilde{e}$ simulates a Q transition or is recorded as a new audit obligation;

(ii) every raw record in the refined boundary reflects to an old raw record or carries an explicit new witness obligation;

(iii) open hypergraph closure in the refinement reflects open closure in the original protocol or is marked as a new regeneration edge.

The set of such successors is denoted $S_x^{\mathrm{ref}}(e)$.

Definition 4.4 (Non-resurrection closure operator)**.** Let $S_x^{\mathrm{tr}}(e)$ be the ordinary transition successors of e admitted by $\mathcal{T}$. Let $S_x^{\mathrm{desc}}(e)$ be the descendant-process successors generated by the finite descendant rule family in $\mathcal{R}$ from records, residues, or certificates preserving an x-indexed trace; in the finite regime each rule has finite output, so $S_x^{\mathrm{desc}}(e)$ is finite. The audit-admissible successor system is the finite union

$$S_x(e) = S_x^{\mathrm{tr}}(e) \cup S_x^{\mathrm{desc}}(e) \cup S_x^{\mathrm{ref}}(e).$$

Define

$$\mathcal{N}_x(A) = \{e \in \mathsf{Tgt}_{\mathsf{Ext}}(x) \cap B_x : S_x(e) \subseteq A\}.$$

The finite non-resurrection closure is

$$C_{\mathsf{Q}}^{\mathrm{nr}}(x) = \nu A. \mathcal{N}_x(A).$$

Definition 4.5 (Kernel-support successor)**.** For a fixed Markov kernel K on the declared topology, the transition-support successor set is

$$S_x^K(e) = \{e' : K(U \mid e) > 0 \text{ for every measurable neighborhood } U \ni e'\}.$$

In a finite state space this is $S_x^K(e) = \{e' : K(e, e') > 0\}$. A set D is kernel absorbing when $S_x^K(d) \subseteq D$ for every $d \in D$, equivalently $K(D \mid d) = 1$ in the finite Markov-chain setting.

Definition 4.6 (Kernel-audit adequacy)**.** A Markov kernel K is audit-reflected for x when every positive-support transition is represented by an ordinary audit transition:

$$S_x^K(e) \subseteq S_x^{\mathrm{tr}}(e) \quad \text{for all audited } e.$$

It is audit-complete on a set A when, conversely, every ordinary audit transition in A is either in $S_x^K(e)$ or is labelled as a zero-probability but audit-required nondeterministic successor. Kernel-audit adequacy means audit-reflection plus audit-completeness on the certified window. Only audit-reflection is used to infer Markov absorption from audit closure. Audit-completeness is a separate condition for claiming that the audit successor system contains all protocol-relevant future obligations, including zero-probability but audit-required branches.

Definition 4.7 (Continuous-time support)**.** In the continuous-time audit regime, support is not inferred from a single one-step kernel unless the protocol declares an audit skeleton. For a cadlag law on $[0, T]$ with lifetime ζ and $\tau_D = \inf\{t : X_t \in D\}$, define the finite-horizon hitting and post-hit safety events

$$\mathsf{Hit}_D^T(e) = \{\omega : X_0 = e,\ \tau_D \le T,\ \tau_D < \zeta\},$$

$$\mathsf{PostSafe}_D^T(e) = \{\omega : X_0 = e,\ X_t \in D \text{ for every audited } t \in [\tau_D, T] \text{ whenever } \tau_D \le T < \zeta\}.$$

The stopped non-resurrection event is

$$\mathsf{SafePath}_D^T(e) = \mathsf{Hit}_D^T(e) \cap \mathsf{PostSafe}_D^T(e).$$

For a skeleton Skel, support successors are the kernel-support successors of the inter-skeleton kernels. Continuous-time non-resurrection claims below are interpreted only through one of these two declared supports.

Definition 4.8 (Controlled extinction game and attractor)**.** A finite controlled extinction game for x is a finite lifted state space $E^\sharp$, a projection $\pi^\sharp : E^\sharp \to \mathcal{E}_{\mathsf{Q}}$, a lifted target $D^\sharp_x \subseteq E^\sharp$, a controller action set $\mathsf{Ctrl}_x(e)$, and scheduler successor sets $\mathsf{Succ}_x(e,a) \subseteq E^\sharp$. The lifted burden-bounded region is

$$B^\sharp_x = \{e^\sharp \in E^\sharp : \pi^\sharp(e^\sharp) \in B_x\}.$$

The lifted target satisfies $\pi^\sharp(D^\sharp_x) \subseteq C^{\mathrm{nr}}_{\mathsf{Q}}(x)$. Finite-game kernels and certificates are subsets of $E^\sharp$; the concrete non-resurrection meaning is obtained only by projection through $\pi^\sharp$. Define

$$A_0 = D^\sharp_x,$$

$$A_{n+1} = A_n \cup \left\{ e \in B^\sharp_x : \exists a \in \mathsf{Ctrl}_x(e),\ \mathsf{Succ}_x(e,a) \subseteq A_n \right\}.$$

The complete-extinction kernel is

$$K^{\mathrm{ext}}_{\mathsf{Q}}(x) = \bigcup_{n\geq 0} A_n.$$

Definition 4.9 (Liberation kernel and certificate)**.** In a finite controlled extinction game, define lifted witness predicates on $E^\sharp$ by

$$\mathsf{DebtAdm}^\sharp_x(e^\sharp) \iff \mathsf{DebtAdm}^{\mathrm{ext}}_x(\pi^\sharp(e^\sharp)), \quad B^\sharp_{\mathrm{queue}}(x,e^\sharp) = B_{\mathrm{queue}}(x,\pi^\sharp(e^\sharp)),$$

and let $\mathsf{Disch}^{\sharp,\mathrm{open}}_x(e^\sharp)$ be the residual-status-register version of $\mathsf{Disch}^{\mathrm{open}}_x$ on the lifted state. The seven OK predicates are:

$$\begin{array}{lcl}
\mathsf{DebtOK}_x(e^\sharp) & \iff & \mathsf{DebtAdm}^\sharp_x(e^\sharp),\\
\mathsf{QueueOK}_x(e^\sharp) & \iff & B^\sharp_{\mathrm{queue}}(x,e^\sharp) \leq \beta_{\mathrm{queue}} \text{ and a queue-quench witness is present},\\
\mathsf{OpenOK}_x(e^\sharp) & \iff & \mathsf{Disch}^{\sharp,\mathrm{open}}_x(e^\sharp) \text{ and lifted open closure avoids the active domain},\\
\mathsf{PartOK}_x(e^\sharp) & \iff & \text{the audit-critical partition witness is present},\\
\mathsf{RefOK}_x(e^\sharp) & \iff & \text{the refinement-simulation and reflection witness is present},\\
\mathsf{FailOK}_x(e^\sharp) & \iff & e^\sharp \neq \perp,\\
\mathsf{BdryOK}_x(e^\sharp) & \iff & e^\sharp \text{ lies inside the closed audit boundary.}
\end{array}$$

The finite bad-state predicates are the corresponding complements inside $E^\sharp$:

$$\mathsf{Bad}^{\mathrm{debt}}_x,\ \mathsf{Bad}^{\mathrm{queue}}_x,\ \mathsf{Bad}^{\mathrm{open}}_x,\ \mathsf{Bad}^{\mathrm{part}}_x,\ \mathsf{Bad}^{\mathrm{ref}}_x,\ \mathsf{Bad}^{\mathrm{fail}}_x,\ \mathsf{Bad}^{\mathrm{bdry}}_x \subseteq E^\sharp.$$

Thus $\mathsf{Bad}^{\mathrm{debt}}_x = E^\sharp \setminus \mathsf{DebtOK}_x$, $\mathsf{Bad}^{\mathrm{queue}}_x = E^\sharp \setminus \mathsf{QueueOK}_x$, and similarly for open residue, partition, refinement, failure, and boundary conditions. A protocol extension may add a finite family $\mathsf{Bad}^{\mathrm{add}}_x$ only when each added predicate is measurable on $E^\sharp$, obligation-reflecting, and names the audit obligation whose missing witness it records. Define

$$\mathsf{Safe}_x = B^\sharp_x \cap \mathsf{DebtOK}_x \cap \mathsf{QueueOK}_x \cap \mathsf{OpenOK}_x \cap \mathsf{PartOK}_x \cap \mathsf{RefOK}_x \cap \mathsf{FailOK}_x \cap \mathsf{BdryOK}_x \setminus \bigcup \mathsf{Bad}^{\mathrm{add}}_x,$$

with $\bigcup \mathsf{Bad}^{\mathrm{add}}_x = \varnothing$ when no extension predicate is declared. For $A \subseteq E^\sharp$, set

$$\mathsf{SafePred}_x(A) = \{e \in \mathsf{Safe}_x : \exists a \in \mathsf{Ctrl}_x(e),\ \mathsf{Succ}_x(e,a) \subseteq A\}.$$

Let

$$L_0 = D^\sharp_x, \qquad L_{n+1} = L_n \cup \mathsf{SafePred}_x(L_n).$$

The finite liberation kernel is the safe attractor

$$K_{\mathsf{Q}}^{\mathrm{lib}}(x) = \bigcup_{n \geq 0} L_n.$$

A raw liberation certificate for x at state e is a finite tuple

$$L = (P_L, \sigma, \rho, C, \Theta)$$

with typed components

$$P_L \subseteq E^\sharp, \quad e \in P_L, \quad \sigma : P_L \setminus C \to \bigcup_{u \in P_L} \mathsf{Ctrl}_x(u), \quad \rho : P_L \to \{0, \dots, |E^\sharp|\}, \quad C \subseteq D_x^\sharp.$$

Here P_L is a finite certified prefix cone, $\sigma(u) \in \mathsf{Ctrl}_x(u)$ is required for each $u \in P_L \setminus C$, ρ is a finite rank, C is the successor-closed invariant reached by σ, and

$$\Theta = (\Theta_{\mathrm{debt}}, \Theta_{\mathrm{queue}}, \Theta_{\mathrm{open}}, \Theta_{\mathrm{part}}, \Theta_{\mathrm{ref}}, \Theta_{\mathrm{fail}}, \Theta_{\mathrm{end}})$$

is a typed finite witness ledger. The entries are finite maps on the certified prefix cone: Θ_{debt} assigns closure, safe transfer, or non-regenerative certificate witnesses to active debt; Θ_{queue} assigns queue-quench witnesses; Θ_{open} assigns open-residue discharge witnesses; Θ_{part} records audit-critical partition separation; Θ_{ref} records refinement simulation and reflection witnesses; Θ_{fail} records exclusion of $\mathsf{Bad}_x^{\mathrm{fail}}$; and Θ_{end} records that no scheduler-maintainable avoiding end component remains inside the certified cone.

The predicate $\mathsf{RawLibCert}_{\mathsf{Q}}(x, e, L)$ holds when $P_L \setminus C \subseteq \mathsf{Safe}_x$, every successor in $\mathsf{Succ}_x(u, \sigma(u))$ lies in P_L with strictly smaller rank for each $u \in P_L \setminus C$, C is successor-closed and contained in $D_x^\sharp$, and the ledger entries validate all OK predicates and added bad-predicate exclusions on P_L. No kernel-membership premise is part of this predicate. Finally, $\mathsf{Liber}_x(e) \iff e \in K_{\mathsf{Q}}^{\mathrm{lib}}(x)$.

Definition 4.10 (Typed crystal update algebra)**.** The candidate-update component $\mathcal{U}$ of a protocol contains four typed finite-witness update families

$$U^{\mathrm{grow}}, \quad U^{\mathrm{diss}}, \quad U^{\mathrm{res}}, \quad U^{\mathrm{recrys}}.$$

Each family is a measurable relation or partial kernel on typed pairs (z, e), with output (z', e'). Its type is an element of the finite disjoint union of source and target sorts generated by TailSort, HeadSort, and ActSort; the corresponding update-label space is finite in the finite regime and standard Borel in the standard Borel regime. The family U^{grow} adds a protocol-visible structure component. The family U^{diss} removes or redissolves one while writing the resulting residue into the residue register in the register mode declared by Q. The family U^{res} composes declared non-regenerative residues. The family U^{recrys} reuses a certified residue and is admissible only when the post-update open closure remains disjoint from the active domain:

$$\mathsf{FCl}_x(R_x(e') \cup \{z'\}) \cap \mathsf{Act}_x(e') = \varnothing.$$

A typed update is admissible only when its source and target sorts match the schema signatures, its cost and delay fit the relevant witness budget, its register writes preserve the declared register mode, its output remains debt-admissible, and the displayed non-regeneration condition holds for members of U^{recrys}.

Definition 4.11 (Active-domain profile and candidate reduction)**.** For any candidate or structure y, the active-domain node set at e is

$$\mathsf{Act}_y(e) = \bigcup_{r \in \mathsf{Bdry}_{\mathsf{Q}}(e,y)} \mathcal{J}^{\mathsf{Q}}_{\mathrm{act}}(r,e) \subseteq \mathsf{ActNode}_{\mathsf{Q}}.$$

The associated active-configuration set is

$$\mathsf{ActCfg}_y = \{e \in \mathcal{E}_{\mathsf{Q}} : \mathsf{Act}_y(e) \neq \varnothing\}.$$

Its active-threshold status is the finite vector

$$\mathsf{ActStatus}_e(y) = \big(\mathbf{1}\{\theta_{\mathrm{act},j}(y,e) \leq \beta_{\mathrm{act},j}\}\big)_j$$

over the active-domain thresholds declared by Q. The mandatory witness profile is

$$\mathsf{WitProf}_e(y) = \big(\mathsf{Mand}_{\mathsf{Q}}(y,e), (\mathsf{mode}_o, \mathsf{Tgt}_o(y), N_o(y,e), C_o(y,e), T_o(y,e))_{o \in \mathsf{Mand}_{\mathsf{Q}}(y,e)}, J_{\mathrm{wit}}(y,e)\big),$$

where $J_{\mathrm{wit}}(y,e)$ records which mandatory witness coordinates are used by those obligations. The active-domain profile of a candidate s at e is

$$\mathsf{ActProf}_e(s) = (\mathsf{Act}_s(e), \mathsf{ActStatus}_e(s)).$$

Candidates s and s' are active-domain equivalent at e, written $s \equiv_{\mathsf{Act},e} s'$, when their active-domain profiles are equal. A witness-preserving candidate reduction is a relation $s' \lhd_e s$ such that $s' \equiv_{\mathsf{Act},e} s$, $\mathsf{WitProf}_e(s') = \mathsf{WitProf}_e(s)$, its burden vector is componentwise no larger, and

$$D_{\mathsf{Q}}(e,s') \prec_{\mathrm{R}} D_{\mathsf{Q}}(e,s).$$

The relation is evaluated on the finite candidate set in the finite regime and on the declared analytic candidate graph in the standard Borel regime.

Definition 4.12 (Redissolution interval and irreducibility)**.** A candidate s at (e, Q) has a viable redissolution interval when the protocol records

$$\rho_{\mathrm{ord}}(s,e) < R_{\mathrm{D}}(s,e) < \rho_{\mathrm{block}}(s,e)$$

and supplies a finite U^{diss} witness from the crystallized state to a burden-bounded residue state. When a protocol records residue or load budgets for a seed, this interval is a seed-admissibility requirement. A seed is irreducible at e when there is no seed s' with $s' \lhd_e s$.

Definition 4.13 (Meaning seed and crystal under extinction priority)**.** A candidate s is a meaning seed at e, written $\mathsf{Seed}(s; e, \mathsf{Q})$, when there exists an admitted update from e to e', written either $e \xrightarrow{\mathrm{rel}}_{\mathsf{Q},s} e'$, $e \xrightarrow{\mathrm{pos}}_{\mathsf{Q},s} e'$, or $e \xrightarrow{\mathrm{wit}}_{\mathsf{Q},s,o} e'$ for some obligation o, such that

$$D_{\mathsf{Q}}(e',s) \prec_{\mathrm{R}} D_{\mathsf{Q}}(e,s),$$

no mandatory witness coordinate is worsened, the update satisfies $\mathsf{DebtAdm}^{\mathrm{rel}}_s(e')$, every mandated load-quench or access obligation for s is witnessed in the selected predecessor mode, and the redissolution interval is viable for every seed whose load, residue, or crystal budget records redissolution. If $\mathsf{Ext} \notin \mathsf{Mand}_{\mathsf{Q}}(s,e')$, then $e' \in K^{\mathrm{rel}}_{\mathsf{Q}}(s)$ is required. If $\mathsf{Ext} \in \mathsf{Mand}_{\mathsf{Q}}(s,e')$, then $\mathsf{DebtAdm}^{\mathrm{ext}}_s(e')$ and $e' \in K^{\mathrm{ext}}_{\mathsf{Q}}(s)$ are required.

A meaning crystal, written $\mathsf{Crystal}(c; e, \mathsf{Q})$, is a finite derivation

$$c = ((z_0, e_0), u_0, (z_1, e_1), \ldots, u_{m-1}, (z_m, e_m)), \qquad e_0 = e,$$

where z_0 is either a seed satisfying $\mathsf{Seed}(z_0; e_0, \mathsf{Q})$ or a declared non-regenerative residue. Each u_i belongs to one of U^{grow}, U^{diss}, U^{res}, or U^{recrys} and maps (z_i, e_i) to (z_{i+1}, e_{i+1}) with the declared type, cost, delay, and debt witnesses. Each prefix state e_{i+1} lies in $K^{\mathrm{ext}}_{\mathsf{Q}}(z_{i+1})$ if $\mathsf{Ext} \in \mathsf{Mand}_{\mathsf{Q}}(z_{i+1}, e_{i+1})$, otherwise in $K^{\mathrm{rel}}_{\mathsf{Q}}(z_{i+1})$. The finite derivation and its update labels are part of the witness.

Definition 4.14 (Lock-in, false relief, and false resurrection)**.** A candidate or crystal is lock-in at e when it gives strict short-horizon diagnostic improvement but the post-update configuration is outside the required relief or complete-extinction kernel. False resurrection is a refinement or descendant transition that appears harmless in the quotient but creates an open forward closure from certified residues, records, keys, or descendants back to $\mathsf{Act}_x(e')$ at the post-transition configuration.

5 Nucleation and Stochastic Loop Liberation

Definition 5.1 (Candidate proposal and seed certification)**.** Candidate appearance is represented by a proposal kernel $G_t(ds \mid e)$ or finite proposal relation. Seed certification is the indicator

$$\mathsf{Cert}(s, e) = \mathbf{1}\{\mathsf{Seed}(s; e, \mathsf{Q})\}.$$

Candidate proposal and seed certification are distinct: a proposal can occur without producing a seed.

Definition 5.2 (Barrier hazard)**.** The protocol may supply normalized measurable coordinates $\mathsf{Sup}(e, t)$, $\mathsf{Ten}(e, t)$, $\mathsf{Pert}(e, t)$, a barrier $\mathsf{Bar}(e, t) \geq 0$, and either a deterministic measurable fluctuation $\xi(e, t)$ or a stochastic process ξ_t adapted to the protocol filtration. The coordinates are dimensionless protocol diagnostics; no physical unit identity is asserted. A local effective barrier coordinate is

$$\Delta(e, t) = \mathsf{Bar}(e, t) - a\mathsf{Sup}(e, t) - b\mathsf{Ten}(e, t) - c\mathsf{Pert}(e, t) - d\xi(e, t),$$

with nonnegative scales. A hazard is

$$\lambda_t(e) = \lambda_0(t)\rho(\Delta(e, t), \Theta(t)), \qquad \Theta(t) > 0,$$

where ρ is nonincreasing in the barrier argument and measurable in all displayed variables. Arrhenius and logistic forms are admitted only when specified by the protocol; the crystallization analogy is used only through this barrier-and-hazard interface [7, 8].

Definition 5.3 (Stochastic non-resurrection recursion)**.** Let E be finite, $D = C^{\mathrm{nr}}_{\mathsf{Q}}(x)$, and $B \subseteq E$ a burden-bounded region. A finite controlled stochastic substrate consists of finite controller actions $A(e)$, finite scheduler actions $G(e, a)$, and a kernel $K(e' \mid e, a, g)$. Add an absorbing failure sink $\perp$ for every transition leaving $B \setminus D$. A triple $(p, f, h) \in [0, 1] \times [0, 1] \times [0, \infty]$ means success probability at least p, failure probability at most f, and expected burden at most h.

Let $\succeq_{\mathsf{Tri}}$ be the strong-guarantee order

$$(p, f, h) \succeq_{\mathsf{Tri}} (p', f', h') \quad \iff \quad p \geq p', \quad f \leq f', \quad h \leq h'.$$

For $S \subseteq [0, 1] \times [0, 1] \times [0, \infty]$, define the weakening closure

$$\mathsf{WeakCl}_{\mathsf{Tri}}(S) = \{t : \exists s \in S,\ s \succeq_{\mathsf{Tri}} t\}.$$

Define weakening-closed sets $\mathcal{G}_n(e)$ by

$$\mathcal{G}_0(e) = \begin{cases} \mathsf{WeakCl}_{\mathsf{Tri}}\{(1,0,0)\}, & e \in D, \\ \mathsf{WeakCl}_{\mathsf{Tri}}\{(0,1,+\infty)\}, & e = \perp \text{ or } e \notin B, \\ \mathsf{WeakCl}_{\mathsf{Tri}}\{(0,0,0)\}, & e \in B \setminus D, \end{cases}$$

where $\mathsf{WeakCl}_{\mathsf{Tri}}$ closes strong guarantees under weaker requested guarantees. The failure-sink base set contains no triple with $p > 0$, no triple with $f < 1$, and no triple with finite h. For $e \in B \setminus D$, let $\mathsf{Gen}_{n+1}(e)$ be the set of all triples (p, f, h) for which there exists $a \in A(e)$ such that for every $g \in G(e, a)$ there are choices $(p_{e'}, f_{e'}, h_{e'}) \in \mathcal{G}_n(e')$ satisfying

$$p \leq \sum_{e'} K(e' \mid e, a, g) p_{e'}, \quad f \geq \sum_{e'} K(e' \mid e, a, g) f_{e'}, \quad h \geq c(e, a, g) + \sum_{e'} K(e' \mid e, a, g) h_{e'}.$$

Set

$$\mathcal{G}_{n+1}(e) = \mathsf{WeakCl}_{\mathsf{Tri}}(\mathsf{Gen}_{n+1}(e)) \quad \text{for } e \in B \setminus D.$$

For $e \in D$, $e = \perp$, or $e \notin B$, the displayed base clauses are absorbing. A strict stochastic certificate at horizon N is the existence of $(p, f, h) \in \mathcal{G}_N(e)$ with

$$p \geq \alpha, \qquad f = 0, \qquad h \leq C.$$

A practical stochastic certificate with declared tolerance ε replaces $f = 0$ by $f \leq \varepsilon$. Thus the robust quantifier order is built into the set-valued recursion: one controller action must work for every scheduler action at each step.

Remark 5.4 (Pure actions). The recursion uses finite pure controller and scheduler actions. Randomized actions are admitted only when the protocol explicitly replaces the action sets by declared convexified action spaces and preserves the same success, failure, and burden coordinates.

Definition 5.5 (Regular path-law recursion)**.** In the compact Feller regime, let $K_a(de' \mid e)$ be a controlled kernel, D a closed non-resurrection set, B a Borel burden-bounded region, $c \geq 0$ a locally bounded Borel burden, and $A(e)$ the compact admissible-action set. For a fixed measurable selector σ, finite-horizon hitting probabilities and expected burdens are defined by

$$p_0(e) = \mathbf{1}_D(e), \qquad p_{n+1}(e) = \mathbf{1}_D(e) + \mathbf{1}_{B\setminus D}(e) \int p_n(e') K_{\sigma(e)}(de' \mid e),$$

$$h_0(e) = 0, \qquad h_{n+1}(e) = \mathbf{1}_{B\setminus D}(e) \left(c(e, \sigma(e)) + \int h_n(e') K_{\sigma(e)}(de' \mid e) \right).$$

A companion failure recursion is obtained by replacing p_0 by the indicator of $W \setminus B$ or of a declared failure sink. Controller optimization over $A(e)$ is used only when the theorem states the required measurable-selector hypotheses.

Definition 5.6 (Continuous-time audit path law)**.** In the continuous-time audit regime, let $(X_t)_{t\geq 0}$ be a cadlag Markov family with lifetime ζ, and let $\tau_D = \inf\{t : X_t \in D\}$ for a non-resurrection set D. Certification on $[0, T]$ uses the cadlag law on the audited horizon; burden accounting after hitting may be expressed by the stopped process $(X_{t\wedge\tau_D\wedge\zeta})_{0\leq t\leq T}$. The certificate is admissible only when $\Pr_e(\zeta \leq T,\ \zeta < \tau_D) = 0$, or when the protocol replaces the path by a discrete audit skeleton whose inter-skeleton kernels are explicitly declared. The assertion that D is non-resurrecting in continuous time on $[0, T]$ means either that the skeleton kernels hit D and are absorbing in D at audited skeleton times, or that the cadlag law assigns probability one to $\mathsf{SafePath}^T_D(e)$ on the audited horizon. A pure post-hit safety claim omits $\mathsf{Hit}^T_D(e)$ and certifies only conditional non-resurrection after hitting.

6 Main Results

Finite Kernels and Liberation

Proposition 6.1 (Crystal derivation soundness)**.** *Let c be a meaning crystal derivation from e under Q. Then every prefix state of c satisfies the kernel required by its current obligation: $K^{\mathrm{rel}}_{\mathsf{Q}}(z_i)$ when no extinction request for z_i is mandated, and $K^{\mathrm{ext}}_{\mathsf{Q}}(z_i)$ when an extinction request is mandated. In particular, a derivation cannot certify crystallization by passing through an unbounded load, unresolved active-continuation debt, or open regeneration route outside the relevant kernel.*

Proof. The base prefix is either a certified seed or a declared non-regenerative residue. The seed clause requires debt admissibility, load-quench witnesses when mandated, and membership in the appropriate kernel after the seed update. For the induction step, each typed update in U^{grow}, U^{diss}, U^{res}, or U^{recrys} is admissible only with its displayed type, budget, register-mode, and debt witnesses. Recrystallization updates additionally require post-update open-closure disjointness from the active domain. The definition of $\mathsf{Crystal}$ explicitly requires the resulting prefix state to lie in K^{rel} or K^{ext} according to the extinction-request clause. Induction over the finite derivation proves the assertion. □

Proposition 6.2 (Cessation stratification)**.** *For every indexed instance and configuration e,*

$$\mathsf{NRExt}_x(e) \Longrightarrow \mathsf{Diss}_x(e) \Longrightarrow \mathsf{Cess}_x(e).$$

Neither converse is valid in finite audit systems. There is a finite system with $\mathsf{Cess}_x(e)$ but not $\mathsf{Diss}_x(e)$, and a finite system with $\mathsf{Diss}_x(e)$ but not $\mathsf{NRExt}_x(e)$.

Proof. The first implication follows from the displayed clauses defining $\mathsf{NRExt}_x(e)$, and the second from the displayed clauses defining $\mathsf{Diss}_x(e)$. For the first converse failure, take one inactive state e with $\mathsf{Act}_x(e) = \varnothing$ and a positive queue burden above β_{queue}; then cessation holds but dissipative quench fails. For the second, take one inactive state with a declared residue r, an open hyperedge $\{r\} \to a$, and $a \in \mathsf{Act}_x(e)$; the residue is declared and cessation is dissipative except for non-resurrection, but $\mathsf{FCl}_x(\{r\})$ meets $\mathsf{Act}_x(e)$, so $\mathsf{NRExt}_x(e)$ fails. □

Proposition 6.3 (Open-residue discharge obstruction)**.** *Assume an extinction request for x is mandated. If some $r \in R^{\mathrm{open}}_x(e)$ has no finite erasure witness below the regeneration threshold, no transfer witness to a process whose regeneration closure avoids $\mathsf{Act}_x(e)$, and no non-regenerative certificate, then $e \notin \mathsf{Tgt}_{\mathsf{Ext}}(x)$. Consequently $e \notin C^{\mathrm{nr}}_{\mathsf{Q}}(x)$ and no complete-extinction or liberation certificate starts at e without first discharging r.*

Proof. The target requires $\mathsf{Disch}^{\mathrm{open}}_x(e)$ and open-closure disjointness. The displayed r fails every allowed discharge form, so the target predicate fails before any attractor or successor-closure argument is applied. Since $C^{\mathrm{nr}}_{\mathsf{Q}}(x)$ is a subset of $\mathsf{Tgt}_{\mathsf{Ext}}(x)$, the state is outside the non-resurrection closure. A liberation certificate must eventually enter a successor-closed subset of that closure while remaining in the certified safe region, so it cannot start from e unless a prior certified transition discharges r. □

Lemma 6.4 (Residual-status lifting)**.** *Assume a finite audit window and a fixed finite ambient residue register ResAmb^I_x. Every transition system in which open-residue status can change over time has a finite lift*

$$E^\flat = E \times \{0, 1, 2, 3\}^{\mathsf{ResAmb}^I_x}$$

such that discharge, erasure, safe transfer, and non-regenerative certification are ordinary successor-state predicates on $E^\flat$. Projection to E preserves the underlying process transition, while the lifted status coordinate preserves whether an open residue remains unresolved.

Proof. The ambient register is finite, so the product lift is finite. Each transition of the original system is paired with the protocol-declared status update for each ambient residue token. The first projection is the original transition. Since the second coordinate records all four allowed statuses for every token, whether a residue is unresolved, erased, safely transferred, or certified is evaluated at one lifted state rather than inferred from a time-dependent family. ☐

Proposition 6.5 (Open-residue completion)**.** *In a finite audit window with fixed ambient residual-status lift $E^\flat$, the open-residue obstruction for x is absent at $e^\flat$ if and only if every currently open residue token has a finite erasure, safe-transfer, or non-regenerative-certificate witness in the residual-status register and the resulting audited family has open forward closure disjoint from $\mathsf{Act}_x(e)$.*

Proof. If the obstruction is absent, $\mathsf{Disch}_x^{\mathrm{open}}(e^\flat)$ holds and the target clause supplies exactly one of the three finite witness forms for every open token; the target also requires open-closure disjointness. Conversely, if every open token carries one of the three witnesses and the resulting family avoids $\mathsf{Act}_x(e)$ under FCl_x, then the two clauses that can fail solely because of open residue, discharge and closure disjointness, both hold. Thus open residue contributes no remaining obstruction to target membership. Finiteness of the window makes the witness family finite. ☐

Proposition 6.6 (Finite active-domain irreducible seed)**.** *Assume the finite regime and fix an active-domain equivalence class of candidates at e. If this class contains at least one seed satisfying the specified mandatory witness profile and burden bounds, then it contains a seed that is minimal for the candidate-reduction relation $\lhd_e$.*

Proof. Let S be the nonempty finite set of seeds in the fixed active-domain equivalence class satisfying the specified witness profile and burden bounds. Restrict $\lhd_e$ to S. If every element of S had a predecessor under $\lhd_e$, repeatedly choosing predecessors would produce an infinite descending sequence in a finite set, hence a cycle. A cycle is impossible because each reduction requires strict diagnostic improvement in the quotient preorder. Therefore some element has no $\lhd_e$-predecessor and is an irreducible seed. ☐

Theorem 6.7 (Non-resurrection closure)**.** *Let E be a finite audit-saturated transition system. Then $C^{\mathrm{nr}}_{\mathsf{Q}}(x) = \nu A.\mathcal{N}_x(A)$ is the largest subset of $\mathsf{Tgt}_{\mathsf{Ext}}(x)$ such that every audit-admissible future transition, descendant, and audit-stable refinement successor remains in the same subset. If $e \in C^{\mathrm{nr}}_{\mathsf{Q}}(x)$, then no finite audit-admissible future path from e reaches ActCfg_x.*

Proof. The operator $\mathcal{N}_x$ is monotone on the finite lattice $\mathcal{P}(E)$ because $A \subseteq A'$ and $S_x(e) \subseteq A$ imply $S_x(e) \subseteq A'$. Finite descending iteration from E gives its greatest fixed point. The fixed point is contained in $\mathsf{Tgt}_{\mathsf{Ext}}(x)$ and is successor-closed by construction. Conversely, any successor-closed subset of $\mathsf{Tgt}_{\mathsf{Ext}}(x)$ is post-fixed for $\mathcal{N}_x$ and is contained in the greatest fixed point. Since every state in $\mathsf{Tgt}_{\mathsf{Ext}}(x)$ has open forward closure disjoint from $\mathsf{Act}_x(e)$ and every successor remains in the fixed point, induction on path length excludes finite audit-admissible resurrection. ☐

Theorem 6.8 (Complete extinction with non-resurrection)**.** *Let $E^\sharp$ be a finite controlled extinction game. The set $K^{\mathrm{ext}}_{\mathsf{Q}}(x)$ is exactly the set of states from which the controller has a finite-budget strategy forcing every scheduler successor path into $D^\sharp_x$ while remaining in $B^\sharp_x$. The attractor iteration stabilizes in at most $|E^\sharp|$ strict growth steps. Its complement is the greatest fixed point of the scheduler's avoid-non-resurrection operator.*

Proof. By induction, A_n is the set of states from which the controller can force entrance into $D_x^\sharp$ in at most n game steps. The induction step is exactly the existence of a controller action whose every scheduler successor is already in A_n. Finiteness gives stabilization after at most $|E^\sharp|$ strict additions. The complement consists of states from which the scheduler can keep the play outside the attractor for every controller action, which is the greatest avoid-non-resurrection fixed point. □

Proposition 6.9 (Lifted target projection soundness)**.** *Let $D_x^\sharp \subseteq E^\sharp$ be the lifted target of a finite controlled extinction game and assume $\pi^\sharp(D_x^\sharp) \subseteq C_{\mathsf{Q}}^{\mathrm{nr}}(x)$. If a finite lifted path enters $D_x^\sharp$, then its projected terminal configuration lies in the concrete non-resurrection closure. If $C \subseteq D_x^\sharp$ is successor-closed in the lifted game, then every projected lifted path that remains in C remains inside $C_{\mathsf{Q}}^{\mathrm{nr}}(x)$ after projection.*

Proof. The first claim is the inclusion $\pi^\sharp(D_x^\sharp) \subseteq C_{\mathsf{Q}}^{\mathrm{nr}}(x)$. For the second, every lifted state on the path lies in $C \subseteq D_x^\sharp$, so every projected state lies in $\pi^\sharp(D_x^\sharp) \subseteq C_{\mathsf{Q}}^{\mathrm{nr}}(x)$. □

Proposition 6.10 (Attractor-end-component dichotomy)**.** *In a finite controlled extinction game, every state is in the complete-extinction attractor $K_{\mathsf{Q}}^{\mathrm{ext}}(x)$ or belongs to the scheduler's greatest avoid-non-resurrection region. From any state in the avoid region, the scheduler has a memoryless choice keeping the play in that region against every controller action. Every infinite play kept in the finite avoid region has a tail contained in an avoiding end component. Hence practical persistence outside the attractor is witnessed by a reachable avoiding end component disjoint from $D_x^\sharp$.*

Proof. The attractor iteration removes exactly the states from which the controller can force the next rank decrease toward $D_x^\sharp$. Its complement is therefore the greatest set in which, for every controller action, the scheduler has a successor remaining in the complement; choosing such a successor gives a memoryless avoiding scheduler. In a finite graph, the set of states visited infinitely often by any infinite play is strongly connected after deleting transient vertices and is closed under the scheduler choices used on the tail. Taking a maximal closed strongly connected subgraph gives an avoiding end component. Conversely, inside an avoiding end component the scheduler chooses internal successors forever, so the play never reaches $D_x^\sharp$. □

Theorem 6.11 (Non-circular liberation certificate)**.** *In a finite controlled extinction game, the liberation kernel $K_{\mathsf{Q}}^{\mathrm{lib}}(x) = \bigcup_n L_n$ generated by the safe-attractor recursion is the least set containing $D_x^\sharp$ such that every state in $K_{\mathsf{Q}}^{\mathrm{lib}}(x) \setminus D_x^\sharp$ lies in Safe_x and has a controller action whose scheduler successors remain in lower rank. The iteration stabilizes in at most $|E^\sharp|$ strict growth steps. Moreover*

$$e \in K_{\mathsf{Q}}^{\mathrm{lib}}(x) \iff \exists L,\ \mathsf{RawLibCert}_{\mathsf{Q}}(x, e, L).$$

Proof. Monotonicity follows from $L_n \subseteq L_{n+1}$ and from the predecessor clause $\mathsf{Succ}_x(e, a) \subseteq L_n$. Since $E^\sharp$ is finite, the increasing sequence stabilizes after at most $|E^\sharp|$ strict additions. The leastness property is induction on construction rank: any set satisfying the displayed closure condition contains $L_0 = D_x^\sharp$ and, if it contains L_n, also contains every state added to L_{n+1}.

If $e \in L_k$, choose a witnessing action for each state at its least construction rank, let P_L be the finite cone of states reachable under these choices before reaching $D_x^\sharp$, set $C = D_x^\sharp \cap P_L$, and let ρ be the least construction rank on P_L. The typed ledger entries are exactly the finite witnesses required by Safe_x at each nonterminal prefix. The rank decreases on every selected scheduler successor until C is reached, so $\mathsf{RawLibCert}_{\mathsf{Q}}(x, e, L)$ holds.

Conversely, assume $\mathsf{RawLibCert}_{\mathsf{Q}}(x, e, L)$. Prove by induction on $\rho(u)$ that every $u \in P_L$ lies in $K_{\mathsf{Q}}^{\mathrm{lib}}(x)$. Rank-zero states lie in the successor-closed invariant $C \subseteq D_x^\sharp = L_0$. If $\rho(u) > 0$, then $u \in \mathsf{Safe}_x$ and every selected scheduler successor has smaller rank, hence belongs to $K_{\mathsf{Q}}^{\mathrm{lib}}(x)$ by induction. Thus u is added by $\mathsf{SafePred}_x$ at a finite stage. Applying this to $u = e$ gives $e \in K^{\mathrm{lib}}_{\mathsf{Q}}(x)$ without assuming kernel membership in the certificate predicate. $\square$

Remark 6.12 (Membership-certified abbreviation). After the equivalence theorem is established, $\mathsf{LibCert}_{\mathsf{Q}}(x, e, L)$ may be used as an abbreviation for $\mathsf{RawLibCert}_{\mathsf{Q}}(x, e, L)$ together with the derived fact $e \in K^{\mathrm{lib}}_{\mathsf{Q}}(x)$. The primitive finite proof object remains $\mathsf{RawLibCert}$.

Proposition 6.13 (Safe-attractor refinement)**.** *For every finite controlled extinction game, $K^{\mathrm{lib}}_{\mathsf{Q}}(x) \subseteq K^{\mathrm{ext}}_{\mathsf{Q}}(x)$. If every OK predicate holds on $B^\sharp_x$ and no added bad predicate is declared, so that $\mathsf{Safe}_x = B^\sharp_x$, then $K^{\mathrm{lib}}_{\mathsf{Q}}(x) = K^{\mathrm{ext}}_{\mathsf{Q}}(x)$.*

Proof. The definition of Safe_x includes the intersection with $B^\sharp_x$, so $\mathsf{Safe}_x \subseteq B^\sharp_x$. The safe predecessor is the complete-extinction predecessor restricted to Safe_x. Induction on the attractor stages gives $L_n \subseteq A_n$ for every n, hence $K^{\mathrm{lib}}_{\mathsf{Q}}(x) \subseteq K^{\mathrm{ext}}_{\mathsf{Q}}(x)$. If $\mathsf{Safe}_x = B^\sharp_x$, the two predecessor clauses are identical, so the two increasing sequences are identical. $\square$

Theorem 6.14 (Safe-attractor obstruction dichotomy)**.** *In the same finite game, every state e satisfies exactly one of the following:*

(i) e carries some finite raw liberation certificate L;

(ii) e is outside the safe attractor, and the scheduler can either keep the play outside $D^\sharp_x$ forever inside an avoiding end component of $E^\sharp \setminus K^{\mathrm{lib}}_{\mathsf{Q}}(x)$, or force entrance into the complement of Safe_x, including $\perp$, against every proposed liberation strategy.

Proof. If $e \in K^{\mathrm{lib}}_{\mathsf{Q}}(x)$, the non-circular certificate theorem supplies a raw certificate and the decreasing rank excludes infinite avoidance or unsafe exit under the certified strategy. If $e \notin K^{\mathrm{lib}}_{\mathsf{Q}}(x)$, then for every controller action either some scheduler successor stays outside $K^{\mathrm{lib}}_{\mathsf{Q}}(x)$ or the action leaves Safe_x. Choosing such successors gives a scheduler strategy. If unsafe exit occurs, the second alternative holds through the failure/complement clause. If the play stays forever in the finite complement, the set of states seen infinitely often contains a closed avoiding end component disjoint from $D^\sharp_x$. These cases are mutually exclusive with certified rank decrease. $\square$

Proposition 6.15 (Almost-sure non-resurrection)**.** *Let (X_n) be a finite Markov chain and let $D = C^{\mathrm{nr}}_{\mathsf{Q}}(x)$ be kernel absorbing for the chain:*

$$S^K_x(d) \subseteq D \quad \textit{equivalently} \quad K(D \mid d) = 1 \quad \textit{for all } d \in D.$$

For an initial state e, $\Pr_e(\tau_D < \infty) = 1$ if and only if every recurrent class reachable from e is contained in D.

Proof. If a recurrent class $C \not\subseteq D$ is reachable from e, then $C \cap D = \varnothing$: if C met D, communication inside C would force a positive-probability path from D to $C \setminus D$, impossible by kernel absorption of D. Hence the chain enters C with positive probability and then never hits D, contradicting almost-sure hitting. Conversely, a finite Markov chain reaches some recurrent class almost surely. If every recurrent class reachable from e is contained in D, then entry into the eventual recurrent class entails hitting D in finite time almost surely. $\square$

Proposition 6.16 (Kernel reflection absorption)**.** *Let $D \subseteq E$ be closed under the audit successor set S_x. If the Markov kernel is audit-reflected for x, then D is kernel absorbing. Without audit-reflection, audit successor closure need not imply kernel absorption; kernel absorption still need not imply closure under descendant or refinement successors.*

Proof. If $d \in D$ and $e' \in S_x^K(d)$, audit-reflection gives $e' \in S_x^{\text{tr}}(d) \subseteq S_x(d)$. Since D is audit-successor closed, $e' \in D$; hence D is kernel absorbing. For failure without audit-reflection, take states $\{d, a\}$ with $D = \{d\}$, no declared audit transition from d to a, and an enriched kernel satisfying $K(a \mid d) = 1/2$. Audit closure holds for the declared successor system, but kernel absorption fails because the positive-support transition is latent to the audit. For the converse separation, take $K(d \mid d) = 1$ and add a refinement successor from d to $a \in \mathsf{ActCfg}_x$. Then D is kernel absorbing but not closed under S_x. ☐

Lemma 6.17 (Portmanteau closed-set step)**.** *Let $\mu_n \Rightarrow \mu$ be weak convergence of probability measures on a metric space. If C is closed and $\mu_n(C) = 1$ for all n, then $\mu(C) = 1$. If $C_n \downarrow C_\infty = \bigcap_n C_n$ is a decreasing sequence of measurable sets, then $\mu(C_n) \downarrow \mu(C_\infty)$. These are the only weak-convergence facts used in the closed Feller kernel and reach-avoid proofs [34].*

Proof. The Portmanteau theorem gives $\mu(C) \geq \limsup_n \mu_n(C) = 1$ for closed C, hence $\mu(C) = 1$. The second claim is continuity from above; the finiteness condition is automatic because μ is a probability measure. ☐

Regular Stochastic Kernels

Theorem 6.18 (Closed Feller non-resurrection kernel)**.** *Assume the compact Feller regime on a compact audit window W. Let $T = W \cap \mathsf{Tgt}_{\mathsf{Ext}}(x)$ be closed and disjoint from ActCfg_x. For closed $C \subseteq W$, define*

$$\mathsf{Pred}_{\text{cl}}(C) = \{e \in T : \exists a \in A(e),\ K(C \mid e, a) = 1\}.$$

Then $\mathsf{Pred}_{\text{cl}}(C)$ is closed. The descending iteration

$$C_0 = T, \qquad C_{n+1} = C_n \cap \mathsf{Pred}_{\text{cl}}(C_n)$$

has a closed limit $C_\infty = \bigcap_{n \geq 0} C_n$, and C_∞ is the largest closed subset of T from which a controller can keep the next state inside the same subset with probability one.

Proof. For closed C, joint weak continuity and the Portmanteau closed-set step imply that the set of admissible pairs with $K(C \mid e, a) = 1$ is closed in the closed action graph. The graph over compact W and compact action space is compact, so its closed subset has closed projection to e. Thus $\mathsf{Pred}_{\text{cl}}(C)$ is closed. Since $C_0 = T$ is closed, induction gives closedness of every C_n.

If $e \in C_\infty$, choose $a_n \in A(e)$ with $K(C_n \mid e, a_n) = 1$. Compactness gives a convergent subsequence $a_{n_j} \to a \in A(e)$, using closedness of the action graph. Since $(e, a_{n_j}) \to (e, a)$, joint weak continuity applies in this fixed-e special case and gives $K(\cdot \mid e, a_{n_j}) \Rightarrow K(\cdot \mid e, a)$. For each fixed m, all sufficiently large n_j satisfy $C_{n_j} \subseteq C_m$, so $K(C_m \mid e, a_{n_j}) = 1$. The Portmanteau step gives $K(C_m \mid e, a) = 1$ for each m. Since $C_m \downarrow C_\infty$ and $K(\cdot \mid e, a)$ is a probability measure, continuity from above gives $K(C_\infty \mid e, a) = 1$. Maximality follows because any closed controlled-invariant subset of T is contained in every C_n by induction. ☐

Theorem 6.19 (Closed Feller invariant selector)**.** *Assume the compact Feller regime on Polish state and action spaces. Let $C \subseteq W$ be a closed controlled-invariant non-resurrection set. Suppose the admissible-action correspondence has Borel graph and nonempty compact values, and that the restricted correspondence below has measurable graph in the standard Borel structure induced by the Polish topology. Then the invariant action correspondence*

$$A_C(e) = \{a \in A(e) : K(C \mid e, a) = 1\}, \qquad e \in C,$$

has nonempty compact values and admits a universally measurable selector. If its graph is Borel, the selector may be chosen Borel under the Kuratowski–Ryll-Nardzewski/Castaing measurable-multifunction hypotheses [38, 25].

Proof. Nonemptiness follows from controlled invariance. Compactness follows because $A_C(e)$ is a closed subset of the compact set $A(e)$; closedness is obtained from joint weak continuity and the Portmanteau closed-set step. The graph is the intersection of the action graph with the measurable condition $K(C \mid e, a) = 1$. The universally measurable conclusion follows from the standard Borel measurable-selection form. When the graph is Borel, the Borel selector conclusion follows from the stated measurable multifunction theorem. □

Proposition 6.20 (Absorbing-kernel non-resurrection invariance)**.** *Let $D \subseteq \mathcal{E}_{\mathsf{Q}}$ be a non-resurrection set with $D \cap \mathsf{ActCfg}_x = \varnothing$. If every audit-admissible controlled kernel satisfies $K_a(D \mid e) = 1$ for all $e \in D$ and all $a \in A(e)$, then every path law starting in D assigns probability one to paths that remain in D at every finite time and never reach ActCfg_x.*

Proof. For one step the assertion is exactly $K_a(D \mid e) = 1$. Iterating the tower property gives $\Pr_e(X_0, \ldots, X_n \in D) = 1$ for every finite n. The countable intersection over n still has probability one. Since D is disjoint from ActCfg_x, active-domain resurrection has probability zero along these audited paths. □

Proposition 6.21 (Corrected analytic predecessor)**.** *Assume the standard Borel analytic regime. Let $G \subseteq E \times A \times E$ be the analytic graph of admissible one-step transitions. For analytic $T \subseteq E$,*

$$\mathsf{Pred}_{\exists}(T) = \{e : \exists a \in A(e),\ \exists e' \in T,\ (e, a, e') \in G\}$$

is analytic, and universally measurable selectors exist for nonempty existential witness sections. For stochastic kernels on standard Borel spaces, finite-horizon robust feasibility is represented by upper semianalytic value functions such as

$$\mathsf{Val}_{n+1}(e) = \sup_{a \in A(e)} \int \mathsf{Val}_n(e') K(de' \mid e, a),$$

with universally measurable ε-selectors. Exact robust selectors are not asserted unless the protocol supplies compactness, closedness, or a selector theorem that makes the argmax sections measurable and nonempty.

Proof. The existential predecessor is the projection of $G \cap (E \times A \times T)$ onto E, hence analytic. Selection for existential witnesses follows from the stated analytic-section hypothesis. For the value recursion, integration of an upper semianalytic function against a Borel kernel and supremum over analytic action sections preserve upper semianalyticity. Standard measurable-selection results give universally measurable ε-maximizers. Without extra compactness or closed graph assumptions, the exact argmax correspondence can be empty or nonmeasurable, so no exact robust selector is claimed. □

Lemma 6.22 (Closed safe approximation)**.** *Assume the compact Feller regime on W. If every burden coordinate used in B_x is lower semicontinuous on W, then replacing strict inequalities by closed sublevel inequalities gives a closed set $B_x^{\mathrm{cl}} \subseteq W$. Any reach-avoid certificate obtained inside B_x^{cl} is also valid for the original burden-bounded region B_x whenever the protocol chooses thresholds with a positive margin.*

Proof. For a lower semicontinuous burden coordinate b, the sublevel set $\{e : b(e) \le c\}$ is closed. A finite intersection of such sublevel sets is closed in W. The positive-margin assumption makes the closed thresholds no larger than the original strict-threshold budget, so membership in B_x^{cl} implies membership in B_x. □

Proposition 6.23 (Finite-horizon Feller reach-avoid)**.** *Assume the compact Feller regime on a compact audit window W. Let $D \subseteq W$ be a closed non-resurrection target and $B \subseteq W$ a closed burden-bounded safe set. Define*

$$\mathsf{RA}_0 = D, \qquad \mathsf{RA}_{n+1} = D \cup \{e \in B : \exists a \in A(e),\ K(\mathsf{RA}_n \mid e, a) = 1\}.$$

Then every RA_n is closed. For each n, a universally measurable selector exists on $\mathsf{RA}_{n+1} \setminus D$ that certifies one-step progress into RA_n. Membership in RA_N is a finite-horizon reach-avoid certificate and does not assert infinite-horizon optimality.

Proof. The base $\mathsf{RA}_0 = D$ is closed. If RA_n is closed, the set of admissible pairs with $K(\mathsf{RA}_n \mid e, a) = 1$ is closed by the Portmanteau closed-set step and joint weak continuity. Intersecting with closed B and projecting from the compact action graph gives a closed predecessor set; union with closed D gives closed RA_{n+1}. The witnessing action sections are nonempty closed subsets of a compact action space on $\mathsf{RA}_{n+1} \setminus D$, and the measurable-selection condition supplies a selector. The final sentence is the finite-horizon definition. □

Theorem 6.24 (Robust stochastic feasibility recursion)**.** *In a finite controlled stochastic substrate with finite controller and scheduler action sets, the set-valued recursion $\mathcal{G}_n$ computes exactly the triples that the controller can guarantee for n steps against all scheduler choices, closed under weakening of the requested guarantee. Hence a strict horizon-N stochastic certificate exists at e if and only if some $(p, f, h) \in \mathcal{G}_N(e)$ satisfies*

$$p \ge \alpha, \qquad f = 0, \qquad h \le C,$$

and a practical certificate with tolerance ε exists if and only if some $(p, f, h) \in \mathcal{G}_N(e)$ satisfies

$$p \ge \alpha, \qquad f \le \varepsilon, \qquad h \le C.$$

Proof. The proof is induction on n. At $n = 0$, the three base cases are exactly target success, failure, and safe non-hit. The failure sink contributes only the weakening closure of $(0, 1, +\infty)$, so it cannot imply positive success, smaller failure probability, or finite burden. For the induction step, a controller action is chosen before the scheduler action, and the displayed inequalities require one triple from $\mathcal{G}_n(e')$ at every successor state for every scheduler action. Linearity of expectation gives the three one-step bounds, and weakening closure records that weaker success, larger failure allowance, or larger burden allowance is still guaranteed. Conversely, any $n+1$ step robust strategy has a first controller action; conditioning on each scheduler action and successor state gives triples in $\mathcal{G}_n(e')$ satisfying the same inequalities. The certificate clauses are therefore exactly membership tests in $\mathcal{G}_N(e)$. □

Proposition 6.25 (Failure coordinate is not hidden by burden)**.** *No scalar expected-burden coordinate can replace the failure coordinate uniformly over all protocols. There are two one-step systems with the same scalar expected burden, one with failure probability $\delta > 0$ and one with failure probability 0, such that only the triple recursion separates strict feasibility from practical feasibility with tolerance at least δ.*

Proof. In the first system, failure occurs with probability δ and the nonfailed branch has burden 0. In the second, failure never occurs and ordinary burden is adjusted so that the scalar expected burden matches the first system. Any certificate seeing only the scalar burden gives the same judgment to both systems. The triple recursion separates them by the second coordinate: the first can satisfy only $f \geq \delta$, whereas the second can satisfy $f = 0$. ☐

Proposition 6.26 (Tolerance monotonicity). *For fixed α, C, N, e, strict feasibility is the case $\varepsilon = 0$ of practical feasibility. If $0 \leq \varepsilon \leq \varepsilon'$, then practical feasibility at tolerance ε implies practical feasibility at tolerance ε'.*

Proof. The practical certificate condition differs from the strict condition only by replacing $f = 0$ with $f \leq \varepsilon$. Setting $\varepsilon = 0$ recovers strict feasibility. If $f \leq \varepsilon$ and $\varepsilon \leq \varepsilon'$, then the same triple satisfies $f \leq \varepsilon'$. ☐

Proposition 6.27 (Ranking certificate for loop liberation). *Let E be finite. Suppose there is a function $V : E \to \{0, \dots, M\}$ such that for every $e \in B_x \setminus C^{\mathrm{nr}}_{\mathsf{Q}}(x)$ the controller has an action a with every scheduler successor $e' \in \mathsf{Succ}_x(e, a)$ satisfying $V(e') < V(e)$. Then the controller forces entry into $C^{\mathrm{nr}}_{\mathsf{Q}}(x)$ or exit from B_x in at most $M + 1$ steps. In a stochastic kernel, let τ be the hitting time of $C^{\mathrm{nr}}_{\mathsf{Q}}(x) \cup (E \setminus B_x)$. If $V(X_{n\wedge\tau})$ is integrable, nonnegative, and satisfies*

$$\mathbb{E}[V(X_{(n+1)\wedge\tau}) \mid \mathcal{F}_n] \leq V(X_{n\wedge\tau}) - \epsilon \mathbf{1}_{\{\tau > n\}}$$

for some $\epsilon > 0$, then $\mathbb{E}[\tau] \leq V(e)/\epsilon$ and the target is reached almost surely.

Proof. In the finite adversarial case, V decreases by at least one natural-number rank at every non-target bounded step, so more than M such steps are impossible. Thus the play reaches $C^{\mathrm{nr}}_{\mathsf{Q}}(x)$ or leaves B_x. In the stochastic case, taking expectations and summing the drift inequality gives $\epsilon\mathbb{E}[n \wedge \tau] \leq V(e)$. Monotone convergence yields $\mathbb{E}[\tau] \leq V(e)/\epsilon$, hence $\Pr(\tau < \infty) = 1$. ☐

Regeneration, Debt, and Refinement

Definition 6.28 (Lifted dynamic regeneration hypergraph). For a finite audited hypergraph with edge set $\mathcal{E}_x$, let

$$E^\sharp = \mathcal{E}_{\mathsf{Q}} \times \{0, 1\}^{\mathcal{E}_x}$$

where the second coordinate records whether each edge is open. A lifted state is $e^\sharp = (e, \chi)$, and the lifted hypergraph $\mathcal{H}^\sharp_{\mathsf{Q}}(x, e^\sharp)$ has the same nodes and edge labels as $\mathcal{H}_{\mathsf{Q}}(x, e)$, with edge a open exactly when $\chi(a) = 1$ and its current threshold predicates are satisfied. Lifted transitions update both e and χ by the protocol's edge-status rule.

Proposition 6.29 (Frozen hypergraph non-resurrection cut). *Let $\mathcal{H}_{\mathsf{Q}}(x, e)$ be finite with edge-open status frozen at the audited configuration e, and let $R_x(e)$ be the post-cessation family of audited residues, recovery keys, load states, descendants, records, and certificates. Then all audited combination-regeneration routes from $R_x(e)$ to $\mathsf{Act}_x(e)$ are cut if and only if*

$$\mathsf{FCl}_x(R_x(e)) \cap \mathsf{Act}_x(e) = \varnothing.$$

Proof. The open forward closure is exactly the set of nodes derivable by finite sequences of open hyperedges whose tails are already derived. If it intersects $\mathsf{Act}_x(e)$, the derivation sequence is an audited combination-regeneration route. If such a route exists, every node on it is generated by the closure rule, so the active node lies in $\mathsf{FCl}_x(R_x(e))$. The two conditions are equivalent. ☐

Proposition 6.30 (Lifted dynamic hypergraph cut)**.** *If edge-open status can change over time, include the edge-status register in the configuration state and form the lifted finite hypergraph* $\mathcal{H}^{\sharp}_{\mathsf{Q}}(x, e^{\sharp})$ *with frozen status at each lifted state. Then non-resurrection along all audited finite paths is equivalent to the frozen cut condition holding at every lifted successor state in the successor-closed non-resurrection closure.*

Proof. In the lifted system, each transition updates both the process configuration and the edge-status register. The frozen cut proposition applies at each lifted state. If the cut holds throughout the successor-closed closure, induction on audited path length prevents any open derivation to the active-node set at the current lifted configuration. Conversely, if some lifted successor violates the frozen cut, the corresponding finite open hyperedge derivation is an audited resurrection route at that state. □

Proposition 6.31 (Recrystallization non-regeneration preservation)**.** *Let a crystal derivation use only recrystallization steps satisfying*

$$\mathsf{FCl}_x(R_x(e') \cup \{z'\}) \cap \mathsf{Act}_x(e') = \varnothing$$

at every post-update prefix. Then no finite prefix of the derivation creates a false-resurrection route from preserved residues, records, keys, load states, or certificates to the active domain.

Proof. The assertion is proved by induction over the finite derivation. The base prefix is non-regenerative by the seed or residue clause. Growth, dissolution, and residue composition steps are admitted only with the required kernel membership and register witnesses. At a recrystallization step, the displayed open-closure disjointness is exactly the absence of a finite hyperedge derivation to the active domain from the updated audited family. Thus the invariant is preserved at each prefix. □

Theorem 6.32 (Regeneration Galois abstraction soundness)**.** *Let* (α, γ) *be a closure-sound finite regeneration abstraction with sound abstract active domain* $\bar{\mathsf{Act}}_x$*. If*

$$\bar{\mathsf{FCl}}_x(\alpha(R_x(e))) \cap \bar{\mathsf{Act}}_x = \varnothing,$$

then $\mathsf{FCl}_x(R_x(e)) \cap \mathsf{Act}_x(e) = \varnothing$*. Thus abstract non-resurrection soundly implies concrete non-resurrection. The converse need not hold for over-approximating abstractions.*

Proof. Closure soundness gives

$$\alpha(\mathsf{FCl}_x(R_x(e))) \subseteq \bar{\mathsf{FCl}}_x(\alpha(R_x(e))).$$

If $\mathsf{FCl}_x(R_x(e))$ met $\mathsf{Act}_x(e)$, active-domain soundness applied to the audited family $\mathsf{FCl}_x(R_x(e))$ would force $\alpha(\mathsf{FCl}_x(R_x(e)))$ to meet $\bar{\mathsf{Act}}_x$. Closure soundness places this abstract image inside $\bar{\mathsf{FCl}}_x(\alpha(R_x(e)))$, contradicting the displayed abstract disjointness. Over-approximation can introduce abstract routes without concrete witnesses, so the converse can fail. □

Proposition 6.33 (Galois update soundness)**.** *Suppose a finite regeneration abstraction is closure-sound, has sound abstract active domain, and every abstract residue composition or recrystallization step is Galois-sound. If the abstract post-update closure avoids* $\bar{\mathsf{Act}}_x$ *at every prefix of an abstract crystal derivation, then every represented concrete derivation avoids the corresponding concrete active-node set* $\mathsf{Act}_x(e_i)$ *at each prefix* e_i*.*

Proof. Let S_i be the concrete audited family at prefix i and $\bar{S}_i$ the abstract family. Galois-soundness gives $\alpha(S_i) \subseteq \bar{S}_i$ after each represented update. Closure soundness then gives $\alpha(\mathsf{FCl}_x(S_i)) \subseteq \bar{\mathsf{FCl}}_x(\bar{S}_i)$. If a concrete active-domain node were reachable, active-domain reflection would force the abstract closure to meet $\bar{\mathsf{Act}}_x$, contradicting the assumed abstract disjointness. Induction over prefixes gives the result. □

Proposition 6.34 (Debt return path induction)**.** *Fix a finite certified path $\gamma = (e_0, \ldots, e_n)$. Suppose every debt token that requires active continuation of x at some prefix has, before the terminal prefix, either a closure witness, a transfer witness to a process whose regeneration closure is disjoint from $\mathsf{Act}_x(e_k)$ at that prefix, or a non-regenerative certificate witness. Suppose also that every deferred return kernel on the path satisfies*

$$B_{\text{return}}(d, \gamma_{\leq k}) \leq \beta_{\text{return}}$$

for every still-deferred debt token d and every prefix k. Then the path is debt-admissible and its recognized return burden remains within the protocol threshold. Conversely, if an active-continuation debt d has no closure, transfer, or non-regenerative certificate on any prefix and every path-dependent return kernel available at every prefix carrying d has burden bounded below by a value exceeding β_{return}, no debt-admissible witness path through that debt exists.

Proof. The first assertion follows by induction over prefixes. At prefix k, a token is either discharged by a closure, moved by a transfer that does not reconstruct x, certified as non-regenerative, or remains deferred with return burden within threshold. Monotonicity keeps any non-discharged token in the ledger, so the induction hypothesis covers the next prefix. For the converse, ledger monotonicity implies that a token with no closure, transfer, or non-regenerative certificate remains present at every prefix on which the path continues. The prefix-wise lower bound applies to the history-dependent kernel at that same prefix and violates the return-threshold clause. Hence no prefix carrying that debt can be part of a debt-admissible witness path. □

Proposition 6.35 (Debt-mode separation)**.** *Ordinary relief debt admissibility does not imply extinction debt admissibility. There is a finite ledger state e such that $\mathsf{DebtAdm}_x^{\text{rel}}(e)$ holds and $\mathsf{DebtAdm}_x^{\text{ext}}(e)$ fails.*

Proof. Take one active-continuation debt token d with return burden below β_{return} and a recorded reconstruction-bearing residue r that is accepted as an ordinary relief deferral. Add one open hyperedge $\{r\} \to a$ with $a \in \mathsf{Act}_x(e)$, and assume there is no closure witness, no transfer to a non-reconstructing process, and no non-regenerative certificate. Then the ordinary relief clause holds by the recorded residue and bounded return burden. The extinction clause fails because the only residue reconstructs x and no extinction-compatible witness is present. □

Proposition 6.36 (Load-queue dissipation obstruction)**.** *If a cessation transition for x moves the active commitment into a work queue whose queue burden cannot be driven below β_{queue} within the declared quench budget, and the queue is not transferred to a process whose regeneration closure is disjoint from the post-transition active-node set $\mathsf{Act}_x(e')$, then that transition is excluded from both the relief kernel and the complete-extinction kernel.*

Proof. The relief kernel requires membership in B_x and satisfaction of mandated load-quench obligations. The complete-extinction target additionally requires dissipative cessation and queue quench when a queue component is present. A transition that merely converts visible activity into a growing queue fails these clauses because B_{queue} remains above threshold without a non-regenerative transfer. Since K^{rel} and K^{ext} are built from these target and burden conditions, the transition is not admissible in either kernel. □

Proposition 6.37 (Debt non-resurrection obstruction). *Assume an extinction request for x is mandated. If some recognized debt requires active continuation of x and has no closure witness, no transfer witness to a process that does not reconstruct x, and no non-regenerative certificate, then $C^{\mathrm{nr}}_{\mathsf{Q}}(x)$ is empty on the region in which that debt is active. Consequently the complete-extinction kernel cannot certify liberation from that region. A mere declaration that the debt no longer requires continuation does not remove the obstruction.*

Proof. The target definition requires non-resurrection-compatible debt. An active debt without closure, transfer, or non-regenerative certificate fails that clause. Since $C^{\mathrm{nr}}_{\mathsf{Q}}(x) \subseteq \mathsf{Tgt}_{\mathsf{Ext}}(x)$, no state carrying such a debt belongs to the closure. The attractor to an empty target within that region cannot certify complete extinction. The final sentence follows because the definition requires a witness, not a declaration. □

Proposition 6.38 (Refinement-simulation preservation). *Let $\mathsf{Q}_2 \sqsubseteq \mathsf{Q}_1$ be represented by measurable maps and a simulation relation $\mathsf{Sim} \subseteq \mathcal{E}_{\mathsf{Q}_1} \times \mathcal{E}_{\mathsf{Q}_2}$. Let*

$$\iota_A : \mathcal{A}_{\mathsf{Q}_1} \cup \mathcal{M}_{\mathsf{Q}_1} \to \mathcal{A}_{\mathsf{Q}_2} \cup \mathcal{M}_{\mathsf{Q}_2}$$

be the candidate and active-domain lift induced by the refinement maps, preserving active-domain equivalence and diagnostic coordinates up to the declared cone embedding. Assume:

(i) every Q_2 audit obligation, extinction request, debt token, residue, key, certificate, and active-domain event is reflected by a Q_1 obligation or by one of finitely many explicit new Q_2 witnesses on the certified window;

(ii) every certified Q_1 transition or kernel step has a Q_2 simulated transition or kernel coupling that keeps related states related;

(iii) open forward closure, non-regenerative certificates, debt closure or transfer witnesses, burden thresholds, Pareto cone embeddings, audit-critical partitions, and path-law divergence diagnostics are preserved;

(iv) if $(e, \tilde{e}) \in \mathsf{Sim}$ and all Q_1 successors of e remain in a set A, then all Q_2 successors of $\tilde{e}$ either remain related to A or carry explicit new witnesses satisfying the same target clauses;

Then non-resurrection complete-extinction certification under Q_1 implies certification of the lifted candidate under Q_2. If the Q_1 certificate is a finite liberation certificate, the lifted data determine a finite liberation certificate under Q_2.

Proof. First prove by descending induction on the finite non-resurrection closure iteration that simulated states in $C^{\mathrm{nr}}_{\mathsf{Q}_1}(x)$ satisfy the target clauses and successor-closure clauses of $C^{\mathrm{nr}}_{\mathsf{Q}_2}(\iota_A(x))$. Target preservation follows from obligation, debt, closure, cone, and partition reflection. Successor closure follows from the local successor-simulation clause. For the liberation part, assign each original certified state its safe-attractor rank r. For each lifted state, let m be the finite number of newly reflected obligations still requiring explicit discharge. The proof order is the lexicographic rank (r, m) constructed from these two quantities. A refined controller step that simulates an old step decreases r; a step introduced only to discharge reflected obligations keeps r fixed and decreases m. Thus no infinite lifted proof branch can avoid either progress in the original safe attractor or completion of all reflected obligations. The reflected debt, queue, open-residue, partition, refinement, failure, and end-component witnesses form the lifted typed ledger Θ, so the lifted strategy remains inside the refined liberation kernel. □

Proposition 6.39 (False-certification refinement counterexamples)**.** *If audit-obligation reflection, transition simulation, or hypergraph-closure reflection fails, a finite refinement can preserve coarse diagnostics while invalidating complete-extinction certification.*

Proof. For obligation failure, add one unreflected active-continuation debt in the refinement. For simulation failure, add one refined transition from a certified state to a state outside the lifted attractor. For closure-reflection failure, add one open hyperedge from a preserved certificate to the refined active-node set. In each case the old diagnostic coordinates can be copied unchanged, while a mandatory witness clause fails in the refined protocol. $\square$

Information and Audit Boundaries

Proposition 6.40 (Dual-cone witness preservation)**.** *Work in a finite-dimensional coordinate diagnostic space, with smaller scalar values interpreted as better. Let $\mathcal{K}_{\mathrm{break}}$ be the cone generated by mandatory witness-worsening directions. If a linear scalarization $\lambda \cdot y$ is not positive on $\mathcal{K}_{\mathrm{break}} \setminus \{0\}$, then there is a finite diagnostic counterexample in which the scalar strictly improves while a mandatory witness is broken. Consequently scalar seed certification is sound only when witness predicates are checked separately or the scalarization is strictly positive on every protocol-admissible witness-breaking direction and belongs to the diagnostic dual cone used for order comparison.*

Proof. If some $b \in \mathcal{K}_{\mathrm{break}} \setminus \{0\}$ has $\lambda \cdot b < 0$, set $y = z + b$. Then $\lambda \cdot y < \lambda \cdot z$ although y worsens a mandatory witness. If $\lambda \cdot b = 0$ and the protocol has a reuse-improving direction r with $\lambda \cdot r < 0$ that does not repair the broken witness, set $y = z + b + \epsilon r$ for small $\epsilon > 0$. Again the scalar improves while the witness remains broken. Thus scalar improvement alone cannot certify seedhood or non-resurrection-compatible improvement. The stated sufficient alternatives either keep the witness predicate outside the scalar or make every witness-breaking direction visible to the scalar. $\square$

Lemma 6.41 (Dual-cone Pareto certification)**.** *In a finite coordinate diagnostic space, suppose $y \preceq_{\mathrm{R}} z$, $y \not\succeq_{\mathrm{R}} z$, and the witness-coordinate displacement is non-worsening. If $\lambda \in \mathcal{K}_{\mathrm{R}}^*$ and $\lambda \cdot (z - y) > 0$, then $\lambda \cdot y < \lambda \cdot z$. The converse fails in general.*

Proof. Since $z - y \in \mathcal{K}_{\mathrm{R}}$ and $\lambda \in \mathcal{K}_{\mathrm{R}}^*$, $\lambda \cdot (z - y) \geq 0$; strict positivity gives the displayed strict scalar decrease. Conversely, a scalar can decrease along a direction outside $\mathcal{K}_{\mathrm{R}}$, so scalar improvement need not imply Pareto improvement or witness preservation. $\square$

Lemma 6.42 (Partition-KL diagnostic)**.** *Let P and Q be path laws on a measurable path space. If Π' refines a finite partition Π, then*

$$\mathrm{KL}_{\Pi}(P \| Q) \leq \mathrm{KL}_{\Pi'}(P \| Q).$$

If $P \ll Q$ and $\mathrm{KL}(P \| Q)$ exists, then every finite partition satisfies $\mathrm{KL}_{\Pi}(P \| Q) \leq \mathrm{KL}(P \| Q)$.

Proof. The refinement inequality is the log-sum inequality applied inside each cell of Π. The bound by full relative entropy follows by applying the same inequality to the conditional densities on each finite partition cell, equivalently by the data-processing inequality for the measurable map that sends a path to its partition cell. $\square$

Proposition 6.43 (Audit-critical finite two-path indistinguishability)**.** *Let the finite audited path model admit Dirac path laws on every admissible audited path, and let a certificate rule depend on a path law only through a finite partition Π and its partition-level probabilities. If Π is not audit-critical for (x, e, Q) and some partition cell contains two admissible audited paths, one with* *a critical regeneration, debt, request, residue, certificate, or active-domain event and one without it, then there exist two admissible Dirac path laws with the same induced law on* Π *but different critical-event probability.*

Proof. Let ω_1 and ω_0 be admissible audited paths in the same partition cell, with ω_1 in the critical event C and ω_0 outside it. The Dirac laws δ_{ω_1} and δ_{ω_0} have the same induced law on Π because the paths occupy the same cell. Their probabilities of C are 1 and 0. A certificate rule seeing only Π cannot distinguish them. $\square$

Proposition 6.44 (Audit-critical law-rich identifiability). *Let $\mathcal{L}$ be the protocol-declared class of admissible path laws, and fix the protocol reference pushforward Q_Π for partition-level KL. Assume law-richness relative to a finite partition Π: whenever a partition cell contains two admissible audited paths with different membership in a critical event $C \in \mathsf{Crit}_x(e)$, the class $\mathcal{L}$ contains two path laws P_0, P_1 with the same pushforward to Π and $P_0(C) \neq P_1(C)$. If a certificate rule depends only on the partition pushforward law or KL against the fixed Q_Π, and Π is not audit-critical, then the rule cannot soundly certify non-resurrection over all laws in $\mathcal{L}$. If Π is audit-critical and every cell contained in a critical violation event has certified probability zero, then all partition-measurable critical violations have probability zero.*

Proof. By law-richness, non-criticality supplies two admissible laws with identical partition data and different critical-event probability. Any rule depending only on that partition data gives the same output on both laws, so it cannot distinguish the violation. In the audit-critical case, each critical violation is a union of partition cells; if all such cells have probability zero, the union has probability zero. $\square$

Proposition 6.45 (Audit hyperproperty witness). *Assume a finite audit window and an audit-critical partition Π. If a non-resurrection certificate fails on the audit path family $\mathcal{T}_{\mathrm{aud}}(x, e)$, then there is a finite witness consisting of a finite path prefix, a finite raw-boundary subset, and a finite open hyperedge derivation into $\mathsf{Act}_x(e')$ at some prefix state e'. Conversely, if Π is not audit-critical and the path-law class is law-rich, there are two admissible path families with the same partition-level data and different non-resurrection status.*

Proof. In a finite audit window, every violation of the declared non-resurrection condition is generated by finitely many successor steps and finitely many open hyperedges. Audit-criticality makes the triggering event, the active-domain head event, and the relevant debt, request, residue, certificate, and queue events unions of partition cells. Hence the violating prefix and hyperedge derivation form a finite witness. The converse is exactly the law-rich indistinguishability construction applied to a critical event not separated by the partition. $\square$

Proposition 6.46 (Persistence-selection identity). *Let (X_n) be a finite Markov chain with extinction set D and $\tau_D = \inf\{n : X_n \in D\}$. For $m \geq 0$ and $i \notin D$, set $r_m(i) = \Pr_i(\tau_D > m)$. Whenever $\Pr(\tau_D > n) > 0$,*

$$\Pr(X_n = i \mid \tau_D > n + m) = \frac{\Pr(X_n = i \mid \tau_D > n)\, r_m(i)}{\sum_{j \notin D} \Pr(X_n = j \mid \tau_D > n)\, r_m(j)}.$$

Thus survival-conditioned observations overrepresent states with above-average future survival probability.

Proof. The identity is Bayes' rule and the Markov property. $\square$

Proposition 6.47 (Survival bias and complete-extinction independence)**.** *There are finite systems with identical survival-conditioned observational Markov laws on non-extinct states but different controller extensions and hence different complete-extinction kernels. Hence persistence-conditioned observation alone does not decide membership in* $K_{\mathsf{Q}}^{\mathrm{ext}}(x)$*.*

Proof. Let the observed chain in both systems have states $\{a, d\}$, extinction state d, and initial state a, with the non-extinct observational law conditioned on survival assigning probability one to a at every time. In the first protocol extension the controller action set contains a certified action sending a to d, so $a \in K_{\mathsf{Q}}^{\mathrm{ext}}(x)$. In the second protocol extension the controller action set at a contains only a persistence action whose successors remain at a, so $a \notin K_{\mathsf{Q}}^{\mathrm{ext}}(x)$. The survival-conditioned observational law is identical because the controller extension is not part of that conditioned observation, while complete-extinction accessibility is different. □

Proposition 6.48 (Non-explosion loop liberation)**.** *Assume the continuous-time audit regime on* $[0, T]$ *with non-resurrection set* D *and lifetime* ζ*. If* $\Pr_e(\zeta < T,\ \zeta < \tau_D) > 0$*, then the cadlag audit law does not certify loop liberation to non-resurrection on* $[0, T]$*. If* $\Pr_e(\zeta \le T,\ \zeta < \tau_D) = 0$*,* $D \cap \mathsf{ActCfg}_x = \varnothing$*, and the cadlag law assigns probability one to* $\mathsf{SafePath}_D^T(e)$*, then finite-horizon loop liberation to* D *and post-hit non-resurrection are certified on* $[0, T]$*. If only* $\Pr_e(\mathsf{PostSafe}_D^T(e) \mid \tau_D \le T) = 1$ *is known, then the conclusion is only conditional non-resurrection after hitting.*

Proof. On the event $\{\zeta < T, \zeta < \tau_D\}$ the process leaves the audited path space before certified non-resurrection, so the protocol has no path-law witness that loop liberation occurred. The event $\mathsf{SafePath}_D^T(e)$ is the intersection of hitting D before T and remaining in D at audited post-hit times. Since $D \cap \mathsf{ActCfg}_x = \varnothing$, this gives finite-horizon hitting and post-hit non-resurrection. If only the conditional post-hit event is probability one, the hitting part is not certified, so only the conditional safety conclusion follows. □

Theorem 6.49 (Finite evidence diagonal impossibility)**.** *Let a model class be extension-closed: whenever a finite observed horizon* N *is admitted, the class also admits an observationally identical extension that adds a delayed regeneration channel after* N *without changing raw records, quotient observations, diagnostics, or witness certificates up to* N*. Then no certificate rule depending only on the finite horizon can soundly certify absolute extinction over the whole class. The sound predicate must be protocol-certified non-resurrection under the audited horizon and declared audit-stable extensions.*

Proof. For any finite certificate input, choose the extension that agrees on all observed data up to the certificate horizon and adds delayed regeneration afterward. The rule has identical input in the base model and in the extension, hence identical output. If it certifies absolute extinction in the base model, it also certifies the extension, where absolute non-reconstructability is false. □

7 Phase Regions and Boundaries

Definition 7.1 (Operational phase regions)**.** Phase regions are subsets of diagnostic and witness space:

$$\Phi_{\mathrm{seed}} = \{e : \exists s,\ \mathsf{Seed}(s; e, \mathsf{Q})\}, \quad \Phi_{\mathrm{crystal}} = \{e : \exists c,\ \mathsf{Crystal}(c; e, \mathsf{Q})\},$$

$$\Phi_{\mathrm{nr}} = \{e : \exists x,\ e \in C_{\mathsf{Q}}^{\mathrm{nr}}(x)\},$$

$$\Phi_{\mathrm{ext}} = \{e : \exists x,\ e \in K_{\mathsf{Q}}^{\mathrm{ext}}(x)\},$$

$$\Phi_{\mathrm{loop}} = \{e : \exists x,\ e \text{ reaches a closed component disjoint from } C_{\mathsf{Q}}^{\mathrm{nr}}(x)\}.$$

Define Φ_{reqTrap}, Φ_{loadTrap}, Φ_{debtTrap}, Φ_{loopTrap}, and Φ_{regen} respectively as the states violating the access-witness, load-quench, debt-closure, loop-liberation, and open-closure clauses required for the relevant kernel. In finite and standard Borel regimes these are finite or universally measurable whenever the underlying witness and predecessor sets are.

Proposition 7.2 (Phase measurability)**.** *In the finite regime all displayed phase regions are finite subsets. In the standard Borel analytic regime, assume the candidate, configuration, and finite update-label spaces are standard Borel, and seed, typed-update, witness, and kernel-membership graphs are analytic. Then Φ_{seed} and Φ_{crystal} are analytic projections of finite derivation graphs, and the trap regions defined by finite witness failure are universally measurable.*

Proof. The finite case is immediate. For the standard Borel analytic case, Φ_{seed} is the projection of the analytic seed graph. A crystal of length m lives in a finite product of standard Borel spaces for states, candidates, and update labels. The length-m derivation constraints are finite intersections of analytic typed-update, witness, and kernel-membership graphs; hence the length-m crystal set is analytic. The union over $m \in \mathbb{N}$ is analytic, and projection to the initial state gives Φ_{crystal}. The remaining trap regions are obtained from finite Boolean combinations and projections of the declared witness and predecessor graphs, so they are universally measurable under the stated regime. $\square$

Remark 7.3 (Phase entry). Entry into the seeded phase requires a finite or stochastic transition into the required kernel after strict reuse improvement. Entry into the complete-extinction phase requires a controller-certified transition into $C^{\mathrm{nr}}_{\mathsf{Q}}(x)$ or satisfaction of the stochastic non-resurrection threshold.

Remark 7.4 (Boundary with viability and invariance). Classical viability and invariance theories study remaining inside constraint sets under dynamics [9, 10]. OSCT uses related fixed-point constructions, but non-resurrection complete extinction is a reachability-to-safety construction: the process must reach a successor-closed non-regeneration region.

Remark 7.5 (Boundary with model checking, liveness, and termination). Finite OSCT witness conditions can be represented as bounded reachability, request-response checks, game attractors, and ranking certificates on transition systems [11, 12, 13, 14, 15]. OSCT differs by coupling those checks to reuse-diagnostic improvement, audit saturation, regeneration closure, non-regenerative certificates, and debt ledgers.

Remark 7.6 (Boundary with simulation and abstraction). Symbolic control, approximate simulation, and hybrid-system verification study finite abstractions and simulation relations for continuous systems [17, 18]. Interface automata study component assumptions, guarantees, and alternating refinement [19]. OSCT uses similar local-simulation discipline, but the abstraction target is regeneration closure and liberation certification: a refinement is admissible only when it over-approximates concrete resurrection routes, reflects audit-critical active-domain events, and reconstructs the finite rank certificate rather than merely preserving a component interface.

Remark 7.7 (Boundary with stochastic control and recurrent classes). Stochastic non-resurrection recursions are finite-horizon feasibility recursions related to Markov decision processes and stochastic control on finite and Borel spaces [21, 22, 23]. The robust version keeps controller/scheduler quantifier order, as in stochastic-game analysis, but uses OSCT-specific success, failure, and burden certificates rather than an ω-regular winning objective [30]. Analytic and semianalytic measurability are standard descriptive-set-theoretic tools [24]. Closed-set invariance, Feller regularity, and non-explosive cadlag path laws use standard Markov-process structure [26, 27]. Almost-sure loop liberation uses recurrent-class and drift ideas from Markov-chain stability and probabilistic termination [28, 29]. Persistence-conditioned observations are related to quasi-stationary analysis of absorbing chains [31, 32].

Remark 7.8 (Boundary with secure deletion and no-deleting results). Media sanitization standards and secure deletion study effort-bounded data recovery [35, 36]. Quantum no-deleting results mark a different physical limit for unknown quantum states [37]. OSCT does not identify complete extinction with any one substrate theory; it abstracts the common structure as audit saturation, regeneration closure, non-resurrection invariance, and non-regenerative certificates.

Remark 7.9 (Boundary with vector payoff and information geometry). The reuse preorder may include vector-valued coordinates, Pareto frontiers, information divergences, or geometric diagnostics [20, 3, 4]. OSCT uses these as witness-constrained diagnostic coordinates. It does not reduce them to a universal scalar unless the scalarization preserves mandatory witness coordinates and non-resurrection certificates.

Remark 7.10 (Boundary with thermodynamic entropy and free energy). Local reduction in reuse dispersion and complete-extinction certification do not imply global thermodynamic entropy decrease, because substrate dissipation and erasure costs are represented separately [1, 5, 6]. On non-countable path spaces, OSCT uses finite partitions or path-law relative entropy rather than unconstrained Shannon sums; weak-convergence and large-deviation tools are background only when the protocol supplies the required path laws [33]. A free-energy quantity may appear as one protocol-visible diagnostic coordinate [39].

8 Core Result Dependency Map

D1. Result: finite audit substrate. Depends-on: finite-set measurability, typed protocol interpretation, boundary closure. Output: finite state and witness spaces for all finite kernels.

D2. Result: non-resurrection closure, Theorem 6.7. Depends-on: extinction target, regeneration closure, audit-admissible successors. Output: greatest successor-closed non-resurrection fixed point.

D3. Result: complete-extinction attractor, Theorem 6.8. Depends-on: lifted target $D_x^\sharp$ and lifted burden region $B_x^\sharp$. Output: finite controlled attractor and scheduler avoid region.

D4. Result: non-circular liberation certificate, Theorem 6.11. Depends-on: Safe_x, $\mathsf{SafePred}_x$, finite rank, and $\mathsf{RawLibCert}$. Output: equivalence between K^{lib} membership and a raw finite certificate.

D5. Result: safe-attractor obstruction dichotomy, Theorem 6.14. Depends-on: finite safe attractor, avoiding end components, and failure/unsafe-complement reachability. Output: liberation certificate or scheduler obstruction.

D6. Result: robust stochastic feasibility, Theorem 6.24. Depends-on: finite controller/scheduler kernel and $\mathsf{WeakCl}_{\mathsf{Tri}}$. Output: strict/practical success-failure-burden certificate.

D7. Result: compact Feller kernels, Theorems 6.18 and 6.19. Depends-on: closed targets, lower semicontinuous burdens, compact-valued actions, closed graphs, and joint weak continuity. Output: closed safety kernel and selector.

D8. Result: abstraction and refinement preservation. Depends-on: audit reflection, regeneration-closure over-approximation, typed witness ledgers, and active-domain reflection. Output: sound Galois abstraction, audit hyperproperty, and refinement preservation statements.

9 Condensed Formal Statement

S1. Objects are indexed process-context-protocol configurations.

S2. The theory has no terminal objective primitive and no hidden subject primitive.

S3. Raw records induce mandatory audit obligations before diagnostic comparison.

S4. Relief has three levels: cessation, dissipative cessation, and non-resurrection complete extinction.

S5. Absolute extinction is not certified without audit completeness; the formal predicate is protocol-relative.

S6. Non-resurrection is a successor-closed safety invariant after extinction.

S7. Non-resurrection certification is a finite audit hyperproperty over generated paths and refinement families.

S8. Liberation is a safe-attractor fixed point over a typed lifted safe predicate excluding debt, queue, open-residue, partition, refinement, failure, and boundary bad states.

S9. Open forward closure in the regeneration hypergraph must not intersect the active domain.

S10. Practical infinite persistence is represented by closed components, end components, or recurrent classes disjoint from non-resurrection.

S11. Loop liberation is certified by a non-circular finite $\mathsf{RawLibCert}$ ledger, a safe-attractor rank, recurrent-class exclusion, ranking functions, or non-explosive cadlag audit path laws.

S12. Under an extinction request, preserved records, keys, descendants, and certificates are allowed only when jointly non-regenerative.

S13. Continuous-state claims require standard Borel analytic or compact Feller regularity declared at the theorem level; finite stochastic claims use weakening-closed robust feasibility triples rather than scalarized expectations.

S14. Half-ordered diagnostics are treated by cone/Pareto comparison and mandatory witness preservation, not by default scalarization.

S15. Non-countable path-space information diagnostics use audit-critical finite partitions with fixed reference pushforward laws, or path-law relative entropy, not arbitrary Shannon sums.

S16. Open-residue dynamics are audited on fixed finite ambient status registers, so discharge and non-regeneration are state predicates rather than time-dependent side conditions.

S17. Regeneration abstraction is sound only when it over-approximates concrete open closure and reflects active-domain events.

S18. Refinement preserves complete-extinction and liberation certification only when local reflection and simulation reconstruct the finite witness ledger and safe-attractor rank certificate.

S19. A meaning seed is strict admissible diagnostic improvement into the required relief or complete-extinction kernel.

10 Conclusion

OSCT formalizes reusable ordering as audit-adequate crystallization over observable process configurations, while separating survival-biased observation from relief accessibility. The strongest relief notion developed here is non-resurrection complete extinction with liberation certification: a typed safe-attractor or regular stochastic proof that the active commitment reaches a successor-closed region from which audited residues, records, keys, descendants, debts, queues, certificates, and refinements cannot reconstruct it. In finite audit windows, the raw certificate consists of a strategy, a safe-attractor rank, a successor-closed invariant, and typed witness ledgers for debt, queue, open residue, partition, refinement, failure, and end-component exclusion; it is equivalent to liberation-kernel membership without assuming that membership. Stochastic certification keeps success probability, failure probability, and expected burden as a weakening-closed robust feasibility object, so safety failure is not hidden by scalar averaging. This is not an absolute erasure claim outside the protocol. It is a mathematical condition under which a process selected for persistence can still have an auditable route out of practical infinity, loop retention, failure displacement, and future resurrection.

References

[1] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

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

[3] I. Csiszár. Information-type measures of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica*, 2:299–318, 1967.

[4] S. Amari and H. Nagaoka. *Methods of Information Geometry.* American Mathematical Society and Oxford University Press, 2000.

[5] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961. doi:10.1147/rd.53.0183.

[6] C. H. Bennett. The thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21:905–940, 1982. doi:10.1007/BF02084158.

[7] D. Kashchiev. *Nucleation: Basic Theory with Applications*. Butterworth-Heinemann, 2000.

[8] K. F. Kelton and A. L. Greer. *Nucleation in Condensed Matter: Applications in Materials and Biology.* Pergamon, 2010.

[9] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre. *Viability Theory: New Directions*. Springer, 2nd edition, 2011. doi:10.1007/978-3-642-16684-6.

[10] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999. doi:10.1016/S0005-1098(99)00113-2.

[11] R. W. Floyd. Assigning meanings to programs. In J. T. Schwartz, editor, *Mathematical Aspects of Computer Science*, Proceedings of Symposia in Applied Mathematics, volume 19, pages 19–32. American Mathematical Society, 1967.

[12] A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*, pages 46–57, 1977. doi:10.1109/SFCS.1977.32.

[13] B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985. doi:10.1016/0020-0190(85)90056-0.

[14] A. Podelski and A. Rybalchenko. Transition invariants. In *Proceedings of the Nineteenth Annual IEEE Symposium on Logic in Computer Science*, pages 32–41, 2004.

[15] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008. ISBN 9780262026499.

[16] M. R. Clarkson and F. B. Schneider. Hyperproperties. *Journal of Computer Security*, 18(6):1157–1210, 2010. doi:10.3233/JCS-2009-0393.

[17] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009. doi:10.1007/978-1-4419-0224-5.

[18] A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5):782–798, 2007. doi:10.1109/TAC.2007.895849.

[19] L. de Alfaro and T. A. Henzinger. Interface automata. In *Proceedings of the 8th European Software Engineering Conference*, pages 109–120. ACM, 2001. doi:10.1145/503209.503226.

[20] D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956. doi:10.2140/pjm.1956.6.1.

[21] D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Academic Press, 1978.

[22] O. Hernández-Lerma and J. B. Lasserre. *Discrete-Time Markov Control Processes: Basic Optimality Criteria*. Springer, 1996. doi:10.1007/978-1-4612-0729-0.

[23] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994. doi:10.1002/9780470316887.

[24] A. S. Kechris. *Classical Descriptive Set Theory*. Springer, 1995. doi:10.1007/978-1-4612-4190-4.

[25] C. Castaing and M. Valadier. *Convex Analysis and Measurable Multifunctions*. Lecture Notes in Mathematics 580. Springer, 1977. doi:10.1007/BFb0087685.

[26] S. N. Ethier and T. G. Kurtz. *Markov Processes: Characterization and Convergence*. Wiley, 1986. doi:10.1002/9780470316658.

[27] O. Kallenberg. *Foundations of Modern Probability*. Springer, 3rd edition, 2021. doi:10.1007/978-3-030-61871-1.

[28] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, 2nd edition, 2009. doi:10.1017/CBO9780511626630.

[29] K. Chatterjee, H. Fu, and A. K. Goharshady. Termination analysis of probabilistic programs through Positivstellensatz's. *Formal Methods in System Design*, 52:260–296, 2018. doi:10.1007/s10703-018-0313-8.

[30] K. Chatterjee and T. A. Henzinger. A survey of stochastic ω-regular games. *Journal of Computer and System Sciences*, 78(2):394–413, 2012. doi:10.1016/j.jcss.2011.05.002.

[31] J. N. Darroch and E. Seneta. On quasi-stationary distributions in absorbing discrete-time finite Markov chains. *Journal of Applied Probability*, 2(1):88–100, 1965. doi:10.2307/3211876.

[32] P. Collet, S. Martinez, and J. San Martin. *Quasi-Stationary Distributions: Markov Chains, Diffusions and Dynamical Systems.* Springer, 2013. doi:10.1007/978-3-642-33131-2.

[33] P. Dupuis and R. S. Ellis. *A Weak Convergence Approach to the Theory of Large Deviations.* Wiley, 1997.

[34] P. Billingsley. *Convergence of Probability Measures.* Wiley, 2nd edition, 1999. doi:10.1002/9780470316962.

[35] R. Kissel, A. Regenscheid, M. Scholl, and K. Stine. *Guidelines for Media Sanitization.* NIST Special Publication 800-88 Revision 1, National Institute of Standards and Technology, 2014. doi:10.6028/NIST.SP.800-88r1.

[36] P. Gutmann. Secure deletion of data from magnetic and solid-state memory. In *Proceedings of the Sixth USENIX Security Symposium*, 1996.

[37] A. K. Pati and S. L. Braunstein. Impossibility of deleting an unknown quantum state. *Nature*, 404:164–165, 2000. doi:10.1038/404130b0.

[38] K. Kuratowski and C. Ryll-Nardzewski. A general theorem on selectors. *Bulletin de l'Academie Polonaise des Sciences, Serie des Sciences Mathematiques, Astronomiques et Physiques*, 13:397–403, 1965.

[39] K. Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11:127–138, 2010. doi:10.1038/nrn2787.

[40] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970.