Audited Residue Calculus

ARC 1.0

Formal Specification of the Canonical Audited Generator

Noah Robideau

A standalone mathematical system for residue-preserving reasoning, scoped answers, certified erasure, and obstruction-null completion.

Version	1.0
Author	Noah Robideau
Document status	Formal mathematical specification
Generator	CAG[Γ] = G[Γ]
Foundation	Univalent / homotopy-coherent dependent type theory
Export date	2026-06-28

Every erased distinction must return as flatness, reconstruction, ledgering, or registered obstruction.

Contents

1. Executive Abstract

Audited Residue Calculus, abbreviated ARC, is a formal system for reasoning without silent erasure. It treats a problem as a structured presentation, computes the unresolved content of that presentation as a residue, and permits conclusions only when every generated residue has a certified fate.

The canonical object of the system is the Canonical Audited Generator, denoted CAG[Γ] or G[Γ]. For a context Γ, G[Γ] is the initial structure generated by the ARC formation rules, constructors, equations, legal-transition rules, scope rules, answer-record rules, and failure-registration rules.

ARC is not a universal solver. It does not assert that every residue is flat, every problem is solvable, or every answer is globally valid. Its stronger value is that it provides a universal exact continuation discipline: every presentable distinction can be converted into exact residue form, and every attempted erasure of information must expose its corresponding fiber, reconstruct the lost information, ledger it forward, or register an obstruction.

Central Statement. For every context Γ, the canonical generator G[Γ] is initial among ARC models over Γ. Every derivable ARC answer is one of four kinds: a global answer, a scoped answer with provenance, a legally promoted scoped answer with fiber accounting, or a registered failure residue.

2. Design Intent and Governing Law

The design intent of ARC is to prevent the most common failure of complex reasoning: a generated distinction disappears before it is solved. The system therefore separates intuition, presentation, residue, support, scope, proof, and completion.

The method begins from a direction of search, but it does not treat that direction as proof. It compiles the raw input into a formal presentation, generates all residue forced by that presentation, and makes every later claim conditional on the fate of that residue.

ARC Master Law: no generated residue may be discarded without flatness, reconstruction, ledgering, or failure registration.

Flatness. A residue is flat when it is equivalent to the terminal object in its typed carrier. Flatness is the formal certificate of canonical solvedness.

Reconstruction. A residue is reconstructed when there is auditable data from which an equivalent object is generated, together with the required equivalence certificate.

Ledgering. A residue is ledgered when it is carried forward to a successor component through a typed comparison, with the necessary obligation and transfer certificates.

Failure registration. A missing certificate, missing construction, ambiguity, obstruction, or unavailable decision procedure is not ignored. It becomes a typed failure residue.

3. Foundation and Metatheory

ARC is stated relative to an ambient univalent or homotopy-coherent dependent type theory. Equivalent interpretations may be given in a sufficiently well-behaved infinity-categorical foundation. The formal system assumes universes, homotopy pullbacks, homotopy fibers, dependent sums, dependent products, mapping objects, section objects, and the particular initial algebra or colimit constructions explicitly invoked below.

Universe hierarchy:
$U_0 : U_1 : U_2 : \cdots$

Small spaces:
$\mathscr{S} = \mathscr{S}_{\mathcal{U}}$

Flatness of a space E:
Flat(E) := Equiv(E, 𝟙)

Obstruction:
Obs(E) := (E → 𝟘)

Constructive ambiguity:
Amb(E) := ‖E‖ × ¬Flat(E)

Regime classification is treated as a certificate problem. ARC does not assume a global decision procedure that classifies every residue as empty, ambiguous, or flat. When a decision is needed but unavailable, the missing decision certificate is registered as residue.

Regime certificate:
RegimeCert(E) := Obs(E) + Amb(E) + Flat(E)

Missing regime decision:
RegimeDecisionRes(E) := RegimeCert(E)

Metatheoretic Discipline. Every assertion of closure, reconstruction, soundness, completeness, support conservativity, termination, evidence calibration, resource validity, or scope promotion is represented by a certificate space. A certificate space may itself be obstructed, ambiguous, or flat.

4. Syntax and Judgment Forms

ARC is a judgmental calculus. The following judgment forms define what may be built, used, transformed, certified, or reported.

Judgment	Meaning
Γ ctx	Γ is a well-formed context.
$\Gamma \vdash \tau$ type	τ is a generated or admitted type.
$\Gamma \vdash E : V_\tau$	E is a carrier object at type τ.
$\Gamma \vdash f : E \to_\tau F$	f is a carrier map at type τ.
$\Gamma \vdash P : \mathrm{Prob}_\tau$	P is a typed formal problem.
$\Gamma \vdash \mathrm{Res}(P) : V_\tau$	The residue of P is generated.
$\Gamma \vdash \alpha : \mathrm{State}$	α is an audited state.
$\Gamma \vdash T : \mathrm{Leg}_1(\alpha,\beta)$	T is a legal transition.
$\Gamma \vdash F : \mathrm{OutDoc}_\lambda$	F is an output doctrine at target type λ.
$\Gamma \vdash A_o = \mathrm{GloF}\lambda(X;\Gamma)$	A_o is the maximal declared answer object.
$\Gamma \vdash$scope $q : A_o \to A$	q is a generated scope map.
$\Gamma \vdash a : \mathbb{1} \to A$	a is a visible answer at scope A.

Judgment	Meaning
$\Gamma \vdash R : \mathrm{AnsRec}(\Gamma, X, F, \lambda)$	R is a valid answer record.
$\Gamma \vdash \mathrm{Failure}(\mathrm{label}, C)$	C is registered as a failure residue.

A derivation is valid only if each name, map, certificate, and scope used by the derivation is generated from the current context or admitted by an explicit context extension. Otherwise the admissibility residue is produced.

5. Formal Problems and Residues

The primitive problem shape is the walking cospan. This single shape separates candidates, interfaces, and selected specifications.

Walking cospan:
$\Lambda = (0 \to 1 \leftarrow 2)$

Typed formal problems:
$\mathrm{Prob}_\tau := \mathrm{Fun}(\Lambda, V_\tau)$

A problem P:
$P = (C_P \to B_P \leftarrow S_P)$

Residue of P:
$E_P := C_P \times_{B_P} S_P$

Candidate space C. The space of possible witnesses, constructions, models, proofs, explanations, states, histories, or realizations.

Interface space B. The space of observables, constraints, boundary data, verification outputs, law conditions, tests, or target-interface data.

Selected specification S → B. The selected datum, allowed condition, boundary constraint, target requirement, evidence record, or specification being imposed.

Residue E_P. The homotopy pullback $C_P \times_{B_P} S_P$. It is the unresolved content of the problem under the chosen presentation.

Formation rule:

$$\frac{\Gamma \vdash C : V_\tau \quad \Gamma \vdash B : V_\tau \quad \Gamma \vdash S : V_\tau \quad \Gamma \vdash r : C \to B \quad \Gamma \vdash s : S \to B}{\Gamma \vdash (C \to_r B \leftarrow_s S) : \mathrm{Prob}_\tau}$$

Residue rule:

$$\frac{\Gamma \vdash P = (C \to B \leftarrow S) : \mathrm{Prob}_\tau}{\Gamma \vdash \mathrm{Res}(P) := C \times_B S : V_\tau}$$

Residue Regimes. A residue may be obstructed, ambiguous, or flat. A point of E_P certifies existence; flatness of E_P certifies canonical solvedness. Inhabitation and determinacy are distinct certificates.

6. Exact Normal Forms and Residue Localization

Exact normal forms preserve unresolved content. They are not solutions unless the preserved residue is already flat.

For $E : V_\tau$, define:
$J_\tau(E) := (E \to \mathbb{1}_\tau \leftarrow \mathbb{1}_\tau)$
$\Delta_\tau(E) := (E \to_{id} E \leftarrow_{id} E)$

Computation rules:
$\mathrm{Res}(J_\tau(E)) \simeq E$
$\mathrm{Res}(\Delta_\tau(E)) \simeq E$

Exact Continuation. For every typed residue E, both $J_\tau(E)$ and $\Delta_\tau(E)$ are exact continuations. They preserve residue equivalently and therefore never delete unresolved content.
Proof. By direct computation of the pullback defining the residue. The residue of $E \to 1 \leftarrow 1$ is E, and the residue of $E \to E \leftarrow E$ along identities is E.

Residue Localization. For every type τ, typed formal problems localized at residue-equivalences are equivalent to typed residues: $\mathrm{Prob}_\tau[W_{res}^{-1}] \simeq V_\tau$.
Proof. The residue functor $\mathrm{Res}_\tau : \mathrm{Prob}_\tau \to V_\tau$ has the constant-cospan embedding Δ_τ as a fully faithful left adjoint in the relevant homotopy-coherent sense. The counit $\Delta_\tau(\mathrm{Res}(P)) \to P$ is a residue-equivalence. Therefore every problem is identified, after localization, with its residue-only normal form.

This theorem is the mathematical reason ARC can treat a problem by preserving its residue. Presentation matters operationally and semantically, but modulo residue-equivalence the problem content is exactly the residue.

7. Typed Carriers, Support, and Conservativity

A type τ determines a carrier V_τ. ARC distinguishes structured carrier-level claims from support-level claims. Support may be easier to compute, but support-level flatness does not automatically imply structured flatness.

Typed carrier data:
V_τ — carrier category / type of objects
$\mathbb{1}_\tau$ — terminal object
Pull_τ — finite homotopy pullbacks
$\mathrm{Map}_\tau(E,F)$ — mapping space
$U_\tau : V_\tau \to \mathscr{S}$ — support functor

Structured flatness:
$\mathrm{Flat}_\tau(E) := \mathrm{Equiv}_{V_\tau}(E, \mathbb{1}_\tau)$

Support flatness:
$\mathrm{SuppFlat}_\tau(E) := \mathrm{Equiv}_{\mathscr{S}}(U_\tau(E), \mathbb{1})$

Support-conservativity for an arrow $a : E \to F$:
$\mathrm{SC}_\tau(a) := \mathrm{Map}(\mathrm{isEquiv}_{\mathscr{S}}(U_\tau(a)), \mathrm{isEquiv}_{V_\tau}(a))$

Specific-Arrow Conservativity. Support flatness may be promoted to structured flatness only with a support-conservativity certificate for the exact arrow involved. A certificate for a similar arrow or a support-equivalent object cannot be substituted.

This rule prevents a common collapse: treating a support-level summary as if it proved structure-level equivalence. ARC permits this transfer only through an explicit certificate space.

8. States, Registries, and Total Remainder

An ARC state records active problem presentations, typed residue components, dependencies, evidence, parameters, and history. The registry is primary; aggregate remainder is derived from it.

State:
$\alpha = (D_\alpha, P_\alpha, \chi_\alpha, \mathrm{Reg}_\alpha, \Theta_\alpha, \mathrm{Data}_\alpha, \mathrm{Hist}_\alpha)$

Registry:
$\mathrm{Reg}_\alpha = (J_\alpha, R_\alpha, \tau_\alpha, \mathrm{Dep}_\alpha, \mathrm{status}_\alpha, \mathrm{prov}_\alpha)$

For each component $j \in J_\alpha$:
$R_\alpha(j) : V_{\tau\alpha(j)}$

Aggregate support remainder:
$\mathrm{Rem}(\alpha) := \lim_{j \in J_\alpha} U_{\tau(j)}\, R_\alpha(j)$

Componentwise flatness:
$\mathrm{CompFlat}(\alpha) := \prod_{j \in J_\alpha} \mathrm{Equiv}_{V_{\tau(j)}}(R_\alpha(j), \mathbb{1}_{\tau(j)})$

Componentwise Flatness Implies Aggregate Flatness. If CompFlat(α) is inhabited, then Rem(α) is flat.
Proof. Each component has terminal support. A homotopy limit of a terminal diagram is terminal. The converse is not generally valid without additional dependency and support-conservativity certificates.

ARC uses the componentwise registry rather than only the aggregate remainder because dependencies can hide unresolved structure inside a flat-looking aggregate object.

9. Legal Transitions and Anti-Collapse

A transition is legal only if it carries forward the old residue obligations. State-time runs forward, while residue-obligation maps backward.

A legal transition $\alpha \to \beta$ is:
$T = (f, \rho, \mathrm{obl}, \mathrm{trans}, \mathrm{ac})$

where:
f : raw transition $\alpha \to \beta$
$\rho : \mathrm{Rem}(\beta) \to \mathrm{Rem}(\alpha)$
$\mathrm{obl} : \mathrm{Obl}(f,\rho)$
$\mathrm{trans} : \mathrm{Trans}(\rho)$
$\mathrm{ac} : \mathrm{AntiCollapse}(f)$

Therefore:
$\mathrm{Leg}_1(\alpha,\beta) := \sum_f \sum_\rho \mathrm{Obl}(f,\rho) \times \mathrm{Trans}(\rho) \times \mathrm{AntiCollapse}(f)$

For each old component $j \in J_\alpha$:
$\mathrm{AntiCollapse}_j(f) := \mathrm{Flat}_j(f) + \mathrm{Recon}_j(f) + \mathrm{Ledger}_j(f)$

Total anti-collapse:
$\mathrm{AntiCollapse}(f) := \prod_{j \in J_\alpha} \mathrm{AntiCollapse}_j(f)$

Flat branch. The old component is solved by a structured flatness certificate.

Reconstruction branch. The old component is reconstructed from auditable successor history, with an equivalence to the reconstructed object.

Ledger branch. The old component is compared to a successor component through a typed comparison, obligation certificate, and transfer certificate.

Anti-Collapse Theorem. AntiCollapse(f) is inhabited exactly when no old registry component is silently deleted by f.
Proof. The product over old components requires one lawful fate for each component. If any component lacks flatness, reconstruction, and ledgering, the product is empty and the transition is not legal.

Compositionality of Legal Transitions. Legal transitions compose. If $\alpha \to \beta$ and $\beta \to \gamma$ are legal, then the composite $\alpha \to \gamma$ is legal, with backward comparison $\mathrm{Rem}(\gamma) \to \mathrm{Rem}(\beta) \to \mathrm{Rem}(\alpha)$.

10. Failure Registry

Failure is internal to ARC. A missing certificate is a residue, not a side note. The failure registry is stratified to avoid circularity: failures required to operate a failure layer appear one stage higher.

Stage-zero failure examples:
SmallnessRes, AccessibilityRes, PresentationRes, PresentationAmbiguityRes, SelectionRes, CandidateGenerationRes, InterfaceSelectionRes, TypeLocalizationRes, TypeTranslationCoherenceRes, SupportConsRes, RuleCompleteRes, CanonRuleCompleteRes, ReconRes, LawRes, ObsRes, SubstrateRes, ResourceRes, FinalCoalgebraRes, RegimeDecisionRes

Successor stage:
$\mathrm{FailReg}_{n+1} := \mathrm{FailReg}_n + \mathrm{OperateCert}(\mathrm{FailReg}_n) + \mathrm{AuditCert}(\mathrm{FailReg}_n) + \mathrm{TransportCert}(\mathrm{FailReg}_n) + \mathrm{ComposeCert}(\mathrm{FailReg}_n) + \mathrm{SmallnessCert}(\mathrm{FailReg}_n) + \mathrm{AccessibilityCert}(\mathrm{FailReg}_n)$

Omega stage, when certified:
$\mathrm{FailReg}_\omega := \mathrm{colim}_{n<\omega}\, \mathrm{FailReg}_n$

Failure Closure. If $\mathrm{FailReg}_\omega$ exists in the ambient foundation, then the stratified registry is closed under every generated finite-stage failure mode. If the colimit or its hypotheses are unavailable, that absence is itself registered one level higher.

11. Admissibility Compiler

A raw name is not a generated object. ARC distinguishes namedness, generatedness, and context admission. This prevents proofs and constructions from smuggling in ungenerated maps, certificates, scopes, or presentations.

For a sort S and a named object x : S:

Generatedness:
$\mathrm{Gen}_\Gamma(S,x) := \sum_{t \in \mathrm{Term}_\Gamma(S)} \mathrm{Eq}_S(t,x)$

Context admission:
$\mathrm{CtxAdm}_\Gamma(S,x)$:= certificates that Γ, x:S is well formed

Admissibility residue:
$\mathrm{Adm}_\Gamma(S,x) := \mathrm{Gen}_\Gamma(S,x) + \mathrm{CtxAdm}_\Gamma(S,x)$

Admissibility branch	Operational effect
Generated branch	x is represented by a term of G[Γ] and may be used internally.
Context branch	x becomes a variable in the extended context Γ,x:S; all dependent answers retain that scope.
Failure branch	$\mathrm{Adm}_\Gamma(S,x)$ is exact-normalized or registered as failure residue.

Admissibility Soundness. Every accepted named object is generated from the current context or explicitly admitted by context extension. Every unaccepted name produces an admissibility residue.

This rule applies to inputs, types, carrier objects, maps, presentations, residues, output doctrines, answer objects, scope maps, visible answers, certificates, transitions, rules, and practical closed-form claims.

12. Global Answer Objects and Descent

An answer to one presentation is not automatically an answer to the input itself. ARC defines global answers as coherent sections over the presentation atlas, guarded against vacuous truth when no presentation exists.

Presentation space:
$\mathrm{Pres}_\Gamma(X) := \sum_p \mathrm{PresAdm}_\Gamma(X;p)$

For $p \in \mathrm{Pres}_\Gamma(X)$:
$p = (\tau_p, C_p \to B_p \leftarrow S_p)$
$E_p := C_p \times_{B_p} S_p$

Target-normalized residue:
$E_p^{\lambda} := L_\lambda\, e_{\tau_p}(E_p)$

Guarded global answer object:
$\mathrm{Glo}_F^{\lambda}(X;\Gamma) := \|\mathrm{Pres}_\Gamma(X)\| \times \prod_{p \in \mathrm{Pres}\Gamma(X)} F(E_p^{\lambda})$

Abbreviation:
$A_0 := \mathrm{Glo}_F^{\lambda}(X;\Gamma)$

Presentation Bridge. A claimed F-answer is presentation-global for X exactly when it is a point of $\mathrm{Glo}_F^{\lambda}(X;\Gamma)$. A single answer at one presentation p is an answer to (X,p), not automatically to X.

Local or interface-relative data are handled by descent. A local answer is globally usable only if the fiber of global answers over that local datum is flat, reconstructed, or ledgered.

Local-compatible answer object:
$\mathrm{Loc}_F^{\lambda}(X;U_\bullet;\Gamma) := \mathrm{holim}\, A_F^{\lambda}(U_\bullet;\Gamma)$

Restriction map:
$\mathrm{res}_U : \mathrm{Glo}_F^{\lambda}(X;\Gamma) \to \mathrm{Loc}_F^{\lambda}(X;U_\bullet;\Gamma)$

Descent residue for local datum ℓ:
$\mathrm{Desc}_F^{\lambda}(X;U_\bullet;\Gamma;\ell) := \mathrm{Glo}_F^{\lambda}(X;\Gamma) \times_{\mathrm{Loc}F\lambda(X;U\bullet;\Gamma)} \{\ell\}$

Legal descent:
LegalDesc := Flat(Desc) + Recon(Desc) + Ledger(Desc)

Descent Soundness. Compatible local answer data may be reported as global exactly when LegalDesc is inhabited. Otherwise the answer remains local-scoped or becomes a registered descent obstruction.

13. Scope Logic and Answer Records

ARC treats every answer as a value at a scope. The maximal declared answer object A_0 is relative to Γ, X, F, and λ. Any reported visible answer through a map from A_0 has an outside-scope residue.

Scope map:
$q : A_0 \to A$

Visible answer:
$a : \mathbb{1} \to A$

Outside-scope residue:
$\mathrm{Out}_q(a) := A_0 \times_A \{a\}$

Canonical promotion:
$\mathrm{CanonProm}_q(a) := \mathrm{Equiv}(\mathrm{Out}_q(a), \mathbb{1})$

Legal promotion:
$\mathrm{LegalProm}_q(a) := \mathrm{Flat}(\mathrm{Out}_q(a)) + \mathrm{Recon}(\mathrm{Out}_q(a)) + \mathrm{Ledger}(\mathrm{Out}_q(a))$

No Unscoped Answer. A visible answer a at scope q may be reported globally only with $\mathrm{LegalProm}_q(a)$. It may be reported as a naked canonical global answer only with $\mathrm{CanonProm}_q(a)$. Without either promotion or retained scope provenance, the output is invalid.

Answer record form	Required data
Global	$a_0 : \mathbb{1} \to A_0$.
Scoped	$q : A_0 \to A$, $a : \mathbb{1} \to A$, and provenance retaining q.
Promoted	$q : A_0 \to A$, $a : \mathbb{1} \to A$, and $\mathrm{LegalProm}_q(a)$.
Failure	A label and registered certificate space C.

Total answer-record space:
$\mathrm{AnsRec}(\Gamma,X,F,\lambda) := \mathrm{AnsGlobal} + \mathrm{AnsScoped} + \mathrm{AnsPromoted} + \mathrm{AnsFailure}$

13.1 Composite Scope Loss

If an answer is produced through several scopes, the total outside-scope residue is the fiber of the composite. Stagewise loss decomposes by pullback associativity.

For $A \to^f B \to^g C$ and $c : \mathbb{1} \to C$:
$A \times_C \{c\} \simeq \Sigma[b \in B \times_C \{c\}] (A \times_B \{b\})$

For an answer stack $A_0 \to A_1 \to \cdots \to A_n$ and $a_n : \mathbb{1} \to A_n$:
$\mathrm{OutStack}(a_n) := A_0 \times_{A_0} \{a_n\}$

Composite Loss Theorem. Total answer loss decomposes into the dependent aggregation of all stagewise losses. Stagewise accounting is sufficient for global accounting; a direct certificate for the composite fiber is also sufficient.

14. Practical Closed Forms

A practical form is a representation with certified loss accounting. A finite summary, statistic, model, memory, formula, numerical answer, or simplified expression is not a closed form merely because it is useful or compact.

Practical compression:
$q : E \to R$

Loss fiber over $r : \mathbb{1} \to R$:
$\mathrm{Loss}_q(r) := E \times_R \{r\}$

Selected-value legality:
$\mathrm{LegalValuePCF}(q,r) := \mathrm{Flat}(\mathrm{Loss}_q(r)) + \mathrm{Recon}(\mathrm{Loss}_q(r)) + \mathrm{Ledger}(\mathrm{Loss}_q(r))$

Representation-level legality:
$\mathrm{LegalRepPCF}(q) := \prod_{r \in R} (\mathrm{Flat}(\mathrm{Loss}_q(r)) + \mathrm{Recon}(\mathrm{Loss}_q(r)) + \mathrm{Ledger}(\mathrm{Loss}_q(r)))$

Practical Closed Form Criterion. A selected practical value r is globally reportable relative to the declared base object E, or to A0 when $q : A0 \to R$, exactly when LegalValuePCF(q,r) is inhabited. The whole representation R is a legal practical closed form exactly when LegalRepPCF(q) is inhabited.

Exact mode always preserves the full residue. Practical mode is useful only when it accounts for what it omits.

15. Proof Trees and Closure

Proof is modeled as a generated object, not as an informal explanation. A proof tree is sound only through a rule system whose rules carry closure-transfer data.

For a certified rule system R, define the proof-tree polynomial:

$(\Phi_R X)(\alpha) := \coprod_{n \geq 0} \coprod_{\beta_1,\ldots,\beta_n} R_n(\alpha;\beta_1,\ldots,\beta_n) \times \prod_i X(\beta_i)$

Assuming accessibility:
$\mathrm{Sol}_R := \mu\Phi_R$

Soundness map:
$\mathrm{sound}_\alpha : \mathrm{Sol}_R(\alpha) \to \mathrm{Cl}(I(\alpha))$

Proof-Tree Soundness. Every proof tree maps to a closure certificate. If $\mathrm{Sol}_R(\alpha)$ is inhabited, then α is closed under the closure doctrine represented by Cl.
Proof. By initial-algebra recursion. Each proof node contains a certified rule, and each certified rule contains the closure map from child closures to parent closure.

Rule completeness:
$\mathrm{RuleComplete}_R := \prod_\alpha \mathrm{Map}(\|\mathrm{Cl}(I(\alpha))\|, \|\mathrm{Sol}_R(\alpha)\|)$

Canonical rule completeness:
$\mathrm{CanonRuleComplete}_R := \prod_\alpha \mathrm{Equiv}(\mathrm{Sol}_R(\alpha), \mathrm{Cl}(I(\alpha)))$

Soundness is built into the proof tree. Completeness is never assumed. Missing proof completeness is a certificate residue.

16. Models and the Term Generator

An ARC model interprets the entire calculus: types, carriers, cospans, residues, exact forms, registries, transitions, proof trees, answer scopes, admissibility, and failure strata. A morphism of ARC models preserves this structure up to coherent equivalence.

ARC model component	Required interpretation
Types and carriers	A type object and a carrier V_τ for each type, with terminal object, pullbacks, support, and mapping spaces.

ARC model component	Required interpretation
Formal problems	Cospans $C \to B \leftarrow S$ and residue pullbacks $C \times_B S$.
Exact forms	J_τ and Δ_τ with residue-equivalence computation rules.
States and registries	Audited states, typed residue components, dependencies, and histories.
Legal transitions	Raw transitions, backward residue maps, obligations, transfers, and anti-collapse.
Proofs and closure	Rule systems, proof-tree initial algebras, soundness maps, and completeness certificate spaces.
Answers and scopes	Global answer objects, scope maps, outside-scope residues, promotion certificates, and answer records.
Failures	Stratified failure registries and failure-registration constructors.

The term generator $G[\Gamma]$ is the freely generated ARC model over the context Γ. It starts with the variables of Γ, closes under all constructors, and quotients by the homotopy-coherent equations required by ARC.

Generated constructors include:
type formation; typed comparison; cospan formation; residue pullback; J; Δ; readout fiber; conjunction; alternative; dependent sequencing; universal aggregation; compression loss; registry formation; anti-collapse; obligation; legal transition; proof-tree construction; reconstruction fiber; substrate fiber; calibration fiber; law residue; failure successor; exact normalization; presentation atlas; global answer section; scope map; outside-scope residue; legal promotion; answer record; admissibility residue.

17. Initiality

Initiality states that $G[\Gamma]$ is the canonical audited generator over Γ. Any other system satisfying the ARC laws receives a unique structure-preserving interpretation of $G[\Gamma]$.

Let ARCMod_Γ be the infinity-category of ARC models with an interpretation of Γ.

Initiality theorem:
$\mathrm{Map}_{\mathrm{ARCMod}_\Gamma}(G[\Gamma], A) \simeq \mathbb{1}$

for every ARC model A over Γ.

Initiality Theorem. For every context Γ, $G[\Gamma]$ is homotopy-initial in ARCMod_Γ.
Proof. Existence is by recursive interpretation of generated terms in any target ARC model. Well-definedness follows because every equation imposed in $G[\Gamma]$ is required to hold in every ARC model. Uniqueness follows by induction on term formation: any two structure-preserving maps agree on Γ and must agree on every generated constructor output up to contractible coherence.

Initiality does not imply a universal flat solver. If a universal flat solver existed, every target ARC model would have to interpret it as a flatness certificate for every residue. Target models may contain empty or noncontractible residues, so such a term cannot exist.

Universal Exact Continuation. Exact continuation exists for every residue, but flat solution does not. The terms $J_\tau(E)$ and $\Delta_\tau(E)$ are universal because they preserve residue; they do not assert that the residue is terminal.

18. Non-Collapse, Answer Soundness, and Operational Adequacy

The major soundness theorems are structural consequences of the formation rules. Every rule either preserves residue exactly, accounts for loss fibers, transfers closure with a certificate, retains scope, or registers failure.

Non-Collapse Soundness. Every derivable ARC answer either preserves every generated distinction, assigns each erased distinction a lawful fate, or registers the missing fate as failure residue.
Proof. Proceed by induction on derivations. Cospan formation generates residue. Exact normal forms preserve residue. Compression computes loss fibers. Scope promotion computes outside-scope fibers. Descent computes descent fibers. Legal transition requires flatness, reconstruction, or ledgering of every old component. Admissibility prevents ungenerated names. Failure registration converts missing certificates into typed residues.

Answer Soundness. If $\Gamma \vdash R : \mathrm{AnsRec}(\Gamma,X,F,\lambda)$, then R is exactly one of: a global answer, a scoped answer with provenance, a legally promoted answer with outside-scope fiber accounting, or a registered failure residue.
Proof. The answer-record type is defined as the coproduct of those four constructors. No fifth constructor exists.

Operational Adequacy. Every validated operational output is derivable as an ARC answer record. Solve mode requires structured flatness, Exact mode returns residue-preserving normal form, Practical mode requires loss-fiber accounting, and Explore mode returns legal futures.

19. Completion Certificate and Obstruction Register

ARC 1.0 is complete as a formal mathematical specification relative to its ambient foundation and the explicit certificate obligations below. When an implementation lacks one of these obligations, ARC does not collapse; it registers the missing certificate as residue.

ARCComplete := (Foundation, Judgments, FormationRules, ComputationRules, AdmissibilityCompiler, ModelSemantics, TermModel, Initiality, NonCollapseSoundness, ScopeSoundness, OperationalAdequacy, FailureClosure)

Meta-obligation	Status inside ARC
UniverseCert	Required to bound carriers, syntax, and certificate spaces.
SmallnessCert	Required for small indexing categories, registries, and presentation spaces.
AccessibilityCert	Required for generated localizations, initial algebras, and some colimits.
LocalizationCert	Required when a generated type is a reflective localization.
TypeTranslationCoherenceCert	Required for cross-type comparison and ledger composition.
HITQuotientCert	Required for a proof-assistant implementation of homotopy-coherent quotient syntax.
InitialAlgebraCert	Required for proof-tree objects $\mathrm{Sol}_R = \mu\, \Phi_R$.

FinalCoalgebraCert	Required for infinite legal process objects.
SequentialColimitCert	Required for FailReg$_\omega$.

Completion is obstruction-nullity: every generated survivor is legally neutralized, lawfully supported, preserved, or registered as exact obstruction.

Appendix A. Core Inference Rules

A.1 Problem and Residue Rules

Problem formation:

$$\frac{\Gamma \vdash C : V_\tau \quad \Gamma \vdash B : V_\tau \quad \Gamma \vdash S : V_\tau \quad \Gamma \vdash r : C \to B \quad \Gamma \vdash s : S \to B}{\Gamma \vdash (C \to_r B \leftarrow_s S) : \mathrm{Prob}_\tau}$$

Residue formation:

$$\frac{\Gamma \vdash P = (C \to B \leftarrow S) : \mathrm{Prob}_\tau}{\Gamma \vdash \mathrm{Res}(P) := C \times_B S : V_\tau}$$

A.2 Exact Form Rules

Exact forms:

$$\frac{\Gamma \vdash E : V_\tau}{\Gamma \vdash J_\tau(E) : \mathrm{Prob}_\tau} \qquad \frac{\Gamma \vdash E : V_\tau}{\Gamma \vdash \Delta_\tau(E) : \mathrm{Prob}_\tau}$$

Computation:

$$\mathrm{Res}(J_\tau(E)) \simeq E \qquad \mathrm{Res}(\Delta_\tau(E)) \simeq E$$

A.3 Legal Transition Rule

Legal transition:

$$\frac{\Gamma \vdash f : \mathrm{Raw}(\alpha,\beta) \quad \Gamma \vdash \rho : \mathrm{Rem}(\beta) \to \mathrm{Rem}(\alpha) \quad \Gamma \vdash o : \mathrm{Obl}(f,\rho) \quad \Gamma \vdash t : \mathrm{Trans}(\rho) \quad \Gamma \vdash a : \mathrm{AntiCollapse}(f)}{\Gamma \vdash (f,\rho,o,t,a) : \mathrm{Leg}_1(\alpha,\beta)}$$

A.4 Scope and Answer Rules

Scoped answer:
$\Gamma \vdash \text{scope } q : A_0 \to A$
$\Gamma \vdash a : \mathbb{1} \to A$
$\Gamma \vdash \text{retain}(q,a)$

$\Gamma \vdash \text{Scoped}(q,a,\text{retain}) : \text{AnsRec}(\Gamma,X,F,\lambda)$

Promoted answer:
$\Gamma \vdash \text{scope } q : A_0 \to A$
$\Gamma \vdash a : \mathbb{1} \to A$
$\Gamma \vdash c : \text{LegalProm}_q(a)$

$\Gamma \vdash \text{Promoted}(q,a,c) : \text{AnsRec}(\Gamma,X,F,\lambda)$

Failure answer:
$\Gamma \vdash C : \text{CertSpace}$
$\Gamma \vdash \text{Registered}(\text{label},C)$

$\Gamma \vdash \text{Failure}(\text{label},C) : \text{AnsRec}(\Gamma,X,F,\lambda)$

A.5 Admissibility Rule

Admissibility:
$\text{Adm}_\Gamma(S,x) := \text{Gen}_\Gamma(S,x) + \text{CtxAdm}_\Gamma(S,x)$

Generated branch:
$g : \text{Gen}_\Gamma(S,x)$

$\Gamma \vdash x : S$

Context branch:
$a : \text{CtxAdm}_\Gamma(S,x)$

$\Gamma, x{:}S \vdash x : S$

Failure branch:
$\text{Adm}_\Gamma(S,x)$ required but no branch supplied

$\Gamma \vdash \text{Failure}(\text{AdmissibilityRes}, \text{Adm}_\Gamma(S,x))$

Appendix B. Theorem Index

Theorem	Claim
Residue Localization	Formal problems modulo residue-equivalence are typed residues.
Exact Continuation	$J_\tau(E)$ and $\Delta_\tau(E)$ preserve residue for every typed residue E.
Specific-Arrow Conservativity	Support flatness promotes to structured flatness only through the exact support-conservativity certificate.
Componentwise Flatness	Componentwise registry flatness implies aggregate remainder flatness.
Anti-Collapse	A legal transition deletes no old residue silently.

Theorem	Claim
Failure Closure	Stratified failure registries close generated finite-stage failures.
Admissibility Soundness	No name is usable without generatedness, context admission, or failure registration.
Presentation Bridge	Global answers are coherent sections over the presentation atlas.
Descent Soundness	Local answers globalize only through a flat, reconstructed, or ledgered descent fiber.
No Unscoped Answer	A visible scoped answer promotes only through outside-scope fiber accounting.
Composite Loss	Total scope loss decomposes by pullback associativity.
Practical Closed Form	A representation is closed exactly when all selected loss fibers have lawful fates.
Proof-Tree Soundness	Proof trees map to closure certificates.
Initiality	$G[\Gamma]$ is homotopy-initial in ARCMod_{Γ}.
Non-Collapse Soundness	Every derivable answer preserves, accounts for, or registers all erased distinctions.
Answer Soundness	Every answer record is global, scoped, promoted, or failure-registered.
Operational Adequacy	Every validated operational output is derivable as an answer record.

Appendix C. Implementation Notes

This manuscript is a formal specification. A machine-checked implementation requires selecting a proof assistant or formal foundation capable of representing the relevant homotopy-coherent structures, or else replacing those structures with a stricter 1-categorical or set-level fragment.

A practical mechanization can proceed module by module. A minimal first implementation may restrict carriers to ordinary types or sets, replace homotopy pullbacks with ordinary pullbacks, and treat certificate spaces as ordinary propositions or types. The full ARC system is recovered by adding higher identity structure, homotopy coherence, dependent sums and products, reflective localizations, and higher-categorical model semantics.

1. Implement the core cospan and residue calculus: Prob_{τ}, $\mathrm{Res}(P)$, J, Δ, and residue-equivalence.
2. Implement certificate spaces for flatness, reconstruction, ledgering, and failure registration.
3. Implement registries and legal transitions with anti-collapse products over old components.
4. Implement scope maps, outside-scope residues, answer records, and legal promotion.
5. Implement admissibility: generatedness, context admission, and failure registration for names.
6. Implement proof trees as initial algebras only after the required accessibility hypotheses are fixed.
7. State and prove initiality for the chosen syntax/model category.

The implementation should preserve the governing invariant: every constructor that forgets structure must produce an explicit fiber or a certificate explaining the fate of that fiber. This invariant is the central regression test for any implementation of ARC.

Final Compression

A problem is a cospan:
$C \to B \leftarrow S$

Its content is a pullback residue:
$E = C \times_B S$

A solution is flatness of that residue.
An exact answer preserves that residue.
A practical answer accounts for every loss fiber.
A scoped answer retains its scope.
A global answer descends across presentations and local data.
A legal transition preserves old residue by flatness, reconstruction, or ledgering.
A missing certificate is residue.
A generated object must be admissible.
The canonical generator is initial among all systems obeying these laws.

ARC 1.0: the initial non-collapsing audited residue calculus.