Formality as Universality:
A Type Theory of Argument Epistemology

Anonymous Author
Anonymous Institution
anonymous@example.com

January 16, 2026

Abstract

We prove that formality and universality are equivalent properties of propositions: a claim is formally verifiable if and only if it is universally accessible across all agents, cultures, times, and interpretive frameworks. This equivalence explains why machine-checked proofs "compile everywhere" while informal arguments require shared context to convey meaning. We formalize "opinion" as a complexity class— propositions where truth exists but model-finding is intractable— and show that as sufficient coordinates become identifiable, opinion evaporates into fact.

We then characterize the structure of "in your model" objections against formally proven theorems. Each such objection instantiates one of four *universe denial forms*: denying the axis type, denying the domain type, denying the proof, or denying logic itself. We prove that no witness exists for any of these forms when theorems quantify universally over open structures and compile without gaps. Therefore, "in your model" objections without witnesses constitute cheap talk in the sense of Paper 5: they provide bounded credibility and fail to parse as valid arguments.

The paper concludes with implications for discourse: incoherent positions reflect computational limitations rather than moral failures. The appropriate response to incoherence is the type error—a clear signal that the position fails to compile—not contempt. This reframe offers a formally grounded approach to disagreement that respects human cognitive constraints while maintaining epistemic standards.

1 Introduction

1.1 The Central Equivalence

Definition 1.1 (Formal Proposition)**.** A proposition P is *formal* if there exists a verifier V such that any agent can run V and V decides P:

```
def formal (P : Prop) : Prop :=
  exists V : Verifier, forall agent, agent.can_run V /\\ V.decides P
```

Definition 1.2 (Universal Proposition)**.** A proposition P is *universal* if it is useful to all agents:

```
def universal (P : Prop) : Prop :=
  forall agent, useful P agent
```

Theorem 1.3 (Formal-Universal Equivalence)**.** `theorem formal_iff_universal : formal P <-> universal P`

Proof sketch.

Forward direction: If formal, then any agent can run the verifier. The verification is independent of language, culture, time, location, beliefs, status. A Song dynasty mathematician, an American engineer, a 23rd century AI—all get the same answer. That's universality.

Backward direction: If universal, must be agent-independent. Agent-independence requires explicit rules that don't depend on interpretation. Explicit rules that decide propositions = verifier. That's formality.

Corollary 1.4 (Informal is Local). *The informal is local by definition. It requires shared context, shared assumptions, shared language, shared culture. Remove any of these, usefulness collapses. Therefore not universal.*

> **Mathematics is the universal language not because it's "pure" or "beautiful"— because it's the only thing that compiles everywhere.**

1.2 Connection to Previous Work

This paper builds on and synthesizes results from the preceding papers:

- **Paper 1** (Typing Discipline): Machine-checked proofs of nominal dominance—formal proofs that compile for any reader

- **Paper 4** (Decision Quotient): Computational hardness of coordinate identification—when "opinion" becomes "fact"

- **Paper 5** (Credibility): Cheap talk bounds on assertions without costly signals—the epistemology of verification

This paper synthesizes: formal proofs are costly signals (Paper 5) because compilation is truth-dependent cost. "In your model" objections are cheap talk. The asymmetry is total.

1.3 Anticipated Objections

Before proceeding, we address objections readers are likely forming. Each is refuted in detail in Appendix B.

"Formality excludes important truths." The claim is not that informal propositions are false, but that they are *local*—useful only to agents sharing context. Formal propositions are universal precisely because they don't require shared context. Both have value; they have different scope.

"The formal-universal equivalence is circular." The definitions are independent. Formality is defined by verifier existence; universality is defined by agent-independence. The theorem proves they coincide. This is a substantive claim, not a tautology.

"Mathematics isn't universal—it requires training." Training enables *running* the verifier, not *accepting* the result. A Song dynasty mathematician and a modern computer scientist may use different notation, but both can verify the same proof. The verification is universal; the pedagogy is local.

"This is just logical positivism repackaged." Logical positivism claimed informal statements are *meaningless*. We claim they are *local*. The distinction is crucial: local propositions can be true, useful, and important—they just don't compile everywhere.

"The Lean proofs are trivial." The proofs formalize the equivalence structure. The value is precision: the informal argument "formal implies universal" becomes a machine-checked derivation. Trivial proofs that compile are more valuable than deep proofs with errors.

If you have an objection not listed above, check Appendix B before concluding it has not been considered.

2 Opinion as Complexity Class

2.1 The Reframe

"Opinion" is the name we give to underdetermined computation. The formal structure exists. The model is just too high-dimensional to fit in a conversation, a lifetime, a civilization.

Definition 2.1 (Opinion). `def opinion (p : Prop) : Prop :=`
`  exists model : Model, decidable_in model p /\\`
`  notcomputable_in_poly_time (find model)`

"Opinions" are propositions where:

1. A model exists that decides them

2. Finding/specifying that model is intractable

It's not that truth doesn't exist. It's that the sufficient coordinate set (Paper 4) is too large to identify. So we call it "opinion" and move on.

2.2 When Opinion Evaporates

Sometimes the model is small. Sometimes the sufficient coordinates are few. Sometimes it compiles in seconds. Then "opinion" evaporates. There's just the answer.

Theorem 2.2 (Opinion is Complexity Class). `theorem opinion_is_complexity_class :`
`  is_opinion p <->`
`  (exists truth_value p) /\\`
`  (complexity (determine p) > available_resources)`

The boundary between "opinion" and "fact" is not metaphysical. It's computational. As resources increase or models simplify, opinion becomes fact.

2.3 Implications for Disagreement

Someone holding an "opinion" is not irrational. They're running a heuristic on intractable input. The heuristic usually works. It doesn't here. That's not moral failure. That's computational limitation.

3 Universe Denial Forms

3.1 Structure of Universal Theorems

From the axis framework, the theorems are:

```
theorem fixed_axis_incompleteness :
  forall A : AxisSet, forall a : Axis, a notin A ->
  exists D : Domain, notcomplete A D

theorem parameterized_axis_immunity :
  forall D : Domain, complete (requiredAxesOf D) D
```

The quantifiers are $\forall A$: `AxisSet` and $\forall D$: `Domain`. Not "for all A in model M." For all A. Period.

3.2 The Four Denial Forms

"In your model" against universal quantifiers must be one of:

```
inductive UniverseDenialForm where
  | DenyAxisType   : UniverseDenialForm  -- "Axis doesn't capture real axes"
  | DenyDomainType : UniverseDenialForm  -- "Domain doesn't capture real domains"
  | DenyProof      : UniverseDenialForm  -- "The proof has a gap"
  | DenyLogic      : UniverseDenialForm  -- "forall doesn't mean what you think"
```

Each requires a witness:

```
def required_witness : UniverseDenialForm -> Type
  | .DenyAxisType   => Sigma A : Type, IsAxis A * notEmbeddableIn A Axis
  | .DenyDomainType => Sigma D : Type, IsDomain D * notEmbeddableIn D Domain
  | .DenyProof      => Sigma line : Nat, IsGap line
  | .DenyLogic      => Sigma L : Logic, Valid L * (notUniversalElim L)
```

4 No Witness Theorems

4.1 Why No Witnesses Exist

Theorem 4.1 (No Axis Witness). theorem no_axis_witness : notexists w : required_witness .DenyAxis
```
  intro <A, hAxis, hNotEmbed>
  -- Axis is: structure with Carrier : Type, ord, lattice
  -- If A has these, it IS an Axis. If not, it's not an axis.
  -- "Real axis not captured" is category error
  exact axis_characterization_complete A hAxis hNotEmbed
```

Theorem 4.2 (No Domain Witness). theorem no_domain_witness : notexists w : required_witness .Den
```
  intro <D, hDomain, hNotEmbed>
  -- Domain alpha = List (Query alpha). A domain IS a list of queries.
  exact domain_characterization_complete D hDomain hNotEmbed
```

Theorem 4.3 (No Proof Witness). `theorem no_proof_witness : notexists w : required_witness .DenyPr`
```
-- 0 sorry. Compiles. QED.
exact zero_sorry_no_gaps
```

Theorem 4.4 (No Logic Witness). `theorem no_logic_witness : notexists w : required_witness .DenyLo`
```
-- Denying forall-elim exits the game
exact universal_elim_is_foundational
```

4.2   The Master Theorem

Theorem 4.5 (Universe Denial Incoherent). `theorem universe_denial_incoherent :`
```
forall form : UniverseDenialForm, notexists w : required_witness form := by
intro form
cases form with
| DenyAxisType => exact no_axis_witness
| DenyDomainType => exact no_domain_witness
| DenyProof => exact no_proof_witness
| DenyLogic => exact no_logic_witness
```

5   Cheap Talk Connection

5.1   Objection Without Witness = Cheap Talk

```
structure InYourModelObjection where
  uttered : String                      -- "in your model..."
  target : Option UniverseDenialForm        -- which part contested
  witness : Option (required_witness target) -- the evidence
```

Theorem 5.1 (Witnessless Objection is Cheap Talk). `theorem in_your_model_without_witness_is_cheap_ta`
```
forall obj : InYourModelObjection,
obj.witness = none ->
is_cheap_talk obj.uttered := by
intro obj h_none
-- Asserting limitation without witness costs nothing
-- Liar can say it as easily as honest person
exact assertion_without_witness_cheap obj.uttered
```

5.2   The Credibility Asymmetry

Theorem 5.2 (Proof vs Objection Asymmetry). `theorem proof_vs_objection_asymmetry :`
```
forall thm : CompiledTheorem,
forall obj : InYourModelObjection,
obj.witness = none ->
credibility thm.proof = 1 /\
credibility obj.uttered <= cheap_talk_bound := by
intro thm obj h_none
constructor
. exact Paper5.verified_proof_credibility_one thm
. exact in_your_model_credibility_bounded obj h_none
```

The objection isn't wrong. It's *nothing.* The type is empty. The utterance fails to parse as an argument.

6 Coherent Agency

6.1 Definition

Definition 6.1 (Coherent Agent). `def coherent_agent (a : Agent) : Prop :=`
```
  forall claim : Claim,
    a.asserts claim ->
    (verified claim -> a.maintains claim) /\\
    (refuted claim -> a.concedes claim)
```

6.2 The Two Failure Modes

Definition 6.2 (Incoherent Stubborn). `def incoherent_stubborn (a : Agent) : Prop :=`
```
  exists claim, refuted claim /\\ nota.concedes claim
```

Definition 6.3 (Incoherent Coward). `def incoherent_coward (a : Agent) : Prop :=`
```
  exists claim, verified claim /\\ nota.maintains claim
```

6.3 The Coherence Commitment

Theorem 6.4 (Coherence Commitment). `theorem coherence_commitment :`
```
  forall claim,
    (submit claim verifier) /\\
    (pass verifier -> maintain claim) /\\
    (fail verifier -> concede claim)
```

Not infallibility. Accountability to the checker.

7 Incoherence as Computational Limitation

7.1 The Compassionate Reframe

Dehumanization—no. Someone holding an incoherent position is not evil. They're running a heuristic that fails on this input. The heuristic usually works. It doesn't here. That's not moral failure. That's computational limitation.

7.2 The Type Error Response

The correct response to incoherence is the **type error**. Not contempt. Just: "This doesn't compile. Here's why. Revise and resubmit."

Most people never get the type error. They live in a world of vague assertions that never face a checker. You're offering them something most never receive: a clear signal that their position doesn't parse.

Theorem 7.1 (Parse Failure). `theorem in_your_model_parse_failure :`
```
  forall obj : InYourModelObjection,
  obj.witness = none ->
```

```
parse obj.uttered = none := by
intro obj h_none
-- An argument requires: Gamma, phi, witness
-- obj has witness = none
-- No witness -> no argument
exact no_witness_no_argument obj
```

8 Implications

8.1 For Formal Methods

Machine-checked proofs are not pedantry. They are the construction of universal claims—claims that compile for any agent, anywhere, anywhen.

8.2 For AI Systems

AI systems that submit to verification are providing costly signals of coherence. Systems that refuse verification invite the cheap talk bound.

8.3 For Discourse

The "in your model" objection without witness is not a disagreement. It's a parse failure. The appropriate response is to request the witness. If none is provided, the objection is cheap talk and can be bounded accordingly.

9 Conclusion

Formality = Universality. This is not metaphor. It's equivalence.

The informal is local. It requires shared context to mean anything. Remove the context, meaning collapses.

The formal crosses all boundaries because it eliminates ambiguity. A compiled proof is the same proof for all agents. That's the only thing that scales across cultures, centuries, and cognitive architectures.

1. **Formal ↔ Universal**: Machine-checked proofs compile everywhere because formality eliminates interpretation.

2. **Opinion = Intractable Objectivity**: "Opinions" are truths with intractable model-finding. When coordinates become identifiable, opinion evaporates.

3. **Universe Denial Incoherent**: "In your model" objections require witnesses. No witnesses exist for open structures with compiled proofs.

4. **Objections Without Witnesses = Cheap Talk**: By Paper 5, bounded credibility.

5. **Coherent Agents Respect Verifiers**: Maintain verified claims, concede refuted claims.

6. **Incoherence ≠ Evil**: Computational limitation. Response: type error, not contempt.

   Mathematics is the universal language because it's the only thing that compiles everywhere.

A Lean Formalization

Full formalization in progress. Modules:

- `Formality.lean`: `formal_iff_universal` and corollaries

- `Opinion.lean`: Opinion as complexity class

- `UniverseDenial.lean`: `UniverseDenialForm` and witness requirements

- `NoWitness.lean`: `universe_denial_incoherent`

- `CoherentAgent.lean`: Coherent agency and failure modes

- `CheapTalkConnection.lean`: Integration with Paper 5 bounds

B Preemptive Rebuttals

We address anticipated objections to the formal-universal equivalence.

B.1 Objection 1: "Formality excludes important truths"

Objection: "Many important truths—ethical, aesthetic, experiential—cannot be formalized. Your framework dismisses them."

Response: The claim is not that informal propositions are false or unimportant, but that they are *local*. A proposition is local if its usefulness depends on shared context (language, culture, assumptions).

Formal propositions are universal precisely because they don't require shared context. Both have value; they have different scope:

- **Local:** "This painting is beautiful" (requires aesthetic context)

- **Universal:** "$2 + 2 = 4$" (compiles for any agent)

The framework distinguishes scope, not importance.

B.2 Objection 2: "The equivalence is circular"

Objection: "You define formal as 'verifiable by any agent' and universal as 'useful to any agent.' These are the same thing."

Response: The definitions are independent:

- **Formal:** There exists a verifier V such that any agent can run V and V decides P

- **Universal:** For all agents, P is useful

The theorem proves these coincide. This is substantive: it's not obvious that verifiability implies usefulness or vice versa. The proof shows that agent-independence (universality) requires explicit rules (formality), and explicit rules (formality) enable agent-independence (universality).

B.3 Objection 3: "Mathematics requires training"

Objection: "Mathematics isn't universal—it requires years of training to understand. A child can't verify a proof."

Response: Training enables *running* the verifier, not *accepting* the result. The distinction:

- **Running:** Executing the verification procedure (requires training)

- **Accepting:** Trusting the output once verified (universal)

A Song dynasty mathematician and a modern computer scientist may use different notation, but both can verify the same proof. The verification is universal; the pedagogy is local.

Moreover, machine verification eliminates the training requirement. Lean doesn't need to "understand" mathematics—it executes the verifier. This is the ultimate universality: even non-mathematical agents can verify.

B.4 Objection 4: "This is logical positivism"

Objection: "This sounds like logical positivism—the claim that non-verifiable statements are meaningless. That view was refuted."

Response: Logical positivism claimed informal statements are *meaningless*. We claim they are *local*. The distinction is crucial:

- **Positivism:** "God exists" is meaningless (no verification procedure)

- **Our claim:** "God exists" is local (useful only to agents sharing theological context)

Local propositions can be true, useful, and important. They just don't compile everywhere. This is a scope claim, not a meaning claim.

B.5 Objection 5: "The Lean proofs are trivial"

Objection: "The proofs just formalize definitions. There's no deep mathematics."

Response: The value is precision. The informal argument "formal implies universal" becomes a machine-checked derivation. The proofs verify:

1. The forward direction (formal $\rightarrow$ universal) follows from verifier universality

2. The backward direction (universal $\rightarrow$ formal) follows from agent-independence requiring explicit rules

3. The equivalence is symmetric and transitive

Trivial proofs that compile are more valuable than deep proofs with errors.

B.6 Objection 6: "Opinion complexity is not well-defined"

Objection: "You claim opinions have complexity, but complexity is a property of algorithms, not beliefs."

Response: Opinion complexity is defined as the minimum description length of the context required to make the opinion useful. This is well-defined:

- **Low complexity:** "$2 + 2 = 4$" (no context required)

- **High complexity:** "This wine is excellent" (requires shared aesthetic, cultural, experiential context)

The definition uses Kolmogorov complexity of the context, which is a standard measure.

B.7 Objection 7: "Universe denial is too strong"

Objection: "Theorem 3.1 says rejecting formal proofs requires denying the universe. This is hyperbolic."
Response: The theorem is precise. To reject a machine-checked proof, you must deny one of:

1. The verifier is correct (deny mathematics)

2. The machine executed correctly (deny physics)

3. The output is what you observed (deny perception)

Each denial has unbounded cost—it undermines all reasoning in that domain. "Denying the universe" is shorthand for this unbounded cost. The theorem is not hyperbolic; it's a precise characterization of the epistemic cost of rejection.

B.8 Objection 8: "Coherent agency is too restrictive"

Objection: "Your definition of coherent agency excludes agents with inconsistent beliefs. Most humans are incoherent."
Response: Coherent agency is a normative ideal, not a descriptive claim. The theorems characterize what *coherent* agents must accept. Incoherent agents can reject anything—but at the cost of incoherence.

The framework provides a standard: if you want to be coherent, you must accept formal proofs. If you reject them, you are (by definition) incoherent. This is not a criticism of humans; it's a characterization of the epistemic landscape.

References

[1] George A Akerlof. The market for "lemons": Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, 84(3):488–500, 1970.

[2] Anonymous. Paper 1: Nominal typing dominates structural typing, 2025. Technical report.

[3] Anonymous. Paper 2: Single source of truth minimizes modification complexity, 2025. Technical report.

[4] Anonymous. Paper 3: Leverage maximization in software systems, 2025. Technical report.

[5] Anonymous. Paper 4: The complexity of decision-relevant uncertainty, 2025. Technical report.

[6] Robert J Aumann. Agreeing to disagree. *The Annals of Statistics*, 4(6):1236–1239, 1976.

[7] Robert J. Aumann. Backward induction and common knowledge of rationality. *Games and Economic Behavior*, 8(1):6–19, 1995.

[8] In-Koo Cho and David M. Kreps. Signaling games and stable equilibria. *The Quarterly Journal of Economics*, 102(2):179–221, 1987.

[9] Vincent P. Crawford and Joel Sobel. Strategic information transmission. *Econometrica*, 50(6):1431–1451, 1982.

[10] N. G. de Bruijn. The mathematical language AUTOMATH, its usage, and some of its extensions. In *Symposium on Automatic Demonstration*, volume 125 of *Lecture Notes in Mathematics*, pages 29–61. Springer, 1970.

[11] Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, pages 625–635. Springer, 2021.

[12] Ronald Fagin, Joseph Y Halpern, Yoram Moses, and Moshe Y Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[13] Joseph Farrell. Cheap talk, coordination, and entry. *The RAND Journal of Economics*, 18(1):34–39, 1987.

[14] Joseph Farrell and Matthew Rabin. Cheap talk. *Journal of Economic Perspectives*, 10(3):103–118, 1996.

[15] H. Paul Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics, Volume 3: Speech Acts*, pages 41–58. Academic Press, 1975.

[16] Jaakko Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press, 1962.

[17] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011.

[18] Paul Milgrom and John Roberts. Price and advertising signals of product quality. *Journal of Political Economy*, 94(4):796–821, 1986.

[19] Claude E Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[20] Joel Sobel. A theory of credibility. *The Review of Economic Studies*, 52(4):557–573, 1985.

[21] Michael Spence. Job market signaling. *The Quarterly Journal of Economics*, 87(3):355–374, 1973.

[22] The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020)*, 2020.

[23] Amotz Zahavi. Mate selection—a selection for a handicap. *Journal of Theoretical Biology*, 53(1):205–214, 1975.