Formality as Universality: A Type Theory of Argument Epistemology

Anonymous Author
Anonymous Institution
anonymous@example.com

January 3, 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 :=
   V : Verifier,  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 :=
   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.

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 :=
  model : Model, decidable_in model p
¬computable_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
( 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 :
   A : AxisSet,  a : Axis, a  A →
   D : Domain, ¬complete A D

theorem parameterized_axis_immunity :
   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  -- " doesn't mean what you think"
```

Each requires a witness:

```
def required_witness : UniverseDenialForm → Type
  | .DenyAxisType   =>  A : Type, IsAxis A × ¬EmbeddableIn A Axis
  | .DenyDomainType =>  D : Type, IsDomain D × ¬EmbeddableIn D Domain
  | .DenyProof      =>  line : Nat, IsGap line
  | .DenyLogic      =>  L : Logic, Valid L × (¬UniversalElim L)
```

4 No Witness Theorems

4.1 Why No Witnesses Exist

Theorem 4.1 (No Axis Witness). `theorem no_axis_witness : ¬ w : required_witness .DenyAxisType :=`

```
  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 : ¬ w : required_witness .DenyDomainT`

```
  intro ⟨D, hDomain, hNotEmbed⟩
  -- Domain  = List (Query ). A domain IS a list of queries.
  exact domain_characterization_complete D hDomain hNotEmbed
```

Theorem 4.3 (No Proof Witness). `theorem no_proof_witness : ¬ w : required_witness .DenyProof := b`

```
  -- 0 sorry. Compiles. QED.
  exact zero_sorry_no_gaps
```

Theorem 4.4 (No Logic Witness). `theorem no_logic_witness : ¬ w : required_witness .DenyLogic := b`

```
  -- Denying -elim exits the game
  exact universal_elim_is_foundational
```

4.2 The Master Theorem

Theorem 4.5 (Universe Denial Incoherent). `theorem universe_denial_incoherent :`

```
   form : UniverseDenialForm, ¬ 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`

```
   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 :`

```
   thm : CompiledTheorem,
   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 :=`

```
   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 :=`

```
   claim, refuted claim ∧ ¬a.concedes claim
```

Definition 6.3 (Incoherent Coward). `def incoherent_coward (a : Agent) : Prop :=`

```
   claim, verified claim ∧ ¬a.maintains claim
```

6.3 The Coherence Commitment

Theorem 6.4 (Coherence Commitment). `theorem coherence_commitment :`

```
   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 :
  obj : InYourModelObjection,
obj.witness = none →
parse obj.uttered = none := by
intro obj h_none
-- An argument requires: , , 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