OBINexus Truth: Formal Proof Framework

Eziokwu - Mathematical Foundation for AI Accountability

OBINexus Systems
Nnamdi Michael Okpala, Architect

June 2, 2025

Abstract

This document establishes the formal mathematical framework for OBINexus Truth's proof verification system, implementing the Declarative-Constructive hybrid logic through rigorous set theory and procedural definitions. The framework provides structured mechanisms for handling claimed proofs within AI accountability infrastructure while maintaining cultural sensitivity and human-centered design principles.

Contents

1 Foundation Set Definitions

Definition 1.1 (Universal Evidence Space). *Let $\mathcal{U}$ be the universal evidence space containing all possible evidence artifacts:*

$$\mathcal{U} = \{e : e \text{ is a digitally representable evidence artifact}\}$$

Definition 1.2 (Claim Bubble). *A claim bubble $\mathcal{C}$ is defined as a 4-tuple:*

$$\mathcal{C} = \langle A, E, T, \Phi \rangle$$

where:

- *A is the assertion set: $A = \{a_1, a_2, \ldots, a_n\}$ where each a_i is an atomic claim*
- *$E \subseteq \mathcal{U}$ is the evidence set supporting the claims*
- *T is the temporal constraint set: $T = [t_{start}, t_{end}] \times \mathbb{N}$*
- *Φ is the attribution mapping: $\Phi : A \to \mathcal{P}(Actors)$*

Definition 1.3 (Proof Space). *The proof space $\mathcal{PS}$ for a claim bubble $\mathcal{C}$ is defined as:*

$$\mathcal{PS}(\mathcal{C}) = \{p : p \text{ is a valid proof sequence for } \mathcal{C}\}$$

where each proof $p \in \mathcal{PS}(\mathcal{C})$ must satisfy both declarative and constructive constraints.

2 Declarative Proof Logic (7-Method Model)

Definition 2.1 (Declarative Proof Sequence). *A declarative proof sequence $D = \langle d_1, d_2, \ldots, d_7 \rangle$ for claim bubble $\mathcal{C}$ is defined where:*

$$d_1 : Assertion \to A \text{ (traceable claim establishment)} \tag{1}$$
$$d_2 : Evidence \to E \text{ (data and context provision)} \tag{2}$$
$$d_3 : Attribution \to \Phi \text{ (actor/system association)} \tag{3}$$
$$d_4 : Cross\text{-}Verification \to V(\mathcal{C}) \text{ (independent validation)} \tag{4}$$
$$d_5 : Contradiction\ Check \to \neg\exists c \in \mathcal{C}^c : c \models \neg A \tag{5}$$
$$d_6 : Temporal\ Validation \to T \models occurrence\ constraints \tag{6}$$
$$d_7 : Archival\ Recording \to H(\mathcal{C}, D) \text{ (cryptographic storage)} \tag{7}$$

Definition 2.2 (Cross-Verification Function). *The cross-verification function $V : \mathcal{C} \to \{0, 1\}$ is defined as:*

$$V(\mathcal{C}) = \begin{cases} 1 & \text{if } \exists E_{ind} \subseteq \mathcal{U} : E_{ind} \cap E = \emptyset \wedge E_{ind} \models A \\ 0 & \text{otherwise} \end{cases}$$

where E_{ind} represents independently sourced evidence.

3 Constructive Proof Overlay

Definition 3.1 (Constructive Proof Properties). *A constructive proof K for claim bubble $\mathcal{C}$ must satisfy:*

1. ***Executable Sequence**: $\exists f : \mathcal{C} \to \mathcal{O}$ where $\mathcal{O}$ is the set of observable outcomes*
2. ***Fault Traceability**: $\forall failure\ \phi \in K, \exists trace\ \tau : \tau \models reconstruction(\phi)$*
3. ***Evidence Transformability**: $\exists g : E \times Context \to E'$ where $E' \models A$ in new context*
4. ***No Black Box Doctrine**: $\forall result\ r \in K, \exists logic\ chain\ \ell : \ell \vdash r$*

Theorem 3.1 (Proof Completeness). *A proof p is complete for claim bubble $\mathcal{C}$ if and only if:*

$$p \in \mathcal{PS}(\mathcal{C}) \wedge D(p) = 1 \wedge K(p) = 1$$

where $D(p)$ indicates declarative validity and $K(p)$ indicates constructive validity.

4 Formal Procedure for Claimed Proof Handling

Procedure 4.1 (Claimed Proof Verification Protocol). *Given a claimed proof $\hat{p}$ for claim bubble $\mathcal{C}$, execute the following verification sequence:*

***Input**: Claimed proof $\hat{p}$, Claim bubble $\mathcal{C}$ **Output**: Verification result $v \in \{VALID, INVALID, INSUF$*

1. ***Claim Bubble Validation**:*

$$\begin{aligned} Validate(\mathcal{C}) &= check(A \neq \emptyset) \wedge check(E \subseteq \mathcal{U}) && (8)\\ &\wedge check(T \text{ is well-formed}) && (9)\\ &\wedge check(\Phi \text{ is total}) && (10) \end{aligned}$$

2. ***Evidence Integrity Check**:*

$$\forall e \in E : verify(hash(e)) \wedge validate(metadata(e))$$

3. ***Declarative Verification**:*

$$D_{valid}(\hat{p}) = \bigwedge_{i=1}^{7} d_i(\hat{p}) = true$$

4. ***Constructive Verification**:*

$$K_{valid}(\hat{p}) = executable(\hat{p}) \wedge traceable(\hat{p}) \wedge transformable(\hat{p}) \wedge transparent(\hat{p})$$

5. ***Cultural Sensitivity Assessment**:*

$$Cultural(\hat{p}) = respect(context) \wedge accommodate(neurodivergence) \wedge preserve(dignity)$$

6. ***Final Verification Decision**:*

$$v = \begin{cases} VALID & \text{if } D_{valid}(\hat{p}) \wedge K_{valid}(\hat{p}) \wedge Cultural(\hat{p}) \\ INVALID & \text{if } \neg D_{valid}(\hat{p}) \vee \neg K_{valid}(\hat{p}) \vee \neg Cultural(\hat{p}) \\ INSUFFICIENT & \text{otherwise} \end{cases}$$

5 Proof Space Topology

Definition 5.1 (Proof Distance Metric). *For two proofs $p_1, p_2 \in \mathcal{PS}(\mathcal{C})$, define the proof distance:*

$$d(p_1, p_2) = \sum_{i=1}^{7} w_i \cdot |d_i(p_1) - d_i(p_2)| + \lambda \cdot |K(p_1) - K(p_2)|$$

where w_i are weights for declarative components and λ weights constructive differences.

Definition 5.2 (Proof Convergence). *A sequence of proofs $\{p_n\}$ converges to truth $\mathcal{T}$ if:*

$$\lim_{n\to\infty} d(p_n, \mathcal{T}) = 0$$

where $\mathcal{T}$ represents the ground truth for claim bubble $\mathcal{C}$.

6 Implementation Architecture

Definition 6.1 (Evidence Processing Pipeline). *The evidence processing function $\Pi : \mathcal{U} \to \mathcal{PS}$ is decomposed as:*

$$\Pi = H_7 \circ T_6 \circ C_5 \circ V_4 \circ A_3 \circ E_2 \circ I_1$$

where each Π_i corresponds to the i-th step of the declarative proof model.

Lemma 6.1 (Pipeline Composability). *If each stage Π_i preserves evidence integrity and cultural sensitivity, then the composed pipeline Π maintains these properties.*

7 Governance Integration

Definition 7.1 (Policy-Procedure Coupling). *For governance framework integration, define the coupling function:*

$$PMP : \mathcal{PS} \times \textit{Policies} \to \textit{Procedures}$$

such that every proof verification maps to executable governance procedures while maintaining the No Ghosting Protocol.

Theorem 7.1 (Accountability Preservation). *The OBINexus Truth framework preserves accountability if for every verified proof $p \in \mathcal{PS}(\mathcal{C})$:*

$$\exists \textit{audit trail } \alpha : \alpha \models \textit{complete traceability of } p$$

8 Conclusion

This formal framework establishes the mathematical foundation for OBINexus Truth's proof verification system, ensuring that AI accountability mechanisms maintain rigorous logical standards while preserving human-centered values and cultural sensitivity. The hybrid Declarative-Constructive approach provides both theoretical soundness and practical implementability for real-world AI transparency requirements.

Eziokwu - Truth as correct speech, implemented through heart-centered computational logic.