Title: Unified OBIAI Specification Document for GitHub Repository Integration

Author: Nnamdi Michael Okpala Organization: OBINexus Computing Repository: https://github.com/obinexus/pyobiai Date: June 2025

Abstract

This unified document integrates the key architectural, mathematical, and implementation specifications for the Ontological Bayesian Intelligence Architecture Infrastructure (OBIAI) and its associated symbolic and debiasing components. It consolidates elements from the Conceptual Symbolic Language Layer (CSL), the Formal Mathematical Reasoning System, and the Bayesian Bias Mitigation Framework into a cohesive documentation suite for the GitHub repository.

1. Architectural Overview

OBIAI is a tiered, modular framework organized into Stable, Experimental, and Legacy tiers. Each component supports transparent, deterministic AI for high-stakes applications, particularly in healthcare.

1.1 Component Tiers

- **Stable Tier**: Includes mathematically verified functions (e.g., Cost-Knowledge, Traversal Cost)
- **Experimental Tier**: In-progress modules like Triangle Convergence and Filter-Flash Inference
- **Legacy Tier**: Archived implementations maintained for auditability

1.2 Core Engine

Implements deterministic function resolution and semantic derivation trees to ensure architectural traceability and output consistency.

2. Formal Mathematical Foundations

2.1 Cost-Knowledge Function

Defined as: $C(K_t, S) = H(S) \cdot e^{-K_t}$ Ensures exponential decay of cost with increasing knowledge.

2.2 Traversal Cost Function

Defined as: $C(Node_i \to Node_j) = \alpha \cdot KL(P_i \| P_j) + \beta \cdot \Delta H(S_{i,j})$ Used to calculate the semantic cost of transitioning between belief states.

2.3 Verification Properties

- Monotonicity
- Non-negativity
- Numerical stability under entropy transitions

3. Conceptual Symbolic Language Layer (CSL)

3.1 Glyph Grammar

- Atomic Concept Mapping (e.g., $G_{node}, G_{seed}, G_{cloud}$)
- Compositional Grammar with operators: causal, temporal, intensity, uncertainty

3.2 Semantic Salience Function

$\Sigma(G_i, K_t, C_{cultural}) = \alpha P(concept_i|evidence_t) + \beta A(G_i) + \gamma C(K_t, S_i)$ Weights cultural and probabilistic relevance.

3.3 Cultural Validation

Uses tiered protocols: automated pattern checking, historical precedent, and community validation.

4. Bayesian Bias Mitigation Framework

4.1 Causal DAG Modeling

Defines relationships between confounders (S), conditions (C), outcomes (T), and protected attributes (A).

4.2 Hierarchical Bayesian Estimation

Marginalizes bias parameters: $P(\theta|D) = \int P(\theta, \phi|D)d\phi$

4.3 Fairness Guarantees

- Demographic parity enforcement: $|P(\hat{Y} = 1|A = a) - P(\hat{Y} = 1|A = a')| \leq \epsilon$
- Bias Reduction Theorem: $E[B(\theta_{Bayes}, D)] \leq E[B(\theta_{MLE}, D)] - \Delta$

5. Implementation Strategy

- Structured as per Aegis Waterfall Methodology
- Deployment-ready stable modules
- Cultural glyph visualizations integrated in UI layer
- Unit-tested Python implementations in `/stable`, `/experimental`, `/legacy` branches

6. Repository Notes

- Main codebase: https://github.com/obinexus/pyobiai
- CSL visualization tools and UI engines to be merged under `ui/` branch
- Future integration plans include `polygon` module for semantic cost-space mapping and glyph inference resolution

7. Conclusion

This unified technical specification provides a complete foundation for the GitHub `pyobiai` repository. It harmonizes rigorous mathematical proofs, debiasing strategies, and culturally grounded UI semantics to deliver a robust AI reasoning system.
