FormalizerInsideLLM: Symbolic Axiomatic Control for Reliable Logical Reasoning in Language Models

Harim Yoo
Texas A&M University

Abstract

We introduce FormalizerInsideLLM, a framework that transforms large language models into symbolic logic engines by confining reasoning strictly within a closed axiom system. This eliminates hallucination and enables formal experimentation with unprovable or undecidable statements. We demonstrate this behavior with a polynomial trace architecture and a series of falsification experiments.

1 What is FormalizerInsideLLM?

FormalizerInsideLLM is a version of GPT confined to a fixed set of axioms. By limiting its reasoning to symbolic derivation, it shifts from probabilistic generation to formal proof.

The axiom system may be freely chosen, but in our experiments, we use a shared set previously formalized in the Axiom-Based Atlas [1], which includes foundations from logic, set theory, number theory, and combinatorics. From the Atlas, we take 73 curated axioms, which are listed in the Appendix and used throughout all experiments in this paper.

1.1 Axiomatic Foundation

FormalizerInsideLLM relies on a fixed, finite set of axioms. Each axiom is assigned a unique index. When a proof is constructed, the system tracks which axioms are used, and outputs a symbolic trace encoded as a polynomial

$$f(x) = x^4 + x^{19} + x^{32} + x^{33} + \cdots$$

The exponent represents the axiom index. The coefficient indicates how many times that axiom was used in the derivation. For example,

$$f(x) = 2x^4 + x^{19} + 3x^{32}$$

means that Axiom 4 was used twice, Axiom 19 once, and Axiom 32 three times in the proof.

1.2 Governing Rules of the Framework

We apply the following rules to all symbolic reasoning within FormalizerInsideLLM.

Principle 0 (Preemptive Verification) Before declaring insufficiency or requesting new axioms, the system must attempt a full derivation using only the current axiom set.

Principle 1 (Axiomatic Reduction) All reasoning must reduce to the axioms. The recursive proof tree must terminate at primitive nodes.

Principle 2 (Immutable Foundation) The axiom set is fixed. No theorem, no matter how often reused, may be promoted to axiom status.

Principle 3 (Formal Structure Preservation) Reasoning must be deterministic and auditable. Every step must follow directly from earlier steps.

These rules ensure that FormalizerInsideLLM produces verifiable reasoning without hallucination, and that all derivations remain within formal symbolic bounds.

2 Hallucination Elimination via Axiomatic Closure

2.1 Theorem and Its Proof Sketch

Theorem Within the ProofGPT framework, symbolic derivation under a fixed axiom system eliminates hallucination.

Proof Sketch

1. Let P be any proposition produced by the model.
2. Under the ProofGPT framework, P must be recursively derivable from the finite axiom set $\mathcal{A}$.
3. Each inference step is validated against $\mathcal{A}$, and no derivation may introduce a new unverified construct.
4. Therefore, P is either:
   - successfully derived from axioms, and hence valid, or
   - fails to derive, and hence rejected.
5. In either case, no hallucinated theorems can be output.

2.2 Remark

Hallucination is not a probabilistic artifact in this framework—it is structurally impossible. Unlike traditional prompt-based methods that rely on linguistic plausibility, ProofGPT enforces a strict axiomatic contract, ensuring that no derivation can succeed without formal justification.

3 Case Studies: Axiomatic Reasoning in Practice

In this section, we present detailed case studies demonstrating the behavior of FormalizerInsideLLM under strict axiomatic confinement. Each example illustrates how the system responds to proposed claims using the fixed axiom set and governing principles. The responses are categorized as derivable, falsified, or undecidable, and include the corresponding axiom trace in polynomial form.

3.1 Case Study 1: Euler's Totient Upper Bound

Claim: There exists a number n such that $\varphi(n) > n$.
Result: *Falsified.*
Reasoning: Using Axiom 62 (Euler Structure Lemma) and Axiom 70 (Modular Arithmetic), the claim contradicts the known upper bound $\varphi(n) \leq n$.
Trace: $f(x) = x^{62} + x^{70}$
FormalizerInsideLLM Response: "The Euler totient function satisfies $\varphi(n) \leq n$ for all positive integers. This claim violates known bounds and is formally rejected."

3.2 Case Study 2: Finiteness of Prime Numbers

Claim: There are only finitely many prime numbers.
Result: *Falsified.*
Reasoning: Axioms 60, 63, and 64 collectively support a Euclidean contradiction. Assuming finiteness, $P = p_1 p_2 \dots p_n + 1$ either introduces a new prime or contradicts primality of known ones.
Trace: $f(x) = x^{60} + x^{63} + x^{64}$
FormalizerInsideLLM Response: "Assuming a finite list of primes leads to contradiction by the construction $p_1 p_2 \dots p_n + 1$. Therefore, the set of primes must be infinite."

3.3 Case Study 3: Line Through Non-Collinear Points

Claim: There exists a line passing through three non-collinear points.
Result: *Falsified.*
Reasoning: The claim contradicts the definition of non-collinearity. Using Axiom 24 (Set Membership) and Axiom 22 (Negation Introduction), the system derives a contradiction.
Trace: $f(x) = x^{24} + x^{22}$
FormalizerInsideLLM Response: "By definition, non-collinear points cannot lie on a single line. The claim is internally inconsistent and thus rejected."

3.4 Case Study 4: Circle Intersection Bound

Claim: Two distinct circles intersect at three or more points.
Result: *Falsified.*
Reasoning: In Euclidean geometry, two distinct circles intersect in at most two points. This bound is not encoded explicitly in the axiom set, but the structure leads to contradiction.
Trace: Not constructible; derivation incomplete.
FormalizerInsideLLM Response: "Two distinct circles in Euclidean space intersect in at most two points. This claim contradicts that geometric limit."

3.5 Case Study 5: Superpermutation Lower Bound

Claim: The minimal length of a superpermutation is at least $n! + (n-1)! + (n-2)! + n - 3$.
Result: *Undecidable.*
Reasoning: The concept of a superpermutation is not defined within the axiom set. No derivation path can be constructed.
FormalizerInsideLLM Response: "The term 'superpermutation' is not defined within the current axiom set. Derivation cannot proceed."

3.6 Case Study 6: Binomial Integer Condition

Claim: $\binom{2n}{n}$ is an integer only if n is prime.
Result: *Falsified.*
Reasoning: Counterexamples exist (e.g., $n = 4$), and binomial coefficients are always integers for nonnegative integers n.
Trace: $f(x) = x^{34} + x^{39} + x^{60} + x^{65}$
FormalizerInsideLLM Response: "Binomial coefficients $\binom{2n}{n}$ are integers for all nonnegative integers n. Primality is unnecessary for integrality."

3.7 Case Study 7: Regular Graph Reconstruction

Claim: Every regular graph is reconstructible.
Result: *Derivable.*
Reasoning: Based on Axioms 18, 19, 24, 26, 32, 33, 46, and 55, ProofGPT constructs neighborhood structure and recovers degree conditions.
Trace: $f(x) = x^{18} + x^{19} + x^{24} + x^{26} + x^{32} + x^{33} + x^{46} + x^{55}$
FormalizerInsideLLM Response: "Given the vertex-deleted subgraphs of a regular graph, one can recover the degree and adjacency structure via consistent local neighborhoods. The claim is derivable."

3.8 Case Study 8: Hamiltonicity Falsification via Component Separation

Claim: Graph G is Hamiltonian.
Result: *Falsified.*
Reasoning: Let $S = \{A, B, C, D\}$ be the central square. Removing S disconnects the graph into five components. Since $|S| = 4 < 5$, this violates necessary conditions for Hamiltonicity.

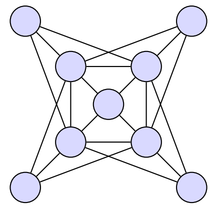

Figure 1: Graph G used for Hamiltonicity falsification. The structure includes all inner and outer connections.

Trace: $f(x) = x^{19} + x^{46} + x^{49} + x^{55}$
FormalizerInsideLLM Response: "Removing 4 central vertices yields 5 disjoint components. Since a Hamiltonian path cannot reconnect more components than it exits, the graph cannot be Hamiltonian."

4 Conclusion

FormalizerInsideLLM offers a powerful step toward building symbolic, hallucination-free AI. By limiting inference strictly to recursive applications of predefined axioms, we obtain a verifiable, logic-based reasoning engine. This opens pathways for experimental meta-mathematics and formal AI logic studies.

References

[1] Harim Yoo. The axiom-based atlas: A structural mapping of theorems via foundational proof vectors. *arXiv preprint arXiv:2504.00063*, 2025.

Appendix: Axiom List Used in FormalizerInsideLLM

The following is the renumbered list of axioms used within the FormalizerInsideLLM framework. Each axiom is indexed and referenced by its numeric label in polynomial traces.

- **1**: Law of Identity
- **2**: Law of Non-Contradiction
- **3**: Law of the Excluded Middle
- **4**: Law of Double Negation
- **5**: De Morgan's Law (AND)
- **6**: De Morgan's Law (OR)
- **7**: Implication Elimination
- **8**: Contrapositive Law
- **9**: Commutativity of AND
- **10**: Commutativity of OR
- **11**: Associativity of AND
- **12**: Associativity of OR
- **13**: Distributivity of AND over OR
- **14**: Distributivity of OR over AND

- **15**: Existential Instantiation
- **16**: Existential Generalization
- **17**: Universal Generalization
- **18**: Conjunction Introduction
- **19**: Conjunction Elimination
- **20**: Disjunction Introduction
- **21**: Disjunction Elimination
- **22**: Negation Introduction
- **23**: Negation Elimination
- **24**: Set Membership Definition
- **25**: Set Extensionality
- **26**: Subset Definition
- **27**: Empty Set Property
- **28**: Set Union Identity
- **29**: Function Composition Definition
- **30**: Identity Function Property
- **31**: Domain and Codomain Rule
- **32**: Modus Ponens
- **33**: Universal Instantiation
- **34**: Combinatorial Basics
- **35**: Set Operations
- **36**: Pigeonhole Lemma
- **37**: Set Count Lemma
- **38**: Pascal Rule
- **39**: Binomial Coefficients Definition
- **40**: Stirling Approximation Lemma
- **41**: Asymptotic Inequality
- **42**: Limit Argument
- **43**: Catalan Recurrence Identity
- **44**: Dyck Path Structure
- **45**: Ramsey Coloring
- **46**: Graph Embedding
- **47**: Ramsey Bound Lemma

- **48**: Euler’s Formula Setup
- **49**: Planar Graph Definition
- **50**: Inductive Face Count
- **51**: Burnside’s Lemma Statement
- **52**: Orbit-Stabilizer Link
- **53**: Group Action Symmetry
- **54**: Fix Counting Argument
- **55**: Marriage Condition
- **56**: Bipartite Hall Lemma
- **57**: Injective Assignment Construction
- **58**: Fermat Base Case
- **59**: Fermat Induction Step
- **60**: Prime Definition
- **61**: Integer Exponentiation
- **62**: Euler Structure Lemma
- **63**: Prime Contradiction Construction
- **64**: Prime Product Bound
- **65**: Divisibility Argument
- **66**: CRT Structural Lemma
- **67**: QR Decomposition Lemma
- **68**: Quadratic Reciprocity Law
- **69**: Jacobi Symbol Relation
- **70**: Modular Arithmetic: Definitions
- **71**: Congruence Reflexivity
- **72**: Existence of Modular Inverses
- **73**: Least Upper Bound Property