

UNIVERSAL LOGIC LANGUAGE

Reliable AI Reasoning

(c) Leopoldo Cano Guardiola - leo@ulogicmind.ai

March 9, 2026

Abstract

Current paradigms of mathematical formalization (based on Dependent Type Theory like Lean) present a fundamental barrier for Artificial Intelligence: **grammatical dissonance**. Encoding informal mathematical discourse into a language with a radically different grammar (Lean or similar) does not preserve meaning, but rather alters it. This “Representational Rupture” crystallizes ideas rigidly, blocking the AI’s ability to detect analogies and intuitions necessary for discovery. We propose **ULOGIC**, a new paradigm based on an underlying grammar isomorphic to semi-formal mathematical language (using nested syntactic chains *HAL-Chains* “Headed-Atomic-List”). ULOGIC allows everyday semi-formal mathematical expression to be computationally verifiable without suffering the semantic distortion of translation into distant logical code, enabling a new generation of neuro-symbolic AI systems capable of exact reasoning, capable of reusing knowledge, and ultimately capable of acting as reliable AI agents for discovery in science and industry.

1 Introduction: The Crisis of Non-Isomorphic Translation

Generative Artificial Intelligence (LLMs) has demonstrated an unprecedented capacity to manipulate natural language, but it systematically fails in rigorous logical reasoning: it can repeat what it has already seen but cannot reliably generate new knowledge based on rules (mathematics, science). On the other hand, Interactive Theorem Provers (ITPs) like Lean guarantee correctness, but are opaque to heuristic intuition and invention (due to the very limitations of the underlying formal language).

Currently, AIs face a dead end in exact mathematical learning (or more broadly, in all types of “learning for rule-based systems”):

1. **They cannot learn from textbooks:** The language of books is ambiguous and, crucially, does not explicitly express the logical steps followed nor the rules applied. There is no “exact” underlying grammar in informal text upon which the AI can apply inference rules.
2. **They cannot learn from formalisms like LEAN:** Being “another language” with a radically different grammar, it possesses “another meaning”. Training an AI in Lean is teaching it to compile code, not to reason mathematically.

This work argues that encoding mathematics into a language with an alien and distant grammar **radically alters the meaning** of the propositions. This alteration prevents AI from participating in the creative process of discovery, relegating it to a mere syntactic verification of a representation detached from the original meaning.

2 The Problem of Representation and Meaning

2.1 Meaning as an Intensional Relation

In any logical-mathematical language (whether informal or formal), the meaning of an expression is not a reference to a physical object, but is strictly **intensional**: it consists of the network of relations that said expression maintains with the rest of the expressions in the system.

Consequently, if we drastically change the grammar of the system (as happens when moving from the Intuitive Set Theory of a textbook to the Inductive Type Theory of Lean), we break the original network of relations. Meaning is not preserved; it is destroyed and replaced by a new meaning, characteristic of the code, but alien to the original intuition.

2.2 The Barrier of Non-Isomorphic Encoding

Translating everyday semi-formal reasoning from mathematics and science into a radically different formal language breaks the meaning and creates an insurmountable barrier for AI. That is the **Barrier of Non-Isomorphic Encoding**.

Languages like Lean or HOL are code representations that demand a technical explicitation (implementation) that does not exist in fluid mathematical thought.

- By forcing this encoding, a **crystallized rigidity** is introduced.
- Concepts that in informal language are analogous and allow transferring intuitions from one field to another, in Lean become disparate and incompatible type structures.
- This encoding acts as a “post-mortem” representation: it captures the logical corpse of a proof, but eliminates the heuristic pathways that allowed its discovery.

2.3 Expressive and Metalinguistic Incapacity

Beyond the change in meaning, current formal languages (FOL, HOL, LEAN) suffer from a profound **expressive incapacity**. Ordinary mathematics is inherently rich in structures that these systems cannot natively capture, and here are two examples:

- **Meta-expressions and Auto-Metalinguistic capacity:** Common expressions like $\forall x_1, x_2 \dots x_n P(x_1, \dots x_n)$ are simultaneously expressions of the system and “meta-expressions” that refer to other expressions (that expression refers to all cases where n varies in length). The ability to talk about one’s own expressions is done “inside the language” acting as a metalanguage of itself (current logic dictates that a language cannot be its own metalanguage, although this is exactly what happens -partially- in any math book from the very first page).
- **Algorithms:** Everyday mathematical discourse has the ability to define algorithms and make proofs about their execution and results. Algorithms are also mathematical expressions! But they have been left out of traditional logic (SOL, FOL, etc.).

We need a language with unified capabilities: **logical, algorithmic, and auto-metalinguistic**. Current systems are far from having this capacity, limiting AI (if forced to use those formal systems) to an impoverished and rigid subset of real reasoning.

2.4 The Blockage of Creativity in AI

An AI trained on these rigid representations (HOL, LEAN, Coq) loses the capacity for **discovery**. Mathematical discovery is based on seeing structural patterns and flexible analogies between seemingly distinct concepts. Lean’s grammar hides these analogies under layers of technical implementation.

3 ULOGIC: A New Foundational Paradigm

To solve this problem we need a grammar that acts as a faithful mirror, not as a distorting translator. We propose **ULOGIC** (Universal Logic Language).

3.1 Grammatical Isomorphism: The Rigorous Mirror

ULOGIC proposes an underlying grammar that runs parallel to the semi-formal language of mathematics and computation. The goal is to achieve a **structural isomorphism**:

$$Expression_{(semi\ formal)} \iff Expression_{(Ulogic\ Formal)}$$

In ULOGIC, the everyday mathematical semi-formal expression (that of common books) allows us to directly find the underlying UlogicFormal formal expression. There is no translation to alien code. By maintaining a grammatical parallelism, we preserve the intensional relations and, therefore, preserve the original meaning.

3.2 HAL Chains Architecture

Technically, ULOGIC eliminates the need for complex type systems through **HAL Chains** (Head + Atoms List):

- **Pure Syntax:** Meaning resides exclusively in string rewriting rules, without depending on external model semantics.
- **HAL-Chains suitable for AI:** An atom is any letter or word that you do not divide. A sequence of atoms with a Header (a special atom that acts as a marker) is a HAL-Chain (Header-Atomic-List). That is all there is, nested chains. Furthermore, there is a 1-1 correspondence between atom and token. Atoms mean one thing or another depending on how they appear in a chain and in their relationship with other chains. This contextual meaning is what LLMs capture efficiently.

3.3 Rehabilitation of Intuitive Theory and Definitions

ULOGIC allows reconstructing the Intuitive Set Theory avoiding classical contradictions through strict control of definitions.

We argue that set-theoretic contradictions (like Russell’s Paradox) do not originate in the idea of a set, but in the **rules for making definitions**. ULOGIC provides a profound revision and understanding of fundamental concepts (mistaken for 120 years):

- **Definitions as Axioms:** A definition is a line that did not exist before and is introduced into the system; it operates functionally as an axiom.
- **Non-Eliminability:** Definitions are not mere eliminable abbreviations. The eliminability of a defined term $A(x, y)$ is only possible if the atom A is always used in its explicit form. However, in mathematics it is common to use the defined atom as an argument, for example

$P(A)$ (e.g., “the set of continuous functions is closed”). Here A cannot be eliminated simply by expanding its definition.

- **Syntactic Solution:** With adequate rules for the introduction of these definitions, contradictions are avoided at a purely syntactic level, allowing the fluid use of sets without the artificial complexity of ZFC or Types. In other words: ULOGIC offers the solution that the logicians and mathematicians of the early 20th century (Russell, Hilbert, and Zermelo...) sought, but did not find.

4 Impact on Reasoning Artificial Intelligence

The adoption of ULOGIC implies a paradigm shift:

4.1 Direct Vision and Intuition Learning

Because there is no distant encoding barrier, the AI can learn directly from mathematical texts. It “sees” the real underlying logic, not an obfuscated implementation. This allows training the AI in the real discovery process, not just in the final verification.

4.2 Discovery by Analogy

By preserving intensional meaning and the flexibility of informal grammar, ULOGIC allows the AI to detect deep analogies between mathematical branches. The AI can propose new theorems by manipulating expressions at their natural level of abstraction, while the deterministic Kernel of ULOGIC (traditional software to verify TekDoc documents written in ULOGIC) verifies the validity of the steps in the background.

5 Conclusion: Towards Rigor without Alienation

True *mathematical rigor* does not consist of constraining thought within an incomprehensible **alien encoding** that alters the meaning of what we want to prove. Rigor is not a change of language, it is a structural anchor. It consists of the ability to apply strict logical rules over a clear underlying grammar.

Everyday informal and semi-formal language, far from being the enemy of precision, is an unparalleled tool for comprehension, capable of concisely encapsulating highly complex ideas. While the **underlying “UlogicFormal” grammatical structures** are necessarily more extensive to eliminate ambiguity, these remain **invisible but accessible**, acting as the firm foundations of a building.

ULOGIC demonstrates that **rigor is compatible with expressive freedom**. Rigor is possible even if we write loosely, *as long as the underlying grammar can be made explicit*. The UlogicFormal expressions and the rules for doing proofs and computation in ULOGIC are the absolutely reliable and automatically machine-verifiable rock. **Universally verifiable and automatically verified reasoning:** The end of the path initiated by Euclid 2350 years ago, dreamed of by Leibniz, and attempted by 20th-century logicians (Frege, Russell, Hilbert, Ackermann, Gentzen...). In AI, only when formalization ceases to be a destructive translation and becomes a structural mirror will we be able to teach machines not only to verify our past, but to imagine our mathematical and scientific future.