UNIVERSAL LOGIC LANGUAGE
Razonamiento IA Confiable

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

9 de marzo de 2026

Resumen

Los paradigmas actuales de formalización matemática (basados en Teoría de Tipos Dependientes como Lean) presentan una barrera fundamental para la Inteligencia Artificial: la **disonancia gramatical**. Codificar el discurso matemático informal en un lenguaje con una gramática radicalmente diferente (Lean o similar) no preserva el significado, sino que lo altera. Esta "Ruptura Representacional" cristaliza las ideas de forma rígida, bloqueando la capacidad de la IA para detectar analogías e intuiciones necesarias para el descubrimiento. Proponemos **ULOGIC**, un nuevo paradigma basado en una gramática subyacente isomorfa al lenguaje matemático semiformal (usando cadenas sintácticas *HAL-Chains* "Headed-Atomic-List" anidadas). ULOGIC permite que la expresión semiformal matemática cotidiana sea computacionalmente verificable sin sufrir la distorsión semántica de la traducción a código lógico distante, habilitando una nueva generación de sistemas de IA neuro-simbólicos capaces de razonar de forma exacta, capaces de reutilizar conocimiento, y capaces en definitiva de actuar como agentes de IA confiables para el descubrimiento en ciencia y en la industria.

---

1. Introducción: La Crisis de la Traducción No-Isomorfa

La Inteligencia Artificial Generativa (LLMs) ha demostrado una capacidad sin precedentes para manipular el lenguaje natural, pero fracasa sistemáticamente en el razonamiento lógico riguroso: puede repetir lo que ya ha visto pero no puede generar de modo confiable conocimiento nuevo basado en reglas (matemáticas, ciencia). Por otro lado, los Asistentes de Demostración (ITPs) como Lean garantizan la corrección, pero son opacos para la intuición heurística e invención (por las propias limitaciones del lenguaje formal subyacente).

Actualmente, las IAs se enfrentan a un callejón sin salida en el aprendizaje matemático exacto (o en general de todo tipo de "aprendizaje de sistemas basados en reglas"):

1. **No pueden aprender de los libros de texto:** El lenguaje de los libros es ambiguo y, crucialmente, no expresa explícitamente los pasos lógicos seguidos ni las reglas aplicadas. No existe una gramática subyacente "exacta" en el texto informal sobre la cual la IA pueda aplicar reglas de inferencia.
2. **No pueden aprender de formalismos como LEAN:** Al ser "otro lenguaje" con una gramática radicalmente distinta, posee "otro significado". Entrenar a una IA en Lean es enseñarle a compilar código, no a razonar matemáticamente.

Este trabajo sostiene que codificar matemáticas en un lenguaje con una gramática ajena y distante **altera radicalmente el significado** de las proposiciones. Esta alteración impide que

la IA participe en el proceso creativo de descubrimiento, relegándola a una mera verificación sintáctica de una representación alejada del significado original.

2. El Problema de la Representación y el Significado

2.1. El Significado como Relación Intensional

En cualquier lenguaje lógico-matemático (sea informal o formal), el significado de una expresión no es una referencia a un objeto físico, sino que es estrictamente **intensional**: consiste en la red de relaciones que dicha expresión mantiene con el resto de las expresiones del sistema.

Por consiguiente, si cambiamos drásticamente la gramática del sistema (como ocurre al pasar de la Teoría de Conjuntos Intuitiva de un libro de texto a la Teoría de Tipos Inductivos de Lean), rompemos la red de relaciones original. El significado no se preserva; se destruye y se sustituye por otro significado nuevo, propio del código, pero ajeno a la intuición original.

2.2. La Barrera de la Codificación No-Isomorfa

Traducir los razonamientos semiformales cotidianos de las matemáticas y la ciencia a un lenguaje formal radicalmente diferente rompe el significado y crear una barrera insuperable para la IA. Esa es la **Barrera de Codificación No-Isomorfa**. Lenguajes como Lean o HOL son representaciones en código que exigen una explicitación técnica (implementación) que no existe en el pensamiento matemático fluido.

- Al forzar esta codificación, se introduce una **rigidez cristalizada**.
- Conceptos que en el lenguaje informal son análogos y permiten transferir intuiciones de un campo a otro, en Lean se convierten en estructuras de tipos dispares e incompatibles.
- Esta codificación actúa como una representación "post-mortem": captura el cadáver lógico de una demostración, pero elimina las vías heurísticas que permitieron su descubrimiento.

2.3. Incapacidad Expresiva y Metalingüística

Más allá del cambio de significado, los lenguajes formales actuales (FOL, HOL, LEAN) sufren de una profunda **incapacidad expresiva**. Las matemáticas ordinarias son inherentemente ricas en estructuras que estos sistemas no pueden capturar nativamente, y aquí van dos ejemplos:

- **Meta-expresiones y capacidad Auto-Metalingüística:** Expresiones comunes como $\forall x_1, x_2 \dots x_n P(x_1, \dots x_n)$ son a la vez expresiones del sistema y "meta-expresiones" que se refieren a otras expresiones (esa expresión se refiere a todas los casos donde n varía en longitud). La capacidad de hablar sobre las propias expresiones se hace "dentro del lenguaje" actuando como metalenguaje de sí mismo (la lógica actual dice que un lenguaje no puede ser su propio meta-lenguaje, aunque es lo que ocurre -parcialmente- en cualquier libro de matemáticas desde la primera página)
- **Algoritmos:** El discurso matemático cotidiano tiene la capacidad de definir algoritmos y hacer demostraciones sobre su ejecución y resultados. ¡Los algoritmos también son expresiones matemáticas! pero han quedado fuera de la lógica tradicional (SOL, FOL, etc).

Necesitamos un lenguaje con capacidades unificadas: **lógicas, algorítmicas y auto-metalingüísticas**. Los sistemas actuales están lejos de tener esa capacidad, limitando a la IA (si se le obliga a usar esos sistemas formales) a una subconjunto empobrecido y rígido del razonamiento real.

2.4. El Bloqueo de la Creatividad en la IA

Una IA entrenada sobre estas representaciones rígidas (HOL, LEAN, Coq) pierde la capacidad de **descubrimiento**. El descubrimiento matemático se basa en ver patrones estructurales y analogías flexibles entre conceptos aparentemente distintos. La gramática de Lean oculta estas analogías bajo capas de implementación técnica.

3. ULOGIC: Un Nuevo Paradigma Fundacional

Para resolver este problema necesitamos una gramática que actúe como un espejo fiel, no como un traductor distorsionante. Proponemos **ULOGIC** (Universal Logic Language)

3.1. Isomorfismo Gramatical: El Espejo Riguroso

ULOGIC propone una gramática subyacente que corre paralela al lenguaje semiformal de las matemáticas y la computación. El objetivo es lograr un **isomorfismo estructural**:

$$Expression_{(semiformal)} \iff Expression_{(UlogicFormal)}$$

En ULOGIC, la expresión semiformal cotidiana matemática (la de los libros comunes) permite hallar de forma directa la expresión formal subyacente UlogicFormal. No hay traducción a un código ajeno. Al mantener un paralelismo gramatical, preservamos las relaciones intensionales y, por tanto preservamos el significado original.

3.2. Arquitectura de Cadenas HAL

Técnicamente, ULOGIC elimina la necesidad de sistemas de tipos complejos mediante **Cadenas HAL** (Head + Atoms List):

- **Sintaxis Pura:** El significado reside exclusivamente en las reglas de reescritura de cadenas, sin depender de semánticas de modelos externas.
- **HAL-Chains adecuados para la IA:** Un átomo es cualquier letra o palabra que no divides. Una secuencia de átomos con un Header (un átomo especial que actúa de marcador) es una HAL-Chain (Header-Atomic-List). Eso es todo lo que hay, cadenas anidadas. Además hay una correspondencia 1-1 entre átomo y token. Los átomos significan una cosa u otra dependiendo de cómo aparecen en una cadenas y en su relación con otras cadenas. Este significado contextual es el que los LLMs captan eficientemente.

3.3. Rehabilitación de la Teoría Intuitiva y Definiciones

ULOGIC permite reconstruir la Teoría Intuitiva de Conjuntos evitando las contradicciones clásicas mediante un control estricto de las definiciones.

Sostenemos que las contradicciones conjuntistas (como la Paradoja de Russell) no se originan en la idea de conjunto, sino en las **reglas para hacer definiciones**. ULOGIC proporciona una profunda revisión y comprensión de conceptos fundamentales (equivocados desde hace 120 años):

- **Definiciones como Axiomas:** Una definición es una línea que antes no existía y se introduce al sistema; funciona operativamente como un axioma.
- **No Eliminabilidad:** Las definiciones no son meras abreviaturas eliminables. La eliminabilidad de un término definido $A(x, y)$ solo es posible si el átomo A se usa siempre en

su forma explicita. Sin embargo, en matemáticas es habitual usar el átomo definido como argumento, por ejemplo $P(A)$ (e.g., "el conjunto de funciones continuas es cerrado"). Aquí A no se puede eliminar simplemente expandiendo su definición.

- **Solución Sintáctica:** Con reglas adecuadas para la introducción de estas definiciones, las contradicciones se evitan a nivel puramente sintáctico, permitiendo el uso fluido de conjuntos sin la complejidad artificial de ZFC o Tipos. En otras palabras: ULOGIC ofrece la solución que los lógicos y matemáticos de principios del siglo XX (Russell, Hilbert y Zermelo...) buscaron, pero no encontraron.

4. Impacto en la Inteligencia Artificial Razonadora

La adopción de ULOGIC implica un cambio de paradigma:

4.1. Visión Directa y Aprendizaje de la Intuición

Al no haber una barrera de codificación distante, la IA puede aprender directamente de los textos matemáticos. "Ve" la lógica real subyacente, no una implementación ofuscada. Esto permite entrenar a la IA en el proceso de descubrimiento real, no solo en la verificación final.

4.2. Descubrimiento por Analogía

Al preservar el significado intensional y la flexibilidad de la gramática informal, ULOGIC permite que la IA detecte analogías profundas entre ramas matemáticas. La IA puede proponer nuevos teoremas manipulando las expresiones en su nivel natural de abstracción, mientras el Kernel determinista de ULOGIC (software tradicional para verificar documentos TekDoc escritos en ULOGIC) verifica la validez de los pasos en segundo plano.

5. Conclusión: Hacia un Rigor sin Alienación

El verdadero *rigor matemático* no consiste en constreñir el pensamiento dentro de una **codificación alienígena** e incomprensible que altera el significado de lo que queremos demostrar. El rigor no es un cambio de lenguaje, es un anclaje estructural. Consiste en la capacidad de aplicar reglas lógicas estrictas sobre una gramática subyacente clara.

El lenguaje informal y semiformal habitual, lejos de ser el enemigo de la precisión, es una herramienta de comprensión inigualable, capaz de encapsular ideas de alta complejidad de forma concisa. Si bien las **estructuras gramaticales subyacentes "UlogicFormal"** son necesariamente más extensas para eliminar la ambigüedad, estas permanecen **invisibles pero accesibles**, actuando como los cimientos firmes de un edificio.

ULOGIC demuestra que **la rigurosidad es compatible con la libertad expresiva**. El rigor es posible aunque escribamos de forma laxa *siempre y cuando la gramática subyacente sea explicitable*. Las expresiones UlogicFormal y las reglas para hacer demostraciones y computación de ULOGIC son la roca absolutamente confiable y automáticamente verificable mediante máquinas. **Razonamiento universal verificable y verificado de forma automática**: El fin del camino que inició Euclides hace 2350 años, con el que soñó Leibniz, e intentaron construir los lógicos del siglo XX (Frege, Russell, Hilbert, Ackerman, Gentzen...). En IA, sólo cuando la formalización deje de ser una traducción destructiva para convertirse en un espejo estructural, podremos enseñar a las máquinas no solo a verificar nuestro pasado, sino a imaginar nuestro futuro matemático y científico.