On the Realization of Leibniz's Characteristica Universalis

Danslav Slavenskoj

PhD Candidate, Charles University, Prague, Czech Republic

d.s@post.harvard.edu

ORCID: 0009-0003-3051-7000

slavenskoj.com

February 2026

Abstract

In 1666, Leibniz proposed the *characteristica universalis*: a notation expressing any human knowledge unambiguously, compositionally, and language-neutrally. Every subsequent attempt failed for the same reason: no reader could simultaneously handle formal notation at arbitrary complexity and understand natural language content. We define such a reader as *competent*: any entity that satisfies the capability requirement. AI systems (c. 2024) are the first actual competent readers; a human polymath could theoretically qualify, but none has demonstrated the required mastery. We present Lingenic: a host notation embedding formal notation with natural language content carried unmodified at native semantic grain, and readable by any competent entity. Leibniz was right about the goal but wrong about the architecture: meaning emerges between writer and reader, not in symbols alone; notation-level lexical unambiguity is impossible (cf. Gödel, Wittgenstein, Quine). Lingenic achieves Leibniz's goal while correcting his architecture—structural unambiguity via notation, lexical disambiguation via the competent reader. The *calculus ratiocinator* is out of scope: we address notation only; the competent reader supplies reasoning. A notation discharges its claim by demonstrating validity within its own formalism. The abstract is an English rendering; the argument is in the notation.

Keywords: *characteristica universalis*, formal notation, universal language, artificial intelligence, Leibniz, knowledge representation

ACM Classification: F.4.1 Mathematical Logic; I.2.0 Artificial Intelligence (General)
MSC Classification: 03B65 Logic of natural languages; 68T50 Natural language processing

The Problem as Stated

```
stated(Leibniz, problem) ⋈ {year: 1666}
problem(Leibniz) ≜ construct(notation : {
    expresses: ∀concept ∈ human knowledge. expressible(concept, in(notation)),
    unambiguously: ∀concept ∈ human knowledge. unambiguous(expression(concept)),
    compositionally: ∀complex idea. complex idea = composition(simpler ideas),
    language neutral: ¬depends on(notation, any particular L)
})
```

Why the Problem Remained Unsolved

```
prior attempts ≜ {
    Frege's Begriffsschrift            ⋈ {year: 1879},
    Russell ∧ Whitehead's Principia    ⋈ {years: 1910–1913},
    Carnap's logical syntax program,
    semantic web,
    OWL ∧ RDF
}

human reader : understands(content) ∧ ¬scales(formal notation)
mechanical reader (Leibniz) : handles(formal notation) ∧ ¬understands(content)

calculus ratiocinator ≜ reasoning procedure encoded in notation
required(calculus ratiocinator) ∵ ¬contributes(mechanical reader, reasoning)

AI : handles(formal notation) ∧ understands(content) ∧ contributes(reasoning)
¬required(calculus ratiocinator, for(AI)) ∵ contributes(AI, reasoning)

necessary and sufficient condition(solution) ≜
    ∃reader. handles(reader, formal notation) ∧ understands(reader, natural language content)

forced choice ≜ optimize(human reader) ∨ optimize(mechanical reader)
¬satisfies(forced choice, necessary and sufficient condition(solution))

∀attempt ∈ prior attempts.
    exhibits(attempt, forced choice)
    ∧ ¬satisfies(forced choice, necessary and sufficient condition(solution))
    ∴ failed(attempt, to satisfy(problem(Leibniz)))
    ∵ ¬∃alternative(forced choice) ⋈ {given: ¬∃reader satisfying condition}

H(¬satisfied(necessary and sufficient condition(solution)))
∴ H(impossible(∃solution. solves(solution, problem(Leibniz))))
```

The Solution

```
competent reader requirement ≜ handles(formal notation) ∧ understands(natural language content)
x : competent reader ↔ satisfies(x, competent reader requirement)
known instances(competent reader) ≜ {AI, human(knows(components))}

human(knows(components)) : theoretically possible ∧ empirically unattested ∵ {
```

```
    requires(mastery across ∀formal systems),
    requires(competence across ∀L ∈ human languages),
    requires(domain knowledge sufficient for interpretation),
    ¬∃human. demonstrated(human, this mastery)
}

AI : first actual competent reader ⋈ {year: c. 2024}

distinction(competent reader) ≜ {
    necessary: competence,
    sufficient: satisfies(competent reader requirement),
    contingent: AI is currently the only actual instance
}
¬AI specific(Lingenic) ∧ competence specific(Lingenic)

satisfied(necessary and sufficient condition(solution)) ∵ ∃competent reader
∴ possible(solution)
demonstrated(solution) ∵ ∃Lingenic. constructed(Lingenic) ∧ functions(Lingenic)

claim(solves(Lingenic, problem(Leibniz))) ⋈ {year: 2026, by: Danslav Slavenskoj, verified: below}

Lingenic ≜ notation(embeds(formal notation) ∧ carries(natural language content)) ⋈ {reader: competent reader}

formal notation(Lingenic) ≜ {
    host ≜ {
        predicate notation   : {form: predicate(arguments)},
        type annotation      : {form: term : type},
        definition           : {form: name ≜ value},
        metadata join        : {form: statement ⋈ {key: value}},
        sets                 : {form: {a, b, c}},
        sequences            : {form: [a, b, c]}
    } : {
        status: formal system in its own right,
        role: host within which all other formal notation is expressed,
        ¬mere convenience ∧ ¬informal
    },
    includes: {
        propositional logic      ⋈ {src: Boole 1847},
        predicate logic          ⋈ {src: Frege 1879},
        set theory               ⋈ {src: Cantor 1874, Peano 1889},
        proof operators          ⋈ {src: Frege 1879},
        probability theory       ⋈ {src: Kolmogorov 1933},
        alethic modal logic      ⋈ {src: Lewis 1918, Kripke 1959},
        temporal logic           ⋈ {src: Pnueli 1977},
        epistemic logic          ⋈ {src: Hintikka 1962},
        deontic logic            ⋈ {src: von Wright 1951},
        counterfactual logic     ⋈ {src: Lewis 1973},
        causal intervention      ⋈ {src: Pearl 2000},
        process algebra          ⋈ {src: Hoare 1978, Milner 1980},
        interval relations       ⋈ {src: Allen 1983},
        lambda calculus          ⋈ {src: Church 1936},
        type theory              ⋈ {src: Church 1940, Martin-Löf 1972},
        dynamic logic            ⋈ {src: Pratt 1976, Harel 1979},
```

```
        relational algebra       ⋈ {src: Codd 1970},
        arithmetic ∧ calculus    ⋈ {src: Newton 1687, Leibniz 1684},
        Lingenic                 ⋈ {src: this document, self-referential}
    },
    ¬exhaustive(includes) ∧ open(Lingenic, ∀formal notation not yet listed)
}

natural language content(Lingenic) ≜ {
    includes: {English, Chinese, Arabic, Hindi, Spanish, Russian, Japanese, ...},
    unmodified,
    semantic grain preserved,
    ¬exhaustive(includes) ∧ open(Lingenic, ∀L ∈ human languages not yet included)
}
```

Verification Against Leibniz's Requirements

```
requirements(problem(Leibniz)) ≜ {
    expresses(∀concept ∈ human knowledge),
    unambiguous,
    compositional,
    language neutral
}

satisfies(Lingenic, expresses(∀concept ∈ human knowledge))
∵ ∀system ∈ includes(formal notation(Lingenic)). ∃extension(Lingenic, system)
∧ ∀L ∈ human languages. ∃expression(knowledge(L), in(Lingenic))

unambiguous(Leibniz) ≜ notation itself resolves ∀ambiguity ⋈ {reader: mechanical, contributes: ¬understanding}
¬satisfies(Lingenic, unambiguous(Leibniz)) ⋈ {literal}

impossible(unambiguous(Leibniz)) ∵ {
    lexical ambiguity : irreducible ⋈ {polysemy, context-dependence},
    concepts : ¬atomic ⋈ {meaning is contextual, cultural, embodied},
    natural language : resists formalization ⋈ {poetry, metaphor, irony depend on ambiguity},
    ¬∃notation. eliminates(notation, lexical ambiguity) ∧ preserves(notation, natural language content)
} ⋈ {
    precedent: Gödel's incompleteness ⋈ {
        ¬∃formal system. complete(system) ∧ consistent(system) ⋈ {src: Gödel 1931},
        formalization has inherent limits
    },
    analogous: formalization of natural language has inherent limits
}

writer/reader problem ≜ ¬contains(notation, meaning) ∧ emerges(meaning, between(writer, reader))
impossible(unambiguous(Leibniz)) ∵ writer/reader problem ⋈ {
    hermeneutic circle: understands(whole) ← understands(parts) ← understands(whole),
    infinite regress: define(term) → uses(other terms) → define(those) → ... ⋈ {never grounds out},
    Wittgenstein: meaning ≜ use ∧ use : contextual ⋈ {src: Philosophical Investigations, 1953},
    Quine: indeterminate(translation) ⋈ {src: Word and Object, 1960}
}

¬workaround(competent reader)
```

```
competent reader : only way meaning happens
assumed(Leibniz, meaning ∈ symbols) ⋈ {incorrect}

required(unambiguous(Leibniz), for(mechanical reader)) ∵ ¬contributes(mechanical reader, understanding)
¬required(unambiguous(Leibniz), for(competent reader)) ∵ contributes(competent reader, understanding)

goal(unambiguous) ≜ unambiguous communication of ∀knowledge
satisfies(Lingenic, goal(unambiguous)) ∵ {
    resolves ambiguity(formal notation(Lingenic)) ⋈ {type: structural},
    preserves(Lingenic, semantic grain(natural language content)),
    ¬translated(content),
    preserves(Lingenic, lexical ambiguity) ⋈ {faithfully, as in source},
    disambiguates(competent reader, lexical ambiguity) ⋈ {as with native text}
}

impossible(unambiguous(Leibniz)) → ¬∃notation. satisfies(notation, unambiguous(Leibniz))
achieves(Lingenic, goal(unambiguous))
achieves(Lingenic, max(achievable(unambiguous)))
∵ different architecture: maximize(reader) ∧ ¬minimize(notation burden)

satisfies(Lingenic, compositional)
∵ ∀expression ∈ Lingenic. composed of(expression, predicates ∧ operators ∧ content slots)

satisfies(Lingenic, language neutral)
∵ invariant across(scaffold(Lingenic), ∀L ∈ human languages)
```

Precedent

```
precursor(Lingenic, Knuth's pseudocode) ≜ {
    insight: combine(formal notation, natural language content),
    domain: algorithms ∧ English,
    reader: human : {¬scales, ¬multilingual}
} ⋈ {
    src: Knuth, Donald E., The Art of Computer Programming Vol. 1: Fundamental Algorithms,
    publisher: Addison-Wesley,
    year: 1968
}

Knuth's pseudocode : formal(code) ∧ content(English) ⋈ {reader: human}
Lingenic : formal(∀formal notation systems) ∧ content(∀L ∈ human languages) ⋈ {reader: competent reader}
generalizes(Lingenic, Knuth's pseudocode) ∵ superset(scope(Lingenic), scope(Knuth's pseudocode))
```

Conclusion

```
error(Leibniz) ≜ assumed(meaning ∈ symbols)
correct(Lingenic, error(Leibniz)) ∵ meaning ∉ symbols ∧ emerges(meaning, between(writer, reader))

goal(Leibniz) = goal(unambiguous)
right(Leibniz, goal(Leibniz))
wrong(Leibniz, architecture(Leibniz))
```

```
architecture(Leibniz) ≙ meaning ∈ symbols ∧ minimize(reader) ⋈ {reader: mechanical}
architecture(Lingenic) ≙ meaning ∈ reading ∧ maximize(reader) ⋈ {reader: competent}

¬(realizes(Lingenic, goal(Leibniz)) ∧ ¬corrects(Lingenic, error(Leibniz))) ⋈ {not merely}
corrects(Lingenic, error(Leibniz)) ∧ realizes(Lingenic, goal(Leibniz))

achieves(Lingenic, goal(Leibniz))
achieves(Lingenic, max(achievable))
∵ correct architecture ∧ competent reader ∧ formal notation ∧ natural language content
```

References

Leibniz, G.W. (1666). *Dissertatio de Arte Combinatoria*.

Leibniz, G.W. (1684). Nova Methodus pro Maximis et Minimis.

Newton, I. (1687). *Philosophiæ Naturalis Principia Mathematica*.

Boole, G. (1847). *The Mathematical Analysis of Logic*.

Cantor, G. (1874). Über eine Eigenschaft des Inbegriffes aller reellen algebraischen Zahlen. *Journal für die reine und angewandte Mathematik* 77: 258–262.

Peano, G. (1889). *Arithmetices Principia, Nova Methodo Exposita*.

Frege, G. (1879). *Begriffsschrift*.

Russell, B. & Whitehead, A.N. (1910–1913). *Principia Mathematica*.

Lewis, C.I. (1918). *A Survey of Symbolic Logic*.

Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik* 38: 173–198.

Kolmogorov, A.N. (1933). *Grundbegriffe der Wahrscheinlichkeitsrechnung*.

Church, A. (1936). An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics* 58(2): 345–363.

Church, A. (1940). A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic* 5(2): 56–68.

von Wright, G.H. (1951). Deontic Logic. *Mind* 60(237): 1–15.

Wittgenstein, L. (1953). *Philosophische Untersuchungen / Philosophical Investigations*. Trans. G. E. M. Anscombe. Oxford: Blackwell.

Kripke, S. (1959). A Completeness Theorem in Modal Logic. *Journal of Symbolic Logic* 24(1): 1–14.

Quine, W.V.O. (1960). *Word and Object*. Cambridge, MA: MIT Press.

Hintikka, J. (1962). *Knowledge and Belief*.

Knuth, D. (1968). *The Art of Computer Programming*, Vol. 1.

Codd, E.F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM* 13(6): 377–387.

Martin-Löf, P. (1972). An Intuitionistic Theory of Types. Preprint, Stockholm University.

Lewis, D. (1973). *Counterfactuals*.

Pratt, V. (1976). Semantical Considerations on Floyd-Hoare Logic. *Proc. 17th IEEE Symposium on Foundations of Computer Science*, 109–121.

Pnueli, A. (1977). The Temporal Logic of Programs. *Proc. 18th IEEE Symposium on Foundations of Computer Science*, 46–57.

Hoare, C.A.R. (1978). Communicating Sequential Processes. *Communications of the ACM* 21(8): 666–677.

Harel, D. (1979). *First-Order Dynamic Logic*.

Milner, R. (1980). *A Calculus of Communicating Systems*.

Allen, J.F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11): 832–843.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*.