

Semantic Representation and Computation for Pure Mathematics in the Wolfram Language

Ian Ford 

Wolfram Research, Inc., Champaign, Illinois, USA
`ianf@wolfram.com`

Abstract. The Wolfram Institute Pure Mathematics Project is an ongoing effort to extend Wolfram Language with symbolic representations of concepts and operations across pure mathematics. It investigates how a general computational language can express mathematical ideas in forms that remain recognizable to mathematicians while supporting computation. We present the current implementation, a Wolfram Language packet integrating executable definitions with usage messages, reference pages, guides, tutorials, notebook formatting, and tests. Its public interface spans 11 broad mathematical categories. Examples from topology, algebra, and proof theory illustrate a common design in which symbolic expressions support computation and domain-specific algorithms while preserving mathematical structure for symbolic transformation and reasoning at multiple levels. During research and development we also used large language models with the same symbolic source and documentation, suggesting that the representation can serve as a common interface for human and machine use. We outline plans to broaden the language and connect it with mathematical knowledgebases and proof assistants.

Keywords: Semantic representation · Mathematical software · Mathematical knowledge management · AI for mathematics

Wolfram Language represents programs, data, and mathematical structures as symbolic expressions. Its environment combines mathematical computation, automated theorem proving [9], computable knowledge, and notebooks. The project extends this vocabulary with concepts and operations from pure mathematics. Here, *semantic representation* means encoding mathematical objects, operations, properties, and relations as structured symbolic expressions that can be composed, transformed, queried, computed with, and displayed in conventional notation. Each addition is developed as part of the language rather than as an isolated routine: its behavior, usage messages, reference documentation, and notebook formatting are designed together.

The broader vision of computational knowledge for pure mathematics in Wolfram Language was set out by Wolfram [7]. Distinct Wolfram projects have pursued aspects of that vision in continued fractions [5], function spaces [6], and general topology [2]. Research on reusable proof-assistant structures [3] and unified mathematical data infrastructure [1] offers complementary approaches.

On 27 August 2026 the public packet contained 2,171 public symbols, 122 guides, 99 tutorials, and 6,689 test declarations across 11 broad categories [8]. Examples include finite topological spaces and algebras, proof trees, and sequents. Users can apply domain-specific algorithms to whole expressions while using general Wolfram Language operations to inspect, transform, and compose their constituent parts.

During research and development we used large language models with the same symbolic source and documentation, suggesting that semantic representations can provide a shared interface for mathematicians and computational tools. Current work is broadening and refining the language. Planned connections to the Wolfram Knowledgebase and MathWorld would provide access to curated knowledge; future work will investigate alignments with Lean [4].

Acknowledgments. The author thanks Stephen Wolfram for project direction and Bradley Janes, Jeremy Stratton-Smith, Nikolay Murzin, and Peter Barendse for contributions.

Disclosure of Interests. The author is employed by Wolfram Research, Inc.

References

1. Berčič, K., Kohlhase, M., Rabe, F.: Towards a unified mathematical data infrastructure: Database and interface generation. In: Intelligent Computer Mathematics. Lecture Notes in Computer Science, vol. 11617, pp. 28–43. Springer (2019). https://doi.org/10.1007/978-3-030-23250-4_3
2. Ford, I.: Semantic representation of general topology in the Wolfram Language. In: Intelligent Computer Mathematics. Lecture Notes in Computer Science, vol. 10383, pp. 163–177. Springer (2017). https://doi.org/10.1007/978-3-319-62075-6_12
3. Garillot, F., Gonthier, G., Mahboubi, A., Rideau, L.: Packaging mathematical structures. In: Theorem Proving in Higher Order Logics. Lecture Notes in Computer Science, vol. 5674, pp. 327–342. Springer (2009). https://doi.org/10.1007/978-3-642-03359-9_23
4. de Moura, L., Ullrich, S.: The Lean 4 theorem prover and programming language. In: Automated Deduction—CADE 28. Lecture Notes in Computer Science, vol. 12699, pp. 625–635. Springer (2021). https://doi.org/10.1007/978-3-030-79876-5_37
5. Trott, M., Weisstein, E.W.: Computational knowledge of continued fractions. Wolfram|Alpha Blog (2013), <https://blog.wolframalpha.com/2013/05/16/computational-knowledge-of-continued-fractions/>
6. Weisstein, E.W., Ford, I.: The semantic representation of pure mathematics. Wolfram Blog (2016), <https://blog.wolfram.com/2016/12/22/the-semantic-representation-of-pure-mathematics/>
7. Wolfram, S.: Computational knowledge and the future of pure mathematics. Stephen Wolfram Writings (2014), <https://writings.stephenwolfram.com/2014/08/computational-knowledge-and-the-future-of-pure-mathematics/>
8. Wolfram Institute: Pure Mathematics Project, <https://www.wolframcloud.com/obj/wolfram institute/PureMath>, accessed 27 August 2026
9. Wolfram Research: FindEquationalProof. Wolfram Language Documentation (2018), <https://reference.wolfram.com/language/ref/FindEquationalProof>