Kleis: A Unified Framework for Knowledge Production

Jane Smith[1], Alice Chen[1]

[1]*MIT CSAIL, Cambridge, MA*

Abstract. We present Kleis, a domain-specific language for knowledge production that unifies notation, verification, and document generation. Our approach is based on dependent type theory and enables both symbolic computation and formal verification within a single framework.

Keywords: formal verification, type theory, scientific computing, domain-specific languages
PACS: 03.65.-w, 02.10.Yn
MSC: 81P05, 03G12
arXiv: 2501.12345

1. Introduction

The landscape of scientific computing is characterized by a growing gap between expressive notation and formal verification. Existing tools either prioritize expressiveness (Mathematica, Maple) or verification (Coq, Lean), but rarely both.

The well-known quadratic formula demonstrates our notation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

2. Related Work

Prior work in this area includes the Lean theorem prover, which provides a foundation for mathematical formalization, and Typst, which enables high-quality document generation.

3. Methodology

Our approach combines three key innovations: a dependent type system, a template-based rendering engine, and an axiom verification system.

> **Theorem 1.** For any well-formed Kleis program P, if P type-checks, then P terminates with a value or diverges.

4. Results

We evaluated Kleis on a benchmark of 100 mathematical proofs from various domains including analysis, algebra, and topology.

Domain	Problems	Verified	Time (avg)
Analysis	35	35	0.3s
Algebra	40	38	0.5s
Topology	25	24	0.4s

Table 1: Benchmark results across mathematical domains

5. Conclusion

We have presented Kleis, a unified framework that bridges the gap between expressive notation and formal verification. Future work will extend the system to support interactive theorem proving.

Acknowledgments. This work was supported by NSF grant XXX-YYYY.