Book Announcement: "Proof Theory and Logic Programming: Computation as Proof Search"

Dale Miller

Cambridge University Press has recently published my book, "Proof Theory and Logic Programming: Computation as Proof Search" (December 2025). A preprint of the full text is available for download [1].

While the **proof-as-program** approach—where computation is driven by proof normalization—is well-documented in the literature, this book explores an alternative formalization: the **proof-search** paradigm. Often identified with logic or relational programming, proof search is traditionally introduced through resolution and Horn clauses. This work, however, anchors the paradigm in structural proof theory and the sequent calculus, reframing computation as the systematic search for cut-free proofs. By treating computation as a goal-directed process, this book bridges the gap between high-level logical specifications and their execution.

1 Key Themes and Coverage

This book establishes the sequent calculus as a robust framework for describing proof search, with a rigorous focus on structural rules, focused proofs, and cut-elimination theorems. By tracing the transition from classical to intuitionistic and linear logic, it demonstrates how these systems expand the expressive power of logic programming. Using the lens of focusing, the text develops the logical foundations for Prolog, λProlog, and two distinct linear logic programming languages. Furthermore, the framework accommodates both first-order and higher-order quantification, with the completeness of focused proofs proven via detailed cut-elimination arguments.

2 From Theory to Application

Beyond theoretical foundations, the book offers numerous examples of applying proof theory to reason about logic programs. It features several chapter-length

case studies that apply these principles to modern computer science problems:

- **Encoding Security Protocols**: Modeling communication and cryptographic properties.
- **Formalizing Operational Semantics**: Using logic to specify and analyze the behavior of programming languages.
- **Static Analysis**: Performing collection analysis and formal reasoning for Horn clauses.

A central question addressed is: "Why should one incorporate higher-order quantification or linear logic into a logic program?" The application-oriented chapters provide several examples that leverage these rich logical specifications.

3 Audience

Whether you are a graduate student in logic and computation, a researcher in programming language design, or an educator seeking a principled textbook on logic programming, this book serves as a comprehensive resource. It assumes minimal prerequisites, making it accessible to newcomers while offering insights for seasoned experts. The text contains over 90 exercises, and many are given full or partial solutions. "Proof Theory and Logic Programming" distills forty years of research into a clear, principled journey from theory to design and application. It invites the computer science community to reconsider how logic can specify computation, and encourages the proof theory community to explore new dynamics and applications of the sequent calculus.

For more information, including the table of contents and endorsements, please visit the book's webpage at the link provided below.

References

[1] Dale Miller. *Proof Theory and Logic Programming: Computation as Proof Search*. Cambridge University Press, 2025. DOI: 10.1017/9781009561280 Preprint available at `https://www.lix.polytechnique.fr/Labo/Dale.Miller/ptlp/`.