Proof Complexity and Its Relations to SAT Solving

Albert Atserias ⌂ 🆔
Universitat Politècnica de Catalunya, Barcelona, Spain
Centre de Recerca Matemàtica, Bellaterra, Spain

─── **Abstract** ───────────────

Propositional proof complexity is the study of algorithms that recognize the set of tautologies in propositional logic. Initially conceived as an approach to address the "P versus NP" problem in computational complexity, the field has gradually expanded its focus to include new objectives. Among these is the goal of providing a theoretical foundation for comparing the effectiveness of heuristics for algorithms that exhaustively explore the solution spaces of combinatorial problems. Dually, and complementarily, the methods of proof complexity can also be used to assess how to certify that a given exploration path of such an algorithm ultimately leads to a dead end. A notable challenge faced by this methodology lies in the fact that, despite the theoretically proved modelling power of propositional logic, as established by the theory of NP-completeness, propositional logic is not always the best specification language for all application domains. Addressing this challenge involves studying the expressive power of various languages and their associated proof systems through the lens of computational complexity.
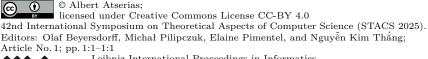
The first part of this talk will be a survey of how the emergence of these new objectives for propositional proof complexity came to be, and what the theory's methods offer in pursuing them. The second part will review the current state of the art on the computational complexity of automating the proof search problem for various proof systems for propositional logic and other languages. While it is now known and well understood that fully automating propositional Resolution as a proof system for propositional logic is NP-hard, it remains an open question whether it is possible to distinguish satisfiable formulas from unsatisfiable ones with short Resolution proofs of unsatisfiability in polynomial time. As of the time of writing, there is no consensus among experts on whether this problem should be considered computationally intractable.