Mathematisches Forschungsinstitut Oberwolfach

Report No. 34/2020

DOI: 10.4171/OWR/2020/34

Mathematical Logic: Proof Theory, Constructive Mathematics

Organized by
Samuel R. Buss, La Jolla
Rosalie Iemhoff, Utrecht
Ulrich Kohlenbach, Darmstadt
Michael Rathjen, Leeds

8 November – 14 November 2020

Abstract. Invariants of topological spaces of dimension three play a major role in many areas, in particular . . .

Introduction by the Organizers

The workshop *Invariants of topological spaces of dimension three*, organised by Max Muster (München) and Bill E. Xample (New York) was well attended with over 30 participants with broad geographic representation from all continents. This workshop was a nice blend of researchers with various backgrounds . . .

Acknowledgement: The MFO and the workshop organizers would like to thank the National Science Foundation for supporting the participation of junior researchers in the workshop by the grant DMS-1641185, “US Junior Oberwolfach Fellows”. Moreover, the MFO and the workshop organizers would like to thank the Simons Foundation for supporting NAME(S) in the “Simons Visiting Professors” program at the MFO.

Workshop: Mathematical Logic: Proof Theory, Constructive Mathematics

Table of Contents

Jan Krajíček

Proof search problem 5

Abstracts

Proof search problem

Jan Krajíček

Propositional proof complexity is linked with SAT solving by interpreting the run of a complete SAT algorithm that fails to find a satisfying assignment for φ as a *proof* that $\neg\varphi$ is a tautology. Often such an "abstract" proof system is equal to (or close to) a standard proof system as is R (resolution). Various technical results (and lower bounds, in particular) known in proof complexity for the proof system can then be interpreted as results about the original algorithm. That is, proof complexity contributes to the *analysis of SAT algorithms*.

This seems to be too narrow and proof complexity ought to attempt to precisely formalize and to answer some of the outstanding informal problems. These include:

(1) How do you compare two proof search algorithms and is there an optimal way to search for propositional proofs?

(2) Why it does not seem to be particularly helpful to search for proofs in stronger proof systems?

(3) How is it possible that real-world algorithms (SAT or automated theorem proving) perform well even for very long formulas while we have exponential lower bounds for the associated proof systems?

Basic notions of proof complexity as are propositional proof systems and simulations and p-simulations among them, can be found in [1]. The fundamental problems are the NP vs. coNP problem, asking whether for some proof system P is the length-of-proof function

$$s_P(\tau) := \min\ (|w| \mid\ w \text{ is a } P\text{-proof of } \tau\ )$$

bounded by $|\tau|^{O(1)}$, and the optimality problem: Is there a proof system that is maximal in the quasi-ordering induced by (p-)simulation? The optimality problem relates to a number if questions in a surprisingly varied areas and there are quite a few relevant statements known cf. [1, Chpt.21]).

We define a *proof search algorithm* to be a pair (A, P), where P is a proof system and A is a deterministic algorithm such that $A(\tau)$ is a P-proof of τ, for all tautologies τ. We note two statements:

Lemma *For any fixed proof system P there is A such that (A, P) is time-optimal among all (B, P); it has at most polynomial slow-down:*

$$time_A(\tau)\ \leq\ time_B(\tau)^{O(1)}\ .$$

Let (A_P, P) denote some proof search algorithm time-optimal among all (B, P).

Theorem *Let P be any proof system containing R and having the property that for some $c \geq 1$, for every τ and every τ' obtained from τ by substituting constants for some atoms it holds $s_P(\tau') \leq s_P(\tau)^c$.*

Then P is p-optimal iff (A_P, P) is time-optimal among all proof search algorithms (B, Q).

The proof of the non-trivial if-direction uses the fact that for any Q there is a *p-time construable sequence* of tautologies

$$\langle Ref_Q \rangle_n \ , \ n \geq 1$$

such that if it is feasible to construct P-proofs of these formulas then P p-simulates Q.

Another context where *easy sequences of hard formulas* appear are length-of-proofs lower bounds: whenever we can show that Q is stronger than P we can demonstrate it on such a sequence.

I would like to have a definition of a quasi-ordering on proof search algorithms that does not declare (B, Q) stronger only because B will recognize a simple sequence of formulas that have short Q-proofs but long P-proofs. The idea is that we compare proof search algorithms only on *special test sets* T that do not contain easy to recognize sets of tautologies. Having such a notion, we put

Definition *Define that (A, P) is as good as (B, Q), denoted by $(A, P) \succeq (B, Q)$, iff for all test sets T:*

$$time_A(\tau) \ \leq \ time_B(\tau)^{O(1)} \ \ \textit{for all} \ \ \tau \in T \ .$$

In [1, Sec.21.5] I took test sets to be of the form TAUT $\setminus H$ with $H \in \mathrm{P}/poly$, allowing to disregard those easy sequences of hard formulas. But maybe one ought to disallow all such easy sets at the same time, and to declare a set easy if it is computable in sub-exp-time $2^{o(n)}$ rather than in p-time. Such "subexp-time-immune" subsets of TAUT can be constructed by a diagonalization process but there are also candidates that are more transparent, constructed from conjectured proof complexity generators: tautologies in such test sets express that a string is outside of the range of a suitable map.

An open problem is whether for some natural test sets there is (A, P) that is $\succeq$-maximal among all proof search algorithms. It would be interesting if some such $\succeq$ allowed for an unconditional affirmative answer and if the proof system P would be one of the weaker proof systems (this would offer answers to informal problems 1 and 2 mentioned above).

While we have easy sequences of hard formulas for various proof system they are in a sense rather rare (e.g. combinatorial principles or reflection principles). This can be an explanation why real life algorithms solve problems of huge size (cf. informal problem 3 above): the formulas are instances from easy to describe sets and such sets of hard formulas are rare.

Slides from the talk are available at:
`www.karlin.mff.cuni.cz/~krajicek/talk-proofsearch-mfo-11-20.pdf`

References

[1] J. Krajíček, *Proof complexity*, Encyclopedia of Mathematics and Its Applications, Vol. **170**, Cambridge University Press, (2019).

Reporter: Anton Freund