Proof search problem

Jan Krajíček

Charles University

Oberwolfach, 12.November 2020

SAT solving and proof complexity

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.

sample informal questions

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 thm proving) perform well even for very long formulas while we have exponential lower bounds for the associated proof systems?

basic notions

Cook-Reckhow's definition

A propositional proof system (abbreviated pps) is a p-time function whose range is exactly TAUT, the set of propositional tautologies:

$$P \ : \ \{0,1\}^* \rightarrow_{onto} \ \text{TAUT} \ .$$

Equivalently, P may be a p-time provability predicate admitting proofs for tautologies but not for other formulas.

Fundamental problem

Is NP closed under complementation? Equivalently, is there a pps P such that the function

$$s_P(\tau) \ := \ \min\{|w| \mid P(w) = \tau\}$$

is bounded by $|\tau|^{O(1)}$?

optimality of pps

Two pps P and Q can be compared by their proof lengths:

$$P \geq Q \;\Leftrightarrow\; s_P(\tau) \leq s_Q(\tau)^{O(1)}$$

or by the possibility to efficiently translate proofs:

$$P \geq_p Q \;\Leftrightarrow\; \exists \text{ p-time } f \text{ s.t. } \forall w, P(f(w)) = Q(w) \ .$$

(Function f is called p-simulation.)

Optimality problem

Is there a maximal pps w.r.t. $\geq$ or $\geq_p$?
(The former would be called optimal, the latter p-optimal.)

NO $\Rightarrow$ $NP \neq coNP$ or $P \neq NP$, resp.
(in fact, $E \neq NE$, ...)

known facts

The Optimality problem relates to a number of questions in surprisingly varied areas: structural complexity th. (disjoint NP sets, sparse complete sets, ...), finite model th., Gödel's thms, games on graphs, etc., and quite a few relevant results are known.

In particular, relative to a theory there is an optimal pps ($\geq$-max w.r.t. to all pps that are provably sound in the theory) and uniformity of pps may be important (there is an optimal pps among pps with advice).

Theorem (K.-Pudlak'89)

There exists a p-optimal pps iff there exists a deterministic algorithm computing χ_{TAUT}, the characteristic function of TAUT, that is time-optimal on formulas from TAUT.

proof search alg's

Definition

A proof search algorithm is a pair (A, P) where P is a pps and A is a deterministic algorithm finding P-proofs:

$$P(A(\tau)) \;=\; \tau$$

for all $\tau \in TAUT$.

Remark: There is a notion of automatizable pps which is, however, void: there are essentially no automatizable pps (except truth-tables and alike). (However, the notion gives a nice interpretation of the failure of feasible interpolation: if P fails to admit feasible interpolation then it is not automatizable.)

a couple of statements

Lemma

For any fixed pps 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 a proof search algorithm time-optimal for all (B, P).

Theorem

For any sufficiently strong (ess. containing R) pps P:
P is p-optimal iff (A_P, P) is time-optimal among all proof search algorithms (B, Q).

seq's of hard flas

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 $P \geq_p Q$.

Another context where p-time seq's 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 alg's that does not declare (B, Q) stronger only because B will recognize a p-time sequence of formulas that have short Q-proofs but long P-proofs.

comparing proof search alg's

The idea is that we compare proof search alg's only on special test sets T that do not contain easy to recognize sets of tautologies.

Definition

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

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

If tests are closed under intersection then we can quantify T existentially.

In Sec.25.1 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 p-time 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.

test sets

Possible test sets $T \subseteq TAUT$

T infinite but not containing an infinite subset computable in time $2^{o(n)}$.

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 candidate proof complexity generators: tautologies in such test sets express that a string is outside of the range of a suitable map.

Open problem

Is there (A, P) that is $\succeq$-maximal among all proof search algorithms?

concluding remarks

- It would be interesting if this (or some other) definition of $\succeq$ allowed for an unconditional affirmative answer and if the pps P could be one of the weaker pps.
  This would offer answers to informal problems 1 and 2.

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

a general reference

Chpt.21 in