What do tautologies know about their poofs?

Jan Krajíček

Charles University

FOCS'21 workshop, 8.February 2022

The Cook-Reckhow definition

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

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

Fundamental problem

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

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

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

Holly grail

Prove super-polynomial lower bounds for s_P for P as strong as possible.

But why?

If we manage to prove such lower bounds
for some but not all pps
does it have any significance at all?

Fact

No super-poly lower bounds for s_P are known for the usual text-book calculus based on modus ponens and a finite nb. of axiom schemes (a Frege system in the established terminology).

We may be lucky and prove a super-poly s_P lower bound for optimal P:

- *s_P has at most poly slowdown w.r.t. any other s_Q,*

or even for a p-optimal P:

- *proofs in any Q can be translated in p-time to P-proofs.*

Then super-poly lower bounds follow for all s_Q and NP $\neq$ coNP.

But we do not know if such a pps exists.

The optimality problem

Is there a p-optimal, or at least an optimal, pps?

This problem relates to a surprisingly varied areas: structural complexity th. (disjoint NP sets, sparse complete sets, ...), finite model th., quantitative Gödel's thms, games on graphs, ...

Even if we are not lucky and P is not optimal, a super-poly lower bound for s_P does have, in fact, at least two interesting consequences.

s_p-lower-bound consequence 1

No SAT algorithm from a class $Alg(P)$ of SAT algorithms attached to P runs in p-time.

Alg_P: alg's whose soundness have short proofs in P
[soundness relates to simulation]

Ex. $Alg(F_d)$ contains commonly considered enhancements of DPLL (even for small d)

Fact

Virtually all SAT alg's considered at present are contained in $Alg(P)$ for some P for which we have super-poly s_P lower bounds.

s_p-lower-bound consequence 2

$P \neq NP$ is consistent with a FO theory T_P associated with P: there is a model of T_P in which all p-time clocked alg's fail to solve SAT.

T_P: some base theory plus a universal statement expressing the soundness of P

Ex. T_{ER} is Cook's theory PV and it proves a significant part of complexity theory (e.g. the PCP theorem). In particular, a significant part of complexity theory holds in *any* model of T_{ER}.

Fact:

T_P cannot prove lower bounds for any pps Q stronger than P (in terms of a p-simulation).

A change of perspective:

- *do not ask about the size of proofs but how hard it is to find them.*

The Fundamental problem becomes the P vs. NP problem and the Optimality problem translates into

Proof search problem (informal)

Is there an optimal way to search for propositional proofs?

Definition

A proof search algorithm is a pair (A, P) where P is a pps and A is a deterministic algorithm that stops on all inputs and finds P-proofs for all tautologies:

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

all $\tau \in TAUT$.

In fact, this problem can be "clarified".

Lemma

For any fixed pps P there is A_P such that (A_P, P) is time-optimal among all (B, P): for all τ

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

Theorem

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

Hence the optimal proof search problem reduces to the original p-optimality problem .

A motivation for the notion coming next:

- *The quasi-ordering of proof search alg's by time does not seem quite right and it lead me to consider how to measure the hardness of searching for a proof of an individual formula: the measure should apply to an individual formula (similarly as s_P does) and not to an asymptotic behavior of an algorithm.*

Eventually this lead to an alternative quasi-ordering for which, however, the optimality has the same answer: it is just the p-optimality problem.

But the resulting notion seems to be of an independent interest.

Definition

For a pps P, the information efficiency function is defined as:

$$i_P(\tau) \ := \ \min\{Kt(\pi|\tau) \mid P(\pi) = \tau\} \ .$$

Kt: Levin's time-bounded Kolmogorov complexity:

$$Kt(w|u) \ := \ \min\{|e| + \lceil \log t \rceil \mid \{e\} \text{ computes } w \text{ from } u \text{ in time } \leq t\}$$

Observation

For P whose proofs are not shorter than the formula being proved and which allows to simulate efficiently the truth-table proof:

$$\log|\tau| \leq \log s_P(\tau) \leq i_P(\tau) \leq |\tau| \ .$$

information and time

Lemma 1

Let (A, P) be any proof search algorithm. Then for all $\tau \in TAUT$:

$$i_P(\tau) \leq Kt(A(\tau)|\tau) \leq |A| + \log(time_A(\tau)) \ .$$

In particular, $time_A(\tau) \geq \Omega(2^{i_P(\tau)})$.

Lemma 2

For every proof system P there is an algorithm B_P such that for all $\tau \in TAUT$:

$$Kt(B_P(\tau)|\tau) = i_P(\tau)$$

and

$$time_{B_P}(\tau) \leq 2^{O(i_P(\tau))} \ .$$

[In fact, $A_P \sim_{time} B_P$.]

As always

$$i_P(\tau) \ \geq \ \log s_P(\tau) \ ,$$

a super-poly s_P-lower-bound implies a super-log lower bound for i_P. Does such a lower bound for i_P:

$$i_P(\tau) \geq \omega(\log |\tau|)$$

alone imply anything interesting?

Fact

Assuming a super-logarithmic lower bound for i_P the same two consequences as before follow:

- No SAT algorithm from a class $Alg(P)$ runs in p-time.
- $P \neq NP$ is consistent with theory T_P.

Problem

Prove an *unconditional* lower bound

$$i_P(\tau) \geq \omega(\log|\tau|)$$

for some proof system P for which no super-polynomial lower bounds for s_P are known.

It is possible to formulate various weaker versions of the problem but the emphasis should always be on the qualification unconditional.

Is it easier to prove i_P lower bounds than to prove s_P lower bounds?

Various plausible hypotheses (e.g. $P \neq NP$ or *RSA is secure*) imply that for many P (except some trivial ones):

$$i_P(\tau) \;\geq\; \omega(\log s_P(\tau)) \; .$$

I.e. super-log lower bounds for i_P do not imply, in general, super-poly lower bounds for s_P(keyword: automatizability).

notation/terminology

The main parameter is $m := |\tau|$ and we call a quantity

- small or large iff it is $O(\log m)$ or $\omega(\log m)$, resp.,
- and a string (of any length) simple or complex iff its Kt-complexity is small or large, resp.

$X \subseteq$ TAUT solves the problem iff

- X is a set of formulas of unbounded size,
- $i_P(\tau) \geq \omega(\log m)$ for $\tau \in X$, $m >> 1$.

Remark

As we aim at unconditional lower bound we ought to expect that formulas from X require super-poly size as well (although we may not be able to prove that).

Necessary condition (N)

If X solves the problem then all P-proofs of $\tau \in X$ have to be complex.

Prf.:

$$i_P(\tau) \leq Kt(\pi|\tau) \leq Kt(\pi)$$

□

Sufficient condition (S)

If X satisfies (N) and all $\tau \in X$ are simple then X solves the problem.

Prf.:

$$Kt(\pi) \leq Kt(\tau) + Kt(\pi|\tau) + \log\text{-terms}$$

and so

$$Kt(\pi|\tau) \geq \omega(\log m) - O(\log m) = \omega(\log m) \ .$$

□

The heart of (S) can be reformulated so that, in principle, it applies to complex formulas as well.

Sufficient condition (S')

If X satisfies (N) and for all $\tau \in X$ and for all P-proofs π of τ:

$$It(\tau : \pi) := Kt(\pi) - Kt(\pi|\tau) \text{ is small}$$

the X solves the problem.

This quantity, defined by Kolmogorov, was by him interpreted as

information that τ conveys about π.

An informal summary

We look for $X \subseteq$ TAUT consisting of formulas that have only complex proofs but that convey little information about them.

There are two classes of candidate hard formulas supported by some theory:

- *reflection formulas*,
- *proof complexity generators*.

Reflection formulas

$$\langle Ref_Q \rangle_m$$

express that *all formulas with a Q-proof of size $\leq m$ are tautologies.*

Facts:

- uniform (and hence *simple*),
- probably too general to be useful for *unconditional* lower bounds,
- in principle, (S) can be used.

Proof complexity generators

Given a p-time function g extending n bits to $m = m(n) > n$ bits

$$g_n : \{0,1\}^n \to \{0,1\}^m$$

each $b \in \{0,1\}^m \setminus Rng(g_n)$ defines formula

$$\tau(g)_b(x,y) \ := \ g_n(x) \neq b \ .$$

Facts:

- non-uniform (possibly all complex),
- hard for all pps' for which lower bounds are known,
- (S') needs to be used in place of (S), i.e.

first we need to understand the quantity $It(\tau : \pi)$*.*

Ex.: Let $f : \{0,1\}^\ell \times \{0,1\}^k \to \{0,1\}^{k+1}$ is any p-time function.

Gadget generator

Function $Gad_f : \{0,1\}^n \to \{0,1\}^{n+1}$, where $n := \ell + k(\ell+1)$, takes as an input an n-string that it interprets as $(\ell+2)$-tuple

$$(v, u^1, \dots, u^{\ell+1})$$

where: $|v| = \ell$, $|u^i| = k$, all $i \leq \ell + 1$, and outputs

$$(w^1, \dots, w^{\ell+1})$$

where $w^i := f(v, u^i)$, all $i \leq \ell + 1$.

W.l.o.g. v is a circuit sending k bits to $k+1$ bits and f is circuit evaluation and $\ell \leq k^2$. Such *Gad* is *universal* is a good sense.

The truth-table function sends a circuit $C(x_1, \ldots, x_k)$ in k variables to its truth-table $\mathrm{tt}(C)$, the string of all 2^k values ordered lexicographically.

For w any string define its circuit-size

$$CSize(w) \;:=\; \min\{|C| \mid \mathrm{tt}(C) = w'\}$$

where w' is w extended by zeros so that the length of w' is a power of 2.

Observation

$$Kt(w) \;\leq\; CSize(w) + \log(|w|) + O(1)\ .$$

Remark:
Allender et.al., Power from Random Strings, 2006, characterize $Kt(w)$ as circuit size in a more general model of circuits (may use oracle for a set in E).

Theorem

For any pps P:

1. *either* P is not p-bounded, i.e. there are super-poly lower bounds for s_P and hence super-log lower bounds for i_P,
2. *or* there are simple formulas τ, $|\tau| = m$ and $CSize(\tau) = O(\log m)$ (and hence $Kt(\tau) \leq O(\log m)$ too), such that no P-proof π of τ has small, i.e. $O(\log m)$, circuit size.
   (In fact, $CSize(\pi) \geq m^\delta$ for a fixed constant $\delta > 0$.)

The proof modifies the proof of Thm.2.1 in
J.K., Diagonalization in proof complexity, Fundamenta Mathematicae, 182, pp.181-192, (2004).

[I do not think it can be generalized further to Kt instead of $CSize$.]

proof idea

P: any pps

S: base FO theory plus an axiom stating that anything P proves, *even implicitly*, is valid

$\underline{N}$: dyadic numeral for N, $|\underline{N}| \sim \log N$

Gödel's diagonal lemma

There is an FO formula $A(x)$ such that S proves that for all $N \geq 1$:

$$A(\underline{N}) \;\Leftrightarrow\; [A(\underline{N}) \text{ has no } S\text{-proof of size } \leq N] \;.$$

Note: $|A(\underline{N})| = O(\log N)$.

proof idea cont'd

Assuming both (1) and (2) in the thm fail we construct a $(\log N)^{O(1)}$ size S-proof of $A(\underline{N})$ and reach a contradiction as follows:

- Translate $A(\underline{N})$ into a big tautology $||A||_N$ of size $O(N)$. It is uniform a there is a $O(\log N)$ size C s.t. $tt(C) = ||A||_N$.
- Assuming (2) fails there is a $O(\log N)$ size D s.t. $tt(D)$ is a P-proof of $||A||_N$.
- The fact that D describes a P-proof of $tt(C)$ can be expressed by a $O(\log N)$ size tautology $\sigma_{C,D}$.
- Assuming (1) fails, this fla has a size $(\log N)^{O(1)}$ P-proof π.
- Using the special axiom of S we derive that $A(\underline{N})$ is true.
- Total size is $(\log N)^{O(1)} << N$: a contradiction!

references

- J.K., Information in propositional proofs and algorithmic proof search, J.Symbolic Logic, to appear,

  [available from my web page]

- J.K., *Proof Complexity*, (2019), CUP

  [for all proof background mentioned]