Proof Complexity

Jan Krajíček

Charles University

(BCTCS, Liverpool, 30.March 2021)

topic

proof complexity briefly:

- *study of lengths of propositional proofs*

with closely related research into

- *bounded arithmetic theories* (both proof theory and model theory)
- *computational complexity* (communication compl., NP search problems, circuits complexity, ...)

motivations

Holy grail:

- *Show super-polynomial lower bounds for the size of proofs in all (or as strong as possible) propositional proof systems.*

A large part of the research into proof complexity lower bounds is motivated by

1. the P vs. NP problem
2. independence results for bounded arithmetic
3. SAT algorithms

this talk

Nowadays the emphasis is often in the opposite order, stressing

- *combinatorial analysis of SAT algorithms* of the day.

In general, combinatorial approach prevails today over logic approach.

But I think that a genuine progress, i.e.

- *progress in ideas and not just in technical innovations*,

will more likely from logic than from combinatorics.
This has often been the case in past.
(Of course, developing logic ideas involves a lot of combinatorics too.)

In this talk I will present some

logical facets of proof complexity

that lead me to this view.

computational classes

- P: the class of *problems decidable in p-time*

- NP: the class of *problems defined by a condition*

$$\exists y(|y| \leq |x|^c)R(x,y)$$

  where R is p-time decidable.

p-time alg's $\rightleftharpoons$ feasible alg's:

[Smullyan'61, Bennett'62, Cobham'64, Edmonds'65, ...]

the problem

The P vs. NP problem (Cook'71)

$$\mathsf{P} =_? \mathsf{NP}$$

SAT: the set of satisfiable CNF formulas

Cook's theorem

$$\mathsf{SAT} \in \mathsf{P} \;\Leftrightarrow\; \mathsf{P} = \mathsf{NP} \;.$$

Karp'72, Levin'73

Hilbert's program

D.Hilbert's work on foundations of mathematics (early 1900s)

- Entscheidungsproblem (Hilbert-Ackermann 1928):
  Device an algorithm deciding whether a first-order formula is logically valid.
  (Leibniz's *calculus ratiocinator*, 250 years earlier)

- *Formalize and prove the consistence of mathematics*,
  including the infinitary methods of set theory.

The P vs. NP problem is obtained by scaling down to feasible world:

- any algorithm := *p-time algorithm*
- FO formula := *propositional formula*

spectrum problem

Spectrum

For a FO sentence φ, $Spec(\varphi)$ is the set of all $n \geq 1$ such that φ has a model of size n.

Scholz'52: *Characterize spectra.*

Asser'53: *Is the complement of a spectrum also a spectrum?*

Facts

1. Spectra are exactly NE sets, i.e. sets accepted by a non-deterministic machine in time $2^{O(n)}$.
2. $NE \neq coNE \Rightarrow NP \neq coNP \Rightarrow P \neq NP$.

Gödel's letter to von Neumann

In 20.3.1956 K.Gödel wrote letter to J. von Neumann where he raised the following question:

- *given*: a FO formula ψ and $n \geq 1$,
- *the task*: decide if there is a proof of ψ using $\leq n$ symbols.

Gödel remarks that

- this is algorithmically solvable by exhaustive search,
- there is $c \geq 1$ such that no algorithm using $c \cdot n$ steps works

and he writes that he sees no argument ruling out

an algorithm working in time $O(n)$ or $O(n^2)$,

and points of that

the existence of such an algorithm would have the greatest importance ... .

combinatorics

Savage's theorem

If L can be decided by a Turing machine in polynomial time then there are Boolean circuits C_n, $n \geq 1$, of polynomial size computing the characteristic function of L.

combinatorial approach to P vs. NP:

- *Try to establish super-polynomial circuits lower bounds for SAT.*

This approach is attractive and had some earlier successes but, in fact, it is *unreasonably unsuccessful*: even a non-linear lower bound is not known.

This may be contrasted with *unreasonably successful* combinatorial approach to SAT solving.

logic

logic approach:

Recalling the logic pedigree of the problem, and the fact that logic solved the Entscheidungsproblem (Turing'36, Church'36):

- *Turing machines*,
- *Halting problem and its undecidability*,

it seems sensible to consider the problem via *logic eyes*.

propositional logic

Note:

$$\varphi \notin \mathsf{SAT} \;\Leftrightarrow\; \neg\varphi \in \mathsf{TAUT}$$

TAUT: (DNF) tautologies

We can *certify* that a formula is a tautology by giving its proof in propositional calculus.

A reduction:

- *If no propositional calculus admits p-size proofs of all tautologies then TAUT is not in NP and hence P $\neq$ NP $\neq$ coNP.*

pps

The Cook-Reckhow definition, 1979

A propositional proof system (pps) is a binary relation (provability relation) $P(x, y)$ such that:

- $\tau \in \mathrm{TAUT} \;\Leftrightarrow\; \exists \pi P(\tau, \pi)$,
- $P(x, y)$ is p-time decidable.

In item 1:

$\Rightarrow$ is the *completeness* and $\Leftarrow$ is the *soundness*.

terminology: π is a P-proof of τ

logic examples

Frege systems:

- complete language
- based on a finite number of axiom schemes and inference rules

$$\frac{p \qquad p \rightarrow q}{q}$$

R = resolution:
proves that a set of clauses is unsatisfiable, operates only with clauses using one rule:

$$\frac{C \cup \{p\} \qquad D \cup \{\neg p\}}{C \cup D}$$

derives $\emptyset$.

algebraic example

algebraic proof systems: represents clauses by polynomial equations over a field **K** and proves that the starting system has no 0/1 solution.

Clause $C:\ p, q, \neg r$ is represented by polynomial $f_C:\ (1-p)(1-q)r$ in the sense that:

$$C \textit{is satisfied by } a \in \{0,1\}^3 \ \Leftrightarrow\ f_C(a) = 0\ .$$

This replaces an initial set $\mathcal{C}$ of clauses by a set of polynomial equations $\mathcal{F}$.

Hilbert's Nullstellensatz

$\mathcal{C}$ is unsatisfiable iff $\mathcal{F}' := \mathcal{F} \cup \{p^2 - p, q^2 - q, r^2 - r, \dots\}$ generates the trivial ideal in the ring $\mathbf{K}[p, q, r, \dots]$

algebraic pps: use closure properties of ideals as rules to show that $\mathcal{F}'$ generates 1

geometric example

geometric systems: represents clauses by integer linear inequalities and refutes the initial system by deriving $0 \geq 1$.

Clause C : $p, q, \neg r$ is represented by integer linear function L_C : $p + q + (1 - r)$ in the sense that:

$$C \textit{ is satisfied by } a \in \{0,1\}^3 \;\Leftrightarrow\; L_C(a) \geq 1 \;.$$

Fact

$\mathcal{C}$ is unsatisfiable iff $\mathcal{L}' := \mathcal{L} \cup \{1 \geq p \geq 0, \dots\}$ has no integer solution.

cutting planes pps: few obvious rules plus the Chvátal-Gomory cut

$$\frac{\sum_i a_i p_i \geq b}{\sum_i (a_i/c) p_i \geq \lceil b/c \rceil}$$

where $c > 0$ divides all a_i.

abstract pps

abstract pps based on a theory:

declare an ZFC proof of the formal statement

- τ *is a tautology*

to be a P_{ZFC}-proof of τ.

Any consistent theory interpreting some minimal arithmetic and having a p-time set of axioms can be used.

formal links

Definition

Given a pps P, it lengths-of-proofs function is a function $s_P : \mathrm{TAUT} \to \mathbf{N}$ defined by:

$$s_P(\tau) \ := \ \min\{|\pi| \mid P(\tau, \pi)\} \ .$$

P is p-bounded iff $s_P(\tau) \leq |\tau|^{O(1)}$

Theorem (Cook-Reckhow'79)

NP is closed under complementation iff there exists a p-bounded pps.

task

Hence our ideal task is

- *to prove super-poly lower bounds on s_P for* all *proof systems.*

But we cannot prove it one-by-one. (Maybe yes - next slide.)

What can we derive if we prove such lower bound only for some specific P? We get:

- *Lower bounds for a class of all pps which P simulates*.
- *Time lower bounds for a class of SAT algorithms that P simulates*.
- *Consistency of $P \neq NP \neq coNP$ with a first-order theory associated with P*.

simulation

Definition

$P \geq Q$, P simulates Q iff $s_P(\tau) \leq s_Q(\tau)^{O(1)}$.

It is a quasi-ordering.

The optimality problem

Is there an optimal pps (i.e. maximal) w.r.t. to $\geq$?

It is consistent with present knowledge that a Frege system is optimal.

A super-poly lower bound for an optimal pps implies that no pps is p-bounded.

SAT alg's

By a SAT algorithm we shall mean any algorithm satisfying the following universal sentence $Sound_A$:

$$\forall \varphi \ (\varphi \in \mathrm{SAT} \ \rightarrow \ \varphi(A(\varphi)) = 1) \ .$$

A as a proof system

Define a pps $Q_A(\pi, \tau)$ by

- *π is the transcript of the computation of A on $\neg\tau$ and $\tau(A(\neg\tau)) = 1$.*

$P \geq A$ abbreviates $P \geq Q_A$

from size to time

Observation

If $P \geq A$ then $time_A(\neg\tau) \geq s_P(\tau)^\epsilon$, some $\epsilon > 0$.

In particular: super-poly lower bound on the lengths of P-proofs imply that A is not p-time.

Ex's:

- R^* (tree-like R) $\geq$ DPLL,
- $P(\sim$ R) $\geq$ CDCL, ...,
- $AC^0 - F +$ counting principles, PC, CP simulate many algebro-geometric alg's.

Haken'85, Ajtai'88, ...

towards simulation

How can we show that $P \geq A$? Two options:

1. *A direct translation of computations into proofs.*

   This allows to interpret various proof theoretic results about P-proofs as statements about A-computations (space bounds, trade-offs, ...).

2. *Prove "in P" the soundness of A.*

   More general but less elementary.

PV

language L_{PV}: names for all p-time clocked algorithms

theory PV: universal statements how one algorithm is build from others

Cook'76 (uses Cobham's 1964 characterization of p-time)

Translation

For an open L_{PV}-formula $\psi(x)$ and $n \geq 1$ denote

$$||\psi||^n(p_1, \dots, p_n)$$

a *canonical* circuit evaluating the truth value of ψ on strings of length n.

$$\forall a \in \{0,1\}^n \ \psi(a) \ \Leftrightarrow \ (||\psi||^n(a) = 1) \ .$$

general correspondence

T: any r.e. consistent extension of PV in a language extending L_{PV}

A classic fact

For any such T there is a pps P_T such that:

- If $T \vdash \forall \psi(x)$, ψ open L_{PV}-formula, then formulas $||\psi||^n$ have p-size P_T-proofs.
- T proves the soundness of P_T.
- If T proves the soundness of Q then $P_T \geq Q$.

consistency of $P \neq NP$

Corollary

Any super-polynomial lower bound for P_T-proofs of any sequence of tautologies implies that $P \neq NP$ is consistent with T in the sense that

$$T \; + \; \{\neg Sound_A \mid \; \textit{all p-time clocked algorithms } A \; \}$$

is consistent.

Ex.:

Super-poly lower bounds for AC^0-Frege systems (plus some counting principles as is PHP) imply super-poly time lower bounds for a large class of currently considered SAT algorithms.

Ajtai'88, ...

from T

any given theory T determines:

- universal consequences: pps P_T
- existential consequences: a class of *witnessing functions*

ex's

P-IND: PV + the scheme of IND for open L_{PV}-formulas determines:

- pps ER: Extended R
- FP: p-time functions

NP-IND: PV + IND for NP-formulas (= bounded existential) determines:

- pps G_1: a fragment of quantified propositional calculus
- PLS: a class of NP search problems

strength of P-IND

A large part of contemporary complexity theory around P, NP, ... can be formalized in bounded arithmetic theories as are extensions of PV by

P-IND and NP-IND.

Ex's:

- NP-completeness
- the Goldreich-Levin theorem
- a construction of PRNGs from OWF
- circuit and proof complexity lower bounds
- Nisan-Wigderson generator and its uses
- natural proofs
- sorting networks
- expander constructions
- . . .

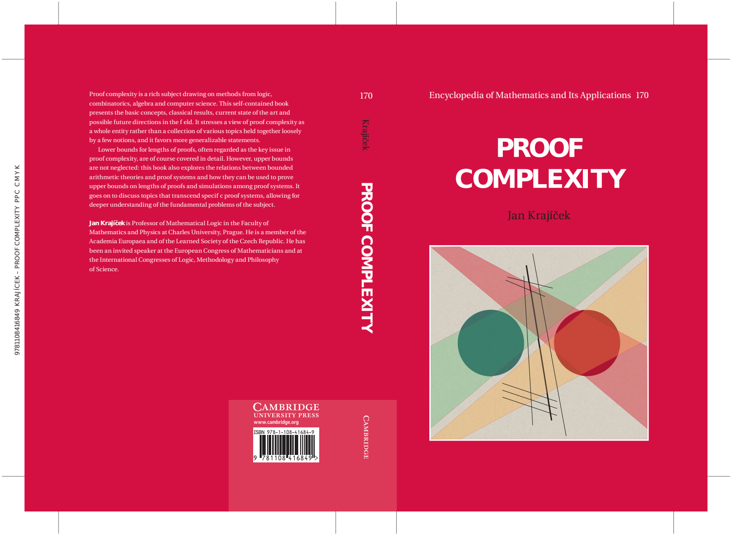
hypothetical situation

Consider the following situation:

- *You have a p-time clocked SAT algorithm A*
- *but you cannot prove $Sound_A$ in P-IND, only in NP-IND (or worse).*

Can you still claim that SAT is feasible?

a general reference



9781108416849 KRAJÍCEK – PROOF COMPLEXITY PPC C M Y K

Proof complexity is a rich subject drawing on methods from logic, combinatorics, algebra and computer science. This self-contained book presents the basic concepts, classical results, current state of the art and possible future directions in the f eld. It stresses a view of proof complexity as a whole entity rather than a collection of various topics held together loosely by a few notions, and it favors more generalizable statements.

Lower bounds for lengths of proofs, often regarded as the key issue in proof complexity, are of course covered in detail. However, upper bounds are not neglected: this book also explores the relations between bounded arithmetic theories and proof systems and how they can be used to prove upper bounds on lengths of proofs and simulations among proof systems. It goes on to discuss topics that transcend specif c proof systems, allowing for deeper understanding of the fundamental problems of the subject.

Jan Krajíček is Professor of Mathematical Logic in the Faculty of Mathematics and Physics at Charles University, Prague. He is a member of the Academia Europaea and of the Learned Society of the Czech Republic. He has been an invited speaker at the European Congress of Mathematicians and at the International Congresses of Logic, Methodology and Philosophy of Science.

CAMBRIDGE
UNIVERSITY PRESS
www.cambridge.org
ISBN 978-1-108-41684-9

170

Krajíček

PROOF COMPLEXITY

CAMBRIDGE

Encyclopedia of Mathematics and Its Applications 170

PROOF COMPLEXITY

Jan Krajíček

optimality

NO optimal pps $\Rightarrow$ NE $\neq$ coNE $\Rightarrow$ NP $\neq$ coNP

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., quantitative Gödel's thms, games on graphs, etc., and quite a results characterizing the existence of optimal systems 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).

back to Hilbert

Hilbert asked to prove by finitary means the consistency of Math. (Gödel'31: impossible.)

quantitative version $Con_T(\tilde{n})$:

- formalizes that T is consistent w.r.t. proofs of size $\leq n$,
- $\tilde{n}$ is the dyadic numeral of size $\log n$.

Theorem (K.-Pudlák'89)

An optimal pps exists iff there is a theory S such that for all T, S proves $Con_T(\tilde{n})$ in size $poly(n)$.

(Leaving out technical assumptions.)