What is Proof Complexity?

Jan Krajíček

Charles University

krajicek@karlin.mff.cuni.cz

The theme

What do we mean by saying that

a proof is complex

and can we show that

all proofs

of some theorem must be complex?

A restriction

We restrict to

formal mathematical proofs.

- Clearly defined.
- Already subject to mathematical investigations.
- General: capture *all there is*.

Ex.: Classification of the finite simple groups

Original proof:

- Many individual papers spanning several decades and covering various parts of the topic.
- No unique statement of the theorem: Experts agreed (then disagreed and agreed again) that the classification is complete.

2nd generation proof:

$$\text{Vol.}1, \ldots, \text{Vol.}8, \ldots, \text{Vol.}13?$$

totaling cca 5000 pages. A specific statement is given.

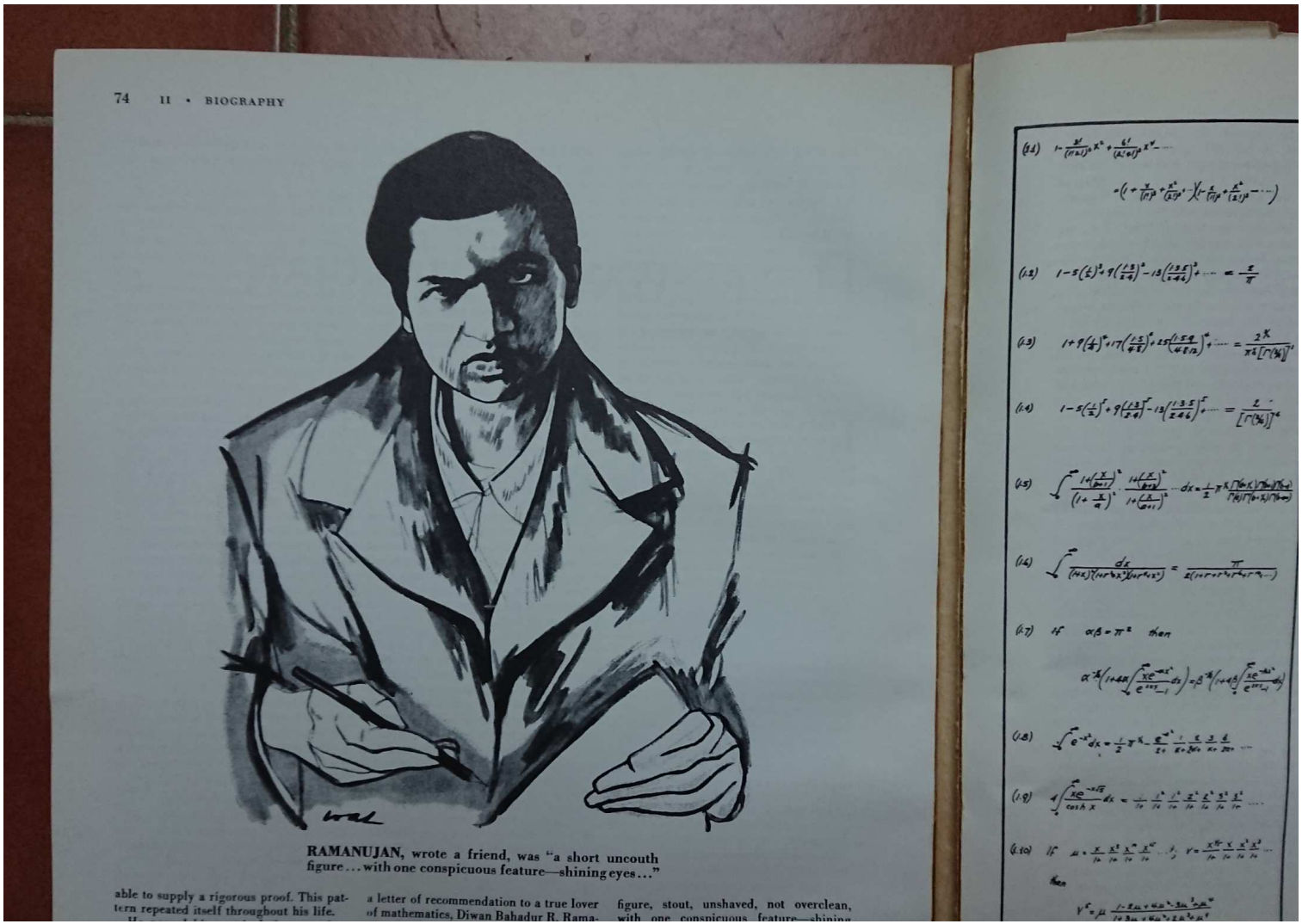
Ex.: Four color theorem

- Many particular cases to check.
- Computer assisted.

$$\downarrow\downarrow\downarrow\downarrow\downarrow$$

Also the correctness of the algorithm needs to be proved.

Ramanujan



74 II • BIOGRAPHY

RAMANUJAN, wrote a friend, was "a short uncouth figure ... with one conspicuous feature—shining eyes ..."

able to supply a rigorous proof. This pattern repeated itself throughout his life.

a letter of recommendation to a true lover of mathematics, Diwan Bahadur R. Rama-

figure, stout, unshaved, not overclean,

Fermat's last theorem

Annals of Mathematics, **142** (1995), 443–551

Modular elliptic curves and Fermat's Last Theorem

By Andrew Wiles*

For Nada, Clare, Kate and Olivia

Cubum autem in duos cubos, aut quadratoquadratum in duos quadratoquadratos, et generaliter nullam in infinitum ultra quadratum potestatem in duos ejusdem nominis fas est dividere: cujus rei demonstrationem mirabilem sane detexi. Hanc marginis exiguitas non caperet.

Pierre de Fermat

Introduction

An elliptic curve over **Q** is said to be modular if it has a finite covering by

A common feature

All these proofs need a lot of time to verify.

$\downarrow\downarrow\downarrow\downarrow\downarrow$

- What is TIME?
- Who is the VERIFIER?

Verifier & time

Verifier: Turing machine V

Algorithmic verifier *forces* that a proof contains all information needed to accept it: No creative input from the verifier is needed.

- Input: a statement φ and a purported proof π.
- Output $V(\varphi, \pi)$: ACCEPT or REJECT.

Time

Time - the number of elementary steps - of the computation checking the proof.

Ex's of verifiers

Predicate calculus

Checking a proof in any of the usual formalizations of predicate calculus can be done algorithmically and quickly.

Recursive theories

Checking a proof in a recursive theory (e.g. ZFC or PA) needs one extra subroutine algorithmically recognizing axioms of the theory.

- Proof complexity questions in these cases often reduce to the Halting problem.

Propositional calculus - our interest

The role of propositional Halting problem is taken - presumably - by propositional satisfiability SAT but it is open whether it can be solved quickly. This is the famous P vs. NP problem.

A technical maneuver

Original V:

- input: φ, π,
- input length: $|\varphi| + |\pi| = m$,
- time: t.

Modified V':

- input: φ and a padded proof $\pi' := \pi\ 1^{(t)}$,
- input length: $|\varphi| + |\pi'| = m + t$,
- time: $\sim t$, i.e. linear in the input size.

We are trading time for length!

Cook-Reckhow proof systems

Propositional proof system

A p-time decidable relation $Q(x, y)$ such that for all $\varphi \in \{0,1\}^*$:

$$\varphi \in \mathrm{TAUT} \;\Leftrightarrow\; \exists \pi\, Q(\varphi, \pi)\ .$$

- $\Rightarrow$: completeness,
- $\Leftarrow$: soundness,
- TAUT: propositional tautologies in a complete language.

Ex.: Frege systems

- a complete language,
- a finite number of sound axioms schemes and schematic inference rules,
- implicationally complete.

A specific Frege system:

Modus ponens and axiom schemes:

$$p \to (q \to p)$$

$$p \to (q \to r) \to [(p \to q) \to (p \to r)]$$

$$(\neg p \to \neg q) \to [(\neg p \to q) \to p]$$

Complexity theory

Decision problem

A problem is represented by its YES instances, a language $L \subseteq 0, 1^*$:

$$u \in L \quad \text{iff} \quad \text{YES to } u \ .$$

Complexity class

A class $\mathcal{C}$ of languages having *some common computational features*; e.g. can be all solved by simple algorithms.

P vs. NP

$\mathcal{P}$:

Problems solvable by a deterministic polynomial-time algorithm.

Ex.: Is $a \in \mathbf{N}$ odd?

$\mathcal{NP}$:

Problems for which the affirmative answers $u \in L$ can be certified by a polynomial size string and checked in p-time.

Ex.: $\exists x, y, ax^2 + by = c$?

P vs. NP problem:

$$\mathcal{P} = \mathcal{NP}?$$

This is a fundamental problem of MATH and theoretical CS.

Proof complexity

Main proof complexity problem

Is there a polynomially bounded propositional proof system?

Q p-bounded: $\exists \pi(|\pi| \leq |\varphi|^{const})\ Q(\varphi, \pi)$.

Remark: It is open if a Frege system is p-bounded.

Theorem (Cook-Reckhow'79)

$$NO \Leftrightarrow \mathcal{NP} \neq co\mathcal{NP} \Rightarrow \mathcal{P} \neq \mathcal{NP} .$$

Our task

Given a propositional proof system Q find a sequence of hard tautologies:

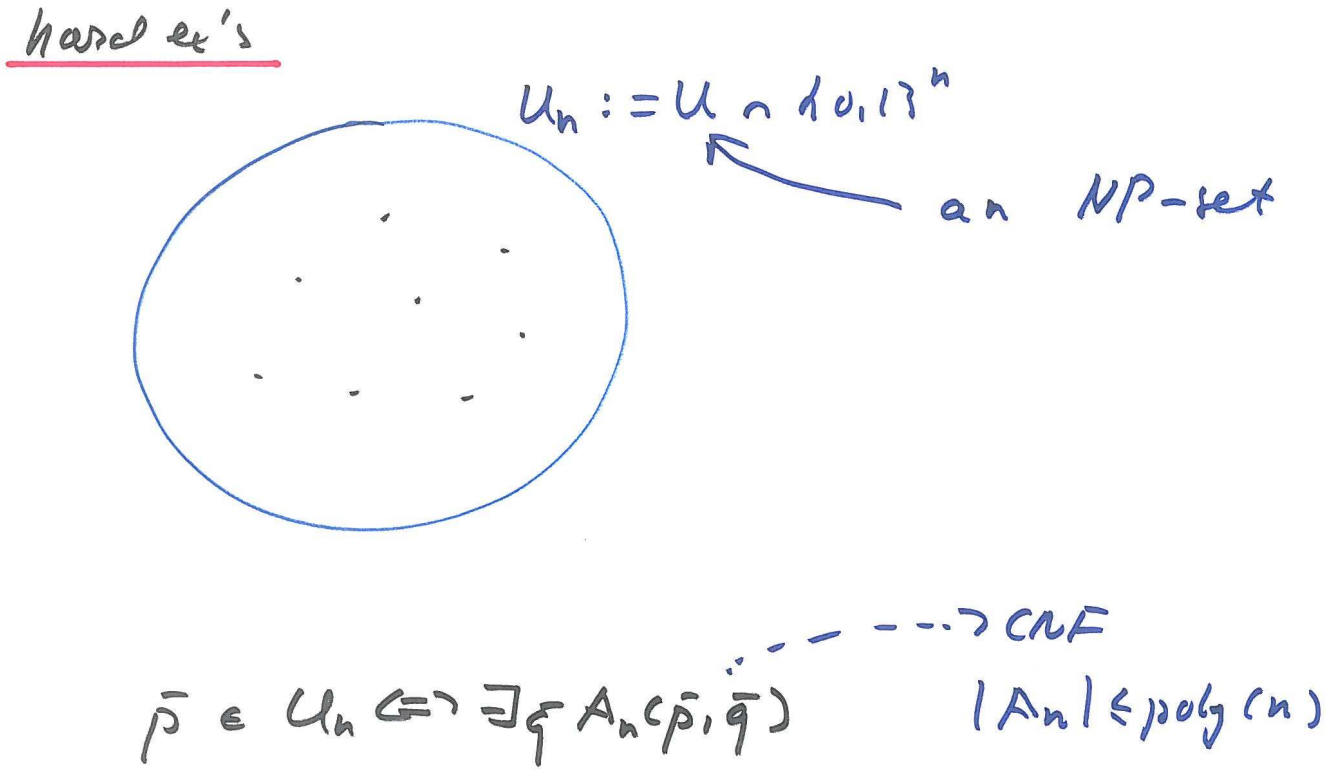
$$\ldots\ ,\ \tau_n(p_1, \ldots, p_n)\ \ \ldots$$

for $n \longrightarrow \infty$ such that

$$\min\{|\pi| \mid Q(\tau_n, \pi)\}$$

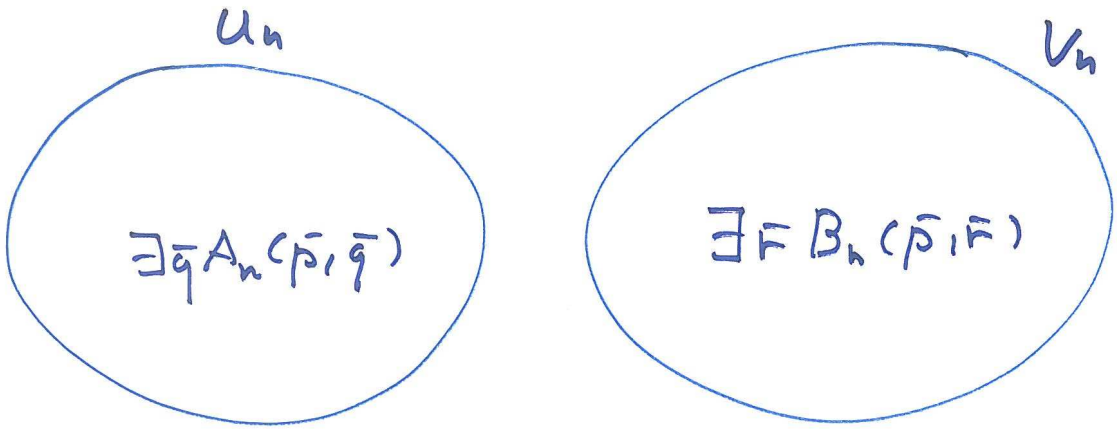
cannot be bounded by a polynomial in $|\tau_n|$.

NP sets and formulas



19.

Disjoint NP pairs



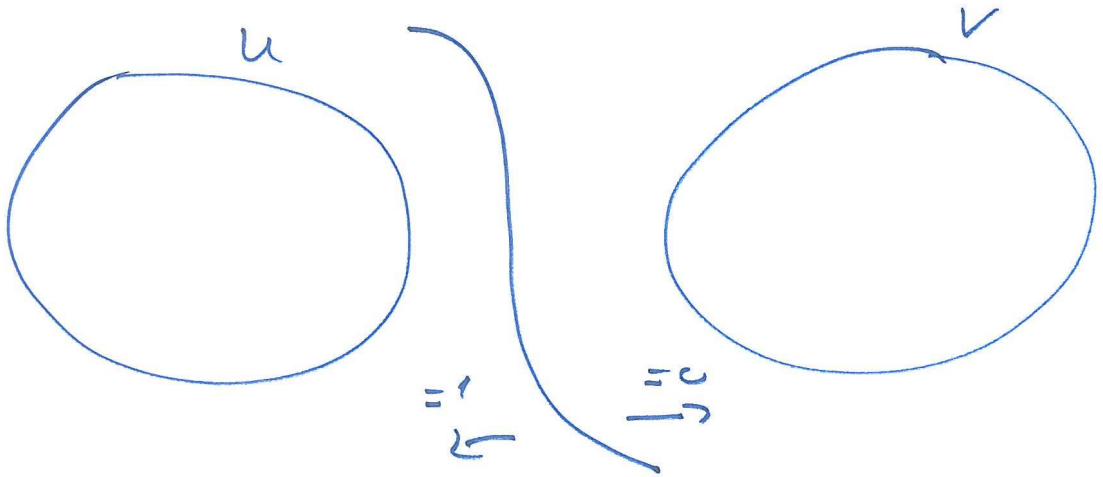
$$\Sigma_n := \neg A_n(\bar{p}, \bar{q}) \vee \neg B_n(\bar{p}, \bar{r})$$

$$[\, \Sigma_n \in TAUT \iff U_n \cap V_n = \emptyset \,]$$

20.

Separation task

a computational task:



a separating alg.

21.

Feasible interpolation method

Feasible interpolation idea

A short Q-proof of $U \cap V = \emptyset$ yields a computational information for the separation task: It can be performed better than by the exhaustive search.

That is:

- If U and V are hard to separate then formulas $\tau_n = \neg A_n \vee \neg B_n$ need long Q-proofs.

This method applies to the widest class of proof systems for which we have non-trivial lower bounds. These include weak logical proof systems (e.g. resolution), geometric proof systems (e.g. cutting planes system) and algebraic proof systems (e.g. polynomial calculus), as well as some ad hoc proof systems (e.g. the OBDD system).

Hard pairs U, V

Encryption of one bit:

$$b = 0, 1 \text{ and random string } w \rightarrow E(b, w) \in \{0, 1\}^* .$$

Hard pair

U: encryptions of $b = 0$, V: encryptions of $b = 1$.

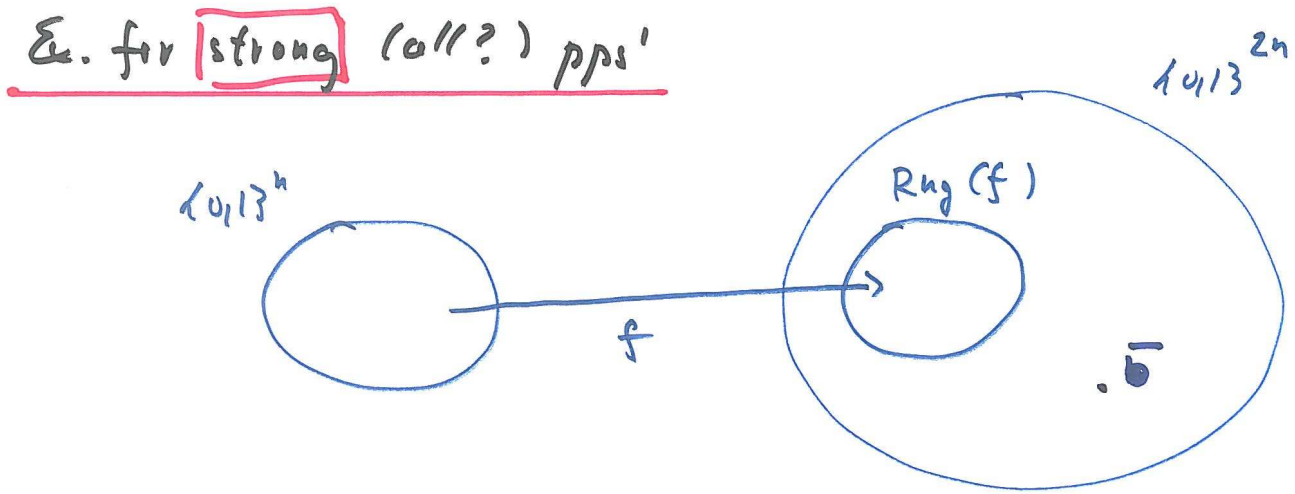
Ex.: RSA, presumably secure: U, V are not feasibly separable.

There is an unconditional monotone variant of the method:

- U closed upwards or V downwards,
- the separation algorithm is *monotone*.

Ex.: U: graphs on n vertices containing a clique of size $\geq n^{1/2}$,
V: graphs that are $< n^{1/2}$-colorable.

Towards length lower bound



$\tau_{\bar{b}} :=$ "$f(p_1, \ldots, p_n) \neq \bar{b}$"

[proof compl. generators]

23.

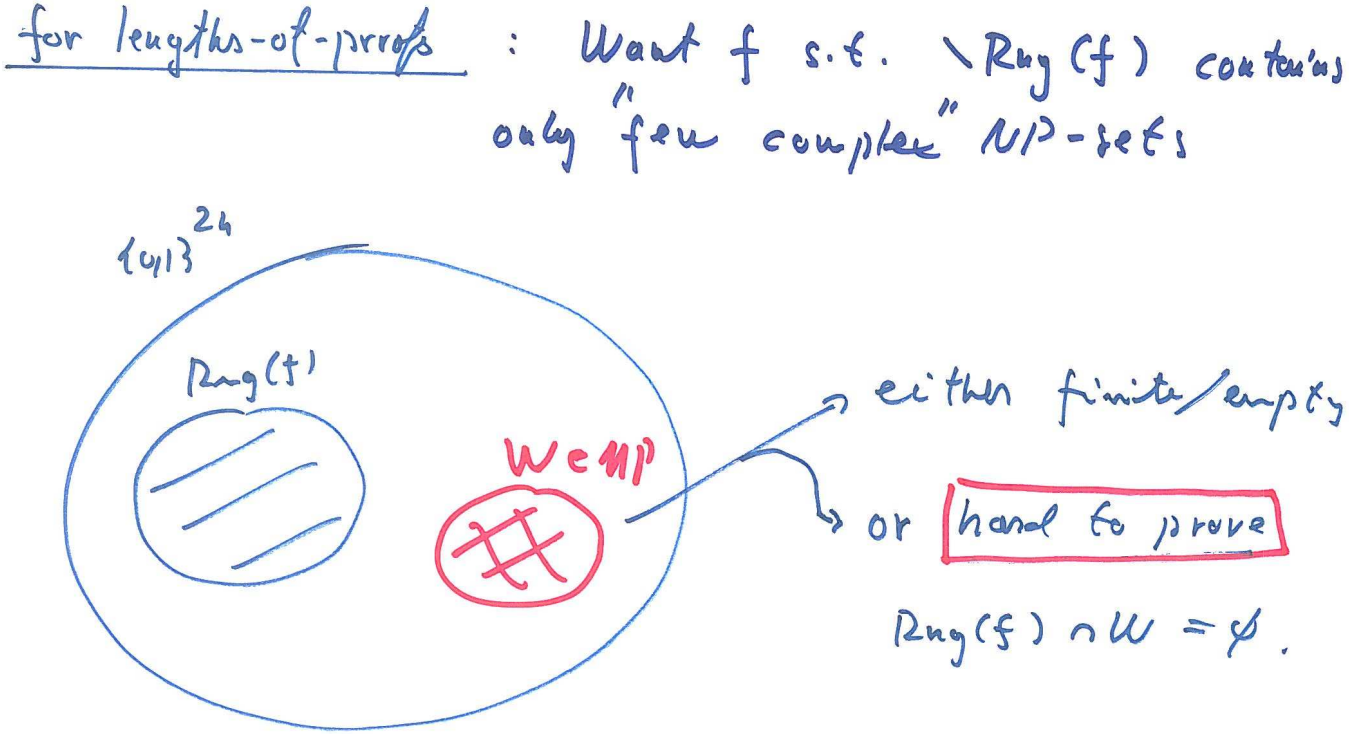
Proof complexity generators

- Work whenever feasible interpolation does but also for some proof systems that are outside the scope of feasible interpolation (e.g. for various subsystems of Frege systems).
- It is consistent with present knowledge that there is a generator hard for all proof systems.
- Allows for various conditional results:

Theorem (Limiting proof search)

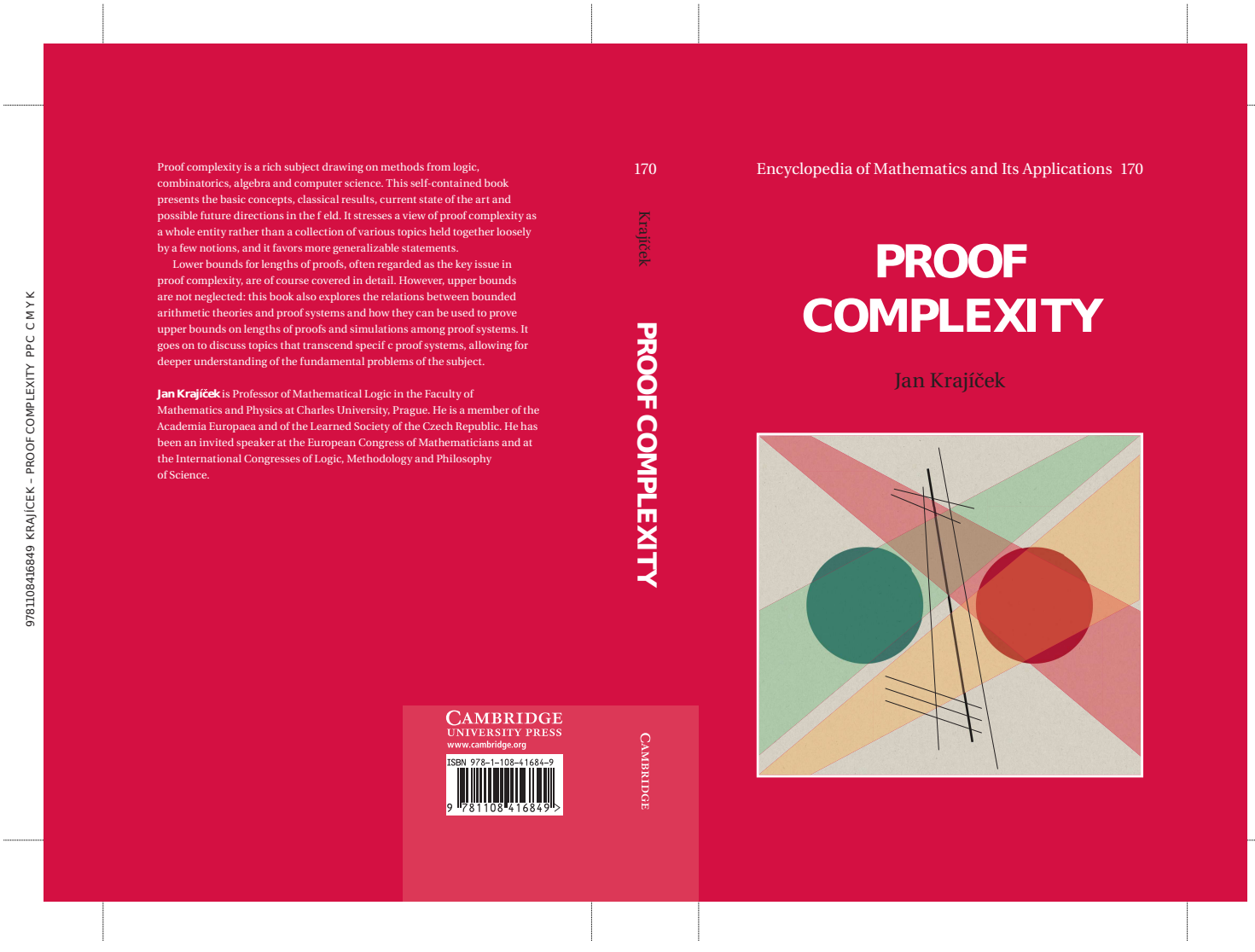
If f is a pseudo-random number generator then for no Q there is a feasible algorithm that given $b \notin Rng(f)$ constructs a Q-proof of $\tau_b(f)$.

Generator requirement



25.

A reference


9781108416849 KRAJÍČEK – 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
9 781108 416849
CAMBRIDGE
170
Krajíček
PROOF COMPLEXITY
Encyclopedia of Mathematics and Its Applications 170
PROOF COMPLEXITY
Jan Krajíček

Candidate generators

All plausible candidates (not too many!) have cca the following form:

Input:

- $x \in \{0,1\}^n$ describes an algorithm of some sort with k input bits,
- $A \subseteq \{0,1\}^k$,
- A is defined from ℓ bits y and $n + \ell < |A|$.

Output:

- string z of $m := |A|$ values of algorithm x on all inputs from A.

$$f \ : \ (x,y) \in \{0,1\}^{n+\ell} \ \rightarrow \ z \in \{0,1\}^m \ .$$