Propositional Proof Complexity

Edward A. Hirsch

http://logic.pdmi.ras.ru/~hirsch

Steklov Institute of Mathematics at St.Petersburg
St.Petersburg State University

Mathematical proofs

- What is a (complete) mathematical proof?

Mathematical proofs

- What is a (complete) mathematical proof?
  - A text.
    Not an interactive procedure.

Mathematical proofs

- What is a (complete) mathematical proof?
  - A text.
    Not an interactive procedure.
  - Can be verified.
    Verification does not need creative work.

Mathematical proofs

- What is a (complete) mathematical proof?
  - A text.
    Not an interactive procedure.
  - Can be verified.
    Verification does not need creative work.
  - Includes everything it cites.
    Does not use unproven statements.

Mathematical proofs

- What is a (complete) mathematical proof?
  - A text.
    Not an interactive procedure.
  - Can be verified.
    Verification does not need creative work.
  - Includes everything it cites.
    Does not use unproven statements.
- Our particular case:

Mathematical proofs

- What is a (complete) mathematical proof?
  - A text.
    Not an interactive procedure.
  - Can be verified.
    Verification does not need creative work.
  - Includes everything it cites.
    Does not use unproven statements.
- Our particular case:
  - Polynomial-time verifiable.

Mathematical proofs

- What is a (complete) mathematical proof?
  - A text.
    Not an interactive procedure.
  - Can be verified.
    Verification does not need creative work.
  - Includes everything it cites.
    Does not use unproven statements.
- Our particular case:
  - Polynomial-time verifiable.
  - Theorems are propositional tautologies:

$$(x_1 \wedge x_2) \vee \overline{x_1} \vee \overline{x_2}$$

Mathematical proofs

- What is a (complete) mathematical proof?
  - A text.
    Not an interactive procedure.
  - Can be verified.
    Verification does not need creative work.
  - Includes everything it cites.
    Does not use unproven statements.
- Our particular case:
  - Polynomial-time verifiable.
  - Theorems are propositional tautologies:

$$(x_1 \& x_2) \vee \neg x_1 \vee \neg x_2$$

Mathematical proofs

- What is a (complete) mathematical proof?
  - A text.
    Not an interactive procedure.
  - Can be verified.
    Verification does not need creative work.
  - Includes everything it cites.
    Does not use unproven statements.
- Our particular case:
  - Polynomial-time verifiable.
  - Theorems are propositional tautologies:

$$(x_1 \wedge x_2) \vee \overline{x_1} \vee \overline{x_2}$$

  - Size 2^n is trivial, interested in shorter proofs.

What do we prove

- Proof by contradiction:
  instead of deriving
  $$F = (x_1 \wedge x_2) \vee \overline{x_1} \vee \overline{x_2}$$
  we can refute
  $$\overline{F} = (\overline{x_1} \vee \overline{x_2}) \wedge x_1 \wedge x_2$$

What do we prove

- Proof by contradiction:
  instead of deriving
  $$F = (x_1 \wedge x_2) \vee \overline{x_1} \vee \overline{x_2}$$
  we can refute
  $$\overline{F} = (\overline{x_1} \vee \overline{x_2}) \wedge x_1 \wedge x_2$$
- W.l.o.g.: Conjunctive Normal Form
  $$\bigwedge_{j=1}^{t} (\ell_{j1} \vee \ldots \vee \ell_{jk}), \qquad (\text{where } \ell_{jt} = x_i \text{ or } = \overline{x_i})$$

  Tseytin's translation:
  $$F = G \vee H \qquad \mapsto \qquad (\overline{x_F} \vee x_G \vee x_H) \wedge (\overline{x_G} \vee x_F) \wedge (\overline{x_H} \vee x_F)$$

  We work with "clauses"!
  $$x \vee \overline{y} \vee \overline{z}$$

What do we prove

- Proof by contradiction:
  instead of deriving
  $$F = (x_1 \wedge x_2) \vee \overline{x_1} \vee \overline{x_2}$$
  we can refute
  $$\overline{F} = (\overline{x_1} \vee \overline{x_2}) \wedge x_1 \wedge x_2$$
- Useful tautologies:

  Propositional Pigeonhole Principle:

  - $x_{ij} \sim i$-th pigeon in j-th hole
    - $i = 1, \ldots, n+1$
    - $j = 1, \ldots, n$
  - Pigeon i sits somewhere:
    $$x_{i1} \vee x_{i2} \vee \ldots \vee x_{in}$$
  - Two pigeons $i \neq i'$ cannot share a hole:
    $$\overline{x_{ij}} \vee \overline{x_{i'j}}$$

What do we prove

- Proof by contradiction:
  instead of deriving
  $$F = (x_1 \wedge x_2) \vee \overline{x_1} \vee \overline{x_2}$$
  we can refute
  $$\overline{F} = (\overline{x_1} \vee \overline{x_2}) \wedge x_1 \wedge x_2$$
- Useful tautologies:

  Tseytin formulas:

  - Graph (V, E),
    x_e is a label for edge $e \in E$.
  - $\bigoplus_{e \ni v} x_e = 1$ for every $v \in V$.
  - Contradictory if $|E|$ is odd.

How do we prove

Logic-like systems

Logical rule:

$$\frac{F_1, \quad \ldots, \quad F_s}{G}$$

if F_i's semantically imply G.

For example, Modus Ponens:

$$\frac{F, \quad F \Rightarrow H}{H}$$

How do we prove

Logic-like systems

Logical rule:

$$\frac{F_1, \quad \ldots, \quad F_s}{G}$$

if F_i's semantically imply G.

For example, Modus Ponens:

$$\frac{F, \quad F \supset H}{H}$$

Resolution

$$\frac{x \vee \ell_1 \vee \ldots \vee \ell_k, \quad \overline{x} \vee m_1 \vee \ldots \vee m_t}{\ell_1 \vee \ldots \vee \ell_k \vee m_1 \vee \ldots \vee m_t} \qquad \textbf{(Resolution)}$$

(Clauses [disjunctions] are treated as sets,
C and D contain no contrary pairs.)

For example,

$$\frac{x \vee y \vee \overline{z}, \quad z \vee y}{x \vee y}$$

Resolution

$$\frac{x \vee C, \quad \overline{x} \vee D}{C \vee D} \qquad \textbf{(Resolution)}$$

(Clauses [disjunctions] are treated as sets,
C and D contain no contrary pairs.)

For example,

$$\frac{x \vee y \vee \overline{z}, \quad z \vee y}{x \vee y}$$

Resolution

$$\frac{C}{C \vee x} \qquad \textbf{(Weakening)}$$

$$\frac{x \vee C, \quad \overline{x} \vee D}{C \vee D} \qquad \textbf{(Resolution)}$$

(Clauses [disjunctions] are treated as sets,
C and D contain no contrary pairs.)

For example,

$$\frac{x \vee y \vee \overline{z}, \quad z \vee y}{x \vee y}$$

How do we prove

Non-logic-like systems

Formulate disjunctions as statements about integers:

$$x_1 \vee x_2 \vee \overline{x_3}$$

$$x_1 + x_2 + (1 - x_3) \geq 1$$

$$(1 - x_1)(1 - x_2)x_3 = 0$$

What can we do?

- make linear combinations,
- multiply by appropriate constants (or polynomials),
- ...and more.

How do we prove

Non-logic-like systems

Formulate disjunctions as statements about integers:

$$x_1 \vee x_2 \vee \overline{x_3}$$

$$x_1 + x_2 + (1 - x_3) \geq 1$$

$$(1 - x_1)(1 - x_2)x_3 = 0$$

What can we do?

- make linear combinations,
- multiply by appropriate constants (or polynomials),
- ...and more.

Nullstellensatz

System of polynomial equations:

- $x_i^2 - x_i = 0$ (denoted $f_i = 0$), where $i = 1, \ldots, n$,
- original clauses as polynomials: $f_j = 0$, where $j = n + 1, \ldots$.

(Weak) Hilbert's Nullstellensatz:

there are no solutions (over an algebraically closed field) iff

- 1 is in the ideal generated by f_k's iff
- there are polynomials g_k's such that $\sum_k f_k g_k \equiv 1$

The proof: g_k's.

Verification: open brackets...

Degree upper bound: exponential in the general case,
linear in the Boolean case (exercise).

Propositional proof system

Formal definition

- Cook, Reckhow, 1974:
  Propositional proof system for L is
  a polynomial-time computable function

$$f : \{0,1\}^* \twoheadrightarrow L$$

Propositional proof system

Formal definition

- Cook, Reckhow, 1974:
  Propositional proof system for L is
  a polynomial-time computable function

$$f : \{0,1\}^* \twoheadrightarrow L$$

  - $L = \{\text{tautologies}\}$
  - Proof contains the statement.

Propositional proof system

Formal definition

- Cook, Reckhow, 1974:
  Propositional proof system for L is
  a polynomial-time computable function

$$f\colon \{0,1\}^* \twoheadrightarrow L$$

  - $L = \{\text{contradictory formulas in CNF}\}$
  - Proof contains the statement.

- Polynomially bounded proof system:
  there is a polynomial p such that

$$\forall F \in L\ \exists \pi\ \underbrace{f(\pi) = F}_{\pi \text{ is a proof}} \text{ and } \mathrm{length}(\pi) \leq p(\mathrm{length}(F))$$

Propositional proof system

Formal definition

- Cook, Reckhow, 1974:
  Propositional proof system for L is
  a polynomial-time computable function

$$f\colon \{0,1\}^* \twoheadrightarrow L$$

  - $L = \{\text{contradictory formulas in CNF}\}$
  - Proof contains the statement.
- Polynomially bounded proof system:
  there is a polynomial p such that

$$\forall F \in L\ \exists \pi\ \underbrace{f(\pi) = F}_{\pi \text{ is a proof}} \text{ and } \mathrm{length}(\pi) \leq p(\mathrm{length}(F))$$

- Polynomially bounded proof system for propositional tautologies $\Leftrightarrow$ $\mathbf{NP} = \mathbf{co\text{-}NP}$.

Questions to study

- Is this particular system polynomially bounded?
  (exponential lower bounds)
- Is this system (strictly) stronger than that system?
  (lattice or proof systems)
- Is there an "optimal" system that provides shortest possible proofs?
- Can we find proofs efficiently (automatizability)?

Exponential lower bounds for Resolution

- Gregory Tseytin, 1968: superpolynomial bound
- Zvi Galil, 1977: exponential bound (regular, Tseytin formulas)
- Armin Haken, 1985: exponential bound (general, PHP)
- Alasdair Urquhart, 1987: exponential bound (general, Tseytin formulas)

Eli Ben-Sasson, Avi Wigderson

Unified framework

- Width of clause: the number of literals $\ell_1 \vee \ldots \vee \ell_k$.
- Width of proof: the width of the widest clause.
- Proof strategy (formula F with n variables):
  - show a lower bound on the width (ideally, $\Omega(n)$),
  - show that short proofs can be made narrow:
    proof of size $S \implies$ proof of width $\sqrt{n \log S} + \mathrm{width}(F)$,
  - conclude there are exponential-size proofs only.
- Similar framework for algebraic systems (degree lower bounds).

Width $\implies$ Size

Notation:

$F|_{\ell=1}$: substitute ℓ by 1 (ℓ can be x or $\overline{x}$)

axioms $\vdash_w$ clause: derivation of width w

Width $\implies$ Size

Lemma 1: $F|_{\ell=1} \vdash_w C \implies F \vdash_{w+1} C \vee \bar{\ell}$

Lemma 2:

$$\left.\begin{array}{ll} F|_{\ell=1} \vdash_{w-1} & \text{False} \\ F|_{\ell=0} \vdash_{w} & \text{False} \end{array}\right\} \implies F \text{ has refutation of width } \max(w, F)$$

Theorem: proof of size $S \implies$ proof of width $O(\sqrt{n \log S}) + \mathrm{width}(F)$.

Proof: induction: split by ℓ killing many wide clauses.

Width lower bound

Complexity measure: the number of axioms

Complexity measure $\mu\colon \{\text{clauses}\} \to \mathbb{R}_{\geq 0}$

- $\mu(\text{axiom}) \leq 1$
- $\mu(\mathit{False})$ is large
- μ changes smoothly throughout the derivation
- $\implies$ there is a clause with intermediate μ

$\mu(C) =$ the minimum number of axioms implying C

$$\frac{A, \quad B}{C} \quad \implies \quad \mu(C) \leq \mu(A) + \mu(B)$$

hence there is a clause C^* such that

$$\mu(C^*)/\mu(\mathit{False}) \in [1/3\,..\,2/3]$$

Width lower bound

Width vs expansion

- There is a clause C^* such that

$$\mu(C^*)/\mu(\mathit{False}) \in [1/3\,..\,2/3]$$

- Set S of axioms implying C^*.
- Boundary: includes variables x such that there is an assignment α such that $S(\alpha) = 0$, but flipping x satisfies S.
- Large if we work on an expander graph.
- Claim: every boundary variable appears in C^*.

More examples: Cutting Planes [Gomory, Chvátal]

Clause $\ell_1 \vee \ell_2 \vee \ldots \vee \ell_k \;\mapsto\;$ inequality $\ell_1 + \ell_2 + \ldots + \ell_k \geq 1$.

$x \vee \overline{y} \;\mapsto\; x + (1 - y) \geq 1 \;\mapsto\; x - y \geq 0$

Additional axioms: $x \geq 0$, $-x \geq -1$.

$$\frac{A \geq a, \quad B \geq b}{\alpha A + \beta B \geq \alpha a + \beta b} \qquad (\alpha, \beta > 0) \qquad \textbf{(Linear combination)}$$

More examples: Cutting Planes [Gomory, Chvátal]

Clause $\ell_1 \vee \ell_2 \vee \ldots \vee \ell_k \;\mapsto\;$ inequality $\ell_1 + \ell_2 + \ldots + \ell_k \geq 1$.

$x \vee \overline{y} \;\mapsto\; x + (1 - y) \geq 1 \;\mapsto\; x - y \geq 0$

Additional axioms: $x \geq 0$, $-x \geq -1$.

$$\frac{A \geq a}{A \geq \lceil a \rceil} \qquad \textbf{(Rounding)}$$

$$\frac{A \geq a, \quad B \geq b}{\alpha A + \beta B \geq \alpha a + \beta b} \qquad (\alpha, \beta > 0) \qquad \textbf{(Linear combination)}$$

More examples: Cutting Planes [Gomory, Chvátal]

Clause $\ell_1 \vee \ell_2 \vee \ldots \vee \ell_k \mapsto$ inequality $\ell_1 + \ell_2 + \ldots + \ell_k \geq 1$.

$x \vee \overline{y} \mapsto x + (1 - y) \geq 1 \mapsto x - y \geq 0$

Additional axioms: $x \geq 0$, $-x \geq -1$.

$$\frac{A \geq a}{A \geq \lceil a \rceil} \qquad \textbf{(Rounding)}$$

$$\frac{A \geq a, \quad B \geq b}{\alpha A + \beta B \geq \alpha a + \beta b} \qquad (\alpha, \beta > 0) \qquad \textbf{(Linear combination)}$$

Propositional Pigeonhole Principle:

- Pigeon i sits somewhere:

$$x_{i1} + x_{i2} + \ldots + x_{in} \geq 1$$

- Two pigeons $i \neq i'$ cannot share a hole:

$$-x_{ij} - x_{i'j} \geq -1$$

Cutting Planes facts

- Stronger than Resolution:
  - CP simulates Resolution step by step.

    Exercise. Where do we need the rounding rule?
  - Short proofs of PHP in CP.
  - No short proofs of PHP in Resolution.

Cutting Planes facts

- Stronger than Resolution:
  - CP simulates Resolution step by step.

    Exercise. Where do we need the rounding rule?
  - Short proofs of PHP in CP.
  - No short proofs of PHP in Resolution.
- Exponential lower bounds for CP.

Clique-coloring tautologies

Either G contains no $n-$clique $\vee$ G is not $(n-1)$-colorable,

Clique-coloring tautologies

Either G contains no $n-$clique $\vee$ G is not $(n-1)$-colorable,

i.e., $\nexists$ two graph homomorphisms $K_n \xrightarrow{q} G \xrightarrow{r} K_{n-1}$.

Clique-coloring tautologies

Either G contains no n–clique $\vee$ G is not $(n-1)$-colorable,

i.e., $\nexists$ two graph homomorphisms $K_n \xrightarrow{q} G \xrightarrow{r} K_{n-1}$.

$G = (V, E)$, $\quad |V| = m$, $\quad p_{ij} \equiv (\{i,j\} \in E)$.

- Each clique node is mapped to the graph: $\sum_{i=1}^{n} q_{ki} \geq 1$.
- ...to a single specific vertex: $\sum_{i=1}^{n} q_{ki} \leq 1$.
- ...different nodes are mapped to different vertices: $\sum_{k=1}^{m} q_{ki} \leq 1$.
- Every two nodes are connected in the graph:

$$q_{ki} + q_{k',j} \leq p_{ij} + 1 \qquad (k \neq k',\ i < j).$$

- Every vertex has a color: $\sum_{\ell=1}^{m-1} r_{i\ell} \geq 1$.
- The coloring is correct: $p_{ij} + r_{i\ell} + r_{j\ell} \leq 2 \qquad (i < j)$.

Clique-coloring tautologies

Either G contains no n–clique $\vee$ G is not $(n-1)$-colorable,

i.e., $\nexists$ two graph homomorphisms $K_n \xrightarrow{q} G \xrightarrow{r} K_{n-1}$.

$G = (V, E)$, $\quad |V| = m$, $\quad p_{ij} \equiv (\{i,j\} \in E)$.

- Each clique node is mapped to the graph: $\sum_{i=1}^{n} q_{ki} \geq 1$.
- ...to a single specific vertex: $\sum_{i=1}^{n} q_{ki} \leq 1$.
- ...different nodes are mapped to different vertices: $\sum_{k=1}^{m} q_{ki} \leq 1$.
- Every two nodes are connected in the graph:

$$q_{ki} + q_{k',j} \leq p_{ij} + 1 \qquad (k \neq k',\ i < j).$$

- Every vertex has a color: $\sum_{\ell=1}^{m-1} r_{i\ell} \geq 1$.
- The coloring is correct: $p_{ij} + r_{i\ell} + r_{j\ell} \leq 2 \qquad (i < j)$.

The composition of q and r is the PHP!

Craig's interpolation theorem

The propositional case

Theorem

If $A(\vec{x}, \vec{y}) \supset B(\vec{x}, \vec{z})$,
then one can construct $C(\vec{x})$ *such that*
$A(\vec{x}, \vec{y}) \supset C(\vec{x})$ *and* $C(\vec{x}) \supset B(\vec{x}, \vec{z})$.

In general, C can be large (exponential-size)!

Our case: $\mathrm{Clique}(\vec{p}, \vec{q}) \supset \overline{\mathrm{Coloring}}(\vec{p}, \vec{r})$

Craig's interpolation theorem

The propositional case

Theorem

If $A(\vec{x}, \vec{y}) \wedge \overline{B}(\vec{x}, \vec{z})$ *is wrong,*
then one can construct $C(\vec{x})$ *telling what is wrong:*
$\overline{C}(\vec{x}) \supset A(\vec{x}, \vec{y})$ *and* $C(\vec{x}) \supset B(\vec{x}, \vec{z})$.

In general, C can be large (exponential-size)!

Our case: $\mathrm{Clique}(\vec{p}, \vec{q}) \wedge \mathrm{Coloring}(\vec{p}, \vec{r})$

Cutting Planes: exponential lower bound

- Monotone interpolation.

- Exponential lower bound on the interpolating circuit size.

Cutting Planes: exponential lower bound

- Monotone interpolation.

Theorem (Krajıček, Pudlák)

Consider $A(\vec{p}, \vec{q}) \supset B(\vec{p}, \vec{r})$, assume that p_i occur without further negations.
Then Cutting Plane proof yields a monotone interpolating [Boolean or arithmetic] circuit $C(\vec{x})$ of almost the same size.

- Exponential lower bound on the interpolating circuit size.

Cutting Planes: exponential lower bound

- Monotone interpolation.

Theorem (Krajiček, Pudlák)

Consider $A(\vec{p}, \vec{q}) \supset B(\vec{p}, \vec{r})$, assume that p_i occur without further negations.
Then Cutting Plane proof yields a monotone interpolating [Boolean or arithmetic] circuit $C(\vec{x})$ of almost the same size.

- Exponential lower bound on the interpolating circuit size.

Theorem (Razborov; Alon-Boppana; Pudlák)

For any Boolean (or arithmetic) circuit separating n–cliques from $(n-1)$–colorable graphs, $|C| = 2^{\Omega(\sqrt{n})}$ where $n = \lfloor \frac{1}{8}(m/\log m)^{2/3} \rfloor$.

Beat Clique Coloring!

Exercise:
extend the system by quadratic inequalities,
give a short proof of "clique coloring"

Frege systems

Definition

Any set of sound (correct) rules

$$\frac{\Phi_1 \quad \Phi_2 \quad \dots \quad \Phi_k}{\Psi},$$

- Φ_i, Ψ are propositional formulas of abstract variables,
- one can substitute any formulas [with certain operations] for abstract variables, variables,
- start with axioms ($k = 0$), derive what you want.

For example,

$$\frac{}{P \supset (Q \supset P)}, \qquad \frac{}{(\neg Q \supset \neg P) \supset ((\neg Q \supset P) \supset Q)},$$

$$\frac{P \qquad P \supset Q}{Q}, \qquad \frac{}{(P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R))}.$$

Frege systems

Equivalence

A Frege system is complete, if for every tautology F, $\exists$ proof of F.

Frege systems

Equivalence

A Frege system is complete, if for every tautology F, we have $\vdash^* F$.
A Frege system is implicationally complete, if $\tilde{\forall}(F \supset G) \implies F \vdash^* G$.

Frege systems

Equivalence

A Frege system is complete, if for every tautology F, we have $\vdash^* F$.
A Frege system is implicationally complete, if $\tilde{\forall}(F \supset G) \implies F \vdash^* G$.

Theorem

All sound and complete, implicationally complete Frege systems p-simulate each other.

Frege systems

Equivalence

A Frege system is complete, if for every tautology F, we have $\vdash^* F$.
A Frege system is implicationally complete, if $\tilde{\forall}(F \supset G) \implies F \vdash^* G$.

Theorem

All sound and complete, implicationally complete Frege systems p-simulate each other.

- Proof for the same operations: simulate each rule.

Extension rule

Introduce a new variable with an axiom: $x \equiv F$.

Frege + extension $\equiv$ Resolution + extension!
(For resolution: axioms $(\overline{x} \vee a_1 \vee \ldots \vee a_k)$ и $(\overline{a_1} \vee x)$, $\ldots$, $(\overline{a_k} \vee x)$.)

Extension rule

Introduce a new variable with an axiom: $x \equiv F$.

Frege + extension $\equiv$ Resolution + extension!
(For resolution: axioms $(\overline{x} \vee a_1 \vee \ldots \vee a_k)$ и $(\overline{a_1} \vee x)$, ..., $(\overline{a_k} \vee x)$.)

Short proof of PHP with extension rule:

prove by induction ($n+1 \to n \to \ldots$) introducing new variables,

the m-th mapping maps m pigeons into $m-1$ holes;

those already sitting there ($j < m$) are untouched;

the hole formerly occupied by the $(m+1)$-st pigeon gets the pigeon from the m-th hole:

Extension rule

Introduce a new variable with an axiom: $x \equiv F$.

Frege + extension $\equiv$ Resolution + extension!
(For resolution: axioms $(\overline{x} \vee a_1 \vee \ldots \vee a_k)$ и $(\overline{a_1} \vee x)$, ..., $(\overline{a_k} \vee x)$.)

<u>Short proof of PHP with extension rule:</u>

prove by induction ($n+1 \to n \to \ldots$) introducing new variables,

the m-th mapping maps m pigeons into $m-1$ holes;

those already sitting there ($j < m$) are untouched;

the hole formerly occupied by the $(m+1)$-st pigeon gets the pigeon from the m-th hole:

$$q_{i,j}^{(m)} \equiv q_{i,j}^{(m+1)} \vee (q_{m+1,j}^{(m+1)} \wedge q_{i,m}^{(m+1)}),$$
$$q_{i,j}^{(n+1)} \equiv p_{i,j}.$$

Derive smaller-PHP clauses for new variables.

Proof systems

Definition (Cook, Reckhow, 70s)

A proof system for L is a polynomial-time surjective $\Pi\colon \{0,1\}^* \to L$.

Definition (almost equivalent)

A proof system for L is a quadratic-time verification procedure V such that

$$F \in L \iff \exists \pi \; V(F, \pi) = 1.$$

Proof systems

Definition (Cook, Reckhow, 70s)

A proof system for L is a polynomial-time surjective $\Pi\colon \{0,1\}^* \to L$.

Definition (almost equivalent)

A proof system for L is a quadratic-time verification procedure V such that

$$F \in L \iff \exists \pi \; V(F, \pi) = 1.$$

Definition (Messner)

An acceptor for L is an algorithm that
accepts every $x \in L$ and does not accept any $x \notin L$.

- Every acceptor yields a proof system (proof: 1^{time}), but not vice versa.
- $\exists$ polynomial-time acceptor for $\texttt{TAUT} \iff \mathbf{P} = \mathbf{NP}$.
- $\exists$ polynomially bounded proof system for $\texttt{TAUT} \iff \mathbf{NP} = \mathbf{co\text{-}NP}$.

Optimal acceptors

Definition

Acceptor S simulates acceptor W if $\exists$ polynomial p such that $\forall x \in L$

$$\mathrm{time}_S(x) \leq p(\mathrm{time}_W(x) + |x|).$$

Optimal acceptor simulates all other acceptors.

Optimal acceptors

Definition

Acceptor S simulates acceptor W if $\exists$ polynomial p such that $\forall x \in L$

$$\mathrm{time}_S(x) \leq p(\mathrm{time}_W(x) + |x|).$$

Optimal acceptor simulates all other acceptors.

Does it exist, e.g., for TAUT?..

Optimal acceptors

Definition

Acceptor S simulates acceptor W if $\exists$ polynomial p such that $\forall x \in L$

$$\mathrm{time}_S(x) \leq p(\mathrm{time}_W(x) + |x|).$$

Optimal acceptor simulates all other acceptors.

Does it exist, e.g., for TAUT?..

Levin's optimal algorithm for SAT as a search problem:
run "in parallel" all possible algorithms outputting satisfying assignments;
check the results and output as soon as a correct one found.

Remark

Levin's algorithm does not give an (optimal) acceptor for TAUT.

Optimal proof systems

Definition

A proof system Σ simulates a proof system Ω iff
Σ-proofs are at most as long as Ω-proofs (up to a polynomial p):

$$\forall F \in L \quad |\text{shortest } \Sigma\text{-proof of } F| \leq p(|\text{shortest } \Omega\text{-proof of } F|).$$

Optimal proof systems

Definition

A proof system Σ simulates a proof system Ω iff
Σ-proofs are at most as long as Ω-proofs (up to a polynomial p):

$$\forall F \in L \quad |\text{shortest } \Sigma\text{-proof of } F| \leq p(|\text{shortest } \Omega\text{-proof of } F|).$$

Definition

p-simulation is a constructive version: For any w-size Ω-proof,
one can compute a $p(w)$-size Σ-proof in polynomial time.

Optimal proof systems

Definition

A proof system Σ simulates a proof system Ω iff
Σ-proofs are at most as long as Ω-proofs (up to a polynomial p):

$$\forall F \in L \quad |\text{shortest } \Sigma\text{-proof of } F| \leq p(|\text{shortest } \Omega\text{-proof of } F|).$$

Definition

p-simulation is a constructive version: For any w-size Ω-proof,
one can compute a $p(w)$-size Σ-proof in polynomial time.

Definition

(p-)optimal proof system (p-)simulates any other proof system.

Optimal proof systems

Definition

A proof system Σ simulates a proof system Ω iff
Σ-proofs are at most as long as Ω-proofs (up to a polynomial p):

$$\forall F \in L \quad |\text{shortest } \Sigma\text{-proof of } F| \leq p(|\text{shortest } \Omega\text{-proof of } F|).$$

Definition

p-simulation is a constructive version: For any w-size Ω-proof,
one can compute a $p(w)$-size Σ-proof in polynomial time.

Definition

(p-)optimal proof system (p-)simulates any other proof system.

Does it exist?..

Optimal acceptors vs p-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For TAUT, $\exists$ *p-optimal proof system iff* $\exists$ *an optimal acceptor.*

Optimal acceptors vs p-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For TAUT, $\exists$ *p-optimal proof system iff* $\exists$ *an optimal acceptor.*

$\Longleftarrow$:

- Now an optimal proof of a size-n tautology includes
  - description of proof system Π;
  - Π-proof of F;

Optimal acceptors vs p-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For TAUT, $\exists$ *p-optimal proof system iff* $\exists$ *an optimal acceptor.*

$\Longleftarrow$:

- Optimal acceptor is polynomial-time on every polynomial-time recognizable set of tautologies.

- Now an optimal proof of a size-n tautology includes
  - description of proof system Π;
  - Π-proof of F;

Optimal acceptors vs p-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For TAUT, $\exists$ *p-optimal proof system iff* $\exists$ *an optimal acceptor.*

$\Longleftarrow$:

- Optimal acceptor is polynomial-time on every polynomial-time recognizable set of tautologies.
- For every proof system Π, one can write in polynomial time the tautology $\mathrm{Con}_{\Pi,n}$ meaning the system is correct for formulas of size n.

- Now an optimal proof of a size-n tautology includes
  - description of proof system Π;
  - Π-proof of F;

Optimal acceptors vs p-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For TAUT, $\exists$ *p-optimal proof system iff* $\exists$ *an optimal acceptor.*

$\Longleftarrow$:

- Optimal acceptor is polynomial-time on every polynomial-time recognizable set of tautologies.
- For every proof system Π, one can write in polynomial time the tautology $\mathrm{Con}_{\Pi,n}$ meaning the system is correct for formulas of size n.
- Thus optimal acceptor is polynomial-time on $\{\mathrm{Con}_{\Pi,n}\}_{n\in\mathbb{N}}$.
- Now an optimal proof of a size-n tautology includes
  - description of proof system Π;
  - Π-proof of F;
  - padding 1^t, where t is the time spent by optimal acceptor on $\mathrm{Con}_{\Pi,n}$.

Optimal acceptors vs p-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For TAUT, $\exists$ *p-optimal proof system iff* $\exists$ *an optimal acceptor.*

$\Longrightarrow$ (for any language, not just TAUT):

- Let Π be a p-optimal proof system.

Optimal acceptors vs p-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For TAUT, $\exists$ *p-optimal proof system iff* $\exists$ *an optimal acceptor.*

$\Longrightarrow$ (for any language, not just TAUT):

- Let Π be a p-optimal proof system.
- Optimal acceptor runs in parallel
  all algorithms B_i trying to produce a Π-proof of F.
- The "proof" is checked by Π. Return 1 if it's valid.

Optimal acceptors vs p-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For TAUT, $\exists$ *p-optimal proof system iff* $\exists$ *an optimal acceptor.*

$\Longrightarrow$ (for any language, not just TAUT):

- Let Π be a p-optimal proof system.
- Optimal acceptor runs in parallel
  all algorithms B_i trying to produce a Π-proof of F.
- The "proof" is checked by Π. Return 1 if it's valid.
- Since Π is p-optimal, for every acceptor A there is a polynomial-time transformation f of its execution into a Π-proof. Thus A together with f are listed in $\{B_i\}_i$.

Optimal proof system with advice

Definition

Proof system with (output non-uniform) advice may use advice string $w_{|x|,|\pi|}$ when verifying proof π for input x.

Optimal proof system with advice

Definition

Proof system with (output non-uniform) advice may use advice string $w_{|x|,|\pi|}$ when verifying proof π for input x.

Theorem (Cook, Krajíček, 07)

For every L, $\exists$ a proof system with 1 bit of advice that simulates any other such system. The simulation can be computed in polynomial time with 1 bit of advice.

Optimal proof for $x \in L$:

- description of proof system Π_i written as 1^i;
- Π_i-proof π of x;
- advice bit b, written as 1^b;
- pairing function must be "length-injective".

Advice bit says whether Π_i with b is correct on all inputs of size $|x|$ and proofs of size $|\pi|$.

Automatizable proof systems

Let $L_\Pi(x)$ be the size of the shortest Π-proof.
Automatizable proof system: has an automatizer A working in output-polynomial-time.

- $A(x)$ is a Π-proof of size polynomial in $L_\Pi(x)$.
  - (recall) Π accepts proofs with probability $> 1/2$;
  - (recall) Π accepts non-proofs of wrong theorems with probability $\leq 1/8$;
  - "almost" proof is what is accepted with probability $> 1/4$.

Fact

For every automatizable proof system Π there is an acceptor with time polynomial in $L_\Pi(x)$, and vice versa.

Disjoint NP pairs

- Just a pair (A, B) of two disjoint sets $A, B \in \mathbf{NP}$.
- The problem is to separate A from B: given x, decide between the two alternatives $x \in A$ vs $x \in B$ (if it is outside both, say anything).

Disjoint NP pairs

- Just a pair (A, B) of two disjoint sets $A, B \in \mathbf{NP}$.
- The problem is to separate A from B: given x, decide between the two alternatives $x \in A$ vs $x \in B$ (if it is outside both, say anything).
- Reduction $(A, B) \to (C, D)$:
  polynomial-time f such that $f(A) \subseteq C$, $f(B) \subseteq D$.
- Are there complete ones? Unknown.

Disjoint NP pairs

- Just a pair (A, B) of two disjoint sets $A, B \in \mathbf{NP}$.
- The problem is to separate A from B: given x, decide between the two alternatives $x \in A$ vs $x \in B$ (if it is outside both, say anything).
- Reduction $(A, B) \to (C, D)$:
  polynomial-time f such that $f(A) \subseteq C$, $f(B) \subseteq D$.
- Are there complete ones? Unknown.

Example (Razborov, 1994: Canonical NP pair for proof system Π for $\mathtt{TAUT}$)

$\overline{\mathtt{TAUT}}_* = \{(F, 1^t) \mid F \in \overline{\mathtt{TAUT}}\}$,
$\mathtt{REF}_\Pi = \{(F, 1^t) \mid F \in \mathtt{TAUT},\ \text{there is a } \Pi\text{-proof of } F \text{ of size} \leq t\}$.
Separation gives automatization (of a possibly stronger system)!

Disjoint NP pairs

- Just a pair (A, B) of two disjoint sets $A, B \in \mathbf{NP}$.
- The problem is to separate A from B: given x, decide between the two alternatives $x \in A$ vs $x \in B$ (if it is outside both, say anything).
- Reduction $(A, B) \to (C, D)$:
  polynomial-time f such that $f(A) \subseteq C$, $f(B) \subseteq D$.
- Are there complete ones? Unknown.

Example (Razborov, 1994: Canonical NP pair for proof system Π for $\mathtt{TAUT}$)

$\overline{\mathtt{TAUT}}_* = \{(F, 1^t) \mid F \in \overline{\mathtt{TAUT}}\}$,
$\mathtt{REF}_\Pi = \{(F, 1^t) \mid F \in \mathtt{TAUT},\ \text{there is a } \Pi\text{-proof of } F \text{ of size} \leq t\}$.
Separation gives automatization (of a possibly stronger system)!

Theorem

If Ω simulates Σ, then $(\overline{\mathtt{TAUT}}_*, \mathtt{REF}_\Omega) \to (\overline{\mathtt{TAUT}}_*, \mathtt{REF}_\Sigma)$.
Optimal proof system yields complete **NP** *pair.*

To be continued. . .

References

Old surveys (useful for fast introduction to the area):

[1] Samuel R. Buss: Propositional Proof Complexity: An Introduction. In Computational Logic, edited by U. Berger and H. Schwichtenberg, Springer-Verlag, Berlin, 1999, pp. 127-178. http://www.math.ucsd.edu/~sbuss/ResearchWeb/marktoberdorf97/index.html

[2] Pavel Pudlák: On the complexity of propositional calculus. Sets and Proofs, Invited papers from Logic Colloquium'97, S. Barry Cooper and John K. Truss eds., Cambridge Univ. Press 1999, pp. 197-218. http://www.math.cas.cz/~pudlak/oncompl.ps

[3] Alasdair Urquhart: The Complexity of Propositional Proofs. Bulletin of Symbolic Logic 1(4):425-467, 1995. https://doi.org/10.2307/421131

Book:

[4] Jan Krajíček: Proof complexity. Encyclopedia of Mathematics and Its Appplications, Vol.170, Cambridge University Press, Cambridge - New York - Melbourne, 2019 (in press)
https://www.karlin.mff.cuni.cz/~krajicek/prfdraft.html