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?

  - 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:
  - Size $2^n$ is trivial, interested in shorter proofs.
What is a (complete) mathematical proof?

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

- A text.
  - Not an interactive procedure.

- Can be verified.
  - Verification does not need creative work.
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.*
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:
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.
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:

  \[(x_1 \land x_2) \lor \overline{x_1} \lor \overline{x_2}\]
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:

  \[(x_1 \& x_2) \lor \neg x_1 \lor \neg x_2\]
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:
  \[(x_1 \land x_2) \lor \overline{x_1} \lor \overline{x_2}\]
- Size $2^n$ is trivial, interested in shorter proofs.
Proof by contradiction:
instead of deriving
\[ F = (x_1 \land x_2) \lor \overline{x_1} \lor \overline{x_2} \]
we can refute
\[ \overline{F} = (\overline{x_1} \lor \overline{x_2}) \land x_1 \land x_2 \]
What do we prove

- Proof by contradiction:
  instead of deriving

  \[ F = (x_1 \land x_2) \lor \overline{x_1} \lor \overline{x_2} \]

  we can refute

  \[ \overline{F} = (\overline{x_1} \lor \overline{x_2}) \land x_1 \land x_2 \]

- W.l.o.g.: Conjunctive Normal Form

  \[ \bigwedge_{j=1}^{t} (\ell_{j1} \lor \ldots \lor \ell_{jk}), \quad \text{(where } \ell_{jt} = x_i \text{ or } = \overline{x_i}) \]

  Tseytin’s translation:

  \[ F = G \lor H \quad \iff \quad (\overline{x_F} \lor x_G \lor x_H) \land (\overline{x_G} \lor x_F) \land (\overline{x_H} \lor x_F) \]

  We work with “clauses”!

  \[ x \lor \overline{y} \lor \overline{z} \]
What do we prove

- Proof by contradiction:
  instead of deriving
  \[ F = (x_1 \land x_2) \lor \overline{x_1} \lor \overline{x_2} \]
  we can refute
  \[ \overline{F} = (\overline{x_1} \lor \overline{x_2}) \land x_1 \land x_2 \]

- Useful tautologies:

  Propositional Pigeonhole Principle:
  \[ x_{ij} \sim i\text{-th pigeon in } j\text{-th hole} \]
  \[ i = 1, \ldots, n + 1 \]
  \[ j = 1, \ldots, n \]
  Pigeon \( i \) sits somewhere:
  \[ x_{i1} \lor x_{i2} \lor \ldots \lor x_{in} \]
  Two pigeons \( i \neq i' \) cannot share a hole:
  \[ \overline{x_{ij}} \lor \overline{x_{i'j}} \]
What do we prove

- Proof by contradiction:
  instead of deriving
  \[ F = (x_1 \land x_2) \lor \overline{x_1} \lor \overline{x_2} \]
  we can refute
  \[ \overline{F} = (\overline{x_1} \lor \overline{x_2}) \land x_1 \land x_2 \]

- 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:

\[
\begin{array}{c}
F_1, \ldots, F_s \\
G
\end{array}
\]

if \( F_i \)'s semantically imply \( G \).

For example, Modus Ponens:

\[
\begin{array}{c}
F, \quad F \Rightarrow H \\
H
\end{array}
\]
How do we prove
Logic-like systems

Logical rule:

\[
\begin{array}{c}
F_1, \ldots, F_s \\
\hline
G
\end{array}
\]

if \( F_i \)'s semantically imply \( G \).

For example, Modus Ponens:

\[
\begin{array}{c}
F, \quad F \supset H \\
\hline
H
\end{array}
\]
Resolution

\[ \frac{x \lor \ell_1 \lor \ldots \lor \ell_k, \quad \overline{x} \lor m_1 \lor \ldots \lor m_t}{\ell_1 \lor \ldots \lor \ell_k \lor m_1 \lor \ldots \lor m_t} \]

(Resolution)

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

For example,

\[ \frac{x \lor y \lor \overline{z}, \quad z \lor y}{x \lor y} \]
Resolution

\[
\frac{x \lor C, \quad \overline{x} \lor D}{C \lor D} \quad \text{(Resolution)}
\]

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

For example,

\[
\frac{x \lor y \lor \overline{z}, \quad z \lor y}{x \lor y}
\]
Resolution

\[
\frac{C}{C \lor x} \quad \text{(Weakening)}
\]

\[
\frac{x \lor C, \; \overline{x} \lor D}{C \lor D} \quad \text{(Resolution)}
\]

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

For example,

\[
\frac{x \lor y \lor \overline{z}, \; z \lor y}{x \lor y}
\]
How do we prove
Non-logic-like systems

Formulate disjunctions as statements about integers:

\[ x_1 \lor x_2 \lor \overline{x_3} \]

\[ x_1 + x_2 + (1 - x_3) \geq 1 \]

\[ (1 - x_1)(1 - x_2)x_3 = 0 \]

What can we do?

- make linear combinations,
- multiply by appropriate constants (or polynomials),
- \ldots and more.
How do we prove
Non-logic-like systems

Formulate disjunctions as statements about integers:

\[ x_1 \lor x_2 \lor \overline{x_3} \]

\[ x_1 + x_2 + (1 - x_3) \geq 1 \]

\[ (1 - x_1)(1 - x_2)x_3 = 0 \]

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

\[ f : \{0, 1\}^* \rightarrow L \]
Propositional proof system

Formal definition

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

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

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

Formal definition

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

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

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

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

  $\forall F \in L \ \exists \pi \ f(\pi) = F \text{ and } \text{length}(\pi) \leq p(\text{length}(F))$

  $\pi$ is a proof
Propositional proof system

Formal definition

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

  \[ f : \{0, 1\}^* \rightarrow L \]

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

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

  \[ \forall F \in L \exists \pi \ f(\pi) = F \text{ and } \text{length}(\pi) \leq p(\text{length}(F)) \]

  - $\pi$ is a proof

- Polynomaically bounded proof system for propositional tautologies $\Leftrightarrow NP = co-NP$. 
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)
▶ Width of clause: the number of literals $\ell_1 \lor \ldots \lor \ell_k$.
▶ 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 $\sqrt{n \log S} + \text{width}(F)$,
   ▶ conclude there are exponential-size proofs only.
▶ Similar framework for algebraic systems (degree lower bounds).
Notation:

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

axioms $\vdash_w$ clause: derivation of width $w$
Lemma 1: $F|_{\ell=1} \vdash^w C \implies F \vdash_{w+1} C \lor \bar{\ell}$

Lemma 2:

$$\begin{align*}
F|_{\ell=1} \vdash^w_1 \text{False} & \implies F \text{ has refutation of width } \max(w, F) \\
F|_{\ell=0} \vdash^w \text{False} & \implies F \text{ has refutation of width } \max(w, F)
\end{align*}$$

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

Proof: induction: split by $\ell$ killing many wide clauses.
Complexity measure $\mu: \{\text{clauses}\} \rightarrow \mathbb{R}_{\geq 0}$

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

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

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

hence there is a clause $C^*$ such that

$$\frac{\mu(C^*)}{\mu(False)} \in [1/3 .. 2/3]$$
There is a clause $C^*$ such that

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

- Set $S$ of axioms implying $C^*$.
- Boundary: includes variables $x$ such that there is an assignment $\alpha$ 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 $\ell_1 \lor \ell_2 \lor \ldots \lor \ell_k \mapsto$ inequality $\ell_1 + \ell_2 + \ldots + \ell_k \geq 1$.

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

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

\[
\frac{A \geq a, \quad B \geq b}{\alpha A + \beta B \geq \alpha a + \beta b} \quad (\alpha, \beta > 0) \quad \text{(Linear combination)}
\]
More examples: Cutting Planes [Gomory, Chvátal]

Clause \( \ell_1 \lor \ell_2 \lor \ldots \lor \ell_k \mapsto \) inequality \( \ell_1 + \ell_2 + \ldots + \ell_k \geq 1 \).

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

Additional axioms: \( x \geq 0, -x \geq -1 \).

\[
\begin{align*}
\frac{A \geq a}{A \geq \lceil a \rceil} & \quad \text{(Rounding)} \\
\frac{A \geq a, \quad B \geq b}{\alpha A + \beta B \geq \alpha a + \beta b} & \quad (\alpha, \beta > 0) \quad \text{(Linear combination)}
\end{align*}
\]
More examples: Cutting Planes [Gomory, Chvátal]

Clause $\ell_1 \lor \ell_2 \lor \ldots \lor \ell_k \iff$ inequality $\ell_1 + \ell_2 + \ldots + \ell_k \geq 1$.

$x \lor \overline{y} \iff x + (1 - y) \geq 1 \iff x - y \geq 0$

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

\[
\begin{align*}
\frac{A \geq a}{A \geq \lceil a \rceil} & \quad \text{(Rounding)} \\
\frac{A \geq a, \, B \geq b}{\alpha A + \beta B \geq \alpha a + \beta b} & \quad (\alpha, \beta > 0) \quad \text{(Linear combination)}
\end{align*}
\]

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 $\lor$ $G$ is not $(n - 1)$-colorable,
Clique-coloring tautologies

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

i.e., $\nexists$ two graph homomorphisms $K_n \xrightarrow{q} G \xrightarrow{r} K_{n-1}$.
Either $G$ contains no $n$–clique $\lor$ $G$ is not $(n – 1)$-colorable,

i.e., $\forall$ 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: $q_{ki} + q_{k'j} \leq p_{ij} + 1 \quad (k \neq k', \; i < j)$.
- Every vertex has a color: $\sum_{\ell=1}^{m-1} r_{i\ell} \geq 1$.
- The coloring is correct: $p_{ij} + r_{i\ell} + r_{j\ell} \leq 2 \quad (i < j)$. 
Clique-coloring tautologies

Either $G$ contains no $n$–clique $\lor$ $G$ is not $(n - 1)$-colorable, i.e., $\not\exists$ two graph homomorphisms $K_n \xrightarrow{q} G \xrightarrow{r} K_{n-1}$.

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

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

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: \( \text{Clique}(\vec{p}, \vec{q}) \supset \text{Coloring}(\vec{p}, \vec{r}) \)
Craig’s interpolation theorem
The propositional case

**Theorem**

If $A(\vec{x}, \vec{y}) \land \overline{B}(\vec{x}, \vec{z})$ 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: $\text{Clique}(\vec{p}, \vec{q}) \land \text{Coloring}(\vec{p}, \vec{r})$
Monotone interpolation.

Exponential lower bound on the interpolating circuit size.
Cutting Planes: exponential lower bound

- Monotone interpolation.

**Theorem (Krajíček, 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.
Monotone interpolation.

**Theorem (Krajíček, 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 $n = \lceil \frac{1}{8} (m/ \log m)^{2/3} \rceil$. 
Exercise:
extend the system by quadratic inequalities,
give a short proof of “clique coloring”
Frege systems

Definition

Any set of sound (correct) rules

\[
\Phi_1 \quad \Phi_2 \quad \ldots \quad \Phi_k \quad \Psi
\]

- \(\Phi_i, \Psi\) 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,

\[
P \supset (Q \supset P)
\]

\[
(\neg Q \supset \neg P) \supset ((\neg Q \supset P) \supset Q)
\]

\[
P \quad P \supset Q
\]

\[
Q
\]

\[
(P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R))
\]
A Frege system is complete, if for every tautology $F$, $\exists$ proof of $F$. 
A Frege system is complete, if for every tautology $F$, we have $\vdash * F$.
A Frege system is implicationally complete, if $\forall (F \supset G) \implies F \vdash * G$. 
A Frege system is **complete**, if for every tautology $F$, we have $\vdash^* F$.

A Frege system is **implicationally complete**, if $\forall (F \supset G) \implies F \vdash^* G$.

**Theorem**

*All sound and complete, implicationally complete Frege systems p-simulate each other.*
A Frege system is **complete**, if for every tautology $F$, we have $\vdash^* F$. 
A Frege system is **implicationally complete**, if $\forall (F \supset G) \implies F \vdash^* G$.

**Theorem**

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

- Proof for the same operations: simulate each rule.
Introduce a new variable with an axiom: $x \equiv F$.

Frege + extension $\equiv$ Resolution + extension!
(For resolution: axioms $(\overline{x} \lor a_1 \lor \ldots \lor a_k) \cup (a_1 \lor x), \ldots, (a_k \lor x)$.)
Introduce a new variable with an axiom: \( x \equiv F \).

Frege + extension \( \equiv \) Resolution + extension!
(For resolution: axioms \((\neg x \lor a_1 \lor \ldots \lor a_k) \land (\neg a_1 \lor x), \ldots, (\neg a_k \lor x)\).)

Short proof of PHP with extension rule:
prove by induction \((n + 1 \rightarrow n \rightarrow \ldots)\) 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 $(\overline{x} \lor a_1 \lor \ldots \lor a_k) \land (\overline{a_1} \lor x), \ldots, (\overline{a_k} \lor x)$.)

Short proof of PHP with extension rule:
prove by induction ($n + 1 \rightarrow n \rightarrow \ldots$) 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:

$$q_{i,j}^{(m)} \equiv q_{i,j}^{(m+1)} \lor (q_{m+1,j}^{(m+1)} \land q_{i,m}^{(m+1)}),$$

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

Derive smaller-PHP clauses for new variables.
### Proof systems

#### Definition (Cook, Reckhow, 70s)
A **proof system** for $L$ is a polynomial-time surjective $\Pi : \{0, 1\}^* \rightarrow L$.

#### 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**

<table>
<thead>
<tr>
<th>Definition (Cook, Reckhow, 70s)</th>
<th>A proof system for $L$ is a polynomial-time surjective $\Pi \colon {0, 1}^* \to L$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition (almost equivalent)</td>
<td>A proof system for $L$ is a quadratic-time verification procedure $V$ such that $F \in L \iff \exists \pi \ V(F, \pi) = 1$.</td>
</tr>
<tr>
<td>Definition (Messner)</td>
<td>An acceptor for $L$ is an algorithm that accepts every $x \in L$ and does not accept any $x \notin L$.</td>
</tr>
</tbody>
</table>

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

**Definition**

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

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

Optimal acceptor simulates all other acceptors.
Optimal acceptors

Definition

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

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

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 $\forall x \in L$

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

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 $\Sigma$ simulates a proof system $\Omega$ iff $\Sigma$-proofs are at most as long as $\Omega$-proofs (up to a polynomial $p$):

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

**Definition**

A proof system $\Sigma$ simulates a proof system $\Omega$ iff $\Sigma$-proofs are at most as long as $\Omega$-proofs (up to a polynomial $p$):

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

**Definition**

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

A proof system $\Sigma$ simulates a proof system $\Omega$ iff $\Sigma$-proofs are at most as long as $\Omega$-proofs (up to a polynomial $p$):

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

Definition

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

Definition

$(p)$-optimal proof system $(p)$-simulates any other proof system.
Optimal proof systems

Definition
A proof system $\Sigma$ simulates a proof system $\Omega$ iff $\Sigma$-proofs are at most as long as $\Omega$-proofs (up to a polynomial $p$):

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

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

Definition
$(p)$optimal proof system $(p)$simulates any other proof system.

Does it exist?..
Theorem (Krajíček, Pudlák, 89)

For $\text{TAUT}$, $\exists \ p$-optimal proof system iff $\exists \ an \ optimal \ acceptor.$
Optimal acceptors vs $p$-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For $\text{TAUT}$, $\exists$ $p$-optimal proof system iff $\exists$ an optimal acceptor.

$\Leftarrow$:

- Now an optimal proof of a size-$n$ tautology includes
  - description of proof system $\Pi$;
  - $\Pi$-proof of $F$;
Theorem (Krajíček, Pudlák, 89)

*For* \( \text{TAUT} \), \( \exists \) \( p \)-optimal proof system iff \( \exists \) an optimal acceptor.

\( \Leftarrow \):

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

\( \Rightarrow \):

- Now an optimal proof of a size-\( n \) tautology includes
  - description of proof system \( \Pi \);
  - \( \Pi \)-proof of \( F \);
Optimal acceptors vs $p$-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For $\text{TAUT}$, $\exists$ $p$-optimal proof system iff $\exists$ an optimal acceptor.

$\Leftarrow$:

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

- Now an optimal proof of a size-$n$ tautology includes
  - description of proof system $\Pi$;
  - $\Pi$-proof of $F$;
Optimal acceptors vs $p$-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For $\text{TAUT}$, $\exists$ $p$-optimal proof system iff $\exists$ an optimal acceptor.

$\Leftarrow$: 

- Optimal acceptor is polynomial-time on every polynomial-time recognizable set of tautologies.
- For every proof system $\Pi$, one can write in polynomial time the tautology $\text{Con}_\Pi,n$ meaning the system is correct for formulas of size $n$.
- Thus optimal acceptor is polynomial-time on $\{\text{Con}_\Pi,n\}_{n \in \mathbb{N}}$.
- Now an optimal proof of a size-$n$ tautology includes
  - description of proof system $\Pi$;
  - $\Pi$-proof of $F$;
  - padding $1^t$, where $t$ is the time spent by optimal acceptor on $\text{Con}_\Pi,n$. 
Theorem (Krajíček, Pudlák, 89)

For \( \text{TAUT} \), \( \exists \) \( p \)-optimal proof system iff \( \exists \) an optimal acceptor.

\[\implies\] (for any language, not just \( \text{TAUT} \)):

- Let \( \Pi \) be a \( p \)-optimal proof system.
Optimal acceptors vs $p$-Optimal proof systems

Theorem (Krajíček, Pudlák, 89)

For $\text{TAUT}$, $\exists$ $p$-optimal proof system iff $\exists$ an optimal acceptor.

$\implies$ (for any language, not just $\text{TAUT}$):

- Let $\Pi$ be a $p$-optimal proof system.
- Optimal acceptor runs in parallel all algorithms $B_i$ trying to produce a $\Pi$-proof of $F$.
- The “proof” is checked by $\Pi$. Return 1 if it’s valid.
Optimal acceptors vs \( p \)-Optimal proof systems

**Theorem (Krajíček, Pudlák, 89)**

For \( \text{TAUT} \), \( \exists p \)-optimal proof system iff \( \exists \) an optimal acceptor.

\[ \implies \text{(for any language, not just } \text{TAUT}): \]
- Let \( \Pi \) be a \( p \)-optimal proof system.
- Optimal acceptor runs in parallel all algorithms \( B_i \) trying to produce a \( \Pi \)-proof of \( F \).
- The “proof” is checked by \( \Pi \). Return 1 if it’s valid.
- Since \( \Pi \) is \( p \)-optimal, for every acceptor \( A \) there is a polynomial-time transformation \( f \) of its execution into a \( \Pi \)-proof. Thus \( A \) together with \( f \) are listed in \( \{B_i\}_i \).
Definition

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

Proof system with (output non-uniform) advice may use advice string $w_{|x|,|\pi|}$ when verifying proof $\pi$ 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 $\Pi_i$ written as $1^i$;
- $\Pi_i$-proof $\pi$ of $x$;
- advice bit $b$, written as $1^b$;
- pairing function must be “length-injective”.

Advice bit says whether $\Pi_i$ with $b$ is correct on all inputs of size $|x|$ and proofs of size $|\pi|$.
Let $L_\Pi(x)$ be the size of the shortest $\Pi$-proof.

Automatizable proof system: has an automatizer $A$ working in output-polynomial-time.

- $A(x)$ is a $\Pi$-proof of size polynomial in $L_\Pi(x)$.
  - (recall) $\Pi$ accepts proofs with probability $> 1/2$;
  - (recall) $\Pi$ 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 $\Pi$ 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 \(A, B \in \text{NP}\).
- 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 \(A, B \in \text{NP}\).
- 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 \((A, B) \rightarrow (C, D)\):
  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 \(A, B \in \text{NP}\).
- 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 \((A, B) \rightarrow (C, D)\):
  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 \(\Pi\) for \(\text{TAUT}\))

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

- Just a pair \((A, B)\) of two disjoint sets \(A, B \in \text{NP}\).
- 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 \((A, B) \rightarrow (C, D)\): 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 \(\Pi\) for \(\text{TAUT}\))

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

Theorem

If \(\Omega\) simulates \(\Sigma\), then \((\overline{\text{TAUT}}_*, \text{REF}_\Omega) \rightarrow (\overline{\text{TAUT}}_*, \text{REF}_\Sigma)\).
Optimal proof system yields complete \(\text{NP}\) pair.
To be continued...
Old surveys (useful for fast introduction to the area):


Book: