Propositional Proof Systems and Search

Sam Buss
Mathematics, U.C. San Diego

Connections II: Fundamentals of Network Science
Pasadena, CA
August 2006

to the memory of

Misha Alekhnovich

Propositional Logic

Boolean connectives: $\wedge$ (AND); $\vee$ (OR); $\neg$ (NOT); etc.

Boolean variables: $p, q, r, \ldots, x, y, z, \ldots$, range over $\{T, F\}$.

A *tautology* is a valid (=always true) formula.

A formula is *satisfiable* iff exists an assignment making it true.

Formula ϕ is a tautology iff $\neg\phi$ is not satisfiable.

Algorithmic goals: Given formula ϕ:

- Determine if ϕ is a satisfiable. If so, find a satisfying assignment.
- Determine if ϕ is a tautology. If so, find a formal proof.

NP-hard!

Combinatorial principles as tautologies.

Example 1: The pigeonhole principle. For $n \geq 1, 0 \leq i \leq n$ and $0 \leq j < n$,

$$\neg \left[\bigwedge_i \bigvee_j p_{i,j} \ \wedge \ \bigwedge_{i<i'} \bigwedge_j \neg(p_{i,j} \wedge p_{i',j}) \right].$$

Example 2: Counting principles. Fix $k > 1$. For $n \not\equiv 0 \pmod{k}$,
"n cannot be partitioned into sets of size k"

$$\neg \left[\bigwedge_i \bigvee_{i \in A} p_A \ \wedge \ \bigwedge_{\substack{A \neq B \\ A \cap B \neq \emptyset}} \neg(p_A \wedge p_B) \right].$$

where $A, B \in [n]^k$.

Example 3: Primality tautologies. Fix $a > 1$ a prime number. Let a have n bits in its binary representation. $a = (a_{n-1}, \ldots, a_0)_2$.
Let $p = (p_{n-1}, \ldots, p_0)_2$ and $q = (q_{n-1}, \ldots, q_0)_2$ be unspecified integers.

$$\neg \left[Product(\vec{p}, \vec{q}, \vec{a}) \wedge (\bigvee_{i>1} p_i) \wedge (\bigvee_{i>1} q_i) \right].$$

The variables are $\vec{p}$, $\vec{q}$. The bits $\vec{a}$ are constants.
Here "$Product$" expresses base-2 multiplication.

Taking a a composite, say product of two $n/2$ bit integers, then the negation of the formula is satisfiable. This allows us to generate formulas that are easy to recognize as satisfiable, but are conjectured to be hard to find satisfying assignments for.

P and NP

Def'n: P is the class of predicates (decision problems) that are decidable in polynomial time in the length n of the input.

NP is the class of predicates for which "Yes" answers are verifiable in polynomial time.

Examples in P. Given integer x, is x perfect square? Given integers x and y, is the middle bit of the product $x \cdot y$ a '1'?

Examples in NP: Is x a composite?

Much less obviously, is x a prime? [Pratt, 1975].
Very much less obviously, the set of primes is in P. [Agarwal et al., 2002].

Open Conjecture: Integer Factorization is not in P.
This conjecture and related ones are the basis of the theory of public key cryptography.

Examples of NP problems

SAT (Satisfiability): Given a propositional formula, over connectives AND ($\wedge$), OR ($\vee$), NOT ($\neg$), and with variables $p, q, r, \ldots$, does it have a satisfying assignment? That is, can the variables be set so as to make the formula true?

Hamiltonian cycle. Given graph G, does it have a Hamiltonian cycle?

k-**Provability.** Given a formula (a theorem), does it have a proof of $\leq k$ symbols in some given formal proof system? [Alekhnovich-Buss-Pitassi-Moran]

All three of these are NP-complete (the third at least for certain proof systems).

Def'n A problem is *co-*NP if its complement is NP. I.e., its "No" answers are verifiable in polynomial time.

Example 1: The set of primes is obviously in co-NP.

Example 2: The set of tautologies is co-NP-complete.

Note that φ is a tautology iff $\neg\varphi \in \mathrm{SAT}$.

Methods for *proving* tautologies: (a) Method of truth tables, (b) decision trees, (c) Give a proof in a formal system (e.g., a Frege system). (d) Etc.

(a),(b) require exponential size proofs. For (c), it open, firstly, whether polynomial size proofs exist and, secondly, whether proofs can be found efficiently.

Cook's Program for the P-NP problem

Def'n. A *Frege proof* system, $\mathcal{F}$, is a proof system for propositional logic, say over $\wedge$, $\vee$, $\neg$, $\rightarrow$, based on a finite set of axiom schemes such as $A \rightarrow (B \rightarrow A)$, and based on a finite set of inference rules, such as modus ponens:

$$\frac{A \qquad A \rightarrow B}{B}$$

A Frege system is sound and complete (implicationally).

These are the usual "textbook" proof systems.

Open: Find good upper bounds on the lengths of tautologies. I.e., find slow-growing function f such that every tautology of length n, has an $\mathcal{F}$-proof of $\leq f(n)$ symbols.

$f(n) = 2^{O(n)}$ suffices. Can $f(n)$ be polynomial, $n^{O(1)}$?

Thm: [Cook'75]. If Frege proof lengths can be polynomially bounded, then $NP =$ co-NP

Pf. The tautologies are co-NP-complete. If they have polynomial size $\mathcal{F}$-proofs, they would be in NP (by simply guessing the proof). From this $NP =$ co-NP would follow. q.e.d.

Cook's program Starting with weak proof systems for propositional logic, prove superpolynomial lower bounds on the size of proofs. Work up to superpolynomial lower bounds on stronger systems such as Frege systems, eventually to all proof systems. This would prove $NP \neq$ co-NP, hence $P \neq NP$.

This has been carried out only for restricted proof systems.

Cook's definition of a proof system:
Any polynomial time function with range equal to the set of tautologies.

Some Complete Proof Systems

Truth tables	–
Resolution	Clauses
Cutting planes	Linear integer inequalities
Nullstellensatz	Single polynomial identity
Gröbner basis / Polynomial calculus	Field polynomial
Constant-depth (cd) Frege	cd poly size formulas
cd-Frege with counting axioms	cd poly size formulas
.....	.....
cd-Frege with counting gates	Poly size fmlas with mod gates
Frege systems	Poly size formulas
Extended Frege systems	Poly size circuits
Quantified Frege systems	Polynomial space predicates
Set theory	–

Superpolynomial lower bounds are known for systems above the dotted line.

Extended Frege systems

[Tseitin] An extended Frege system is a Frege system augmented with the ability to introduce abbreviations on the fly with the *extension rule:*

$$q \leftrightarrow A$$

where A is any formula, and q must be a "new" variable that does appear yet in the proof, in A, or in the formula to be proved. This introduces q as an abbreviation for A.

Introducing abbreviations allows proof length to be shorter (well, this is open), since long formulas can be replaced by abbreviations.

Equivalently: extended Frege systems are Frege systems that use Boolean circuits instead of formulas.
Also equivalent: Extended Frege systems are Frege systems with proof length measured in terms of the number of inferences in the proof. [Statman]

The pigeonhole principle (PHP) as a tautology

Let $[n] = \{0, \ldots, n\}$.

The PHP states there is no 1-1 function $f : [n] \to [n-1]$. To encode this as a tautology, use propositional variables $p_{i,j}$ which express the truth of $f(i) = j$.

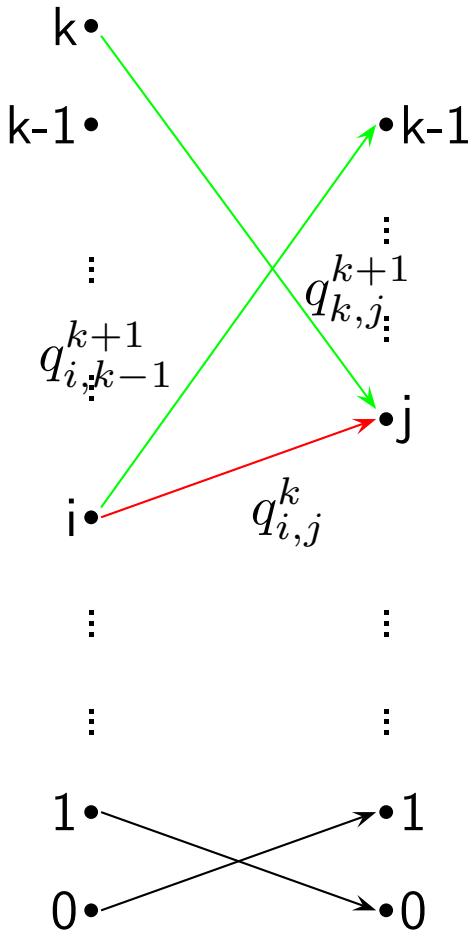
The PHP_n^{n+1} tautology is:

$$\neg \left[\bigwedge_i \bigvee_j p_{i,j} \ \wedge \ \bigwedge_{i \neq i'} \bigwedge_j \neg (p_{i,j} \wedge p_{i',j}) \right].$$

Let's give a proof of this by contradiction ....

Phase 1: Define variables $q^k_{i,j}$ that define a violation of PHP^{k+1}_k. For this, let $q^n_{i,j} = p_{i,j}$ and define

$$q^k_{i,j} \leftrightarrow q^{k+1}_{i,j} \vee (q^{k+1}_{i,k-1} \wedge q^{k+1}_{k,j}).$$



Phase 2: Prove that if q^{k+1}'s violate the PHP, then so do the q^k's.

Phase 3: Proof is done, the PHP^{n+1}_n implies PHP^2_1, which is impossible.

QED

This gives poly size extended Frege proof, but not poly size Frege proof.

Thm. [Buss] The PHP_n^{n+1} tautologies have polynomial size Frege proofs.

Thm. [Haken] The PHP_n^{n+1} tautologies require exponential size resolution proofs. (Actually, refutations.)

Thm. [Chvátal-Szemerédi] Suitably chosen random set of clauses require exponential size resolution refutations.

Def'n The *depth* of a propositional formula is the number of alternations of $\wedge$'s and $\vee$'s. For this count, implications are replaced and negations are pushed to the variables.

A constant depth Frege proof is a (family) of proofs in which the depth of formulas are bounded by a constant.

Thm. [Pitassi-Beame-Impagliazzo, Krajíček-Pudlák-Woods] The PHP_n^{n+1} tautologies require exponential size constant-depth Frege proofs.

Proof used an extension of the Hastad switching lemma.

Algebraic Proof Systems - General formulation

Fix a ring R — usually R is $\mathbb{Z}$ or a field F.

Define a translation (*) from propositional logic to (in)equalities over R. Usually the translation is linear or multilinear.

The elements 1 and 0 correspond to *True* and *False*.

$(\mathit{True})^* = 1$ and $(\mathit{False})^* = 0$

$(\neg\phi)^* = 1 - \phi^*$.

$(\phi \wedge \psi)^* = (\phi^*)(\psi^*)$ (over fields).

Etc.

Details vary based on the proof system....

Cutting Planes Systems - Linear Inequalities over $\mathbb{Z}$

[Gromov, Chvátal, ...] Reason with inequalities of the form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b,$$

for $a_1, \ldots, a_n, b \in \mathbb{Z}$. Identify *True* with 1 and 0 with *False*.

Example: the clause $p \vee \neg q \vee r \vee \neg s$ is translated to

$$p + (1 - q) + r + (1 - s) \geq 1.$$

i.e., to $$1 \cdot p + (-1) \cdot q + 1 \cdot r + (-1) \cdot s \geq -1.$$

Also add, for all variables, the inequalities

$$x_i \geq 0 \quad \text{and} \quad (-1)x_i \geq -1.$$

Cutting Planes - cont'd

Rules of inference are:

a. Addition rule, and

b. Division rule. If $a_1, \ldots, a_n$ are multiples of $d > 0$,

$$\frac{\sum_i a_i x_i \geq b}{\sum_i (a_i/d) x_i \geq \lceil b/d \rceil}$$

Justification: x_i's are integer valued (0,1 valued).

A cutting planes proof consists of a derivation of $0 \geq 1$ from a initial set of clauses. Thus it is a refutation system (like resolution) in that it proves the initial set of clauses is unsatisfiable.

Cutting planes is sound and complete.

Polynomial Calculus & Nullstellensatz systems

[Pudlák, Clegg-Edmonds-Impagliazzo]

Work in a field F, usually $F = \mathbb{R}$ or a finite field, e.g., $\mathbb{Z}_p^*$.

Let $f_1(\vec{x}), \ldots, f_k(\vec{x})$ be polynomials.
The goal is to show that there is *no* solution $\vec{x}$ s.t. $f_i(\vec{x}) = 0$, $\forall i$.

Auxiliary polynomials: $f_{k+i}(\vec{x}) := x_i^2 - x_i$.

Nullstellensatz and the polynomial calculus are *refutation* systems. They try to prove that $1 = 0$ follows from the equations $f_i(\vec{x}) = 0$.

Defn: A **Nullstellsatz refutation** consists of $g_1(\vec{x}), \ldots g_{k+n}(\vec{x})$ such that

$$g_1(\vec{x})f_1(\vec{x}) + \cdots + g_{k+n}(\vec{x})f_{k+n}(\vec{x}) = 1. \qquad (1)$$

Defn: A **polynomial calculus** refutation is a sequence of lines of the form $h(\vec{x}) = 0$ where h is a polynomial (essentially all multilinear), where the rules of inference are:

Axioms: $f_i(\vec{x}) = 0$

Rule of inference: $\dfrac{h_1 = 0 \quad h_2 = 0}{k_1 h_1 + k_2 h_2 = 0}$ where k_1, k_2 are polynomials,

The last line of the refutation must be $1 = 0$.

Thm The Nullstellensatz system and the polynomial calculus are sound and complete.

Defn: The *size* of a refutation is the number of bits needed to write out its description. Sometimes the number of lines or the number of occurrences of monomials is used instead.

More commonly, the **degree** of the polynomials is used instead. In many cases, lower bounds on the degree of refutations translates into lower bounds on the size of refutations.

E.g., Constant degree corresponds to polynomial size proof. And linear lower bounds (ϵn), or even $\sqrt{n}$ lower bounds on degree often give exponential lower bounds on the size of the proofs.

Lower Bounds

- Polynomial calculus refutations of the pigeonhole principle require degree $\sqrt{n}$. [Razborov]

- Polynomial calculus refutations of the counting principles require degree $\Omega(n)$. [Buss-Grigoriev-Impagliazzo-Pitassi, ...]

- The housesitting principles have degree 3 polynomial calculus proofs, but require degree n Nullstellensatz refutations. [CEI]

Positivstellensatz

[Grigoriez-Vorobjov] Working over an ordered field, start with same polynomials as the polynomial calculus. A Positivstellensatz refutation consists of showing that

$$\sum_i g_i f_i = 1 + \sum_j h_j^2.$$

in a the static situation (analogous to the Nullstellensatz system).

Or, in the inference based version using the inference rules of the polynomial calculus, a Positivstellsatz refutation must derive

$$1 + \sum_j h_j^2 = 0.$$

Automatizability and Proof Search

Cook's program concerned only the *existence* of proofs. For practical application, proof *search* is at least as important.

Def'n A proof system T is *automatizable* provided there is an algorithm $f(x)$ and a constant $c > 0$ such that, whenever $T \vdash \varphi$ with a proof of k symbols, then $f(\varphi)$ runs for less than k^c steps and outputs a T-proof of φ.

Automatizability means there is a polynomial time algorithm for proof search.

Some positive automatizability results

Thm [Clegg-Edmonds-Impagliazzo; Beame-Pitassi]
Tree-like resolution is quasiautomatizable.
I.e., If there is a tree-like resolution refutation of size S, then a tree-like resolution refutation can be found in time $n^{O(\log S)}$.
(n is the number of variables.)

Thm [CEI; BP]
Resolution is quasiautomatizable w.r.t. large proofs.
I.e., if there is a resolution refutation of size S, then a resolution refutation can be found in time $n^{O(\sqrt{n \log S})}$.

Proof idea: Bottom-up search for tree-like proofs.

Thm [CEI] Fix d to be constant. The degree d Nullstellensatz and the degree d polynomial calculus are automatizable.

Proof idea: only n^d monomials.

Some negative automatizability results

Def'n A product of two primes congruent to 3 mod 4 is called a *Blum integer*.

Thm [Bonet-Pitassi-Raz] Frege proof systems are not automatizable, unless there is a probabilistic polynomial algorithm for factoring Blum integers.

Similar results hold for cd-Frege and *almost any stronger proof system*.

(Proof idea: Reduction via Craig interpolation.)

Thm [Alekhnovich-Razborov] Resolution is not automatizable unless the parameterized polynomial time hierarchy collapses and the weak parameterized class $W[P]$ is in randomized fixed-parameter polynomial time (FPT).

(Proof idea: reduction to minimum weight monotone circuit satisfiability.)

A discouraging remark and an encouraging remark.

1. High proof complexity does not *always* imply fragility. E.g., resolution requires exponential size proofs of the $2n \longrightarrow n$ pigeonhole principle. [Haken, Buss-Turan]. Or the $2^n \longmapsto n$ pigeonhole principle. [Raz, Razborov].

2. In many cases, it is possible to identify classes of tautologies that have easy decision procedures. E.g., propositional formulas drawn from a random distribution can often be easy to test for satisfiability — unless the random distribution has been deliberately chosen to make it difficult to decide satisfiability.

3. There is still much to discover!