Expressibility and Derivability, and the Complexity of Proofs

Sam Buss
Univ. of California, San Diego

PhilMath Intersem, June 29, 2011.

State of Profound Ignorance

The P versus NP problem is just one example of a range of similar open problems.

Open problem: Show that some natural (combinatorial, say) problem requires superlinear runtime $\Omega(n)$.

Open problem Show that that some such problem requires Boolean circuit size $> 9 \cdot n$.

In essence, the known lower bounds for computational complexity (of natural, constructive problems) are obtainable from the fact that it is necessary to read the entire input.

Similarly, we have profound ignorance on the complexity of propositional proofs.

Definition: A *Frege proof* is a proof in a "textbook" propositional proof system based on modus ponens.

The following is our best lower bound on the length of Frege proofs.

Let ϕ be $(T \wedge (T \wedge (\cdots (T \wedge (T \wedge T)) \cdots)))$.

Then, any Frege proof of ϕ requires $c \cdot n$ steps and $c \cdot n^2$ symbols for some constant $c > 0$. (n is the size of ϕ.)

Proof idea: It is necessary to include some constant fraction of the subformulas of ϕ as lines in the Frege proof.

In general, the *complexity* of a proof is the number of symbols in the proof.

Extended Frege proofs are defined as Frege proofs in which abbreviations may be introduced (to avoid repeating long formulas). Equivalently, an extended Frege proof is a Frege proof but with proof length equal to the number of proof steps (lines).
Open problem: Give superpolynomial lower bounds on the lengths of Frege and extended Frege proofs.

Remarks: 1. This would be a large step towards proving $\mathrm{P} \neq \mathrm{NP}$. Indeed $\mathrm{NP} = \mathrm{coNP}$ is equivalent to the existence of a proof system in which all tautologies have short (=poly size) proofs.
2. It is conjectured ([Tseitin '68] and [Cook-Reckhow '79]) that Frege proofs cannot polynomially *simulate* extended Frege proofs.
3. However, we currently have no reasonable conjectures for natural tautologies for which extended Frege proofs are more efficient than Frege proofs. [BBP '95].

Definition The pigeonhole principle tautologies (PHP) state the following formulas are inconsistent ($n > 0$):

$$\bigvee_{k=1}^{n} x_{i,k} \quad \text{for each } i = 1, \ldots, n+1$$
$$\overline{x}_{i,k} \vee \overline{x}_{j,k} \quad i < j \leq n \text{ and } k \leq n+1.$$

Intuitively, there is no bijection from $[n+1]$ to $[n]$.

Side remark: [Kreisel-Mints-Simpson '75] dismisses the propositional pigeonhole principle as something that one should never try to prove in propositional logic.

However, to the contrary, PHP has proved to be an important test case for understanding the strength of propositional proof systems.

Thm. [CR '79]. PHP has poly size extended Frege proofs.

Thm. [B '87]. PHP has poly size Frege proofs.

[CR '79] used an inductive reduction. [B '87] used a direct argument based on expressing "counting" with poly size formulas.

Explanation(?) in terms of Expressibility

The lines in a polynomial size Frege proof are *polynomial-size (Boolean) formulas*. In an extended Frege proof, they are *polynomial-size circuits.*

Conjecture. Boolean formulas cannot polynomially simulate Boolean circuits.

That is, the expressive power of Frege proof formulas is (conjectured to be) strictly less than that of lines in an extended Frege proof.

This is the basis of the [CR '79] conjecture.

The poly size PHP proofs, [B '87], were based on expressing counting with poly size formulas. This supports the intuition that expressibility helps provability.

Resolution is a *refutation* proof system that acts on clauses. A *clause* is a disjunction of *literals*, namely of variables and negated variables.

The resolution inference rule is:

$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$$

Theorem [Haken '85] Resolution refutations of PHP require size 2^{n^ϵ}.

Clauses cannot express counting, but the proof did not use this. It used instead a detailed analysis of the presence of large clauses in a resolution refutation.

A significant generalization of the fact that clauses cannot express counting is:

Switching Lemma. [Yao '85, Hastad '86] Expressing counting (even mod 2) with constant depth Boolean circuits requires size 2^{n^ϵ}. Here $\epsilon = \Omega(1/\mathrm{depth})$.

In conformance with the intuition that counting is needed to prove PHP:

Theorem. [PBeI, KPW, '93/'95]. Constant depth Frege proofs of PHP require size 2^{n^ϵ}.

But here $\epsilon = (1/6)^{1/\mathrm{depth}}$, so there is a mismatch between expressibility and provability.

Definition The Count^q, counting mod q, principle states that a set of size $n \not\equiv 0 \bmod q$ cannot be partitioned into sets of size q. It uses variables x_U for $U \subseteq [n]$, $|U| = q$.

Definition Let $p > 1$. An $\oplus_p$-Frege proof is a Frege proof in the language augmented with (unbounded fanin) *mod p* gates, along with suitable axioms.

Theorem [Razborov-Smolensky '87] Let p and q be distinct primes. Bounded depth $\oplus_p, \wedge, \vee, \neg$ circuits that count mod q require size 2^{n^ϵ}. Here ϵ is $1/(2 \cdot \mathrm{depth})$.

Open Problem Are there polynomial size, constant depth $\oplus_p$-Frege proofs of the Count^q principles?

A *grid graph* is a graph on a $k \times n$ grid. The *st-connectivity principle* states that in a grid graph where the edges form non-intersecting paths (in-/out-degree ≤ 2), and are colored either red or green, it is not possible to have the vertices $(1,1)$ and (k,n) the only nodes with red degree 1, and the vertices $(1,n)$ and $(k,1)$ the only ones with green degree 1.

Intuition: the red and green paths would have to cross.

Theorem [BaLMS '98] The "st-connectivity *property*" for width k graphs is not expressible by poly-size depth $k-1$ circuits. (In fact, is complete for Π_k-circuits).

However, in spite of this inexpressibility:

Theorem. [B'06] For fixed k, the $k \times n$ st-connectivity principle has poly size resolution proofs.

The grid graphs gave an example of where the seemingly needed concepts are not expressible, but still there are short proofs. For a converse example, monotone propositional logic gives an example where the needed concepts *are* expressible, but for which we do not know any short proofs.

Definition The propositional *monotone sequent calculus* allows sequents of the form $\vec{A} \to \vec{B}$ using conectives $\wedge$ and $\vee$ only (no negation, no negated literals).

Theorem The (non-monotone) sequent calculus is conservative over the monotone sequent calculus.

Theorem [Atserias-Galesi-Pudlák '02] In fact, the monotone calculus can simulate the sequent calculus with quasipolynomial ($= 2^{(\log n)^c}$) size proofs.

Theorem [Valiant '84, Ajtai-Komlós-Szemerédi '83] There are polynomial size monotone formulas for threshold (monotone counting) and majority functions.

However, these formulas use randomized constructions or expander graphs for which it is difficult to prove correctness.

Theorem [AGP '02] If there are polynomial size Frege (or, non-monotone sequent calculus) proofs of the correctness of monotone threshold formulas, then the monotone sequent calculus can polynomially simulate the sequent calculus.

Definition Let LK be a usual sequent calculus formalization (equivalently, Hilbert-style) of first-order logic. Let LK^- be LK without equality.

Fact. LK is conservative over LK^-.

Proof. Use cut-elimination. This gives a superexponential simulation of LK-proofs by LK^-.

Theorem For languages without function symbols, the simulation can be improved to polynomial.

Proof idea: Although equality cannot be directly expressed, one can replace "a=b" with $\forall \vec{x}(R(a,\vec{x}) \leftrightarrow R(b,\vec{x}))$ conjoining for all R's.

Open question. What is the speedup of LK over LK^- for languages that contain function symbols?

Proof search is arguably more important than proof complexity, at least for practical applications. Of course, lower bounds on proof complexity imply lower bounds on proof search, but it seems that the proof search problem is hard even when short proofs exist.

Theorem [Buss '91] For the Gentzen sequent calculus LK the following problem is undecidable: Given a formula ϕ and an integer k, does ϕ have a proof of $\leq k$ steps?

Theorem [Alekhnovitch-Buss-Moran-Pitassi '00] For almost all natural proof systems (resolution, Frege, cutting planes, nullstellensatz, cut-free, etc.), it is impossible to approximate shortest proof length in polynomial time to with a factor of $2^{(\log n)^{1-\epsilon}}$ unless $\mathrm{P} = \mathrm{NP}$. (Where n is the length of a shortest proof.)

Definition A proof system P is *automatizable* if there is a procedure which, given a formula A with a P-proof of length n, finds some P-proof in time polynomial in n.

Theorem [Alekhnovitch-Razborov '01] If resolution is automatizable, then the weak parameterized hierarchy $W[P]$ collapses.

The most striking result along these lines is:

Theorem [Bonet-Pitassi-Raz '97] If Frege proofs are automatizable, then factorization of Blum integers is in polynomial time.

The Bonet-Pitassi-Raz proof uses a kind of Craig interpolation. In fact, they show it is enough to decide the following in polynomial time:

> *Given a Frege proof of $A \vee B$ where the propositional formulas have* no *common variables, correctly identify one of A or B as being a tautology.*

The connection with Craig interpolation is that $A \vee B$ is equivalent to $(\neg A) \rightarrow B$.

Craig interpolation has also been useful for lower bounds in resolution, and in cutting planes (linear integer inequalities over Boolean variables).

Surprisingly, Craig interpolation has become important for the construction of automated theorem provers (SMT solvers) for fragments of first-order logic using for hardware and software verification.

Thank you!

Survey paper: S. Buss, "Towards NP-P via Proof Complexity and Search" available on my web page has many of the citations, plus a lot more about current work in this direction motivated by the P vs NP question.