Theory Open Problem Session, June 2003

1 Avner Magen, embedding NEG into ℓ_1

Definitions

- A nonnegative symmetric function $d : \{1, \ldots, n\}^2 \mapsto \mathbb{R}$ is an $\ell_1 - metric$ if there are $x_1, x_2, \ldots, x_n \in \mathbb{R}^m$ (arbitrary m) such that $d(i,j) = \|x_i - x_j\|_1$ where $\|\cdot\|_1$ is the ℓ_1 norm.
- A nonnegative symmetric function $d : \{1, \ldots, n\}^2 \mapsto \mathbb{R}$ is in ℓ_2^2 if there are $x_1, x_2, \ldots, x_n \in \mathbb{R}^n$ such that $d(i,j) = \|x_i - x_j\|^2$ where $\|\cdot\|$ is the regular Euclidean norm. If in addition the function is a metric (ie it satisfies triangle inequality), we say that it is in NEG.
- The distortion between two functions d and d' is

$$\max_{i \neq j} \frac{d(i,j)}{d'(i,j)} \cdot \max_{i \neq j} \frac{d'(i,j)}{d(i,j)}.$$

A known fact is that all $\ell_1 - metrics$ are in NEG.

Optimizing a linear function over over NEG is easy. Optimizing over ℓ_1-metrics is NP-hard.

The question is: what is the best $\alpha = \alpha(n)$ for which every $d \in NEG$ can be embedded into $d' \in \ell_1$ with distortion $\alpha(n)$. Trivially, $\alpha(n) = \Omega(1)$, and from Bourgain classical result stating that any metric space can be embedded to ℓ_1 with distortion $O(\log n)$ it follows that $\alpha(n) = O(\log n)$. The last fact alone provides the best known approximation algorithm for the very important problem of sparsest cut (given a graph G find a proper subset S of the vertices minimizing $\frac{|e(S,\overline{S})|}{|S||\overline{S}|}$). Improving the upper bound to $\alpha(n) = o(\log n)$ will be a major result. A result of a lesser algorithmic merit, but still very interesting is a nontrivial lower bound for $\alpha(n)$; namely showing $\alpha(n)$ goes to infinity with n. (hey - you might get lucky and prove both...)

Reference: J. Matoušek. *Lectures on discrete geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002. Relevant chapter is online(!) in http://kam.mff.cuni.cz/ matousek/dg-metr.ps.gz

2 Stefan Szeider: Complexity of Resolution with Dynamic Symmetry Rules

A *renaming* ρ is a map of from literals to literals with the property $\rho(\overline{x}) = \overline{\rho(x)}$. For a clause C we write $\rho(C) = \{\rho(x) \mid x \in C\}$, and for a set of clauses F (representing a CNF formula) we write $\rho(F) = \{\rho(C) \mid C \in F\}$.

Krishnamurthy [2] introduced a new rule for the resolution system which takes advantage of symmetries of the initial clauses; it allows the following inference:

> If a clause C is derived from a set of clauses F, and there is an injective renaming ρ such that for every clause $A \in F$ used in the derivation of C, $\rho(A)$ is in F, then the *symmetry rule* allows the derivation of $\rho(C)$.

Adding this rule to the resolution system yields a system which allows linear-length proofs of the pigeon hole principle; actually, this can even be achieved with a weak form of the symmetry rule, the *global* symmetry rule, using only renamings which act as automorphisms on the set of initial clauses.

However, as shown by Arai and Urquhart [1], any formula F which is hard for resolution (i.e., requires an exponential number of steps) can be transformed into a formula F^* which is hard for resolution + symmetry rule: add to each clause C of F a different number of new variables $x_1, \ldots, x_{k(C)}$ and provide the unit clauses $\{\overline{x_1}\}, \ldots, \{\overline{x_{k(C)}}\}$. Now any renaming must map a clause of width ≥ 2 to itself and so the symmetry rule cannot shorten a resolution proof of F^* significantly.

One possibility of generalizing the symmetry rule is to allow non-injective renamings. In fact, such generalization (the *homomorphism rule*) allows an exponential speed-up over the symmetry rule [3]. However, similarly as above (but more involved), any formula F which is hard for resolution can be transformed into a formula F^{**} which is hard for resolution + homomorphism rule. Note that resolution + homomorphism rule and cutting plane proofs are incomparable in terms of p-simulation.

The quoted transformations from F to F^* or F^{**} can be undone using a polynomial number of resolution steps (unit resolution or Davis Putnam resolution). Hence, the hardness of F^* and F^{**} relies crucially on the fact that the considered renamings must act on initial clauses. Therefor we propose a more general rule which relaxes this restriction and allows the following inference:

> Suppose we have derivation of a clause C from a set F of initial clauses containing a sub-derivation of C from clauses $D_1, \ldots, D_k$; if for some injective renaming ρ, $\rho(D_1), \ldots, \rho(D_k)$ are derived clauses, then the *dynamic symmetry rule* allows the derivation of $\rho(C)$.

A *dynamic homomorphism rule* can be formulated analogously allowing non-injective renamings.

The dynamic symmetry rule yields an exponential speed-up over the local homomorphism rule since it allows short proofs of the above formulas F^*, F^{**}.

The complexity of the dynamic systems (resolution + dynamic symmetry rule, resolution + dynamic homomorphism rule) is open. We ask for hard examples for the dynamic systems or a good excuse for not finding such hard examples (e.g., by p-simulating a strong proof system within a dynamic system).

[1] N. H. Arai and A. Urquhart, Local Symmetries in Propositional Logic, Proceedings of TABLEAUX 2000, LNCS 1847, 2000, 40–51.

[2] B. Krishnamurthy, Short Proofs for Tricky Formulas, Acta Informatica 22, 1985, 253–275.

[3] S. Szeider, The Complexity of Resolution with Generalized Symmetry Rules, Proceedings of STACS 2003, LNCS 2607, 2003, 475–486.

3 Alasdair Urquhart: Complexity of the first-order theory of p-adic numbers

What is the complexity of the first-order theory of the p-adic numbers? Let p be a prime. Then the axioms of the first order theory of the field $\mathbf{Q}_p$ of p-adic numbers are those for a field with valuation, together with those that state that the characteristic is 0, the valuation group is a Z-group, the residue field is finite, Hensel's lemma holds and $v_p p = 1$. The problem was originally proved solvable by Ax and Kochen, who used model-theoretic methods. Subsequently, Paul Cohen [1] provided an explicit quantifier-elimination method; this results in an algorithm that is primitive-recursive, but not elementary. Lavinia Egidi [2] later published an argument that there is an elementary decision procedure, but withdrew her claim in a later publication [3]. At the moment, it appears to be an open question whether there is an elementary decision procedure. Egidi provides some arguments that make it plausible that there is no elementary algorithm for the problem. The only known lower bound is the same as that for Presburger arithmetic, that is to say, doubly exponential alternating time with a linear number of alternations.

References

[1] Paul J. Cohen. Decision procedures fo real and p-adic fields. *Communications on Pure and Applied Mathematics*, 22:131–151, 1969.

[2] Lavinia Egidi. The complexity of the theory of p-adic numbers. In *Proceedings of the 34th Annual IEEE Symposium on the Foundations of Computer Science*, pages 412–421, 1993.

[3] Lavinia Egidi. A quantifier elimination for the theory of p-adic numbers. *Computational Complexity*, 7:205–263, 1998.

4 Shlomo Hoory: The Unique neighbour problem

Consider a hypergraph $H = (V, E)$ with $n = |V(H)|$ vertices and $m = |E(H)|$ edges. Let $f(H)$ be the size of the smallest non-empty vertex set S satsifying $|e \cap S| \neq 1$ for every edge $e \in E(H)$. Let $f(n, m)$ be the maximum of $f(H)$ over all such hypergraphs.

From coding theory, it follows that for any $\epsilon > 0$ and a sufficiently large n

$$f(n, (1-\epsilon) \cdot n) \leq \frac{1}{2} \cdot n. \tag{1}$$

The conjecture is that there exists $\delta > 0$ such that for any $\epsilon > 0$ and a sufficiently large n

$$f(n, (1-\epsilon) \cdot n) \leq (\frac{1}{2} - \delta) \cdot n. \tag{2}$$

At this time, I don't even know a proof for (1) that doesn't use coding theory.

One can restrict the problem and define f_k to be the same as f when considering only k-uniform hypergraphs. For $k = 1$ the cojecture is trivial. For $k = 2$ it follows from the fact that the hypergraph must have many connected components and one of them must be small. One can prove the conjecture for any $k > 9$ using a simple probabilistic argument. I think it would be interesting if the cojecture can be proven (or refuted) for $k = 3$.

About lower bounds. The best I have is that for any $\epsilon > 0$ and a sufficiently large n

$$f(n, (1-\epsilon) \cdot n) \geq f_3(n, (1-\epsilon) \cdot n) \geq 0.254 \ldots \cdot n.$$

5 Sam Roweis

We are given a function $M(v) = w$ which computes the result of left multiplying a vector v by a matrix M. The matrix M is assumed to be symmetric positive-definite, with edge size n. We wish to estimate the determinant of M, $\det M$.

- Is there an algorithm which, with probability at least $(1 - \delta)$, returns an estimate V of the determinant of M such that $(1-\epsilon) < \log\det M / \log V < (1 + \epsilon)$ using at most $n \log n$ calls to $M(\cdot)$, and storage linear in n? (Of course, the number of calls will also depend on δ, ϵ.)
- At most n calls? At most $\log n$ calls? A constant number of calls?
- Such that $(1 - \epsilon) < \det M/V < (1 + \epsilon)$? Such that $|\det M - V| < \epsilon$?

6 Toni Pitassi

This question is concerned with finding a better time-space tradeoff for solving the satisfiability problem as a function of the branch width of the underlying hypergraph.

Let f be a 3CNF formula over n variables with m clauses. That is, $f = C_1 \wedge C_2, \wedge \ldots \wedge C_m$. The hypergraph associated with f has m edges and n vertices. There is one vertex corresponding to each of the n variables, and one hyperedge corresponding to each of the m clauses. (The orientation of the variables in f is lost in going from f to the hypergraph corresponding to f.) The *branch width* of a hypergraph is a popular notion that gives a measure of how expanding the graph is. The definition of branch width can be found, for example, in the following paper. (Dalmao, Bacchus, and Pitassi, FOCS 2003.) It is known that if a hypergraph $H = ([n], E)$ has tree width w, then there is a tree decomposition for H of height $logn$ and width $O(w)$. That is, it is possible to obtain a balanced branch decomposition without compromising much in the width.

A good branch decomposition for $H(f)$ gives rise to a natural divide-and-conquer algorithm for solving satisfiability for f. Namely, beginning at the root of the branch decomposition, try all 2^w settings of all variables at the root label. Each such setting breaks f into two variable-disjoint pieces, $f_0 \wedge f_1$, and thus we continue recursively solving satisfiability for f_0 and f_1 using the branch decomposition. The overall runtime here is $2^{O(w \log n)}$ and the space is linear.

By saving intermediate values, an algorithm with better runtime can be obtained. This algorithm acheives time $n^{O(1)}2^w$, but now the space is $n^{O(1)}2^w$.

The open problem is to determine if it is possible to get a better time/space tradeoff. That is, is it possible to solve SAT on formulas with branch width w in time $n^{O(1)}2^w$, and linear space?

7 Alfred Lehman

This question is concerned with whether NP equals $coNP$. Consider f, a monotone (non-decreasing) finite boolean function, which represents an NP-complete problem. For example, let f be the function on $\binom{n}{2}$ variables which distinguishes whether a set of chosen variables constains a Hamiltonian circuit. There is also the directed graph version with twice the variables (edges) and twice the (n!) automorphism group. This version appears to be width-length (ideal matrix). The dual of f (the interchage of variable values and result values) describes the co-problem. If the minimal terms of the dual function can each be verified in polytime, then $NP = coNP$. In any case, their structure (mod the automorphism group) is of interest. Thus far, without computation I have not found any "hard" terms.