Research Statement

Youming Qiao *

November 30, 2011

1 Introduction

My research interest lies in complexity theory, with an emphasis on its interaction with group theory. To put it another way, I would like to understand the interaction between computation and symmetry, motivated by the natural problems in computation. This interaction between computation and symmetry has a long history, and is beautiful and fruitful. It is not hard to interpret the Fundamental Theorem of Algebra and insolvability of quintic equations using modern languages like straightline programs. In the times of Turing, various problems for (infinite) groups, like isomorphism problem and word problem (cf. e.g. [1]), were shown to be unsolvable algorithmic problems. With the advent of information age and complexity theory, there are more instances bonding group theory with the theory of computation, often in a fairly surprising manner. For example, Barrington's theorem [8] (bounded-width branching programs capture NC^1), polynomial-time algorithm for graphs with bounded degree [20], the invention of Arthur-Merlin games (cf. e.g. [6]) (helping to understand the possible status of graph isomorphism problem in complexity classes), and the new, asymptotically-faster algorithm for matrix multiplication using Fourier transform over non-abelian groups [13]. Recently, Mulmuley and Sohoni started geometric complexity theory (cf. e.g. [27, 28, 26]), which tries to reveal the role of symmetry played in the circuit lower bound problems. In particular, it can be seen in [27] that the determinant versus permanent problem can serve as a good starting point to study how symmetry may help in lower bound problems at least in the *algebraic setting*.

With this goal in mind, during my first three years of the Ph.D. study I've worked on problems like group isomorphism problem, polynomial identity test problem, and reconstruction problem for arithmetic formulas. Let me briefly explain the settings for these problems. In group isomorphism problem, the groups are finite and given as Cayley tables. In polynomial identity test problem, our focus is the model of algebraic branching programs, which captures the computation power of determinant [39]. In the reconstruction problem for arithmetic formulas, we consider the problem in the average-case sense. Namely, assume that the formulas are generated by some random procedure, and the goal is to devise an algorithm that works for almost all formulas generated. I also have papers on cryptography at the early stage of my Ph.D. study ([32], [10]). With different collaborators, we are able to achieve progress for the problems mentioned. From these experiences, I find it beneficial to study one discipline with the motivation from the other discipline in mind.

*Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, 100084, China. Email: jimmyqiao86@gmail.com

For example, with the goal of devising a polynomial-time algorithm for group isomorphism problem, I learn about the concepts and constructions in group theory like linear representations and properties of simple groups. On the other hand, in the study of geometric complexity theory, and in the projects on polynomial identity test problem, I begin to appreciate the necessity to delve into algebraic geometry and the theory of Lie groups to understand computation.

In the rest of the article, I will briefly explain my previous works and the next steps, for group isomorphism problem and polynomial identity test problem. The article will be concluded with some maybe naive postulate on the role of symmetry in computation.

2 Group isomorphism problem

2.1 Motivation and past research

I consider group isomorphism problem when groups are finite, and are given as Cayley tables. The starting point of my interest on group isomorphism is due to some of its strange characteristics as a computational problem. For example, it is not known whether the search version of the problem reduces its decision version. Also, though group isomorphism problem is known to reduce to graph isomorphism problem, it is shown that there cannot be AC^0 reduction in the other direction [12]. These observations suggest it to be a possibly outstanding problem in complexity theory.

From the upper bound side, though an $n^{\log n}$ time algorithm has been known at least since 1970's [23], there has been little progress for restricted group classes beyond abelian groups. In 2009, in [15] Le Gall devised a polynomial-time algorithm for groups in the form of a semidirect product of abelian group A by a cyclic group C, under the condition that $(|A|, |C|) = 1$. With different collaborators, my works on group isomorphism problem is to devise polynomial-time algorithms for two non-trivial group classes, described as follows.

Groups formed by coprime extensions: with László Babai [7] (following a previous paper with Jayalal Sarma and Bangsheng Tang [31]), we greatly extend the group class considered by Le Gall to the case as follows. We present a polynomial-time algorithm for groups in the form of a semidirect product of abelian group A by a group C, under the condition that $(|A|, |C|) = 1$, *as long as* $\mathrm{Aut}(C)$ *is given by a set of generators.* From the algorithmic aspect, Luks's dynamic programming idea [22] is applied to give a parameterized algorithm for setwise stabilizer problem, one of the well-known problems in permutation group algorithm that are known as graph isomorphism hard (cf. [21]). From the group-theoretic aspect, linear representation theory, Wedderburn's theorem and in-depth analysis of p'-automorphisms of abelian p-groups arise naturally in this context.

Groups without abelian normal subgroups: following Robinson in [34], we call such groups *semisimple groups.* With László Babai and Paolo Codenotti [5] (following a previous paper with László Babai, Paolo Codenotti and Joshua Grochow [4]), we devise a polynomial-time algorithm for this class of groups. There are two major components in this algorithm: the first is an algorithmic one, which adapts Luks's singly-exponential-time algorithm for hypergraph isomorphism [21] to take into account the size of the alphabet. The second one is group-theoretic, stating that the order of the normalizer of a transitive permutation group $P \leq S_n$ is bounded by $c^n \cdot |P|$, where c is an absolute constant.

2.2 Future directions

p-groups are generally believed to be the bottleneck of the group isomorphism problem. We can draw some analogy from chemistry: if we compare simple groups as the elements in the periodic table, then the abelian simple groups can be compared as the carbon: just like the universality of carbon that makes the carbon-based livings on the Earth possible, p-groups are the main reason why the number of groups reaches $\exp(\Theta(\log^3 n))$ (cf. [9]). Therefore, currently the group classes with polynomial-time algorithms are in some sense avoiding the p-groups.

The major implication of polynomial-time algorithms for groups formed by coprime extensions and for semisimple groups is the possibility of a reduction from general groups to p-groups. There are two relevant group-theoretic constructions.

Extension of a solvable group by a semisimple group: given a group G, take its largest solvable normal subgroup R, called the *solvable radical* of G. Then G/R has no abelian normal subgroups, thus is semisimple. Recall that we have polynomial-time algorithm to test isomorphism of semisimple groups. Thus to reduce the general case to solvable case, the first step should be to understand the way a solvable group is extended by a semisimple group.

Knitted product: given a group G and its two subgroups H and K, G is called the *knitted product* of H and K if $G = HK$ and $H \cap K = \{\mathrm{id}\}$. (Compare this definition with that of the semidirect product, where H is required to be normal.) Hall's theorem establishing the existence of Sylow systems for solvable groups can be interpreted as: a group is solvable if and only if it is an iterated knitted product of (a selection of) its Sylow subgroups. I feel that understanding knitted product will help with reducing the solvable case to p-groups.

We note that in [11] certain heuristics have been developed for isomorphism test, in line with the idea "separating solvable radical" first. Still, I believe that neither of these constructions has received considerable attention from group theorists, at least from the computational perspective. From the lower bound side, there are also interesting problems to ask concerning its status in complexity theory. For example, whether the search version of group isomorphism problem reduces to the decision version, and whether group isomorphism problem can be hard for some low level complexity class.

3 Polynomial identity test problem

3.1 Motivation and past research

Arithmetic circuits can be viewed as an algebraic analogue of boolean circuits, where $\wedge$, $\vee$ and $\neg$ are replaced by $\times$, $+$ and $\times(-1)$ (multiplying by minus 1). The operations can be viewed as performed over a field $\mathbb{F}$ (not necessarily GF(2)), and the output of the circuit is a polynomial over $\mathbb{F}$. To prove non-trivial lower bound of a specific polynomial over this model is thus seen as an algebraic analogue of proving boolean circuit lower bound. For a recent survey see [36]. In [40], Valiant made up an analogous theory of completeness for this model, and thus lead to the wonderful problem: determinant versus permanent. Formally, given a matrix of variables $X = (x_{i,j})_{i,j\in[n]}$, we want to find another matrix $Y = (\ell_{i,j})_{i,j\in[m]}$, where $\ell_{i,j}$ are linear polynomials in X. The conjecture is that, for large enough n, in order to ensure the existence of Y such that $\det(Y) = \mathrm{perm}(X)$, it is necessary that m is super-polynomial in n. In order to formulate the determinant versus permanent problem, Valiant made crucial use of a model called *algebraic branching program* (ABPs), defined as follows. Given a directed acyclic graph G with source s and sink t, we label each edge by a

variable or a field constant. For a path p from s to t, let $\pi(p)$ compute the product of labels on the edges along the path. Then G "computes" $\sum_p \pi(p)$, where summation is over all paths from s to t. Toda [39] observed that the computation power of ABPs exactly captures that of the determinant, making it an interesting computational model.

In the exploration of the power of randomization in computation, people gradually realize the connection between derandomization and circuit lower bound (cf. e.g. [30]). This line of research culminates at the work of Kabanets and Impagliazzo [18], where they show that in some sense derandomization and circuit lower bound are equivalent. The main result is that derandomization of *polynomial identity test problem* (PIT) would result in either a boolean circuit lower bound, or an arithmetic circuit lower bound. Thus devising deterministic, efficient algorithm for PIT is seen as an alternative way to prove lower bound, as advocated by Agrawal [2]. With Maurice Jansen and Jayalal Sarma, we focus on the PIT problem where the underlying circuit model is ABP. We present non-trivial black-box identity tests for ABPs, with the following restrictions.

Sum of constant number of read-once ABPs: given a variable x and an ABP A, the *read of x in A* is the number of edges labeled with x. The *read of A* is the maximum over the reads of all variables appearing in A. For sum of constant number of read-once ABPs, in [16] we achieve quasi-polynomial-time, black-box identity test. The technique is derived from Shpilka and Volkovich [35] for sum of constant number of read-once formulas, while necessary adaptation to accommodate the structural difference between ABPs and formulas is needed.

Ordered ABPs with constant reads: an ABP is *ordered*, if there is an order of the variables, such that each path from source to sink respects this order. For ordered ABPs with constant read, in [17] we devise quasi-polynomial-time, black-box identity test. The idea of the algorithm follows from Nisan's pseudorandom generator [29] for branching programs to reuse the randomness, while we need to combine with basic facts for dimension of an algebraic variety to get the construction.

3.2 Future directions on PIT

In an ongoing work with Neeraj Kayal and Ankit Gupta, we aim at a reconstruction algorithm for general arithmetic formulae. Recall that in the reconstruction problem, we are given black-box access to a polynomial and the goal is to construct the (roughly) smallest formula computing it. It shows interesting applications of some well-known methods in the symbolic computation community, e.g. effective Nullstellenstaz, Nöether normalization, etc., in complexity-theoretic problems.

An arithmetic circuit is *multilinear*, if each gate computes a multilinear polynomials. Raz [33] proved super-polynomial lower bound of determinant and permanent for *multilinear formulas*. A next natural question to ask if one can show non-trivial lower bound for *multilinear ABPs* (cf. a recent paper proving superpolynomial separation between multilinear ABP and multilinear formula [14]). In this aspect, determinant is particularly interesting, as it can be computed by polynomial-size (general) ABPs (cf. e.g. [39]), while the best formula so far computing determinant is quasi-polynomial (matching the lower bound proved by Raz over multilinear correspondent). Thus a super-polynomial lower bound for determinant over multilinear ABPs will lead to a separation between multilinear computation and general computation, thus may shed light on deeper understanding of arithmetic circuits. Another next model to examine is ABPs with constant number of reads, without the ordered constraint. There have been progress on multilinear formulas with constant number of reads [3], but for formulas with constant number of reads, as far as I know, is still open.

4 The role of symmetry in computation

4.1 Turning a lower bound problem to an upper bound problem

The ultimate goal in complexity theory is to understand *efficient computation*. Though in practice, in different situations the focuses can be different, in theory, the ultimate goal would still be to understand polynomial-time algorithms and/or polynomial-size circuits. So what is the possible criterion for a problem to be tractable or intractable, namely, to have polynomial-time algorithm or not to have? In the following, by *lower bound problems*, we mean the various versions of circuit lower bound, e.g. boolean or arithmetic, uniform or non-uniform, formula or circuit or branching programs, etc.. It is also noted that the content in this section is only a personal understanding of geometric complexity theory, *without* contribution from the author.

We first recall certain well-known results in mathematics. The criterion for a graph to be planar is provided as in *Kuratowski's theorem*. Namely, a graph can not be embedded in the plane, if and only if it contains a subgraph *homeomorphic* to either K_5 or $K_{3,3}$ (called Kuratowski subgraphs). Recall that a graph G is homeomorphic to another graph H, if after appropriate insertion and/or deletions of degree-2 vertices in G, the resulting graph is isomorphic to H. This immediately puts non-planarity test in NP. This result in some sense explains the planarity test can be performed in polynomial time, though it is not necessary for planarity test algorithms to use the Kuratowski's criterion. A more suggestive example might be the insolvability of general quintic equations. The criterion is provided by Galois's theorem, namely, a quintic equation is not solvable by radicals, if and only if its associated Galois group is not solvable. Note that until 1980's in [19], a polynomial-time algorithm for determining whether the Galois group of a polynomial is solvable or not is developed.[1]

The spirit embodied in the examples above is that a "lower bound problem" (non-existence of certain objects, e.g. embedding of a graph on the plane, a series of radicals expressing the roots of a quintic equation, etc.) is usually resolved by first turning into an "upper bound problem" in search for some obstructions (e.g. locating K_5 or $K_{3,3}$ in the graph, or determine the insolvability of a group). This philosophy of attacking lower bound problems is made explicit in the *flip strategy* proposed in the geometric complexity theory, partially motivated by contemplations on how to avoid the root difficulty due to the nature of P v.s. NP problem. We refer to [25] for further explanation on this strategy.

4.2 The role of symmetry in lower bound problems

With the understanding of flip strategy in mind, one may wonder what is the corresponding upper bound problem for the determinant versus permanent problem. Let's examine more closely on Galois's theorem, which exploits the symmetry of the roots of a polynomial to provide a criterion for its solvability. To put it in a naive way, given a polynomial, one can consider the algebraic relations to be satisfied by the roots. In order to keep those algebraic relations intact, certain symmetry among the roots will emerge. Naively speaking, some roots are interchangeable, while

[1] We here provide a simple procedure showing that solvability of a group can be determined algorithmically. Given a group G, take an abelian normal subgroup A, if such an A exists. Then form $G_1 = G/A$, and recursively perform the operation above, until we reach some G_k such that it has no abelian normal subgroups. If G_k is trivial, then G is solvable. Otherwise it is not. The algorithm in [19] is essentially different from this, but hopefully it illustrates the point that *insolvability can be determined algorithmically.*

others are not. It turns out that this symmetry captures the solvability of the polynomial, as exhibited in [38].

This is the most well-known instance showing that symmetry can play a role in the transition from lower bound problems to upper bound problems. Its possible role in lower bound problems is again put into a concrete form in geometric complexity theory, where the translation to upper bound problems is tied with symmetry from the very beginning. We recall the construction there for determinant versus permanent problem, over complex number $\mathbb{C}$. Let X be an n by n variable matrix. X can also be viewed as a vector in $\mathbb{C}^N$ where $N = n^2$. Thus every $\pi \in G := \mathrm{GL}(N, \mathbb{C})$ has a group action on a polynomial $f \in \mathbb{C}[X]$, denoted as f^π, by $f^\pi(x) = f(\pi^{-1}(x))$. Denote by V the linear space of homogeneous polynomials of degree n in $\mathbb{C}[X]$. Thus for example, $\det(X)$ is a point in V, and for $f \in \mathbb{C}[X]$, the orbit of f under the action of G we represent as Gf. Take the closure of Gf in Zariski topology to get a projective variety $A = \overline{Gf}$.[2] Then another projective variety B concerning the linear expression of permanent of m by m matrices in terms of determinant is defined in a similar vein, and it is shown that $B \subseteq A$ if and only if permanent can be approximated infinitesimally by a linear expression in terms of determinant. By approximating infinitesimally, it means approximating infinitesimally in the linear space of homogeneous polynomials of degree m. After translating into a geometric problem (to decide if $B \subseteq A$) using a construction motivated by group-theoretic means, it suggests certain group properties may serve as an obstruction for the lower bound problems. This is again conjectured in geometric complexity theory, namely the representations of the stabilizers of the polynomials. I omit the details of the explicit statement but refer to [25], Section 9.

4.3 A specific problem posed by geometric complexity theory

From the above description of geometric complexity, we see that it does have merits to be a source of inspiration for insights into the role of symmetry in computation, though I have not understood the specifics well. Here I would like to mention a well-known problem in algebraic combinatorics that turns out to be crucial in geometric complexity theory. Recall that the irreducible representations of S_n are indexed by partitions of n. Given a partition π of n, let r_π be the irreducible representation of S_n corresponding to π. Let λ, μ and ν be three partitions. Then form the tensor product $r^{\lambda,\mu} = r_\lambda \otimes r_\mu$, and define $c_\nu^{\lambda,\mu}$ as the multiplicity of r_ν in $r^{\lambda,\mu}$. The *Kronecker coefficient problem* then asks for a positive formula for $c_\nu^{\lambda,\mu}$. Its history can be traced back to Weyl's classic monograph Classical Groups [41], and posed as an open problem by Stanley [37]. Mulmuley has put efforts towards settling it via certain nonstandard quantum groups [24]. It will take considerable work to get down to research on this topic, and much more towards making contribution to it, but the appeal seems irresistable.

References

[1] S.I. Adian. The unsolvability of certain algorithmic problems in the theory of groups. *Trudy Moskov. Math. Obshch*, 6:231–298, 1957.

[2] M. Agrawal. Proving lower bounds via pseudo-random generators. pages 92–105, 2005.

[2] It is shown that in [25] the closures in complex topology and in Zariski topology of Gf actually coincide.

[3] Matthew Anderson, Dieter van Melkebeek, and Ilya Volkovich. Derandomizing polynomial identity testing for multilinear constant-read formulae. In *IEEE Conference on Computational Complexity*, pages 273–282, 2011.

[4] László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao. Code equivalence and group isomorphism. In *SODA*, pages 1395–1408, 2011.

[5] László Babai, Paolo Codenotti, and Youming Qiao. Polynomial-time isomorphism test for groups without abelian normal subgroups. Manuscript, 2011.

[6] László Babai and Shlomo Moran. Arthur-merlin games: a randomized proof system, and a hierarchy of complexity class. *J. Comput. Syst. Sci.*, 36:254–276, April 1988.

[7] László Babai and Youming Qiao. Polynomial-time isomorphism test for groups with abelian sylow towers. To appear in STACS, 2012.

[8] D. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38:150–164, 1989.

[9] S.R. Blackburn, P.M. Neumann, and G. Venkataraman. *Enumeration of finite groups*. Cambridge tracts in mathematics. Cambridge University Press, 2007.

[10] Andrej Bogdanov and Youming Qiao. On the security of goldreich's one-way function. In *APPROX-RANDOM*, pages 392–405, 2009.

[11] John J. Cannon and Derek F. Holt. Automorphism group computation and isomorphism testing in finite groups. *J. Symb. Comput.*, 35(3):241–267, 2003.

[12] Arkadev Chattopadhyay, Jacobo Toran, and Fabian Wagner. Graph isomorphism is not AC^0 reducible to Group Isomorphism. In *Proceedings of FSTTCS 2010 (To Appear)*, July 2010. Technical report available at ECCC : TR10-117.

[13] Henry Cohn, Robert D. Kleinberg, Balázs Szegedy, and Christopher Umans. Group-theoretic algorithms for matrix multiplication. In *FOCS*, pages 379–388, 2005.

[14] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:134, 2011.

[15] Francois Le Gall. Efficient isomorphism testing for a class of group extensions. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, volume 3 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 625–636, Dagstuhl, Germany, 2009. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[16] Maurice J. Jansen, Youming Qiao, and Jayalal M. N. Sarma. Deterministic identity testing of read-once algebraic branching programs, 2009. http://arxiv.org/abs/0912.2565.

[17] Maurice J. Jansen, Youming Qiao, and Jayalal M. N. Sarma. Deterministic black-box identity testing pi-ordered algebraic branching programs. In *FSTTCS*, pages 296–307, 2010.

[18] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity testing means proving circuit lower bounds. *Computational Complexity*, 13(1–2):1–44, 2004.

[19] Susan Landau and Gary L. Miller. Solvability by radicals is in polynomial time. *J. Comput. Syst. Sci.*, 30(2):179–208, 1985.

[20] Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comp. Sys. Sci.*, 25:42–65, 1982.

[21] Eugene M. Luks. Permutation groups and polynomial-time computation. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 1993.

[22] Eugene M. Luks. Hypergraph isomorphism and structural equivalence of boolean functions. In *STOC '99: Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 652–658, New York, NY, USA, 1999. ACM.

[23] Gary L. Miller. On the $n \log n$ isomorphism technique (a preliminary report). In *STOC '78: Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 51–58, New York, NY, USA, 1978. ACM.

[24] Ketan Mulmuley. Geometric complexity theory vii: Nonstandard quantum group for the plethysm problem. *CoRR*, abs/0709.0749, 2007.

[25] Ketan Mulmuley. Explicit proofs and the flip. *CoRR*, abs/1009.0246, 2010.

[26] Ketan Mulmuley. On p vs. np and geometric complexity theory: Dedicated to sri ramakrishna. *J. ACM*, 58(2):5, 2011.

[27] Ketan Mulmuley and Milind A. Sohoni. Geometric complexity theory i: An approach to the p vs. np and related problems. *SIAM J. Comput.*, 31(2):496–526, 2001.

[28] Ketan Mulmuley and Milind A. Sohoni. Geometric complexity theory ii: Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.*, 38(3):1175–1206, 2008.

[29] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12:449–461, 1992.

[30] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[31] Youming Qiao, Jayalal M. N. Sarma, and Bangsheng Tang. On isomorphism testing of groups with normal hall subgroups. In *STACS*, pages 567–578, 2011.

[32] Youming Qiao and Christophe Tartary. Counting method for multi-party computation over non-abelian groups. In *CANS*, pages 162–177, 2008.

[33] R. Raz. Multilinear formulas for permanent and determinant are of super-polynomial size. *Journal of the ACM*, 56(2):1–17, 2009.

[34] Derek J.S. Robinson. *A Course in the Theory of Groups*. Springer, 2nd edition, 1996.

[35] A. Shpilka and I. Volkovich. Read-once polynomial identity testing. In *Proceedings of the 40th Annual STOC*, pages 507–516, 2008.

[36] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5:207–388, March 2010.

[37] Richard P. Stanley. Positivity problems and conjectures in algebraic combinatorics. In *in Mathematics: Frontiers and Perspectives*, pages 295–319. American Mathematical Society, 1999.

[38] J.P. Tignol. *Galois' theory of algebraic equations.* World Scientific, 2001.

[39] S. Toda. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Trans. Inf. Syst.*, E75-D:116–124, 1992.

[40] L. Valiant. Completeness classes in algebra. pages 249–261, 1979.

[41] H. Weyl. *The classical groups: their invariants and representations.* Princeton landmarks in mathematics and physics. Princeton University Press, 1997.