Symposium on Theoretical Aspects of Computer Science 2009 (Freiburg), pp. 87–98
www.stacs-conf.org

QUANTUM QUERY COMPLEXITY OF MULTILINEAR IDENTITY TESTING

V. ARVIND [1] AND PARTHA MUKHOPADHYAY [1]

[1] The Institute of Mathematical Sciences
CIT Campus, Taramani
Chennai, India 600113
E-mail address: arvind@imsc.res.in
E-mail address: partham@imsc.res.in
URL: http://www.imsc.res.in

ABSTRACT. Motivated by the quantum algorithm for testing commutativity of black-box groups (Magniez and Nayak, 2007), we study the following problem: Given a black-box finite ring by an additive generating set and a multilinear polynomial over that ring, also accessed as a black-box function (we allow the indeterminates of the polynomial to be commuting or noncommuting), we study the problem of testing if the polynomial is an *identity* for the given ring. We give a quantum algorithm with query complexity sub-linear in the number of generators for the ring, when the number of indeterminates of the input polynomial is small (ideally a constant). Towards a lower bound, we also show a reduction from a version of the collision problem (which is well studied in quantum computation) to a variant of this problem.

1. Introduction

For any finite ring $(R, +, \cdot)$ the ring $R[x_1, x_2, \cdots, x_m]$ is the ring of polynomials in commuting variables $x_1, x_2, \cdots, x_m$ and coefficients in R. The ring $R\{x_1, x_2, \cdots, x_m\}$ is the ring of polynomials where the indeterminates x_i are *noncommuting*. By noncommuting variables, we mean $x_i x_j - x_j x_i \neq 0$ for $i \neq j$.

For the algorithmic problem we study in this paper, we assume that the elements of the ring $(R, +, \cdot)$ are uniformly encoded by binary strings of length n and $R = \langle r_1, r_2, \cdots, r_k \rangle$ is given by an additive generating set $\{r_1, r_2, \cdots, r_k\}$. That is,

$$R = \{\sum_i \alpha_i r_i \mid \alpha_i \in \mathbb{Z}\}.$$

Also, the ring operations of R are performed by black-box oracles for addition and multiplication that take as input two strings encoding ring elements and output their sum or product (as the case may be). Additionally, we assume that the zero element of R is encoded by a fixed string. The black-box model for finite rings was introduced in [ADM06]. We now define the problem which we study in this paper.

1998 ACM Subject Classification: F.2.1 Computation on Polynomials.
Key words and phrases: Quantum Algorithm, Identity Testing, Query Complexity, Multilinear Polynomials.

STACS 2009
Symposium on Theoretical Aspects of Computer Science 2009 (Freiburg), pp. 87-98
http://drops.dagstuhl.de/opus/volltexte/2009/1801

The Multilinear Identity Testing Problem (MIT**):** The input to the problem is a black-box ring $R = \langle r_1, \cdots, r_k \rangle$ given by an additive generating set, and a multilinear polynomial $f(x_1, \cdots, x_m)$ (in the ring $R[x_1, \cdots, x_m]$ or the ring $R\{x_1, \cdots, x_m\}$) that is also given by a black-box access. The problem is to test if f is an *identity* for the ring R. More precisely, the problem is to test if $f(a_1, a_2, \cdots, a_m) = 0$ for all $a_i \in R$.

A natural example of an instance of this problem is the bivariate polynomial $f(x_1, x_2) = x_1x_2 - x_2x_1$ over the ring $R\{x_1, x_2\}$. This is an identity for R precisely when R is a commutative ring. Clearly, it suffices to check if the generators commute with each other, which gives a naive algorithm that makes $O(k^2)$ queries to the ring oracles.

Given a polynomial $f(x_1, \cdots, x_m)$ and a black-box ring R by generators, we briefly discuss some facts about the complexity of checking if $f = 0$ is an identity for R. The problem can be NP-hard when the number of indeterminates m is unbounded, even when R is a fixed ring. To see this, notice that a 3-CNF formula $F(x_1, \cdots, x_n)$ can be expressed as a $O(n)$ degree multilinear polynomial $f(x_1, x_2, \cdots, x_n)$ over $\mathbb{F}_2$, by writing F in terms of addition and multiplication over $\mathbb{F}_2$. It follows that $f = 0$ is an identity for $\mathbb{F}_2$ if and only if F is an unsatisfiable formula. However in this paper we focus only on the upper and lower bounds on the *query complexity* of the problem.

In our query model, each ring operation, which is performed by a query to one of the ring oracles, is of unit cost. Furthermore, we consider each evaluation of $f(a_1, \cdots, a_m)$ to be of unit cost for a given input $(a_1, \cdots, a_m) \in R^m$. This model is reasonable because we consider m as a parameter that is much smaller than k.

The starting point of our study is a result of Magniez and Nayak in [MN07], where the authors study the quantum query complexity of group commutativity testing: Let G be a finite black-box group given by a generating set $g_1, g_2, \cdots, g_k$ and the group operation is performed by a group oracle. The algorithmic task is to check if G is commutative. For this problem the authors in [MN07] give a quantum algorithm with query complexity $O(k^{2/3} \log\ k)$ and time complexity $O(k^{2/3} \log^2\ k)$. Furthermore, a $\Omega(k^{2/3})$ lower bound for the quantum query complexity is also shown. The main technical tool for their upper bound result was a method of quantization of random walks first shown by Szegedy [Sze04]. More recently, Magniez et al in [MNRS07] discovered a simpler and improved description of Szegedy's method.

Our starting point is the observation that Magniez-Nayak result [MN07] for group commutativity can also be easily seen as a commutativity test for arbitrary finite black-box *rings* with similar query complexity. Furthermore, as mentioned earlier, notice that the commutativity testing for a finite ring coincides with testing if the bivariate polynomial $f(x_1, x_2) = x_1x_2 - x_2x_1$ is an identity for the ring. Since $f(x_1, x_2)$ is a multilinear polynomial, a natural question is, whether this approach would extend to testing if any multilinear polynomial is an identity for a given ring. Motivated by this connection, we study the problem of testing multilinear identities for any finite black-box ring.

The upper bound result in [MN07] is based on a group-theoretic lemma of Pak [Pak00]. Our (query complexity) upper bound result takes an analogous approach. The main technical contribution here is a suitable generalization of Pak's lemma to a multilinear polynomial setting. The multilinearity condition is crucially required. The rest of the proof is a suitable adaptation of the Magniez-Nayak result.

For the lower bound result, we show a reduction to a somewhat more general version of MIT from a problem that is closely related to the m-COLLISION problem studied in quantum computation. The m-COLLISION problem is the following. Given a function $f : \{1, 2, \cdots, k\} \to \{1, 2, \cdots, k\}$ as an oracle and a positive integer m, the task is to determine if there is some element in the range of f with exactly m pre-images.

We define the m-SPLIT COLLISION problem that is closely related to m-COLLISION problem. Here the domain $\{1, 2, \cdots, k\}$ is partitioned into m equal-sized intervals (assume k is a multiple of m) and the problem is to determine if there is some element in the range of f with exactly one pre-image in each of the m intervals. We show a reduction from m-SPLIT COLLISION to a general version of MIT. There is an easy randomized reduction from m-COLLISION problem to m-SPLIT COLLISION problem. The best known quantum query complexity lower bound for m-COLLISION problem is $\Omega(k^{\frac{2}{3}})$ [AS04] and thus we get the same lower bound for the general version of MIT that we study. Improving, the current lower bound for m-COLLISION is an important open problem in quantum computation since last few years. [1]

Our reduction for lower bound is conceptually different from the lower bound proof in [MN07]. It uses ideas from automata theory to construct a suitable black-box ring. We recently used similar ideas in the design of a deterministic polynomial-time algorithm for identity testing of noncommutative circuits computing small degree sparse polynomials [AMS08].

2. Black-box Rings and the Quantum Query model

We briefly explain the standard quantum query model. We modify the definition of black-box ring operations by making them unitary transformations that can be used in quantum algorithms. For a black-box ring R, we have two oracles O_R^a and O_R^m for addition and multiplication respectively. For any two ring elements r, s, and a binary string $t \in \{0,1\}^n$ we have $O_R^a|r\rangle|s\rangle = |r\rangle|r+s\rangle$ and $O_R^m|r\rangle|s\rangle|t\rangle = |r\rangle|s\rangle|rs \oplus t\rangle$, where the elements of R are encoded as strings in $\{0,1\}^n$. Notice that O_R^a is a reversible function by virtue of $(R, +)$ being an additive group. On the other hand, $(R, \cdot)$ does not have a group structure. Thus we have made O_R^m reversible by defining it as a 3-place function $O_R^m : \{0,1\}^{3n} \to \{0,1\}^{3n}$. When r or s do not encode ring elements these oracles can compute any arbitrary string.

The query model in quantum computation is a natural extension of classical query model. The basic difference is that a classical algorithm queries deterministically or randomly selected basis states, whereas a quantum algorithm can query a quantum state which is a suitably prepared superposition of basis states. Our query model closely follows the query model of Magniez-Nayak [MN07, Section 2.2]. For black-box ring operations the query operators are simply O_R^a and O_R^m (as defined above). For an arbitrary oracle function $F : X \to Y$, the corresponding unitary operator is $O_F : |g\rangle|h\rangle \to |g\rangle|h \oplus F(g)\rangle$. In the query complexity model, we charge unit cost for a single query to the oracle and all other computations are free. We will assume that the input black-box polynomial $f : R^m \to R$ is given by such an unitary operator U_f.

All the quantum registers used during the computation can be initialised to $|0\rangle$. Then a k-query algorithm for a black-box ring is a sequence of $k+1$ unitary operators and k ring oracle operators: $U_0, Q_1, U_1, \cdots, U_{k-1}, Q_k, U_k$ where $Q_i \in \{O_R^a, O_R^m, O_F\}$ are the oracle queries and U_i's are unitary operators. The final step of the algorithm is to measure designated qubits and decide according to the measurement output.

3. Quantum Algorithm for Multilinear Identity Testing

In this section we describe our quantum algorithm for multilinear identity testing (MIT). Our algorithm is motivated by (and based on) the group commutativity testing algorithm of Magniez and Nayak [MN07]. We briefly explain the algorithm of Magniez-Nayak. Their problem is the

[1] Ambainis in [Amb07] show a quantum query complexity upper bound of $O(k^{m/m+1})$ for m-COLLISION problem.

following: given a black-box group G by a set of generators $g_1, g_2, \cdots, g_k$, the task is to find nontrivial upper bound on the quantum query complexity to determine whether G is commutative. The group operators (corresponding to the oracle) are O_G and $O_{G^{-1}}$.

Note that for this problem, there is a trivial classical algorithm (so as quantum) of query complexity $O(k^2)$. In an interesting paper, Pak showed a classical randomized algorithm of query complexity $O(k)$ for the same problem [Pak00]. Pak's algorithm is based on the following observation ([Pak00, Lemma 1.3]): Consider a subproduct $h = g_1^{e_1} g_2^{e_2} \cdots g_k^{e_k}$ where e_i' s are picked uniformly at random from $\{0, 1\}$. Then for any proper subgroup H of G, $\text{Prob}[h \notin H] \geq 1/2$.

One important step of the algorithm in [MN07] is a generalization of Pak's lemma. Let $\mathcal{V}_\ell$ be the set of all distinct element ℓ tuples of elements from $\{1, 2, \cdots, k\}$. For $u = (u_1, \cdots, u_\ell)$, define $g_u = g_{u_1} \cdot g_{u_2} \cdots g_{u_\ell}$. Let $p = \frac{\ell(\ell-1)+(k-\ell)(k-\ell-1)}{k(k-1)}$.

Lemma 3.1. [MN07] *For any proper subgroup K of G,* $\text{Prob}_{u \in \mathcal{V}_\ell}[g_u \notin K] \geq \frac{1-p}{2}$.

As a simple corollary of this lemma, Magniez and Nayak show in [MN07] that, if G is non abelian then for randomly picked u and v from $\mathcal{V}_\ell$ the elements g_u and g_v will not commute with probability at least $\frac{(1-p)^2}{4}$. Thus, for non abelian G there will be at least $\frac{(1-p)^2}{4}$ fraction of noncommuting pairs (u, v). Call such pairs as *marked pairs*. Next, their idea is to do a random walk in the space of all pairs and to decide whether there exists a marked pair. They achieved this by defining a random walk and quantizing it using [Sze04]. We briefly recall the setting from [MN07, Section 2.3], and the main theorem from [Sze04], which is the central to the analysis of Magniez-Nayak result.

3.0.1. *Quantum Walks.* Let P be an irreducible and aperiodic Markov chain on a graph $G = (V, E)$ with n vertices. A walk following such a Markov chain is always ergodic and has unique stationary distribution. Let $P(u, v)$ denote the transition probability from $u \rightarrow v$, and M be a set of marked nodes of V. The goal is to make a walk on the vertices of G following the transition matrix P and decide whether M is *nonempty*. Assume that every node $v \in V$ is associated with a database $D(v)$ from which we can determine whether $v \in M$. This search procedure is modelled by a quantum walk. To analyze the performance of the search procedure, we need to consider the cost of the following operations:

Set up Cost (S): The cost to set up $D(v)$ for $v \in V$.

Update Cost (U): The cost to update $D(v)$, i.e. to update from $D(v)$ to $D(v')$, where the move $v \rightarrow v'$ is according to the transition matrix P.

Checking Cost (C): To check whether $v \in M$ using $D(v)$.

The costs are specific to the application for e.g. it can be query complexity or time complexity. The problem that we consider or the group commutativity problem of Magniez-Nayak, concern about query complexity. The following theorem due to Szegedy gives a precise analysis of the total cost involved in the quantum walk.

Theorem 3.2. [Sze04] *Let P be the transition matrix of an ergodic, symmetric Markov Chain on a graph $G = (V, E)$ and δ be the spectral gap of P. Also, let M be the set of all marked vertices in V and $|M|/|V| \geq \epsilon > 0$, whenever M is nonempty. Then there is a quantum algorithm which determines whether M is nonempty with constant success probability and cost $S + O((U + C)/\sqrt{\delta\epsilon})$. S is the set up cost of the quantum process, U is the update cost for one step of the walk and C is the checking cost.*

Later, Magniez-Nayak-Ronald-Santha [MNRS07] improve the total cost of the quantum walk. We state their main result.

Theorem 3.3. [MNRS07] *Let P be the transition matrix of a reversible, ergodic Markov Chain on a graph $G = (V, E)$ and δ be the spectral gap of P. Also let M be the set of all marked vertices in V and $|M|/|V| \geq \epsilon > 0$, whenever M is nonempty. Then there is a quantum algorithm which determines whether M is nonempty and in that case finds an element of M, with constant success probability and cost of order $S + \frac{1}{\sqrt{\epsilon}}(\frac{1}{\sqrt{\delta}}U + C)$. S is the set up cost of the quantum process, U is the update cost for one step of the walk and C is the checking cost.*

The analysis of Magniez-Nayak [MN07] is based on Theorem 3.2. For our problem also, we follow similar approach.

3.1. Query Complexity Upper Bound

Now we describe our quantum algorithm for MIT. Our main technical contribution is a suitable generalization of Pak's lemma. For any $i \in [m]$, consider the set $R_i \subseteq R$ defined as follows:

$$R_i = \{u \in R \mid \forall (b_1, \cdots, b_{i-1}, b_{i+1}, \cdots, b_m) \in R^{m-1}, f(b_1, \cdots, b_{i-1}, u, b_{i+1}, \cdots, b_m) = 0\}$$

Clearly, if f is not a zero function from $R^m \to R$, then $|R_i| < |R|$. In the following lemma, we prove that if f is not a zero function then $|R_i| \leq |R|/2$.

Lemma 3.4. *Let R be any finite ring and $f(x_1, x_2, \cdots, x_m)$ be a multilinear polynomial over R such that $f = 0$ is not an identity for R. For $i \in [m]$ define*

$$R_i = \{u \in R \mid \forall (b_1, \cdots, b_{i-1}, b_{i+1}, \cdots, b_m) \in R^{m-1}, f(b_1, \cdots, b_{i-1}, u, b_{i+1}, \cdots, b_m) = 0\}.$$

Then R_i is an additive coset of a proper additive subgroup of R and hence $|R_i| \leq |R|/2$.

Proof. Write $f = A(x_1, \cdots, x_{i-1}, x_i, x_{i+1}, \cdots, x_m) + B(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_m)$ where A is the sum of all the monomials of f containing x_i and B is the sum of the rest of the monomials. Let v_1, v_2 be any two distinct elements in R_i. Then for any fixed $\bar{y} = (y_1, \cdots, y_{i-1}, y_{i+1}, \cdots, y_m) \in R^{m-1}$, consider the evaluation of A and B over the points $(y_1, \cdots, y_{i-1}, v_1, y_{i+1}, \cdots, y_m)$ and $(y_1, \cdots, y_{i-1}, v_2, y_{i+1}, \cdots, y_m)$ respectively. For convenience, we abuse the notation and write,

$$A(v_1, \bar{y}) + B(\bar{y}) = A(v_2, \bar{y}) + B(\bar{y}) = 0,$$

where $\bar{y}$ is an assignment to $x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_k$ and v_1, v_2 are the assignments to x_i respectively. Note that, as f is a multilinear polynomial, the above relation in turns implies that $A(v_1 - v_2, \bar{y}) = 0$.

Consider the set $\hat{R}_i$, defined as follows: Fix any $u^{(i)} \in R_i$,

$$\hat{R}_i = \{w - u^{(i)} \mid w \in R_i\}.$$

We claim that $\hat{R}_i$ is an (additive) subgroup of R. We only need to show that $\hat{R}_i$ is closed under the addition (of R). Consider $(w_1 - u^{(i)}), (w_2 - u^{(i)}) \in \hat{R}_i$. Then $(w_1 - u^{(i)}) + (w_2 - u^{(i)}) = (w_1 + w_2 - u^{(i)}) - u^{(i)}$. It is now enough to show that for any $\bar{y} \in R^{m-1}$, $f(w_1 + w_2 - u^{(i)}, \bar{y}) = 0$ (note that $w_1 + w_2 + u^{(i)}$ is an assignment to x_i). Again using the fact that f is multilinear, we can easily see the following:

$$f(w_1 + w_2 - u^{(i)}, \bar{y}) = A(w_1, \bar{y}) + A(w_2, \bar{y}) - A(u^{(i)}, \bar{y}) + B(\bar{y})$$

and,

$$A(w_1, \bar{y}) + A(w_2, \bar{y}) - A(u^{(i)}, \bar{y}) + B(\bar{y}) = A(w_2, \bar{y}) - A(u^{(i)}, \bar{y}) = 0.$$

Note that the last equality follows because x_2 and u are in R_i. Hence we have proved that $\hat{R}_i$ is a subgroup of R. So $R_i = \hat{R}_i + u^{(i)}$ i.e. R_i is a coset of $\hat{R}_i$ inside R. Also $|R_i| < |R|$ (f is not identically zero over R). Thus, finally we get $|R_i| = |\hat{R}_i| \leq |R|/2$. ∎

Our quantum algorithm is based on the algorithm of [MN07]. In the rest of the paper we denote by S_ℓ the set of all ℓ size *subsets* of $\{1, 2, \cdots, k\}$. We follow a quantization of a random walk on $S_\ell \times \cdots \times S_\ell = S_\ell^m$. For $u = \{u_1, u_2, \cdots, u_\ell\}$, define $r_u = r_{u_1} + \cdots + r_{u_\ell}$. Now, we suitably adapt Lemma 1 of [MN07] in our context. [2]

Let R be a finite ring given by a additive generating set $S = \{r_1, \cdots, r_k\}$. W.l.o.g. assume that r_1 is the zero element of R. Let $\hat{R}$ be a proper additive subgroup of $(R, +)$. Let j be the least integer in $[k]$ such that $r_j \notin \hat{R}$. Since $\hat{R}$ is a proper subgroup of R, such a j always exists.

Lemma 3.5. *Let $\hat{R} < R$ be a proper additive subgroup of R and T be an additive coset of $\hat{R}$ in R. Then* $\text{Prob}_{u \in S_\ell}[r_u \notin T] \geq \frac{1-p}{2}$*, where* $p = \frac{\ell(\ell-1)+(k-\ell)(k-\ell-1)}{k(k-1)}$.

Proof. Let j be the least integer in $[k]$ such that $r_j \notin \hat{R}$. Fix a set u of size ℓ such that $1 \in u$ and $j \notin u$. Denote by v the set obtained from u by deleting 1 and inserting j. This defines a one to one correspondence (matching) between all such pair of (u, v). Moreover $r_v = r_u + r_j$ (notice that $r_1 = 0$). Then at least one of the element r_u or r_v is not in T. For otherwise $(r_v - r_u) \in \hat{R}$ implying $r_j \in \hat{R}$, which is a contradiction.

Therefore,

$$\text{Prob}_{u \in S_\ell}[r_u \in T \mid j \in u \text{ xor } 1 \in u] \leq \frac{1}{2}.$$

For any two indices i, j,

$$\text{Prob}_{u \in S_\ell}[i, j \in u \text{ or } i, j \notin u] = \frac{\ell(\ell-1) + (k-\ell)(k-\ell-1)}{k(k-1)} = p.$$

Thus,

$$\text{Prob}_{u \in S_\ell}[r_u \in T] \leq (1-p)/2 + p \leq (1+p)/2.$$

This completes the proof. ∎

Let $T = R_i$ in Lemma 3.5, where R_i is as defined in Lemma 3.4.

Suppose $f = 0$ is not an identity for the ring R. Then, using Lemma 3.5, it is easy to see that, for $u_1, u_2, \cdots, u_m$ picked uniformly at random from S_ℓ, $f(r_{u_1}, \cdots, r_{u_m})$ is non zero with non-negligible probability. This is analogous to [MN07, Lemma 2]. We include a proof for the sake of completeness.

Lemma 3.6. *Let $f(x_1, \cdots, x_m)$ be a multilinear polynomial (in commuting or noncommuting indeterminates) over R such that $f = 0$ is not an identity for the ring R. Then,*

$$\text{Prob}_{u_1, \cdots, u_m \in S_\ell}[f(r_{u_1}, \cdots, r_{u_m}) \neq 0] \geq \left(\frac{1-p}{2}\right)^m.$$

Proof. For $i \in [m]$, let R_i be the additive coset defined in Lemma 3.4. The proof is by simple induction on m. The proof for the base case of the induction (i.e for $m = 1$) follows easily from the definition of R_i and Lemma 3.5. By induction hypothesis assume that the result holds for all t-variate multilinear polynomials g such that $g = 0$ is not an identity for R with $t \leq m - 1$.

[2] Notice that in [MN07], the author consider the set of all ℓ tuples instead of subsets. This is important for them as they work in non abelian structure in general (where order matters). But we will be interested only over additive abelian structure of a ring and thus order does not matter for us.

Consider the given multilinear polynomial $f(x_1, x_2, \cdots, x_m)$. Then, by Lemma 3.4, R_m is a coset of an additive subgroup $\hat{R}_m$ inside R. Pick $u_m \in \mathrm{S}_\ell$ uniformly at random. If $f = 0$ is not an identity on R then by Lemma 3.5 we get $r_{u_m} \notin R_m$ with probability at least $\frac{1-p}{2}$. Let $g(x_1, x_2, \cdots, x_{m-1}) = f(x_1, \cdots, x_{m-1}, r_{u_m})$. Since $r_{u_m} \notin R_m$ with probability at least $\frac{1-p}{2}$, it follows that $g = 0$ is not an identity on R with probability at least $\frac{1-p}{2}$. Given that g is not an identity for R, by induction hypothesis we have that, $\mathrm{Prob}_{u_1,\cdots,u_{m-1} \in S_\ell}[g(r_{u_1}, \cdots, r_{u_{m-1}}) \neq 0] \geq \left(\frac{1-p}{2}\right)^{m-1}$. Hence we get, $\mathrm{Prob}_{u_1,\cdots,u_m \in S_\ell}[f(r_{u_1}, \cdots, r_{u_m}) \neq 0] \geq \left(\frac{1-p}{2}\right)^m$, which proves the lemma. ■

We observe two simple consequences of Lemma 3.6. Notice that $\frac{1-p}{2} = \frac{\ell(k-\ell)}{k(k-1)}$. Letting $\ell = 1$ we get $\frac{1-p}{2} = 1/k$, and Lemma 3.6 implies that if $f = 0$ is not an identity for R then $f(a_1, \cdots, a_m) \neq 0$ for one of the k^m choices for the a_i from the generating set $\{r_1, \cdots, r_k\}$.

Letting $\ell = k/2$ in Lemma 3.6, we get $\frac{1-p}{2} \geq 1/4$. Hence we obtain the following randomized test which makes $4^m mk$ queries.

Corollary 3.7. *There is a randomized $4^m mk$ query algorithm for* MIT *with constant success probability, where f is m-variate and R is given by an additive generating set of size k. This can be seen as a generalization of Pak's $O(k)$ query randomized test for group commutativity.*

We use Lemma 3.6 to design our quantum algorithm. Technically, our quantum algorithm is similar to the one described in [MN07]. The Lemma 3.6 is used to guarantee that there will at least $\left(\frac{1-p}{2}\right)^m$ fraction of *marked points* in the space S_ℓ^m i.e. the points where f evaluates to non-zero. The underlying graph in our random walk is a Johnson Graph and our analysis require some simple modification of the analysis described in [MN07].

3.1.1. *Random walk on S_ℓ.* Our random walk can be described as a random walk over a graph $G = (V, E)$ defined as follows: The vertices of G are all possible ℓ subsets of $[k]$. Two vertices are connected by an edge whenever the corresponding sets differ by exactly one element. Notice that G is a connected $\ell(k-\ell)$-regular Johnson graph, with parameter $(k, \ell, \ell-1)$ [BCN89]. Let P be the normalized adjacency matrix of G with rows and columns are indexed by the subsets of $[k]$. Then $P_{XY} = 1/\ell(k-\ell)$ if $|X \cap Y| = \ell - 1$ and 0 otherwise. It is well known that the spectral gap δ of P ($\delta = 1 - \lambda$, where λ is the second largest eigenvalue of P) is $\Omega(1/\ell)$ for $\ell \leq k/2$ [BCN89]. Now we describe the random walk on G.

Let the current vertex is $u = \{u_1, u_2, \cdots, u_\ell\}$ and $r_u = r_{u_1} + r_{u_2} + \cdots + r_{u_\ell}$. With probability $1/2$ stay at u and with probability $1/2$ do the following: randomly pick $u_i \in u$ and $j \in [k] \setminus u$. Then move to vertex v such that v is obtained from u by removing u_i and inserting j. Compute r_v by simply subtracting r_{u_i} from r_u and adding r_j to it. That will only cost 2 oracle access. Staying in any vertex with probability $1/2$ ensures that the random walk is ergodic. So the stationary distribution of the random walk is always uniform. It is easy to see that the transition matrix of the random walk is $A = (I + P)/2$ where I is the identity matrix of suitable dimension. So the spectral gap of the transition matrix A is $\hat{\delta} = (1 - \lambda)/2 = \delta/2$.

The query complexity analysis is similar to the analysis of Magniez-Nayak. But to fit it with our requirement, we need some careful parameter setting. We include a brief self-contained proof.

Theorem 3.8. *Let R be a finite black-box ring given as an oracle and $f(x_1, \cdots, x_m)$ be a multilinear polynomial over R given as a black-box. Moreover let $\{r_1, \cdots, r_k\}$ be a given additive* *generating set for R. Then the quantum query complexity of testing whether f is an identity for R, is $O(m(1+\alpha)^{m/2}k^{\frac{m}{m+1}})$, assuming $k \geq (1+1/\alpha)^{m+1}$.*

Proof. **Setup cost(S):** For the quantum walk step we need to start with an uniform distribution on S_ℓ^m. With each $u \in S_\ell$, we maintain a quantum register $|d_u\rangle$ that computes r_u. So we need to prepare the following state $|\Psi\rangle$:

$$|\Psi\rangle = \frac{1}{\sqrt{|S_\ell^m|}} \sum_{u_1,u_2,\cdots,u_m \in S_\ell^m} |u_1, r_{u_1}\rangle \otimes |u_2, r_{u_2}\rangle \otimes \cdots \otimes |u_m, r_{u_m}\rangle.$$

It is easy to see that to compute any r_{u_j}, we need $\ell - 1$ oracle access to the ring oracle. Since in each of m independent walk, quantum queries over all choices of u will be made in parallel (using quantum superposition), the total query cost for setup is $m(\ell-1)$.

Update cost(U): It is clear from the random walk described in the section 3.1.1, that the update cost over S_ℓ is only 2 oracle access. Thus for the random walk on S_ℓ^m which is just m independent random walks, one on each copy of S_ℓ, we need a total update cost $2m$.[3]

Checking cost(C): To check whether f is zero on a point during the walk, we simply query the oracle for f once.

Recall from Szegedy's result [Sze04] (as stated in Theorem 3.2), the total cost for query complexity is $Q = S + \frac{1}{\sqrt{\hat{\delta}\epsilon}}(U+C)$ where $\epsilon = \left(\frac{1-p}{2}\right)^m$ is the proportion of the marked elements and $\hat{\delta}$ is the spectral gap of the transition matrix A described in section 3.1.1. Combining together we get, $Q \leq m\left[(\ell-1) + \frac{3}{\sqrt{\hat{\delta}\epsilon}}\right]$. From the random walk described in the section 3.1.1, we know that $\hat{\delta} \geq \frac{1}{2\ell}$. Hence, $Q \leq m\left[(\ell-1) + \frac{3\sqrt{2\ell}}{\left(\frac{1-p}{2}\right)^{\frac{m}{2}}}\right]$. Notice that, $\frac{1-p}{2} = \frac{\ell}{k}\left(\frac{1-\frac{\ell}{k}}{1-\frac{1}{k}}\right)$. Substituting for $\frac{1-p}{2}$ we get, $Q \leq m\left[(\ell-1) + 3\sqrt{2}k^{m/2}\frac{1}{\ell^{\frac{m-1}{2}}\left(\frac{k-\ell}{k-1}\right)^{m/2}}\right]$. We will choose a suitably small $\alpha > 0$ so that $\frac{k-1}{k-\ell} < 1+\alpha$. Then we can upper bound Q as follows. $Q \leq m\left[(\ell-1) + 3\sqrt{2}\cdot(1+\alpha)^{m/2}k^{m/2}\frac{1}{\ell^{\frac{m-1}{2}}}\right]$. Now our goal is to minimize Q with respect to ℓ and α. For that we choose $\ell = k^t$ where we will fix t appropriately in the analysis. Substituting $\ell = k^t$ we get, $Q \leq m\left[(k^t-1) + 3\sqrt{2}\cdot(1+\alpha)^{m/2}t^{1/2}k^{\frac{m-(m-1)t}{2}}\right]$. Choosing $t = (m/(m+1))$, we can easily see that the query complexity of the algorithm is $O(m(1+\alpha)^{m/2}k^{\frac{m}{m+1}})$. Finally, recall that we need choose an $\alpha > 0$ so that $\frac{k-1}{k-\ell} \leq 1+\alpha$. Clearly, it suffices to choose α so that $(1+\alpha)\ell \leq \alpha k$. Letting $\ell = k^{m/m+1}$ we get the constraint $(1+1/\alpha)^{m+1} \leq k$ which is satisfied if $e^{(m+1)/\alpha} \leq k$. We can choose $\alpha = \frac{m+1}{\ln k}$. ■

Remark 3.9. The choice of α in the above theorem shows some trade-offs in the query complexity between the parameters k and m. For constant m notice that this gives us an $O(k^{m/m+1})$ query complexity upper bound for the quantum algorithm, which is similar to the best known query upper bound for m-COLLISION [Amb07], when the problem instance is a function $f : [k] \to [k]$.

Generalized Multilinear Identity Testing (GMIT**):** We now consider a variant of the MIT problem, which we call GMIT (for generalized-MIT).

[3] In [MN07] the underlying group operation is not necessarily commutative (it is being tested for commutativity). Thus the update cost is more.

Let $f : R^m \to R$ be a black-box multilinear polynomial. Consider any *additive subgroup* A of the black-box ring R, given by a set of generators $r_1, r_2, \cdots, r_k$, so that $A = \{\sum_i \beta_i r_i \mid \beta_i \in \mathbb{Z}\}$. The GMIT$(R, A, f)$ problem is the following: test whether a black-box multilinear polynomial f is an identity for A. In other words, we need to test if $f(a_1, \cdots, a_m) = 0$ for all $a_i \in A$.

It is easy to observe that the quantum algorithm actually solves GMIT and the correctness proof and analysis given in Theorem 3.8 also hold for GMIT problem. We summarize this observation in the following theorem.

Theorem 3.10. *Let R be a black-box finite ring given by ring oracles and $A = \langle r_1, r_2, \cdots, r_k\rangle$ be an* additive subgroup *of R given by generators $r_i \in R$. Let $f(x_1, x_2, \cdots, x_m)$ be a black-box multilinear polynomial $f : R^m \to R$. Then there is a quantum algorithm with query complexity $O(m(1+\alpha)^{m/2}k^{\frac{m}{m+1}})$ for the* GMIT(R, A, f) *problem (assuming $k \geq (1 + 1/\alpha)^{m+1}$).*

4. Query Complexity Lower Bound

In this section we show that GMIT problem of multilinear identity testing for additive subgroups of a black-box ring (described in Section 3.1.1), is at least as hard as m-SPLIT COLLISION (again, m-SPLIT COLLISION problem is defined in Section 1). Also, the well-known m-COLLISION problem can be easily reduced to m-SPLIT COLLISION problem using a simple randomized reduction. In the following lemma, we briefly state the reduction.

Lemma 4.1. *There is a randomized reduction from* m-COLLISION *to* m-SPLIT COLLISION *with success probability close to e^{-m}.*

Proof. Let $f : [k] \to [k]$ be a 'yes' instance of m-COLLISION, and suppose $f^{-1}(i) = \{i_1, i_2, \cdots, i_m\}$. To reduce this instance to m-SPLIT COLLISION we pick a random m-partition $I_1, I_2, \cdots, I_m$ of the domain $[k]$ with each $|I_j| = k/m$. It is easy to see that, with probability close to e^{-m}, the set $\{i_1, i_2, \cdots, i_m\}$ will be a split collision for the function f. ■

Consequently, showing a quantum lower bound of $\Omega(k^\alpha)$ for m-COLLISION will imply a quantum lower bound of $\Omega(k^\alpha/e^m)$ for m-SPLIT COLLISION. It will also show similar lower bound for GMIT because of our reduction.

If $f : [k] \to [k]$ is an instance of m-SPLIT COLLISION problem, then the classical randomized query complexity lower bound is $\Omega(k)$. This is observed in [MN07] for $m = 2$. Due to our reduction, we get similar randomized query complexity lower bound for GMIT.

Currently the best known quantum query complexity lower bound for m-COLLISION problem is $\Omega(k^{2/3})$ (in the case $m = 2$) [AS04]. Thus we obtain the same explicit lower bound for m-SPLIT COLLISION problem due to the random reduction from m-COLLISION to m-SPLIT COLLISION. It also implies quantum query complexity lower bound for GMIT.

Our reduction from m-SPLIT COLLISION to GMIT problem is based on some new automata theoretic ideas. We first describe necessary automata theoretic ideas those are useful for our reduction.

4.1. Automata theory background

We recall some standard automata theory notations (see, for example, [HU78]). Fix a finite automaton $A = (Q, \Sigma, \delta, q_0, q_f)$ which takes as input strings in Σ^*. Q is the set of states of A, Σ is the alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, and q_0 and q_f are the initial and final states respectively (throughout, we only consider automata with unique accepting states). For each letter $b \in \Sigma$, let $\delta_b : Q \to Q$ be the function defined by: $\delta_b(q) = \delta(q, b)$. These functions generate a submonoid of the monoid of all functions from Q to Q. This is the transition monoid of the automaton A and is well-studied in automata theory: for example, see [Str94, page 55]. We now define the 0-1 matrix $M_b \in \mathbb{F}^{|Q|\times|Q|}$ as follows: $M_b(q, q') = 1$ if $\delta_b(q) = q'$, and 0 otherwise.

The matrix M_b is simply the adjacency matrix of the graph of the function δ_b. As the entries of M_b are only zeros and ones, we can consider M_b to be a matrix over any field $\mathbb{F}$.

Furthermore, for any $w = w_1w_2 \cdots w_k \in \Sigma^*$, we define the matrix M_w to be the matrix product $M_{w_1}M_{w_2} \cdots M_{w_k}$. If w is the empty string, define M_w to be the identity matrix of dimension $|Q| \times |Q|$. For a string w, let δ_w denote the natural extension of the transition function to w. If w is the empty string, δ_w is simply the identity function. It is easy to check that: $M_w(q, q') = 1$ if $\delta_w(q) = q'$ and 0 otherwise. Thus, M_w is also a matrix of zeros and ones for any string w. Also, $M_w(q_0, q_f) = 1$ if and only if w is accepted by the automaton A. We now describe the reduction.

Theorem 4.2. *The* m-SPLIT COLLISION *problem reduces to* GMIT *problem for additive subgroups of black-box rings.*

Proof. An instance of m-SPLIT COLLISION is a function $f : [k] \to [k]$ given as an oracle, where we assume w.l.o.g. that $k = nm$. Divide $\{1, 2, \cdots, k\}$ into m intervals $I_1, I_2, \cdots, I_m$, each containing n consecutive points of $[k]$. Recall from Section1 that, f is said to have an m-split collision if for some $j \in [k]$ we have $|f^{-1}(j)| = m$ and $|f^{-1}(j) \cap I_i| = 1$ for each interval I_i.

Consider the alphabet $\Sigma = \{b, c, b_1, b_2, \cdots, b_m\}$. Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, q_f)$ be a deterministic finite state automaton that accepts all strings $w \in \Sigma^*$ such that each $b_j, 1 \leq j \leq m$ occurs at least once in w. It is easy to see that such an automaton with a single final state q_f can be designed with total number of states $|Q| = 2^{O(m)} = t$. W.l.o.g. let the set of states Q be renamed as $\{1, 2, \cdots, t\}$, where 1 is the initial state and t is the final state.

For each letter $a \in \Sigma$, let M_a denote the $t \times t$ transition matrix for δ_a (as defined in Section 4.1). Since each M_a is a $t \times t$ 0-1 matrix, each M_a is in the ring $\mathcal{M}_t(\mathbb{F}_2)$ of $t \times t$ matrices with entries from the field $\mathbb{F}_2$. Let R denote the k-fold product ring $(\mathcal{M}_t(\mathbb{F}_2))^k$. Clearly, R is a finite ring (which is going to play the role of the black-box ring in our reduction). We now define an additive subgroup T of R, where we describe the generating set of T using the m-SPLIT COLLISION instance f.

For each index $i \in [k]$, define an k-tuple $T_i \in R$ as follows. If $i \neq f(i)$, then define $T_i[i] = M_b$, $T_i[f(i)] = M_{b_j}$ (where $i \in I_j$) and and for each index $s \notin \{i, f(i)\}$ define $T_i[s] = M_c$. For $i = f(i)$, define $T[i] = M_{b_j}$ $(i \in I_j)$ and the rest of the entries as M_c. The additive subgroup of R that we consider is $T = \langle T_1, T_2, \cdots, T_k\rangle$ generated by the $T_i, 1 \leq i \leq k$.

Furthermore, define two $t \times t$ matrices A and B in $\mathcal{M}_t(\mathbb{F}_2)$ as follows. Let $A[1, 1] = 1$ and $A[u, \ell] = 0$ for $(u, \ell) \neq (1, 1)$. For the matrix B, let $B[t, 1] = 1$ and $B[u, \ell] = 0$ for $(u, \ell) \neq (t, 1)$.

Claim 1. Let $w = w_1w_2 \cdots w_s \in \Sigma^*$ be any string. Then the automaton $\mathcal{A}$ defined above accepts w if and only if the matrix $AM_{w_1}M_{w_2} \cdots M_{w_s}B$ is nonzero.

Proof of Claim By definition of the matrices M_a, the $(1, t)^{th}$ entry of the product $M_{w_1}M_{w_2} \cdots M_{w_s}$ is 1 if and only if w is accepted by $\mathcal{A}$. By definition of the matrices A and B the claim follows immediately.

Now, consider the polynomial $P(x_1, x_2, \cdots, x_m)$ with coefficients from the matrix ring R defined as follows:

$$P(x_1, x_2, \cdots, x_m) = \bar{A}x_1x_2 \cdots x_m\bar{B},$$

where $\bar{A} = (A, A, \ldots, A) \in R$ and $\bar{B} = (B, B, \cdots, B) \in R$ are k-tuples of A's and B's respectively. We claim that the multilinear polynomial $P(x_1, x_2, \cdots, x_m) = 0$ is an identity for the additive subgroup T if and only if f has no m-split collision.

Claim 2. $P(x_1, \cdots, x_m) = 0$ is an identity for the additive subgroup $T = \langle T_1, \cdots, T_k \rangle$ if and only if f has no m-split collision. In other words, GMIT(R, T, P) is an 'yes' instance if and only if f has no m-split collision.

Proof of Claim Suppose f has an m-split collision. Specifically, let $i_j \in I_j$ ($1 \leq j \leq m$ and $i_1 < i_2 < \cdots < i_m$) be indices such that $f(i_1) = \cdots = f(i_m) = \ell$. In the polynomial P, we substitute the indeterminate x_j by T_{i_j}.

Then $P(T_{i_1}, T_{i_2}, \cdots, T_{i_m}) = \bar{A}M\bar{B}$, where $M = T_{i_1} \cdots T_{i_m}$. M is a k-tuple of $t \times t$ matrices such that the ℓ^{th} component of M is $\prod_{j=1}^{m} M_{b_j}$ where $i_j \in I_j$. Since $b_{i_1} b_{i_2} \cdots b_{i_m} \in \Sigma^*$ is a length m-string containing all the b_j's it will be accepted by the automaton $\mathcal{A}$. Consequently, the $(q_0, q_f)^{th}$ entry of the matrix M, which is the $(1, t)^{th}$ entry, is 1 (as explained in Section 4.1). It follows that the $(1, 1)$ entry of the matrix AMB is 1. Hence $P = 0$ is not an identity over the additive subgroup T.

For the other direction, assume that f has no m-split collision. We need to show that $P = 0$ is an identity for the ring T. For any m elements $S_1, S_2, \cdots, S_m \in T$ consider $P(S_1, S_2, \cdots, S_m) = \bar{A}S_1 S_2 \cdots S_m \bar{B}$. Since Each S_j is an $\mathbb{F}_2$-linear combination of the generators $T_1, \cdots, T_k$, it follows by distributivity in the ring R that $P(S_1, S_2, \cdots, S_m)$ is an $\mathbb{F}_2$-linear combination of terms of the form $P(T_{k_1}, T_{k_2}, \cdots, T_{k_m})$ for some m indices $k_1, \cdots, k_m \in [k]$. Thus, it suffices to show that $P(T_{k_1}, T_{k_2}, \cdots, T_{k_m}) = 0$.

Let $\hat{T} = T_{k_1} T_{k_2} \cdots T_{k_m}$. Then, for each $j \in [k]$ we have $\hat{T}[j] = T_{k_1}[j] T_{k_2}[j] \cdots T_{k_m}[j]$. Since f has no m-split collision, for each $j \in [N]$ the set of matrices $\{M_{b_1}, M_{b_2}, \cdots, M_{b_m}\}$ is *not* contained in the set $\{T_1[j], T_2[j], \cdots, T_k[j]\}$. Thus, $\hat{T}[j] = T_{k_1}[j] T_{k_2}[j] \cdots T_{k_m}[j]$ is a product of matrices $M_{w_1} M_{w_2} \cdots M_{w_m}$ for a word $w = w_1 w_2 \cdots w_m$ that is not accepted by $\mathcal{A}$. It follows from the previous claim that $A\hat{T}[j]B = 0$. Hence $P(T_{k_1}, T_{k_2}, \cdots, T_{k_m}) = 0$ which completes the proof. ■

In Section 3.1, we have already shown a quantum algorithm of query complexity $O(k^{\frac{m}{m+1}})$ for MIT (m is a constant). This bound holds as well for GMIT. We conclude this section by showing that any algorithm of query complexity $q(k, m)$ (q is any function) for GMIT will give an algorithm of similar query complexity for m-COLLISION problem. In particular an algorithm for GMIT of query complexity $k^{o(m/m+1)}$ will improve the best known algorithm for m-COLLISION problem due to Ambainis [Amb07]. The following corollary is an easy consequence of Theorem 4.2.

Corollary 4.3. *Let* $f : [k] \to [k]$ *be an instance of* m-SPLIT COLLISION *problem and* GMIT(R, T, P) *be an instance of* GMIT *problem, where the multilinear polynomial* $P : R^m \to R$ *and* T *is an additive subgroup of* G *given by* k *generators. Then, if we have a quantum algorithm of query complexity* $q(k, m)$ *for* GMIT *problem, we will have a quantum algorithm for* m-SPLIT COLLISION *with query complexity* $O(q(k, m))$.

Proof. Let $\mathcal{A}$ be an algorithm for GMIT with quantum query complexity $q(k, m)$. Given an instance of m-SPLIT COLLISION, the generators for the additive subgroup T is indexed by $1, 2, \cdots, k$ (as defined in the proof of Theorem 4.2). Also, define the polynomial $P(x_1, x_2, \cdots, x_m)$ So the inputs of our GMIT problem are $1, 2, \cdots, k$ and P. Using the algorithm $\mathcal{A}$, we define another algorithm $\mathcal{A}'$ which does the following. When $i \in [k]$ is invoked by $\mathcal{A}$ for the ring operation, the algorithm $\mathcal{A}'$ constructs the generator T_i by making only one query to the oracle for f. One more query to the f-oracle is required to erase the output. Moreover, if $\mathcal{A}$ wants to check whether the output of the ring operation is a valid generator (say T_j for some j), then also $\mathcal{A}'$ uses just two queries to the oracle of f. Thus we have an algorithm $\mathcal{A}'$ for m-SPLIT COLLISION with query complexity $4q(k)$. ■

Recall that the best known lower bound for m-SPLIT COLLISION problem is $\Omega(k^{2/3})$. Then, combining Theorem 4.2 and Corollary 4.3, we get $\Omega(k^{2/3})$ quantum query lower bound for GMIT problem.

Acknowledgement

We thank Ashwin Nayak for comments and suggestions. We are grateful to the anonymous STACS'09 referees for useful comments.

References

[ADM06] Vikraman Arvind, Bireswar Das, and Partha Mukhopadhyay. The complexity of black-box ring problems. In *COCOON*, pages 126–135, 2006.

[Amb07] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM J. Comput.*, 37(1):210–239, 2007.

[AMS08] Vikraman Arvind, Partha Mukhopadhyay, and Srikanth Srinivasan. New results on noncommutative and commutative polynomial identity testing. In *IEEE Conference on Computational Complexity*, pages 268–279, 2008.

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004.

[BCN89] A.E. Brouwer, A.M. Cohen, and A.N. Neumaier. Distance Regular Graphs. *Springer-Verlag*, pages 255–256, 1989.

[HU78] John E. Hopcroft and Jeffrey D. Ullman. Introduction to Automata Theory, Languages and Computation. *Addison-Wesley*, 1978.

[MN07] Frederic Magniez and Ashwin Nayak. Quantum complexity of testing group commutativity. *Algorithmica*, 48(3):221–232, 2007.

[MNRS07] Frederic Magniez, Ashwin Nayak, Jeremie Roland, and Miklos Santha. Search via quantum walk. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 575–584, 2007.

[Pak00] Igor Pak. Testing commutativity of a group and the power of randomization. *Electronic Version at: http://www-math.mit.edu/pak/research.html*, 2000.

[Str94] Howard Straubing. Finite Automata, Formal Logic, and Circuit Complexity. *Progress in Theoretical Computer Science, Birkhser*, 1994.

[Sze04] Mario Szegedy. Quantum speed-up of markov chain based algorithms. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 32–41, 2004.