ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/221426770

The Complexity of Black-Box Ring Problems

Conference Paper · August 2006

DOI: 10.1007/11809678_15 · Source: DBLP

CITATIONS
6

READS
81

3 authors, including:

Vikraman Arvind
The Institute of Mathematical Sciences
163 PUBLICATIONS **1,128** CITATIONS

SEE PROFILE

Bireswar Das
Indian Institute of Technology Gandhinagar
20 PUBLICATIONS **131** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Structural Parameterizations of Intractable Graph Problems View project

All content following this page was uploaded by Bireswar Das on 30 January 2014.

The user has requested enhancement of the downloaded file.

The Complexity of Black-Box Ring Problems

V. Arvind, Bireswar Das and Partha Mukhopadhyay

Institute of Mathematical Sciences
C.I.T Campus,Chennai 600 113, India
{arvind,bireswar,partham}@imsc.res.in

Abstract. We study the complexity of some computational problems on finite black-box rings whose elements are encoded as strings of a given length and the ring operations are performed by a black-box oracle. We give a polynomial-time quantum algorithm to compute a basis representation for a given black-box ring. Using this result we obtain polynomial-time quantum algorithms for several natural computational problems over black-box rings.

1 Introduction

Finite rings often play an important role in the design of algebraic algorithms. Berlekamp's randomized algorithm for factoring univariate polynomials over finite fields is a classic example [VZG03]. More recently, as explained in [AS05], the celebrated AKS primality test [AKS04] can be cast in a ring-theoretic framework. Lenstra's survey [Le92] gives other algorithmic examples. Recently, [AS05,KS05] have shown that Graph Isomorphism and Integer Factoring are polynomial-time reducible to Ring Isomorphism, where the rings are input in the *basis representation* (defined in Section 2).

As pointed out in [AS05], the representation of the finite ring is crucial to complexity of Ring Isomorphism. In this paper, we explore the complexity of ring-theoretic problems where the finite rings are given by a *black-box* (definitions are in Section 2). In a sense, a black-box ring is representation free. This model is motivated by finite black-box groups introduced by Babai and Szemerédi [BS84,Ba92] and intensively studied in algorithmic group theory. It turns out, surprisingly, that there is a polynomial-time *quantum* algorithm to obtain a basis representation for a given black-box ring. Thus, upto quantum polynomial time, the two representations are equivalent. A key procedure we use is an almost-uniform random sampling algorithm for finite black-box rings. Our algorithm is quite simple as compared to Babai's sampling algorithm for black-box groups [Ba91]. Additionally, if the characteristic of the ring is small (polynomially bounded in the input size), then we actually have an NC sampling algorithm. In contrast, for black-box groups it is still open if there is an NC sampler [Ba91].

It is an open question whether there is a randomized polynomial-time algorithm to recover a basis representation from the black-box oracle. The main obstacle is *additive independence testing* in a black-box ring R: given $r_1, r_2, \cdots, r_k \in R$ is there a nontrivial solution to $\sum_{i=0}^{l} x_i r_i = 0$. There is no known classical polynomial-time algorithm for this problem. However, it fits nicely in the hidden subgroup framework and we can solve it in quantum polynomial time as the additive group of R is abelian. As application we obtain quantum algorithms for some black-box rings problems in Sections 5 and 6.

2 Preliminaries

A *finite ring* is a triple $(R, +, *)$, where R is a finite nonempty set such that $(R, +)$ is a commutative group and $(R, *)$ is a semigroup, such that $*$ distributes over addition. A *subring* R' is a subset of R that is a ring under the same operations. Let $S \subset R$. The subring *generated* by S is the smallest subring $\langle S \rangle$ of R containing S. Thus, if $R = \langle S \rangle$ then every element of R can be computed by an arithmetic circuit that takes as input the generators from S and has the ring operations $+$ and $*$

as the gate operations. It is easy to see that every finite ring R has a generator set of size at most $\log |R|$.

A *ring oracle* R takes queries of the form $(q, x, y, +)$, $(q, x, y, *)$, (q, x, addinv), and (q, addid) where q, x, y are strings of *equal length* over Σ. The response to each of these queries is either a string of length $|q|$ or a symbol indicating invalid query. Let $R(q)$ be the set of $x \in \Sigma^{|q|}$ for which (q, x, addinv) is a valid query. Then R is a ring oracle if $R(q)$ is either empty or a ring with ring operations described by the responses to the above queries (where the response to (q, addid) is the string encoding additive identity). The oracle R defines the rings $R(q)$. The subrings of $R(q)$, given by generator sets will be called *black-box rings*.

A *basis representation* of a finite ring R [Le92,KS05] is defined as follows: the additive group $(R, +)$ is described by a direct sum $(R, +) = \mathbb{Z}_{m_1} e_1 \oplus \mathbb{Z}_{m_2} e_2 \oplus \cdots \oplus \mathbb{Z}_{m_n} e_n$, where m_i are the additive orders of e_i. Multiplication in R is specified by the products $e_i e_j = \sum_{k=1}^{n} \gamma_{ij}^k e_k$, for $1 \leq i, j \leq n$, where $\gamma_{ijk} \in \mathbb{Z}_{m_k}$.

Details about the classical and quantum complexity classes discussed in this paper can be found in [BDG88a,BDG88b,BV97].

3 Random sampling from a black-box ring

In this section we present a simple polynomial-time sampling algorithm that samples almost uniformly from finite black box rings. Let R be a black-box ring generated by S.

We will describe a randomized algorithm that takes S as input and with high probability computes an *additive generating set* T for $(R, +)$. I.e. every element of R is expressible as a sum of elements of T.

Using this additive generator set T it turns out that we can easily sample from $(R, +)$. We first prove this fact in the following lemma.

Lemma 1. *Let R be a finite black-box ring given by an additive generator set $\{r_1, r_2, \cdots, r_n\}$. Then there is a polynomial-time almost uniform sampling algorithm for R using $O(n \log(|R|/\epsilon))$ ring additions and $O(n \log(|R|/\epsilon))$ random bits.*

Proof. Let $k_1, k_2, \cdots, k_n$ be the additive orders of $\{r_1, r_2, \cdots, r_n\}$ in R. Define the onto homomorphism $\xi : \mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \times \cdots \mathbb{Z}_{k_n} \longrightarrow R$ as $\xi(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} x_i r_i$. Suppose we can almost uniformly sample from $\mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \times \cdots \mathbb{Z}_{k_n}$. Let $(x_1, x_2, \cdots, x_n)$ be a sample point from $\mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \times \cdots \mathbb{Z}_{k_n}$. Since ξ is an onto homomorphism, $\xi^{-1}(r)$ has the same cardinality for each $r \in R$. Hence, $\xi(x_1, x_2, \cdots, x_n)$ is an almost uniformly distributed random element from R.

Thus, it suffices to show that we can almost uniformly sample from $\mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \times \cdots \mathbb{Z}_{k_n}$. Notice that we do not know the k_i's. But we know an upper bound, namely 2^m, for each of $k_1, k_2, \ldots, k_n$. Take a suitably large $M > 2^m$ to be fixed later in the analysis. The sampling is as follows: pick $(x_1, x_2, \cdots, x_n)$ uniformly at random from $[M]^n$ and output $\sum x_i r_i$. Let $(a_1, a_2, \cdots, a_n) \in \mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \times \cdots \mathbb{Z}_{k_n}$ and let $p = \text{Prob}[x_i \equiv a_i \bmod k_i, 1 \leq i \leq n]$.

The x_i for which $x_i \equiv a_i \bmod k_i$ are precisely $a_i, a_i + k_i, \cdots, a_i + k_i \lfloor (M - a_i)/k_i \rfloor$. Let $M_i' = \lfloor (M - a_i)/k_i \rfloor$. Then $p = (\prod_i M_i')/M^n$. Clearly, $p \leq \prod_i (1/k_i)$. Furthermore, it is also easy to check that $p \geq (1 - 2^{m+1}/M)^n \cdot \prod_i (1/k_i)$. Choose $M > (n 2^{m+1})/\epsilon$. Then $p \geq (1 - \epsilon) \prod_i (1/k_i)$, implying that $\sum x_i r_i$ is ϵ-uniformly distributed in R. The number of ring additions required is $O(n \log((n 2^{m+1})/\epsilon))$ which is $O(n \log(|R|/\epsilon))$. The number of random bits used is also $O(n \log(|R|/\epsilon))$.

Let $R = \langle S \rangle$ be a black-box ring. Denote by $\hat{R}$ the additive subgroup of $(R, +)$ generated by S. I.e. $\hat{R}$ is the smallest additive subgroup of $(R, +)$ containing S. Notice that $\hat{R}$ could be a proper subset of R, and $\hat{R}$ need not be a subring of R in general.

Lemma 2. *Let $R = \langle S \rangle$ be a black-box ring, and $\hat{R}$ be the additive subgroup of $(R, +)$ generated by S. Then $\hat{R} = R$ if and only if $\hat{R}$ is closed under the ring multiplication: i.e. $\hat{R} r \subseteq \hat{R}$ for each $r \in S$.*

Proof. If $\hat{R} = R$ then the condition is obviously true. Conversely, notice that $\hat{R}r \subseteq \hat{R}$ for each $r \in S$ implies that $\hat{R}$ is closed under multiplication and hence $\hat{R} = R$.

Theorem 1. *There is a randomized algorithm that takes as input a black-box ring $R = \langle S \rangle$ and with high probability computes an additive generating set for $(R, +)$ and runs in time polynomial in the input size.*

Proof. The algorithm starts with S and proceeds in stages by including new randomly picked elements into the set at every stage. Thus, it computes a sequence of subsets $S = S_1 \subseteq S_2 \subseteq \ldots \subseteq S_\ell$, where ℓ will be appropriately fixed in the analysis. Let H_i denote the additive subgroup generated additively by S_i for each i. Notice that $H_1 = \hat{R}$. We now describe stage i of the procedure where, given S_i, the algorithm will compute S_{i+1}. First, notice that for each $r \in S$, $H_i r$ is a subgroup of $(R, +)$ that is additively generated by $\{xr \mid x \in S_i\}$. Thus, we can use Lemma 1 to ϵ-uniformly sample in polynomial time an element x_{ir} from $H_i r$, for each $r \in S$ (for a suitable ϵ to be chosen in the analysis). We now define the set $S_{i+1} = S_i \cup \{x_{ir} \mid r \in S\}$. Clearly, if ℓ is polynomially bounded then the above sampling procedure outputs S_ℓ in polynomial time. It thus remains to analyze the probability that S_ℓ additively generates $(R, +)$.
Claim. For $\ell = 4m + 1$ and $\epsilon = 1/2^m$ the probability that S_ℓ additively generates $(R, +)$ is at least $1/6$.
Proof of Claim. The proof is a simple application of Markov's inequality. We define indicator random variables $Y_i, 1 \leq i \leq 4m$ as follows: $Y_i = 1$ if $H_i = H_{i+1}$ and $H_i \neq R$, and $Y_i = 0$ otherwise. Let $Y = \sum_{i=1}^{4m} Y_i$. First, we bound the expected value of each Y_i. If $H_i = R$ then clearly $E[Y_i] = 0$. Suppose $H_i \neq R$. By Lemma 2 there is an $r \in S$ such that $H_i r \not\subseteq H_i$. As $H_i r$ is an additive group it follows that $H_i r \cap H_i$ is a proper subgroup of $H_i r$ and hence $|H_i r \cap H_i| \leq |H_i r|/2$. Therefore, for a random $x \in H_i r$ the probability that it lies in H_i is at most $1/2$. Since x_{ir} is ϵ-uniformly distributed we have $\text{Prob}[Y_i = 1] \leq \text{Prob}[x_{ir} \in H_i] \leq 1/2(1 + 1/2^m)$. Putting it together, we get $\mu = E[Y] \leq 2m(1 + 1/2^m) \leq 2.5m$ for $m > 1$. Now, by Markov's inequality $\text{Prob}[Y > 3m] \leq \text{Prob}[Y > 3\mu/2.5] \leq 5/6$.

Combining Theorem 1 with Lemma 1 we immediately obtain the main theorem of this section.

Theorem 2. *There is a polynomial-time almost uniform sampling algorithm from black-box rings that takes as input $R = \langle S \rangle$ and $\epsilon > 0$, runs in time polynomial in input size and $\log(1/\epsilon)$ and outputs an ϵ-uniform random element from the ring R.*

Remark. We note that if the characteristic of the ring R is unary (in input size) then it is possible to modify the above polynomial-time sampling algorithm into an NC sampling algorithm.

Let $R = \langle r_1, r_2, \ldots, r_n \rangle$ be a black-box ring with elements encoded as strings in Σ^m. Examining the proof of Theorem 1 it is easy to see that every element $r \in R$ can be computed by an arithmetic circuit C_r (a straight-line program) that takes as input the generators $r_1, r_2, \ldots, r_n$ and has gates labeled $+$ and $*$ corresponding to the ring operations, such that C_r evaluates to r, and the size of the circuit C_r is $O(m^3 n^3)$. This is analogous to the reachability lemma for finite black-box groups [BS84].

Lemma 3 (ring reachability lemma). *Let $R = \langle r_1, r_2, \ldots, r_n \rangle$ be a black-box ring with elements encoded as strings in Σ^m. For every $r \in R$ there is an arithmetic circuit C_r of size $O(m^3 n^3)$ that has gates labeled by ring operations $+$ and $*$, takes as input $r_1, r_2, \ldots, r_n$ and evaluates to r.*

4 Quantum algorithm for finding a basis representation

In this section we describe a quantum polynomial-time algorithm that takes a black-box ring and computes a basis representation for it. The algorithm is Monte Carlo with small error probability.

Theorem 3. *There is a quantum polynomial-time algorithm that takes a black-box ring as input and computes a basis representation for the ring with small error probability.*

Proof. Let $R = \langle S \rangle$ be the input black-box ring. By the algorithm in Theorem 1 we first compute an additive generating set $\{r_1, r_2, \cdots, r_n\}$ for R. We first claim that there is a quantum polynomial-time algorithm for computing the additive orders d_i for each r_i. I.e. d_i is the least positive integer such that $d_i r_i = 0$, $1 \leq i \leq n$. To see this notice that 2^m is an upper bound on $|R|$, where m is the length of encodings of elements in R. Thus, the problem of computing d_i is precisely the period finding problem that can be solved in quantum polynomial-time by applying Shor's algorithm [Shor97].

The next step is to extract an *additively independent* set T of generators from $\{r_1, r_2, \cdots, r_n\}$ which will serve as the basis for R in its basis representation. Computing such a subset can be easily done using ideas from Cheung and Mosca in [CM01]. The idea is to first decompose $(R, +)$ as the direct sum of it's Sylow subgroups. This decomposition uses Shor's algorithm. Then each of the Sylow subgroups can further be decomposed into direct sum of cyclic groups by solving instances of hidden subgroup problem.

Finally, it remains to express the products rr', for $r, r' \in T$, as integer linear combinations of elements of T. We can again use Shor's period-finding quantum algorithm to compute the additive order d of rr'. Then we define a homomorphism $\varphi : \mathbb{Z}_d \times \mathbb{Z}_{d_{i_1}} \times \cdots \times \mathbb{Z}_{d_{i_\ell}} \to (R, +)$ as $\varphi(a, a_1, a_{i_1}, \cdots, a_{i_\ell}) = -arr' + \sum_{j=1}^{\ell} a_{i_j} r_{i_j}$. By applying [CM01] we can find an additive generating set for $\mathrm{Ker}(\varphi)$. We can express the generating set for $\mathrm{Ker}(\varphi)$ in terms of the basis for the ring R. Let M be the integer matrix whose columns are the generators of $\mathrm{Ker}(\varphi)$. Let M_h be the corresponding *Hermite Normal Form* for M that can be computed in deterministic polynomial time. For expressing rr' as an integer linear combination of the basis elements, we need to seek a vector of the form $(1, x_1, x_2, \cdots, x_\ell)$ in the column space of M_h. Thus, $(1,1)^{\text{th}}$ entry of M_h has to be an invertible element in the ring $\mathbb{Z}_d$. Let its inverse be λ. If C_1 is the first column of M_h, it is easy to see that λC_1 is a solution of the form $(1, x_1, x_2, \cdots, x_m)$ using which we can express rr' as an integer linear combination of the basis.

5 Testing if a black-box ring is a field

In this section we describe a simple quantum polynomial time algorithm that takes a black-box ring as input and tests if it is a field. This result can be seen as a sort of generalization of primality testing: the ring $\mathbb{Z}_n$ is a field if and only if n is a prime. However, the black-box setting for the problem presents obstacles, like finding the additive order of elements, that seem hard for classical (randomized) polynomial time computation.

Theorem 4. *There is a quantum polynomial-time algorithm with small error probability for testing if a given black-box ring is a field.*

Proof. Let R be the input black-box ring. Applying the algorithm in Theorem 3 we obtain with high probability a basis representation for R: $(R, +) = \mathbb{Z}_{m_1} e_1 \oplus \mathbb{Z}_{m_2} e_2 \oplus \cdots \oplus \mathbb{Z}_{m_n} e_n$, and $e_i e_j = \sum_{k=1}^{n} \gamma_{ijk} e_k, \gamma_{ijk} \in \mathbb{Z}_{m_k}$.

Clearly, R is a field only if all m_i's are equal to a prime p. Using the AKS primality testing [AKS04] (or one of the polynomial time randomized tests) we check if p is prime. If not then the input is rejected. Thus, the basis representation can be written as $(R, +) = \mathbb{F}_p e_1 \oplus \mathbb{F}_p e_2 \oplus \cdots \oplus \mathbb{F}_p e_n$.

We next compute the minimal polynomial of e_1 over $\mathbb{F}_p$. This can be easily done in deterministic polynomial time. Suppose the minimal polynomial is $m_1(x)$ with degree d_1. Then in deterministic polynomial time we can test if $m_1(x)$ is irreducible over $\mathbb{F}_p$ [VZG03]. If it is not then the input R is rejected. Otherwise, $\mathbb{F}_p(e_1) = \{a_0 + a_1 e_1 + a_2 e_1^2 + \cdots + a_{d_1-1} e_1^{d_1-1} | \text{ for } 1 \leq i \leq d_1 - 1,\ a_i \in \mathbb{F}_p\}$ is a finite field isomorphic to $\mathbb{F}_{p^{d_1}}$.

With the above step as the base case, inductively we assume that at the i-th step of the algorithm we have computed the finite field $\mathbb{F}_p(e_1, e_2, \ldots, e_i)$ contained in R with a basis $\{v_1, v_2, \cdots, v_k\}$ where each v_i is expressed as an $\mathbb{F}_p$-linear combination of $\{e_1, e_2, \cdots, e_n\}$. Let $d = \prod_{t=1}^{i} d_t$, where d_t is the degree of the minimal polynomial of e_t over $\mathbb{F}(e_1, \ldots, e_{t-1})$ for each t. By induction hypothesis $\mathbb{F}_p(e_1, e_2, \cdots, e_i) \cong \mathbb{F}_{p^d}$. Proceeding inductively, at the $i+1$-th step we again compute the

minimum polynomial $m_{i+1}(x)$ of e_{i+1} over $\mathbb{F}_p(e_1, e_2, \cdots, e_i)$. Using the product relations defining the basis representation for R, it is easy to see that this computation will also boil down to solving a system of linear equations over $\mathbb{F}_p$. Also, we will similarly be able to check in polynomial time whether the obtained minimal polynomial is irreducible over $\mathbb{F}_{p^d}$ [VZG03].

We continue this procedure for n steps and if in none of the steps the above algorithm rejects the input, we conclude that R is a field. Clearly, if the basis representation for R is correct (which it is with high probability), the algorithm will correctly decide.

In the above theorem the power of quantum computation is used only to recover a basis representation for R. If R is already in basis representation then field testing is in P. We now give a classical complexity upper bound for the field testing.

Theorem 5. *Testing if a black-box ring is a field is in* AM $\cap$ coNP.

Proof. A finite ring $R = \langle r_1, \ldots, r_n \rangle$ is not a field if and only if it has zero divisors: nonzero elements $a, b \in R$ whose product $ab = 0$. An NP test for this would be to guess small circuits C_a and C_b (using Lemma 3) for zero divisors a and b verifying their product $ab = 0$ using the black-box oracle. Thus the problem is in coNP. We now show that the problem is in AM. Merlin will send the basis representation for R to Arthur as follows: Merlin sends a basis $\{u_1, u_2, \cdots, u_l\}$ for $(R, +)$ along with their pairwise products in terms of generators. Also Merlin sends each generator r_i as a linear combination of the basis elements u_j. Arthur can now easily verify that $\{u_1, u_2, \cdots, u_l\}$ is a generating set for $(R, +)$ and that the product relations are correct. It remains to verify that $\{u_1, u_2, \cdots, u_l\}$ is *additively independent*. Merlin sends the additive orders d_i of u_i for each i, with the prime factorizations of d_i using which Arthur can verify that d_i are the additive orders. Now, to verify that $\{u_1, u_2, \cdots, u_\ell\}$ is additively independent it suffices to check that the $|R| = \prod_{i=1}^{\ell} d_i$. By a result of Babai [Ba92], order verification of black-box groups is in AM. This protocol can clearly be applied to $(R, +)$.

5.1 An application of the Chebotarëv density theorem

We now briefly explore a somewhat different problem related to testing if a given ring is a field: suppose we are given a basis $e_1, e_2, \ldots, e_n$ along with product relations $e_i e_j = \sum_{k=1}^{n} \gamma_{ijk} e_k$ for integers γ_{ijk}. The question we ask here is whether there is *some* prime p such that *modulo* p the above is a basis representation for the finite field $\mathbb{F}_{p^n}$. We need some algebraic number theory to develop a polynomial-time randomized algorithm to test if there is such a prime.

First, notice that by using the product relations, we can as before compute the minimal polynomials m_i of the e_i *over the rationals* $\mathbb{Q}$. We can check in polynomial time that the m_i are all indeed irreducible over $\mathbb{Q}$ (using the LLL algorithm). Because if m_i are not irreducible over $\mathbb{Q}$ then they are not irreducible modulo any prime and we can reject the input in that case. Now, we use the product relations to compute the tower of fields $\mathbb{Q} \subseteq \mathbb{Q}(e_1) \subseteq \cdots \subseteq \cdots \subseteq \mathbb{Q}(e_1, \ldots, e_n)$. In fact, by the *primitive element theorem*, starting with $f_1 = e_1$ we can compute in polynomial time a primitive element f_i for the field $\mathbb{Q}(e_1, \ldots, e_i)$ as an integer linear combination of $e_1, \ldots, e_i$. Finally, we will obtain f_n such that $\mathbb{Q}(f_n) = \mathbb{Q}(e_1, e_2, \cdots, e_n)$. Using the product relation the problem of finding the minimal polynomial of f_n over $\mathbb{Q}$ reduces to solving a system of linear equations over $\mathbb{Q}$ which can be done in polynomial time. Let $f(x)$ be this minimal polynomial, which has to be irreducible of degree n. Thus, we have $\mathbb{Q}(f_n) \cong \mathbb{Q}[x]/(f(x))$. Let ℓ denote the lcm of the denominators of coefficients of $f(x)$. Then taking $e = lf_n$, we observe that e has a monic minimal polynomial $g(y)$ with integer coefficients and $\mathbb{Q}(f_n) = \mathbb{Q}(e) = \mathbb{Q}[y]/(g(y))$. Now our goal is to test if there is a prime p such that $g(y)$ is irreducible modulo p, so that $\mathbb{Z}_p[y]/g(y)$ is $\mathbb{F}_{p^n}$ and hence the given basis representation modulo p is $\mathbb{F}_{p^n}$.

Let L be the splitting field of $g(y)$. Consider its Galois group $\mathrm{Gal}(L/\mathbb{Q}) = G$ which is fully described by its action on the roots of $g(y)$. Thus G can be seen as a subgroup of S_n. Clearly, any $\sigma \in G$ is a product of disjoint cycles. If the length of these cycles is $n_1, n_2 \ldots, n_k$ such that $n_1 \leq n_2 \leq \ldots \leq n_k$ we say that σ has *cycle pattern* $(n_1, \ldots, n_k)$. Let p be a prime which does not divide the discriminant of g. Then, modulo p, the polynomial $g(y)$ has no multiple roots. If

we factorize g modulo p (using Berlekamp's algorithm) we get $g(y) = g_1(y)g_2(y)\dots g_k(y)$, where g_i are distinct, irreducible and degrees of the g_i is a partition of n. Writing the degrees d_i of g_i in increasing order we obtain the *decomposition pattern* $(d_1, d_2, \dots, d_k)$ of g modulo p.

Now, the Frobenius density theorem (which is a weaker form of the Chebotarëv density theorem) tells us that the number of primes with a given decomposition is close to number of permutations in G having that particular cycle structure. Assuming GRH, the bounds are tight enough to be algorithmically applied. We describe this theorem in a form tailored to our question.

Let $C = (n_1, n_2, \dots, n_k)$ be any cycle pattern and let $C(G)$ be the subset of G consisting of permutations with cycle pattern C. Notice that $C(G)$ is closed under conjugation. Let $\pi_C(x)$ be the number of primes $p \leq x$ such that p is unramified in L and the decomposition pattern of g modulo p is C. Then we have the following theorem which is a restatement of the Frobenius density theorem using [SS97, Lemma 3].

Theorem 6. *Let L be the splitting field of an irreducible polynomial $g(y) \in \mathbb{Z}[y]$. Let d be the discriminant of L. Assuming GRH, there are absolute constants α, β such that if $x \geq \alpha(\log|d|)^\alpha$, then $\pi_C(x) \geq \beta \frac{\#C(G)}{\#G} \frac{x}{\ln x}$.*

Since we are seeking a prime p such that $g(y)$ is irreducible modulo p, let C_0 be the cycle pattern (n). First we can see by an easy counting argument that if $C(G)$ is nonempty then $\#C(G) \geq \frac{\#G}{n}$. Let size$(g)$ denote the size of the polynomial g. Then it follows that $\log|d| \leq (n+1)!^2 \cdot \text{size}(g)$. We choose $x = \lceil \alpha(\log|d|)^\alpha \rceil$ which is a polynomial-sized integer. By the above theorem it follows that $g(y)$ is irreducible modulo p for a random prime $p \leq x$ with probability at least β/n if there is such a prime at all. This gives us a simple randomized polynomial time procedure that finds such a prime p if it exists with nonnegligible success probability (assuming GRH).

6 Complexity of Nilradical

Let R be a commutative ring. An element $x \in R$ is *nilpotent* if $x^n = 0$ for some $n > 0$. In a commutative ring R, the set of all nilpotent elements of R form an ideal $N(R)$ called the *nilradical* of R. The nilradical is crucial to the structure of rings and it plays an important role in decomposing finite rings. In this section, we show that the nilradical of *commutative* black-box rings can be computed in quantum polynomial time.

Let $R = \langle S \rangle$ be a commutative black-box ring. By Lemma 1 we first compute an additive generating set for R with high probability. Let T denote the computed additive generating set.

Applying Cheung & Mosca's ideas, as explained in Theorem 3, in quantum polynomial time we can compute from T an additive *independent* generating set for $(R, +)$. Call this generating set $T' = \{r_1, r_2, \dots, r_\ell\}$. Now, using Shor's algorithm we can find their additive orders d_i with high probability, implying that $|R| = \prod_{i=1}^{n} d_i$. Again by Shor's integer factoring algorithm we compute the prime factorization $|R| = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$. Let $n_i = \frac{|R|}{p_i^{\alpha_i}}$ for $1 \leq i \leq k$.

By elementary group theory, we know that the additive p_i-Sylow subgroups $(R_i, +)$ of $(R, +)$ is additively generated by $T_i = \{n_i r | r \in T'\}$. It is easy to see that R_i's are actually subrings of R (the p_i-Sylow subrings), in fact even ideals and furthermore $R = R_1 \oplus \cdots \oplus R_k$ is a direct sum ring decomposition with $R_i R_j = 0$, $\forall i \neq j$. Thus, the nilradical N of R is given by $N = N_1 \oplus N_2 \oplus \cdots \oplus N_k$, where each N_i is the nilradical of R_i. Thus, it suffice to explain how to compute an additive generating set for the nilradical of a ring R s.t $|R| = p^\alpha$ for some prime p.

As explained in Theorem 3 we compute a basis representation for R in quantum polynomial time. Let $e_1, e_2, \dots, e_t$ be the basis elements with additive orders p^{α_i} $(1 \leq i \leq t)$ respectively. Then $R \cong \mathbb{Z}_{p_1^{\alpha_1}} e_1 \oplus \cdots \oplus \mathbb{Z}_{p_t^{\alpha_t}} e_t$ with $e_i e_j = \sum_{k=1}^{t} \gamma_{ijk} e_k$ is the basis representation.

Since $p^\alpha r = 0$ $\forall r \in R$, it easily follows that $pR = \{pr | r \in R\}$ is a subring of R contained in the nilradical. Indeed pR is an ideal of R. Thus R/pR is also a finite ring. Moreover, we can easily write down its basis representation as follows: Let $f_i = e_i + pR$, $1 \leq i \leq t$. Then $R/pR = \mathbb{F}_p f_1 \oplus \cdots \oplus \mathbb{F}_p f_t$ where the products $f_i f_j$ are $e_i e_j + pR$ and can be expressed as an $\mathbb{F}_p$-linear combination of the f_i's. The following lemma is easy to see.

Lemma 4. *N is the nilradical of R if and only if N/pR is the nilradical of R/pR.*

Thus, if we can find a basis for the nilradical N/pR of R/pR as linear combinations of the f_i's we can easily pull back into N by replacing the basis elements f_i's by e_i's. Therefore, we have reduced the problem to finding the nilradical of an $\mathbb{F}_p$-algebra R given in basis representation. The proof of the next lemma will be given in the full version.

Lemma 5. *Given an $\mathbb{F}_p$-algebra R in basis representation, its nilradical can be computed in deterministic polynomial time.*

Continuing with the original problem, let S'' be the pullback of S' w.r.t the homomorphism $\phi : R \to R/pR$ (namely replace f_i by e_i). Then $S'' \cup \{pe_i | 1 \leq i \leq t\}$ is an additive generating set for the nilradical of R. Putting it together, we have proved the following theorem.

Theorem 7. *The nilradical of a black-box ring can be computed in quantum polynomial time.*

Also similar kind of techniques as of Theorem 5 easily suggests the following result about the classical complexity of nilradical testing.

Theorem 8. *Let R be a black-box ring and I be an ideal of R given by a generator set.Testing if I is the nilradical of R is in* $\mathrm{AM} \cap \mathrm{coAM}$.

The *square-free part* of a positive integer n is the product of all distinct prime factors of n. We now observe that computing the nilradical of a black-box ring is harder than computing the square-free part of an integer.

Lemma 6. *Computing the square free part of an integer n is polynomial time Turing reducible to computing the nilradical of $\mathbb{Z}_n$.*

Proof. Let $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$. Then the square-free part of n is $s = p_1 p_2 \ldots p_k$. An x in $\mathbb{Z}_n$ is in the nilradical N if and only if $x^m = 0 (\mathrm{mod} n)$, which is possible if and only if s divides x. Now, suppose we have an algorithm that computes a generator set T for N, where T generates N as a ring. Let $n_1 \in T$ be any element. Then $n_1 < n$ and s divides n_1. We again apply the algorithm to find a generator set for the nilradical N_1 of $\mathbb{Z}_{n_1}$. Continuing thus, we obtain a sequence of integers $n > n_1 > n_2 ... > n_t$ where n_i divides n_{i-1} for each i and each n_i is a multiple of s. Thus, this sequence is of length at most $\log n$ and must terminate at some $n_t = s$, which we can detect since the nilradical of $\mathbb{Z}_s$ is $\{0\}$.

References

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics,* 160(2):781-793, 2004.

[AS05] Manindra Agrawal and Nitin Saxena. Automorphisms of Finite Rings and Applications to Complexity of Problems.*STACS'05, Springer LNCS 3404.* 1-17, 2005.

[Ba91] L. Babai.Local Expansion of Vertex-Transitive Graphs and Random Generation in Finite Groups.*STOC* 1991: 164-174.

[Ba92] L. Babai. Bounded Round Interactive Proofs in Finite Groups. *SIAM J. Discrete Math.,* 5(1): 88-111 1992.

[BDG88a] J. L. Balcázar, J. Díaz, and D. Gabarró. Structural Complexity I. *ETACS Monographs on Theoretical Computer Science, Springer Verlag.* 1988 (I).

[BDG88b] J. L. Balcázar, J. Díaz, and D. Gabarró. Structural Complexity II. *ETACS Monographs on Theoretical Computer Science, Springer Verlag.* 1990 (II).

[BM88] L. Babai, S. Moran, Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes. *Journal Comput. Syst. Sciences,* 36(2): 254-276 (1988).

[BMc74] Bernard R. McDonald, Finite Rings with Identity. *Marcel Dekker, Inc.*, 1974.

[BS84] L. Babai and E. Szemerédi, On the complexity of matrix group problems I, *In Proc. 25th IEEE Sympos. on the Foundation of Computer Science,* pp.229-240 (1984).

[BV97] U. Vazirani and E. Bernstein, Quantum Complexity Theory. *Special issue on Quantum Computation of the Siam Journal of Computing*, Oct.(1997).

[CM01] Kevin K.H. Cheung, Michele Mosca Decomposing Finite Abelian Groups. *Los Alamos Preprint Archive,* quant-ph/0101004, 2001.

[Ebr89] Wayne Eberly, Computations for algebras and group representations, PhD thesis. *University of Toronto,* (1989).

[FR85] K. Friedl and L. Rónyai, Polynomial time solutions for some problems in computational algebra. *in Proc. 17th Ann. Symp. Theory of Computing,* 153-162,(1985).

[KS05] Neeraj Kayal, Nitin Saxena, On the Ring Isomorphism and Automorphism Problems. *IEEE Conference on Computational Complexity*, 2-12, 2005.

[Le92] H. W. Lenstra Jr., Algorithms in algebraic number theory. *Bulletin of the AMS,* 26(2): 211-244, 1992.

[Shor97] Peter Shor, Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing,* 26(5):1484-1509, 1997.

[SS97] T. Sanders and M. A. Shokrollahi, Deciding properties of polynomials without factoring. *Proc. 38th IEEE Foundations of Computer Science,* 46-55, 1997.

[VZG03] Joachim v.z Gathen and Jürgen Gerhard, Modern Computer Algebra. *Cambridge University Press, 2nd Ed.*, 2003 .

View publication stats