Automorphisms of Finite Rings and Applications to Complexity of Problems

Manindra Agrawal
NUS / IITK

Motivation

- Automorphisms of an algebraic structure capture its symmetries.

- Many properties can be proved by analyzing the automorphism group of the structure.

Examples in Mathematics

- [Galois,1830] Structure of automorphism group of the splitting field of a polynomial $f(x)$ characterizes the solvability of f using radicals.

- [Hasse,1932] The number of rational points on elliptic curve E_p is between $p+1-2\sqrt{p}$ and $p+1+2\sqrt{p}$.

What About Algorithms & Complexity?

- Not received much attention.
- Used only for few problems like polynomial factorization.
- So are they not of much use?

Automorphisms of finite rings are intimately related to the complexity of many important algebraic problems.

Examples Discussed

1. Primality Testing
2. Integer Factoring
3. Polynomial Factoring
4. Graph Isomorphism
5. Polynomial Equivalence

Problems related to Automorphisms / Isomorphisms

- Ring Automorphism: Given a ring R, does it have a non-trivial automorphism?
- Ring Isomorphism: Given two rings R, S, are they isomorphic?
  - The functional versions of above two require one to find a morphism.
- Automorphism Testing: Given a ring R and a function $\phi: R \mapsto R$, is ϕ an automorphism?

Representations of Finite Rings

- We consider finite commutative rings with identity.

- These rings have three main representations:
  - Table representation
  - Basis representation
  - Polynomial representation

Table Representation

- The ring R is given as
  - $(e_1, e_2, \ldots, e_n)$ – the set of elements in R
  - The table of addition operation
  - The table of multiplication operation

- The size of representation is $\Theta(|R|^2)$.

Table Representation: Complexity

- Problems related to automorphisms can be computed in time $O(n^{\log n})$:
  - The ring has $O(\log n)$-sized generator set under addition.
  - An automorphism maps a generator set to another.

- Too verbose!

Basis Representation

- The ring R is given as
  - $(b_1, m_1, b_2, m_2, \ldots, b_n, m_n)$ where $b_1, \ldots, b_n$ is a generator set for R under addition and m_i is the order of b_i.
  - The table of multiplication operation for generators: $b_i * b_j = \sum_{1 \le k \le n} \alpha_{ijk} b_k$.

- The size of representation is $\Theta(n^3) = O(\log |R|)^3$ – exponentially smaller than table representation.

Basis Representation: Complexity

- Problems related to automorphisms are in the class $FP^{AM \cap coAM}$ [Kayal-Saxena,2004]:
  - An automorphism/isomorphism is a linear map on additive generator set.
  - So guess-and-verify technique works.
  - A variant of Graph Isomorphism in coAM proof works.

Polynomial Representation

- The ring R is given as
  - $Z_m[X_1, \ldots, X_n] / (f_1, \ldots, f_k)$ where $X_1, \ldots, X_n$ is a generator set for R under addition and multiplication and $(f_1, \ldots, f_k)$ is the ideal of polynomials satisfied by $X_1, \ldots, X_n$.
  - Each f_i is given as an arithmetic circuit.

- The size of representation can be exponentially smaller than basis representation:
  - Example: $F_2[X_1, \ldots, X_n] / (X_1^2, \ldots, X_n^2)$

Polynomial Representation: Complexity

- Problems related to automorphisms are NP-hard:
  - An automorphism is completely specified by its action on $X_1, \ldots, X_n$.
  - Verifying membership in the ideal $(f_1, \ldots, f_k)$ can be hard (EXPSPACE-complete in general).
  - Ring Automorphism problem is NP-hard.
  - Ring Isomorphism problem is coNP-hard.
- Too compact!

- So the best representation, from the complexity perspective, is basis representation.

- Often, basis and polynomial representations have similar sizes.
  - In such cases, we use polynomial representation as it is most natural one.

Application to Primality Testing

Automorphism Testing ⇨ Primality Testing

Fermat's Little Theorem: If n is prime then the map $\phi(x) = x^n \pmod n$ is the trivial automorphism of ring Z_n.

- Converse is not true.
- Even if it were, it is expensive to test that the map is indeed an automorphism.

- These problems can be eliminated!

Automorphism Testing ⇨ Primality Testing

- Let $R = Z_n[Y] / (Y^r - 1)$ for some $r > 0$ and define $\phi: R \mapsto R$ as $\phi(x) = x^n$.

Observation: ϕ is an automorphism of R iff for every $g(Y) \in R$,

$$g^n(Y) = \phi(g(Y)) = g(\phi(Y)) = g(Y^n).$$

Automorphism Testing ⇨ Primality Testing

[A-Kayal-Saxena,2002]: For suitably chosen "small" r, if $(Y + a)^n = Y^n + a$ in R for $1 \leq a \leq \sqrt{r} \log n$, then either n is prime or has a divisor $< r$.

- Above is a slight generalization of the original statement.

Automorphism Testing ⇨ Primality Testing

- Let ring $S = Z_n[Y] / (Y^{2r} - Y^r)$.

- The AKS theorem translates to:

Theorem: (1) n is prime iff ϕ is an automorphism in S.

(2) ϕ is an automorphism in S iff $\phi(Y + a) = \phi(Y) + a$ for $1 \leq a \leq \sqrt{r} \log n$.

Application to Polynomial Factoring

Automorphism Testing ⇨ Polynomial Factoring

- Let f be a polynomial of degree d in $F_q[Y]$.
- Let $R = F_q[Y] / (f)$ and $\phi(x) = x^q$.

Observation: (1) ϕ is an automorphism in R and ϕ^d is the trivial automorphism.

(2) ϕ^k is trivial iff degrees of all irreducible factors of f divide k.

(3) ϕ^k is trivial iff $Y^{q^k} = \phi^k(Y) = Y$.

Automorphism Testing ⇨ Polynomial Factoring

- This allows to test for irreducibility of f as well as separate distinct degree factors of f:

  - For $k = 1$ to d do: compute $\gcd(f, Y^{q^k} - Y)$.

Automorphism Testing ⇨ Polynomial Factoring

- Finding equal degree factors of f can be reduced to finding roots of a related polynomial in F_q:

  - Find a $t(Y) \in R \setminus F_q$, with $\phi(t(Y)) = t(Y)$. [use linear algebra]
  - Let $g(x) = \mathrm{Res}(\, t(Y) - x, f(Y)\,)$.
  - For a root α of g, $\gcd(\, t(Y) - \alpha, f(Y)\,)$ is non-trivial.

Automorphism Testing ⇨ Polynomial Factoring

- Roots of g can be computed using distinct degree factorization method.

- Works in randomized polynomial time.

Application to Integer Factoring

Finding Ring Automorphism ⇔ Integer Factoring

- Quadratic Sieve, Number Field Sieve: the fastest two known method for factoring integers.
- Both aim to find a and b in Z_n, $a \neq \pm b$, $a^2 = b^2 \pmod n$.
- Given such a and b, gcd(a+b, n) is non-trivial.

These methods are equivalent to finding an automorphism in a special ring.

Finding Ring Automorphism ⇔ Integer Factoring

- Let $R = Z_n[Y] / (Y^2 - 1)$ for odd n.

Observation: $x \mapsto x$ and $x \mapsto -x$ are two straightforward automorphisms in R.

Lemma: Let ϕ be any automorphism of R. Then, $\phi(Y) = cY$ with $c^2 = 1 \pmod{n}$.

Finding Ring Automorphism ⇔ Integer Factoring

Proof: Let $\phi(Y) = cY + d$. Then,

$$0 = \phi(Y^2 - 1) = (cY+d)^2 - 1$$
$$= 2cdY + c^2 + d^2 - 1.$$

Since ϕ is an automorphism, $(c, n) = 1$. Thus, $d = 0$ and $c^2 = 1$ in Z_n. □

So for any third automorphism, $c \neq \pm 1$.

Therefore, finding a third automorphism is equivalent to factoring n.

Finding Ring Automorphism ⇔ Integer Factoring

- Conversely, finding ring automorphism can be reduced to integer factoring.
- [Kayal-Saxena,2004] showed how:
  - Given ring R, split it as a sum of local rings using integer and polynomial factoring oracles.
  - For each local ring, it is easy to find a non-trivial automorphism if it exists.

Finding Ring Automorphism ⇔ Integer Factoring

- There are many other connections too.
- [Kayal-Saxena,2004] showed that integer factoring reduces to:
  - Counting number of automorphisms of $Z_n[Y] / (Y^2)$.
  - Finding any non-trivial automorphism of $Z_n[Y] / (f)$, f a random degree 3 poly.
  - Finding any isomorphism between $Z_n[Y] / (Y^2-1)$ and $Z_n[Y] / (Y^2-a^2)$, a randomly chosen from Z_n.

Application to Graph Isomorphism

Ring Isomorphism ⇨ Graph Isomorphism

- Shown in [Kayal-Saxena,2004].
- Here, we give a different, more general proof.
- Let $G = (V, E)$ be a graph on n vertices.
- Define polynomial p_G as:

$$p_G(x_1,\ldots,x_n) = \sum_{(i,j)\in E} x_i * x_j.$$

- Define polynomial ideal I_G as:

$$I_G(x_1,\ldots,x_n) = (p_G(x_1,\ldots,x_n), \{x_i^2\}_{1 \le i \le n}, \{x_i x_j x_k\}_{1 \le i < j < k \le n}).$$

Ring Isomorphism ⇨ Graph Isomorphism

- Let $R_{q,G} = F_q[Y_1,\ldots,Y_n] / I_G(Y_1,\ldots,Y_n)$.

Theorem: Graphs G_1 and G_2 are isomorphic iff either $G_1 = G_2 = K_m \cup D_{n-m}$ or rings R_{q,G_1} and R_{q,G_2} are isomorphic.

Here, D_{n-m} is a collection of $n-m$ isolated vertices and q any odd prime power.

Ring Isomorphism ⇨ Graph Isomorphism

Proof: If the graphs are isomorphic via π, the rings are isomorphic via $\phi(Y_i) = Y_{\pi(i)}$.

Suppose the rings are isomorphic and $G_2 \neq K_m \cup D_{n-m}$ for any m.

Let ϕ be an isomorphism,

$$\phi(Y_i) = a_i + \sum_{1 \le j \le n} b_{ij} Y_j + \sum_{1 \le j < k \le n} c_{ijk} Y_j Y_k$$

Ring Isomorphism ⇨ Graph Isomorphism

Since $\phi(Y_i)^2 = \phi(Y_i^2) = 0$:

$0 = \phi(Y_i)^2 = a_i^2 +$ higher degree terms,

implying that $a_i = 0$.

So:

$$0 = \phi(Y_i)^2 = 2 \sum_{1 \le j < k \le n} b_{ij} b_{ik} Y_j Y_k.$$

Ring Isomorphism ⇨ Graph Isomorphism

If two or more b_i's are non-zero, p_{G_2} must divide $\phi(Y)^2$.

This implies $G_2 = K_m \cup D_{n-m}$. Not possible.

If all b_i's are zero then $\phi(Y_i Y_t) = 0$. Not possible.

So, exactly one of b_i's is non-zero.

Ring Isomorphism ⇨ Graph Isomorphism

Let $\pi(i) = j$ where b_{ij} is non-zero.

If $\pi(i) = \pi(t)$, then $\phi(Y_i Y_t) = 0$. Not possible.

So π is a permutation on $[1,n]$.

Ring Isomorphism ⇨ Graph Isomorphism

Also:

$$0 = \pi(p_{G_1}) = \sum_{(i,j)\in E_1} \phi(Y_i)\phi(Y_j)$$
$$= \sum_{(i,j)\in E_1} b_{i,\pi(i)} b_{j,\pi(j)}\, Y_i Y_j.$$

So p_{G_2} must divide above.

This means $\pi(p_{G_1})$ is a constant multiple of p_{G_2} implying that π is an isomorphism.

Application to Polynomial Equivalence

Polynomial Equivalence

The Problem: Given two polynomials f and g in $F[x_1,\ldots,x_n]$, test if there exists an invertible linear transformation T such that

$$g(x_1,\ldots,x_n) = f(Tx_1,\ldots,Tx_n).$$

- [Thierauf,1998] proved it is in $NP \cap coAM$ when T is required to be a permutation.
- His proof works for arbitrary linear transformations too.

Polynomial Equivalence

- Polynomial equivalence for d-forms (homogeneous polynomials of degree d) is well-studied.
- Witt's theorem [1936] implies a polynomial time algorithm for quadratic forms.
- No such algorithm is known for cubic forms.
- There is even a cryptosystem based on (presumed) difficulty of deciding equivalence between collections of cubic forms.

Polynomial Equivalence ⇨ Ring Isomorphism

Theorem: Ring Isomorphism for rings of prime characteristic reduces to Polynomial Equivalence.

Proof: Let R and S be two rings given in basis representation:

$R = (b_1, p, \ldots, b_n, p)$, $b_i b_j = \sum_{1 \le k \le n} \alpha_{ijk} b_k$

$S = (d_1, p, \ldots, d_n, p)$, $d_i d_j = \sum_{1 \le k \le n} \beta_{ijk} d_k$

Polynomial Equivalence ⇨ Ring Isomorphism

Define polynomial $p_R(\mathbf{y},\mathbf{b})$ as:

$$p_R(\mathbf{y},\mathbf{b}) = \sum_{1\le i\le j\le n} y_{ij} \left(b_i b_j - \sum_{1\le k\le n} \alpha_{ijk}\, b_k\right).$$

Similarly define polynomial $p_S(\mathbf{z},\mathbf{d})$.

Claim: If R and S are isomorphic, then p_R and p_S are equivalent.

Proof: Let ϕ be an isomorphism between R and S.

Polynomial Equivalence ⇨ Ring Isomorphism

Then $\phi(b_i b_j - \sum_{1\le k\le n} \alpha_{ijk}\, b_k) = 0$ in S.

This implies that

$$\phi(b_i b_j - \sum_{1\le k\le n}\alpha_{ijk} b_k) = \sum_{l,m} \gamma_{ijlm}(d_l d_m - \sum_{1\le k\le n}\beta_{lmk} d_k).$$

Therefore, the T that extends ϕ to y_{ij}'s as:

$$T(\sum_{ij} \gamma_{ijlm}\, y_{ij}) = z_{lm}$$

is an equivalence between the polynomials.

Polynomial Equivalence ⇨ Ring Isomorphism

Claim: If p_R and p_S are equivalent then R and S are isomorphic.

Proof: Let T be an equivalence. Then:

$$\sum_{1\le i\le j\le n} T(y_{ij})\, T(b_i b_j - \sum_{1\le k\le n} \alpha_{ijk}\, b_k) = \sum_{1\le i\le j\le n} z_{ij}\,(d_i d_j - \sum_{1\le k\le n} \beta_{ijk}\, d_k).$$

By comparing degrees, we get:

$$\sum_{1\le i\le j\le n} T(y_{ij})\, T(b_i b_j) = \sum_{1\le i\le j\le n} z_{ij} d_i d_j.$$

Polynomial Equivalence ⇨ Ring Isomorphism

We first show that $T(b_i)$ is a linear combination of only d's.

Suppose not. Let $T(b_1)$ include z_{11}.

Set z_{11} to make $T(b_1)$ zero. This gives:

$\sum_{1<i\le j\le n} T(y_{ij})\, T(b_i b_j) = \sum_{1\le i\le j\le n,\, j>1} z_{ij}$ (quad d's) + (cubic d's).

Polynomial Equivalence ⇨ Ring Isomorphism

Notice that LHS has only $n(n-1)/2$ terms left while RHS has $n(n+1)/2 - 1$ z's.

For each term on LHS, if any of its component has a z-variable in it, set that variable to make the component zero.

Continuing this way, by setting at most $1+n(n-1)/2$ z-variables, LHS is independent of z's. But RHS still has $n-1$ unset z-variables. Contradiction.

Polynomial Equivalence ⇨ Ring Isomorphism

So each $T(b_i)$ has only d's. The equation is:

$$\sum_{1\le i\le j\le n} T(y_{ij})\, T(b_i b_j - \sum_{1\le k\le n} \alpha_{ijk}\, b_k) = \sum_{1\le i\le j\le n} z_{ij}\, (d_i d_j - \sum_{1\le k\le n} \beta_{ijk}\, d_k).$$

Since there are no cubic d's in RHS, we can ignore d's in $T(y_{ij})$.

Suppose that $T(b_i b_j - \sum_{1\le k\le n} \alpha_{ijk}\, b_k)$ is not in S.

Polynomial Equivalence ⇨ Ring Isomorphism

Then, in S:

$$T(b_i b_j - \sum_{1 \le k \le n} \alpha_{ijk}\, b_k) = \sum_k \gamma_{ijk}\, d_k.$$

Therefore, $\sum_{1 \le i \le j \le n} \gamma_{ijk}\, T(y_{ij}) = 0$ in S. This is not possible since T is invertible on y's.

Therefore, T restricted to b's is an isomorphism from R to S.

Other Connections

- Similar, more involved, proof shows that Graph Isomorphism reduces to cubic form equivalence.

- d-form equivalence over F_q with $(d, q-1) = 1$, reduces to Ring Isomorphism for constant d.

Open Questions

- Can one find connections with problems like discrete-log?

- Can one show that Ring Isomorphism reduces to cubic form equivalence?
  - Our proof only reduces to degree 3 polynomials.

- Most of the effort in Integer Factoring has been concentrated on the ring $Z_n[Y] / (Y^2 - 1)$.
  - Can taking the problem to other rings help?
  - [Kayal-Saxena,2004] provide some alternative rings.

- We reduce Graph Isomorphism to cubic form equivalence (over any field).

- Is the theory of cubic forms of any help in solving Graph Isomorphism?

Thank you!