Automorphisms of Finite Rings and Applications to Complexity of Problems

Manindra Agarwal   Nitin Saxena

National University of Singapore
and
IIT Kanpur

IOS, May 2005

Outline

Part I: Motivation and Definitions
Part II: Applications

Outline of Part I

Motivation

Mathematics

Computer Science

Definitions

Finite Rings

Automorphisms and Isomorphisms

Problems Related to Automorphisms

Complexity of Problems on Different Representations

Ring Automorphism Problem

Complexity of Other Problems

Outline

Part I: Motivation and Definitions
Part II: Applications

Outline of Part II

Primality Testing

Polynomial Factoring

- Over Finite Fields
- Other Variations

Integer Factoring

- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence

- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Part I

Automorphisms: Motivation and Definitions

Outline

Motivation
- Mathematics
- Computer Science

Definitions
- Finite Rings
- Automorphisms and Isomorphisms
- Problems Related to Automorphisms

Complexity of Problems on Different Representations
- Ring Automorphism Problem
- Complexity of Other Problems

Motivation: Mathematics

- Automorphisms of algebraic structures capture its symmetries.
- Many properties of the structure can be proved by analyzing the automorphism group of the structure.

Examples

- Galois (1830) showed that the structure of automorphism group of the splitting field of polynomial $f(x)$ can be used to characterize solvability of f by radicals.
- Wantzel (1836) showed that not all angles can be trisected using ruler and compass.

Examples

- Galois (1830) showed that the structure of automorphism group of the splitting field of polynomial $f(x)$ can be used to characterize solvability of f by radicals.
- Wantzel (1836) showed that not all angles can be trisected using ruler and compass.

Outline

Motivation
- Mathematics
- Computer Science

Definitions
- Finite Rings
- Automorphisms and Isomorphisms
- Problems Related to Automorphisms

Complexity of Problems on Different Representations
- Ring Automorphism Problem
- Complexity of Other Problems

Motivation: Computer Science

- A useful tool in analyzing computational complexity of problems in algebra and number theory.
- Automorphisms and isomorphisms of finite rings are most useful as we will see.
- There are many applications, but only a few are well-known.
- In this talk, we:
    - identify algorithmic problems related to automorphisms and isomorphisms, and
    - present an overview of several applications of these.

Motivation: Computer Science

- A useful tool in analyzing computational complexity of problems in algebra and number theory.
- Automorphisms and isomorphisms of finite rings are most useful as we will see.
- There are many applications, but only a few are well-known.
- In this talk, we:
  - identify algorithmic problems related to automorphisms and isomorphisms, and
  - present an overview of several applications of these.

Motivation: Computer Science

- A useful tool in analyzing computational complexity of problems in algebra and number theory.
- Automorphisms and isomorphisms of finite rings are most useful as we will see.
- There are many applications, but only a few are well-known.
- In this talk, we:
    - identify algorithmic problems related to automorphisms and isomorphisms, and
    - present an overview of several applications of these.

Outline

Motivation
- Mathematics
- Computer Science

Definitions
- Finite Rings
- Automorphisms and Isomorphisms
- Problems Related to Automorphisms

Complexity of Problems on Different Representations
- Ring Automorphism Problem
- Complexity of Other Problems

Finite Rings and Their Representations

- We define a finite ring to be a finite commutative ring with identity.
- There are three main ways to represent these rings:
  - Table Representation.
  - Basis Representation.
  - Polynomial Representation.
- Each representation has a different complexity.

Finite Rings and Their Representations

- We define a finite ring to be a finite commutative ring with identity.
- There are three main ways to represent these rings:
  - Table Representation.
  - Basis Representation.
  - Polynomial Representation.
- Each representation has a different complexity.

Finite Rings and Their Representations

- We define a finite ring to be a finite commutative ring with identity.
- There are three main ways to represent these rings:
  - Table Representation.
  - Basis Representation.
  - Polynomial Representation.
- Each representation has a different complexity.

Finite Rings and Their Representations

- We define a finite ring to be a finite commutative ring with identity.
- There are three main ways to represent these rings:
  - Table Representation.
  - Basis Representation.
  - Polynomial Representation.
- Each representation has a different complexity.

Finite Rings and Their Representations

- We define a finite ring to be a finite commutative ring with identity.
- There are three main ways to represent these rings:
  - Table Representation.
  - Basis Representation.
  - Polynomial Representation.
- Each representation has a different complexity.

Table Representation

- Let R be a finite ring with n elements $e_1, \ldots, e_n$.
- The Table Representation of R is given by two $n \times n$ tables with entries from the interval $[1, n]$:
  - The first table encodes the addition operation with its (i, j)th entry equal to k when $e_i + e_j = e_k$.
  - The second table encodes the multiplication operation similarly.
- The size of the representation is $O(n^2)$.

Table Representation

- Let R be a finite ring with n elements $e_1, \ldots, e_n$.
- The Table Representation of R is given by two $n \times n$ tables with entries from the interval $[1, n]$:
  - The first table encodes the addition operation with its (i, j)th entry equal to k when $e_i + e_j = e_k$.
  - The second table encodes the multiplication operation similarly.
- The size of the representation is $O(n^2)$.

Table Representation

- Let R be a finite ring with n elements $e_1, \ldots, e_n$.
- The Table Representation of R is given by two $n \times n$ tables with entries from the interval $[1, n]$:
  - The first table encodes the addition operation with its (i, j)th entry equal to k when $e_i + e_j = e_k$.
  - The second table encodes the multiplication operation similarly.
- The size of the representation is $O(n^2)$.

Example

- Let R be the ring of polynomials over field F_2 modulo polynomial $x^4 - 1$.
- The ring has $2^4 = 16$ elements.
- Its Table Representation will provide two 16×16 addition and multiplication tables for all elements of the ring.

Example

- Let R be the ring of polynomials over field F_2 modulo polynomial $x^4 - 1$.
- The ring has $2^4 = 16$ elements.
- Its Table Representation will provide two 16×16 addition and multiplication tables for all elements of the ring.

Basis Representation

- Consider the additive structure on R.
- Since R is finite, $(R, +)$ has a finite set of generators.
- Let b_1, b_2, ..., b_m be a set of generators for $(R, +)$ such that
  - The order of b_i is r_i.
  - $(R, +) = Z_{r_1} b_1 \oplus Z_{r_2} b_2 \oplus \cdots \oplus Z_{r_m} b_m$.
- The Basis Representation of R is given by the m-tuple $(r_1, r_2, \ldots, r_m)$ and matrices M_i for $1 \leq i \leq m$ such that:
  - Each M_i is an $m \times m$ matrix.
  - $b_i \cdot b_j = \sum_{k=1}^{m} \alpha_{ijk} b_k$ with $0 \leq \alpha_{ijk} < r_k$.
- The size of the representation is $O(m^3) = O(\log^3 n)$.
- Therefore, this representation is exponentially more succinct than the Table Representation.

Basis Representation

- Consider the additive structure on R.
- Since R is finite, $(R, +)$ has a finite set of generators.
- Let b_1, b_2, …, b_m be a set of generators for $(R, +)$ such that
  - The order of b_i is r_i.
  - $(R, +) = Z_{r_1} b_1 \oplus Z_{r_2} b_2 \oplus \cdots \oplus Z_{r_m} b_m$.
- The Basis Representation of R is given by the m-tuple $(r_1, r_2, \ldots, r_m)$ and matrices M_i for $1 \le i \le m$ such that:
  - Each M_i is an $m \times m$ matrix.
  - $b_i \cdot b_j = \sum_{k=1}^{m} \alpha_{ijk} b_k$ with $0 \le \alpha_{ijk} < r_k$.
- The size of the representation is $O(m^3) = O(\log^3 n)$.
- Therefore, this representation is exponentially more succinct than the Table Representation.

Basis Representation

- Consider the additive structure on R.
- Since R is finite, $(R, +)$ has a finite set of generators.
- Let b_1, b_2, …, b_m be a set of generators for $(R, +)$ such that
  - The order of b_i is r_i.
  - $(R, +) = Z_{r_1} b_1 \oplus Z_{r_2} b_2 \oplus \cdots \oplus Z_{r_m} b_m$.
- The Basis Representation of R is given by the m-tuple $(r_1, r_2, \ldots, r_m)$ and matrices M_i for $1 \leq i \leq m$ such that:
  - Each M_i is an $m \times m$ matrix.
  - $b_i \cdot b_j = \sum_{k=1}^{m} \alpha_{ijk} b_k$ with $0 \leq \alpha_{ijk} < r_k$.
- The size of the representation is $O(m^3) = O(\log^3 n)$.
- Therefore, this representation is exponentially more succinct than the Table Representation.

Basis Representation

- Consider the additive structure on R.
- Since R is finite, $(R, +)$ has a finite set of generators.
- Let b_1, b_2, ..., b_m be a set of generators for $(R, +)$ such that
  - The order of b_i is r_i.
  - $(R, +) = Z_{r_1} b_1 \oplus Z_{r_2} b_2 \oplus \cdots \oplus Z_{r_m} b_m$.
- The Basis Representation of R is given by the m-tuple $(r_1, r_2, \ldots, r_m)$ and matrices M_i for $1 \leq i \leq m$ such that:
  - Each M_i is an $m \times m$ matrix.
  - $b_i \cdot b_j = \sum_{k=1}^{m} \alpha_{ijk} b_k$ with $0 \leq \alpha_{ijk} < r_k$.
- The size of the representation is $O(m^3) = O(\log^3 n)$.
- Therefore, this representation is exponentially more succinct than the Table Representation.

Example

- The ring R defined earlier has 1, x, x^2, x^3 as a set of generators.
- Each generator has order 2.
- The Basis Representation of the ring is given by the four 4×4 matrices $M_1, \ldots, M_4$.
- Matrix M_1 is identity since it codes multiplication by 1.
- Matrix M_2 codes multiplication by x:

$$M_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

- Similarly for M_3 and M_4.

Example

- The ring R defined earlier has 1, x, x^2, x^3 as a set of generators.
- Each generator has order 2.
- The Basis Representation of the ring is given by the four 4×4 matrices $M_1, \ldots, M_4$.
- Matrix M_1 is identity since it codes multiplication by 1.
- Matrix M_2 codes multiplication by x:

$$M_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

- Similarly for M_3 and M_4.

Example

- The ring R defined earlier has 1, x, x^2, x^3 as a set of generators.
- Each generator has order 2.
- The Basis Representation of the ring is given by the four 4×4 matrices $M_1, \ldots, M_4$.
- Matrix M_1 is identity since it codes multiplication by 1.
- Matrix M_2 codes multiplication by x:

$$M_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

- Similarly for M_3 and M_4.

Polynomial Representation

- Let $r = lcm(r_1, r_2, \ldots, r_m)$.
- Let 1, B_1, B_2, ..., B_t be a minimal subset of generators b_1, ..., b_m such that each b_i can be expressed as a polynomial in 1, B_1, ..., B_t over Z_r.
- Let $\mathcal{I}$ be the set of all polynomials $f(x_1, \ldots, x_t)$ over Z_r in t variables such that $f(B_1, \ldots, B_t) = 0$.
  - Set $\mathcal{I}$ forms an ideal of the polynomial ring $Z_r[y_1, \ldots, y_t]$.

Polynomial Representation

- Let $r = lcm(r_1, r_2, \ldots, r_m)$.
- Let 1, B_1, B_2, ..., B_t be a minimal subset of generators b_1, ..., b_m such that each b_i can be expressed as a polynomial in 1, B_1, ..., B_t over Z_r.
- Let $\mathcal{I}$ be the set of all polynomials $f(x_1, \ldots, x_t)$ over Z_r in t variables such that $f(B_1, \ldots, B_t) = 0$.
    - Set $\mathcal{I}$ forms an ideal of the polynomial ring $Z_r[y_1, \ldots, y_t]$.

Polynomial Representation

- The Polynomial Representation is given by numbers t, r, and a generator set $(f_1, f_2, \ldots, f_k)$ for the ideal $\mathcal{I}$.
- We have $R = Z_r[B_1, \ldots, B_t]/\mathcal{I}$.
- The size of the representation is determined by the number and size of the polynomials f_i.
- It is possible that this representation is exponentially more succinct than the Basis Representation.
- For example, consider the ring $F_2[Y_1, \ldots, Y_t]/(Y_1^2, \ldots, Y_t^2)$.
  - Its Polynomial Representation has size $\Theta(t)$.
  - It has an additive basis of size 2^t and hence its Basis Representation has size $\Theta(2^{3t})$.
  - It has 2^{2^t} elements and so its Table Representation has size $\Omega(2^{2^t})$.

Polynomial Representation

- The Polynomial Representation is given by numbers t, r, and a generator set $(f_1, f_2, \ldots, f_k)$ for the ideal $\mathcal{I}$.
- We have $R = Z_r[B_1, \ldots, B_t]/\mathcal{I}$.
- The size of the representation is determined by the number and size of the polynomials f_i.
- It is possible that this representation is exponentially more succinct than the Basis Representation.
- For example, consider the ring $F_2[Y_1, \ldots, Y_t]/(Y_1^2, \ldots, Y_t^2)$.
  - Its Polynomial Representation has size $\Theta(t)$.
  - It has an additive basis of size 2^t and hence its Basis Representation has size $\Theta(2^{3t})$.
  - It has 2^{2^t} elements and so its Table Representation has size $\Omega(2^{2^t})$.

Polynomial Representation

- The Polynomial Representation is given by numbers t, r, and a generator set $(f_1, f_2, \ldots, f_k)$ for the ideal $\mathcal{I}$.
- We have $R = Z_r[B_1, \ldots, B_t]/\mathcal{I}$.
- The size of the representation is determined by the number and size of the polynomials f_i.
- It is possible that this representation is exponentially more succinct than the Basis Representation.
- For example, consider the ring $F_2[Y_1, \ldots, Y_t]/(Y_1^2, \ldots, Y_t^2)$.
  - Its Polynomial Representation has size $\Theta(t)$.
  - It has an additive basis of size 2^t and hence its Basis Representation has size $\Theta(2^{3t})$.
  - It has 2^{2^t} elements and so its Table Representation has size $\Omega(2^{2^t})$.

Example

- Every element of ring R can be expressed as a polynomial in 1 and x.
- The set of polynomials that are zero in R are all multiples of $x^4 - 1$.
- Therefore, $R = F_2[x]/(x^4 - 1)$.

Example

- Every element of ring R can be expressed as a polynomial in 1 and x.
- The set of polynomials that are zero in R are all multiples of $x^4 - 1$.
- Therefore, $R = F_2[x]/(x^4 - 1)$.

Example

- Every element of ring R can be expressed as a polynomial in 1 and x.
- The set of polynomials that are zero in R are all multiples of $x^4 - 1$.
- Therefore, $R = F_2[x]/(x^4 - 1)$.

Outline

Motivation
Mathematics
Computer Science

Definitions
Finite Rings
Automorphisms and Isomorphisms
Problems Related to Automorphisms

Complexity of Problems on Different Representations
Ring Automorphism Problem
Complexity of Other Problems

Automorphisms and Isomorphisms

- Mapping ϕ, $\phi : R \mapsto R$, is an automorphism of ring R if ϕ is a bijection and for every $a, b \in R$:

  $$\phi(a+b) = \phi(a) + \phi(b)$$

  and

  $$\phi(a * b) = \phi(a) * \phi(b).$$

- Given two rings R and S, mapping ϕ, $\phi : R \mapsto S$, is an isomorphism of R and S if ϕ is a bijection and for every $a, b \in R$:

  $$\phi(a+b) = \phi(a) + \phi(b)$$

  and

  $$\phi(a * b) = \phi(a) * \phi(b).$$

Automorphisms and Isomorphisms

- Mapping ϕ, $\phi : R \mapsto R$, is an automorphism of ring R if ϕ is a bijection and for every $a, b \in R$:

$$\phi(a + b) = \phi(a) + \phi(b)$$

  and

$$\phi(a * b) = \phi(a) * \phi(b).$$

- Given two rings R and S, mapping ϕ, $\phi : R \mapsto S$, is an isomorphism of R and S if ϕ is a bijection and for every $a, b \in R$:

$$\phi(a + b) = \phi(a) + \phi(b)$$

  and

$$\phi(a * b) = \phi(a) * \phi(b).$$

Automorphisms for Basis Representation

- Let b_1, ..., b_m be an additive basis for R.
- Then automorphism ϕ is completely specified by its action on basis elements: Let

$$a = \sum_{i=1}^{m} \alpha_i b_i$$

be any element of R. Then,

$$\phi(a) = \phi(\sum_{i=1}^{m} \alpha_i b_i) = \sum_{i=1}^{m} \alpha_i \phi(b_i).$$

- Same holds for isomorphisms between two rings.

Automorphisms for Polynomial Representation

- Let $R = Z_r[X_1, \ldots, X_t]/\mathcal{I}$.
- An automorphism ϕ of R is completely specified by its action on X_1, ..., X_t: Let

$$a = f(X_1, \ldots, X_t)$$

be any element of R where f is a polynomial. Then,

$$\phi(a) = \phi(f(X_1, \ldots, X_t)) = f(\phi(X_1), \ldots, \phi(X_t)).$$

- Same holds for isomorphisms between two rings.

Outline

Motivation
Mathematics
Computer Science

Definitions
Finite Rings
Automorphisms and Isomorphisms
Problems Related to Automorphisms

Complexity of Problems on Different Representations
Ring Automorphism Problem
Complexity of Other Problems

Problems Related to Automorphisms

- Given a ring R, does it have a non-trivial automorphism?
  - This problem is called Ring Automorphism problem.
  - Its search version requires one to find a non-trivial automorphism.
- Given a ring R and a mapping ϕ, $\phi : R \mapsto R$, is ϕ an automorphism of R?
  - This problem is called Automorphism Testing problem.

Problems Related to Automorphisms

- Given a ring R, does it have a non-trivial automorphism?
  - This problem is called Ring Automorphism problem.
  - Its search version requires one to find a non-trivial automorphism.
- Given a ring R and a mapping ϕ, $\phi : R \mapsto R$, is ϕ an automorphism of R?
  - This problem is called Automorphism Testing problem.

Problems Related to Automorphisms

- Given two rings R and S, are they isomorphic?
  - This problem is called Ring Isomorphism Problem.
  - Its search version requires one to find an isomorphism.

Outline

Motivation
- Mathematics
- Computer Science

Definitions
- Finite Rings
- Automorphisms and Isomorphisms
- Problems Related to Automorphisms

Complexity of Problems on Different Representations
- Ring Automorphism Problem
- Complexity of Other Problems

Complexity of Ring Automorphism Problem: Table Representation

Recall:

- The ring R has m additive generators, $m = O(\log n)$ (n is the size of the ring).
- An automorphism of R is completely specified by its action on a set of additive generators.

Complexity of Ring Automorphism Problem: Table Representation

- Hence to test if R has a non-trivial automorphism, do the following:
  1. Compute an ordered set of m additive generators for R. This can be done in time $O(n^2)$
  2. For every ordered subset of m elements, check if mapping the generators to these elements (in order) defines an automorphism. There are $O(n^m)$ such subsets and for each subset checking if the mapping is an automorphism requires time $O(n^2)$.
- The time complexity of this algorithm is $O(n^m) = O(n^{\log n})$.
- This is quasi-polynomial time since size of input is $\Theta(n^2)$.
- The search version of the problem has the same complexity.

Complexity of Ring Automorphism Problem: Table Representation

- Hence to test if R has a non-trivial automorphism, do the following:
  1. Compute an ordered set of m additive generators for R. This can be done in time $O(n^2)$
  2. For every ordered subset of m elements, check if mapping the generators to these elements (in order) defines an automorphism. There are $O(n^m)$ such subsets and for each subset checking if the mapping is an automorphism requires time $O(n^2)$.
- The time complexity of this algorithm is $O(n^m) = O(n^{\log n})$.
- This is quasi-polynomial time since size of input is $\Theta(n^2)$.
- The search version of the problem has the same complexity.

Complexity of Ring Automorphism Problem: Table Representation

- Hence to test if R has a non-trivial automorphism, do the following:
  1. Compute an ordered set of m additive generators for R. This can be done in time $O(n^2)$
  2. For every ordered subset of m elements, check if mapping the generators to these elements (in order) defines an automorphism. There are $O(n^m)$ such subsets and for each subset checking if the mapping is an automorphism requires time $O(n^2)$.
- The time complexity of this algorithm is $O(n^m) = O(n^{\log n})$.
- This is quasi-polynomial time since size of input is $\Theta(n^2)$.
- The search version of the problem has the same complexity.

Complexity of Ring Automorphism Problem: Table Representation

- Hence to test if R has a non-trivial automorphism, do the following:
  1. Compute an ordered set of m additive generators for R. This can be done in time $O(n^2)$
  2. For every ordered subset of m elements, check if mapping the generators to these elements (in order) defines an automorphism. There are $O(n^m)$ such subsets and for each subset checking if the mapping is an automorphism requires time $O(n^2)$.
- The time complexity of this algorithm is $O(n^m) = O(n^{\log n})$.
- This is quasi-polynomial time since size of input is $\Theta(n^2)$.
- The search version of the problem has the same complexity.

Complexity of Ring Automorphism Problem: Table Representation

- Hence to test if R has a non-trivial automorphism, do the following:
  1. Compute an ordered set of m additive generators for R. This can be done in time $O(n^2)$
  2. For every ordered subset of m elements, check if mapping the generators to these elements (in order) defines an automorphism. There are $O(n^m)$ such subsets and for each subset checking if the mapping is an automorphism requires time $O(n^2)$.
- The time complexity of this algorithm is $O(n^m) = O(n^{\log n})$.
- This is quasi-polynomial time since size of input is $\Theta(n^2)$.
- The search version of the problem has the same complexity.

Complexity of Ring Automorphism Problem: Table Representation

- Hence to test if R has a non-trivial automorphism, do the following:
  1. Compute an ordered set of m additive generators for R. This can be done in time $O(n^2)$
  2. For every ordered subset of m elements, check if mapping the generators to these elements (in order) defines an automorphism. There are $O(n^m)$ such subsets and for each subset checking if the mapping is an automorphism requires time $O(n^2)$.
- The time complexity of this algorithm is $O(n^m) = O(n^{\log n})$.
- This is quasi-polynomial time since size of input is $\Theta(n^2)$.
- The search version of the problem has the same complexity.

Complexity of Ring Automorphism Problem: Table Representation

- Hence to test if R has a non-trivial automorphism, do the following:
  1. Compute an ordered set of m additive generators for R. This can be done in time $O(n^2)$
  2. For every ordered subset of m elements, check if mapping the generators to these elements (in order) defines an automorphism. There are $O(n^m)$ such subsets and for each subset checking if the mapping is an automorphism requires time $O(n^2)$.
- The time complexity of this algorithm is $O(n^m) = O(n^{\log n})$.
- This is quasi-polynomial time since size of input is $\Theta(n^2)$.
- The search version of the problem has the same complexity.

Complexity of Ring Automorphism Problem: Basis Representation

- The size of the input is $O(m^3)$ and so the previous algorithm becomes exponential time.
- The problem now is in NP:
  - Given a set of m additive generators, guess the action of an automorphism on these generators and then verify if this results in a non-trivial automorphism. Verification can be done in time $O(m^3)$ since it just requires verifying multiplication property for all pairs of generators.

Complexity of Ring Automorphism Problem: Basis Representation

- The size of the input is $O(m^3)$ and so the previous algorithm becomes exponential time.
- The problem now is in NP:
  - Given a set of m additive generators, guess the action of an automorphism on these generators and then verify if this results in a non-trivial automorphism. Verification can be done in time $O(m^3)$ since it just requires verifying multiplication property for all pairs of generators.

Complexity of Ring Automorphism Problem: Basis Representation

- Kayal-Saxena (2004) show that the problem is in P!
  - They show that ring R has no non-trivial automorphism iff

$$R = \oplus_j \oplus_i Z_{p_i^{\alpha_{i,j}}},$$

    with $\alpha_{1,j} < \alpha_{2,j} < \alpha_{3,j} < \cdots$ for each j.
  - Then they give an efficient algorithm to detect if R is of this form or not.
- Notice that this implies that the Automorphism Problem for Table Representation is also in P.
- However, the search version of the problem is not known to be in P.
  - Kayal-Saxena (2004) show that the problem is in coAM by adopting the protocol for Graph Isomorphism.

Complexity of Ring Automorphism Problem: Basis Representation

- Kayal-Saxena (2004) show that the problem is in P!
  - They show that ring R has no non-trivial automorphism iff
    $$R = \oplus_j \oplus_i Z_{p_i^{\alpha_{i,j}}},$$
    with $\alpha_{1,j} < \alpha_{2,j} < \alpha_{3,j} < \cdots$ for each j.
  - Then they give an efficient algorithm to detect if R is of this form or not.
- Notice that this implies that the Automorphism Problem for Table Representation is also in P.
- However, the search version of the problem is not known to be in P.
  - Kayal-Saxena (2004) show that the problem is in coAM by adopting the protocol for Graph Isomorphism.

Complexity of Ring Automorphism Problem: Basis Representation

- Kayal-Saxena (2004) show that the problem is in P!
  - They show that ring R has no non-trivial automorphism iff
    $$R = \oplus_j \oplus_i Z_{p_i^{\alpha_{i,j}}},$$
    with $\alpha_{1,j} < \alpha_{2,j} < \alpha_{3,j} < \cdots$ for each j.
  - Then they give an efficient algorithm to detect if R is of this form or not.
- Notice that this implies that the Automorphism Problem for Table Representation is also in P.
- However, the search version of the problem is not known to be in P.
  - Kayal-Saxena (2004) show that the problem is in coAM by adopting the protocol for Graph Isomorphism.

Complexity of Ring Automorphism Problem: Polynomial Representation

Theorem

The Ring Automorphism problem for Polynomial Representation is NP-hard.

Complexity of Ring Automorphism Problem: Polynomial Representation

Proof.

- Let $F(x_1, \ldots, x_n)$ be a 3SAT formula with m clauses and n variables.
- For ith clause $c_i = x_{i_1} \vee \bar{x}_{i_2} \vee x_{i_3}$ of F, define polynomial

$$p_i = 1 - (1 - x_{i_1}) \cdot x_{i_2} \cdot (1 - x_{i_3}).$$

- Polynomial p_i equals 1 on any assignment that satisfies clause c_i, 0 otherwise.

Complexity of Ring Automorphism Problem: Polynomial Representation

Proof.

- Let $F(x_1, \ldots, x_n)$ be a 3SAT formula with m clauses and n variables.
- For ith clause $c_i = x_{i_1} \vee \bar{x}_{i_2} \vee x_{i_3}$ of F, define polynomial

$$p_i = 1 - (1 - x_{i_1}) \cdot x_{i_2} \cdot (1 - x_{i_3}).$$

- Polynomial p_i equals 1 on any assignment that satisfies clause c_i, 0 otherwise.

Complexity of Ring Automorphism Problem: Polynomial Representation

- Let $f(x_1, \ldots, x_n) = \prod_{i=1}^{m} p_i$.
- Polynomial f equals 1 on any assignment that satisfies F, 0 otherwise.
- Therefore, F is unsatisfiable iff $f \in (x_1^2 - x_1, x_2^2 - x_2, \ldots, x_n^2 - x_n)$.

Complexity of Ring Automorphism Problem: Polynomial Representation

- Let $f(x_1, \ldots, x_n) = \prod_{i=1}^{m} p_i$.
- Polynomial f equals 1 on any assignment that satisfies F, 0 otherwise.
- Therefore, F is unsatisfiable iff $f \in (x_1^2 - x_1, x_2^2 - x_2, \ldots, x_n^2 - x_n)$.

Complexity of Ring Automorphism Problem: Polynomial Representation

- Define ring R as:

$$R = F_2[Y_1, Y_2, \ldots, Y_n]/(1+f(Y_1, \ldots, Y_n), Y_1^2 - Y_1, \ldots, Y_n^2 - Y_n).$$

- If F is unsatisfiable then $1 \in (1 + f(Y_1, \ldots, Y_n), Y_1^2 - Y_1, \ldots, Y_n^2 - Y_n)$.
  - Implies that ring R is trivial, i.e., has only zero.
- If F is satisfiable, then $1 + f$ will be of the form $(1 +$ multi-linear terms$)$ modulo the ideal $(Y_1^2 - Y_1, \ldots, Y_n^2 - Y_n)$.
- Therefore, R will be non-trivial, in particular, $1 \neq 0$ in R.

Complexity of Ring Automorphism Problem: Polynomial Representation

- Define ring R as:

  $$R = F_2[Y_1, Y_2, \ldots, Y_n]/(1+f(Y_1, \ldots, Y_n), Y_1^2 - Y_1, \ldots, Y_n^2 - Y_n).$$

- If F is unsatisfiable then $1 \in (1 + f(Y_1, \ldots, Y_n), Y_1^2 - Y_1, \ldots, Y_n^2 - Y_n)$.
  - Implies that ring R is trivial, i.e., has only zero.
- If F is satisfiable, then $1 + f$ will be of the form $(1 +$ multi-linear terms$)$ modulo the ideal $(Y_1^2 - Y_1, \ldots, Y_n^2 - Y_n)$.
- Therefore, R will be non-trivial, in particular, $1 \neq 0$ in R.

Complexity of Ring Automorphism Problem: Polynomial Representation

- Define ring R as:

  $R = F_2[Y_1, Y_2, \ldots, Y_n]/(1+f(Y_1, \ldots, Y_n), Y_1^2 - Y_1, \ldots, Y_n^2 - Y_n)$.

- If F is unsatisfiable then $1 \in (1 + f(Y_1, \ldots, Y_n), Y_1^2 - Y_1, \ldots, Y_n^2 - Y_n)$.
  - Implies that ring R is trivial, i.e., has only zero.
- If F is satisfiable, then $1 + f$ will be of the form $(1 + \text{ multi-linear terms})$ modulo the ideal $(Y_1^2 - Y_1, \ldots, Y_n^2 - Y_n)$.
- Therefore, R will be non-trivial, in particular, $1 \neq 0$ in R.

Complexity of Ring Automorphism Problem: Polynomial Representation

- Now consider the ring $R \oplus R$.
  - If R is trivial, $R \oplus R$ has just one element $(0, 0)$ and so has no non-trivial automorphisms.
  - If R is non-trivial, $R \oplus R$ has a non-trivial automorphism that maps the first copy to the second one and vice-versa. □

The search version of the problem is NP-hard too.

Complexity of Ring Automorphism Problem: Polynomial Representation

- Now consider the ring $R \oplus R$.
  - If R is trivial, $R \oplus R$ has just one element $(0, 0)$ and so has no non-trivial automorphisms.
  - If R is non-trivial, $R \oplus R$ has a non-trivial automorphism that maps the first copy to the second one and vice-versa. $\square$

The search version of the problem is NP-hard too.

Complexity of Ring Automorphism Problem: Polynomial Representation

- Now consider the ring $R \oplus R$.
  - If R is trivial, $R \oplus R$ has just one element $(0, 0)$ and so has no non-trivial automorphisms.
  - If R is non-trivial, $R \oplus R$ has a non-trivial automorphism that maps the first copy to the second one and vice-versa. $\square$

The search version of the problem is NP-hard too.

Outline

Motivation
- Mathematics
- Computer Science

Definitions
- Finite Rings
- Automorphisms and Isomorphisms
- Problems Related to Automorphisms

Complexity of Problems on Different Representations
- Ring Automorphism Problem
- Complexity of Other Problems

Complexity of Testing Ring Automorphism

- The complexity of the problem depends on how the map ϕ is given.
- If given as a polynomial, the Table Representation takes quasi-polynomial time.
- For Basis Representation, it is in coNP.
- For Polynomial Representation, it is NP-hard.

Complexity of Ring Isomorphism Problems

- The results are similar for problems related to ring isomorphisms.
- Ring Isomorphism problem (both versions) takes quasi-polynomial time in Table Representation.
- All the problems are in $\mathsf{FP}^{\mathsf{AM} \cap \mathsf{coAM}}$ in Basis Representation.
- All the problems are coNP-hard in Polynomial Representation.
    - The proof is same as for Ring Automorphism: constructed ring R is isomorphic to trivial ring iff F is unsatisfiable.

Complexity of Ring Isomorphism Problems

- The results are similar for problems related to ring isomorphisms.
- Ring Isomorphism problem (both versions) takes quasi-polynomial time in Table Representation.
- All the problems are in $\mathsf{FP}^{\mathsf{AM} \cap \mathsf{coAM}}$ in Basis Representation.
- All the problems are coNP-hard in Polynomial Representation.
  - The proof is same as for Ring Automorphism: constructed ring R is isomorphic to trivial ring iff F is unsatisfiable.

Complexity of Ring Isomorphism Problems

- The results are similar for problems related to ring isomorphisms.
- Ring Isomorphism problem (both versions) takes quasi-polynomial time in Table Representation.
- All the problems are in $\mathsf{FP}^{\mathsf{AM} \cap \mathsf{coAM}}$ in Basis Representation.
- All the problems are coNP-hard in Polynomial Representation.
  - The proof is same as for Ring Automorphism: constructed ring R is isomorphic to trivial ring iff F is unsatisfiable.

Complexity of Ring Isomorphism Problems

- The results are similar for problems related to ring isomorphisms.
- Ring Isomorphism problem (both versions) takes quasi-polynomial time in Table Representation.
- All the problems are in $\mathsf{FP}^{\mathsf{AM} \cap \mathsf{coAM}}$ in Basis Representation. The Ring Isomorphism problem in not known to be in P.
- All the problems are coNP-hard in Polynomial Representation.
  - The proof is same as for Ring Automorphism: constructed ring R is isomorphic to trivial ring iff F is unsatisfiable.

Complexity of Ring Isomorphism Problems

- The results are similar for problems related to ring isomorphisms.
- Ring Isomorphism problem (both versions) takes quasi-polynomial time in Table Representation.
- All the problems are in $\mathsf{FP}^{\mathsf{AM} \cap \mathsf{coAM}}$ in Basis Representation.
- All the problems are coNP-hard in Polynomial Representation.
    - The proof is same as for Ring Automorphism: constructed ring R is isomorphic to trivial ring iff F is unsatisfiable.

The "Right" Representation

Previous discussion indicates that Table Representation is too verbose (all problems are quasi-polynomial time) ...

- We will now restrict our attention to this representation.
- On the other hand, most "natural" representation is the Polynomial Representation.
- Fortunately, nearly all the rings we will consider, have the nice property that their Basis and Polynomial Representations are of the similar size.
- Hence, we get best of both worlds: study rings in Basis Representation while using Polynomial Representation to refer to them!

The "Right" Representation

... and Polynomial Representation is too compact (all problems are NP-hard).

- We will now restrict our attention to this representation.
- On the other hand, most "natural" representation is the Polynomial Representation.
- Fortunately, nearly all the rings we will consider, have the nice property that their Basis and Polynomial Representations are of the similar size.
- Hence, we get best of both worlds: study rings in Basis Representation while using Polynomial Representation to refer to them!

The "Right" Representation

So the right representation, complexity-wise, is the Basis Representation.

- We will now restrict our attention to this representation.
- On the other hand, most "natural" representation is the Polynomial Representation.
- Fortunately, nearly all the rings we will consider, have the nice property that their Basis and Polynomial Representations are of the similar size.
- Hence, we get best of both worlds: study rings in Basis Representation while using Polynomial Representation to refer to them!

The "Right" Representation

So the right representation, complexity-wise, is the Basis Representation.

- We will now restrict our attention to this representation.
- On the other hand, most "natural" representation is the Polynomial Representation.
- Fortunately, nearly all the rings we will consider, have the nice property that their Basis and Polynomial Representations are of the similar size.
- Hence, we get best of both worlds: study rings in Basis Representation while using Polynomial Representation to refer to them!

The "Right" Representation

So the right representation, complexity-wise, is the Basis Representation.

- We will now restrict our attention to this representation.
- On the other hand, most "natural" representation is the Polynomial Representation.
- Fortunately, nearly all the rings we will consider, have the nice property that their Basis and Polynomial Representations are of the similar size.
- Hence, we get best of both worlds: study rings in Basis Representation while using Polynomial Representation to refer to them!

The "Right" Representation

So the right representation, complexity-wise, is the Basis Representation.

- We will now restrict our attention to this representation.
- On the other hand, most "natural" representation is the Polynomial Representation.
- Fortunately, nearly all the rings we will consider, have the nice property that their Basis and Polynomial Representations are of the similar size.
- Hence, we get best of both worlds: study rings in Basis Representation while using Polynomial Representation to refer to them!

Part II

Automorphisms: Applications

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Primality Testing reduces to Automorphism Testing

- Fermat's Little Theorem shows a weak connection of primality testing with Automorphism Testing.
- However, until recently, no reduction was known from primality testing.
- The recent deterministic primality testing algorithm makes the connection and exploits it.

Primality Testing reduces to Automorphism Testing

- Fermat's Little Theorem shows a weak connection of primality testing with Automorphism Testing.
- However, until recently, no reduction was known from primality testing.
- The recent deterministic primality testing algorithm makes the connection and exploits it.

Primality Testing reduces to Automorphism Testing

- Fermat's Little Theorem shows a weak connection of primality testing with Automorphism Testing.
- However, until recently, no reduction was known from primality testing.
- The recent deterministic primality testing algorithm makes the connection and exploits it.

Primality Testing reduces to Automorphism Testing

Let Z_n be the ring of numbers modulo n.

Theorem (Fermat's Little Theorem)

If n is prime then $x^n = x \pmod{n}$ for every $x \in Z_n$.

We need to reformulate the theorem...

Primality Testing reduces to Automorphism Testing

Let Z_n be the ring of numbers modulo n.

Theorem (Fermat's Little Theorem)

If n is prime then $x^n = x \pmod{n}$ for every $x \in Z_n$.

We need to reformulate the theorem...

Primality Testing reduces to Automorphism Testing

Theorem (Fermat's Little Theorem Reformulated)

If n is prime then the map $\phi : Z_n \mapsto Z_n$, $\phi(x) = x^n \pmod{n}$ is an automorphism of Z_n.

- Holds because Z_n has only trivial automorphism.
- The converse does not hold, so it does not show that primality testing reduces to Automorphism Testing.
- A generalization of FLT provides such a reduction.

Primality Testing reduces to Automorphism Testing

Theorem (Fermat's Little Theorem Reformulated)

If n is prime then the map $\phi : Z_n \mapsto Z_n$, $\phi(x) = x^n \pmod n$ is an automorphism of Z_n.

- Holds because Z_n has only trivial automorphism.
- The converse does not hold, so it does not show that primality testing reduces to Automorphism Testing.
- A generalization of FLT provides such a reduction.

Primality Testing reduces to Automorphism Testing

Theorem (Fermat's Little Theorem Reformulated)

If n is prime then the map $\phi : Z_n \mapsto Z_n$, $\phi(x) = x^n \pmod{n}$ is an automorphism of Z_n.

- Holds because Z_n has only trivial automorphism.
- The converse does not hold, so it does not show that primality testing reduces to Automorphism Testing.
- A generalization of FLT provides such a reduction.

PRIMALITY TESTING REDUCES TO AUTOMORPHISM TESTING

THEOREM (FERMAT'S LITTLE THEOREM REFORMULATED)

If n is prime then the map $\phi : Z_n \mapsto Z_n$, $\phi(x) = x^n \pmod n$ is an automorphism of Z_n.

- Holds because Z_n has only trivial automorphism.
- The converse does not hold, so it does not show that primality testing reduces to Automorphism Testing.
- A generalization of FLT provides such a reduction.

PRIMALITY TESTING REDUCES TO AUTOMORPHISM TESTING

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

LEMMA

ϕ is an automorphism of R iff for every $g(Y) \in R$, $\phi(g(Y)) = g(\phi(Y))$.

PROOF.

- ϕ is multiplicative by definition.
- If ϕ is linear then $\phi(x) = \phi(y)$ implies $\phi(x - y) = (x - y)^n = 0$.
- This is not possible since $Y^r - 1$ is not a perfect power and so ϕ is a bijection too. □

Primality Testing reduces to Automorphism Testing

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

Lemma

ϕ is an automorphism of R iff for every $g(Y) \in R$, $\phi(g(Y)) = g(\phi(Y))$.

Proof.

- ϕ is multiplicative by definition.
- If ϕ is linear then $\phi(x) = \phi(y)$ implies $\phi(x - y) = (x - y)^n = 0$.
- This is not possible since $Y^r - 1$ is not a perfect power and so ϕ is a bijection too. $\square$

Primality Testing reduces to Automorphism Testing

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

Lemma

ϕ is an automorphism of R iff for every $g(Y) \in R$, $\phi(g(Y)) = g(\phi(Y))$.

Proof.

- ϕ is multiplicative by definition.
- If ϕ is linear then $\phi(x) = \phi(y)$ implies $\phi(x - y) = (x - y)^n = 0$.
- This is not possible since $Y^r - 1$ is not a perfect power and so ϕ is a bijection too. □

Primality Testing reduces to Automorphism Testing

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

Lemma

ϕ is an automorphism of R iff for every $g(Y) \in R$, $\phi(g(Y)) = g(\phi(Y))$.

Proof.

- ϕ is multiplicative by definition.
- If ϕ is linear then $\phi(x) = \phi(y)$ implies $\phi(x - y) = (x - y)^n = 0$.
- This is not possible since $Y^r - 1$ is not a perfect power and so ϕ is a bijection too. □

Primality Testing reduces to Automorphism Testing

Let $O_r(n)$ denote the order of n modulo r.

Theorem (A-Kayal-Saxena, 2002)

For any r with $O_r(n) > 4\log^2 n$, if $\phi(Y+a) = \phi(Y) + a$ in R for every $a \leq 2\sqrt{r}\log n$ then either n is a prime power or has a divisor $< r$.

The theorem can be generalized to eliminate prime power case.

Primality Testing reduces to Automorphism Testing

Let $O_r(n)$ denote the order of n modulo r.

Theorem (A-Kayal-Saxena, 2002)

For any r with $O_r(n) > 4\log^2 n$, if $\phi(Y+a) = \phi(Y) + a$ in R for every $a \leq 2\sqrt{r}\log n$ then either n is a prime power or has a divisor $< r$.

The theorem can be generalized to eliminate prime power case.

Primality Testing reduces to Automorphism Testing

Theorem (A-Kayal-Saxena, Generalized)

For any r with $O_r(n) > 4\log^2 n$, if $\phi(Y + a) = \phi(Y) + a$ in R for every $a \leq 2\sqrt{r}\log n$ then either n is a prime or has a divisor $< r$.

- Proof
- This basically says that if ϕ is linear on a few elements then n is a prime except when it has a small divisor.
- By changing the ring, one can eliminate the small divisor case too.

Primality Testing reduces to Automorphism Testing

Theorem (A-Kayal-Saxena, Generalized)

For any r with $O_r(n) > 4\log^2 n$, if $\phi(Y+a) = \phi(Y) + a$ in R for every $a \leq 2\sqrt{r}\log n$ then either n is a prime or has a divisor $< r$.

- Proof
- This basically says that if ϕ is linear on a few elements then n is a prime except when it has a small divisor.
- By changing the ring, one can eliminate the small divisor case too.

Primality Testing reduces to Automorphism Testing

Theorem (A-Kayal-Saxena, Generalized)

For any r with $O_r(n) > 4\log^2 n$, if $\phi(Y+a) = \phi(Y) + a$ in R for every $a \leq 2\sqrt{r}\log n$ then either n is a prime or has a divisor $< r$.

- Proof
- This basically says that if ϕ is linear on a few elements then n is a prime except when it has a small divisor.
- By changing the ring, one can eliminate the small divisor case too.

Primality Testing reduces to Automorphism Testing

- Let ring $S = Z_n[Y]/(Y^{2r} - Y^r) = R \oplus Z_n[Y]/(Y^r)$.
- Map ϕ can easily be extended to S.

Primality Testing reduces to Automorphism Testing

Theorem (AKS Reformulated)

Let r be any number with $O_r(n) > 4\log^2 n$.

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 every $a \leq 2\sqrt{r}\log n$.*

- Proof
- The first part of the theorem reduces primality testing to Automorphism Testing.
- The second part shows that Automorphism Testing for the map ϕ in ring S can be done in polynomial time.

Primality Testing reduces to Automorphism Testing

Theorem (AKS Reformulated)

Let r be any number with $O_r(n) > 4\log^2 n$.

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 every $a \leq 2\sqrt{r}\log n$.*

- Proof
- The first part of the theorem reduces primality testing to Automorphism Testing.
- The second part shows that Automorphism Testing for the map ϕ in ring S can be done in polynomial time.

Primality Testing reduces to Automorphism Testing

Theorem (AKS Reformulated)

Let r be any number with $O_r(n) > 4\log^2 n$.

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 every $a \leq 2\sqrt{r}\log n$.*

- Proof
- The first part of the theorem reduces primality testing to Automorphism Testing.
- The second part shows that Automorphism Testing for the map ϕ in ring S can be done in polynomial time.

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Polynomial Factoring Using Automorphisms Over Finite Fields

- A finite field F_q of characteristic p, $q = p^\ell$, has exactly ℓ automorphisms.
- These are ψ, ψ^2, ..., $\psi^{\ell-1}$ with $\psi(x) = x^p$.
- These automorphisms play a crucial role in factoring polynomials over F_q.

Polynomial Factoring Using Automorphisms Over Finite Fields

- A finite field F_q of characteristic p, $q = p^\ell$, has exactly ℓ automorphisms.
- These are ψ, ψ^2, ..., $\psi^{\ell-1}$ with $\psi(x) = x^p$.
- These automorphisms play a crucial role in factoring polynomials over F_q.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Let $f(x)$ be a univariate, degree d polynomial over finite field F_q.
- Assume that f is square-free. If not, its can be factored by computing $\gcd(f(x), f'(x))$.
- Define the ring $R = F_q[Y]/(f(Y))$.
- If f is irreducible, then R is a field of size q^d.
- Else, it is a product of smaller fields.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Let $f(x)$ be a univariate, degree d polynomial over finite field F_q.
- Assume that f is square-free. If not, its can be factored by computing $\gcd(f(x), f'(x))$.
- Define the ring $R = F_q[Y]/(f(Y))$.
- If f is irreducible, then R is a field of size q^d.
- Else, it is a product of smaller fields.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Let $f(x)$ be a univariate, degree d polynomial over finite field F_q.
- Assume that f is square-free. If not, its can be factored by computing $\gcd(f(x), f'(x))$.
- Define the ring $R = F_q[Y]/(f(Y))$.
- If f is irreducible, then R is a field of size q^d.
- Else, it is a product of smaller fields.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Let $f(x)$ be a univariate, degree d polynomial over finite field F_q.
- Assume that f is square-free. If not, its can be factored by computing $\gcd(f(x), f'(x))$.
- Define the ring $R = F_q[Y]/(f(Y))$.
- If f is irreducible, then R is a field of size q^d.
- Else, it is a product of smaller fields.

Polynomial Factoring Using Automorphisms Over Finite Fields

- This difference can be used to factor f into equal degree factors.
- Let $f = \prod_{i=1}^{t} f_i$ with each f_i being a product of irreducible polynomials of degree d_i and $d_1 < d_2 < \cdots < d_t$.
- Then, letting $R_i = F_q[Y]/(f_i(Y))$, $R = \oplus_{i=1}^{t} R_i$.
- Further, ψ^{d_i} is trivial automorphism in ring R_i but not in any other R_j.
- Notice that ψ^{d_i} is trivial in R_i iff $f_i(Y)$ divides $Y^{q^{d_i}} - Y$.
- Therefore, $\gcd(Y^{q^{d_i}} - Y, f(Y)) = f_i(Y)$.

Polynomial Factoring Using Automorphisms Over Finite Fields

- This difference can be used to factor f into equal degree factors.
- Let $f = \prod_{i=1}^{t} f_i$ with each f_i being a product of irreducible polynomials of degree d_i and $d_1 < d_2 < \cdots < d_t$.
- Then, letting $R_i = F_q[Y]/(f_i(Y))$, $R = \oplus_{i=1}^{t} R_i$.
- Further, ψ^{d_i} is trivial automorphism in ring R_i but not in any other R_j.
- Notice that ψ^{d_i} is trivial in R_i iff $f_i(Y)$ divides $Y^{q^{d_i}} - Y$.
- Therefore, $\gcd(Y^{q^{d_i}} - Y, f(Y)) = f_i(Y)$.

Polynomial Factoring Using Automorphisms Over Finite Fields

- This difference can be used to factor f into equal degree factors.
- Let $f = \prod_{i=1}^{t} f_i$ with each f_i being a product of irreducible polynomials of degree d_i and $d_1 < d_2 < \cdots < d_t$.
- Then, letting $R_i = F_q[Y]/(f_i(Y))$, $R = \oplus_{i=1}^{t} R_i$.
- Further, ψ^{d_i} is trivial automorphism in ring R_i but not in any other R_j.
- Notice that ψ^{d_i} is trivial in R_i iff $f_i(Y)$ divides $Y^{q^{d_i}} - Y$.
- Therefore, $\gcd(Y^{q^{d_i}} - Y, f(Y)) = f_i(Y)$.

Polynomial Factoring Using Automorphisms Over Finite Fields

- This difference can be used to factor f into equal degree factors.
- Let $f = \prod_{i=1}^{t} f_i$ with each f_i being a product of irreducible polynomials of degree d_i and $d_1 < d_2 < \cdots < d_t$.
- Then, letting $R_i = F_q[Y]/(f_i(Y))$, $R = \oplus_{i=1}^{t} R_i$.
- Further, ψ^{d_i} is trivial automorphism in ring R_i but not in any other R_j.
- Notice that ψ^{d_i} is trivial in R_i iff $f_i(Y)$ divides $Y^{q^{d_i}} - Y$.
- Therefore, $\gcd(Y^{q^{d_i}} - Y, f(Y)) = f_i(Y)$.

Polynomial Factoring Using Automorphisms Over Finite Fields

- This difference can be used to factor f into equal degree factors.
- Let $f = \prod_{i=1}^{t} f_i$ with each f_i being a product of irreducible polynomials of degree d_i and $d_1 < d_2 < \cdots < d_t$.
- Then, letting $R_i = F_q[Y]/(f_i(Y))$, $R = \oplus_{i=1}^{t} R_i$.
- Further, ψ^{d_i} is trivial automorphism in ring R_i but not in any other R_j.
- Notice that ψ^{d_i} is trivial in R_i iff $f_i(Y)$ divides $Y^{q^{d_i}} - Y$.
- Therefore, $\gcd(Y^{q^{d_i}} - Y, f(Y)) = f_i(Y)$.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Next step is to transform the problem to root finding in F_q.
- Let f be a polynomial of degree d such that all its irreducible factors have degree d_0.
- Let $f = \prod_{i=1}^{\frac{d}{d_0}} f_i$ and consider ring $R = F_q[Y]/(f(Y))$.
- Find a $h(Y) \in R - F_q$ such that $\psi(h(Y)) = h(Y)$.
- If f is reducible then $h(Y)$ exists, and can be computed easily using linear algebra.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Next step is to transform the problem to root finding in F_q.
- Let f be a polynomial of degree d such that all its irreducible factors have degree d_0.
- Let $f = \prod_{i=1}^{\frac{d}{d_0}} f_i$ and consider ring $R = F_q[Y]/(f(Y))$.
- Find a $h(Y) \in R - F_q$ such that $\psi(h(Y)) = h(Y)$.
- If f is reducible then $h(Y)$ exists, and can be computed easily using linear algebra.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Next step is to transform the problem to root finding in F_q.
- Let f be a polynomial of degree d such that all its irreducible factors have degree d_0.
- Let $f = \prod_{i=1}^{\frac{d}{d_0}} f_i$ and consider ring $R = F_q[Y]/(f(Y))$.
- Find a $h(Y) \in R - F_q$ such that $\psi(h(Y)) = h(Y)$.
- If f is reducible then $h(Y)$ exists, and can be computed easily using linear algebra.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Next step is to transform the problem to root finding in F_q.
- Let f be a polynomial of degree d such that all its irreducible factors have degree d_0.
- Let $f = \prod_{i=1}^{\frac{d}{d_0}} f_i$ and consider ring $R = F_q[Y]/(f(Y))$.
- Find a $h(Y) \in R - F_q$ such that $\psi(h(Y)) = h(Y)$.
- If f is reducible then $h(Y)$ exists, and can be computed easily using linear algebra.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Now compute $u(x) = \text{Res}(h(Y) - x, f(Y))$.
- Notice that $h(Y) = c_i \ (mod\ f_i(Y))$ for $c_i \in F_q$ for each i.
- Fix any i. c_i is a root of $u(x)$ by the property of resultants.
- Since $h(Y) \notin F_q$, there exist j such that $c_i \neq c_j$.
- So, f_i will divide $h(Y) - c_i$ but not f_j.
- Therefore, any root of $u(x)$ in F_q will lead to a factor of f.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Now compute $u(x) = \text{Res}(h(Y) - x, f(Y))$.
- Notice that $h(Y) = c_i \ (mod\ f_i(Y))$ for $c_i \in F_q$ for each i.
- Fix any i. c_i is a root of $u(x)$ by the property of resultants.
- Since $h(Y) \notin F_q$, there exist j such that $c_i \neq c_j$.
- So, f_i will divide $h(Y) - c_i$ but not f_j.
- Therefore, any root of $u(x)$ in F_q will lead to a factor of f.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Now compute $u(x) = \text{Res}(h(Y) - x, f(Y))$.
- Notice that $h(Y) = c_i \ (mod\ f_i(Y))$ for $c_i \in F_q$ for each i.
- Fix any i. c_i is a root of $u(x)$ by the property of resultants.
- Since $h(Y) \notin F_q$, there exist j such that $c_i \neq c_j$.
- So, f_i will divide $h(Y) - c_i$ but not f_j.
- Therefore, any root of $u(x)$ in F_q will lead to a factor of f.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Now compute $u(x) = \text{Res}(h(Y) - x, f(Y))$.
- Notice that $h(Y) = c_i \ (mod\ f_i(Y))$ for $c_i \in F_q$ for each i.
- Fix any i. c_i is a root of $u(x)$ by the property of resultants.
- Since $h(Y) \notin F_q$, there exist j such that $c_i \neq c_j$.
- So, f_i will divide $h(Y) - c_i$ but not f_j.
- Therefore, any root of $u(x)$ in F_q will lead to a factor of f.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Now compute $u(x) = \text{Res}(h(Y) - x, f(Y))$.
- Notice that $h(Y) = c_i \ (mod\ f_i(Y))$ for $c_i \in F_q$ for each i.
- Fix any i. c_i is a root of $u(x)$ by the property of resultants.
- Since $h(Y) \notin F_q$, there exist j such that $c_i \neq c_j$.
- So, f_i will divide $h(Y) - c_i$ but not f_j.
- Therefore, any root of $u(x)$ in F_q will lead to a factor of f.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Finally, to find a root of $u(x)$ in F_q, first compute $v(x) = \text{gcd}(u(x), \psi(x) - x)$.
- Polynomial $v(x)$ contains all the roots of $u(x)$ and factors completely over F_q.
- If $\deg(v) > 1$, for a random $a \in F_q$, consider $v(x^2 + a)$.
- With high probability, at least one irreducible factor of $v(x^2 + a)$ will be linear and at least one will be quadratic.
- Now use earlier equal degree factorization to factor $v(x^2 + a)$ and hence $v(x)$.
- Repeat this until all factors of v are computed giving all the roots of u.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Finally, to find a root of $u(x)$ in F_q, first compute $v(x) = \mathrm{gcd}(u(x), \psi(x) - x)$.
- Polynomial $v(x)$ contains all the roots of $u(x)$ and factors completely over F_q.
- If $\deg(v) > 1$, for a random $a \in F_q$, consider $v(x^2 + a)$.
- With high probability, at least one irreducible factor of $v(x^2 + a)$ will be linear and at least one will be quadratic.
- Now use earlier equal degree factorization to factor $v(x^2 + a)$ and hence $v(x)$.
- Repeat this until all factors of v are computed giving all the roots of u.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Finally, to find a root of $u(x)$ in F_q, first compute $v(x) = \text{gcd}(u(x), \psi(x) - x)$.
- Polynomial $v(x)$ contains all the roots of $u(x)$ and factors completely over F_q.
- If $\deg(v) > 1$, for a random $a \in F_q$, consider $v(x^2 + a)$.
- With high probability, at least one irreducible factor of $v(x^2 + a)$ will be linear and at least one will be quadratic.
- Now use earlier equal degree factorization to factor $v(x^2 + a)$ and hence $v(x)$.
- Repeat this until all factors of v are computed giving all the roots of u.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Finally, to find a root of $u(x)$ in F_q, first compute $v(x) = \text{gcd}(u(x), \psi(x) - x)$.
- Polynomial $v(x)$ contains all the roots of $u(x)$ and factors completely over F_q.
- If $\deg(v) > 1$, for a random $a \in F_q$, consider $v(x^2 + a)$.
- With high probability, at least one irreducible factor of $v(x^2 + a)$ will be linear and at least one will be quadratic.
- Now use earlier equal degree factorization to factor $v(x^2 + a)$ and hence $v(x)$.
- Repeat this until all factors of v are computed giving all the roots of u.

Polynomial Factoring Using Automorphisms Over Finite Fields

- Finally, to find a root of $u(x)$ in F_q, first compute $v(x) = \mathrm{gcd}(u(x), \psi(x) - x)$.
- Polynomial $v(x)$ contains all the roots of $u(x)$ and factors completely over F_q.
- If $\deg(v) > 1$, for a random $a \in F_q$, consider $v(x^2 + a)$.
- With high probability, at least one irreducible factor of $v(x^2 + a)$ will be linear and at least one will be quadratic.
- Now use earlier equal degree factorization to factor $v(x^2 + a)$ and hence $v(x)$.
- Repeat this until all factors of v are computed giving all the roots of u.

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Factoring Polynomials Over Rationals

- Let f be given univariate polynomial.
- Choose a small prime p and factor f over F_p.
- Use Hensel Lifting to obtain factors of f over Z_{p^ℓ} for a small ℓ.
- Use LLL algorithm for computing short vector in a lattice to compute a factor of f over rationals.

Factoring Polynomials Over Rationals

- Let f be given univariate polynomial.
- Choose a small prime p and factor f over F_p.
- Use Hensel Lifting to obtain factors of f over Z_{p^ℓ} for a small ℓ.
- Use LLL algorithm for computing short vector in a lattice to compute a factor of f over rationals.

FACTORING POLYNOMIALS OVER RATIONALS

- Let f be given univariate polynomial.
- Choose a small prime p and factor f over F_p.
- Use Hensel Lifting to obtain factors of f over Z_{p^ℓ} for a small ℓ.
- Use LLL algorithm for computing short vector in a lattice to compute a factor of f over rationals.

Factoring Polynomials Over Rationals

- Let f be given univariate polynomial.
- Choose a small prime p and factor f over F_p.
- Use Hensel Lifting to obtain factors of f over Z_{p^ℓ} for a small ℓ.
- Use LLL algorithm for computing short vector in a lattice to compute a factor of f over rationals.

Factoring Multivariate Polynomials

- Use Hilbert's Irreducibility Theorem to reduce the problem of factoring multivariate polynomials to that of factoring bivariate polynomials.
- Use a generalization of univariate factoring to compute factors of bivariate polynomials.

Factoring Multivariate Polynomials

- Use Hilbert's Irreducibility Theorem to reduce the problem of factoring multivariate polynomials to that of factoring bivariate polynomials.
- Use a generalization of univariate factoring to compute factors of bivariate polynomials.

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Factoring Integers Using Ring Automorphism Problem

- There exist several algorithms for factoring integers.
- The most important ones are: Elliptic Curve Factoring, Quadratic Sieve, Number Field Sieve.
- The fastest known algorithm is Number Field Sieve with a conjectured time complexity of $e^{c(\log n)^{1/3}(\log\log n)^{2/3}}$, $c \approx 1.903$.
  - This is discounting the factoring algorithm on quantum computers.
- Many of these algorithms are closely connected to computing automorphisms in rings.
- We will consider the two sieve algorithms.

Factoring Integers Using Ring Automorphism Problem

- There exist several algorithms for factoring integers.
- The most important ones are: Elliptic Curve Factoring, Quadratic Sieve, Number Field Sieve.
- The fastest known algorithm is Number Field Sieve with a conjectured time complexity of $e^{c(\log n)^{1/3}(\log\log n)^{2/3}}$, $c \approx 1.903$.
  - This is discounting the factoring algorithm on quantum computers.
- Many of these algorithms are closely connected to computing automorphisms in rings.
- We will consider the two sieve algorithms.

Factoring Integers Using Ring Automorphism Problem

- There exist several algorithms for factoring integers.
- The most important ones are: Elliptic Curve Factoring, Quadratic Sieve, Number Field Sieve.
- The fastest known algorithm is Number Field Sieve with a conjectured time complexity of $e^{c(\log n)^{1/3}(\log\log n)^{2/3}}$, $c \approx 1.903$.
  - This is discounting the factoring algorithm on quantum computers.
- Many of these algorithms are closely connected to computing automorphisms in rings.
- We will consider the two sieve algorithms.

Quadratic and Number Field Sieve

- Both the algorithms aim to compute a non-trivial solution of the equation
$$x^2 = y^2 \ (mod\ n).$$

- Given a non-trivial solution (x_0, y_0), i.e., $x_0 \neq y_0 \ (mod\ n)$, n can be factored easily:
  - n divides $x_0^2 - y_0^2$ but not $x_0 - y_0$ or $x_0 + y_0$.
  - Hence $\gcd(n, x_0 + y_0)$ will yield a factor of n.
- The process of computing the solution is different in both though.
- For our purposes, the process used is not relevant.

Quadratic and Number Field Sieve

- Both the algorithms aim to compute a non-trivial solution of the equation
$$x^2 = y^2 \ (mod\ n).$$
- Given a non-trivial solution (x_0, y_0), i.e., $x_0 \neq y_0\ (mod\ n)$, n can be factored easily:
  - n divides $x_0^2 - y_0^2$ but not $x_0 - y_0$ or $x_0 + y_0$.
  - Hence $\gcd(n, x_0 + y_0)$ will yield a factor of n.
- The process of computing the solution is different in both though.
- For our purposes, the process used is not relevant.

Quadratic and Number Field Sieve

- Both the algorithms aim to compute a non-trivial solution of the equation
$$x^2 = y^2 \ (mod\ n).$$
- Given a non-trivial solution (x_0, y_0), i.e., $x_0 \neq y_0 \ (mod\ n)$, n can be factored easily:
  - n divides $x_0^2 - y_0^2$ but not $x_0 - y_0$ or $x_0 + y_0$.
  - Hence $\gcd(n, x_0 + y_0)$ will yield a factor of n.
- The process of computing the solution is different in both though.
- For our purposes, the process used is not relevant.

Quadratic and Number Field Sieve

- Both the algorithms aim to compute a non-trivial solution of the equation
$$x^2 = y^2 \ (mod\ n).$$
- Given a non-trivial solution (x_0, y_0), i.e., $x_0 \neq y_0 \ (mod\ n)$, n can be factored easily:
  - n divides $x_0^2 - y_0^2$ but not $x_0 - y_0$ or $x_0 + y_0$.
  - Hence $\gcd(n, x_0 + y_0)$ will yield a factor of n.
- The process of computing the solution is different in both though.
- For our purposes, the process used is not relevant.

Sieve Algorithms and Finding Automorphisms

- Let ring $R = Z_n[Y]/(Y^2 - 1)$.
- This ring has two trivial automorphisms specified by: $\phi_0(Y) = Y$ and $\phi_1(Y) = -Y$.
- Finding any other automorphism in the ring is equivalent to factoring n!

Sieve Algorithms and Finding Automorphisms

- Let ring $R = Z_n[Y]/(Y^2 - 1)$.
- This ring has two trivial automorphisms specified by: $\phi_0(Y) = Y$ and $\phi_1(Y) = -Y$.
- Finding any other automorphism in the ring is equivalent to factoring n!

Sieve Algorithms and Finding Automorphisms

Theorem

Factoring odd n is equivalent to finding a non-trivial automorphism of ring R.

Sieve Algorithms and Finding Automorphisms

Proof.

- Let $\phi(Y) = a \cdot Y + b$ be a non-trivial automorphism of R.
- Let $d = (a, n)$.
- Consider $\phi(\frac{n}{d}Y) = \frac{n}{d} \cdot a \cdot Y + \frac{n}{d} \cdot b = \frac{n}{d} \cdot b$.
- Since ϕ is a 1-1 map, this is only possible when $d = (a, n) = 1$.

Sieve Algorithms and Finding Automorphisms

Proof.

- Let $\phi(Y) = a \cdot Y + b$ be a non-trivial automorphism of R.
- Let $d = (a, n)$.
- Consider $\phi(\frac{n}{d} Y) = \frac{n}{d} \cdot a \cdot Y + \frac{n}{d} \cdot b = \frac{n}{d} \cdot b$.
- Since ϕ is a 1-1 map, this is only possible when $d = (a, n) = 1$.

Sieve Algorithms and Finding Automorphisms

- We have:

$$0 = \phi(Y^2 - 1) = (aY + b)^2 - 1 = 2abY + a^2 + b^2 - 1$$

in the ring.

- This gives $2ab = 0 = a^2 + b^2 - 1 \ (mod\ n)$.
- Since n is odd and $(a, n) = 1$, we get $b = 0 \ (mod\ n)$ and $a^2 = 1 \ (mod\ n)$.
- Therefore, $\phi(Y) = a \cdot Y$ with $a^2 = 1 \ (mod\ n)$.
- As ϕ is non-trivial, $a \neq \pm 1 \ (mod\ n)$.
- So, given ϕ, we can use a to factor n.

Sieve Algorithms and Finding Automorphisms

- We have:

$$0 = \phi(Y^2 - 1) = (aY + b)^2 - 1 = 2abY + a^2 + b^2 - 1$$

  in the ring.
- This gives $2ab = 0 = a^2 + b^2 - 1 \ (mod\ n)$.
- Since n is odd and $(a, n) = 1$, we get $b = 0 \ (mod\ n)$ and $a^2 = 1 \ (mod\ n)$.
- Therefore, $\phi(Y) = a \cdot Y$ with $a^2 = 1 \ (mod\ n)$.
- As ϕ is non-trivial, $a \neq \pm 1 \ (mod\ n)$.
- So, given ϕ, we can use a to factor n.

Sieve Algorithms and Finding Automorphisms

- We have:

$$0 = \phi(Y^2 - 1) = (aY + b)^2 - 1 = 2abY + a^2 + b^2 - 1$$

  in the ring.
- This gives $2ab = 0 = a^2 + b^2 - 1 \ (mod\ n)$.
- Since n is odd and $(a, n) = 1$, we get $b = 0 \ (mod\ n)$ and $a^2 = 1 \ (mod\ n)$.
- Therefore, $\phi(Y) = a \cdot Y$ with $a^2 = 1 \ (mod\ n)$.
- As ϕ is non-trivial, $a \neq \pm 1 \ (mod\ n)$.
- So, given ϕ, we can use a to factor n.

Sieve Algorithms and Finding Automorphisms

- We have:

$$0 = \phi(Y^2 - 1) = (aY + b)^2 - 1 = 2abY + a^2 + b^2 - 1$$

  in the ring.
- This gives $2ab = 0 = a^2 + b^2 - 1 \ (mod\ n)$.
- Since n is odd and $(a, n) = 1$, we get $b = 0 \ (mod\ n)$ and $a^2 = 1 \ (mod\ n)$.
- Therefore, $\phi(Y) = a \cdot Y$ with $a^2 = 1 \ (mod\ n)$.
- As ϕ is non-trivial, $a \neq \pm 1 \ (mod\ n)$.
- So, given ϕ, we can use a to factor n.

Sieve Algorithms and Finding Automorphisms

- Conversely, assume that we know a prime factorization of n.
- Then, it is easy to construct a number a such that $a \neq \pm 1 \ (mod\ n)$ and $a^2 = 1 \ (mod\ n)$.
- This a defines a non-trivial automorphism of R. □

Therefore, the Sieve methods are equivalent to finding a non-trivial automorphism in a ring.

SIEVE ALGORITHMS AND FINDING AUTOMORPHISMS

- Conversely, assume that we know a prime factorization of n.
- Then, it is easy to construct a number a such that $a \neq \pm 1 \ (mod\ n)$ and $a^2 = 1 \ (mod\ n)$.
- This a defines a non-trivial automorphism of R. $\square$

Therefore, the Sieve methods are equivalent to finding a non-trivial automorphism in a ring.

Sieve Algorithms and Finding Automorphisms

- Conversely, assume that we know a prime factorization of n.
- Then, it is easy to construct a number a such that $a \neq \pm 1 \ (mod\ n)$ and $a^2 = 1 \ (mod\ n)$.
- This a defines a non-trivial automorphism of R. □

Therefore, the Sieve methods are equivalent to finding a non-trivial automorphism in a ring.

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Reducing Factoring to Other Rings

- Let $R_f = Z_n[Y]/(f(Y))$ where f is a degree 3 polynomial.
- For the sake of simplicity, assume that $n = p \cdot q$ where p and q are distinct primes.

Theorem (Kayal and Saxena, 2004)

Number n can be efficiently factored iff a non-trivial automorphism of R_f can be efficiently computed for every f.

Reducing Factoring to Other Rings

- Let $R_f = Z_n[Y]/(f(Y))$ where f is a degree 3 polynomial.
- For the sake of simplicity, assume that $n = p \cdot q$ where p and q are distinct primes.

Theorem (Kayal and Saxena, 2004)

Number n can be efficiently factored iff a non-trivial automorphism of R_f can be efficiently computed for every f.

Reducing Factoring to Other Rings

- Let $R_f = Z_n[Y]/(f(Y))$ where f is a degree 3 polynomial.
- For the sake of simplicity, assume that $n = p \cdot q$ where p and q are distinct primes.

Theorem (Kayal and Saxena, 2004)

Number n can be efficiently factored iff a non-trivial automorphism of R_f can be efficiently computed for every f.

Reducing Factoring to Other Rings

Proof.

- If factors of n are known, a non-trivial automorphism of R_f can be computed easily.
  - If f factors completely modulo p, then construct a non-trivial automorphism by permuting roots of f modulo p.
  - If f does not factor completely, then $\phi(x) = x^p$ is a non-trivial automorphism modulo p.
  - Either of above two can be combined with trivial automorphism modulo q to yield a non-trivial automorphism of R_f.

Reducing Factoring to Other Rings

Proof.

- If factors of n are known, a non-trivial automorphism of R_f can be computed easily.
  - If f factors completely modulo p, then construct a non-trivial automorphism by permuting roots of f modulo p.
  - If f does not factor completely, then $\phi(x) = x^p$ is a non-trivial automorphism modulo p.
  - Either of above two can be combined with trivial automorphism modulo q to yield a non-trivial automorphism of R_f.

REDUCING FACTORING TO OTHER RINGS

PROOF.

- If factors of n are known, a non-trivial automorphism of R_f can be computed easily.
  - If f factors completely modulo p, then construct a non-trivial automorphism by permuting roots of f modulo p.
  - If f does not factor completely, then $\phi(x) = x^p$ is a non-trivial automorphism modulo p.
  - Either of above two can be combined with trivial automorphism modulo q to yield a non-trivial automorphism of R_f.

Reducing Factoring to Other Rings

Proof.

- If factors of n are known, a non-trivial automorphism of R_f can be computed easily.
  - If f factors completely modulo p, then construct a non-trivial automorphism by permuting roots of f modulo p.
  - If f does not factor completely, then $\phi(x) = x^p$ is a non-trivial automorphism modulo p.
  - Either of above two can be combined with trivial automorphism modulo q to yield a non-trivial automorphism of R_f.

Reducing Factoring to Other Rings

- Conversely, assume that a non-trivial automorphism of R_f can be computed for any f.
- Randomly select an f of degree 3.
- With probability at least $\frac{1}{9}$, f will be irreducible modulo p and factor into two irreducible factors modulo q.
- This implies

$$R_f = F_{p^3} \oplus F_q \oplus F_{q^2}.$$

- Let ψ be a non-trivial automorphism of R_f.
- Compute the set $S = \{x \in R_f \mid \psi(x) = x\}$.

Reducing Factoring to Other Rings

- Conversely, assume that a non-trivial automorphism of R_f can be computed for any f.
- Randomly select an f of degree 3.
- With probability at least $\frac{1}{9}$, f will be irreducible modulo p and factor into two irreducible factors modulo q.
- This implies

$$R_f = F_{p^3} \oplus F_q \oplus F_{q^2}.$$

- Let ψ be a non-trivial automorphism of R_f.
- Compute the set $S = \{x \in R_f \mid \psi(x) = x\}$.

Reducing Factoring to Other Rings

- Conversely, assume that a non-trivial automorphism of R_f can be computed for any f.
- Randomly select an f of degree 3.
- With probability at least $\frac{1}{9}$, f will be irreducible modulo p and factor into two irreducible factors modulo q.
- This implies

$$R_f = F_{p^3} \oplus F_q \oplus F_{q^2}.$$

- Let ψ be a non-trivial automorphism of R_f.
- Compute the set $S = \{x \in R_f \mid \psi(x) = x\}$.

Reducing Factoring to Other Rings

There are now three cases:

Case 1. ψ fixes F_{p^3}.

- In this case, $|S| = p^3 \cdot q^2$.

Case 2. ψ fixes F_{q^2}.

- In this case, $|S| = p \cdot q^3$.

Case 3. ψ fixes neither.

- In this case, $|S| = p \cdot q^2$.

Reducing Factoring to Other Rings

There are now three cases:

Case 1. ψ fixes F_{p^3}.

- In this case, $|S| = p^3 \cdot q^2$.

Case 2. ψ fixes F_{q^2}.

- In this case, $|S| = p \cdot q^3$.

Case 3. ψ fixes neither.

- In this case, $|S| = p \cdot q^2$.

Reducing Factoring to Other Rings

There are now three cases:

Case 1. ψ fixes F_{p^3}.

- In this case, $|S| = p^3 \cdot q^2$.

Case 2. ψ fixes F_{q^2}.

- In this case, $|S| = p \cdot q^3$.

Case 3. ψ fixes neither.

- In this case, $|S| = p \cdot q^2$.

Reducing Factoring to Other Rings

- In either of the three cases, $\frac{|S|}{n}$ or $\frac{|S|}{n^2}$ will yield a factor of n.
- Notice that S can be computed by linear algebra. $\square$

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Graph Isomorphism Using Ring Isomorphism Problem

- Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two undirected graphs on n vertices.
- The Graph Isomorphism problem is to test if G and H are isomorphic.
- Kayal-Saxena (2004) show that the problem reduces to Ring Isomorphism problem.

Graph Isomorphism Using Ring Isomorphism Problem

- Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two undirected graphs on n vertices.
- The Graph Isomorphism problem is to test if G and H are isomorphic.
- Kayal-Saxena (2004) show that the problem reduces to Ring Isomorphism problem.

Graph Isomorphism Using Ring Isomorphism Problem

- For graph G, define the following polynomial:

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

- Now associate an ideal with G:

$$\mathcal{I}_G = (p_G, \{x_i^2\}_{1 \le i \le n}, \{x_i x_j x_k\}_{1 \le i < j < k \le m}).$$

- Finally, define ring R_G as:

$$R_G = F[Y_1, \ldots, Y_n]/\mathcal{I}_G,$$

where F is a field of characteristic $\neq 2$.

Graph Isomorphism Using Ring Isomorphism Problem

- For graph G, define the following polynomial:

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

- Now associate an ideal with G:

$$\mathcal{I}_G = (p_G, \{x_i^2\}_{1 \le i \le n}, \{x_i x_j x_k\}_{1 \le i < j < k \le m}).$$

- Finally, define ring R_G as:

$$R_G = F[Y_1, \ldots, Y_n]/\mathcal{I}_G,$$

where F is a field of characteristic $\neq 2$.

Graph Isomorphism Using Ring Isomorphism Problem

- For graph G, define the following polynomial:

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

- Now associate an ideal with G:

$$\mathcal{I}_G = (p_G, \{x_i^2\}_{1 \leq i \leq n}, \{x_i x_j x_k\}_{1 \leq i < j < k \leq m}).$$

- Finally, define ring R_G as:

$$R_G = F[Y_1, \ldots, Y_n]/\mathcal{I}_G,$$

where F is a field of characteristic $\neq 2$.

Graph Isomorphism Using Ring Isomorphism Problem

- Say that graph G is k-trivial if it is a union of a k-clique and an $n-k$-independent set.

Theorem

Graph G and H are isomorphic iff either they are both k-trivial or ring R_G is isomorphic to R_H.

Graph Isomorphism Using Ring Isomorphism Problem

Proof.

- Forward direction is simple.
- Suppose G and H are isomorphic under isomorphism π.
- Then, $p_G(\pi(Y_1), \ldots, \pi(Y_n)) = p_H(Y_1, \ldots, Y_n)$.
- The other two sets of polynomials in the ideals $\mathcal{I}_G$ and $\mathcal{I}_H$ are closed under permutations.
- Therefore, $R_G \equiv R_H$ under isomorphism $\phi(Y_i) = Y_{\pi(i)}$.

Graph Isomorphism Using Ring Isomorphism Problem

Proof.

- Forward direction is simple.
- Suppose G and H are isomorphic under isomorphism π.
- Then, $p_G(\pi(Y_1), \ldots, \pi(Y_n)) = p_H(Y_1, \ldots, Y_n)$.
- The other two sets of polynomials in the ideals $\mathcal{I}_G$ and $\mathcal{I}_H$ are closed under permutations.
- Therefore, $R_G \equiv R_H$ under isomorphism $\phi(Y_i) = Y_{\pi(i)}$.

Graph Isomorphism Using Ring Isomorphism Problem

Proof.

- Forward direction is simple.
- Suppose G and H are isomorphic under isomorphism π.
- Then, $p_G(\pi(Y_1), \ldots, \pi(Y_n)) = p_H(Y_1, \ldots, Y_n)$.
- The other two sets of polynomials in the ideals $\mathcal{I}_G$ and $\mathcal{I}_H$ are closed under permutations.
- Therefore, $R_G \equiv R_H$ under isomorphism $\phi(Y_i) = Y_{\pi(i)}$.

Graph Isomorphism Using Ring Isomorphism Problem

- Conversely, if both G and H are k-trivial then they are clearly isomorphic.
- So assume that R_G and R_H are isomorphic but H is not k-trivial.
- Let ϕ be an isomorphism between R_G and R_H.
- Fix an i, $1 \le i \le n$.
- Let

$$\phi(Y_i) = \alpha + \sum_{j=1}^{n} \beta_j Y_j + \text{ higher order terms.}$$

Graph Isomorphism Using Ring Isomorphism Problem

- Conversely, if both G and H are k-trivial then they are clearly isomorphic.
- So assume that R_G and R_H are isomorphic but H is not k-trivial.
- Let ϕ be an isomorphism between R_G and R_H.
- Fix an i, $1 \leq i \leq n$.
- Let

$$\phi(Y_i) = \alpha + \sum_{j=1}^{n} \beta_j Y_j + \text{ higher order terms.}$$

Graph Isomorphism Using Ring Isomorphism Problem

- We have:

$$0 = \phi(Y_i^2) = \phi^2(Y_i) = \alpha^2 + \text{ higher order terms.}$$

- This gives $\alpha = 0$.
- So,

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

- Therefore,

$$P = \sum_{1 \le j < k \le n} \beta_j \beta_k Y_j Y_k \in \mathcal{I}_H.$$

Graph Isomorphism Using Ring Isomorphism Problem

- We have:

$$0 = \phi(Y_i^2) = \phi^2(Y_i) = \alpha^2 + \text{ higher order terms.}$$

- This gives $\alpha = 0$.
- So,

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

- Therefore,

$$P = \sum_{1 \le j < k \le n} \beta_j \beta_k Y_j Y_k \in \mathcal{I}_H.$$

Graph Isomorphism Using Ring Isomorphism Problem

- We have:

$$0 = \phi(Y_i^2) = \phi^2(Y_i) = \alpha^2 + \text{ higher order terms.}$$

- This gives $\alpha = 0$.
- So,

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

- Therefore,

$$P = \sum_{1 \le j < k \le n} \beta_j \beta_k Y_j Y_k \in \mathcal{I}_H .$$

Graph Isomorphism Using Ring Isomorphism Problem

- This is possible only when polynomial p_H divides P.
- Let $B = \{\beta_j \mid \beta_j \neq 0\}$.
- Then,

$$P = \sum_{j,k \in B, j \neq k} \beta_j \beta_k Y_j Y_j.$$

- Since polynomial p_H is also of degree 2, P must be a constant multiple of p_H.
- Assume that P is not identically zero.

Graph Isomorphism Using Ring Isomorphism Problem

- This is possible only when polynomial p_H divides P.
- Let $B = \{\beta_j \mid \beta_j \neq 0\}$.
- Then,

$$P = \sum_{j,k \in B, j \neq k} \beta_j \beta_k Y_j Y_j.$$

- Since polynomial p_H is also of degree 2, P must be a constant multiple of p_H.
- Assume that P is not identically zero.

Graph Isomorphism Using Ring Isomorphism Problem

- This is possible only when polynomial p_H divides P.
- Let $B = \{\beta_j \mid \beta_j \neq 0\}$.
- Then,

$$P = \sum_{j,k \in B, j \neq k} \beta_j \beta_k Y_j Y_j.$$

- Since polynomial p_H is also of degree 2, P must be a constant multiple of p_H.
- Assume that P is not identically zero.

Graph Isomorphism Using Ring Isomorphism Problem

- Since all non-zero coefficients of p_H are 1, $\beta_j \beta_k$'s must all be the equal.
- Since P is not a zero polynomial, we get

$$p_H = \sum_{j,k \in B, j \neq k} Y_j Y_k,$$

  implying that H is $|B|$-trivial.
- This is not possible by assumption.
- Therefore, P must be a zero polynomial and so, $\beta_j \beta_k = 0$ for $1 \leq j < k \leq n$.

Graph Isomorphism Using Ring Isomorphism Problem

- Since all non-zero coefficients of p_H are 1, $\beta_j\beta_k$'s must all be the equal.
- Since P is not a zero polynomial, we get

$$p_H = \sum_{j,k \in B, j \neq k} Y_j Y_k,$$

implying that H is $|B|$-trivial.
- This is not possible by assumption.
- Therefore, P must be a zero polynomial and so, $\beta_j\beta_k = 0$ for $1 \leq j < k \leq n$.

Graph Isomorphism Using Ring Isomorphism Problem

- Since all non-zero coefficients of p_H are 1, $\beta_j\beta_k$'s must all be the equal.
- Since P is not a zero polynomial, we get

$$p_H = \sum_{j,k \in B, j \neq k} Y_j Y_k,$$

  implying that H is $|B|$-trivial.
- This is not possible by assumption.
- Therefore, P must be a zero polynomial and so, $\beta_j\beta_k = 0$ for $1 \leq j < k \leq n$.

Graph Isomorphism Using Ring Isomorphism Problem

- If $\beta_j = 0$ for all j, then

$$\begin{aligned}\phi(Y_i Y_{i'}) &= \phi(Y_i) \cdot \phi(Y_{i'}) \\ &= (\text{ degree 2 terms}) \cdot (\text{ degree} \geq 1 \text{ terms}) \\ &= 0.\end{aligned}$$

- Since ϕ is 1-1, this is not possible.

Graph Isomorphism Using Ring Isomorphism Problem

- So, there is exactly one β_j which is non-zero.
- Let $\pi(i) = j$.
- Mapping π is 1-1, since if $\pi(i) = \pi(i') = j$ then

$$\begin{aligned}\phi(Y_i Y_{i'}) &= (Y_j + \text{ degree 2 terms}) \cdot (Y_j + \text{ degree 2 terms}) \\ &= 0.\end{aligned}$$

- So, π is a permutation.

Graph Isomorphism Using Ring Isomorphism Problem

- So, there is exactly one β_j which is non-zero.
- Let $\pi(i) = j$.
- Mapping π is 1-1, since if $\pi(i) = \pi(i') = j$ then

$$\begin{aligned} \phi(Y_i Y_{i'}) &= (Y_j + \text{ degree 2 terms}) \cdot (Y_j + \text{ degree 2 terms}) \\ &= 0. \end{aligned}$$

- So, π is a permutation.

Graph Isomorphism Using Ring Isomorphism Problem

- So, there is exactly one β_j which is non-zero.
- Let $\pi(i) = j$.
- Mapping π is 1-1, since if $\pi(i) = \pi(i') = j$ then

$$\begin{aligned}\phi(Y_i Y_{i'}) &= (Y_j + \text{ degree 2 terms}) \cdot (Y_j + \text{ degree 2 terms}) \\ &= 0.\end{aligned}$$

- So, π is a permutation.

Graph Isomorphism Using Ring Isomorphism Problem

- Now apply ϕ to p_G:

$$0 = \phi(p_G) = \sum_{(i,j)\in E_G} \phi(Y_i Y_j) = \sum_{(i,j)\in E_G} Y_{\pi(i)} Y_{\pi(j)}.$$

- Again, this means that p_H divides $\phi(p_G)$.
- This is possible only when $p_H = \phi(p_G)$.
- Therefore, π is an isomorphism between G and H. $\square$

Graph Isomorphism Using Ring Isomorphism Problem

- Now apply ϕ to p_G:

$$0 = \phi(p_G) = \sum_{(i,j)\in E_G} \phi(Y_i Y_j) = \sum_{(i,j)\in E_G} Y_{\pi(i)} Y_{\pi(j)}.$$

- Again, this means that p_H divides $\phi(p_G)$.
- This is possible only when $p_H = \phi(p_G)$.
- Therefore, π is an isomorphism between G and H. $\square$

Graph Isomorphism Using Ring Isomorphism Problem

- Now apply ϕ to p_G:

$$0 = \phi(p_G) = \sum_{(i,j)\in E_G} \phi(Y_i Y_j) = \sum_{(i,j)\in E_G} Y_{\pi(i)} Y_{\pi(j)}.$$

- Again, this means that p_H divides $\phi(p_G)$.
- This is possible only when $p_H = \phi(p_G)$.
- Therefore, π is an isomorphism between G and H. $\square$

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

The Polynomial Equivalence Problem

- Let $p(x_1, \ldots, x_n)$ and $q(x_1, \ldots, x_n)$ be two polynomials over field F.
- Given a $n \times n$ matrix A, an A-transformation of p is the polynomial $p(A(x_1, x_2, \ldots, x_n))$.
- For $A = [a_{i,j}]$,

$$A(x_1, \ldots, x_n) = (\sum_{i=1}^{n} a_{i,1}x_i, \ldots, \sum_{i=1}^{n} a_{i,n}x_i).$$

- Polynomials p and q are equivalent if there exists an invertible matrix A such that

$$q(x_1, \ldots, x_n) = p(A(x_1, \ldots, x_n)).$$

The Polynomial Equivalence Problem

- Let $p(x_1, \ldots, x_n)$ and $q(x_1, \ldots, x_n)$ be two polynomials over field F.
- Given a $n \times n$ matrix A, an A-transformation of p is the polynomial $p(A(x_1, x_2, \ldots, x_n))$.
- For $A = [a_{i,j}]$,

$$A(x_1, \ldots, x_n) = (\sum_{i=1}^{n} a_{i,1} x_i, \ldots, \sum_{i=1}^{n} a_{i,n} x_i).$$

- Polynomials p and q are equivalent if there exists an invertible matrix A such that

$$q(x_1, \ldots, x_n) = p(A(x_1, \ldots, x_n)).$$

The Polynomial Equivalence Problem

- Let $p(x_1, \ldots, x_n)$ and $q(x_1, \ldots, x_n)$ be two polynomials over field F.
- Given a $n \times n$ matrix A, an A-transformation of p is the polynomial $p(A(x_1, x_2, \ldots, x_n))$.
- For $A = [a_{i,j}]$,

$$A(x_1, \ldots, x_n) = (\sum_{i=1}^{n} a_{i,1}x_i, \ldots, \sum_{i=1}^{n} a_{i,n}x_i).$$

- Polynomials p and q are equivalent if there exists an invertible matrix A such that

$$q(x_1, \ldots, x_n) = p(A(x_1, \ldots, x_n)).$$

The Polynomial Equivalence Problem

- Let $p(x_1, \ldots, x_n)$ and $q(x_1, \ldots, x_n)$ be two polynomials over field F.
- Given a $n \times n$ matrix A, an A-transformation of p is the polynomial $p(A(x_1, x_2, \ldots, x_n))$.
- For $A = [a_{i,j}]$,

$$A(x_1, \ldots, x_n) = (\sum_{i=1}^{n} a_{i,1} x_i, \ldots, \sum_{i=1}^{n} a_{i,n} x_i).$$

- Polynomials p and q are equivalent if there exists an invertible matrix A such that

$$q(x_1, \ldots, x_n) = p(A(x_1, \ldots, x_n)).$$

Example

- Let $p(x_1, x_2) = x_1^2 + x_2^2$ and $q(x_1, x_2) = x_1^2 + 2x_2^2 + 2x_1x_2$.
- These two are equivalent under transformation $A(x_1) = x_1 + x_2$ and $A(x_2) = x_2$.

Example

- Let $p(x_1, x_2) = x_1^2 + x_2^2$ and $q(x_1, x_2) = x_1^2 + 2x_2^2 + 2x_1x_2$.
- These two are equivalent under transformation $A(x_1) = x_1 + x_2$ and $A(x_2) = x_2$.

The Polynomial Equivalence Problem

- This problem has been studied for a long time in mathematics.
- Especially, the equivalence of d-forms: homogeneous polynomials of degree d.
- Witt (1937) proved that equivalence of quadratic forms ($=2$-forms) can be decided in polynomial time.
- The question is open for higher degree forms.
- Thomas Thierauf (1998) showed that the problem for general polynomials is in $\mathsf{NP} \cap \mathsf{coAM}$.

The Polynomial Equivalence Problem

- This problem has been studied for a long time in mathematics.
- Especially, the equivalence of d-forms: homogeneous polynomials of degree d.
- Witt (1937) proved that equivalence of quadratic forms ($= 2$-forms) can be decided in polynomial time.
- The question is open for higher degree forms.
- Thomas Thierauf (1998) showed that the problem for general polynomials is in $\mathrm{NP} \cap \mathrm{coAM}$.

The Polynomial Equivalence Problem

- This problem has been studied for a long time in mathematics.
- Especially, the equivalence of d-forms: homogeneous polynomials of degree d.
- Witt (1937) proved that equivalence of quadratic forms ($= 2$-forms) can be decided in polynomial time.
- The question is open for higher degree forms.
- Thomas Thierauf (1998) showed that the problem for general polynomials is in $\mathsf{NP} \cap \mathsf{coAM}$.

The Polynomial Equivalence Problem

We show that:

- The Ring Isomorphism problem reduces to degree 3 polynomial equivalence.
- The Graph Isomorphism problem reduces to cubic form equivalence.
- d-form equivalence, for constant d, reduces to Ring Isomorphism problem (except when the $(d, q-1) > 1$ where q is the size of the underlying field F).

The Polynomial Equivalence Problem

We show that:

- The Ring Isomorphism problem reduces to degree 3 polynomial equivalence.
- The Graph Isomorphism problem reduces to cubic form equivalence.
- d-form equivalence, for constant d, reduces to Ring Isomorphism problem (except when the $(d, q-1) > 1$ where q is the size of the underlying field F).

The Polynomial Equivalence Problem

We show that:

- The Ring Isomorphism problem reduces to degree 3 polynomial equivalence.
- The Graph Isomorphism problem reduces to cubic form equivalence.
- d-form equivalence, for constant d, reduces to Ring Isomorphism problem (except when the $(d, q-1) > 1$ where q is the size of the underlying field F).

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Reducing Ring Isomorphism to Polynomial Equivalence

- Let R and S be two given rings in the Basis Representation.
- Let the given basis for R be $b_1, \ldots, b_m$ and for S be $c_1, \ldots, c_m$.
- Also, let $b_i \cdot b_j = \sum_{k=1}^{m} \beta_{ijk} b_k$ and $c_i \cdot c_j = \sum_{k=1}^{m} \gamma_{ijk} c_k$.
- Define polynomial p_R as:

$$p_R(x_1, \ldots, x_m, z_{1,1}, z_{1,2}, \ldots, z_{m,m}) = \sum_{i=1}^{m} \sum_{j=1}^{m} z_{i,j} \cdot (x_i \cdot x_j - \sum_{k=1}^{m} \beta_{ijk} x_k).$$

- Similarly define the polynomial p_S.

Reducing Ring Isomorphism to Polynomial Equivalence

- Let R and S be two given rings in the Basis Representation.
- Let the given basis for R be $b_1, \ldots, b_m$ and for S be $c_1, \ldots, c_m$.
- Also, let $b_i \cdot b_j = \sum_{k=1}^{m} \beta_{ijk} b_k$ and $c_i \cdot c_j = \sum_{k=1}^{m} \gamma_{ijk} c_k$.
- Define polynomial p_R as:

$$p_R(x_1, \ldots, x_m, z_{1,1}, z_{1,2}, \ldots, z_{m,m}) = \sum_{i=1}^{m} \sum_{j=1}^{m} z_{i,j} \cdot (x_i \cdot x_j - \sum_{k=1}^{m} \beta_{ijk} x_k).$$

- Similarly define the polynomial p_S.

Reducing Ring Isomorphism to Polynomial Equivalence

Theorem

Rings R and S are isomorphic iff polynomials p_R and p_S are equivalent.

Reducing Ring Isomorphism to Polynomial Equivalence

Proof.

- Suppose R and S are isomorphic via isomorphism ϕ.
- Clearly, $\phi(b_i \cdot b_j - \sum_{k=1}^{m} \beta_{ijk} b_k) = 0$ in S.
- So let

$$\phi(b_i \cdot b_j - \sum_{k=1}^{m} \beta_{ijk} b_k) = \sum_{s=1}^{m} \sum_{t=1}^{m} \delta_{ij,st}(c_s \cdot c_t - \sum_{u=1}^{m} \gamma_{stu} c_u).$$

Reducing Ring Isomorphism to Polynomial Equivalence

Proof.

- Suppose R and S are isomorphic via isomorphism ϕ.
- Clearly, $\phi(b_i \cdot b_j - \sum_{k=1}^{m} \beta_{ijk} b_k) = 0$ in S.
- So let

$$\phi(b_i \cdot b_j - \sum_{k=1}^{m} \beta_{ijk} b_k) = \sum_{s=1}^{m} \sum_{t=1}^{m} \delta_{ij,st}(c_s \cdot c_t - \sum_{u=1}^{m} \gamma_{stu} c_u).$$

Reducing Ring Isomorphism to Polynomial Equivalence

- Define map A as:

$$\begin{aligned} A(x_i) &= \phi(x_i) \\ A\Big(\sum_{i=1}^{m}\sum_{j=1}^{m}\delta_{ij,st}z_{i,j}\Big) &= z_{s,t}. \end{aligned}$$

Reducing Ring Isomorphism to Polynomial Equivalence

- Then,

$$
\begin{aligned}
p_R(A(\bar{x},\bar{z})) &= \sum_{i=1}^{m}\sum_{j=1}^{m} A(z_{i,j})\cdot \phi(x_i x_j - \sum_{k=1}^{m}\beta_{ijk}x_k) \\
&= \sum_{i=1}^{m}\sum_{j=1}^{m} A(z_{i,j})\cdot \sum_{s=1}^{m}\sum_{t=1}^{m}\delta_{ij,st}\cdot(x_s x_t - \sum_{u=1}^{m}\gamma_{stu}x_u) \\
&= \sum_{s=1}^{m}\sum_{t=1}^{m} A(\sum_{i=1}^{m}\sum_{j=1}^{m}\delta_{ij,st}z_{i,j})\cdot(x_s x_t - \sum_{u=1}^{m}\gamma_{stu}x_u) \\
&= \sum_{s=1}^{m}\sum_{t=1}^{m} z_{s,t}\cdot(x_s x_t - \sum_{u=1}^{m}\gamma_{stu}x_u) \\
&= p_S.
\end{aligned}
$$

Reducing Ring Isomorphism to Polynomial Equivalence

- Conversely, assume that polynomials p_R and p_S are equivalent.
- Let A be the linear transformation from p_R to p_S.
- It can be shown that $A(z_{i,j})$ is a linear combination of only $z_{s,t}$'s.

We will not prove it as it is messy.

Reducing Ring Isomorphism to Polynomial Equivalence

- Conversely, assume that polynomials p_R and p_S are equivalent.
- Let A be the linear transformation from p_R to p_S.
- It can be shown that $A(z_{i,j})$ is a linear combination of only $z_{s,t}$'s.

We will not prove it as it is messy.

Reducing Ring Isomorphism to Polynomial Equivalence

- Conversely, assume that polynomials p_R and p_S are equivalent.
- Let A be the linear transformation from p_R to p_S.
- It can be shown that $A(z_{i,j})$ is a linear combination of only $z_{s,t}$'s.

We will not prove it as it is messy.

Reducing Ring Isomorphism to Polynomial Equivalence

- Now suppose that $A(x_k)$ contains some $z_{s,t}$'s.
- These $z_{s,t}$'s will all occur in terms of $p_R(A(\bar{x},\bar{z}))$ that have z-degree at least two (follows since $A(z_{i,j})$'s have only $z_{s,t}$'s).
- Since p_S has no terms of z-degree more than one, these terms will cancel out each other.
- Therefore, we can drop $z_{s,t}$'s from $A(x_k)$ and the modified transformation is still an equivalence.
- Now suppose $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is not a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s.

Reducing Ring Isomorphism to Polynomial Equivalence

- Now suppose that $A(x_k)$ contains some $z_{s,t}$'s.
- These $z_{s,t}$'s will all occur in terms of $p_R(A(\bar{x}, \bar{z}))$ that have z-degree at least two (follows since $A(z_{i,j})$'s have only $z_{s,t}$'s).
- Since p_S has no terms of z-degree more than one, these terms will cancel out each other.
- Therefore, we can drop $z_{s,t}$'s from $A(x_k)$ and the modified transformation is still an equivalence.
- Now suppose $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is not a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s.

Reducing Ring Isomorphism to Polynomial Equivalence

- Now suppose that $A(x_k)$ contains some $z_{s,t}$'s.
- These $z_{s,t}$'s will all occur in terms of $p_R(A(\bar{x}, \bar{z}))$ that have z-degree at least two (follows since $A(z_{i,j})$'s have only $z_{s,t}$'s).
- Since p_S has no terms of z-degree more than one, these terms will cancel out each other.
- Therefore, we can drop $z_{s,t}$'s from $A(x_k)$ and the modified transformation is still an equivalence.
- Now suppose $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is not a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s.

Reducing Ring Isomorphism to Polynomial Equivalence

- Now suppose that $A(x_k)$ contains some $z_{s,t}$'s.
- These $z_{s,t}$'s will all occur in terms of $p_R(A(\bar{x}, \bar{z}))$ that have z-degree at least two (follows since $A(z_{i,j})$'s have only $z_{s,t}$'s).
- Since p_S has no terms of z-degree more than one, these terms will cancel out each other.
- Therefore, we can drop $z_{s,t}$'s from $A(x_k)$ and the modified transformation is still an equivalence.
- Now suppose $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is not a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s.

Reducing Ring Isomorphism to Polynomial Equivalence

- Now suppose that $A(x_k)$ contains some $z_{s,t}$'s.
- These $z_{s,t}$'s will all occur in terms of $p_R(A(\bar{x}, \bar{z}))$ that have z-degree at least two (follows since $A(z_{i,j})$'s have only $z_{s,t}$'s).
- Since p_S has no terms of z-degree more than one, these terms will cancel out each other.
- Therefore, we can drop $z_{s,t}$'s from $A(x_k)$ and the modified transformation is still an equivalence.
- Now suppose $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is not a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s.

Reducing Ring Isomorphism to Polynomial Equivalence

- Then
$$A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k) = \sum_{s=1}^{m} \sum_{t=1}^{m} \delta_{ij,st} (x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u) + a_{ij} x_\ell + \cdots$$
for some x_ℓ and $a_{ij} \neq 0$.
- Consider the coefficients of x_ℓ for all i and j.
- The sum of these coefficients must be zero since $p_R(A(\cdot)) = p_S$.
- Therefore,
$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} A(z_{i,j}) = 0.$$
- However, this is not possible since A is invertible.

Reducing Ring Isomorphism to Polynomial Equivalence

- Then

$$A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k) = \sum_{s=1}^{m} \sum_{t=1}^{m} \delta_{ij,st} (x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u) + a_{ij} x_\ell + \cdots$$

  for some x_ℓ and $a_{ij} \neq 0$.
- Consider the coefficients of x_ℓ for all i and j.
- The sum of these coefficients must be zero since $p_R(A(\cdot)) = p_S$.
- Therefore,

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} A(z_{i,j}) = 0.$$

- However, this is not possible since A is invertible.

Reducing Ring Isomorphism to Polynomial Equivalence

- Then

$$A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k) = \sum_{s=1}^{m} \sum_{t=1}^{m} \delta_{ij,st} (x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u) + a_{ij} x_\ell + \cdots$$

  for some x_ℓ and $a_{ij} \neq 0$.
- Consider the coefficients of x_ℓ for all i and j.
- The sum of these coefficients must be zero since $p_R(A(\cdot)) = p_S$.
- Therefore,

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} A(z_{i,j}) = 0.$$

- However, this is not possible since A is invertible.

Reducing Ring Isomorphism to Polynomial Equivalence

- Then
$$A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k) = \sum_{s=1}^{m} \sum_{t=1}^{m} \delta_{ij,st}(x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u) + a_{ij} x_\ell + \cdots$$
for some x_ℓ and $a_{ij} \neq 0$.
- Consider the coefficients of x_ℓ for all i and j.
- The sum of these coefficients must be zero since $p_R(A(\cdot)) = p_S$.
- Therefore,
$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} A(z_{i,j}) = 0.$$
- However, this is not possible since A is invertible.

Reducing Ring Isomorphism to Polynomial Equivalence

- Therefore, $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s for all i and j.
- Let $\phi(b_i) = A(b_i)$ with c_j's replacing x_j's in the RHS.
- ϕ maps ring R to S.
- ϕ is invertible since A is.
- ϕ is a homomorphism since it preserves the zeroes as shown above.
- Hence, ϕ is an isomorphism between R and S. $\square$

Reducing Ring Isomorphism to Polynomial Equivalence

- Therefore, $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s for all i and j.
- Let $\phi(b_i) = A(b_i)$ with c_j's replacing x_j's in the RHS.
- ϕ maps ring R to S.
- ϕ is invertible since A is.
- ϕ is a homomorphism since it preserves the zeroes as shown above.
- Hence, ϕ is an isomorphism between R and S. $\square$

Reducing Ring Isomorphism to Polynomial Equivalence

- Therefore, $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s for all i and j.
- Let $\phi(b_i) = A(b_i)$ with c_j's replacing x_j's in the RHS.
- ϕ maps ring R to S.
- ϕ is invertible since A is.
- ϕ is a homomorphism since it preserves the zeroes as shown above.
- Hence, ϕ is an isomorphism between R and S. □

REDUCING RING ISOMORPHISM TO POLYNOMIAL EQUIVALENCE

- Therefore, $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s for all i and j.
- Let $\phi(b_i) = A(b_i)$ with c_j's replacing x_j's in the RHS.
- ϕ maps ring R to S.
- ϕ is invertible since A is.
- ϕ is a homomorphism since it preserves the zeroes as shown above.
- Hence, ϕ is an isomorphism between R and S. $\square$

Reducing Ring Isomorphism to Polynomial Equivalence

- Therefore, $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s for all i and j.
- Let $\phi(b_i) = A(b_i)$ with c_j's replacing x_j's in the RHS.
- ϕ maps ring R to S.
- ϕ is invertible since A is.
- ϕ is a homomorphism since it preserves the zeroes as shown above.
- Hence, ϕ is an isomorphism between R and S. $\square$

Reducing Ring Isomorphism to Polynomial Equivalence

- Therefore, $A(x_i x_j - \sum_{k=1}^{m} \beta_{ijk} x_k)$ is a linear combination of $x_s x_t - \sum_{u=1}^{m} \gamma_{stu} x_u$'s for all i and j.
- Let $\phi(b_i) = A(b_i)$ with c_j's replacing x_j's in the RHS.
- ϕ maps ring R to S.
- ϕ is invertible since A is.
- ϕ is a homomorphism since it preserves the zeroes as shown above.
- Hence, ϕ is an isomorphism between R and S. $\square$

Reducing Graph Isomorphism to Cubic Form Equivalence

- The polynomials p_R and p_S constructed above are of degree 3 but not homogeneous.
- They can be made homogeneous by multiplying all smaller degree terms with appropriate power of a new variable y.
- However, then the above proof breaks down.
- For rings arising out of Graph Isomorphism reduction, the proof goes through.

Reducing Graph Isomorphism to Cubic Form Equivalence

- The polynomials p_R and p_S constructed above are of degree 3 but not homogeneous.
- They can be made homogeneous by multiplying all smaller degree terms with appropriate power of a new variable y.
- However, then the above proof breaks down.
- For rings arising out of Graph Isomorphism reduction, the proof goes through.

Reducing Graph Isomorphism to Cubic Form Equivalence

- The polynomials p_R and p_S constructed above are of degree 3 but not homogeneous.
- They can be made homogeneous by multiplying all smaller degree terms with appropriate power of a new variable y.
- However, then the above proof breaks down.
- For rings arising out of Graph Isomorphism reduction, the proof goes through.

Outline

Primality Testing

Polynomial Factoring

- Over Finite Fields
- Other Variations

Integer Factoring

- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence

- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Reducing d-Form Equivalence to Ring Isomorphism

- Let p and q be two n-variable d-forms over finite field F of size s.
- Let ring R_p be:

$$R_p = F[x_1, \ldots, x_n]/(p(x_1, \ldots, x_n), \{\prod_{j=1}^{d+1} x_{i_j}\}_{1 \le i_1, \ldots, i_{d+1} \le n}).$$

- Similarly, define ring R_q.

Reducing d-Form Equivalence to Ring Isomorphism

Theorem

For $(d, s-1) = 1$, polynomials p and q are equivalent iff rings R_p and R_q are isomorphic.

Reducing d-Form Equivalence to Ring Isomorphism

Proof.

- If p and q are equivalent via A, then A defines an isomorphism between R_p and R_q.
- Conversely, suppose that R_p and R_q are isomorphic via ϕ.
- Let

$$\phi(x_i) = \alpha + \text{ degree 1 terms } + \text{ higher degree terms.}$$

- $\phi^{d+1}(x_i) = \phi(x_i^{d+1}) = 0$ implies that $\alpha = 0$.

Reducing d-Form Equivalence to Ring Isomorphism

Proof.

- If p and q are equivalent via A, then A defines an isomorphism between R_p and R_q.
- Conversely, suppose that R_p and R_q are isomorphic via ϕ.
- Let

$$\phi(x_i) = \alpha + \text{ degree 1 terms } + \text{ higher degree terms.}$$

- $\phi^{d+1}(x_i) = \phi(x_i^{d+1}) = 0$ implies that $\alpha = 0$.

Reducing d-Form Equivalence to Ring Isomorphism

Proof.

- If p and q are equivalent via A, then A defines an isomorphism between R_p and R_q.
- Conversely, suppose that R_p and R_q are isomorphic via ϕ.
- Let

$$\phi(x_i) = \alpha + \text{ degree 1 terms } + \text{ higher degree terms.}$$

- $\phi^{d+1}(x_i) = \phi(x_i^{d+1}) = 0$ implies that $\alpha = 0$.

Reducing d-Form Equivalence to Ring Isomorphism

Proof.

- If p and q are equivalent via A, then A defines an isomorphism between R_p and R_q.
- Conversely, suppose that R_p and R_q are isomorphic via ϕ.
- Let

$$\phi(x_i) = \alpha + \text{ degree 1 terms } + \text{ higher degree terms.}$$

- $\phi^{d+1}(x_i) = \phi(x_i^{d+1}) = 0$ implies that $\alpha = 0$.

Reducing d-Form Equivalence to Ring Isomorphism

- Let ψ be the "linear part" of ϕ.
- ψ remains an isomorphism between R_p and R_q.
- Moreover, $\psi(p) = cq$ for some $c \in F$.
- Therefore, ψ', $\psi'(x_i) = c^{1/d}\psi(x_i)$, is an equivalence between p and q.
- The d-th root of c will always exist in F if $(d, s-1) = 1$. $\square$

REDUCING d-FORM EQUIVALENCE TO RING ISOMORPHISM

- Let ψ be the "linear part" of ϕ.
- ψ remains an isomorphism between R_p and R_q.
- Moreover, $\psi(p) = cq$ for some $c \in F$.
- Therefore, ψ', $\psi'(x_i) = c^{1/d}\psi(x_i)$, is an equivalence between p and q.
- The d-th root of c will always exist in F if $(d, s-1) = 1$. $\square$

Reducing d-Form Equivalence to Ring Isomorphism

- Let ψ be the "linear part" of ϕ.
- ψ remains an isomorphism between R_p and R_q.
- Moreover, $\psi(p) = cq$ for some $c \in F$.
- Therefore, ψ', $\psi'(x_i) = c^{1/d}\psi(x_i)$, is an equivalence between p and q.
- The d-th root of c will always exist in F if $(d, s-1) = 1$. $\square$

Reducing d-Form Equivalence to Ring Isomorphism

- Let ψ be the "linear part" of ϕ.
- ψ remains an isomorphism between R_p and R_q.
- Moreover, $\psi(p) = cq$ for some $c \in F$.
- Therefore, ψ', $\psi'(x_i) = c^{1/d}\psi(x_i)$, is an equivalence between p and q.
- The d-th root of c will always exist in F if $(d, s-1) = 1$. $\square$

Reducing d-Form Equivalence to Ring Isomorphism

- Let ψ be the "linear part" of ϕ.
- ψ remains an isomorphism between R_p and R_q.
- Moreover, $\psi(p) = cq$ for some $c \in F$.
- Therefore, ψ', $\psi'(x_i) = c^{1/d}\psi(x_i)$, is an equivalence between p and q.
- The d-th root of c will always exist in F if $(d, s-1) = 1$. $\square$

Outline

Primality Testing

Polynomial Factoring
- Over Finite Fields
- Other Variations

Integer Factoring
- Reduction to 2-dim Rings
- Reduction to 3-dim Rings

Graph Isomorphism

Polynomial Equivalence
- Problem Definition
- Reducing Ring Isomorphism to Polynomial Equivalence
- Reducing d-form Equivalence to Ring Isomorphism

Open Questions

Open Questions

- Can integer factoring be done faster using rings other than $Z_n[Y]/(Y^2 - 1)$?
- Can the theory of cubic forms be used to derive an efficient algorithm for Graph Isomorphism?
- Do other algebraic problems, e.g., Discrete Log, reduce to any of automorphism problems?

Open Questions

- Can integer factoring be done faster using rings other than $Z_n[Y]/(Y^2 - 1)$?
- Can the theory of cubic forms be used to derive an efficient algorithm for Graph Isomorphism?
- Do other algebraic problems, e.g., Discrete Log, reduce to any of automorphism problems?

Open Questions

- Can integer factoring be done faster using rings other than $Z_n[Y]/(Y^2 - 1)$?
- Can the theory of cubic forms be used to derive an efficient algorithm for Graph Isomorphism?
- Do other algebraic problems, e.g., Discrete Log, reduce to any of automorphism problems?

Open Questions

- Does Ring Isomorphism problem, at least for small characteristic, reduce to Graph Isomorphism?
- Does the Ring Isomorphism problem reduce to equivalence of cubic forms?
  - We can prove it only for degree 3 polynomials.
- Does the equivalence of degree d polynomials reduce to Ring Isomorphism?
  - We can prove it only for homogeneous degree d polynomials.

Open Questions

- Does Ring Isomorphism problem, at least for small characteristic, reduce to Graph Isomorphism?
- Does the Ring Isomorphism problem reduce to equivalence of cubic forms?
  - We can prove it only for degree 3 polynomials.
- Does the equivalence of degree d polynomials reduce to Ring Isomorphism?
  - We can prove it only for homogeneous degree d polynomials.

Open Questions

- Does Ring Isomorphism problem, at least for small characteristic, reduce to Graph Isomorphism?
- Does the Ring Isomorphism problem reduce to equivalence of cubic forms?
  - We can prove it only for degree 3 polynomials.
- Does the equivalence of degree d polynomials reduce to Ring Isomorphism?
  - We can prove it only for homogeneous degree d polynomials.

Open Questions

- Does Ring Isomorphism problem, at least for small characteristic, reduce to Graph Isomorphism?
- Does the Ring Isomorphism problem reduce to equivalence of cubic forms?
  - We can prove it only for degree 3 polynomials.
- Does the equivalence of degree d polynomials reduce to Ring Isomorphism?
  - We can prove it only for homogeneous degree d polynomials.

Open Questions

- Does Ring Isomorphism problem, at least for small characteristic, reduce to Graph Isomorphism?
- Does the Ring Isomorphism problem reduce to equivalence of cubic forms?
  - We can prove it only for degree 3 polynomials.
- Does the equivalence of degree d polynomials reduce to Ring Isomorphism?
  - We can prove it only for homogeneous degree d polynomials.

Thank You!

Removing Prime Powers

Proof.

- Suppose that $(Y+a)^n = Y^n + a \ (mod\ n, Y^r - 1)$ for $a \leq 2\sqrt{r}\log n$.
- Therefore, $a^n = a \ (mod\ n)$ for $a \leq 2\sqrt{r}\log n$.
- Since $r > 4\log^2 n$, above equation holds for at least $4\log^2 n$ a's.

Removing Prime Powers

Proof.

- Suppose that $(Y + a)^n = Y^n + a \ (mod \ n, Y^r - 1)$ for $a \leq 2\sqrt{r} \log n$.
- Therefore, $a^n = a \ (mod \ n)$ for $a \leq 2\sqrt{r} \log n$.
- Since $r > 4 \log^2 n$, above equation holds for at least $4 \log^2 n$ a's.

Removing Prime Powers

Lemma (Hendrik Lenstra, Jr., 1984)

If $a^n = a \pmod{n}$ for every $a \leq 4\log^2 n$ then n is square-free.

The lemma shows that n cannot be a prime power. $\square$

Back

Removing Small Divisors

Proof.

- Suppose that $(Y + a)^n = Y^n + a \ (mod\ n, Y^{2r} - Y^r)$ for $a \leq 2\sqrt{r} \log n$.
- By previous theorem, this means that n is either prime or has a divisor $< r$.
- In addition, we have $(Y + 1)^n = Y^n + 1 \ (mod\ n, Y^r) = 1 \ (mod\ n, Y^r)$.
- Expanding left side, we get: $\sum_{j=1}^{r-1} \binom{n}{j} Y^j = 0 \ (mod\ n)$.
- Therefore, $\binom{n}{j} = 0 \ (mod\ n)$ for $1 \leq j < r$.
- Let p be the smallest divisor of n and assume that $p < r$.
- Then, $\binom{n}{p} = \frac{n}{p} = 0 \ (mod\ n)$. Contradiction. $\square$

Removing Small Divisors

Proof.

- Suppose that $(Y+a)^n = Y^n + a \ (mod\ n, Y^{2r} - Y^r)$ for $a \leq 2\sqrt{r}\log n$.
- By previous theorem, this means that n is either prime or has a divisor $< r$.
- In addition, we have $(Y+1)^n = Y^n + 1 \ (mod\ n, Y^r) = 1 \ (mod\ n, Y^r)$.
- Expanding left side, we get: $\sum_{j=1}^{r-1} \binom{n}{j} Y^j = 0 \ (mod\ n)$.
- Therefore, $\binom{n}{j} = 0 \ (mod\ n)$ for $1 \leq j < r$.
- Let p be the smallest divisor of n and assume that $p < r$.
- Then, $\binom{n}{p} = \frac{n}{p} = 0 \ (mod\ n)$. Contradiction. □

Removing Small Divisors

Proof.

- Suppose that $(Y+a)^n = Y^n + a \ (mod\ n, Y^{2r} - Y^r)$ for $a \leq 2\sqrt{r}\log n$.
- By previous theorem, this means that n is either prime or has a divisor $< r$.
- In addition, we have $(Y+1)^n = Y^n + 1 \ (mod\ n, Y^r) = 1 \ (mod\ n, Y^r)$.
- Expanding left side, we get: $\sum_{j=1}^{r-1} \binom{n}{j} Y^j = 0 \ (mod\ n)$.
- Therefore, $\binom{n}{j} = 0 \ (mod\ n)$ for $1 \leq j < r$.
- Let p be the smallest divisor of n and assume that $p < r$.
- Then, $\binom{n}{p} = \frac{n}{p} = 0 \ (mod\ n)$. Contradiction. □

Removing Small Divisors

Proof.

- Suppose that $(Y+a)^n = Y^n + a \ (mod\ n, Y^{2r} - Y^r)$ for $a \leq 2\sqrt{r}\log n$.
- By previous theorem, this means that n is either prime or has a divisor $< r$.
- In addition, we have $(Y+1)^n = Y^n + 1 \ (mod\ n, Y^r) = 1 \ (mod\ n, Y^r)$.
- Expanding left side, we get: $\sum_{j=1}^{r-1} \binom{n}{j} Y^j = 0 \ (mod\ n)$.
- Therefore, $\binom{n}{j} = 0 \ (mod\ n)$ for $1 \leq j < r$.
- Let p be the smallest divisor of n and assume that $p < r$.
- Then, $\binom{n}{p} = \frac{n}{p} = 0 \ (mod\ n)$. Contradiction. □

Removing Small Divisors

Proof.

- Suppose that $(Y+a)^n = Y^n + a \; (mod \; n, Y^{2r} - Y^r)$ for $a \leq 2\sqrt{r} \log n$.
- By previous theorem, this means that n is either prime or has a divisor $< r$.
- In addition, we have $(Y+1)^n = Y^n + 1 \; (mod \; n, Y^r) = 1 \; (mod \; n, Y^r)$.
- Expanding left side, we get: $\sum_{j=1}^{r-1} \binom{n}{j} Y^j = 0 \; (mod \; n)$.
- Therefore, $\binom{n}{j} = 0 \; (mod \; n)$ for $1 \leq j < r$.
- Let p be the smallest divisor of n and assume that $p < r$.
- Then, $\binom{n}{p} = \frac{n}{p} = 0 \; (mod \; n)$. Contradiction. $\square$