Isomorphism Problems of Graphs, F-algebras and Cubic Forms

Manindra Agrawal, Nitin Saxena

IIT Kanpur

IRISS, Jan 2006
The **Graph Isomorphism** problem is to *efficiently* check whether two given graphs are isomorphic.

- This is a fundamental problem in computer science and not even a subexponential time algorithm is known yet.
- In this talk we will display connections of Graph Isomorphism to the isomorphism problems of basic algebraic structures like $\mathbb{F}$-algebras and cubic forms.
- The hope is that a better understanding of these algebraic structures might shed light on the graph isomorphism problem.
Motivation

- The **Graph Isomorphism** problem is to *efficiently* check whether two given graphs are isomorphic.
- This is a fundamental problem in computer science and not even a subexponential time algorithm is known yet.
- In this talk we will display connections of Graph Isomorphism to the isomorphism problems of basic algebraic structures like $\mathbb{F}$-algebras and cubic forms.
- The hope is that a better understanding of these algebraic structures might shed light on the graph isomorphism problem.
Motivation

The Graph Isomorphism problem is to *efficiently* check whether two given graphs are isomorphic.

This is a fundamental problem in computer science and not even a subexponential time algorithm is known yet.

In this talk we will display connections of Graph Isomorphism to the isomorphism problems of basic algebraic structures like $F$-algebras and cubic forms.

The hope is that a better understanding of these algebraic structures might shed light on the graph isomorphism problem.
**Motivation**

- The **Graph Isomorphism** problem is to *efficiently* check whether two given graphs are isomorphic.
- This is a fundamental problem in computer science and not even a subexponential time algorithm is known yet.
- In this talk we will display connections of Graph Isomorphism to the isomorphism problems of basic algebraic structures like $F$-algebras and cubic forms.
- The hope is that a better understanding of these algebraic structures might shed light on the graph isomorphism problem.
GI is in NP

- Given two graphs $G_1, G_2$ and a map $\pi$, it is easy to check whether $\pi$ is an isomorphism from $G_1 \to G_2$.
- Thus, GI can be verified in polynomial time or GI $\in$ NP.
- Is graph non-isomorphism, i.e. $\overline{\text{GI}}$, in NP too?
- Whether $\overline{\text{GI}} \in \text{NP}$ is not known but it can be shown that $\overline{\text{GI}}$ is verifiable in randomized polynomial time.
Given two graphs $G_1, G_2$ and a map $\pi$, it is easy to check whether $\pi$ is an isomorphism from $G_1 \rightarrow G_2$.

Thus, GI can be verified in polynomial time or $\text{GI} \in \text{NP}$.

Is graph non-isomorphism, i.e. $\overline{\text{GI}}$, in NP too?

Whether $\overline{\text{GI}} \in \text{NP}$ is not known but it can be shown that $\overline{\text{GI}}$ is verifiable in randomized polynomial time.
GI is in NP

- Given two graphs $G_1, G_2$ and a map $\pi$, it is easy to check whether $\pi$ is an isomorphism from $G_1 \rightarrow G_2$.
- Thus, GI can be verified in polynomial time or $\text{GI} \in \text{NP}$.
- Is graph non-isomorphism, i.e. $\overline{\text{GI}}$, in NP too?
- Whether $\overline{\text{GI}} \in \text{NP}$ is not known but it can be shown that $\overline{\text{GI}}$ is verifiable in randomized polynomial time.
GI is in NP

• Given two graphs $G_1, G_2$ and a map $\pi$, it is easy to check whether $\pi$ is an isomorphism from $G_1 \rightarrow G_2$.
• Thus, GI can be verified in polynomial time or GI $\in$ NP.
• Is graph non-isomorphism, i.e. $\overline{GI}$, in NP too?
• Whether $\overline{GI} \in NP$ is not known but it can be shown that $\overline{GI}$ is verifiable in randomized polynomial time.
**GI is in AM**

- Suppose the verifier has two graphs $G_1, G_2$ and he wants to verify whether the graphs are non-isomorphic by querying a prover.
- The verifier randomly chooses a permutation $\pi$ on the vertex set and an $i \in \{1, 2\}$.
- The verifier sends the graph $\pi(G_i)$ to the prover and asks the prover to send back a $j \in \{1, 2\}$ and an isomorphism $\sigma : G_j \rightarrow \pi(G_i)$. The verifier accepts iff $j = i$.
- Observe that:
  
  $G_1 \not\sim G_2 \Rightarrow \Pr[\text{Verifier accepts}] = 1$
  
  $G_1 \sim G_2 \Rightarrow \Pr[\text{Verifier accepts}] \leq \frac{1}{2}$
GI IS IN AM

• Suppose the verifier has two graphs $G_1, G_2$ and he wants to verify whether the graphs are non-isomorphic by querying a prover.

• The verifier randomly chooses a permutation $\pi$ on the vertex set and an $i \in \{1, 2\}$.

• The verifier sends the graph $\pi(G_i)$ to the prover and asks the prover to send back a $j \in \{1, 2\}$ and an isomorphism $\sigma : G_j \rightarrow \pi(G_i)$. The verifier accepts iff $j = i$.

• Observe that:

\[
G_1 \not\cong G_2 \quad \Rightarrow \quad \Pr[\text{Verifier accepts}] = 1
\]

\[
G_1 \cong G_2 \quad \Rightarrow \quad \Pr[\text{Verifier accepts}] \leq \frac{1}{2}
\]
GI IS IN AM

- Suppose the verifier has two graphs $G_1, G_2$ and he wants to verify whether the graphs are non-isomorphic by querying a prover.
- The verifier randomly chooses a permutation $\pi$ on the vertex set and an $i \in \{1, 2\}$.
- The verifier sends the graph $\pi(G_i)$ to the prover and asks the prover to send back a $j \in \{1, 2\}$ and an isomorphism $\sigma : G_j \rightarrow \pi(G_i)$. The verifier accepts iff $j = i$.
- Observe that:

\[
G_1 \not\cong G_2 \Rightarrow \Pr[\text{Verifier accepts}] = 1
\]
\[
G_1 \cong G_2 \Rightarrow \Pr[\text{Verifier accepts}] \leq \frac{1}{2}
\]
Suppose the verifier has two graphs $G_1, G_2$ and he wants to verify whether the graphs are non-isomorphic by querying a prover.

The verifier randomly chooses a permutation $\pi$ on the vertex set and an $i \in \{1, 2\}$.

The verifier sends the graph $\pi(G_i)$ to the prover and asks the prover to send back a $j \in \{1, 2\}$ and an isomorphism $\sigma : G_j \rightarrow \pi(G_i)$. The verifier accepts iff $j = i$.

Observe that:

$$G_1 \not\cong G_2 \quad \Rightarrow \quad \Pr[\text{Verifier accepts}] = 1$$

$$G_1 \cong G_2 \quad \Rightarrow \quad \Pr[\text{Verifier accepts}] \leq \frac{1}{2}$$
GI “cannot be” NP-hard

- The previous two slides tell us that \( GI \in \text{NP} \cap \text{coAM} \).
- This means that GI is unlikely to be NP-hard or else polynomial hierarchy will collapse.
GI “cannot be” NP-hard

- The previous two slides tell us that $\text{GI} \in \text{NP} \cap \text{coAM}$.  
- This means that GI is unlikely to be NP-hard or else polynomial hierarchy will collapse.
Outline

Motivation

Complexity of GI

F-algebra Isomorphism
  Definitions
  The Complexity

Cubic Form Equivalence
  Definitions
  The Complexity

Conclusion
**F-algebras**

- Let \( F \) be a finite field. **F**-algebra is a set of elements with operations of addition and multiplication *suitably* defined on the elements.
- For example, \( F_p[x]/(x^2) \) is an \( F \)-algebra with elements of the form \( (a + bx) \), \( a, b \in F_p \). Addition is natural while multiplication is defined as:
  \[
  (a + bx)(c + dx) = ac + (ad + bc)x \pmod{p}.
  \]
- Let \( R \) be an \( F \)-algebra such that its elements look like:
  \[
  (\alpha_1 b_1 + \cdots + \alpha_n b_n), \quad \alpha_1, \ldots, \alpha_n \in F.
  \]
- \( b_1, \ldots, b_n \) are called **basis** elements and \( R \) is completely defined by specifying the products \( b_i \cdot b_j \).
F-algebras

- Let $\mathbb{F}$ be a finite field. $\mathbb{F}$-algebra is a set of elements with operations of addition and multiplication suitably defined on the elements.
- For example, $\mathbb{F}_p[x]/(x^2)$ is an $\mathbb{F}$-algebra with elements of the form $(a + bx)$, $a, b \in \mathbb{F}_p$. Addition is natural while multiplication is defined as:
  $$(a + bx)(c + dx) = ac + (ad + bc)x \pmod{p}.$$ 
- Let $R$ be an $\mathbb{F}$-algebra such that its elements look like:
  $$(\alpha_1 b_1 + \cdots + \alpha_n b_n), \, \alpha_1, \ldots, \alpha_n \in \mathbb{F}.$$ 
- $b_1, \ldots, b_n$ are called basis elements and $R$ is completely defined by specifying the products $b_i \cdot b_j$. 
Let $\mathbb{F}$ be a finite field. $\mathbb{F}$-algebra is a set of elements with operations of addition and multiplication *suitably* defined on the elements.

For example, $\mathbb{F}_p[x]/(x^2)$ is an $\mathbb{F}$-algebra with elements of the form $(a + bx)$, $a, b \in \mathbb{F}_p$. Addition is natural while multiplication is defined as:

$$(a + bx)(c + dx) = ac + (ad + bc)x \pmod{p}.$$ 

Let $R$ be an $\mathbb{F}$-algebra such that its elements look like:

$$(\alpha_1 b_1 + \cdots + \alpha_n b_n), \quad \alpha_1, \ldots, \alpha_n \in \mathbb{F}.$$ 

$b_1, \ldots, b_n$ are called *basis* elements and $R$ is completely defined by specifying the products $b_i \cdot b_j$. 

**F-algebras**
\textbf{F-ALGEBRAS}

- Let \( F \) be a finite field. \( F \)-algebra is a set of elements with operations of addition and multiplication \textit{suitably} defined on the elements.
- For example, \( \mathbb{F}_p[x]/(x^2) \) is an \( F \)-algebra with elements of the form \( (a + bx) \), \( a, b \in \mathbb{F}_p \). Addition is natural while multiplication is defined as:
  \[
  (a + bx)(c + dx) = ac + (ad + bc)x \pmod{p}.
  \]
- Let \( R \) be an \( F \)-algebra such that its elements look like:
  \[
  (\alpha_1 b_1 + \cdots + \alpha_n b_n), \quad \alpha_1, \ldots, \alpha_n \in F.
  \]
- \( b_1, \ldots, b_n \) are called \textit{basis} elements and \( R \) is completely defined by specifying the products \( b_i \cdot b_j \).
Let $\mathbb{F}$ be a finite field. $\mathbb{F}$-algebra is a set of elements with operations of addition and multiplication *suitably* defined on the elements.

For example, $\mathbb{F}_p[x]/(x^2)$ is an $\mathbb{F}$-algebra with elements of the form $(a + bx)$, $a, b \in \mathbb{F}_p$. Addition is natural while multiplication is defined as:

$$(a + bx)(c + dx) = ac + (ad + bc)x \pmod{p}.$$ 

Let $R$ be an $\mathbb{F}$-algebra such that its elements look like:

$$(\alpha_1 b_1 + \cdots + \alpha_n b_n), \quad \alpha_1, \ldots, \alpha_n \in \mathbb{F}.$$ 

$b_1, \ldots, b_n$ are called *basis* elements and $R$ is completely defined by specifying the products $b_i \cdot b_j$. 

**F-ALGEBRAS**
Problem Statement

- The $\mathbb{F}$-algebra Isomorphism problem is to check whether two given $\mathbb{F}$-algebras $R_1, R_2$ are isomorphic,

- For example, $\mathbb{F}_p[x]/(x^2)$ and $\mathbb{F}_p[x]/((x - 1)^2)$ are isomorphic $\mathbb{F}$-algebras.

- Of course, we want to solve this problem in time polynomial in the size of the basis representations of $R_1$ and $R_2$. 
Problem Statement

• The $\mathbb{F}$-algebra Isomorphism problem is to check whether two given $\mathbb{F}$-algebras $R_1, R_2$ are isomorphic, i.e. whether there is a bijective map from $R_1 \rightarrow R_2$ that preserves the addition and multiplication operations.

• For example, $\mathbb{F}_p[x]/(x^2)$ and $\mathbb{F}_p[x]/((x - 1)^2)$ are isomorphic $\mathbb{F}$-algebras.

• Of course, we want to solve this problem in time polynomial in the size of the basis representations of $R_1$ and $R_2$. 
The $F$-algebra Isomorphism problem is to check whether two given $F$-algebras $R_1, R_2$ are isomorphic, i.e. whether there is a bijective map from $R_1 \rightarrow R_2$ that preserves the addition and multiplication operations.

For example, $F_p[x]/(x^2)$ and $F_p[x]/((x - 1)^2)$ are isomorphic $F$-algebras.

Of course, we want to solve this problem in time polynomial in the size of the basis representations of $R_1$ and $R_2$. 
Problem Statement

- The $\mathbb{F}$-algebra Isomorphism problem is to check whether two given $\mathbb{F}$-algebras $R_1, R_2$ are isomorphic, i.e. whether there is a bijective map from $R_1 \rightarrow R_2$ that preserves the addition and multiplication operations.

- For example, $\mathbb{F}_p[x]/(x^2)$ and $\mathbb{F}_p[x]/((x - 1)^2)$ are isomorphic $\mathbb{F}$-algebras.

- Of course, we want to solve this problem in time polynomial in the size of the basis representations of $R_1$ and $R_2$. 
Outline

Motivation

Complexity of GI

F-algebra Isomorphism
- Definitions
- The Complexity

Cubic Form Equivalence
- Definitions
- The Complexity

Conclusion
Clearly, $\mathbb{F}$-algebra Isomorphism is in NP.

The proof of GI in coAM can be modified to show $\mathbb{F}$-algebra Isomorphism in coAM.

- The verifier applies random invertible linear transformation on the basis $b_1, \ldots, b_n$.

Thus, $\mathbb{F}$-algebra Isomorphism is in $\text{NP} \cap \text{coAM}$.
Clearly, $\mathbf{F}$-algebra Isomorphism is in NP.

The proof of GI in coAM can be modified to show $\mathbf{F}$-algebra Isomorphism in coAM.

- The verifier applies random invertible linear transformation on the basis $b_1, \ldots, b_n$.

Thus, $\mathbf{F}$-algebra Isomorphism is in $\text{NP} \cap \text{coAM}$.
Unlikely to be NP-hard

- Clearly, $F$-algebra Isomorphism is in NP.
- The proof of GI in coAM can be modified to show $F$-algebra Isomorphism in coAM.
  - The verifier applies random invertible linear transformation on the basis $b_1, \ldots, b_n$.
- Thus, $F$-algebra Isomorphism is in $NP \cap coAM$. 
UNLIKELY TO BE NP-HARD

- Clearly, $F$-algebra Isomorphism is in NP.
- The proof of GI in coAM can be modified to show $F$-algebra Isomorphism in coAM.
  - The verifier applies random invertible linear transformation on the basis $b_1, \ldots, b_n$.
- Thus, $F$-algebra Isomorphism is in $NP \cap coAM$. 
We will now outline how a solution to $\mathbb{F}$-algebra isomorphism can solve the graph isomorphism problem too!

Given a graph $G$ with $n$ vertices and edge set $E$ we construct the $\mathbb{F}$-algebra: $R(G) := \mathbb{F}[x_1, \ldots, x_n]/\mathcal{I}_G$

where, $\mathcal{I}_G$ is an ideal generated by the polynomials:

$$\{x_i^2\}_{i \in [n]} \cup \left\{ \sum_{(i,j) \in E} x_ix_j \right\} \cup \{x_ix_jx_k\}_{i,j,k \in [n]}$$

It can be shown that $G \cong G'$ iff $R(G) \cong R(G')$. 
Reduction from Graph Isomorphism

- We will now outline how a solution to $\mathbb{F}$-algebra isomorphism can solve the graph isomorphism problem too!
- Given a graph $G$ with $n$ vertices and edge set $E$ we construct the $\mathbb{F}$-algebra: 
  \[ R(G) := \mathbb{F}[x_1, \ldots, x_n]/\mathcal{I}_G \]
  where, $\mathcal{I}_G$ is an ideal generated by the polynomials:
  \[
  \{ x_i^2 \}_{i \in [n]} \cup \left\{ \sum_{(i,j) \in E} x_i x_j \right\} \cup \{ x_i x_j x_k \}_{i,j,k \in [n]} \]
- It can be shown that $G \cong G'$ iff $R(G) \cong R(G')$. 
Reduction from Graph Isomorphism

- We will now outline how a solution to \( \mathbb{F} \)-algebra isomorphism can solve the graph isomorphism problem too!

- Given a graph \( G \) with \( n \) vertices and edge set \( E \) we construct the \( \mathbb{F} \)-algebra: \( R(G) := \mathbb{F}[x_1, \ldots, x_n]/\mathcal{I}_G \)

where, \( \mathcal{I}_G \) is an ideal generated by the polynomials:

\[
\{ x_i^2 \}_{i \in [n]} \cup \left\{ \sum_{(i,j) \in E} x_i x_j \right\} \cup \{ x_i x_j x_k \}_{i,j,k \in [n]}
\]

- It can be shown that \( G \cong G' \) iff \( R(G) \cong R(G') \).
We will now outline how a solution to $F$-algebra isomorphism can solve the graph isomorphism problem too!

Given a graph $G$ with $n$ vertices and edge set $E$ we construct the $F$-algebra: $R(G) := F[x_1, \ldots, x_n]/I_G$ where, $I_G$ is an ideal generated by the polynomials:

$$\{x_i^2\}_{i \in [n]} \cup \left\{ \sum_{(i,j) \in E} x_i x_j \right\} \cup \left\{ x_i x_j x_k \right\}_{i,j,k \in [n]}$$

It can be shown that $G \cong G'$ iff $R(G) \cong R(G')$. 
Outline

Motivation

Complexity of GI

F-algebra Isomorphism
- Definitions
- The Complexity

Cubic Form Equivalence
- Definitions
- The Complexity

Conclusion
Cubic Forms

- Cubic Forms are degree 3 homogeneous polynomials over a field $\mathbb{F}$.
- Given two cubic forms $f(x_1, \ldots, x_n)$, $g(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$, we say that $f$ is equivalent to $g$ if there is an invertible linear transformation $\tau$ such that:
  \[ f(\tau(x_1), \ldots, \tau(x_n)) = g(x_1, \ldots, x_n). \]
- For example, $x_1^3 + x_2^2x_3$ is equivalent to $x_2^3 - (x_1 + x_2)^2x_3$.
- Cubic Form Equivalence is the problem of checking whether two given cubic forms are equivalent in time polynomial in the size of the cubic forms.
Cubic Forms

- Cubic Forms are degree 3 homogeneous polynomials over a field $\mathbb{F}$. We assume that $\mathbb{F}$ is a finite field.

- Given two cubic forms $f(x_1, \ldots, x_n), g(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$, we say that $f$ is equivalent to $g$ if there is an invertible linear transformation $\tau$ such that:

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

- For example, $x_1^3 + x_2^2 x_3$ is equivalent to $x_2^3 - (x_1 + x_2)^2 x_3$.

- Cubic Form Equivalence is the problem of checking whether two given cubic forms are equivalent in time polynomial in the size of the cubic forms.
Cubic Forms

- Cubic Forms are degree 3 homogeneous polynomials over a field $\mathbb{F}$. We assume that $\mathbb{F}$ is a finite field.
- Given two cubic forms $f(x_1, \ldots, x_n), g(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$, we say that $f$ is equivalent to $g$ if there is an invertible linear transformation $\tau$ such that:
  $$f(\tau(x_1), \ldots, \tau(x_n)) = g(x_1, \ldots, x_n).$$
- For example, $x_1^3 + x_2^2 x_3$ is equivalent to $x_2^3 - (x_1 + x_2)^2 x_3$.
- Cubic Form Equivalence is the problem of checking whether two given cubic forms are equivalent in time polynomial in the size of the cubic forms.
Cubic Forms

- Cubic Forms are degree 3 homogeneous polynomials over a field $\mathbb{F}$. We assume that $\mathbb{F}$ is a finite field.
- Given two cubic forms $f(x_1, \ldots, x_n), g(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$, we say that $f$ is equivalent to $g$ if there is an invertible linear transformation $\tau$ such that:
  $$f(\tau(x_1), \ldots, \tau(x_n)) = g(x_1, \ldots, x_n).$$
- For example, $x_1^3 + x_2^2x_3$ is equivalent to $x_2^3 - (x_1 + x_2)^2x_3$.
- Cubic Form Equivalence is the problem of checking whether two given cubic forms are equivalent in time polynomial in the size of the cubic forms.
Cubic Forms

- Cubic Forms are degree 3 homogeneous polynomials over a field $\mathbb{F}$. We assume that $\mathbb{F}$ is a finite field.

- Given two cubic forms $f(x_1, \ldots, x_n), g(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$, we say that $f$ is equivalent to $g$ if there is an invertible linear transformation $\tau$ such that:

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

- For example, $x_1^3 + x_2^2x_3$ is equivalent to $x_2^3 - (x_1 + x_2)^2x_3$.

- Cubic Form Equivalence is the problem of checking whether two given cubic forms are equivalent in time polynomial in the size of the cubic forms.
Outline

Motivation

Complexity of GI

F-algebra Isomorphism

Definitions

The Complexity

Cubic Form Equivalence

Definitions

The Complexity

Conclusion
Unlikely to be NP hard

- Clearly, Cubic Form Equivalence is in NP.
- The proof of GI in coAM can be modified to show Cubic Form Equivalence in coAM.
  - The verifier applies random invertible linear transformation on the variables $x_1, \ldots, x_n$.
- Thus, Cubic Form Equivalence is in $\mathsf{NP} \cap \mathsf{coAM}$. 
Unlikely to be NP hard

- Clearly, Cubic Form Equivalence is in NP.
- The proof of GI in coAM can be modified to show Cubic Form Equivalence in coAM.
  - The verifier applies random invertible linear transformation on the variables $x_1, \ldots, x_n$.
- Thus, Cubic Form Equivalence is in $\text{NP} \cap \text{coAM}$. 
Unlikely to be NP hard

- Clearly, Cubic Form Equivalence is in NP.
- The proof of GI in coAM can be modified to show Cubic Form Equivalence in coAM.
  - The verifier applies random invertible linear transformation on the variables $x_1, \ldots, x_n$.
- Thus, Cubic Form Equivalence is in $\text{NP} \cap \text{coAM}$. 
Unlikely to be NP hard

- Clearly, Cubic Form Equivalence is in NP.
- The proof of GI in coAM can be modified to show Cubic Form Equivalence in coAM.
  - The verifier applies random invertible linear transformation on the variables $x_1, \ldots, x_n$.
- Thus, Cubic Form Equivalence is in $NP \cap coAM$. 
**Reduction from $F$-algebra Isomorphism**

- Interestingly, $F$-algebra isomorphism reduces to cubic form equivalence.
- Let $R$ be an $F$-algebra given by its basis elements $b_1, \ldots, b_n$ and the multiplication defined as: $b_i \cdot b_j = \sum_{k=1}^{n} a_{i,j,k} b_k$ where for all $i, j, k \in [n], \ a_{i,j,k} \in F$.
- From $R$ we construct a cubic form $f_R$ as:

$$f_R(b, z, y) := \sum_{1 \leq i \leq j \leq n} z_{i,j} \left( b_i \cdot b_j - y \cdot \sum_{k=1}^{n} a_{i,j,k} b_k \right)$$

- It can be shown that for two given $F$-algebras $R$ and $R'$ we have: $R \cong R'$ iff $f_R \sim f_{R'}$. 


Reduction from $\mathbb{F}$-algebra Isomorphism

- Interestingly, $\mathbb{F}$-algebra isomorphism reduces to cubic form equivalence.

- Let $R$ be an $\mathbb{F}$-algebra given by its basis elements $b_1, \ldots, b_n$ and the multiplication defined as: $b_i \cdot b_j = \sum_{k=1}^{n} a_{i,j,k} b_k$ where for all $i, j, k \in [n]$, $a_{i,j,k} \in \mathbb{F}$.

- From $R$ we construct a cubic form $f_R$ as:

$$f_R(b, z, y) := \sum_{1 \leq i \leq j \leq n} z_{i,j} \left( b_i \cdot b_j - y \cdot \sum_{k=1}^{n} a_{i,j,k} b_k \right)$$

- It can be shown that for two given $\mathbb{F}$-algebras $R$ and $R'$ we have: $R \cong R'$ iff $f_R \sim f_{R'}$. 
Reduction from $\mathbb{F}$-algebra Isomorphism

- Interestingly, $\mathbb{F}$-algebra isomorphism reduces to cubic form equivalence.

- Let $R$ be an $\mathbb{F}$-algebra given by its basis elements $b_1, \ldots, b_n$ and the multiplication defined as: $b_i \cdot b_j = \sum_{k=1}^{n} a_{i,j,k} b_k$ where for all $i, j, k \in [n]$, $a_{i,j,k} \in \mathbb{F}$.

- From $R$ we construct a cubic form $f_R$ as:

$$f_R(b, z, y) := \sum_{1 \leq i \leq j \leq n} z_{i,j} \left( b_i \cdot b_j - y \cdot \sum_{k=1}^{n} a_{i,j,k} b_k \right)$$

- It can be shown that for two given $\mathbb{F}$-algebras $R$ and $R'$ we have: $R \simeq R'$ iff $f_R \sim f_{R'}$. 
**Reduction from \( \mathbb{F} \)-algebra Isomorphism**

- Interestingly, \( \mathbb{F} \)-algebra isomorphism reduces to cubic form equivalence.

- Let \( R \) be an \( \mathbb{F} \)-algebra given by its basis elements \( b_1, \ldots, b_n \) and the multiplication defined as: \( b_i \cdot b_j = \sum_{k=1}^{n} a_{i,j,k} b_k \) where for all \( i, j, k \in [n], \ a_{i,j,k} \in \mathbb{F} \).

- From \( R \) we construct a cubic form \( f_R \) as:

\[
f_R(\overline{b}, \overline{z}, y) := \sum_{1 \leq i \leq j \leq n} z_{i,j} \left( b_i \cdot b_j - y \cdot \sum_{k=1}^{n} a_{i,j,k} b_k \right)
\]

- It can be shown that for two given \( \mathbb{F} \)-algebras \( R \) and \( R' \) we have: \( R \cong R' \) iff \( f_R \sim f_{R'} \).
The Results

- The isomorphism problems of graphs, $F$-algebras and $F$-cubic forms are of intermediate complexity (for finite $F$).
- These problems satisfy the following relation:

  \[
  \text{Graph Isomorphism} \leq F - \text{algebra Isomorphism} \leq \text{Cubic Form Equivalence}
  \]
The Results

- The isomorphism problems of graphs, $\mathbb{F}$-algebras and $\mathbb{F}$-cubic forms are of intermediate complexity (for finite $\mathbb{F}$).
- These problems satisfy the following relation:

\[
\text{Graph Isomorphism} \leq \mathbb{F} - \text{algebra Isomorphism} \leq \text{Cubic Form Equivalence}
\]
The Results

- The isomorphism problems of graphs, $\mathbb{F}$-algebras and $\mathbb{F}$-cubic forms are of intermediate complexity (for finite $\mathbb{F}$).
- These problems satisfy the following relation:

\[
\text{Graph Isomorphism} \leq \mathbb{F} - \text{algebra Isomorphism} \leq \text{Cubic Form Equivalence}
\]
The Results

- The isomorphism problems of graphs, $F$-algebras and $F$-cubic forms are of intermediate complexity (for finite $F$).
- These problems satisfy the following relation:

\[
\text{Graph Isomorphism} \leq \text{F} \text{— algebra Isomorphism} \leq \text{Cubic Form Equivalence}
\]
Open Problems

We find the following problems of interest:

• Is there a way to solve cubic form equivalence in subexponential time?

• Is the cubic form equivalence problem over an infinite field $F$ decidable?
Open Problems

We find the following problems of interest:

- Is there a way to solve cubic form equivalence in subexponential time?
- Is the cubic form equivalence problem over an infinite field \( F \) decidable?
Open Problems

We find the following problems of interest:

- Is there a way to solve cubic form equivalence in subexponential time?
- Is the cubic form equivalence problem over an infinite field \( \mathbb{F} \) decidable?
Thank You!

Questions?