Tensor Isomorphism
Complexity, Algorithms, and Cryptography

Youming Qiao
University of Technology Sydney (*)

Workshop on Algebraic Complexity Theory (WACT) 2025
3 April, 2025

* Currently visiting IAS Princeton supported by the Ky Fang and Yu-Fen Fan Endowment Fund

Talk outline

1. Isomorphism problems: from graphs and matrices to tensors
2. Complexity: Tensor Isomorphism as a unifying problem for some algebraic isomorphism problems
3. Algorithms: Exciting progress, but still exponential…
4. Cryptography: Group action based cryptography
5. Conclusion and open problems

Based on joint works with many collaborators, including Josh Grochow, Gábor Ivanyos, Markus Bläser, Alexander Rogovskyy, Xiaorui Sun, Kate Stange, Yinan Li, Chuanqi Zhang, Antoine Joux, Anand Narayanan…

Talk outline

1. Isomorphism problems: from graphs and matrices to tensors

Isomorphism testing in computer science

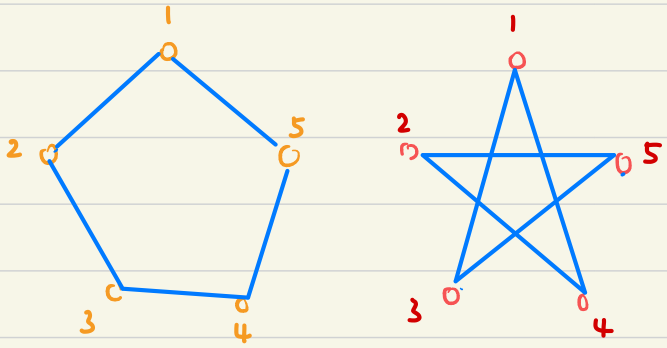
- Isomorphism problems: given two (combinatorial or algebraic) structures, whether they are essentially the same
- The most famous example is Graph Isomorphism
  - Given two graphs, decide if they are the same up to relabelling the vertices

Isomorphism testing in computer science

* Isomorphism problems: given two (combinatorial or algebraic) structures, whether they are essentially the same

* The most famous example is Graph Isomorphism

    - Given two graphs, decide if they are the same up to relabelling the vertices

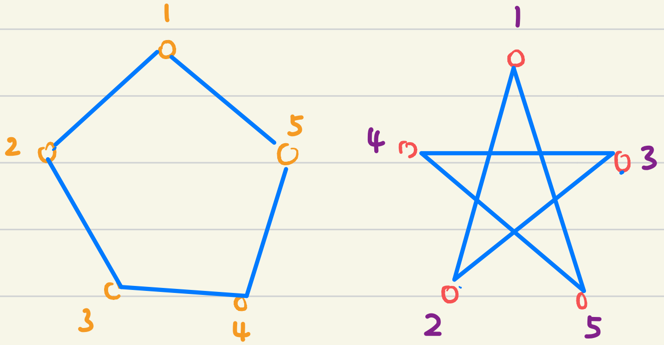


Isomorphism testing in computer science

* Isomorphism problems: given two (combinatorial or algebraic) structures, whether they are essentially the same

* The most famous example is Graph Isomorphism

- Given two graphs, decide if they are the same up to relabelling the vertices



Isomorphism testing in computer science

* Isomorphism problems: given two (combinatorial or algebraic) structures, whether they are essentially the same

* The most famous example is Graph Isomorphism

- Given two graphs, decide if they are the same up to relabelling the vertices
- One of the earliest problems considered in the framework of P and NP
- Motivated permutation group algorithms [Babai, Luks...];
  Classical examples for interactive protocols [Goldwasser—Sipser, Schöning], zero-knowledge [Goldreich—Micali—Wigderson]

Isomorphism testing in computer science

* Isomorphism problems: given two (combinatorial or algebraic) structures, whether they are essentially the same

* The most famous example is Graph Isomorphism

- Given two graphs, decide if they are the same up to relabelling the vertices

* Another classical example is Matrix Equivalence

- Given two matrices A and B, decide if $A=LBR$ for invertible matrices L and R

Isomorphism testing in computer science

* Isomorphism problems: given two (combinatorial or algebraic) structures, whether they are essentially the same

* The most famous example is Graph Isomorphism

- Given two graphs, decide if they are the same up to relabelling the vertices

* Another classical example is Matrix Equivalence

- Given two matrices A and B, decide if $A=LBR$ for invertible matrices L and R

$$A = \begin{bmatrix} -49 & 17 \\ 33 & -89 \end{bmatrix} \qquad B = \begin{bmatrix} 13 & 7 \\ -8 & 3 \end{bmatrix}$$

Isomorphism testing in computer science

* Isomorphism problems: given two (combinatorial or algebraic) structures, whether they are essentially the same

* The most famous example is Graph Isomorphism

- Given two graphs, decide if they are the same up to relabelling the vertices

* Another classical example is Matrix Equivalence

- Given two matrices A and B, decide if $A=LBR$ for invertible matrices L and R

$$A = \begin{bmatrix} -49 & 17 \\ 33 & -89 \end{bmatrix} \qquad B = \begin{bmatrix} 13 & 7 \\ -8 & 3 \end{bmatrix}$$

$$\Rightarrow \quad L = \begin{bmatrix} 1 & 2 \\ -1 & 2 \end{bmatrix} \qquad R = \begin{bmatrix} -1 & 3 \\ -4 & 2 \end{bmatrix} \qquad \text{then} \qquad A = LBR$$

Isomorphism testing in computer science

* Isomorphism problems: given two (combinatorial or algebraic) structures, whether they are essentially the same

* The most famous example is Graph Isomorphism

- Given two graphs, decide if they are the same up to relabelling the vertices

* Another classical example is Matrix Equivalence

- Given two matrices A and B, decide if $A=LBR$ for invertible matrices L and R
- A basic linear algebra fact: A and B are equivalent iff $\mathrm{rank}(A)=\mathrm{rank}(B)$
- Two related notions: matrix similarity ($R=L^{-1}$) and matrix congruence ($R=L^{t}$) are important topics in linear algebra

From matrices to tensors

* A main object of interest in this workshop is tensors, or multiway arrays

From matrices to tensors

* A main object of interest in this workshop is tensors, or multiway arrays

  - Note: a matrix is a 2-way array

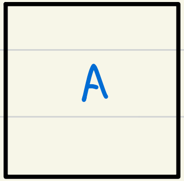
$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

From matrices to tensors

* A main object of interest in this workshop is tensors, or multiway arrays

- Note: a matrix is a 2-way array

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

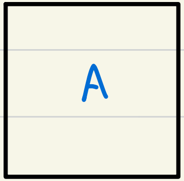


From matrices to tensors

* A main object of interest in this workshop is tensors, or multiway arrays

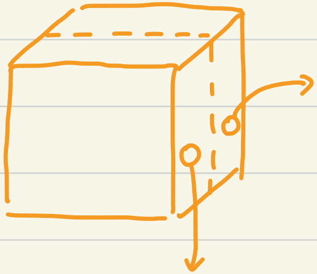
- Note: a matrix is a 2-way array

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$



- Next step: 3-way arrays

$\bar{C} =$



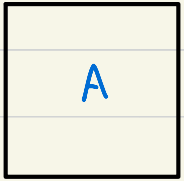
$$\begin{bmatrix} c_{211} & c_{212} \\ c_{221} & c_{222} \end{bmatrix}$$

$$\begin{bmatrix} c_{111} & c_{112} \\ c_{121} & c_{122} \end{bmatrix}$$

From matrices to tensors

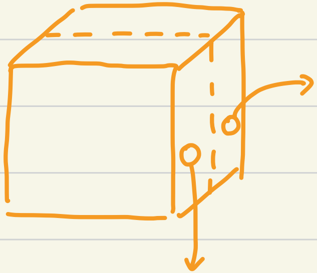
* A main object of interest in this workshop is tensors, or multiway arrays
  - Note: a matrix is a 2-way array

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$



  - Next step: 3-way arrays

$\bar{C} =$



$$\begin{bmatrix} C_{211} & C_{212} \\ C_{221} & C_{222} \end{bmatrix}$$

$$\begin{bmatrix} C_{111} & C_{112} \\ C_{121} & C_{122} \end{bmatrix}$$

$\bar{C} = (C_1, C_2)$: a matrix tuple

From matrix equivalence to tensor equivalence

* A main object of interest in this workshop is tensors, or multiway arrays

    - Note: a matrix is a 2-way array. Matrix equivalence is defined as

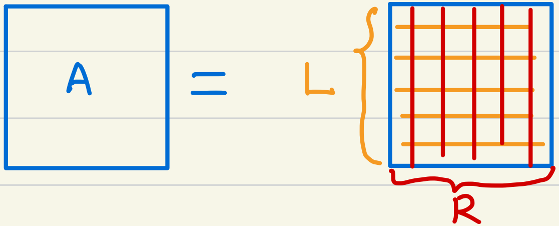
$\exists$ L, R
Invertible matrices

$$\boxed{A} = L \boxed{B} R$$

From matrix equivalence to tensor isomorphism

* A main object of interest in this workshop is tensors, or multiway arrays

  - Note: a matrix is a 2-way array. Matrix equivalence is defined as



L: row operations

R: column operations

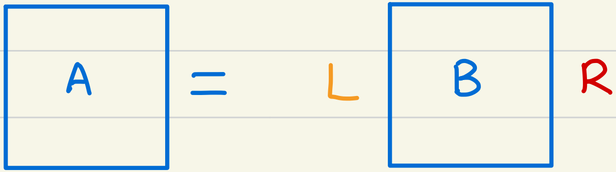
From matrix equivalence to tensor isomorphism

* A main object of interest in this workshop is tensors, or multiway arrays

- Note: a matrix is a 2-way array. Matrix equivalence is defined as

∃ L, R

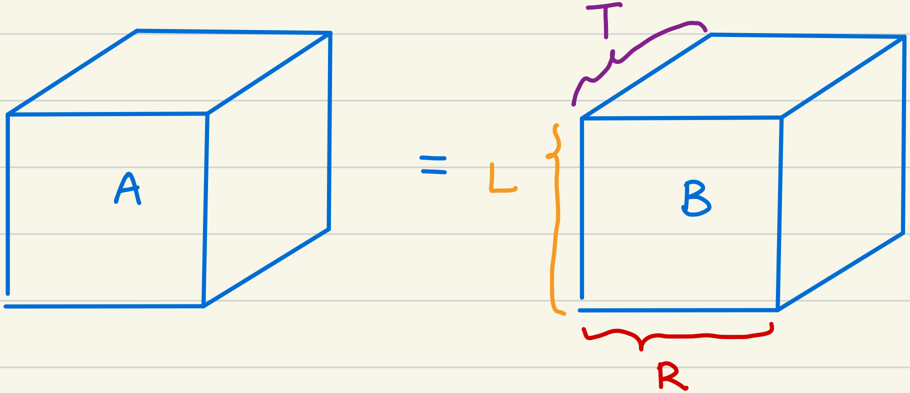
Invertible matrices



- Next step: 3-way arrays. Tensor isomorphism is defined as

∃ L, R, T

Invertible matrices



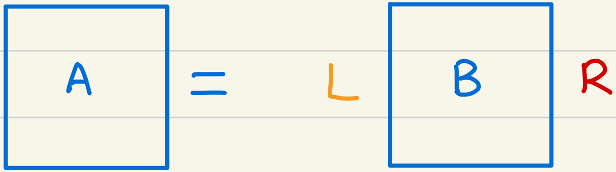
From matrix equivalence to tensor isomorphism

* A main object of interest in this workshop is tensors, or multiway arrays

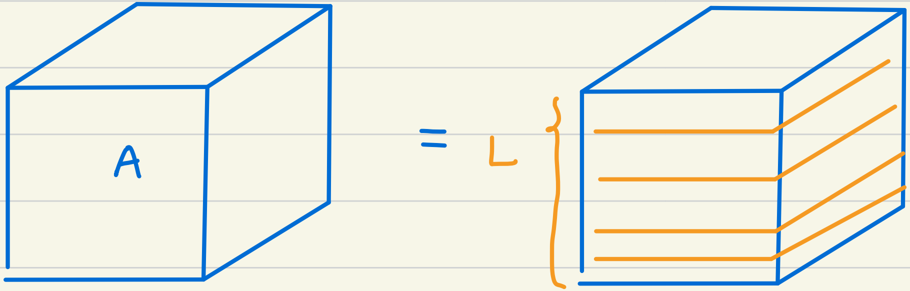
- Note: a matrix is a 2-way array. Matrix equivalence is defined as

$\exists$ L, R

Invertible matrices



- Next step: 3-way arrays. Tensor isomorphism is defined as



From matrix equivalence to tensor isomorphism

* A main object of interest in this workshop is tensors, or multiway arrays

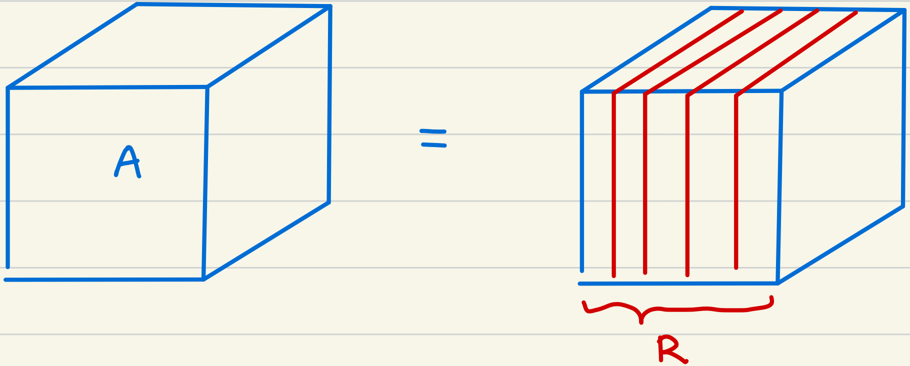
- Note: a matrix is a 2-way array. Matrix equivalence is defined as

$\exists\, L, R$

Invertible matrices

$$\boxed{A} = L\,\boxed{B}\,R$$

- Next step: 3-way arrays. Tensor isomorphism is defined as



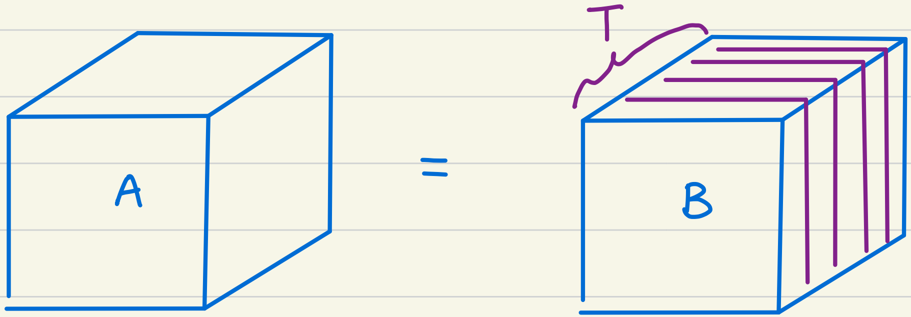
From matrix equivalence to tensor isomorphism

* A main object of interest in this workshop is tensors, or multiway arrays

- Note: a matrix is a 2-way array. Matrix equivalence is defined as

$\exists\, L, R$
Invertible matrices
$$\boxed{A} = L \boxed{B} R$$

- Next step: 3-way arrays. Tensor isomorphism is defined as



Tensor isomorphism problem

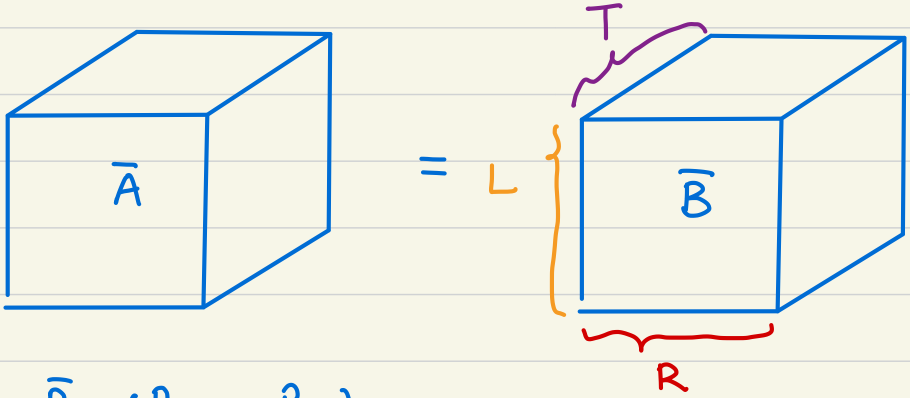
Definition. Let $\bar{A} = (A_1, \cdots, A_n)$, $\bar{B} = (B_1, \cdots, B_n)$, A_i, B_j : $n \times n$ matrices over $\mathbb{F}$.

Decide if $\exists$ $n \times n$ invertible matrices L, R, $T = (t_{ij})$, s.t.

$$\forall i \in [n], \quad A_i = \sum_{j=1}^{n} t_{ij} L B_j R$$

$\exists$ L, R, T

Invertible matrices



$\bar{A} = (A_1, \cdots, A_n)$, $\bar{B} = (B_1, \cdots, B_n)$

Some basic facts and relations

* Tensor Iso appears in coding theory (matrix codes) and quantum info (SLOCC equivalence between quantum states)

Some basic facts and relations

* Tensor Iso appears in coding theory (matrix codes) and quantum info (SLOCC equivalence between quantum states)

* The complexity of Tensor Iso depends on the underlying field

- Finite field: in NP $\cap$ coAM
- Complex number field: AM assuming Generalised Riemann Hypothesis [Koiran]

  Question: TI over C in AM $\cap$ coAM?

Some basic facts and relations

* Tensor Iso appears in coding theory (matrix codes) and quantum info (SLOCC equivalence between quantum states)

* The complexity of Tensor Iso depends on the underlying field

- Finite field: in NP $\cap$ coAM
- Complex number field: AM assuming GRH [Koiran] Q: TI over C in AM $\cap$ coAM?

* The following problems are shown to be poly-time reducible to Tensor Iso:

- Graph Iso and Code Equivalence: whether two linear codes are the same up to permuting the coordinates. Studied in coding theory since 1980s

Graph Iso $\leq_p$ Code Eq $\leq_p$ Tensor Iso

[Petrank—Roth] [Grochow—Q]

Talk outline

1. Isomorphism problems: from graphs and matrices to tensors

2. Complexity: Tensor Isomorphism as a unifying problem for algebraic isomorphism problems

Tensor Isomorphism as a new complexity class

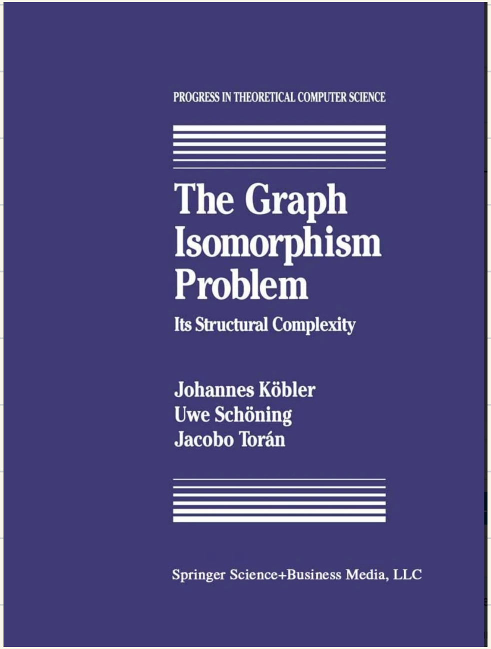
Definition. [Grochow-Q.] The Tensor Isomorphism (TI) complexity class consists of problems that are poly-time reducible to the tensor isomorphism problem.

Tensor Isomorphism as a new complexity class

Definition. [Grochow-Q.] The Tensor Isomorphism (TI) complexity class consists of problems that are poly-time reducible to the tensor isomorphism problem.

* This is in analogy with the Graph Isomorphism (GI) complexity class

- Consisting of problems poly-time reducible to Graph Iso



PROGRESS IN THEORETICAL COMPUTER SCIENCE

The Graph Isomorphism Problem

Its Structural Complexity

Johannes Köbler
Uwe Schöning
Jacobo Torán

Springer Science+Business Media, LLC

Tensor Isomorphism as a new complexity class

Definition. [Grochow-Q.] The Tensor Isomorphism (TI) complexity class consists of problems that are poly-time reducible to the tensor isomorphism problem.

* This is in analogy with the Graph Isomorphism (GI) complexity class

* So far a series of five papers: Tensor Isomorphism I to V, from 2021 to 2025

  - III: Also with Zhili Chen, Gang Tang, Chuanqi Zhang
  - V: Also with Kate Stange, Xiaorui Sun

* Motivated by complexity considerations, leading to unexpected connections :)

A synopsis of TI series

1. TensorIso captures iso problems for many algebraic structures (TI1)

2. TensorIso acted by different groups leads to connections to quantum information, geometry, and number theory (TI3 and TI5)
   - TI3: from GL(n, F) to O/U/Sp
   - TI5: from GL(n, F) to GL(n, R), R a commutative ring

3. TI2 and TI4: more on the technical aspects of complexity
   - TI2: search- and counting-to-decision reductions for Tensor Iso
   - TI4: more efficient reductions

A synopsis of TI series

1. This talk: TensorIso captures iso problems for many algebraic structures (TI1)

2. TensorIso acted by different groups leads to connections to quantum information, geometry, and number theory (TI3 and TI5)
   - TI3: from GL(n, F) to O/U/Sp
   - TI5: from GL(n, F) to GL(n, R), R a commutative ring

(There is a recorded talk at IAS on these aspects on YouTube)

3. TI2 and TI4: more on the technical aspects of complexity
   - TI2: search- and counting-to-decision reductions for Tensor Iso
   - TI4: more efficient reductions

Some algebraic isomorphism problems: Group Isomorphism

* Finite group isomorphism: Given two finite groups, decide if they are isomorphic

- Studied in TCS and computational group theory since 1970s
- For two groups of order N, a natural $N^{\log(N)+O(1)}$-time algorithm [Tarjan]
- Verbose version: Cayley tables are given
- Succinct version: generators of matrix groups over finite fields

Some algebraic isomorphism problems: Group Isomorphism

* Finite group isomorphism: Given two finite groups, decide if they are isomorphic
    - Studied in TCS and computational group theory since 1970s
    - For two groups of order N, a natural $N^{\log(N)+O(1)}$-time algorithm [Tarjan]
    - Verbose version: Cayley tables are given
    - Succinct version: generators of matrix groups over finite fields

* Polynomial-time algorithms are known for some groups
    - Groups without abelian normal subgroups [Babai—Codenotti—Grochow—Q], Groups with abelian Sylow towers [Babai—Q], Quotients of generalised Heisenberg groups [Lewis—Wilson]

Some algebraic isomorphism problems: Group Isomorphism

- Finite group isomorphism: Given two finite groups, decide if they are isomorphic
  - Studied in TCS and computational group theory since 1970s
  - For two groups of order N, a natural $N^{\log(N)+O(1)}$-time algorithm [Tarjan]
  - Verbose version: Cayley tables are given
  - Succinct version: generators of matrix groups over finite fields

- Polynomial-time algorithms are known for some groups
  - Groups without abelian normal subgroups [Babai—Codenotti—Grochow—Q], Groups with abelian Sylow towers [Babai—Q], Quotients of generalised Heisenberg groups [Lewis—Wilson]

- One group class that resisted decades of efforts:

  p-groups of nilpotency class 2 and exponent p

  - A group G, $|G| = p^\ell$, $Z(G) \supseteq [G,G]$, $\forall g \in G$, $g^p = \mathrm{id}$

Bilinear maps underlying groups

* Group Isomorphism: for p-groups of class 2 and exponent p (odd p), by taking the commutator map, we get:

* Skew-symmetric bilinear map isomorphism: finite-dim vector spaces U, V over GF(p)

Input: Skew-sym bilinear maps f, g: $U \times U \to V$

Output: "True" if $\exists$ A in GL(U), B in GL(V), s.t. $\forall u, u'$ in U, $f(A(u), A(u'))=B(g(u, u'))$

"False" otherwise

Bilinear maps underlying groups

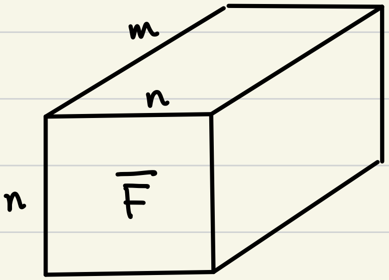
* Group Isomorphism: for p-groups of class 2 and exponent p (odd p), by taking the commutator map, we get:

* Skew-symmetric bilinear map isomorphism: finite-dim vector spaces U, V over GF(p)

Input: Skew-sym bilinear maps f, g: $U \times U \to V$

Output: "True" if $\exists$ A in $GL(U)$, B in $GL(V)$, s.t. $\forall u, u'$ in U, $f(A(u), A(u')) = B(g(u, u'))$
"False" otherwise

* Suppose $U \cong \mathbb{F}_p^n$, $V \cong \mathbb{F}_p^m$.
Then $f: U \times U \to V$ is stored in algorithms as a 3-way array $\bar{F}$
$\bar{F}(i, j, k) = f(e_i, e_j)_k$



Bilinear map isometry

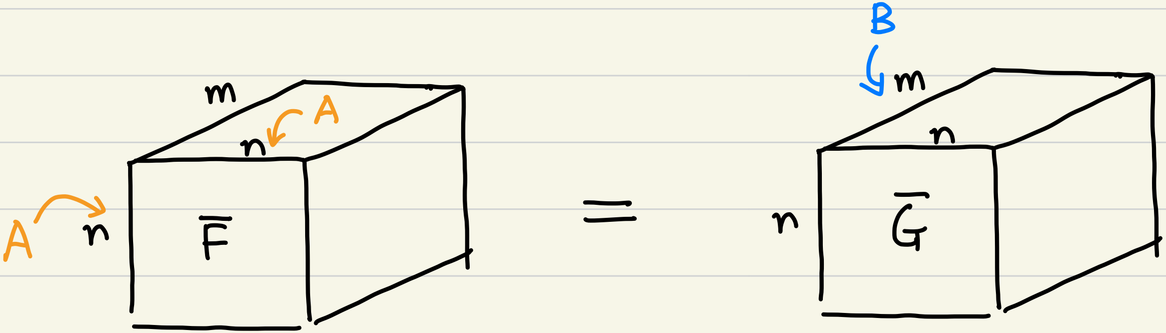
* Skew-symmetric bilinear map isomorphism: finite-dim vector spaces U, V over GF(p)

Input: Skew-sym bilinear maps f, g: U×U → V

Output: "True" if ∃ A in GL(U), B in GL(V), s.t. ∀u, u′ in U, f(A(u), A(u′))=B(g(u, u′))

"False" otherwise

* Testing if f and g are isomorphic as bimaps translates to find $A \in GL(n,p)$, $B \in GL(m,p)$



Algebra isomorphism

* Algebra isomorphism: finite-dim vector space U over a field F

Input: Bilinear maps $f, g: U \times U \rightarrow U$

Output: "True" if $\exists$ A in $GL(U)$ s.t. $\forall u, u'$ in U, $f(A(u), A(u'))=A(g(u, u'))$

"False" otherwise

Algebra isomorphism

* Algebra isomorphism: finite-dim vector space U over a field F

Input: Bilinear maps $f, g: U \times U \to U$

Output: "True" if $\exists$ A in $GL(U)$ s.t. $\forall u, u'$ in U, $f(A(u), A(u'))=A(g(u, u'))$

"False" otherwise

* Imposing conditions (alternating, associativity, Jacobi) give associative or Lie algebras

* Studied in theoretical computer science and computer algebra [Agrawal–Saxena, Saxena–Kayal, Grochow, Brooksbank–Wilson]

Algebra isomorphism

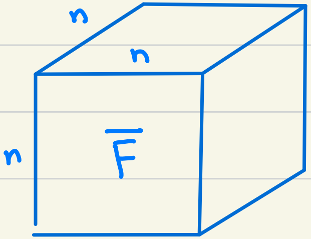
* Algebra isomorphism: finite-dim vector space U over a field F

Input: Bilinear maps f, g: U×U → U

Output: "True" if ∃ A in GL(U) s.t. ∀u, u′ in U, f(A(u), A(u′))=A(g(u, u′))

"False" otherwise

* Suppose $V \cong \mathbb{F}^n$. Represent f by its structure constants



$$\overline{F}(i,j,k) = f(e_i, e_j)_k$$

Algebra isomorphism

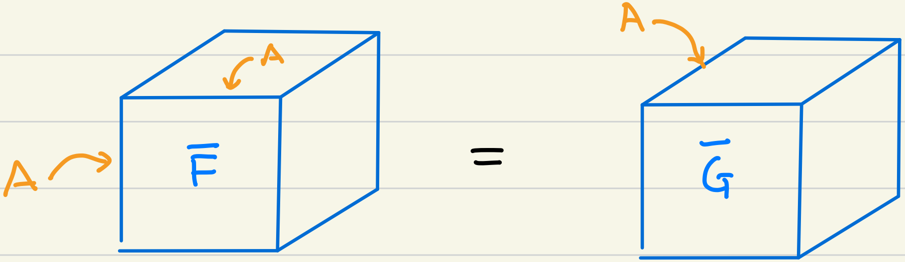
* Algebra isomorphism: finite-dim vector space U over a field F

Input: Bilinear maps f, g: $U \times U \to U$

Output: "True" if $\exists$ A in GL(U) s.t. $\forall u, u'$ in U, $f(A(u), A(u')) = A(g(u, u'))$

"False" otherwise

* Testing if f and g are isomorphic as algebras translates to find $A \in GL(n, \mathbb{F})$ s.t.



Cubic form equivalence

* Cubic form equivalence:

Input: cubic forms $f, g \in \mathbb{F}[x_1, x_2, \cdots, x_n]$

Output: True if $\exists\ A = (a_{ij}) \in GL(n, \mathbb{F})$, s.t. $f(x_1, \cdots, x_n) = g\left(\sum_{i=1}^{n} a_{1i} x_i, \cdots, \sum_{i=1}^{n} a_{ni} x_i\right)$

False otherwise

Cubic form equivalence

* Cubic form equivalence:

Input: cubic forms $f, g \in \mathbb{F}[x_1, x_2, \cdots, x_n]$

Output: True if $\exists\ A = (a_{ij}) \in GL(n, \mathbb{F})$, s.t. $f(x_1, \cdots, x_n) = g(\sum_{i=1}^{n} a_{1i} x_i, \cdots, \sum_{i=1}^{n} a_{ni} x_i)$

False otherwise

* Studied in multivariate cryptography [Patarin, Bouillaguet—Fouque—Véber, Beullens] and complexity theory [Agrawal—Saxena]

- Agrawal—Saxena: poly-time equivalence between cubic form iso and algebra iso

Cubic form equivalence

* Cubic form equivalence:

Input: cubic forms $f, g \in \mathbb{F}[x_1, x_2, \cdots, x_n]$

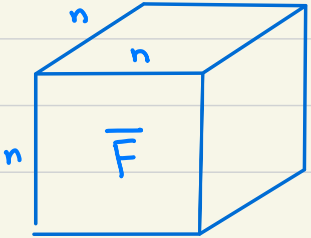
Output: True if $\exists\, A = (a_{ij}) \in GL(n, \mathbb{F})$, s.t. $f(x_1, \cdots, x_n) = g\left(\sum_{i=1}^{n} a_{1i} x_i, \cdots, \sum_{i=1}^{n} a_{ni} x_i\right)$

False otherwise

* Suppose $\operatorname{char}(\mathbb{F}) \neq 2$ or 3. $f : \mathbb{F}^n \to \mathbb{F}$.

Let $\hat{f}(u,v,w) = f(u+v+w) - f(u+v) - f(u+w) - f(v+w) + f(u) + f(v) + f(w)$

$\hat{f} : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ is a symmetric trilinear form



$$\bar{F}(i,j,k) = \hat{f}(e_i, e_j, e_k)$$

Cubic form equivalence

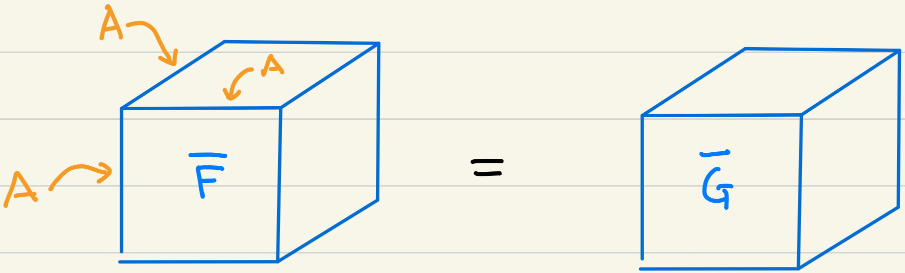
* Cubic form equivalence:

Input: cubic forms $f, g \in \mathbb{F}[x_1, x_2, \cdots, x_n]$

Output: True if $\exists\ A = (a_{ij}) \in GL(n, \mathbb{F})$, s.t. $f(x_1, \cdots, x_n) = g(\sum_{i=1}^{n} a_{1i} x_i, \cdots, \sum_{i=1}^{n} a_{ni} x_i)$

False otherwise

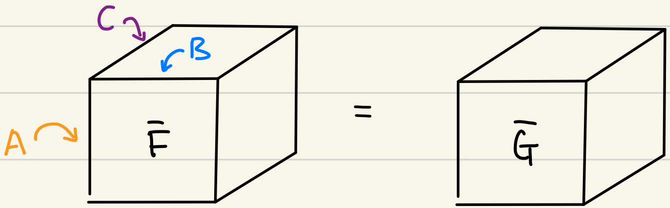
* Suppose char$(\mathbb{F}) \neq 2$ or 3. By examining symmetric trilinear forms we need to find $A \in GL(n, \mathbb{F})$ s.t



A brief recap...

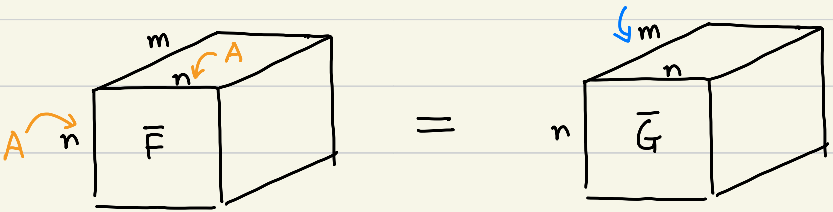
- Tensor iso:

$f,g : U \times V \times W \to \mathbb{F}$



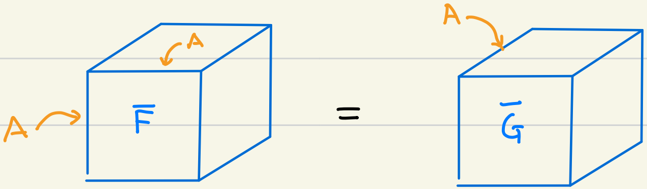
- class-2 exp-p p-group iso:

$f,g : U \times U \to V$



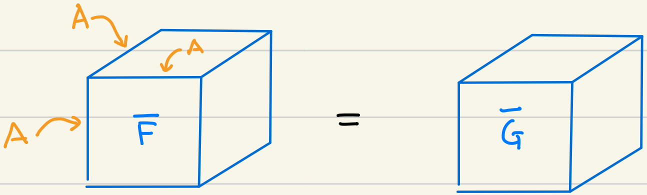
- Algebra iso:

$f,g : U \times U \to U$



- Cubic form iso:

$f,g : U \times U \times U \to \mathbb{F}$



TI-complete problems

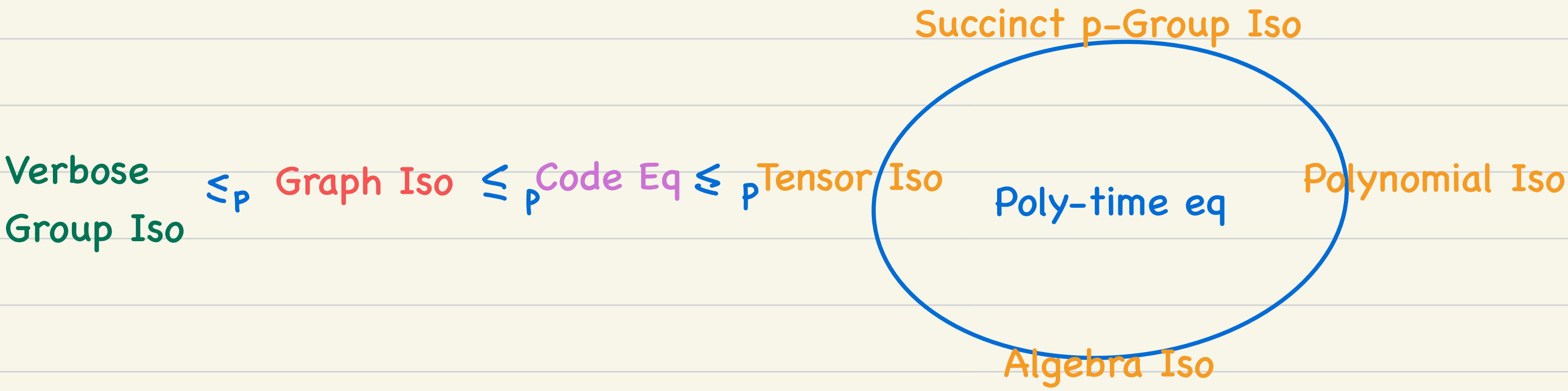
Theorem. [Futorny-Grochow-Sergeichuk, TI1] These problems are TI-complete:

* Succinct Group Isomorphism with p-groups of class 2 and exponent p
* Polynomial Isomorphism (for cubic forms)
* Algebra Isomorphism (for associative or Lie algebras)

TI-complete problems

Theorem. [Futorny-Grochow-Sergeichuk, TI1] These problems are TI-complete:

* Succinct Group Isomorphism with p-groups of class 2 and exponent p
* Polynomial Isomorphism (for cubic forms)
* Algebra Isomorphism (for associative or Lie algebras)



Verbose Group Iso $\leq_p$ Graph Iso $\leq_p$ Code Eq $\leq_p$ Tensor Iso

Succinct p-Group Iso

Polynomial Iso

Algebra Iso

Poly-time eq

TI-complete problems

Theorem. [Futorny-Grochow-Sergeichuk, TI1] These problems are TI-complete:
* Succinct Group Isomorphism with p-groups of class 2 and exponent p
* Polynomial Isomorphism (for cubic forms)
* Algebra Isomorphism (for associative or Lie algebras)

Note. Subject to appropriate underlying fields.
- p-Group Iso is over GF(p)
- Cubic form iso: field characteristic not 2 or 3

Technical version: U, V, W are vector spaces. The orbit structures of

$$U\otimes V\otimes W,\ U\otimes U\otimes V,\ U\otimes U^*\otimes V,\ U\otimes U\otimes U,\ U\otimes U\otimes U^*$$

are equivalent under the containment relation in the sense of [Gelfand—Panomerav] (even assuming natural symmetries and certain algebraic conditions)

d-Tensor Iso and 3-Tensor Iso

* Recall that matrix (2-tensor) equivalence is in P

* As we will see, 3-Tensor Iso is much harder

* How about 4-Tensor Iso, or d-Tensor Iso in general?

– $\hat{A} = (a_{ijk\ell})$ and $\hat{B} = (b_{ijk\ell})$ are the same up to invertible matrices L, R, T, S.

d-Tensor Iso and 3-Tensor Iso

* Recall that matrix (2-tensor) equivalence is in P

* As we will see, 3-Tensor Iso is much harder

* How about 4-Tensor Iso, or d-Tensor Iso in general?

  – $\hat{A} = (a_{ijk\ell})$ and $\hat{B} = (b_{ijk\ell})$ are the same up to invertible matrices L, R, T, S.

Theorem. [Grochow-Q] For d>3, d-Tensor Iso poly-time reduces to 3-Tensor Iso

* This is like: 2SAT is in P, but d-SAT reduces to 3-SAT which is NP-complete

Talk outline

1. Isomorphism problems: from graphs and matrices to tensors

2. Complexity: Tensor Isomorphism as a unifying problem

3. Algorithms: Exciting progress, but still exponential…

Algorithms for Tensor Isomorphism

* Unlike Graph Isomorphism, tensor/group/algebra/polynomial isomorphism problems seem to be much more difficult

	Graphs with n vertices	$n \times n \times n$ tensors over $\mathbb{F}_q$
Brute-force	$n!$	q^{n^2}
Worst-case		
Average-case		
Practical		

Algorithms for Tensor Isomorphism

* Unlike Graph Isomorphism, tensor/group/algebra/polynomial isomorphism problems seem to be much more difficult

	Graphs with n vertices	$n \times n \times n$ tensors over $\mathbb{F}_q$
Brute-force	$n!$	q^{n^2}
Worst-case	$n^{O(\log^2 n)}$ [Babai]	$q^{\tilde{O}(n^{1.5})}$ [Ivanyos-Mendoza-Q-Sun-Zhang]
Average-case		
Practical		

Algorithms for Tensor Isomorphism

* Unlike Graph Isomorphism, tensor/group/algebra/polynomial isomorphism problems seem to be much more difficult

	Graphs with n vertices	$n \times n \times n$ tensors over $\mathbb{F}_q$
Brute-force	$n!$	q^{n^2}
Worst-case	$n^{O(\log^2 n)}$ [Babai]	$q^{\tilde{O}(n^{1.5})}$ [Ivanyos-Mendoza-Q-Sun-Zhang]
Average-case	$O(n^2)$ [Babai-Erdős-Selkow]	$q^{O(n)}$ [Brooksbank-Li-Q-Wilson]
Practical		

Algorithms for Tensor Isomorphism

* Unlike Graph Isomorphism, tensor/group/algebra/polynomial isomorphism problems seem to be much more difficult

	Graphs with n vertices	$n \times n \times n$ tensors over $\mathbb{F}_q$
Brute-force	$n!$	q^{n^2}
Worst-case	$n^{O(\log^2 n)}$ [Babai]	$q^{\tilde{O}(n^{1.5})}$ [Ivanyos-Mendoza-Q-Sun-Zhang]
Average-case	$O(n^2)$ [Babai-Erdős-Selkow]	$q^{O(n)}$ [Brooksbank-Li-Q-Wilson]
Practical	$n \approx 10^6$ [McKay-Piperno]	$q^{\frac{1}{2}n}$ [Narayanan-Q-Tang]

↳ Not effective for $n=20$, $q=11$

On worst-case algorithms for TensorIso

Theorem. [Ivanyos-Mendoza-Q-Sun-Zhang] There exists a $q^{\tilde{O}(n^{1.5})}$-time algorithm to test isomorphism of nxnxn tensors over GF(q) for odd q.

On worst-case algorithms for TensorIso

Theorem. [Ivanyos-Mendoza-Q-Sun-Zhang] There exists a $q^{\tilde{O}(n^{1.5})}$-time algorithm to test isomorphism of nxnxn tensors over GF(q) for odd q.

* Improving from the $q^{O(n^{1.8})}$-time algorithm by Sun (the first breakthrough!)

* A wonderful combination of probabilistic methods, maximum versus non-commutative rank of matrix spaces, and classification of simple algebras with involutions!

On worst-case algorithms for TensorIso

Theorem. [Ivanyos-Mendoza-Q-Sun-Zhang] There exists a $q^{\tilde{O}(n^{1.5})}$-time algorithm to test isomorphism of nxnxn tensors over GF(q) for odd q.

* Improving from the $q^{O(n^{1.8})}$-time algorithm by Sun (the first breakthrough!)

* Using linear-length reductions in [TI4], we have:

Corollary. [Ivanyos-Mendoza-Q-Sun-Zhang] For odd p, there is an $N^{\tilde{O}(\sqrt{\log N})}$-time algorithm to test isomorphism of p-groups of class 2, exponent p, and order N.

On worst-case algorithms for TensorIso

Theorem. [Ivanyos-Mendoza-Q-Sun-Zhang] There exists a $q^{\tilde{O}(n^{1.5})}$-time algorithm to test isomorphism of nxnxn tensors over GF(q) for odd q.

* Improving from the $q^{O(n^{1.8})}$-time algorithm by Sun (the first breakthrough!)

* Using linear-length reductions in [TI4], we have:

Corollary. [Ivanyos-Mendoza-Q-Sun-Zhang] For odd p, there is an $N^{\tilde{O}(\sqrt{\log N})}$-time algorithm to test isomorphism of p-groups of class 2, exponent p, and order N.

* Again, the first breakthrough was by Sun ($N^{O((\log N)^{5/6})}$ -time)

- Breaking the decades-long barrier of $N^{\log N + O(1)}$

Talk outline

1. Isomorphism problems: from graphs and matrices to tensors
2. Complexity: Tensor Isomorphism as a unifying problem
3. Algorithms: Exciting progress, but still exponential…
4. Cryptography: Group action based cryptography

Isomorphism problems in cryptography

- Can GraphIso be used in cryptography?
  - Pondered in [Brassard—Crépeau, Brassard—Yung, ~1990]
  - Seems unlikely, not just because of Babai, but also McKay (Nauty, ~1980)

Isomorphism problems in cryptography

* Can GraphIso be used in cryptography?
  - Pondered in [Brassard—Crépeau, Brassard—Yung, ~1990]
  - Seems unlikely, not just because of Babai, but also McKay (Nauty, ~1980)

* Are there useful isomorphism problems in cryptography?
  - Yes — discrete logarithm!

Isomorphism problems in cryptography

- * Can GraphIso be used in cryptography?
  - Pondered in [Brassard—Crépeau, Brassard—Yung, ~1990]
  - Seems unlikely, not just because of Babai, but also McKay (Nauty, ~1980)

- * Are there useful isomorphism problems in cryptography?
  - Yes — discrete logarithm!

- * Group action framework: let f:GxS→S be a group action of G on S
  - Suppose group operations, actions, and sampling from G and S, are efficient
  - Orbit problem: given s and t in S, are they in the same orbit?
  - Search version: given s and t in the same orbit, compute g in G sending s to t

Isomorphism problems in cryptography

* Group action framework: let $f: G \times S \to S$ be a group action of G on S

- Suppose group operations, actions, and sampling from G and S, are efficient
- Orbit problem: given s and t in S, are they in the same orbit?
- Search version: given s and t in the same orbit, compute g in G sending s to t

* Discrete logarithm: $S = C_p \setminus \{id\}$, $G = \mathrm{Aut}(C_p) \cong \mathbb{Z}_p^\times$

Given g and h in S, compute a in G such that $g^a = h$

Isomorphism problems in cryptography

* Group action framework: let $f: G \times S \to S$ be a group action of G on S

- Suppose group operations, actions, and sampling from G and S, are efficient
- Orbit problem: given s and t in S, are they in the same orbit?
- Search version: given s and t in the same orbit, compute g in G sending s to t

* Discrete logarithm: $S = C_p \setminus \{id\}$, $G = \mathrm{Aut}(C_p) \cong \mathbb{Z}_p^\times$

Given g and h in S, compute a in G such that $g^a = h$

* GraphIso: S is the set of subsets of $\binom{[n]}{2}$, $G = \mathrm{Sym}([n])$

Given E and F in S, compute g in G sending E to F (as sets)

Isomorphism problems in cryptography

* Group action framework: let $f: G \times S \to S$ be a group action of G on S

- Suppose group operations, actions, and sampling from G and S, are efficient
- Orbit problem: given s and t in S, are they in the same orbit?
- Search version: given s and t in the same orbit, compute g in G sending s to t

* Discrete logarithm: $S = C_p \setminus \{id\}$, $G = \mathrm{Aut}(C_p) \cong \mathbb{Z}_p^\times$

Given g and h in S, compute a in G such that $g^a = h$

* GraphIso: S is the set of subsets of $\binom{[n]}{2}$, $G = \mathrm{Sym}([n])$

Given E and F in S, compute g in G sending E to F (as sets)

* TensorIso: S is the set of trilinear forms $U \times V \times W \to F$, $G = GL(U) \times GL(V) \times GL(W)$

Isomorphism problems in cryptography

- **Group action framework**: let f:G×S→S be a group action of G on S
  - Search orbit: given s and t in the same orbit, compute g in G sending s to t

Definition. [Brassard—Yung] A group action f is one-way, if for some s in S, f_s:G→S by $f_s(g):=f(g, s)$ is a one-way function.

Isomorphism problems in cryptography

* Group action framework: let $f:G\times S\to S$ be a group action of G on S

- Search orbit: given s and t in the same orbit, compute g in G sending s to t

Definition. [Brassard–Yung] A group action f is one-way, if for some s in S, $f_s:G\to S$ by $f_s(g):=f(g, s)$ is a one-way function.

* One-way group action - computational discrete logarithm.

There is a pseudorandom group action - decisional Diffie–Hellman.

Isomorphism problems in cryptography

* Group action framework: let $f:G\times S\to S$ be a group action of G on S

  - Search orbit: given s and t in the same orbit, compute g in G sending s to t

Definition. [Brassard–Yung] A group action f is one-way, if for some s in S, $f_s:G\to S$ by $f_s(g):=f(g, s)$ is a one-way function.

* One-way group action - computational discrete logarithm.

There is a pseudorandom group action - decisional Diffie–Hellman.

Definition. [Ji-Q-Song-Yun] Let G be a group acting on a set S.

- Random distribution: (s, t), where s and t are randomly sampled from S.
- Pseudorandom distribution: (s, t), where s is randomly sampled from S, and t is randomly sampled from the orbit of s

This action is pseudorandom, if no poly-time algorithms distinguish these two.

Isomorphism problems in cryptography

Definition. [Ji-Q-Song-Yun] Let G be a group acting on a set S.

- Random distribution: (s, t), where s and t are randomly sampled from S.
- Pseudorandom distribution: (s, t), where s is randomly sampled from S, and t is randomly sampled from the orbit of s

This action is pseudorandom, if no poly-time algorithms distinguish these two.

* To recover decisional Deffie—Hellman, consider

$$S = C_p\setminus\{id\} \times C_p\setminus\{id\}, \quad G = \mathrm{Aut}(C_p), \quad (g, h) \to (g^a, h^a)$$

- Random distribution: $(s, t) \in_R S \times S$, $(s,t) = ((g_1, h_1), (g_2, h_2)) = (g_1, g_1^a, g_1^b, g_1^c)$
- Pseudorandom distribution: $s = (g_1, h_1) = (g_1, g_1^a) \in_R S$

  $t = s^b = (g_1^b, g_1^{ab})$, i.e $(g_1, g_1^a, g_1^b, g_1^{ab})$

Isomorphism problems in cryptography

* Cryptographic applications of cryptographic group actions: bit commitment [Brassard—Yung], digital signature [Goldreich—Micali—Wigderson, Fiat—Shamir], quantum public-key encryption [Hhan—Morimae—Yamakawa]...

Question. Candidates for one-way or pseudorandom group actions?

Isomorphism problems in cryptography

- Cryptographic applications of cryptographic group actions: bit commitment [Brassard—Yung], digital signature [Goldreich—Micali—Wigderson, Fiat—Shamir], quantum public-key encryption [Hhan—Morimae—Yamakawa]...

Question. Candidates for one-way or pseudorandom group actions?

- DiscreteLog group action is one-way for classical but not quantum [Shor]

- GraphIso group action is not one-way for classical, but the "standard technique" from Shor's algorithm does not work [Hallgren-Morre-Rotteler-Russell-Sen]
  - Moore-Russell-Vazirani: "The strongest such evidence we have about the limits of quantum algorithms"

Isomorphism problems in cryptography

* Cryptographic applications of cryptographic group actions: bit commitment [Brassard—Yung], digital signature [Goldreich—Micali—Wigderson, Fiat—Shamir], quantum public-key encryption [Hhan—Morimae—Yamakawa]...

Question. Candidates for one-way or pseudorandom group actions?

* DiscreteLog group action is one-way for classical but not quantum [Shor]

* GraphIso group action is not one-way for classical, but the "standard technique" from Shor's algorithm does not work [Hallgren-Morre-Rotteler-Russell-Sen]

* TensorIso seems to be difficult in practice and also inherits the resistance to quantum "standard techniques"

Tensor Iso as a pseudorandom group action?

* For Tensor Iso to be pseudorandom, we need to distinguish between

  - Random distribution: Two random nxnxn tensors A and B over GF(q)
  - Pseudorandom distribution: A random nxnxn tensor A, and another B from the same orbit of A

Tensor Iso as a pseudorandom group action?

* For Tensor Iso to be pseudorandom, we need to distinguish between

- Random distribution: Two random nxnxn tensors A and B over GF(q)
- Pseudorandom distribution: A random nxnxn tensor A, and another B from the same orbit of A

* To distinguish these two distributions, it is enough to find a useful invariant

- Efficiently computable
- Distinguishing enough: two random tensors have different values

Tensor Iso as a pseudorandom group action?

* For Tensor Iso to be pseudorandom, we need to distinguish between

- Random distribution: Two random nxnxn tensors A and B over GF(q)
- Pseudorandom distribution: A random nxnxn tensor A, and another B from the same orbit of A

* To distinguish these two distributions, it is enough to find a useful invariant

- Efficiently computable
- Distinguishing enough: two random tensors have different values

* TI-complete problems are also eligible [Tang-Duong-Joux-Plantard-Q-Susilo]

Crypto as a nice motivation for math questions

* One desirable feature of digital signature schemes is to have the so-called Quantum Random Oracle Model (QROM) security

* For G acting on S, a digital signature design has QROM security, if the stabiliser group of a random s is trivial

Crypto as a nice motivation for math questions

* One desirable feature of digital signature schemes is to have the so-called Quantum Random Oracle Model (QROM) security

* For G acting on S, a digital signature design has QROM security, if the stabiliser group of a random s is trivial

Theorem. [Bläser—Li—Q—Rogovskyy] When n is large enough, a random nxnxn tensor over GF(q) has the trivial stabiliser group.

Crypto as a nice motivation for math questions

* One desirable feature of digital signature schemes is to have the so-called Quantum Random Oracle Model (QROM) security

* For G acting on S, a digital signature design has QROM security, if the stabiliser group of a random s is trivial

Theorem. [Bläser—Li—Q—Rogovskyy] When n is large enough, a random nxnxn tensor over GF(q) has the trivial stabiliser group.

* As a consequence, we could improve the estimation on the number of isomorphism classes of p-groups of class 2 and exponent p by Higman from the 1960's

Summary

1. Tensor Isomorphism problem

2. Complexity: Tensor Isomorphism as a unifying problem for some algebraic isomorphism problems

3. Algorithms: Exciting progress, but still exponential…

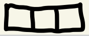
4. Cryptography: Group action based cryptography

Many questions remain...

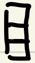
* Symmetric trilinear forms are alternating trilinear forms are irreducible reps of GL(n, C)

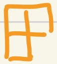
- Orbit problems for these actions are TI-complete
- How about the other one?

$U \cong \mathbb{F}^n$, $U \odot U \odot U$



$U \wedge U \wedge U$





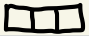
Many questions remain...

* Symmetric trilinear forms are alternating trilinear forms are irreducible reps of GL(n, C)

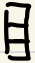
- Orbit problems for these actions are TI-complete
- How about the other one?

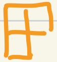
* Tensor Iso over GL(n, Q): is this decidable?

$U \cong \mathbb{F}^n$, $U \odot U \odot U$



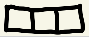
$U \wedge U \wedge U$

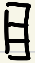


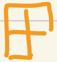


Many questions remain...

* Symmetric trilinear forms are alternating trilinear forms are irreducible reps of GL(n, C)

$U \cong \mathbb{F}^n$, $U \odot U \odot U$ 

$U \wedge U \wedge U$ 



- Orbit problems for these actions are TI-complete
- How about the other one?

* Tensor Iso over GL(n, Q): is this decidable?

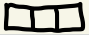
* Lysikov and Walter introduced a complexity class Tensor Orbit Closure Intersection

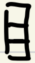
- The relation between TOCI and TI?

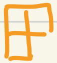
Many questions remain...

* Symmetric trilinear forms are alternating trilinear forms are irreducible reps of GL(n, C)

  - Orbit problems for these actions are TI-complete
  - How about the other one?

$U \cong \mathbb{F}^n$, $U \odot U \odot U$ 

$U \wedge U \wedge U$ 



* Tensor Iso over GL(n, Q): is this decidable?

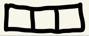
* Lysikov and Walter introduced a complexity class Tensor Orbit Closure Intersection
  - The relation between TOCI and TI?

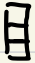
* 2x2x2 TensorIso over GL(2, Z) is in BQP [Bhargava, Hallgren]
  - How about 3x3x3? (Only known to be decidable)

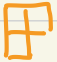
Many questions remain…

* Symmetric trilinear forms are alternating trilinear forms are irreducible reps of GL(n, C)

- Orbit problems for these actions are TI-complete
- How about the other one?

$U \cong \mathbb{F}^n$, $U \odot U \odot U$ 

$U \wedge U \wedge U$ 



* Tensor Iso over GL(n, Q): is this decidable?

* Lysikov and Walter introduced a complexity class Tensor Orbit Closure Intersection

- The relation between TOCI and TI?

* 2x2x2 TensorIso over GL(2, Z) is in BQP [Bhargava, Hallgren]

- How about 3x3x3? (Only known to be decidable)

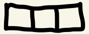
* Is Tensor Iso group action pseudorandom?

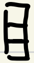
- A possible approach via hyperdeterminant [Joux—Narayanan]

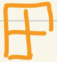
Many questions remain...

* Symmetric trilinear forms are alternating trilinear forms are irreducible reps of GL(n, C)

- Orbit problems for these actions are TI-complete
- How about the other one?

$U \cong \mathbb{F}^n$, $U \odot U \odot U$ 

$U \wedge U \wedge U$ 



* Tensor Iso over GL(n, Q): is this decidable?

* Lysikov and Walter introduced a complexity class Tensor Orbit Closure Intersection

- The relation between TOCI and TI?

* 2x2x2 TensorIso over GL(2, Z) is in BQP [Bhargava, Hallgren]

- How about 3x3x3? (Only known to be decidable)

* Is Tensor Iso group action pseudorandom?

- A possible approach via hyperdeterminant [Joux–Narayanan]

* nxnx2 TensorIso over GF(q): polynomial-time?

Thank you!

And questions please :)