How to make Algebraic Computations GRH free?

Nitin Saxena[1]
(with Gábor Ivanyos[2], Marek Karpinski[1] and Lajos Rónyai[2])

[1]Hausdorff Center for Mathematics, Bonn

[2]Computer and Automation Research Institute, Budapest

NTACC Workshop 2010
Warsaw

Introduction

Polynomial Factoring

The Problem
GRH Connection
Finite Algebra Questions

Standard Algebraic Terms

Outline of Part II

Our Results: Commutative Algebras

New Concepts / Tools

- Semiregularity
- Lagrange Resolvent
- Kummer Extension

A Warmup Application

Proof of the Main Result

Outline of Part III

Our Results: Noncommutative Algebras

Proof of the Main Result

Part I

Introduction

Outline

Polynomial Factoring over Finite Fields

- Given a polynomial $f(x) \in \mathbb{F}_q[x]$ we want a nontrivial factor.
- It is not only a fundamental problem but also has practical applications: coding theory, integer factoring algorithms, cryptography, computer algebra, ...
- Berlekamp (1967) showed that the problem reduces in deterministic polynomial time to the problem of: *factoring a degree n polynomial with n distinct roots in a prime field $\mathbb{F}_p$.*

Polynomial Factoring over Finite Fields

- Given a polynomial $f(x) \in \mathbb{F}_q[x]$ we want a nontrivial factor.
- It is not only a fundamental problem but also has practical applications: coding theory, integer factoring algorithms, cryptography, computer algebra, ...
- Berlekamp (1967) showed that the problem reduces in deterministic polynomial time to the problem of: *factoring a degree n polynomial with n distinct roots in a prime field $\mathbb{F}_p$.*

Polynomial Factoring over Finite Fields

- Given a polynomial $f(x) \in \mathbb{F}_q[x]$ we want a nontrivial factor.
- It is not only a fundamental problem but also has practical applications: coding theory, integer factoring algorithms, cryptography, computer algebra, ...
- Berlekamp (1967) showed that the problem reduces in deterministic polynomial time to the problem of: *factoring a degree n polynomial with n distinct roots in a prime field $\mathbb{F}_p$.*

Polynomial Factoring Methods

- Let $f(x)$ be the input polynomial of degree n with distinct n roots in $\mathbb{F}_p$.
- Factoring is very well studied: (Legendre 1700s), (Berlekamp 1967), (Moenck 1977), (Rabin 1980), (Cantor, Zassenhaus 1981), (Camion 1983), (Huang 1985), (Schoof 1985), (von zur Gathen 1987), (Mignotte, Schnorr 1988), (Evdokimov 1989, 1994), (von zur Gathen, Shoup 1992), (Kaltofen, Shoup 1995), (Cheng, Huang 2000), (Bach, von zur Gathen, Lenstra 2001), (Gao 2001), (Stein 2001), (van de Woestijne 2005), (Kedlaya, Umans 2008), (Ivanyos, Karpinski, Saxena 2009), (Zrałek 2010),.....
- The best deterministic algorithm known takes time $O^{\sim}(n^2\sqrt{p})$ (S90).
- The really useful algorithms - (B67), (CZ81), (vzGS92), (KS95) - are all randomized and take $\mathrm{poly}(n \log p)$ time.
- It is an open question to derandomize them.

Polynomial Factoring Methods

- Let $f(x)$ be the input polynomial of degree n with distinct n roots in $\mathbb{F}_p$.
- Factoring is very well studied: (Legendre 1700s), (Berlekamp 1967), (Moenck 1977), (Rabin 1980), (Cantor, Zassenhaus 1981), (Camion 1983), (Huang 1985), (Schoof 1985), (von zur Gathen 1987), (Mignotte, Schnorr 1988), (Evdokimov 1989, 1994), (von zur Gathen, Shoup 1992), (Kaltofen, Shoup 1995), (Cheng, Huang 2000), (Bach, von zur Gathen, Lenstra 2001), (Gao 2001), (Stein 2001), (van de Woestijne 2005), (Kedlaya, Umans 2008), (Ivanyos, Karpinski, Saxena 2009), (Zrałek 2010),.....
- The best deterministic algorithm known takes time $O^{\sim}(n^2\sqrt{p})$ (S90).
- The really useful algorithms - (B67), (CZ81), (vzGS92), (KS95) - are all randomized and take $\mathrm{poly}(n \log p)$ time.
- It is an open question to derandomize them.

Polynomial Factoring Methods

- Let $f(x)$ be the input polynomial of degree n with distinct n roots in $\mathbb{F}_p$.
- Factoring is very well studied: (Legendre 1700s), (Berlekamp 1967), (Moenck 1977), (Rabin 1980), (Cantor, Zassenhaus 1981), (Camion 1983), (Huang 1985), (Schoof 1985), (von zur Gathen 1987), (Mignotte, Schnorr 1988), (Evdokimov 1989, 1994), (von zur Gathen, Shoup 1992), (Kaltofen, Shoup 1995), (Cheng, Huang 2000), (Bach, von zur Gathen, Lenstra 2001), (Gao 2001), (Stein 2001), (van de Woestijne 2005), (Kedlaya, Umans 2008), (Ivanyos, Karpinski, Saxena 2009), (Zrałek 2010),.....
- The best deterministic algorithm known takes time $O^{\sim}(n^2\sqrt{p})$ (S90).
- The really useful algorithms - (B67), (CZ81), (vzGS92), (KS95) - are all randomized and take $\mathrm{poly}(n \log p)$ time.
- It is an open question to derandomize them.

Polynomial Factoring Methods

- Let $f(x)$ be the input polynomial of degree n with distinct n roots in $\mathbb{F}_p$.
- Factoring is very well studied: (Legendre 1700s), (Berlekamp 1967), (Moenck 1977), (Rabin 1980), (Cantor, Zassenhaus 1981), (Camion 1983), (Huang 1985), (Schoof 1985), (von zur Gathen 1987), (Mignotte, Schnorr 1988), (Evdokimov 1989, 1994), (von zur Gathen, Shoup 1992), (Kaltofen, Shoup 1995), (Cheng, Huang 2000), (Bach, von zur Gathen, Lenstra 2001), (Gao 2001), (Stein 2001), (van de Woestijne 2005), (Kedlaya, Umans 2008), (Ivanyos, Karpinski, Saxena 2009), (Zrałek 2010),.....
- The best deterministic algorithm known takes time $O^{\sim}(n^2\sqrt{p})$ (S90).
- The really useful algorithms - (B67), (CZ81), (vzGS92), (KS95) - are all randomized and take $\mathrm{poly}(n\log p)$ time.
- It is an open question to derandomize them.

Polynomial Factoring Methods

- Let $f(x)$ be the input polynomial of degree n with distinct n roots in $\mathbb{F}_p$.
- Factoring is very well studied: (Legendre 1700s), (Berlekamp 1967), (Moenck 1977), (Rabin 1980), (Cantor, Zassenhaus 1981), (Camion 1983), (Huang 1985), (Schoof 1985), (von zur Gathen 1987), (Mignotte, Schnorr 1988), (Evdokimov 1989, 1994), (von zur Gathen, Shoup 1992), (Kaltofen, Shoup 1995), (Cheng, Huang 2000), (Bach, von zur Gathen, Lenstra 2001), (Gao 2001), (Stein 2001), (van de Woestijne 2005), (Kedlaya, Umans 2008), (Ivanyos, Karpinski, Saxena 2009), (Zrałek 2010),.....
- The best deterministic algorithm known takes time $O^{\sim}(n^2\sqrt{p})$ (S90).
- The really useful algorithms - (B67), (CZ81), (vzGS92), (KS95) - are all randomized and take $\mathrm{poly}(n \log p)$ time.
- It is an open question to derandomize them.

Reminder: Randomized Factoring

- The simplest (and practical) algorithm was already suggested by Legendre (1752-1833).
- Given $f(x)$ of degree n having that many roots in $\mathbb{F}_p$.
- Choose a *random* $a \in \mathbb{F}_p$.
- Compute $g(x) := \gcd(f(x+a), x^{\frac{p-1}{2}} - 1)$.
- With more than 50% chance $g(x)$ is a nontrivial factor!
- Key fact: $(x^{\frac{p-1}{2}} - 1)$ 'collects' the *squares* mod p, and is easy to compute $\pmod{f(x)}$.

□

Fig: Legendre?

Reminder: Randomized Factoring

- The simplest (and practical) algorithm was already suggested by Legendre (1752-1833).
- Given $f(x)$ of degree n having that many roots in $\mathbb{F}_p$.
- Choose a *random* $a \in \mathbb{F}_p$.
- Compute $g(x) := \gcd(f(x+a), x^{\frac{p-1}{2}} - 1)$.
- With more than 50% chance $g(x)$ is a nontrivial factor!
- Key fact: $(x^{\frac{p-1}{2}} - 1)$ 'collects' the *squares* mod p, and is easy to compute $\pmod{f(x)}$.

Fig: Legendre?

Reminder: Randomized Factoring

- The simplest (and practical) algorithm was already suggested by Legendre (1752-1833).
- Given $f(x)$ of degree n having that many roots in $\mathbb{F}_p$.
- Choose a *random* $a \in \mathbb{F}_p$.
- Compute $g(x) := \gcd(f(x+a), x^{\frac{p-1}{2}} - 1)$.
- With more than 50% chance $g(x)$ is a nontrivial factor!
- Key fact: $(x^{\frac{p-1}{2}} - 1)$ 'collects' the *squares* mod p, and is easy to compute $\pmod{f(x)}$.

□

Fig: Legendre?

Reminder: Randomized Factoring

- The simplest (and practical) algorithm was already suggested by Legendre (1752-1833).
- Given $f(x)$ of degree n having that many roots in $\mathbb{F}_p$.
- Choose a *random* $a \in \mathbb{F}_p$.
- Compute $g(x) := \gcd(f(x+a), x^{\frac{p-1}{2}} - 1)$.
- With more than 50% chance $g(x)$ is a nontrivial factor!
- Key fact: $(x^{\frac{p-1}{2}} - 1)$ 'collects' the *squares* mod p, and is easy to compute $\pmod{f(x)}$.

Fig: Legendre?

Reminder: Randomized Factoring

- The simplest (and practical) algorithm was already suggested by Legendre (1752-1833).
- Given $f(x)$ of degree n having that many roots in $\mathbb{F}_p$.
- Choose a *random* $a \in \mathbb{F}_p$.
- Compute $g(x) := \gcd(f(x+a), x^{\frac{p-1}{2}} - 1)$.
- With more than 50% chance $g(x)$ is a nontrivial factor!
- Key fact: $(x^{\frac{p-1}{2}} - 1)$ 'collects' the *squares* mod p, and is easy to compute $\pmod{f(x)}$.

Fig: Legendre?

Reminder: Randomized Factoring

- The simplest (and practical) algorithm was already suggested by Legendre (1752-1833).
- Given $f(x)$ of degree n having that many roots in $\mathbb{F}_p$.
- Choose a *random* $a \in \mathbb{F}_p$.
- Compute $g(x) := \gcd(f(x+a), x^{\frac{p-1}{2}} - 1)$.
- With more than 50% chance $g(x)$ is a nontrivial factor!
- Key fact: $(x^{\frac{p-1}{2}} - 1)$ 'collects' the *squares* mod p, and is easy to compute $\pmod{f(x)}$.

□

Fig: Legendre?

Reminder: Randomized Factoring

- The simplest (and practical) algorithm was already suggested by Legendre (1752-1833).
- Given $f(x)$ of degree n having that many roots in $\mathbb{F}_p$.
- Choose a *random* $a \in \mathbb{F}_p$.
- Compute $g(x) := \gcd(f(x+a), x^{\frac{p-1}{2}} - 1)$.
- With more than 50% chance $g(x)$ is a nontrivial factor!
- Key fact: $(x^{\frac{p-1}{2}} - 1)$ 'collects' the *squares* mod p, and is easy to compute $\pmod{f(x)}$.

Outline

Riemann Hypothesis & Polynomial Factoring

Generalized Riemann Hypothesis (GRH)

For any Dirichlet character χ and a complex root s of the Dirichlet L-function $L(\chi, s) = \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s}$: if $\text{Re}(s) \in [0, 1]$ then $\text{Re}(s) = \frac{1}{2}$.

- Generalized Riemann Hypothesis (GRH) has been useful in understanding the deterministic complexity of polynomial factoring, albeit only in special cases.
- Most prominently, a degree n polynomial $f(x)$ can be nontrivially factored in deterministic $\text{poly}(\log p, n^{\log n})$ time by GRH (Evdokimov 1994).
- From such results we eliminate GRH (with a caveat!).

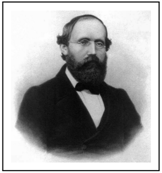

Fig: Riemann

Riemann Hypothesis & Polynomial Factoring

Generalized Riemann Hypothesis (GRH)

For any Dirichlet character χ and a complex root s of the Dirichlet L-function $L(\chi, s) = \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s}$: if $\mathrm{Re}(s) \in [0, 1]$ then $\mathrm{Re}(s) = \frac{1}{2}$.

- Generalized Riemann Hypothesis (GRH) has been useful in understanding the deterministic complexity of polynomial factoring, albeit only in special cases.
- Most prominently, a degree n polynomial $f(x)$ can be nontrivially factored in deterministic $\mathrm{poly}(\log p, n^{\log n})$ time by GRH (Evdokimov 1994).
- From such results we eliminate GRH (with a caveat!).

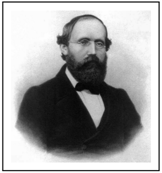

Fig: Riemann

Riemann Hypothesis & Polynomial Factoring

Generalized Riemann Hypothesis (GRH)

For any Dirichlet character χ and a complex root s of the Dirichlet L-function $L(\chi, s) = \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s}$: if $\mathrm{Re}(s) \in [0,1]$ then $\mathrm{Re}(s) = \frac{1}{2}$.

- Generalized Riemann Hypothesis (GRH) has been useful in understanding the deterministic complexity of polynomial factoring, albeit only in special cases.
- Most prominently, a degree n polynomial $f(x)$ can be nontrivially factored in deterministic $\mathrm{poly}(\log p, n^{\log n})$ time by GRH (Evdokimov 1994).
- From such results we eliminate GRH (with a caveat!).

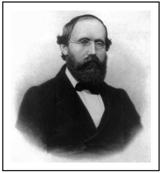

Fig: Riemann

Riemann Hypothesis & Polynomial Factoring

Generalized Riemann Hypothesis (GRH)

For any Dirichlet character χ and a complex root s of the Dirichlet L-function $L(\chi, s) = \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s}$: if $\mathrm{Re}(s) \in [0, 1]$ then $\mathrm{Re}(s) = \frac{1}{2}$.

- Generalized Riemann Hypothesis (GRH) has been useful in understanding the deterministic complexity of polynomial factoring, albeit only in special cases.
- Most prominently, a degree n polynomial $f(x)$ can be nontrivially factored in deterministic $\mathrm{poly}(\log p, n^{\log n})$ time by GRH (Evdokimov 1994).
- From such results we eliminate GRH (with a caveat!).

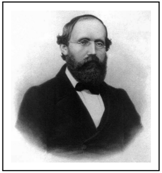

Fig: Riemann

Outline

Finite Algebra over a Finite Field

- Polynomial factoring applies to structural questions in finite algebras.
- Friedl & Rónyai (1985) showed that finding zero divisors in finite algebras over finite fields reduces to polynomial factoring.
- Thus, under GRH, they gave a $\mathrm{poly}(\log p, n^{\log n})$ time deterministic algorithm for finding zero divisors.
- Our methods, in noncommutative algebras, make this algorithm completely GRH free.

Finite Algebra over a Finite Field

- Polynomial factoring applies to structural questions in finite algebras.
- Friedl & Rónyai (1985) showed that finding zero divisors in finite algebras over finite fields reduces to polynomial factoring.
- Thus, under GRH, they gave a $\mathrm{poly}(\log p, n^{\log n})$ time deterministic algorithm for finding zero divisors.
- Our methods, in noncommutative algebras, make this algorithm completely GRH free.

Finite Algebra over a Finite Field

- Polynomial factoring applies to structural questions in finite algebras.
- Friedl & Rónyai (1985) showed that finding zero divisors in finite algebras over finite fields reduces to polynomial factoring.
- Thus, under GRH, they gave a $\mathrm{poly}(\log p, n^{\log n})$ time deterministic algorithm for finding zero divisors.
- Our methods, in noncommutative algebras, make this algorithm completely GRH free.

Finite Algebra over a Finite Field

- Polynomial factoring applies to structural questions in finite algebras.
- Friedl & Rónyai (1985) showed that finding zero divisors in finite algebras over finite fields reduces to polynomial factoring.
- Thus, under GRH, they gave a $\mathrm{poly}(\log p, n^{\log n})$ time deterministic algorithm for finding zero divisors.
- Our methods, in noncommutative algebras, make this algorithm completely GRH free.

How to make our world GRH free?

- Assuming GRH, there is a poly-time algorithm to compute $\sqrt[r]{a} \pmod{p}$ (Huang 1985).
- Any algorithm that assumes GRH, invokes the above routine to compute r-th roots in an algebra $\mathcal{A}$.
- What if instead of computing the r-th root explicitly, we use an implicit root?
- I.e., we simply go to the extension algebra $\mathcal{A}[\zeta_r][\sqrt[r]{a}]$, explicitly $\mathcal{A}[X, Y]/(\sum_{i=0}^{r-1} X^i, Y^r - a)$.
- We make this idea work by developing a Galois theory for algebras.

How to make our world GRH free?

- Assuming GRH, there is a poly-time algorithm to compute $\sqrt[r]{a} \pmod{p}$ (Huang 1985).
- Any algorithm that assumes GRH, invokes the above routine to compute r-th roots in an algebra $\mathcal{A}$.
- What if instead of computing the r-th root explicitly, we use an implicit root?
- I.e., we simply go to the extension algebra $\mathcal{A}[\zeta_r][\sqrt[r]{a}]$, explicitly $\mathcal{A}[X, Y]/(\sum_{i=0}^{r-1} X^i, Y^r - a)$.
- We make this idea work by developing a Galois theory for algebras.

How to make our world GRH free?

- Assuming GRH, there is a poly-time algorithm to compute $\sqrt[r]{a} \pmod{p}$ (Huang 1985).
- Any algorithm that assumes GRH, invokes the above routine to compute r-th roots in an algebra $\mathcal{A}$.
- What if instead of computing the r-th root explicitly, we use an implicit root?
- I.e., we simply go to the extension algebra $\mathcal{A}[\zeta_r][\sqrt[r]{a}]$, explicitly $\mathcal{A}[X, Y]/(\sum_{i=0}^{r-1} X^i, Y^r - a)$.
- We make this idea work by developing a Galois theory for algebras.

How to make our world GRH free?

- Assuming GRH, there is a poly-time algorithm to compute $\sqrt[r]{a} \pmod{p}$ (Huang 1985).
- Any algorithm that assumes GRH, invokes the above routine to compute r-th roots in an algebra $\mathcal{A}$.
- What if instead of computing the r-th root explicitly, we use an implicit root?
- I.e., we simply go to the extension algebra $\mathcal{A}[\zeta_r][\sqrt[r]{a}]$, explicitly $\mathcal{A}[X, Y]/(\sum_{i=0}^{r-1} X^i, Y^r - a)$.
- We make this idea work by developing a Galois theory for algebras.

How to make our world GRH free?

- Assuming GRH, there is a poly-time algorithm to compute $\sqrt[r]{a} \pmod{p}$ (Huang 1985).
- Any algorithm that assumes GRH, invokes the above routine to compute r-th roots in an algebra $\mathcal{A}$.
- What if instead of computing the r-th root explicitly, we use an implicit root?
- I.e., we simply go to the extension algebra $\mathcal{A}[\zeta_r][\sqrt[r]{a}]$, explicitly $\mathcal{A}[X,Y]/(\sum_{i=0}^{r-1} X^i, Y^r - a)$.
- We make this idea work by developing a Galois theory for algebras.

Reminder: Galois theory

- Galois (1811-1832) studied *fields* by *groups*.
- For a field *extension* $K \subset L$ consider the group G_L of *automorphisms* of L that *fix* K elementwise.
- Essentially, $[L : K] = |G_L|$.
- Essentially, there is a 1-1 *correspondence* between the subfields of L and subgroups of G_L.
- The first triumph of Galois theory: *quintic polynomials cannot be solved by radicals*.
- 'Positive' side-effect: for special polynomials the theory gives a systematic way to express roots using radicals!

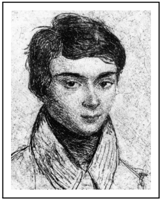

Fig: Galois

Reminder: Galois theory

- Galois (1811-1832) studied *fields* by *groups*.
- For a field *extension* $K \subset L$ consider the group G_L of *automorphisms* of L that *fix* K elementwise.
- Essentially, $[L : K] = |G_L|$.
- Essentially, there is a 1-1 *correspondence* between the subfields of L and subgroups of G_L.
- The first triumph of Galois theory: *quintic polynomials cannot be solved by radicals*.
- 'Positive' side-effect: for special polynomials the theory gives a systematic way to express roots using radicals!

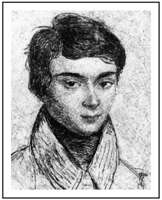

Fig: Galois

Reminder: Galois theory

- Galois (1811-1832) studied *fields* by *groups*.
- For a field *extension* $K \subset L$ consider the group G_L of *automorphisms* of L that *fix* K elementwise.
- Essentially, $[L : K] = |G_L|$.
- Essentially, there is a 1-1 *correspondence* between the subfields of L and subgroups of G_L.
- The first triumph of Galois theory: *quintic polynomials cannot be solved by radicals*.
- 'Positive' side-effect: for special polynomials the theory gives a systematic way to express roots using radicals!

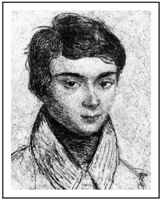

Fig: Galois

Reminder: Galois theory

- Galois (1811-1832) studied *fields* by *groups*.
- For a field *extension* $K \subset L$ consider the group G_L of *automorphisms* of L that *fix* K elementwise.
- Essentially, $[L : K] = |G_L|$.
- Essentially, there is a 1-1 *correspondence* between the subfields of L and subgroups of G_L.
- The first triumph of Galois theory: *quintic polynomials cannot be solved by radicals.*
- 'Positive' side-effect: for special polynomials the theory gives a systematic way to express roots using radicals!

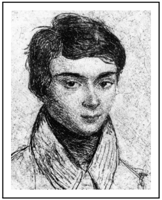

Fig: Galois

Reminder: Galois theory

- Galois (1811-1832) studied *fields* by *groups*.
- For a field *extension* $K \subset L$ consider the group G_L of *automorphisms* of L that *fix* K elementwise.
- Essentially, $[L : K] = |G_L|$.
- Essentially, there is a 1-1 *correspondence* between the subfields of L and subgroups of G_L.
- The first triumph of Galois theory: *quintic polynomials cannot be solved by radicals*.
- 'Positive' side-effect: for special polynomials the theory gives a systematic way to express roots using radicals!

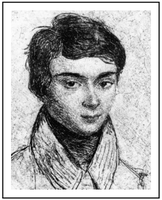

Fig: Galois

Reminder: Galois theory

- Galois (1811-1832) studied *fields* by *groups*.
- For a field *extension* $K \subset L$ consider the group G_L of *automorphisms* of L that *fix* K elementwise.
- Essentially, $[L : K] = |G_L|$.
- Essentially, there is a 1-1 *correspondence* between the subfields of L and subgroups of G_L.
- The first triumph of Galois theory: *quintic polynomials cannot be solved by radicals*.
- 'Positive' side-effect: for special polynomials the theory gives a systematic way to express roots using radicals!

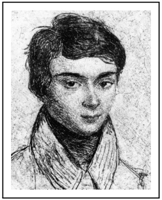

Fig: Galois

Outline

Module Terms

- Let R be a ring. A left R-module M consists of an abelian group $(M, +)$ and a *scalar multiplication* $R \times M \to M$ satisfying natural conditions.
- Just *R-module* when scalar multiplication commutes.
- Free R-module M if there is a free basis $B \subset M$ s.t. every element in M has a *unique* representation as $\sum_{b \in B} r_b b$ ($r_b \in R$).
- Rank $\mathrm{rk}_R M$ is *the* size of a free basis.
- Example: a vector space is a free module of rank equal to its dimension.

Module Terms

- Let R be a ring. A left R-module M consists of an abelian group $(M, +)$ and a *scalar multiplication* $R \times M \to M$ satisfying natural conditions.
- Just R-*module* when scalar multiplication commutes.
- Free R-module M if there is a free basis $B \subset M$ s.t. every element in M has a *unique* representation as $\sum_{b \in B} r_b b$ $(r_b \in R)$.
- Rank $\mathrm{rk}_R M$ is *the* size of a free basis.
- Example: a vector space is a free module of rank equal to its dimension.

Module Terms

- Let R be a ring. A left R-module M consists of an abelian group $(M, +)$ and a *scalar multiplication* $R \times M \to M$ satisfying natural conditions.
- Just R-*module* when scalar multiplication commutes.
- Free R-module M if there is a free basis $B \subset M$ s.t. every element in M has a *unique* representation as $\sum_{b \in B} r_b b$ $(r_b \in R)$.
- Rank $\mathrm{rk}_R M$ is *the* size of a free basis.
- Example: a vector space is a free module of rank equal to its dimension.

Module Terms

- Let R be a ring. A left R-module M consists of an abelian group $(M, +)$ and a *scalar multiplication* $R \times M \to M$ satisfying natural conditions.
- Just R-*module* when scalar multiplication commutes.
- Free R-module M if there is a free basis $B \subset M$ s.t. every element in M has a *unique* representation as $\sum_{b \in B} r_b b$ $(r_b \in R)$.
- Rank $\mathrm{rk}_R M$ is *the* size of a free basis.
- Example: a vector space is a free module of rank equal to its dimension.

Module Terms

- Let R be a ring. A left R-module M consists of an abelian group $(M, +)$ and a *scalar multiplication* $R \times M \to M$ satisfying natural conditions.
- Just R-*module* when scalar multiplication commutes.
- Free R-module M if there is a free basis $B \subset M$ s.t. every element in M has a *unique* representation as $\sum_{b \in B} r_b b$ ($r_b \in R$).
- Rank $\mathrm{rk}_R M$ is *the* size of a free basis.
- Example: a vector space is a free module of rank equal to its dimension.

Algebra Terms

- An R-algebra $\mathcal{A}$ consists of an R-module $(\mathcal{A}, +)$ and a *multiplication operation in* $\mathcal{A}$ that commutes with the scalar multiplication.
- Example: if $\mathcal{A}$ is a ring with a subring $\mathcal{B}$ in its *center* then $\mathcal{A}$ is a $\mathcal{B}$-algebra.
- A zero divisor $x \in \mathcal{A}$ is a nonzero element s.t. for some nonzero $y, y' \in \mathcal{A}$, $yx = xy' = 0$. (Factor ⇐ Zero divisor)
- An ideal I of the R-algebra $\mathcal{A}$ is an R-submodule s.t. $\mathcal{A}I \subset I$ and $I\mathcal{A} \subset I$. (Trivial: $\{0\}$ and $\mathcal{A}$.)
- Simple algebra has no nontrivial ideals. Example: a field $\mathbb{F}$.
- Semisimple algebra is a *direct sum* of finitely many simple algebras. Example: $\mathbb{F}_p[x]/(f(x))$ for a squarefree $f(x)$.

Algebra Terms

- An R-algebra $\mathcal{A}$ consists of an R-module $(\mathcal{A}, +)$ and a *multiplication operation in* $\mathcal{A}$ that commutes with the scalar multiplication.
- Example: if $\mathcal{A}$ is a ring with a subring $\mathcal{B}$ in its *center* then $\mathcal{A}$ is a $\mathcal{B}$-algebra.
- A zero divisor $x \in \mathcal{A}$ is a nonzero element s.t. for some nonzero $y, y' \in \mathcal{A}$, $yx = xy' = 0$. (Factor ⇐ Zero divisor)
- An ideal I of the R-algebra $\mathcal{A}$ is an R-submodule s.t. $\mathcal{A}I \subset I$ and $I\mathcal{A} \subset I$. (Trivial: $\{0\}$ and $\mathcal{A}$.)
- Simple algebra has no nontrivial ideals. Example: a field $\mathbb{F}$.
- Semisimple algebra is a *direct sum* of finitely many simple algebras. Example: $\mathbb{F}_p[x]/(f(x))$ for a squarefree $f(x)$.

Algebra Terms

- An R-algebra $\mathcal{A}$ consists of an R-module $(\mathcal{A}, +)$ and a *multiplication operation in* $\mathcal{A}$ that commutes with the scalar multiplication.
- Example: if $\mathcal{A}$ is a ring with a subring $\mathcal{B}$ in its *center* then $\mathcal{A}$ is a $\mathcal{B}$-algebra.
- A zero divisor $x \in \mathcal{A}$ is a nonzero element s.t. for some nonzero $y, y' \in \mathcal{A}$, $yx = xy' = 0$. (Factor $\leftrightharpoons$ Zero divisor)
- An ideal I of the R-algebra $\mathcal{A}$ is an R-submodule s.t. $\mathcal{A}I \subset I$ and $I\mathcal{A} \subset I$. (Trivial: $\{0\}$ and $\mathcal{A}$.)
- Simple algebra has no nontrivial ideals. Example: a field $\mathbb{F}$.
- Semisimple algebra is a *direct sum* of finitely many simple algebras. Example: $\mathbb{F}_p[x]/(f(x))$ for a squarefree $f(x)$.

Algebra Terms

- An R-algebra $\mathcal{A}$ consists of an R-module $(\mathcal{A}, +)$ and a *multiplication operation in* $\mathcal{A}$ that commutes with the scalar multiplication.
- Example: if $\mathcal{A}$ is a ring with a subring $\mathcal{B}$ in its *center* then $\mathcal{A}$ is a $\mathcal{B}$-algebra.
- A zero divisor $x \in \mathcal{A}$ is a nonzero element s.t. for some nonzero $y, y' \in \mathcal{A}$, $yx = xy' = 0$. (Factor $\leftrightharpoons$ Zero divisor)
- An ideal I of the R-algebra $\mathcal{A}$ is an R-submodule s.t. $\mathcal{A}I \subset I$ and $I\mathcal{A} \subset I$. (Trivial: $\{0\}$ and $\mathcal{A}$.)
- Simple algebra has no nontrivial ideals. Example: a field $\mathbb{F}$.
- Semisimple algebra is a *direct sum* of finitely many simple algebras. Example: $\mathbb{F}_p[x]/(f(x))$ for a squarefree $f(x)$.

Algebra Terms

- An *R*-algebra $\mathcal{A}$ consists of an R-module $(\mathcal{A}, +)$ and a *multiplication operation in* $\mathcal{A}$ that commutes with the scalar multiplication.
- Example: if $\mathcal{A}$ is a ring with a subring $\mathcal{B}$ in its *center* then $\mathcal{A}$ is a $\mathcal{B}$-algebra.
- A zero divisor $x \in \mathcal{A}$ is a nonzero element s.t. for some nonzero $y, y' \in \mathcal{A}$, $yx = xy' = 0$. (Factor $\leftrightharpoons$ Zero divisor)
- An ideal I of the R-algebra $\mathcal{A}$ is an R-submodule s.t. $\mathcal{A}I \subset I$ and $I\mathcal{A} \subset I$. (Trivial: $\{0\}$ and $\mathcal{A}$.)
- Simple algebra has no nontrivial ideals. Example: a field $\mathbb{F}$.
- Semisimple algebra is a *direct sum* of finitely many simple algebras. Example: $\mathbb{F}_p[x]/(f(x))$ for a squarefree $f(x)$.

Algebra Terms

- An *R*-algebra $\mathcal{A}$ consists of an R-module $(\mathcal{A}, +)$ and a *multiplication operation in* $\mathcal{A}$ that commutes with the scalar multiplication.
- Example: if $\mathcal{A}$ is a ring with a subring $\mathcal{B}$ in its *center* then $\mathcal{A}$ is a $\mathcal{B}$-algebra.
- A zero divisor $x \in \mathcal{A}$ is a nonzero element s.t. for some nonzero $y, y' \in \mathcal{A}$, $yx = xy' = 0$. (Factor $\leftrightharpoons$ Zero divisor)
- An ideal I of the R-algebra $\mathcal{A}$ is an R-submodule s.t. $\mathcal{A}I \subset I$ and $I\mathcal{A} \subset I$. (Trivial: $\{0\}$ and $\mathcal{A}$.)
- Simple algebra has no nontrivial ideals. Example: a field $\mathbb{F}$.
- Semisimple algebra is a *direct sum* of finitely many simple algebras. Example: $\mathbb{F}_p[x]/(f(x))$ for a squarefree $f(x)$.

Algebra Terms

- An R-algebra $\mathcal{A}$ consists of an R-module $(\mathcal{A}, +)$ and a *multiplication operation in* $\mathcal{A}$ that commutes with the scalar multiplication.
- Example: if $\mathcal{A}$ is a ring with a subring $\mathcal{B}$ in its *center* then $\mathcal{A}$ is a $\mathcal{B}$-algebra.
- A zero divisor $x \in \mathcal{A}$ is a nonzero element s.t. for some nonzero $y, y' \in \mathcal{A}$, $yx = xy' = 0$. (Factor $\leftrightharpoons$ Zero divisor)
- An ideal I of the R-algebra $\mathcal{A}$ is an R-submodule s.t. $\mathcal{A}I \subset I$ and $I\mathcal{A} \subset I$. (Trivial: $\{0\}$ and $\mathcal{A}$.)
- Simple algebra has no nontrivial ideals. Example: a field $\mathbb{F}$.
- Semisimple algebra is a *direct sum* of finitely many simple algebras. Example: $\mathbb{F}_p[x]/(f(x))$ for a squarefree $f(x)$.

Algebra Terms (Contd.)

- An algebra $\mathcal{A}$ is an extension of a subalgebra $\mathcal{B}$ if $\mathcal{A}$ is a free $\mathcal{B}$-module. Example: $\mathbb{F}_p[x]/(f(x))$ extends $\mathbb{F}_p$.
- Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be $\mathcal{B}$-algebras. The tensor product $\mathcal{A}_1 \otimes_{\mathcal{B}} \mathcal{A}_2$ is a $\mathcal{B}$-module with generators $a_1 \otimes a_2$. It is also an algebra with multiplication: $(a_1 \otimes a_2) \cdot (a_1' \otimes a_2') = (a_1 a_1' \otimes a_2 a_2')$.
- Example: $\mathbb{F}[x]/(f(x)) \otimes_{\mathbb{F}} \mathbb{F}[y]/(g(y)) \cong \mathbb{F}[x, y]/(f(x), g(y))$.
- $\mathcal{B}$-homomorphism from algebra $\mathcal{A}_1$ to $\mathcal{A}_2$ is a map that preserves *all* operations and fixes $\mathcal{B}$ elementwise.
- Homomorphisms can be injective, surjective or both.
- The group of $\mathbb{F}$-automorphisms of $\mathcal{A}$, $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.

Algebra Terms (Contd.)

- An algebra $\mathcal{A}$ is an extension of a subalgebra $\mathcal{B}$ if $\mathcal{A}$ is a free $\mathcal{B}$-module. Example: $\mathbb{F}_p[x]/(f(x))$ extends $\mathbb{F}_p$.
- Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be $\mathcal{B}$-algebras. The tensor product $\mathcal{A}_1 \otimes_{\mathcal{B}} \mathcal{A}_2$ is a $\mathcal{B}$-module with generators $a_1 \otimes a_2$. It is also an algebra with multiplication: $(a_1 \otimes a_2) \cdot (a_1' \otimes a_2') = (a_1 a_1' \otimes a_2 a_2')$.
- Example: $\mathbb{F}[x]/(f(x)) \otimes_{\mathbb{F}} \mathbb{F}[y]/(g(y)) \cong \mathbb{F}[x, y]/(f(x), g(y))$.
- $\mathcal{B}$-homomorphism from algebra $\mathcal{A}_1$ to $\mathcal{A}_2$ is a map that preserves *all* operations and fixes $\mathcal{B}$ elementwise.
- Homomorphisms can be injective, surjective or both.
- The group of $\mathbb{F}$-automorphisms of $\mathcal{A}$, $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.

Algebra Terms (Contd.)

- An algebra $\mathcal{A}$ is an extension of a subalgebra $\mathcal{B}$ if $\mathcal{A}$ is a free $\mathcal{B}$-module. Example: $\mathbb{F}_p[x]/(f(x))$ extends $\mathbb{F}_p$.
- Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be $\mathcal{B}$-algebras. The tensor product $\mathcal{A}_1 \otimes_{\mathcal{B}} \mathcal{A}_2$ is a $\mathcal{B}$-module with generators $a_1 \otimes a_2$. It is also an algebra with multiplication: $(a_1 \otimes a_2) \cdot (a_1' \otimes a_2') = (a_1 a_1' \otimes a_2 a_2')$.
- Example: $\mathbb{F}[x]/(f(x)) \otimes_{\mathbb{F}} \mathbb{F}[y]/(g(y)) \cong \mathbb{F}[x,y]/(f(x), g(y))$.
- $\mathcal{B}$-homomorphism from algebra $\mathcal{A}_1$ to $\mathcal{A}_2$ is a map that preserves *all* operations and fixes $\mathcal{B}$ elementwise.
- Homomorphisms can be injective, surjective or both.
- The group of $\mathbb{F}$-automorphisms of $\mathcal{A}$, $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.

Algebra Terms (Contd.)

- An algebra $\mathcal{A}$ is an extension of a subalgebra $\mathcal{B}$ if $\mathcal{A}$ is a free $\mathcal{B}$-module. Example: $\mathbb{F}_p[x]/(f(x))$ extends $\mathbb{F}_p$.
- Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be $\mathcal{B}$-algebras. The tensor product $\mathcal{A}_1 \otimes_{\mathcal{B}} \mathcal{A}_2$ is a $\mathcal{B}$-module with generators $a_1 \otimes a_2$. It is also an algebra with multiplication: $(a_1 \otimes a_2) \cdot (a_1' \otimes a_2') = (a_1 a_1' \otimes a_2 a_2')$.
- Example: $\mathbb{F}[x]/(f(x)) \otimes_{\mathbb{F}} \mathbb{F}[y]/(g(y)) \cong \mathbb{F}[x, y]/(f(x), g(y))$.
- $\mathcal{B}$-homomorphism from algebra $\mathcal{A}_1$ to $\mathcal{A}_2$ is a map that preserves *all* operations and fixes $\mathcal{B}$ elementwise.
- Homomorphisms can be injective, surjective or both.
- The group of $\mathbb{F}$-automorphisms of $\mathcal{A}$, $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.

Algebra Terms (Contd.)

- An algebra $\mathcal{A}$ is an extension of a subalgebra $\mathcal{B}$ if $\mathcal{A}$ is a free $\mathcal{B}$-module. Example: $\mathbb{F}_p[x]/(f(x))$ extends $\mathbb{F}_p$.
- Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be $\mathcal{B}$-algebras. The tensor product $\mathcal{A}_1 \otimes_{\mathcal{B}} \mathcal{A}_2$ is a $\mathcal{B}$-module with generators $a_1 \otimes a_2$. It is also an algebra with multiplication: $(a_1 \otimes a_2) \cdot (a_1' \otimes a_2') = (a_1 a_1' \otimes a_2 a_2')$.
- Example: $\mathbb{F}[x]/(f(x)) \otimes_{\mathbb{F}} \mathbb{F}[y]/(g(y)) \cong \mathbb{F}[x,y]/(f(x), g(y))$.
- $\mathcal{B}$-homomorphism from algebra $\mathcal{A}_1$ to $\mathcal{A}_2$ is a map that preserves *all* operations and fixes $\mathcal{B}$ elementwise.
- Homomorphisms can be injective, surjective or both.
- The group of $\mathbb{F}$-automorphisms of $\mathcal{A}$, $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.

Algebra Terms (Contd.)

- An algebra $\mathcal{A}$ is an extension of a subalgebra $\mathcal{B}$ if $\mathcal{A}$ is a free $\mathcal{B}$-module. Example: $\mathbb{F}_p[x]/(f(x))$ extends $\mathbb{F}_p$.
- Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be $\mathcal{B}$-algebras. The tensor product $\mathcal{A}_1 \otimes_{\mathcal{B}} \mathcal{A}_2$ is a $\mathcal{B}$-module with generators $a_1 \otimes a_2$. It is also an algebra with multiplication: $(a_1 \otimes a_2) \cdot (a_1' \otimes a_2') = (a_1 a_1' \otimes a_2 a_2')$.
- Example: $\mathbb{F}[x]/(f(x)) \otimes_{\mathbb{F}} \mathbb{F}[y]/(g(y)) \cong \mathbb{F}[x,y]/(f(x), g(y))$.
- $\mathcal{B}$-homomorphism from algebra $\mathcal{A}_1$ to $\mathcal{A}_2$ is a map that preserves *all* operations and fixes $\mathcal{B}$ elementwise.
- Homomorphisms can be injective, surjective or both.
- The group of $\mathbb{F}$-automorphisms of $\mathcal{A}$, $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.

Input/Output Representation

- We only consider finite algebras over finite fields.
- An algebra $\mathcal{A}$ over a finite field $\mathbb{F}$ is given in basis form.
- Basis elements $b_1, \ldots, b_n \in \mathcal{A}$ are given together with the relations $b_i \cdot b_j = \sum_{\ell=1}^{n} \alpha_{i,j,\ell} b_\ell$ (α-s in $\mathbb{F}$).
- Homomorphisms between algebras are also presented in basis form, i.e. by giving the respective images of $b_1, \ldots, b_n$.

Input/Output Representation

- We only consider finite algebras over finite fields.
- An algebra $\mathcal{A}$ over a finite field $\mathbb{F}$ is given in basis form.
- Basis elements $b_1, \ldots, b_n \in \mathcal{A}$ are given together with the relations $b_i \cdot b_j = \sum_{\ell=1}^{n} \alpha_{i,j,\ell} b_\ell$ (α-s in $\mathbb{F}$).
- Homomorphisms between algebras are also presented in basis form, i.e. by giving the respective images of $b_1, \ldots, b_n$.

Input/Output Representation

- We only consider finite algebras over finite fields.
- An algebra $\mathcal{A}$ over a finite field $\mathbb{F}$ is given in basis form.
- Basis elements $b_1, \ldots, b_n \in \mathcal{A}$ are given together with the relations $b_i \cdot b_j = \sum_{\ell=1}^{n} \alpha_{i,j,\ell} b_\ell$ (α-s in $\mathbb{F}$).
- Homomorphisms between algebras are also presented in basis form, i.e. by giving the respective images of $b_1, \ldots, b_n$.

Input/Output Representation

- We only consider finite algebras over finite fields.
- An algebra $\mathcal{A}$ over a finite field $\mathbb{F}$ is given in basis form.
- Basis elements $b_1, \ldots, b_n \in \mathcal{A}$ are given together with the relations $b_i \cdot b_j = \sum_{\ell=1}^{n} \alpha_{i,j,\ell} b_\ell$ (α-s in $\mathbb{F}$).
- Homomorphisms between algebras are also presented in basis form, i.e. by giving the respective images of $b_1, \ldots, b_n$.

Part II

Commutative

Outline

Either Factor or Find an Automorphism

- Input: A polynomial $f(x)$, over a finite field $\mathbb{F}$, of degree n.
- Output: *Either* we find a nontrivial factor of $f(x)$ *or* a nontrivial automorphism σ, of $\mathcal{A} = \mathbb{F}[x]/(f(x))$, of order n.
- Complexity: Deterministic $\mathrm{poly}(\log |\mathbb{F}|, n^{\log n})$ time.
- In a sense we do find all the roots of $f(X)$. But they live in $\mathcal{A}$, namely, $x, \sigma(x), \ldots, \sigma^{n-1}(x) \in \mathcal{A}$.
- Such a σ is easy to find in $\mathbb{F}[x]/(x^2 - a)$, eg. $x \mapsto -x$ works. But in other cases is a very nontrivial question.

Either Factor or Find an Automorphism

- Input: A polynomial $f(x)$, over a finite field $\mathbb{F}$, of degree n.
- Output: *Either* we find a nontrivial factor of $f(x)$ *or* a nontrivial automorphism σ, of $\mathcal{A} = \mathbb{F}[x]/(f(x))$, of order n.
- Complexity: Deterministic $\mathrm{poly}(\log|\mathbb{F}|, n^{\log n})$ time.
- In a sense we do find all the roots of $f(X)$. But they live in $\mathcal{A}$, namely, $x, \sigma(x), \ldots, \sigma^{n-1}(x) \in \mathcal{A}$.
- Such a σ is easy to find in $\mathbb{F}[x]/(x^2 - a)$, eg. $x \mapsto -x$ works. But in other cases is a very nontrivial question.

Either Factor or Find an Automorphism

- Input: A polynomial $f(x)$, over a finite field $\mathbb{F}$, of degree n.
- Output: *Either* we find a nontrivial factor of $f(x)$ *or* a nontrivial automorphism σ, of $\mathcal{A} = \mathbb{F}[x]/(f(x))$, of order n.
- Complexity: Deterministic $\mathrm{poly}(\log|\mathbb{F}|, n^{\log n})$ time.
- In a sense we do find all the roots of $f(X)$. But they live in $\mathcal{A}$, namely, $x, \sigma(x), \ldots, \sigma^{n-1}(x) \in \mathcal{A}$.
- Such a σ is easy to find in $\mathbb{F}[x]/(x^2 - a)$, eg. $x \mapsto -x$ works. But in other cases is a very nontrivial question.

Either Factor or Find an Automorphism

- Input: A polynomial $f(x)$, over a finite field $\mathbb{F}$, of degree n.
- Output: *Either* we find a nontrivial factor of $f(x)$ *or* a nontrivial automorphism σ, of $\mathcal{A} = \mathbb{F}[x]/(f(x))$, of order n.
- Complexity: Deterministic $\mathrm{poly}(\log|\mathbb{F}|, n^{\log n})$ time.
- In a sense we do find all the roots of $f(X)$. But they live in $\mathcal{A}$, namely, $x, \sigma(x), \ldots, \sigma^{n-1}(x) \in \mathcal{A}$.
- Such a σ is easy to find in $\mathbb{F}[x]/(x^2 - a)$, eg. $x \mapsto -x$ works. But in other cases is a very nontrivial question.

Either Factor or Find an Automorphism

- Input: A polynomial $f(x)$, over a finite field $\mathbb{F}$, of degree n.
- Output: *Either* we find a nontrivial factor of $f(x)$ *or* a nontrivial automorphism σ, of $\mathcal{A} = \mathbb{F}[x]/(f(x))$, of order n.
- Complexity: Deterministic $\mathrm{poly}(\log|\mathbb{F}|, n^{\log n})$ time.
- In a sense we do find all the roots of $f(X)$. But they live in $\mathcal{A}$, namely, $x, \sigma(x), \ldots, \sigma^{n-1}(x) \in \mathcal{A}$.
- Such a σ is easy to find in $\mathbb{F}[x]/(x^2 - a)$, eg. $x \mapsto -x$ works. But in other cases is a very nontrivial question.

Application to Algebras

- As a *direct* application we have the following algorithm.
- Input: Given a commutative semisimple algebra $\mathcal{A}$, over a finite field $\mathbb{F}$.
- Output: We can find a decomposition, $\mathcal{A} = \mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_t$, with an automorphism of $\mathcal{A}_i$ of order $\dim_{\mathbb{F}} \mathcal{A}_i$.
- Complexity: Deterministic quasipolynomial time.

Application to Algebras

- As a *direct* application we have the following algorithm.
- Input: Given a commutative semisimple algebra $\mathcal{A}$, over a finite field $\mathbb{F}$.
- Output: We can find a decomposition, $\mathcal{A} = \mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_t$, with an automorphism of $\mathcal{A}_i$ of order $\dim_{\mathbb{F}} \mathcal{A}_i$.
- Complexity: Deterministic quasipolynomial time.

Application to Algebras

- As a *direct* application we have the following algorithm.
- Input: Given a commutative semisimple algebra $\mathcal{A}$, over a finite field $\mathbb{F}$.
- Output: We can find a decomposition, $\mathcal{A} = \mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_t$, with an automorphism of $\mathcal{A}_i$ of order $\dim_{\mathbb{F}} \mathcal{A}_i$.
- Complexity: Deterministic quasipolynomial time.

Application to Algebras

- As a *direct* application we have the following algorithm.
- Input: Given a commutative semisimple algebra $\mathcal{A}$, over a finite field $\mathbb{F}$.
- Output: We can find a decomposition, $\mathcal{A} = \mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_t$, with an automorphism of $\mathcal{A}_i$ of order $\dim_{\mathbb{F}} \mathcal{A}_i$.
- Complexity: Deterministic quasipolynomial time.

Application to Cyclotomic Polynomials

- Our methods can be used to actually factor certain polynomials.
- Let $\Phi_m(x)$ be the m-th cyclotomic polynomial.
- Examples: $\Phi_1(x) = (x - 1)$, $\Phi_2(x) = (x^2 - 1)/\Phi_1(x)$, $\Phi_3(x) = (x^3 - 1)/\Phi_1(x)$, $\Phi_4(x) = (x^4 - 1)/\Phi_1(x)\Phi_2(x)$,...
- We can factor $\Phi_m(x)$ over $\mathbb{F}$ in deterministic polynomial time, if $\mathbb{Z}_m^*$ is noncyclic.
- I.e. When $m \notin \{1, 2, 4, p^i, 2p^i\}$, we can find a nontrivial factor of $\Phi_m(x)$ over a finite field $\mathbb{F}$.

Application to Cyclotomic Polynomials

- Our methods can be used to actually factor certain polynomials.
- Let $\Phi_m(x)$ be the m-th cyclotomic polynomial.
- Examples: $\Phi_1(x) = (x-1)$, $\Phi_2(x) = (x^2-1)/\Phi_1(x)$, $\Phi_3(x) = (x^3-1)/\Phi_1(x)$, $\Phi_4(x) = (x^4-1)/\Phi_1(x)\Phi_2(x)$,...
- We can factor $\Phi_m(x)$ over $\mathbb{F}$ in deterministic polynomial time, if $\mathbb{Z}_m^*$ is noncyclic.
- I.e. When $m \notin \{1, 2, 4, p^i, 2p^i\}$, we can find a nontrivial factor of $\Phi_m(x)$ over a finite field $\mathbb{F}$.

Application to Cyclotomic Polynomials

- Our methods can be used to actually factor certain polynomials.
- Let $\Phi_m(x)$ be the m-th cyclotomic polynomial.
- Examples: $\Phi_1(x) = (x-1)$, $\Phi_2(x) = (x^2-1)/\Phi_1(x)$, $\Phi_3(x) = (x^3-1)/\Phi_1(x)$, $\Phi_4(x) = (x^4-1)/\Phi_1(x)\Phi_2(x)$,...
- We can factor $\Phi_m(x)$ over $\mathbb{F}$ in deterministic polynomial time, if $\mathbb{Z}_m^*$ is noncyclic.
- I.e. When $m \notin \{1, 2, 4, p^i, 2p^i\}$, we can find a nontrivial factor of $\Phi_m(x)$ over a finite field $\mathbb{F}$.

Application to Cyclotomic Polynomials

- Our methods can be used to actually factor certain polynomials.
- Let $\Phi_m(x)$ be the m-th cyclotomic polynomial.
- Examples: $\Phi_1(x) = (x-1)$, $\Phi_2(x) = (x^2-1)/\Phi_1(x)$, $\Phi_3(x) = (x^3-1)/\Phi_1(x)$, $\Phi_4(x) = (x^4-1)/\Phi_1(x)\Phi_2(x)$,...
- We can factor $\Phi_m(x)$ over $\mathbb{F}$ in deterministic polynomial time, if $\mathbb{Z}_m^*$ is noncyclic.
- I.e. When $m \notin \{1, 2, 4, p^i, 2p^i\}$, we can find a nontrivial factor of $\Phi_m(x)$ over a finite field $\mathbb{F}$.

Application to Cyclotomic Polynomials

- Our methods can be used to actually factor certain polynomials.
- Let $\Phi_m(x)$ be the m-th cyclotomic polynomial.
- Examples: $\Phi_1(x) = (x-1)$, $\Phi_2(x) = (x^2-1)/\Phi_1(x)$, $\Phi_3(x) = (x^3-1)/\Phi_1(x)$, $\Phi_4(x) = (x^4-1)/\Phi_1(x)\Phi_2(x)$,...
- We can factor $\Phi_m(x)$ over $\mathbb{F}$ in deterministic polynomial time, if $\mathbb{Z}_m^*$ is noncyclic.
- I.e. When $m \notin \{1, 2, 4, p^i, 2p^i\}$, we can find a nontrivial factor of $\Phi_m(x)$ over a finite field $\mathbb{F}$.

Other Applications

- Our methods also "eliminate" GRH from other known results.

Using Galois Group

Let $f(x)$ have a *Galois group* over $\mathbb{Q}$ of size m. Then we can either factor $f(x) \pmod{p}$ *or* find an automorphism of $\mathbb{F}_p[x]/(f(x))$ of order $\deg f$, in $\mathrm{poly}(m, \log p)$ time.

Using Special Fields

Let $f(x)$ be a polynomial of degree n with that many roots in $\mathbb{F}_p$. Let r be the *largest* prime factor of $(p-1)$. Then we can either factor $f(x) \pmod{p}$ *or* find an automorphism of $\mathbb{F}_p[x]/(f(x))$ of order n, in $\mathrm{poly}(r, n, \log p)$ time.

Other Applications

- Our methods also "eliminate" GRH from other known results.

Using Galois Group

Let $f(x)$ have a *Galois group* over $\mathbb{Q}$ of size m. Then we can either factor $f(x) \pmod{p}$ *or* find an automorphism of $\mathbb{F}_p[x]/(f(x))$ of order $\deg f$, in $\mathrm{poly}(m, \log p)$ time.

Using Special Fields

Let $f(x)$ be a polynomial of degree n with that many roots in $\mathbb{F}_p$. Let r be the *largest* prime factor of $(p-1)$. Then we can either factor $f(x) \pmod{p}$ *or* find an automorphism of $\mathbb{F}_p[x]/(f(x))$ of order n, in $\mathrm{poly}(r, n, \log p)$ time.

Other Applications

- Our methods also "eliminate" GRH from other known results.

Using Galois Group

Let $f(x)$ have a *Galois group* over $\mathbb{Q}$ of size m. Then we can either factor $f(x) \pmod p$ *or* find an automorphism of $\mathbb{F}_p[x]/(f(x))$ of order $\deg f$, in $\mathrm{poly}(m, \log p)$ time.

Using Special Fields

Let $f(x)$ be a polynomial of degree n with that many roots in $\mathbb{F}_p$. Let r be the *largest* prime factor of $(p-1)$. Then we can either factor $f(x) \pmod p$ *or* find an automorphism of $\mathbb{F}_p[x]/(f(x))$ of order n, in $\mathrm{poly}(r, n, \log p)$ time.

Outline

Semiregular Automorphisms

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and subgroup $G \leq \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- We call G semiregular if none of its elements *fixes* a nontrivial ideal of $\mathcal{A}$.
- We call a $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$ semiregular if $\langle \sigma \rangle$ is semiregular.
- Example: Let $\mathcal{A} = \mathbb{F}_p \oplus \mathbb{F}_p \oplus \mathbb{F}_{p^2} \oplus \mathbb{F}_{p^2}$. It has an automorphism σ that swaps the two $\mathbb{F}_p$ components, and also the two $\mathbb{F}_{p^2}$ components. Then $G = \{1, \sigma\}$ is a semiregular group of automorphisms of $\mathcal{A}$.

Semiregular Automorphisms

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and subgroup $G \leq \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- We call G semiregular if none of its elements *fixes* a nontrivial ideal of $\mathcal{A}$.
- We call a $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$ semiregular if $\langle \sigma \rangle$ is semiregular.
- Example: Let $\mathcal{A} = \mathbb{F}_p \oplus \mathbb{F}_p \oplus \mathbb{F}_{p^2} \oplus \mathbb{F}_{p^2}$. It has an automorphism σ that swaps the two $\mathbb{F}_p$ components, and also the two $\mathbb{F}_{p^2}$ components. Then $G = \{1, \sigma\}$ is a semiregular group of automorphisms of $\mathcal{A}$.

Semiregular Automorphisms

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and subgroup $G \leq \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- We call G semiregular if none of its elements *fixes* a nontrivial ideal of $\mathcal{A}$.
- We call a $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$ semiregular if $\langle \sigma \rangle$ is semiregular.
- Example: Let $\mathcal{A} = \mathbb{F}_p \oplus \mathbb{F}_p \oplus \mathbb{F}_{p^2} \oplus \mathbb{F}_{p^2}$. It has an automorphism σ that swaps the two $\mathbb{F}_p$ components, and also the two $\mathbb{F}_{p^2}$ components. Then $G = \{1, \sigma\}$ is a semiregular group of automorphisms of $\mathcal{A}$.

Semiregular Automorphisms

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and subgroup $G \leq \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- We call G semiregular if none of its elements *fixes* a nontrivial ideal of $\mathcal{A}$.
- We call a $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$ semiregular if $\langle \sigma \rangle$ is semiregular.
- Example: Let $\mathcal{A} = \mathbb{F}_p \oplus \mathbb{F}_p \oplus \mathbb{F}_{p^2} \oplus \mathbb{F}_{p^2}$. It has an automorphism σ that swaps the two $\mathbb{F}_p$ components, and also the two $\mathbb{F}_{p^2}$ components. Then $G = \{1, \sigma\}$ is a semiregular group of automorphisms of $\mathcal{A}$.

Semiregular Automorphisms

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and subgroup $G \leq \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- We call G semiregular if none of its elements *fixes* a nontrivial ideal of $\mathcal{A}$.
- We call a $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$ semiregular if $\langle \sigma \rangle$ is semiregular.
- Example: Let $\mathcal{A} = \mathbb{F}_p \oplus \mathbb{F}_p \oplus \mathbb{F}_{p^2} \oplus \mathbb{F}_{p^2}$. It has an automorphism σ that swaps the two $\mathbb{F}_p$ components, and also the two $\mathbb{F}_{p^2}$ components. Then $G = \{1, \sigma\}$ is a semiregular group of automorphisms of $\mathcal{A}$.

Semiregular Automorphisms (Contd.)

- Let G be a subgroup of $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$. We denote by $\mathcal{A}_G$ the elements of $\mathcal{A}$ fixed by G.
- Theorem: G is semiregular iff $\mathcal{A}$ is a free $\mathcal{A}_G$-module of rank $|G|$.
- It can be seen as a generalized *Galois extension*.
- If G is not semiregular then while trying to find a free basis of $\mathcal{A}$ over $\mathcal{A}_G$ *we will discover a zero divisor of* $\mathcal{A}$.
- Thus, in this work we can always assume that an automorphism at hand is semiregular.

Semiregular Automorphisms (Contd.)

- Let G be a subgroup of $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$. We denote by $\mathcal{A}_G$ the elements of $\mathcal{A}$ fixed by G.
- Theorem: G is semiregular iff $\mathcal{A}$ is a free $\mathcal{A}_G$-module of rank $|G|$.
- It can be seen as a generalized *Galois extension*.
- If G is not semiregular then while trying to find a free basis of $\mathcal{A}$ over $\mathcal{A}_G$ *we will discover a zero divisor of* $\mathcal{A}$.
- Thus, in this work we can always assume that an automorphism at hand is semiregular.

Semiregular Automorphisms (Contd.)

- Let G be a subgroup of $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$. We denote by $\mathcal{A}_G$ the elements of $\mathcal{A}$ fixed by G.
- Theorem: G is semiregular iff $\mathcal{A}$ is a free $\mathcal{A}_G$-module of rank $|G|$.
- It can be seen as a generalized *Galois extension*.
- If G is not semiregular then while trying to find a free basis of $\mathcal{A}$ over $\mathcal{A}_G$ *we will discover a zero divisor of* $\mathcal{A}$.
- Thus, in this work we can always assume that an automorphism at hand is semiregular.

Semiregular Automorphisms (Contd.)

- Let G be a subgroup of $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$. We denote by $\mathcal{A}_G$ the elements of $\mathcal{A}$ fixed by G.
- Theorem: G is semiregular iff $\mathcal{A}$ is a free $\mathcal{A}_G$-module of rank $|G|$.
- It can be seen as a generalized *Galois extension*.
- If G is not semiregular then while trying to find a free basis of $\mathcal{A}$ over $\mathcal{A}_G$ *we will discover a zero divisor of* $\mathcal{A}$.
- Thus, in this work we can always assume that an automorphism at hand is semiregular.

Semiregular Automorphisms (Contd.)

- Let G be a subgroup of $\mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$. We denote by $\mathcal{A}_G$ the elements of $\mathcal{A}$ fixed by G.
- Theorem: G is semiregular iff $\mathcal{A}$ is a free $\mathcal{A}_G$-module of rank $|G|$.
- It can be seen as a generalized *Galois extension*.
- If G is not semiregular then while trying to find a free basis of $\mathcal{A}$ over $\mathcal{A}_G$ *we will discover a zero divisor of* $\mathcal{A}$.
- Thus, in this work we can always assume that an automorphism at hand is semiregular.

Outline

Reminder: Classical Lagrange resolvent

- As we know: *a cubic polynomial is solvable by radicals.*
- Lagrange (1736-1813) gave an elegant formula by reducing cubic to a *quadratic*.
- Say α, β, γ are roots of a cubic $f(x)$ (in $\mathbb{C}$).
- Say ω is a primitive 3-rd root of unity.
- Lagrange considered the combinations: $r_1 := (\alpha + \omega\beta + \omega^2\gamma)$ and $r_2 := (\alpha + \omega^2\beta + \omega\gamma)$.
- These are Lagrange resolvents.
- Note: $\sigma(r_1) = \omega r_1$ where σ *permutes* the roots.

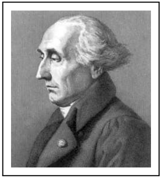

Fig: Lagrange

Reminder: Classical Lagrange resolvent

- As we know: *a cubic polynomial is solvable by radicals*.
- Lagrange (1736-1813) gave an elegant formula by reducing cubic to a *quadratic*.
- Say α, β, γ are roots of a cubic $f(x)$ (in $\mathbb{C}$).
- Say ω is a primitive 3-rd root of unity.
- Lagrange considered the combinations: $r_1 := (\alpha + \omega\beta + \omega^2\gamma)$ and $r_2 := (\alpha + \omega^2\beta + \omega\gamma)$.
- These are Lagrange resolvents.
- Note: $\sigma(r_1) = \omega r_1$ where σ *permutes* the roots.

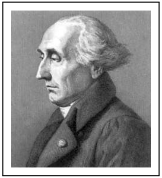

Fig: Lagrange

Reminder: Classical Lagrange resolvent

- As we know: *a cubic polynomial is solvable by radicals*.
- Lagrange (1736-1813) gave an elegant formula by reducing cubic to a *quadratic*.
- Say α, β, γ are roots of a cubic $f(x)$ (in $\mathbb{C}$).
- Say ω is a primitive 3-rd root of unity.
- Lagrange considered the combinations: $r_1 := (\alpha + \omega\beta + \omega^2\gamma)$ and $r_2 := (\alpha + \omega^2\beta + \omega\gamma)$.
- These are Lagrange resolvents.
- Note: $\sigma(r_1) = \omega r_1$ where σ *permutes* the roots.

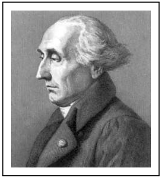

Fig: Lagrange

Reminder: Classical Lagrange resolvent

- As we know: *a cubic polynomial is solvable by radicals*.
- Lagrange (1736-1813) gave an elegant formula by reducing cubic to a *quadratic*.
- Say α, β, γ are roots of a cubic $f(x)$ (in $\mathbb{C}$).
- Say ω is a primitive 3-rd root of unity.
- Lagrange considered the combinations: $r_1 := (\alpha + \omega\beta + \omega^2\gamma)$ and $r_2 := (\alpha + \omega^2\beta + \omega\gamma)$.
- These are Lagrange resolvents.
- Note: $\sigma(r_1) = \omega r_1$ where σ *permutes* the roots.

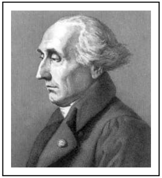

Fig: Lagrange

Reminder: Classical Lagrange resolvent

- As we know: *a cubic polynomial is solvable by radicals*.
- Lagrange (1736-1813) gave an elegant formula by reducing cubic to a *quadratic*.
- Say α, β, γ are roots of a cubic $f(x)$ (in $\mathbb{C}$).
- Say ω is a primitive 3-rd root of unity.
- Lagrange considered the combinations: $r_1 := (\alpha + \omega\beta + \omega^2\gamma)$ and $r_2 := (\alpha + \omega^2\beta + \omega\gamma)$.
- These are Lagrange resolvents.
- Note: $\sigma(r_1) = \omega r_1$ where σ *permutes* the roots.

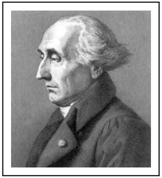

Fig: Lagrange

Reminder: Classical Lagrange resolvent

- As we know: *a cubic polynomial is solvable by radicals*.
- Lagrange (1736-1813) gave an elegant formula by reducing cubic to a *quadratic*.
- Say α, β, γ are roots of a cubic $f(x)$ (in $\mathbb{C}$).
- Say ω is a primitive 3-rd root of unity.
- Lagrange considered the combinations: $r_1 := (\alpha + \omega\beta + \omega^2\gamma)$ and $r_2 := (\alpha + \omega^2\beta + \omega\gamma)$.
- These are Lagrange resolvents.
- Note: $\sigma(r_1) = \omega r_1$ where σ *permutes* the roots.

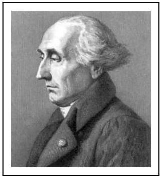

Fig: Lagrange

Reminder: Classical Lagrange resolvent

- As we know: *a cubic polynomial is solvable by radicals*.
- Lagrange (1736-1813) gave an elegant formula by reducing cubic to a *quadratic*.
- Say α, β, γ are roots of a cubic $f(x)$ (in $\mathbb{C}$).
- Say ω is a primitive 3-rd root of unity.
- Lagrange considered the combinations:
  $r_1 := (\alpha + \omega\beta + \omega^2\gamma)$ and $r_2 := (\alpha + \omega^2\beta + \omega\gamma)$.
- These are Lagrange resolvents.
- Note: $\sigma(r_1) = \omega r_1$ where σ *permutes* the roots.

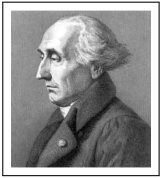

Fig: Lagrange

Computing (our) Lagrange Resolvent

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- Let σ be of prime order r and ζ be a primitive r-th root of unity in $\mathcal{A}_\sigma$.
- We call a nonzero element $x \in \mathcal{A}$ Lagrange resolvent, if $\sigma(x) = \zeta x$.
- Theorem: Given $\mathcal{A}$, σ and ζ, we can efficiently compute a Lagrange resolvent.
- *Proof idea*: We pick a $y \in \mathcal{A} \setminus \mathcal{A}_\sigma$. Consider $(y, \zeta^j) := \sum_{i=0}^{r-1} \zeta^{ij} \sigma^i(y)$.
- One of these (y, ζ^j) gives the Lagrange resolvent!

Computing (our) Lagrange Resolvent

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- Let σ be of prime order r and ζ be a primitive r-th root of unity in $\mathcal{A}_\sigma$.
- We call a nonzero element $x \in \mathcal{A}$ Lagrange resolvent, if $\sigma(x) = \zeta x$.
- Theorem: Given $\mathcal{A}$, σ and ζ, we can efficiently compute a Lagrange resolvent.
- *Proof idea*: We pick a $y \in \mathcal{A} \setminus \mathcal{A}_\sigma$. Consider $(y, \zeta^j) := \sum_{i=0}^{r-1} \zeta^{ij} \sigma^i(y)$.
- One of these (y, ζ^j) gives the Lagrange resolvent!

Computing (our) Lagrange Resolvent

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- Let σ be of prime order r and ζ be a primitive r-th root of unity in $\mathcal{A}_\sigma$.
- We call a nonzero element $x \in \mathcal{A}$ Lagrange resolvent, if $\sigma(x) = \zeta x$.
- Theorem: Given $\mathcal{A}$, σ and ζ, we can efficiently compute a Lagrange resolvent.
- *Proof idea*: We pick a $y \in \mathcal{A} \setminus \mathcal{A}_\sigma$. Consider $(y, \zeta^j) := \sum_{i=0}^{r-1} \zeta^{ij} \sigma^i(y)$.
- One of these (y, ζ^j) gives the Lagrange resolvent!

Computing (our) Lagrange Resolvent

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- Let σ be of prime order r and ζ be a primitive r-th root of unity in $\mathcal{A}_\sigma$.
- We call a nonzero element $x \in \mathcal{A}$ Lagrange resolvent, if $\sigma(x) = \zeta x$.
- Theorem: Given $\mathcal{A}$, σ and ζ, we can efficiently compute a Lagrange resolvent.
- *Proof idea*: We pick a $y \in \mathcal{A} \setminus \mathcal{A}_\sigma$. Consider $(y, \zeta^j) := \sum_{i=0}^{r-1} \zeta^{ij} \sigma^i(y)$.
- One of these (y, ζ^j) gives the Lagrange resolvent!

Computing (our) Lagrange Resolvent

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- Let σ be of prime order r and ζ be a primitive r-th root of unity in $\mathcal{A}_\sigma$.
- We call a nonzero element $x \in \mathcal{A}$ Lagrange resolvent, if $\sigma(x) = \zeta x$.
- Theorem: Given $\mathcal{A}$, σ and ζ, we can efficiently compute a Lagrange resolvent.
- *Proof idea*: We pick a $y \in \mathcal{A} \setminus \mathcal{A}_\sigma$. Consider $(y, \zeta^j) := \sum_{i=0}^{r-1} \zeta^{ij} \sigma^i(y)$.
- One of these (y, ζ^j) gives the Lagrange resolvent!

Computing (our) Lagrange Resolvent

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $\sigma \in \mathrm{Aut}_{\mathbb{F}}(\mathcal{A})$.
- Let σ be of prime order r and ζ be a primitive r-th root of unity in $\mathcal{A}_\sigma$.
- We call a nonzero element $x \in \mathcal{A}$ Lagrange resolvent, if $\sigma(x) = \zeta x$.
- Theorem: Given $\mathcal{A}$, σ and ζ, we can efficiently compute a Lagrange resolvent.
- *Proof idea*: We pick a $y \in \mathcal{A} \setminus \mathcal{A}_\sigma$. Consider $(y, \zeta^j) := \sum_{i=0}^{r-1} \zeta^{ij} \sigma^i(y)$.
- One of these (y, ζ^j) gives the Lagrange resolvent!

Outline

Reminder: Classical Kummer Extension

- Kummer (1810-1893) developed them while studying *Fermat's last "theorem"*.
- A field extension $K \subset L$ is called Kummer extension if :
- K has an r-th primitive root of unity, and
- G_L is *abelian* of size r.
- For example, $K[\zeta_r][\sqrt[r]{c}]$ over K (where $c \in K[\zeta_r]$ but $\sqrt[r]{c} \notin K[\zeta_r]$).

Fig: Kummer

Reminder: Classical Kummer Extension

- Kummer (1810-1893) developed them while studying *Fermat's last "theorem"*.
- A field extension $K \subset L$ is called Kummer extension if :
- K has an r-th primitive root of unity, and
- G_L is *abelian* of size r.
- For example, $K[\zeta_r][\sqrt[r]{c}]$ over K (where $c \in K[\zeta_r]$ but $\sqrt[r]{c} \notin K[\zeta_r]$).

Fig: Kummer

Reminder: Classical Kummer Extension

- Kummer (1810-1893) developed them while studying *Fermat's last "theorem"*.
- A field extension $K \subset L$ is called Kummer extension if :
- K has an r-th primitive root of unity, and
- G_L is *abelian* of size r.
- For example, $K[\zeta_r][\sqrt[r]{c}]$ over K (where $c \in K[\zeta_r]$ but $\sqrt[r]{c} \notin K[\zeta_r]$).

Fig: Kummer

Reminder: Classical Kummer Extension

- Kummer (1810-1893) developed them while studying *Fermat's last "theorem"*.
- A field extension $K \subset L$ is called Kummer extension if :
- K has an r-th primitive root of unity, and
- G_L is *abelian* of size r.
- For example, $K[\zeta_r][\sqrt[r]{c}]$ over K (where $c \in K[\zeta_r]$ but $\sqrt[r]{c} \notin K[\zeta_r]$).

Fig: Kummer

Reminder: Classical Kummer Extension

- Kummer (1810-1893) developed them while studying *Fermat's last "theorem"*.
- A field extension $K \subset L$ is called Kummer extension if :
- K has an r-th primitive root of unity, and
- G_L is *abelian* of size r.
- For example, $K[\zeta_r][\sqrt[r]{c}]$ over K (where $c \in K[\zeta_r]$ but $\sqrt[r]{c} \notin K[\zeta_r]$).

Fig: Kummer

Cyclotomic Extension

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and r be a prime.
- The r-th cyclotomic extension is simply $\mathcal{A}[X]/(\sum_{i=0}^{r-1} X^i)$, denoted by $\mathcal{A}[\zeta_r]$.
- $\mathcal{A}[\zeta_r]$ is also a semisimple $\mathbb{F}$-algebra. If $\mathcal{A} \cong \mathcal{A}_1 \oplus \mathcal{A}_2$ then $\mathcal{A}[\zeta_r] \cong \mathcal{A}_1[\zeta_r] \oplus \mathcal{A}_2[\zeta_r]$.
- For $a \in \mathbb{Z}_r^*$ the map $\rho_a : \zeta_r \mapsto \zeta_r^a$ is an automorphism of $\mathcal{A}[\zeta_r]$.
- The set of these ρ_a-s is a group of automorphisms, denoted by Δ_r.

Cyclotomic Extension

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and r be a prime.
- The r-th cyclotomic extension is simply $\mathcal{A}[X]/(\sum_{i=0}^{r-1} X^i)$, denoted by $\mathcal{A}[\zeta_r]$.
- $\mathcal{A}[\zeta_r]$ is also a semisimple $\mathbb{F}$-algebra. If $\mathcal{A} \cong \mathcal{A}_1 \oplus \mathcal{A}_2$ then $\mathcal{A}[\zeta_r] \cong \mathcal{A}_1[\zeta_r] \oplus \mathcal{A}_2[\zeta_r]$.
- For $a \in \mathbb{Z}_r^*$ the map $\rho_a : \zeta_r \mapsto \zeta_r^a$ is an automorphism of $\mathcal{A}[\zeta_r]$.
- The set of these ρ_a-s is a group of automorphisms, denoted by Δ_r.

Cyclotomic Extension

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and r be a prime.
- The r-th cyclotomic extension is simply $\mathcal{A}[X]/(\sum_{i=0}^{r-1} X^i)$, denoted by $\mathcal{A}[\zeta_r]$.
- $\mathcal{A}[\zeta_r]$ is also a semisimple $\mathbb{F}$-algebra. If $\mathcal{A} \cong \mathcal{A}_1 \oplus \mathcal{A}_2$ then $\mathcal{A}[\zeta_r] \cong \mathcal{A}_1[\zeta_r] \oplus \mathcal{A}_2[\zeta_r]$.
- For $a \in \mathbb{Z}_r^*$ the map $\rho_a : \zeta_r \mapsto \zeta_r^a$ is an automorphism of $\mathcal{A}[\zeta_r]$.
- The set of these ρ_a-s is a group of automorphisms, denoted by Δ_r.

Cyclotomic Extension

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and r be a prime.
- The r-th cyclotomic extension is simply $\mathcal{A}[X]/(\sum_{i=0}^{r-1} X^i)$, denoted by $\mathcal{A}[\zeta_r]$.
- $\mathcal{A}[\zeta_r]$ is also a semisimple $\mathbb{F}$-algebra. If $\mathcal{A} \cong \mathcal{A}_1 \oplus \mathcal{A}_2$ then $\mathcal{A}[\zeta_r] \cong \mathcal{A}_1[\zeta_r] \oplus \mathcal{A}_2[\zeta_r]$.
- For $a \in \mathbb{Z}_r^*$ the map $\rho_a : \zeta_r \mapsto \zeta_r^a$ is an automorphism of $\mathcal{A}[\zeta_r]$.
- The set of these ρ_a-s is a group of automorphisms, denoted by Δ_r.

Cyclotomic Extension

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and r be a prime.
- The r-th cyclotomic extension is simply $\mathcal{A}[X]/(\sum_{i=0}^{r-1} X^i)$, denoted by $\mathcal{A}[\zeta_r]$.
- $\mathcal{A}[\zeta_r]$ is also a semisimple $\mathbb{F}$-algebra. If $\mathcal{A} \cong \mathcal{A}_1 \oplus \mathcal{A}_2$ then $\mathcal{A}[\zeta_r] \cong \mathcal{A}_1[\zeta_r] \oplus \mathcal{A}_2[\zeta_r]$.
- For $a \in \mathbb{Z}_r^*$ the map $\rho_a : \zeta_r \mapsto \zeta_r^a$ is an automorphism of $\mathcal{A}[\zeta_r]$.
- The set of these ρ_a-s is a group of automorphisms, denoted by Δ_r.

Cyclotomic Extension

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and r be a prime.
- The r-th cyclotomic extension is simply $\mathcal{A}[X]/(\sum_{i=0}^{r-1} X^i)$, denoted by $\mathcal{A}[\zeta_r]$.
- $\mathcal{A}[\zeta_r]$ is also a semisimple $\mathbb{F}$-algebra. If $\mathcal{A} \cong \mathcal{A}_1 \oplus \mathcal{A}_2$ then $\mathcal{A}[\zeta_r] \cong \mathcal{A}_1[\zeta_r] \oplus \mathcal{A}_2[\zeta_r]$.
- For $a \in \mathbb{Z}_r^*$ the map $\rho_a : \zeta_r \mapsto \zeta_r^a$ is an automorphism of $\mathcal{A}[\zeta_r]$.
- The set of these ρ_a-s is a group of automorphisms, denoted by Δ_r.

Teichmüller Subgroup

- First group: Consider the subgroup $\mathcal{A}[\zeta_r]^*_r$ of units, in $\mathcal{A}[\zeta_r]$, whose order are powers of r. (r-Sylow)
- Its automorphism: Consider the map $\omega_a : x \mapsto x^{a^{r^u}}$ where $\mathrm{ord}(x) = r^u$.
- Second group: $T_{\mathcal{A},r} := \{x \in \mathcal{A}[\zeta_r]^*_r \mid \rho_a(x) = \omega_a(x) \text{ for every } \rho_a \in \Delta_r\}$.
- This generalizes the classical Teichmüller subgroup. It is a subgroup on which Δ_r-action is "well behaved".
- Note: Since $\rho_a(\zeta_r) = \zeta_r^a = \zeta_r^{a^r} = \omega_a(\zeta_r)$, thus $\zeta_r \in T_{\mathcal{A},r}$.
- Bottomline: $T_{\mathcal{A},r}$ is a "nice" subgroup, of units of $\mathcal{A}[\zeta_r]$, of size an r-power.

Teichmüller Subgroup

- First group: Consider the subgroup $\mathcal{A}[\zeta_r]_r^*$ of units, in $\mathcal{A}[\zeta_r]$, whose order are powers of r. (r-Sylow)
- Its automorphism: Consider the map $\omega_a : x \mapsto x^{a^{r^u}}$ where $\text{ord}(x) = r^u$.
- Second group:
  $T_{\mathcal{A},r} := \{x \in \mathcal{A}[\zeta_r]_r^* \mid \rho_a(x) = \omega_a(x) \text{ for every } \rho_a \in \Delta_r\}$.
- This generalizes the classical Teichmüller subgroup. It is a subgroup on which Δ_r-action is "well behaved".
- Note: Since $\rho_a(\zeta_r) = \zeta_r^a = \zeta_r^{a^r} = \omega_a(\zeta_r)$, thus $\zeta_r \in T_{\mathcal{A},r}$.
- Bottomline: $T_{\mathcal{A},r}$ is a "nice" subgroup, of units of $\mathcal{A}[\zeta_r]$, of size an r-power.

Teichmüller Subgroup

- First group: Consider the subgroup $\mathcal{A}[\zeta_r]_r^*$ of units, in $\mathcal{A}[\zeta_r]$, whose order are powers of r. (r-Sylow)
- Its automorphism: Consider the map $\omega_a : x \mapsto x^{a^{r^u}}$ where $\mathrm{ord}(x) = r^u$.
- Second group: $T_{\mathcal{A},r} := \{x \in \mathcal{A}[\zeta_r]_r^* \mid \rho_a(x) = \omega_a(x) \text{ for every } \rho_a \in \Delta_r\}$.
- This generalizes the classical Teichmüller subgroup. It is a subgroup on which Δ_r-action is "well behaved".
- Note: Since $\rho_a(\zeta_r) = \zeta_r^a = \zeta_r^{a^r} = \omega_a(\zeta_r)$, thus $\zeta_r \in T_{\mathcal{A},r}$.
- Bottomline: $T_{\mathcal{A},r}$ is a "nice" subgroup, of units of $\mathcal{A}[\zeta_r]$, of size an r-power.

Teichmüller Subgroup

- First group: Consider the subgroup $\mathcal{A}[\zeta_r]_r^*$ of units, in $\mathcal{A}[\zeta_r]$, whose order are powers of r. (r-Sylow)
- Its automorphism: Consider the map $\omega_a : x \mapsto x^{a^{r^u}}$ where $\text{ord}(x) = r^u$.
- Second group: $T_{\mathcal{A},r} := \{x \in \mathcal{A}[\zeta_r]_r^* \mid \rho_a(x) = \omega_a(x) \text{ for every } \rho_a \in \Delta_r\}$.
- This generalizes the classical Teichmüller subgroup. It is a subgroup on which Δ_r-action is "well behaved".
- Note: Since $\rho_a(\zeta_r) = \zeta_r^a = \zeta_r^{a^r} = \omega_a(\zeta_r)$, thus $\zeta_r \in T_{\mathcal{A},r}$.
- Bottomline: $T_{\mathcal{A},r}$ is a "nice" subgroup, of units of $\mathcal{A}[\zeta_r]$, of size an r-power.

Teichmüller Subgroup

- First group: Consider the subgroup $\mathcal{A}[\zeta_r]_r^*$ of units, in $\mathcal{A}[\zeta_r]$, whose order are powers of r. (r-Sylow)
- Its automorphism: Consider the map $\omega_a : x \mapsto x^{a^{r^u}}$ where $\mathrm{ord}(x) = r^u$.
- Second group:
$T_{\mathcal{A},r} := \{x \in \mathcal{A}[\zeta_r]_r^* \mid \rho_a(x) = \omega_a(x) \text{ for every } \rho_a \in \Delta_r\}$.
- This generalizes the classical Teichmüller subgroup. It is a subgroup on which Δ_r-action is "well behaved".
- Note: Since $\rho_a(\zeta_r) = \zeta_r^a = \zeta_r^{a^r} = \omega_a(\zeta_r)$, thus $\zeta_r \in T_{\mathcal{A},r}$.
- Bottomline: $T_{\mathcal{A},r}$ is a "nice" subgroup, of units of $\mathcal{A}[\zeta_r]$, of size an r-power.

Teichmüller Subgroup

- First group: Consider the subgroup $\mathcal{A}[\zeta_r]_r^*$ of units, in $\mathcal{A}[\zeta_r]$, whose order are powers of r. (r-Sylow)
- Its automorphism: Consider the map $\omega_a : x \mapsto x^{a^{r^u}}$ where $\mathrm{ord}(x) = r^u$.
- Second group:
  $T_{\mathcal{A},r} := \{x \in \mathcal{A}[\zeta_r]_r^* \mid \rho_a(x) = \omega_a(x) \text{ for every } \rho_a \in \Delta_r\}$.
- This generalizes the classical Teichmüller subgroup. It is a subgroup on which Δ_r-action is "well behaved".
- Note: Since $\rho_a(\zeta_r) = \zeta_r^a = \zeta_r^{a^r} = \omega_a(\zeta_r)$, thus $\zeta_r \in T_{\mathcal{A},r}$.
- Bottomline: $T_{\mathcal{A},r}$ is a "nice" subgroup, of units of $\mathcal{A}[\zeta_r]$, of size an r-power.

Kummer Extension of an Algebra

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $T_{\mathcal{A},r} \leq \mathcal{A}[\zeta_r]^*$ be the Teichmüller subgroup.
- For $c \in T_{\mathcal{A},r}$ consider the algebra $\mathcal{A}[\zeta_r][Y]/(Y^r - c)$.
- It generalizes the classical Kummer extension and is denoted by $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$.
- It is again a semisimple $\mathbb{F}$-algebra.
- The automorphism ρ_a of $\mathcal{A}[\zeta_r]$ beautifully extends to $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$

  via $\sqrt[r]{c} \mapsto \omega_a(\sqrt[r]{c}) = (\sqrt[r]{c})^{a^{r^u}}$ where $\mathrm{ord}(\sqrt[r]{c}) = r^u$.
- Bottomline: Automorphisms Δ_r of the cyclotomic extension *extend* to the Kummer extension.

Kummer Extension of an Algebra

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $T_{\mathcal{A},r} \leq \mathcal{A}[\zeta_r]^*$ be the Teichmüller subgroup.
- For $c \in T_{\mathcal{A},r}$ consider the algebra $\mathcal{A}[\zeta_r][Y]/(Y^r - c)$.
- It generalizes the classical Kummer extension and is denoted by $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$.
- It is again a semisimple $\mathbb{F}$-algebra.
- The automorphism ρ_a of $\mathcal{A}[\zeta_r]$ beautifully extends to $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$

  via $\sqrt[r]{c} \mapsto \omega_a(\sqrt[r]{c}) = (\sqrt[r]{c})^{a^{r^u}}$ where $\mathrm{ord}(\sqrt[r]{c}) = r^u$.
- Bottomline: Automorphisms Δ_r of the cyclotomic extension *extend* to the Kummer extension.

Kummer Extension of an Algebra

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $T_{\mathcal{A},r} \leq \mathcal{A}[\zeta_r]^*$ be the Teichmüller subgroup.
- For $c \in T_{\mathcal{A},r}$ consider the algebra $\mathcal{A}[\zeta_r][Y]/(Y^r - c)$.
- It generalizes the classical Kummer extension and is denoted by $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$.
- It is again a semisimple $\mathbb{F}$-algebra.
- The automorphism ρ_a of $\mathcal{A}[\zeta_r]$ beautifully extends to $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$ via $\sqrt[r]{c} \mapsto \omega_a(\sqrt[r]{c}) = (\sqrt[r]{c})^{a^{r^u}}$ where $\mathrm{ord}(\sqrt[r]{c}) = r^u$.
- Bottomline: Automorphisms Δ_r of the cyclotomic extension *extend* to the Kummer extension.

Kummer Extension of an Algebra

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $T_{\mathcal{A},r} \leq \mathcal{A}[\zeta_r]^*$ be the Teichmüller subgroup.
- For $c \in T_{\mathcal{A},r}$ consider the algebra $\mathcal{A}[\zeta_r][Y]/(Y^r - c)$.
- It generalizes the classical Kummer extension and is denoted by $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$.
- It is again a semisimple $\mathbb{F}$-algebra.
- The automorphism ρ_a of $\mathcal{A}[\zeta_r]$ beautifully extends to $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$

  via $\sqrt[r]{c} \mapsto \omega_a(\sqrt[r]{c}) = (\sqrt[r]{c})^{a^{r^u}}$ where $\mathrm{ord}(\sqrt[r]{c}) = r^u$.
- Bottomline: Automorphisms Δ_r of the cyclotomic extension *extend* to the Kummer extension.

Kummer Extension of an Algebra

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $T_{\mathcal{A},r} \leq \mathcal{A}[\zeta_r]^*$ be the Teichmüller subgroup.
- For $c \in T_{\mathcal{A},r}$ consider the algebra $\mathcal{A}[\zeta_r][Y]/(Y^r - c)$.
- It generalizes the classical Kummer extension and is denoted by $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$.
- It is again a semisimple $\mathbb{F}$-algebra.
- The automorphism ρ_a of $\mathcal{A}[\zeta_r]$ beautifully extends to $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$
  via $\sqrt[r]{c} \mapsto \omega_a(\sqrt[r]{c}) = (\sqrt[r]{c})^{a^{r^u}}$ where $\mathrm{ord}(\sqrt[r]{c}) = r^u$.
- Bottomline: Automorphisms Δ_r of the cyclotomic extension *extend* to the Kummer extension.

Kummer Extension of an Algebra

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $T_{\mathcal{A},r} \leq \mathcal{A}[\zeta_r]^*$ be the Teichmüller subgroup.
- For $c \in T_{\mathcal{A},r}$ consider the algebra $\mathcal{A}[\zeta_r][Y]/(Y^r - c)$.
- It generalizes the classical Kummer extension and is denoted by $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$.
- It is again a semisimple $\mathbb{F}$-algebra.
- The automorphism ρ_a of $\mathcal{A}[\zeta_r]$ beautifully extends to $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$ via $\sqrt[r]{c} \mapsto \omega_a(\sqrt[r]{c}) = (\sqrt[r]{c})^{a^{r^u}}$ where $\mathrm{ord}(\sqrt[r]{c}) = r^u$.
- Bottomline: Automorphisms Δ_r of the cyclotomic extension *extend* to the Kummer extension.

Kummer Extension of an Algebra

- Let $\mathcal{A}$ be a commutative semisimple $\mathbb{F}$-algebra and $T_{\mathcal{A},r} \leq \mathcal{A}[\zeta_r]^*$ be the Teichmüller subgroup.
- For $c \in T_{\mathcal{A},r}$ consider the algebra $\mathcal{A}[\zeta_r][Y]/(Y^r - c)$.
- It generalizes the classical Kummer extension and is denoted by $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$.
- It is again a semisimple $\mathbb{F}$-algebra.
- The automorphism ρ_a of $\mathcal{A}[\zeta_r]$ beautifully extends to $\mathcal{A}[\zeta_r][\sqrt[r]{c}]$
  via $\sqrt[r]{c} \mapsto \omega_a(\sqrt[r]{c}) = (\sqrt[r]{c})^{a^{r^u}}$ where $\mathrm{ord}(\sqrt[r]{c}) = r^u$.
- Bottomline: Automorphisms Δ_r of the cyclotomic extension *extend* to the Kummer extension.

Embedding $\mathcal{A}$ in a Kummer Extension of $\mathcal{A}_\sigma$

- Given a commutative semisimple $\mathbb{F}$-algebra $\mathcal{A}$ and a semiregular automorphism σ of *prime* order r.
- We could consider σ as also an automorphism of $\mathcal{A}[\zeta_r]$ by fixing ζ_r.
- We can efficiently find an $x \in T_{\mathcal{A},r}$ which is a Lagrange resolvent i.e. $\sigma(x) = \zeta_r x$. As before and a trick!
- Clearly $\sigma(x^r) = x^r$, which means that $c := x^r \in T_{\mathcal{A}_\sigma,r}$.
- Theorem: There is a *natural* isomorphism, $\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}] \cong \mathcal{A}_\sigma[\zeta_r][x] = \mathcal{A}[\zeta_r]$.
- Thus, we can efficiently compute a $c \in T_{\mathcal{A}_\sigma,r}$ s.t. $\mathcal{A} \cong (\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}])_{\Delta_r}$. Bottomline: canonical embedding of $\mathcal{A}$.

Embedding $\mathcal{A}$ in a Kummer Extension of $\mathcal{A}_\sigma$

- Given a commutative semisimple $\mathbb{F}$-algebra $\mathcal{A}$ and a semiregular automorphism σ of *prime* order r.
- We could consider σ as also an automorphism of $\mathcal{A}[\zeta_r]$ by fixing ζ_r.
- We can efficiently find an $x \in T_{\mathcal{A},r}$ which is a Lagrange resolvent i.e. $\sigma(x) = \zeta_r x$. As before and a trick!
- Clearly $\sigma(x^r) = x^r$, which means that $c := x^r \in T_{\mathcal{A}_\sigma,r}$.
- Theorem: There is a *natural* isomorphism, $\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}] \cong \mathcal{A}_\sigma[\zeta_r][x] = \mathcal{A}[\zeta_r]$.
- Thus, we can efficiently compute a $c \in T_{\mathcal{A}_\sigma,r}$ s.t. $\mathcal{A} \cong (\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}])_{\Delta_r}$. Bottomline: canonical embedding of $\mathcal{A}$.

Embedding $\mathcal{A}$ in a Kummer Extension of $\mathcal{A}_\sigma$

- Given a commutative semisimple $\mathbb{F}$-algebra $\mathcal{A}$ and a semiregular automorphism σ of *prime* order r.
- We could consider σ as also an automorphism of $\mathcal{A}[\zeta_r]$ by fixing ζ_r.
- We can efficiently find an $x \in T_{\mathcal{A},r}$ which is a Lagrange resolvent i.e. $\sigma(x) = \zeta_r x$. As before and a trick!
- Clearly $\sigma(x^r) = x^r$, which means that $c := x^r \in T_{\mathcal{A}_\sigma,r}$.
- Theorem: There is a *natural* isomorphism, $\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}] \cong \mathcal{A}_\sigma[\zeta_r][x] = \mathcal{A}[\zeta_r]$.
- Thus, we can efficiently compute a $c \in T_{\mathcal{A}_\sigma,r}$ s.t. $\mathcal{A} \cong (\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}])_{\Delta_r}$. Bottomline: canonical embedding of $\mathcal{A}$.

Embedding $\mathcal{A}$ in a Kummer Extension of $\mathcal{A}_\sigma$

- Given a commutative semisimple $\mathbb{F}$-algebra $\mathcal{A}$ and a semiregular automorphism σ of *prime* order r.
- We could consider σ as also an automorphism of $\mathcal{A}[\zeta_r]$ by fixing ζ_r.
- We can efficiently find an $x \in T_{\mathcal{A},r}$ which is a Lagrange resolvent i.e. $\sigma(x) = \zeta_r x$. As before and a trick!
- Clearly $\sigma(x^r) = x^r$, which means that $c := x^r \in T_{\mathcal{A}_\sigma,r}$.
- Theorem: There is a *natural* isomorphism, $\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}] \cong \mathcal{A}_\sigma[\zeta_r][x] = \mathcal{A}[\zeta_r]$.
- Thus, we can efficiently compute a $c \in T_{\mathcal{A}_\sigma,r}$ s.t. $\mathcal{A} \cong (\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}])_{\Delta_r}$. Bottomline: canonical embedding of $\mathcal{A}$.

Embedding $\mathcal{A}$ in a Kummer Extension of $\mathcal{A}_\sigma$

- Given a commutative semisimple $\mathbb{F}$-algebra $\mathcal{A}$ and a semiregular automorphism σ of *prime* order r.
- We could consider σ as also an automorphism of $\mathcal{A}[\zeta_r]$ by fixing ζ_r.
- We can efficiently find an $x \in T_{\mathcal{A},r}$ which is a Lagrange resolvent i.e. $\sigma(x) = \zeta_r x$. As before and a trick!
- Clearly $\sigma(x^r) = x^r$, which means that $c := x^r \in T_{\mathcal{A}_\sigma,r}$.
- Theorem: There is a *natural* isomorphism, $\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}] \cong \mathcal{A}_\sigma[\zeta_r][x] = \mathcal{A}[\zeta_r]$.
- Thus, we can efficiently compute a $c \in T_{\mathcal{A}_\sigma,r}$ s.t. $\mathcal{A} \cong (\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}])_{\Delta_r}$. Bottomline: canonical embedding of $\mathcal{A}$.

Embedding $\mathcal{A}$ in a Kummer Extension of $\mathcal{A}_\sigma$

- Given a commutative semisimple $\mathbb{F}$-algebra $\mathcal{A}$ and a semiregular automorphism σ of *prime* order r.
- We could consider σ as also an automorphism of $\mathcal{A}[\zeta_r]$ by fixing ζ_r.
- We can efficiently find an $x \in T_{\mathcal{A},r}$ which is a Lagrange resolvent i.e. $\sigma(x) = \zeta_r x$. As before and a trick!
- Clearly $\sigma(x^r) = x^r$, which means that $c := x^r \in T_{\mathcal{A}_\sigma,r}$.
- Theorem: There is a *natural* isomorphism, $\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}] \cong \mathcal{A}_\sigma[\zeta_r][x] = \mathcal{A}[\zeta_r]$.
- Thus, we can efficiently compute a $c \in T_{\mathcal{A}_\sigma,r}$ s.t. $\mathcal{A} \cong (\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}])_{\Delta_r}$. Bottomline: canonical embedding of $\mathcal{A}$.

Embedding $\mathcal{A}$ in a Kummer Extension of $\mathcal{A}_\sigma$

- Given a commutative semisimple $\mathbb{F}$-algebra $\mathcal{A}$ and a semiregular automorphism σ of *prime* order r.
- We could consider σ as also an automorphism of $\mathcal{A}[\zeta_r]$ by fixing ζ_r.
- We can efficiently find an $x \in T_{\mathcal{A},r}$ which is a Lagrange resolvent i.e. $\sigma(x) = \zeta_r x$. As before and a trick!
- Clearly $\sigma(x^r) = x^r$, which means that $c := x^r \in T_{\mathcal{A}_\sigma,r}$.
- Theorem: There is a *natural* isomorphism, $\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}] \cong \mathcal{A}_\sigma[\zeta_r][x] = \mathcal{A}[\zeta_r]$.
- Thus, we can efficiently compute a $c \in T_{\mathcal{A}_\sigma,r}$ s.t. $\mathcal{A} \cong (\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}])_{\Delta_r}$. Bottomline: canonical embedding of $\mathcal{A}$.

Embedding $\mathcal{A}$ in a Kummer Extension of $\mathcal{A}_\sigma$

- Given a commutative semisimple $\mathbb{F}$-algebra $\mathcal{A}$ and a semiregular automorphism σ of *prime* order r.
- We could consider σ as also an automorphism of $\mathcal{A}[\zeta_r]$ by fixing ζ_r.
- We can efficiently find an $x \in T_{\mathcal{A},r}$ which is a Lagrange resolvent i.e. $\sigma(x) = \zeta_r x$. As before and a trick!
- Clearly $\sigma(x^r) = x^r$, which means that $c := x^r \in T_{\mathcal{A}_\sigma,r}$.
- Theorem: There is a *natural* isomorphism, $\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}] \cong \mathcal{A}_\sigma[\zeta_r][x] = \mathcal{A}[\zeta_r]$.
- Thus, we can efficiently compute a $c \in T_{\mathcal{A}_\sigma,r}$ s.t. $\mathcal{A} \cong (\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}])_{\Delta_r}$. Bottomline: canonical embedding of $\mathcal{A}$.

Embedding $\mathcal{A}$ in a Kummer Extension of $\mathcal{A}_\sigma$

- Given a commutative semisimple $\mathbb{F}$-algebra $\mathcal{A}$ and a semiregular automorphism σ of *prime* order r.
- We could consider σ as also an automorphism of $\mathcal{A}[\zeta_r]$ by fixing ζ_r.
- We can efficiently find an $x \in T_{\mathcal{A},r}$ which is a Lagrange resolvent i.e. $\sigma(x) = \zeta_r x$. As before and a trick!
- Clearly $\sigma(x^r) = x^r$, which means that $c := x^r \in T_{\mathcal{A}_\sigma,r}$.
- Theorem: There is a *natural* isomorphism, $\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}] \cong \mathcal{A}_\sigma[\zeta_r][x] = \mathcal{A}[\zeta_r]$.
- Thus, we can efficiently compute a $c \in T_{\mathcal{A}_\sigma,r}$ s.t. $\mathcal{A} \cong (\mathcal{A}_\sigma[\zeta_r][\sqrt[r]{c}])_{\Delta_r}$. Bottomline: canonical embedding of $\mathcal{A}$.

Outline

Factoring Certain Cyclotomic Polynomials

- Input: Let $\Phi_m(X) \in \mathbb{F}[X]$ be a cyclotomic polynomial with $\mathbb{Z}_m^*$ being noncyclic.
- Consider $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$. It is a semisimple $\mathbb{F}$-algebra.
- For $i \in \mathbb{Z}_m^*$, the map $X \mapsto X^i$ gives an $\mathbb{F}$-automorphism of $\mathcal{A}$.
- Such maps form a group G of automorphisms. $G \cong \mathbb{Z}_m^*$ and hence noncyclic.
- For prime $r|\#G$, let P_r be the r-Sylow subgroup of G i.e. elements of G of r-power order.
- The factoring algorithm is based on computing these subgroups P_r and the various Lagrange resolvents in $\mathcal{A}$.

Factoring Certain Cyclotomic Polynomials

- Input: Let $\Phi_m(X) \in \mathbb{F}[X]$ be a cyclotomic polynomial with $\mathbb{Z}_m^*$ being noncyclic.
- Consider $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$. It is a semisimple $\mathbb{F}$-algebra.
- For $i \in \mathbb{Z}_m^*$, the map $X \mapsto X^i$ gives an $\mathbb{F}$-automorphism of $\mathcal{A}$.
- Such maps form a group G of automorphisms. $G \cong \mathbb{Z}_m^*$ and hence noncyclic.
- For prime $r | \#G$, let P_r be the r-Sylow subgroup of G i.e. elements of G of r-power order.
- The factoring algorithm is based on computing these subgroups P_r and the various Lagrange resolvents in $\mathcal{A}$.

Factoring Certain Cyclotomic Polynomials

- Input: Let $\Phi_m(X) \in \mathbb{F}[X]$ be a cyclotomic polynomial with $\mathbb{Z}_m^*$ being noncyclic.
- Consider $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$. It is a semisimple $\mathbb{F}$-algebra.
- For $i \in \mathbb{Z}_m^*$, the map $X \mapsto X^i$ gives an $\mathbb{F}$-automorphism of $\mathcal{A}$.
- Such maps form a group G of automorphisms. $G \cong \mathbb{Z}_m^*$ and hence noncyclic.
- For prime $r | \#G$, let P_r be the r-Sylow subgroup of G i.e. elements of G of r-power order.
- The factoring algorithm is based on computing these subgroups P_r and the various Lagrange resolvents in $\mathcal{A}$.

Factoring Certain Cyclotomic Polynomials

- Input: Let $\Phi_m(X) \in \mathbb{F}[X]$ be a cyclotomic polynomial with $\mathbb{Z}_m^*$ being noncyclic.
- Consider $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$. It is a semisimple $\mathbb{F}$-algebra.
- For $i \in \mathbb{Z}_m^*$, the map $X \mapsto X^i$ gives an $\mathbb{F}$-automorphism of $\mathcal{A}$.
- Such maps form a group G of automorphisms. $G \cong \mathbb{Z}_m^*$ and hence noncyclic.
- For prime $r | \#G$, let P_r be the r-Sylow subgroup of G i.e. elements of G of r-power order.
- The factoring algorithm is based on computing these subgroups P_r and the various Lagrange resolvents in $\mathcal{A}$.

Factoring Certain Cyclotomic Polynomials

- Input: Let $\Phi_m(X) \in \mathbb{F}[X]$ be a cyclotomic polynomial with $\mathbb{Z}_m^*$ being noncyclic.
- Consider $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$. It is a semisimple $\mathbb{F}$-algebra.
- For $i \in \mathbb{Z}_m^*$, the map $X \mapsto X^i$ gives an $\mathbb{F}$-automorphism of $\mathcal{A}$.
- Such maps form a group G of automorphisms. $G \cong \mathbb{Z}_m^*$ and hence noncyclic.
- For prime $r|\#G$, let P_r be the r-Sylow subgroup of G i.e. elements of G of r-power order.
- The factoring algorithm is based on computing these subgroups P_r and the various Lagrange resolvents in $\mathcal{A}$.

Factoring Certain Cyclotomic Polynomials

- Input: Let $\Phi_m(X) \in \mathbb{F}[X]$ be a cyclotomic polynomial with $\mathbb{Z}_m^*$ being noncyclic.
- Consider $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$. It is a semisimple $\mathbb{F}$-algebra.
- For $i \in \mathbb{Z}_m^*$, the map $X \mapsto X^i$ gives an $\mathbb{F}$-automorphism of $\mathcal{A}$.
- Such maps form a group G of automorphisms. $G \cong \mathbb{Z}_m^*$ and hence noncyclic.
- For prime $r | \# G$, let P_r be the r-Sylow subgroup of G i.e. elements of G of r-power order.
- The factoring algorithm is based on computing these subgroups P_r and the various Lagrange resolvents in $\mathcal{A}$.

Factoring Certain Cyclotomic Polynomials

- Input: Let $\Phi_m(X) \in \mathbb{F}[X]$ be a cyclotomic polynomial with $\mathbb{Z}_m^*$ being noncyclic.
- Consider $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$. It is a semisimple $\mathbb{F}$-algebra.
- For $i \in \mathbb{Z}_m^*$, the map $X \mapsto X^i$ gives an $\mathbb{F}$-automorphism of $\mathcal{A}$.
- Such maps form a group G of automorphisms. $G \cong \mathbb{Z}_m^*$ and hence noncyclic.
- For prime $r|\#G$, let P_r be the r-Sylow subgroup of G i.e. elements of G of r-power order.
- The factoring algorithm is based on computing these subgroups P_r and the various Lagrange resolvents in $\mathcal{A}$.

Factoring Cyclotomic Polynomials (Contd.)

- So $G \cong \mathbb{Z}_m^*$ is the noncyclic automorphism group of $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$.
- P_r is its r-Sylow subgroup and Π_r is the subset of elements of order r.
- For each $\sigma \in \Pi_r$ compute the Lagrange resolvent $x_\sigma \in T_{\mathcal{A},r}$ i.e. $\sigma(x_\sigma) = \zeta_r x_\sigma$.
- Consider the subgroup $H_r := \langle x_\sigma \mid \sigma \in \Pi_r \rangle$ of $T_{\mathcal{A},r}$.
- If H_r is noncyclic then we *can* compute a zero divisor in $\mathcal{A}$.
- If H_r is cyclic then P_r, which can be seen embedded in $\mathrm{Aut}(H_r)$, is also cyclic.
- Since G is noncyclic, one of the P_r is noncyclic and we are guaranteed to get a zero divisor in $\mathcal{A}$.

Factoring Cyclotomic Polynomials (Contd.)

- So $G \cong \mathbb{Z}_m^*$ is the noncyclic automorphism group of $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$.
- P_r is its r-Sylow subgroup and Π_r is the subset of elements of order r.
- For each $\sigma \in \Pi_r$ compute the Lagrange resolvent $x_\sigma \in T_{\mathcal{A},r}$ i.e. $\sigma(x_\sigma) = \zeta_r x_\sigma$.
- Consider the subgroup $H_r := \langle x_\sigma \mid \sigma \in \Pi_r \rangle$ of $T_{\mathcal{A},r}$.
- If H_r is noncyclic then we *can* compute a zero divisor in $\mathcal{A}$.
- If H_r is cyclic then P_r, which can be seen embedded in $\mathrm{Aut}(H_r)$, is also cyclic.
- Since G is noncyclic, one of the P_r is noncyclic and we are guaranteed to get a zero divisor in $\mathcal{A}$.

Factoring Cyclotomic Polynomials (Contd.)

- So $G \cong \mathbb{Z}_m^*$ is the noncyclic automorphism group of $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$.
- P_r is its r-Sylow subgroup and Π_r is the subset of elements of order r.
- For each $\sigma \in \Pi_r$ compute the Lagrange resolvent $x_\sigma \in T_{\mathcal{A},r}$ i.e. $\sigma(x_\sigma) = \zeta_r x_\sigma$.
- Consider the subgroup $H_r := \langle x_\sigma \mid \sigma \in \Pi_r \rangle$ of $T_{\mathcal{A},r}$.
- If H_r is noncyclic then we *can* compute a zero divisor in $\mathcal{A}$.
- If H_r is cyclic then P_r, which can be seen embedded in $\mathrm{Aut}(H_r)$, is also cyclic.
- Since G is noncyclic, one of the P_r is noncyclic and we are guaranteed to get a zero divisor in $\mathcal{A}$.

Factoring Cyclotomic Polynomials (Contd.)

- So $G \cong \mathbb{Z}_m^*$ is the noncyclic automorphism group of $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$.
- P_r is its r-Sylow subgroup and Π_r is the subset of elements of order r.
- For each $\sigma \in \Pi_r$ compute the Lagrange resolvent $x_\sigma \in T_{\mathcal{A},r}$ i.e. $\sigma(x_\sigma) = \zeta_r x_\sigma$.
- Consider the subgroup $H_r := \langle x_\sigma \mid \sigma \in \Pi_r \rangle$ of $T_{\mathcal{A},r}$.
- If H_r is noncyclic then we *can* compute a zero divisor in $\mathcal{A}$.
- If H_r is cyclic then P_r, which can be seen embedded in $\mathrm{Aut}(H_r)$, is also cyclic.
- Since G is noncyclic, one of the P_r is noncyclic and we are guaranteed to get a zero divisor in $\mathcal{A}$.

Factoring Cyclotomic Polynomials (Contd.)

- So $G \cong \mathbb{Z}_m^*$ is the noncyclic automorphism group of $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$.
- P_r is its r-Sylow subgroup and Π_r is the subset of elements of order r.
- For each $\sigma \in \Pi_r$ compute the Lagrange resolvent $x_\sigma \in T_{\mathcal{A},r}$ i.e. $\sigma(x_\sigma) = \zeta_r x_\sigma$.
- Consider the subgroup $H_r := \langle x_\sigma \mid \sigma \in \Pi_r \rangle$ of $T_{\mathcal{A},r}$.
- If H_r is noncyclic then we *can* compute a zero divisor in $\mathcal{A}$.
- If H_r is cyclic then P_r, which can be seen embedded in $\mathrm{Aut}(H_r)$, is also cyclic.
- Since G is noncyclic, one of the P_r is noncyclic and we are guaranteed to get a zero divisor in $\mathcal{A}$.

Factoring Cyclotomic Polynomials (Contd.)

- So $G \cong \mathbb{Z}_m^*$ is the noncyclic automorphism group of $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$.
- P_r is its r-Sylow subgroup and Π_r is the subset of elements of order r.
- For each $\sigma \in \Pi_r$ compute the Lagrange resolvent $x_\sigma \in T_{\mathcal{A},r}$ i.e. $\sigma(x_\sigma) = \zeta_r x_\sigma$.
- Consider the subgroup $H_r := \langle x_\sigma \mid \sigma \in \Pi_r \rangle$ of $T_{\mathcal{A},r}$.
- If H_r is noncyclic then we *can* compute a zero divisor in $\mathcal{A}$.
- If H_r is cyclic then P_r, which can be seen embedded in $\mathrm{Aut}(H_r)$, is also cyclic.
- Since G is noncyclic, one of the P_r is noncyclic and we are guaranteed to get a zero divisor in $\mathcal{A}$.

Factoring Cyclotomic Polynomials (Contd.)

- So $G \cong \mathbb{Z}_m^*$ is the noncyclic automorphism group of $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$.
- P_r is its r-Sylow subgroup and Π_r is the subset of elements of order r.
- For each $\sigma \in \Pi_r$ compute the Lagrange resolvent $x_\sigma \in T_{\mathcal{A},r}$ i.e. $\sigma(x_\sigma) = \zeta_r x_\sigma$.
- Consider the subgroup $H_r := \langle x_\sigma \mid \sigma \in \Pi_r \rangle$ of $T_{\mathcal{A},r}$.
- If H_r is noncyclic then we *can* compute a zero divisor in $\mathcal{A}$.
- If H_r is cyclic then P_r, which can be seen embedded in $\mathrm{Aut}(H_r)$, is also cyclic.
- Since G is noncyclic, one of the P_r is noncyclic and we are guaranteed to get a zero divisor in $\mathcal{A}$.

Factoring Cyclotomic Polynomials (Contd.)

- So $G \cong \mathbb{Z}_m^*$ is the noncyclic automorphism group of $\mathcal{A} = \mathbb{F}[X]/(\Phi_m(X))$.
- P_r is its r-Sylow subgroup and Π_r is the subset of elements of order r.
- For each $\sigma \in \Pi_r$ compute the Lagrange resolvent $x_\sigma \in T_{\mathcal{A},r}$ i.e. $\sigma(x_\sigma) = \zeta_r x_\sigma$.
- Consider the subgroup $H_r := \langle x_\sigma \mid \sigma \in \Pi_r \rangle$ of $T_{\mathcal{A},r}$.
- If H_r is noncyclic then we *can* compute a zero divisor in $\mathcal{A}$.
- If H_r is cyclic then P_r, which can be seen embedded in $\mathrm{Aut}(H_r)$, is also cyclic.
- Since G is noncyclic, one of the P_r is noncyclic and we are guaranteed to get a zero divisor in $\mathcal{A}$.

Outline

Our Results: Commutative Algebras

New Concepts / Tools

- Semiregularity
- Lagrange Resolvent
- Kummer Extension

A Warmup Application

Proof of the Main Result

Recall the Result

- Given $\mathcal{A}_0 = \mathbb{F}[X]/(f(X))$. We want to either find a zero divisor of $\mathcal{A}_0$ *or* an automorphism of order $\deg f$.
- We will actually solve a more general problem: given commutative semisimple finite algebras $\mathcal{B} \leq \mathcal{A}$, we compute either a zero divisor in $\mathcal{B}$ *or* a semiregular $\mathcal{B}$-automorphism of $\mathcal{A}$.
- Our algorithm is recursive. It recurses to an instance with a smaller $\dim_{\mathcal{B}} \mathcal{A}$ (but *larger* $\mathcal{A}$).
- Let us denote the algorithm by $\mathcal{F}(\mathcal{A}, \mathcal{B})$.
- The *initial* call is $\mathcal{F}(\mathcal{A}_0, \mathbb{F})$. The *terminal* call is when $\dim_{\mathcal{B}} \mathcal{A} = 1$.

Recall the Result

- Given $\mathcal{A}_0 = \mathbb{F}[X]/(f(X))$. We want to either find a zero divisor of $\mathcal{A}_0$ *or* an automorphism of order $\deg f$.
- We will actually solve a more general problem: given commutative semisimple finite algebras $\mathcal{B} \leq \mathcal{A}$, we compute either a zero divisor in $\mathcal{B}$ *or* a semiregular $\mathcal{B}$-automorphism of $\mathcal{A}$.
- Our algorithm is recursive. It recurses to an instance with a smaller $\dim_{\mathcal{B}} \mathcal{A}$ (but *larger* $\mathcal{A}$).
- Let us denote the algorithm by $\mathcal{F}(\mathcal{A}, \mathcal{B})$.
- The *initial* call is $\mathcal{F}(\mathcal{A}_0, \mathbb{F})$. The *terminal* call is when $\dim_{\mathcal{B}} \mathcal{A} = 1$.

Recall the Result

- Given $\mathcal{A}_0 = \mathbb{F}[X]/(f(X))$. We want to either find a zero divisor of $\mathcal{A}_0$ *or* an automorphism of order $\deg f$.
- We will actually solve a more general problem: given commutative semisimple finite algebras $\mathcal{B} \leq \mathcal{A}$, we compute either a zero divisor in $\mathcal{B}$ *or* a semiregular $\mathcal{B}$-automorphism of $\mathcal{A}$.
- Our algorithm is recursive. It recurses to an instance with a smaller $\dim_{\mathcal{B}} \mathcal{A}$ (but *larger* $\mathcal{A}$).
- Let us denote the algorithm by $\mathcal{F}(\mathcal{A}, \mathcal{B})$.
- The *initial* call is $\mathcal{F}(\mathcal{A}_0, \mathbb{F})$. The *terminal* call is when $\dim_{\mathcal{B}} \mathcal{A} = 1$.

Recall the Result

- Given $\mathcal{A}_0 = \mathbb{F}[X]/(f(X))$. We want to either find a zero divisor of $\mathcal{A}_0$ *or* an automorphism of order $\deg f$.
- We will actually solve a more general problem: given commutative semisimple finite algebras $\mathcal{B} \leq \mathcal{A}$, we compute either a zero divisor in $\mathcal{B}$ *or* a semiregular $\mathcal{B}$-automorphism of $\mathcal{A}$.
- Our algorithm is recursive. It recurses to an instance with a smaller $\dim_{\mathcal{B}} \mathcal{A}$ (but *larger* $\mathcal{A}$).
- Let us denote the algorithm by $\mathcal{F}(\mathcal{A}, \mathcal{B})$.
- The *initial* call is $\mathcal{F}(\mathcal{A}_0, \mathbb{F})$. The *terminal* call is when $\dim_{\mathcal{B}} \mathcal{A} = 1$.

Recall the Result

- Given $\mathcal{A}_0 = \mathbb{F}[X]/(f(X))$. We want to either find a zero divisor of $\mathcal{A}_0$ *or* an automorphism of order $\deg f$.
- We will actually solve a more general problem: given commutative semisimple finite algebras $\mathcal{B} \leq \mathcal{A}$, we compute either a zero divisor in $\mathcal{B}$ *or* a semiregular $\mathcal{B}$-automorphism of $\mathcal{A}$.
- Our algorithm is recursive. It recurses to an instance with a smaller $\dim_{\mathcal{B}} \mathcal{A}$ (but *larger* $\mathcal{A}$).
- Let us denote the algorithm by $\mathcal{F}(\mathcal{A}, \mathcal{B})$.
- The *initial* call is $\mathcal{F}(\mathcal{A}_0, \mathbb{F})$. The *terminal* call is when $\dim_{\mathcal{B}} \mathcal{A} = 1$.

Recall the Result

- Given $\mathcal{A}_0 = \mathbb{F}[X]/(f(X))$. We want to either find a zero divisor of $\mathcal{A}_0$ *or* an automorphism of order $\deg f$.
- We will actually solve a more general problem: given commutative semisimple finite algebras $\mathcal{B} \leq \mathcal{A}$, we compute either a zero divisor in $\mathcal{B}$ *or* a semiregular $\mathcal{B}$-automorphism of $\mathcal{A}$.
- Our algorithm is recursive. It recurses to an instance with a smaller $\dim_{\mathcal{B}} \mathcal{A}$ (but *larger* $\mathcal{A}$).
- Let us denote the algorithm by $\mathcal{F}(\mathcal{A}, \mathcal{B})$.
- The *initial* call is $\mathcal{F}(\mathcal{A}_0, \mathbb{F})$. The *terminal* call is when $\dim_{\mathcal{B}} \mathcal{A} = 1$.

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- Now we sketch the recursive algorithm $\mathcal{F}$ for inputs $\mathcal{A}, \mathcal{B}$.
- Check whether $\mathcal{A}$ is a free $\mathcal{B}$-module. If not then we have a zero divisor in $\mathcal{B}$.
- Case I: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *even*.
- Tensor idea: Consider the algebra $\mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$, and its homomorphism $\mu : x \otimes y \mapsto xy$ onto $\mathcal{A}$. The *kernel* of μ is an algebra $\mathcal{A}'$ of dimension $m(m-1)$ over $\mathcal{B}$.
- $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ and advantage of $\mathcal{A}'$: we know its $\mathcal{B}$-automorphism $\sigma : x \otimes y \mapsto y \otimes x$.

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- Now we sketch the recursive algorithm $\mathcal{F}$ for inputs $\mathcal{A}, \mathcal{B}$.
- Check whether $\mathcal{A}$ is a free $\mathcal{B}$-module. If not then we have a zero divisor in $\mathcal{B}$.
- Case I: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *even*.
- Tensor idea: Consider the algebra $\mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$, and its homomorphism $\mu : x \otimes y \mapsto xy$ onto $\mathcal{A}$. The *kernel* of μ is an algebra $\mathcal{A}'$ of dimension $m(m-1)$ over $\mathcal{B}$.
- $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ and advantage of $\mathcal{A}'$: we know its $\mathcal{B}$-automorphism $\sigma : x \otimes y \mapsto y \otimes x$.

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- Now we sketch the recursive algorithm $\mathcal{F}$ for inputs $\mathcal{A}, \mathcal{B}$.
- Check whether $\mathcal{A}$ is a free $\mathcal{B}$-module. If not then we have a zero divisor in $\mathcal{B}$.
- Case I: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *even*.
- Tensor idea: Consider the algebra $\mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$, and its homomorphism $\mu : x \otimes y \mapsto xy$ onto $\mathcal{A}$. The *kernel* of μ is an algebra $\mathcal{A}'$ of dimension $m(m-1)$ over $\mathcal{B}$.
- $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ and advantage of $\mathcal{A}'$: we know its $\mathcal{B}$-automorphism $\sigma : x \otimes y \mapsto y \otimes x$.

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- Now we sketch the recursive algorithm $\mathcal{F}$ for inputs $\mathcal{A}, \mathcal{B}$.
- Check whether $\mathcal{A}$ is a free $\mathcal{B}$-module. If not then we have a zero divisor in $\mathcal{B}$.
- Case I: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *even*.
- Tensor idea: Consider the algebra $\mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$, and its homomorphism $\mu : x \otimes y \mapsto xy$ onto $\mathcal{A}$. The *kernel* of μ is an algebra $\mathcal{A}'$ of dimension $m(m-1)$ over $\mathcal{B}$.
- $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ and advantage of $\mathcal{A}'$: we know its $\mathcal{B}$-automorphism $\sigma : x \otimes y \mapsto y \otimes x$.

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- Now we sketch the recursive algorithm $\mathcal{F}$ for inputs $\mathcal{A}, \mathcal{B}$.
- Check whether $\mathcal{A}$ is a free $\mathcal{B}$-module. If not then we have a zero divisor in $\mathcal{B}$.
- Case I: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *even*.
- Tensor idea: Consider the algebra $\mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$, and its homomorphism $\mu : x \otimes y \mapsto xy$ onto $\mathcal{A}$. The *kernel* of μ is an algebra $\mathcal{A}'$ of dimension $m(m-1)$ over $\mathcal{B}$.
- $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ and advantage of $\mathcal{A}'$: we know its $\mathcal{B}$-automorphism $\sigma : x \otimes y \mapsto y \otimes x$.

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- Now we sketch the recursive algorithm $\mathcal{F}$ for inputs $\mathcal{A}, \mathcal{B}$.
- Check whether $\mathcal{A}$ is a free $\mathcal{B}$-module. If not then we have a zero divisor in $\mathcal{B}$.
- Case I: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *even*.
- Tensor idea: Consider the algebra $\mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$, and its homomorphism $\mu : x \otimes y \mapsto xy$ onto $\mathcal{A}$. The *kernel* of μ is an algebra $\mathcal{A}'$ of dimension $m(m-1)$ over $\mathcal{B}$.
- $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ and advantage of $\mathcal{A}'$: we know its $\mathcal{B}$-automorphism $\sigma : x \otimes y \mapsto y \otimes x$.

The Algorithm $\mathcal{F}(\mathcal{A},\mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A},\mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot,\mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A},\mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A},\mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot,\mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A},\mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A},\mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot,\mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A}, \mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot, \mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A}, \mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot, \mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A}, \mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot, \mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A},\mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A},\mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot,\mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A},\mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A},\mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot,\mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A}, \mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot, \mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A},\mathcal{B})$: Even Case

- So we have embeddings $\mathcal{B} \hookrightarrow \mathcal{A} \hookrightarrow \mathcal{A}' \hookrightarrow \mathcal{A} \otimes_{\mathcal{B}} \mathcal{A}$. And a $\mathcal{B}$-automorphism σ of $\mathcal{A}'$.
- *We intend to bring σ down to $\mathcal{A}$.* Important: $\dim_{\mathcal{A}} \mathcal{A}'$ is odd while σ is semiregular of order 2.
- Compute a Lagrange resolvent $x \in T_{\mathcal{A}',2}$ i.e. $\sigma(x) = \zeta_2 x$.
- If $x \in \mathcal{A}$ then $\mathcal{C} := \mathcal{A}_\sigma[\zeta_2][x] = \mathcal{A}_\sigma[x]$ is a subalgebra of $\mathcal{A}$ with automorphism σ. So we call $\mathcal{F}(\mathcal{A},\mathcal{C})$ and glue the output with σ. Done!
- If $x \notin \mathcal{A}$ then $\mathcal{A}'$ cannot be a free $\mathcal{A}[x]$-module, as x is of order 2-power while $\dim_{\mathcal{A}} \mathcal{A}'$ is odd.
- Thus, we can find a zero divisor in $\mathcal{A}'$, decompose it and recursively call $\mathcal{F}(\cdot,\mathcal{A})$. Done!

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$: Odd Case

- Case II: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *odd*. Thus $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ is even.
- Note: There are two natural ways to embed $\mathcal{A}$ into $\mathcal{A}'$. Via $\phi_1 : x \mapsto x \otimes 1$ or $\phi_2 : x \mapsto 1 \otimes x$.
- We recurse to the even case and compute:
  a $\phi_1(\mathcal{A})$-automorphism σ_1 of $\mathcal{A}'$ and
  a $\phi_2(\mathcal{A})$-automorphism σ_2 of $\mathcal{A}'$.
- Essentially, $\phi_2^{-1}\sigma_1\phi_2$ is an automorphism of $\mathcal{A}$, and we are done!

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$: Odd Case

- Case II: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *odd*. Thus $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ is even.
- Note: There are two natural ways to embed $\mathcal{A}$ into $\mathcal{A}'$. Via $\phi_1 : x \mapsto x \otimes 1$ or $\phi_2 : x \mapsto 1 \otimes x$.
- We recurse to the even case and compute:
  a $\phi_1(\mathcal{A})$-automorphism σ_1 of $\mathcal{A}'$ and
  a $\phi_2(\mathcal{A})$-automorphism σ_2 of $\mathcal{A}'$.
- Essentially, $\phi_2^{-1}\sigma_1\phi_2$ is an automorphism of $\mathcal{A}$, and we are done!

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$: Odd Case

- Case II: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *odd*. Thus $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ is even.
- Note: There are two natural ways to embed $\mathcal{A}$ into $\mathcal{A}'$. Via $\phi_1 : x \mapsto x \otimes 1$ or $\phi_2 : x \mapsto 1 \otimes x$.
- We recurse to the even case and compute:
  a $\phi_1(\mathcal{A})$-automorphism σ_1 of $\mathcal{A}'$ and
  a $\phi_2(\mathcal{A})$-automorphism σ_2 of $\mathcal{A}'$.
- Essentially, $\phi_2^{-1}\sigma_1\phi_2$ is an automorphism of $\mathcal{A}$, and we are done!

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$: Odd Case

- Case II: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *odd*. Thus $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ is even.
- Note: There are two natural ways to embed $\mathcal{A}$ into $\mathcal{A}'$. Via $\phi_1 : x \mapsto x \otimes 1$ or $\phi_2 : x \mapsto 1 \otimes x$.
- We recurse to the even case and compute:
  a $\phi_1(\mathcal{A})$-automorphism σ_1 of $\mathcal{A}'$ and
  a $\phi_2(\mathcal{A})$-automorphism σ_2 of $\mathcal{A}'$.
- Essentially, $\phi_2^{-1}\sigma_1\phi_2$ is an automorphism of $\mathcal{A}$, and we are done!

The Algorithm $\mathcal{F}(\mathcal{A}, \mathcal{B})$: Odd Case

- Case II: $m := \dim_{\mathcal{B}} \mathcal{A}$ is *odd*. Thus $\dim_{\mathcal{A}} \mathcal{A}' = (m-1)$ is even.
- Note: There are two natural ways to embed $\mathcal{A}$ into $\mathcal{A}'$. Via $\phi_1 : x \mapsto x \otimes 1$ or $\phi_2 : x \mapsto 1 \otimes x$.
- We recurse to the even case and compute:
  a $\phi_1(\mathcal{A})$-automorphism σ_1 of $\mathcal{A}'$ and
  a $\phi_2(\mathcal{A})$-automorphism σ_2 of $\mathcal{A}'$.
- Essentially, $\phi_2^{-1}\sigma_1\phi_2$ is an automorphism of $\mathcal{A}$, and we are done!

Time Complexity of $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- In any recursive call $\mathcal{F}(\mathcal{C}, \mathcal{D})$ with $d := \dim_{\mathcal{D}} \mathcal{C}$ odd, we recurse to a bigger algebra $\mathcal{C} \otimes_{\mathcal{D}} \mathcal{C}$. $\dim_{\mathcal{D}} \mathcal{C}$ does not increase but $\dim_{\mathcal{B}} \mathcal{C}$ increases d times.
- Whenever we find a zero divisor of $\mathcal{C}$ we decompose it and always recurse to the smallest component. Thus, halving the $\dim_{\mathcal{D}} \mathcal{C}$.
- As we start with dimension $m = \dim_{\mathcal{B}} \mathcal{A}$, we are always in algebras of dimension at most $m^{O(\log m)}$ above $\mathcal{B}$.
- Overall the deterministic algorithm takes $\mathrm{poly}(m^{\log m}, \log |\mathcal{B}|)$ time.

Time Complexity of $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- In any recursive call $\mathcal{F}(\mathcal{C}, \mathcal{D})$ with $d := \dim_{\mathcal{D}} \mathcal{C}$ odd, we recurse to a bigger algebra $\mathcal{C} \otimes_{\mathcal{D}} \mathcal{C}$. $\dim_{\mathcal{D}} \mathcal{C}$ does not increase but $\dim_{\mathcal{B}} \mathcal{C}$ increases d times.
- Whenever we find a zero divisor of $\mathcal{C}$ we decompose it and always recurse to the smallest component. Thus, halving the $\dim_{\mathcal{D}} \mathcal{C}$.
- As we start with dimension $m = \dim_{\mathcal{B}} \mathcal{A}$, we are always in algebras of dimension at most $m^{O(\log m)}$ above $\mathcal{B}$.
- Overall the deterministic algorithm takes $\mathrm{poly}(m^{\log m}, \log |\mathcal{B}|)$ time.

Time Complexity of $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- In any recursive call $\mathcal{F}(\mathcal{C}, \mathcal{D})$ with $d := \dim_{\mathcal{D}} \mathcal{C}$ odd, we recurse to a bigger algebra $\mathcal{C} \otimes_{\mathcal{D}} \mathcal{C}$. $\dim_{\mathcal{D}} \mathcal{C}$ does not increase but $\dim_{\mathcal{B}} \mathcal{C}$ increases d times.
- Whenever we find a zero divisor of $\mathcal{C}$ we decompose it and always recurse to the smallest component. Thus, halving the $\dim_{\mathcal{D}} \mathcal{C}$.
- As we start with dimension $m = \dim_{\mathcal{B}} \mathcal{A}$, we are always in algebras of dimension at most $m^{O(\log m)}$ above $\mathcal{B}$.
- Overall the deterministic algorithm takes $\mathrm{poly}(m^{\log m}, \log |\mathcal{B}|)$ time.

Time Complexity of $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- In any recursive call $\mathcal{F}(\mathcal{C}, \mathcal{D})$ with $d := \dim_{\mathcal{D}} \mathcal{C}$ odd, we recurse to a bigger algebra $\mathcal{C} \otimes_{\mathcal{D}} \mathcal{C}$. $\dim_{\mathcal{D}} \mathcal{C}$ does not increase but $\dim_{\mathcal{B}} \mathcal{C}$ increases d times.
- Whenever we find a zero divisor of $\mathcal{C}$ we decompose it and always recurse to the smallest component. Thus, halving the $\dim_{\mathcal{D}} \mathcal{C}$.
- As we start with dimension $m = \dim_{\mathcal{B}} \mathcal{A}$, we are always in algebras of dimension at most $m^{O(\log m)}$ above $\mathcal{B}$.
- Overall the deterministic algorithm takes $\mathrm{poly}(m^{\log m}, \log |\mathcal{B}|)$ time.

Time Complexity of $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- In any recursive call $\mathcal{F}(\mathcal{C}, \mathcal{D})$ with $d := \dim_{\mathcal{D}} \mathcal{C}$ odd, we recurse to a bigger algebra $\mathcal{C} \otimes_{\mathcal{D}} \mathcal{C}$. $\dim_{\mathcal{D}} \mathcal{C}$ does not increase but $\dim_{\mathcal{B}} \mathcal{C}$ increases d times.
- Whenever we find a zero divisor of $\mathcal{C}$ we decompose it and always recurse to the smallest component. Thus, halving the $\dim_{\mathcal{D}} \mathcal{C}$.
- As we start with dimension $m = \dim_{\mathcal{B}} \mathcal{A}$, we are always in algebras of dimension at most $m^{O(\log m)}$ above $\mathcal{B}$.
- Overall the deterministic algorithm takes $\mathrm{poly}(m^{\log m}, \log |\mathcal{B}|)$ time.

Time Complexity of $\mathcal{F}(\mathcal{A}, \mathcal{B})$

- In any recursive call $\mathcal{F}(\mathcal{C}, \mathcal{D})$ with $d := \dim_{\mathcal{D}} \mathcal{C}$ odd, we recurse to a bigger algebra $\mathcal{C} \otimes_{\mathcal{D}} \mathcal{C}$. $\dim_{\mathcal{D}} \mathcal{C}$ does not increase but $\dim_{\mathcal{B}} \mathcal{C}$ increases d times.
- Whenever we find a zero divisor of $\mathcal{C}$ we decompose it and always recurse to the smallest component. Thus, halving the $\dim_{\mathcal{D}} \mathcal{C}$.
- As we start with dimension $m = \dim_{\mathcal{B}} \mathcal{A}$, we are always in algebras of dimension at most $m^{O(\log m)}$ above $\mathcal{B}$.
- Overall the deterministic algorithm takes $\mathrm{poly}(m^{\log m}, \log |\mathcal{B}|)$ time.

Part III

NONCOMMUTATIVE

Outline

Our Results: Noncommutative Algebras

Proof of the Main Result

Find a Zero Divisor

- Input: A noncommutative algebra $\mathcal{A}$ of dimension n over a finite field $\mathbb{F}$.
- Output: A zero divisor z in $\mathcal{A}$. I.e. for some nonzero $y, y' \in \mathcal{A}$, $yz = zy' = 0$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log |\mathbb{F}|)$ time.
- Note that it is a genuine elimination of GRH!

Find a Zero Divisor

- Input: A noncommutative algebra $\mathcal{A}$ of dimension n over a finite field $\mathbb{F}$.
- Output: A zero divisor z in $\mathcal{A}$. I.e. for some nonzero $y, y' \in \mathcal{A}$, $yz = zy' = 0$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log|\mathbb{F}|)$ time.
- Note that it is a genuine elimination of GRH!

Find a Zero Divisor

- Input: A noncommutative algebra $\mathcal{A}$ of dimension n over a finite field $\mathbb{F}$.
- Output: A zero divisor z in $\mathcal{A}$. I.e. for some nonzero $y, y' \in \mathcal{A}$, $yz = zy' = 0$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log|\mathbb{F}|)$ time.
- Note that it is a genuine elimination of GRH!

Find a Zero Divisor

- Input: A noncommutative algebra $\mathcal{A}$ of dimension n over a finite field $\mathbb{F}$.
- Output: A zero divisor z in $\mathcal{A}$. I.e. for some nonzero $y, y' \in \mathcal{A}$, $yz = zy' = 0$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log |\mathbb{F}|)$ time.
- Note that it is a genuine elimination of GRH!

Find a Zero Divisor

- Input: A noncommutative algebra $\mathcal{A}$ of dimension n over a finite field $\mathbb{F}$.
- Output: A zero divisor z in $\mathcal{A}$. I.e. for some nonzero $y, y' \in \mathcal{A}$, $yz = zy' = 0$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log |\mathbb{F}|)$ time.
- Note that it is a genuine elimination of GRH!

Find Isomorphism with Full Matrix Algebra

- Let K be a finite field and $M_n(K)$ be the full matrix algebra $K^{n\times n}$.
- Input: An $\mathbb{F}$-algebra $\mathcal{A}$ that is isomorphic to $M_n(K)$, for some $K \supset \mathbb{F}$.
- Output: Construct an *explicit* isomorphism $\mathcal{A} \cong M_n(K)$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log|K|)$ time.
- In other words, we solve the isomorphism problem for *finite simple algebras* in quasipolynomial time.
- Aside: Isomorphism problem for *finite algebras* is known to be graph isomorphism hard!

Find Isomorphism with Full Matrix Algebra

- Let K be a finite field and $M_n(K)$ be the full matrix algebra $K^{n \times n}$.
- Input: An $\mathbb{F}$-algebra $\mathcal{A}$ that is isomorphic to $M_n(K)$, for some $K \supset \mathbb{F}$.
- Output: Construct an *explicit* isomorphism $\mathcal{A} \cong M_n(K)$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log |K|)$ time.
- In other words, we solve the isomorphism problem for *finite simple algebras* in quasipolynomial time.
- Aside: Isomorphism problem for *finite algebras* is known to be graph isomorphism hard!

Find Isomorphism with Full Matrix Algebra

- Let K be a finite field and $M_n(K)$ be the full matrix algebra $K^{n \times n}$.
- Input: An $\mathbb{F}$-algebra $\mathcal{A}$ that is isomorphic to $M_n(K)$, for some $K \supset \mathbb{F}$.
- Output: Construct an *explicit* isomorphism $\mathcal{A} \cong M_n(K)$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log |K|)$ time.
- In other words, we solve the isomorphism problem for *finite simple algebras* in quasipolynomial time.
- Aside: Isomorphism problem for *finite algebras* is known to be graph isomorphism hard!

Find Isomorphism with Full Matrix Algebra

- Let K be a finite field and $M_n(K)$ be the full matrix algebra $K^{n\times n}$.
- Input: An $\mathbb{F}$-algebra $\mathcal{A}$ that is isomorphic to $M_n(K)$, for some $K \supset \mathbb{F}$.
- Output: Construct an *explicit* isomorphism $\mathcal{A} \cong M_n(K)$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log|K|)$ time.
- In other words, we solve the isomorphism problem for *finite simple algebras* in quasipolynomial time.
- Aside: Isomorphism problem for *finite algebras* is known to be graph isomorphism hard!

Find Isomorphism with Full Matrix Algebra

- Let K be a finite field and $M_n(K)$ be the full matrix algebra $K^{n \times n}$.
- Input: An $\mathbb{F}$-algebra $\mathcal{A}$ that is isomorphic to $M_n(K)$, for some $K \supset \mathbb{F}$.
- Output: Construct an *explicit* isomorphism $\mathcal{A} \cong M_n(K)$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log |K|)$ time.
- In other words, we solve the isomorphism problem for *finite simple algebras* in quasipolynomial time.
- Aside: Isomorphism problem for *finite algebras* is known to be graph isomorphism hard!

Find Isomorphism with Full Matrix Algebra

- Let K be a finite field and $M_n(K)$ be the full matrix algebra $K^{n\times n}$.
- Input: An $\mathbb{F}$-algebra $\mathcal{A}$ that is isomorphic to $M_n(K)$, for some $K \supset \mathbb{F}$.
- Output: Construct an *explicit* isomorphism $\mathcal{A} \cong M_n(K)$.
- Complexity: In deterministic $\mathrm{poly}(n^{\log n}, \log |K|)$ time.
- In other words, we solve the isomorphism problem for *finite simple algebras* in quasipolynomial time.
- Aside: Isomorphism problem for *finite algebras* is known to be graph isomorphism hard!

Outline

Our Results: Noncommutative Algebras

Proof of the Main Result

Preprocessing

- Let $\mathcal{A}$ be the finite noncommutative $\mathbb{F}$-algebra whose zero divisor we need to find.
- We could assume it to be *semisimple*, otherwise there are methods to compute the radical.
- By a linear system we compute the center $\mathcal{C}$ of $\mathcal{A}$, i.e. elements that commute with $\mathcal{A}$.
- Let $\mathcal{C}_1, \ldots, \mathcal{C}_r$ be simple components of $\mathcal{C}$. (We do not compute them.)
- Now structurally, $\mathcal{A} \cong \bigoplus_{i=1}^{r} M_{n_i}(\mathcal{C}_i)$. (Artin-Wedderburn)
- If n_i-s are not the same then $\mathcal{A}$ is *not* a free $\mathcal{C}$-module. Thus we compute a zero divisor.
- We can assume $\mathcal{A} \cong M_n(\mathcal{C})$, where we know n and the commutative semisimple $\mathcal{C}$.

Preprocessing

- Let $\mathcal{A}$ be the finite noncommutative $\mathbb{F}$-algebra whose zero divisor we need to find.
- We could assume it to be *semisimple*, otherwise there are methods to compute the radical.
- By a linear system we compute the center $\mathcal{C}$ of $\mathcal{A}$, i.e. elements that commute with $\mathcal{A}$.
- Let $\mathcal{C}_1, \ldots, \mathcal{C}_r$ be simple components of $\mathcal{C}$. (We do not compute them.)
- Now structurally, $\mathcal{A} \cong \bigoplus_{i=1}^{r} M_{n_i}(\mathcal{C}_i)$. (Artin-Wedderburn)
- If n_i-s are not the same then $\mathcal{A}$ is *not* a free $\mathcal{C}$-module. Thus we compute a zero divisor.
- We can assume $\mathcal{A} \cong M_n(\mathcal{C})$, where we know n and the commutative semisimple $\mathcal{C}$.

Preprocessing

- Let $\mathcal{A}$ be the finite noncommutative $\mathbb{F}$-algebra whose zero divisor we need to find.
- We could assume it to be *semisimple*, otherwise there are methods to compute the radical.
- By a linear system we compute the center $\mathcal{C}$ of $\mathcal{A}$, i.e. elements that commute with $\mathcal{A}$.
- Let $\mathcal{C}_1, \ldots, \mathcal{C}_r$ be simple components of $\mathcal{C}$. (We do not compute them.)
- Now structurally, $\mathcal{A} \cong \bigoplus_{i=1}^{r} M_{n_i}(\mathcal{C}_i)$. (Artin-Wedderburn)
- If n_i-s are not the same then $\mathcal{A}$ is *not* a free $\mathcal{C}$-module. Thus we compute a zero divisor.
- We can assume $\mathcal{A} \cong M_n(\mathcal{C})$, where we know n and the commutative semisimple $\mathcal{C}$.

Preprocessing

- Let $\mathcal{A}$ be the finite noncommutative $\mathbb{F}$-algebra whose zero divisor we need to find.
- We could assume it to be *semisimple*, otherwise there are methods to compute the radical.
- By a linear system we compute the center $\mathcal{C}$ of $\mathcal{A}$, i.e. elements that commute with $\mathcal{A}$.
- Let $\mathcal{C}_1, \ldots, \mathcal{C}_r$ be simple components of $\mathcal{C}$. (We do not compute them.)
- Now structurally, $\mathcal{A} \cong \bigoplus_{i=1}^{r} M_{n_i}(\mathcal{C}_i)$. (Artin-Wedderburn)
- If n_i-s are not the same then $\mathcal{A}$ is *not* a free $\mathcal{C}$-module. Thus we compute a zero divisor.
- We can assume $\mathcal{A} \cong M_n(\mathcal{C})$, where we know n and the commutative semisimple $\mathcal{C}$.

Preprocessing

- Let $\mathcal{A}$ be the finite noncommutative $\mathbb{F}$-algebra whose zero divisor we need to find.
- We could assume it to be *semisimple*, otherwise there are methods to compute the radical.
- By a linear system we compute the center $\mathcal{C}$ of $\mathcal{A}$, i.e. elements that commute with $\mathcal{A}$.
- Let $\mathcal{C}_1, \ldots, \mathcal{C}_r$ be simple components of $\mathcal{C}$. (We do not compute them.)
- Now structurally, $\mathcal{A} \cong \bigoplus_{i=1}^{r} M_{n_i}(\mathcal{C}_i)$. (Artin-Wedderburn)
- If n_i-s are not the same then $\mathcal{A}$ is *not* a free $\mathcal{C}$-module. Thus we compute a zero divisor.
- We can assume $\mathcal{A} \cong M_n(\mathcal{C})$, where we know n and the commutative semisimple $\mathcal{C}$.

Preprocessing

- Let $\mathcal{A}$ be the finite noncommutative $\mathbb{F}$-algebra whose zero divisor we need to find.
- We could assume it to be *semisimple*, otherwise there are methods to compute the radical.
- By a linear system we compute the center $\mathcal{C}$ of $\mathcal{A}$, i.e. elements that commute with $\mathcal{A}$.
- Let $\mathcal{C}_1, \ldots, \mathcal{C}_r$ be simple components of $\mathcal{C}$. (We do not compute them.)
- Now structurally, $\mathcal{A} \cong \bigoplus_{i=1}^{r} M_{n_i}(\mathcal{C}_i)$. (Artin-Wedderburn)
- If n_i-s are not the same then $\mathcal{A}$ is *not* a free $\mathcal{C}$-module. Thus we compute a zero divisor.
- We can assume $\mathcal{A} \cong M_n(\mathcal{C})$, where we know n and the commutative semisimple $\mathcal{C}$.

Preprocessing

- Let $\mathcal{A}$ be the finite noncommutative $\mathbb{F}$-algebra whose zero divisor we need to find.
- We could assume it to be *semisimple*, otherwise there are methods to compute the radical.
- By a linear system we compute the center $\mathcal{C}$ of $\mathcal{A}$, i.e. elements that commute with $\mathcal{A}$.
- Let $\mathcal{C}_1, \ldots, \mathcal{C}_r$ be simple components of $\mathcal{C}$. (We do not compute them.)
- Now structurally, $\mathcal{A} \cong \bigoplus_{i=1}^{r} M_{n_i}(\mathcal{C}_i)$. (Artin-Wedderburn)
- If n_i-s are not the same then $\mathcal{A}$ is *not* a free $\mathcal{C}$-module. Thus we compute a zero divisor.
- We can assume $\mathcal{A} \cong M_n(\mathcal{C})$, where we know n and the commutative semisimple $\mathcal{C}$.

Preprocessing

- Let $\mathcal{A}$ be the finite noncommutative $\mathbb{F}$-algebra whose zero divisor we need to find.
- We could assume it to be *semisimple*, otherwise there are methods to compute the radical.
- By a linear system we compute the center $\mathcal{C}$ of $\mathcal{A}$, i.e. elements that commute with $\mathcal{A}$.
- Let $\mathcal{C}_1, \ldots, \mathcal{C}_r$ be simple components of $\mathcal{C}$. (We do not compute them.)
- Now structurally, $\mathcal{A} \cong \bigoplus_{i=1}^{r} M_{n_i}(\mathcal{C}_i)$. (Artin-Wedderburn)
- If n_i-s are not the same then $\mathcal{A}$ is *not* a free $\mathcal{C}$-module. Thus we compute a zero divisor.
- We can assume $\mathcal{A} \cong M_n(\mathcal{C})$, where we know n and the commutative semisimple $\mathcal{C}$.

Preprocessing

- Let $\mathcal{A}$ be the finite noncommutative $\mathbb{F}$-algebra whose zero divisor we need to find.
- We could assume it to be *semisimple*, otherwise there are methods to compute the radical.
- By a linear system we compute the center $\mathcal{C}$ of $\mathcal{A}$, i.e. elements that commute with $\mathcal{A}$.
- Let $\mathcal{C}_1, \dots, \mathcal{C}_r$ be simple components of $\mathcal{C}$. (We do not compute them.)
- Now structurally, $\mathcal{A} \cong \bigoplus_{i=1}^{r} M_{n_i}(\mathcal{C}_i)$. (Artin-Wedderburn)
- If n_i-s are not the same then $\mathcal{A}$ is *not* a free $\mathcal{C}$-module. Thus we compute a zero divisor.
- We can assume $\mathcal{A} \cong M_n(\mathcal{C})$, where we know n and the commutative semisimple $\mathcal{C}$.

Automorphism gives Conjugation

- So we are given an $\mathbb{F}$-algebra $\mathcal{A}$ isomorphic to $M_n(\mathcal{C})$.
- *We intend to compute an automorphism of a commutative subalgebra and use it to construct a zero divisor in $\mathcal{A}$.*
- Theorem (Skolem-Noether): Let σ be a $\mathcal{C}$-automorphism of a commutative semisimple $\mathcal{B} \leq M_n(\mathcal{C})$. Then $\exists y \in M_n(\mathcal{C})$ s.t. $\forall x \in \mathcal{B}$, $\sigma(x) = y^{-1}xy$.
- Bottomline: Automorphism gives a conjugation.

Automorphism gives Conjugation

- So we are given an $\mathbb{F}$-algebra $\mathcal{A}$ isomorphic to $M_n(\mathcal{C})$.
- *We intend to compute an automorphism of a commutative subalgebra and use it to construct a zero divisor in* $\mathcal{A}$.
- Theorem (Skolem-Noether): Let σ be a $\mathcal{C}$-automorphism of a commutative semisimple $\mathcal{B} \leq M_n(\mathcal{C})$. Then $\exists y \in M_n(\mathcal{C})$ s.t. $\forall x \in \mathcal{B}$, $\sigma(x) = y^{-1}xy$.
- Bottomline: Automorphism gives a conjugation.

Automorphism gives Conjugation

- So we are given an $\mathbb{F}$-algebra $\mathcal{A}$ isomorphic to $M_n(\mathcal{C})$.
- *We intend to compute an automorphism of a commutative subalgebra and use it to construct a zero divisor in $\mathcal{A}$.*
- Theorem (Skolem-Noether): Let σ be a $\mathcal{C}$-automorphism of a commutative semisimple $\mathcal{B} \leq M_n(\mathcal{C})$. Then $\exists y \in M_n(\mathcal{C})$ s.t. $\forall x \in \mathcal{B}$, $\sigma(x) = y^{-1}xy$.
- Bottomline: Automorphism gives a conjugation.

Automorphism gives Conjugation

- So we are given an $\mathbb{F}$-algebra $\mathcal{A}$ isomorphic to $M_n(\mathcal{C})$.
- *We intend to compute an automorphism of a commutative subalgebra and use it to construct a zero divisor in $\mathcal{A}$.*
- Theorem (Skolem-Noether): Let σ be a $\mathcal{C}$-automorphism of a commutative semisimple $\mathcal{B} \leq M_n(\mathcal{C})$. Then $\exists y \in M_n(\mathcal{C})$ s.t. $\forall x \in \mathcal{B}$, $\sigma(x) = y^{-1}xy$.
- Bottomline: Automorphism gives a conjugation.

Conjugation gives Zero Divisor

- Let $y \in M_n(\mathcal{C})$ be of order r, and it induces a nontrivial automorphism on $\mathcal{B}$ i.e. $y^{-1}\mathcal{B}y = \mathcal{B}$.
- Then it can be shown that $(X^r - 1)$ is the minimal polynomial of y over $\mathbb{F}$.
- Consequently, $(y - 1)$ and $(y^{r-1} + \cdots + y + 1)$ are both zero divisors in $M_n(\mathcal{C}) \cong \mathcal{A}$.
- This observation suggests us a plan: Find a commutative semisimple $\mathcal{B} \leq \mathcal{A}$ and
- a $y \in \mathcal{A}$ of order r, that induces a nontrivial conjugation automorphism of $\mathcal{B}$.

Conjugation gives Zero Divisor

- Let $y \in M_n(\mathcal{C})$ be of order r, and it induces a nontrivial automorphism on $\mathcal{B}$ i.e. $y^{-1}\mathcal{B}y = \mathcal{B}$.
- Then it can be shown that $(X^r - 1)$ is the minimal polynomial of y over $\mathbb{F}$.
- Consequently, $(y - 1)$ and $(y^{r-1} + \cdots + y + 1)$ are both zero divisors in $M_n(\mathcal{C}) \cong \mathcal{A}$.
- This observation suggests us a plan: Find a commutative semisimple $\mathcal{B} \leq \mathcal{A}$ and
- a $y \in \mathcal{A}$ of order r, that induces a nontrivial conjugation automorphism of $\mathcal{B}$.

Conjugation gives Zero Divisor

- Let $y \in M_n(\mathcal{C})$ be of order r, and it induces a nontrivial automorphism on $\mathcal{B}$ i.e. $y^{-1}\mathcal{B}y = \mathcal{B}$.
- Then it can be shown that $(X^r - 1)$ is the minimal polynomial of y over $\mathbb{F}$.
- Consequently, $(y - 1)$ and $(y^{r-1} + \cdots + y + 1)$ are both zero divisors in $M_n(\mathcal{C}) \cong \mathcal{A}$.
- This observation suggests us a plan: Find a commutative semisimple $\mathcal{B} \leq \mathcal{A}$ and
- a $y \in \mathcal{A}$ of order r, that induces a nontrivial conjugation automorphism of $\mathcal{B}$.

Conjugation gives Zero Divisor

- Let $y \in M_n(\mathcal{C})$ be of order r, and it induces a nontrivial automorphism on $\mathcal{B}$ i.e. $y^{-1}\mathcal{B}y = \mathcal{B}$.
- Then it can be shown that $(X^r - 1)$ is the minimal polynomial of y over $\mathbb{F}$.
- Consequently, $(y - 1)$ and $(y^{r-1} + \cdots + y + 1)$ are both zero divisors in $M_n(\mathcal{C}) \cong \mathcal{A}$.
- This observation suggests us a plan: Find a commutative semisimple $\mathcal{B} \leq \mathcal{A}$ and
- a $y \in \mathcal{A}$ of order r, that induces a nontrivial conjugation automorphism of $\mathcal{B}$.

Conjugation gives Zero Divisor

- Let $y \in M_n(\mathcal{C})$ be of order r, and it induces a nontrivial automorphism on $\mathcal{B}$ i.e. $y^{-1}\mathcal{B}y = \mathcal{B}$.
- Then it can be shown that $(X^r - 1)$ is the minimal polynomial of y over $\mathbb{F}$.
- Consequently, $(y-1)$ and $(y^{r-1} + \cdots + y + 1)$ are both zero divisors in $M_n(\mathcal{C}) \cong \mathcal{A}$.
- This observation suggests us a plan: Find a commutative semisimple $\mathcal{B} \leq \mathcal{A}$ and
- a $y \in \mathcal{A}$ of order r, that induces a nontrivial conjugation automorphism of $\mathcal{B}$.

The Algorithm: Almost

- Given $\mathcal{A}$ and (its center) $\mathcal{C}$. Compute a *maximal* commutative semisimple $\mathcal{B} \leq \mathcal{A}$.
- $\mathcal{B}$ is a free $\mathcal{C}$-module of rank n.
- Using our commutative algorithm: find a semiregular $\mathcal{C}$-automorphism σ of $\mathcal{B}$.
- Compute a $y \in \mathcal{A}$ s.t. $\forall b \in \mathcal{B}$, $\sigma(b) = y^{-1}by$ (guaranteed by Skolem-Noether).
- We can replace σ and y by an appropriate power s.t. $\mathrm{ord}(\sigma) = r$ and $\mathrm{ord}(y)$ is an r-power.
- Put $z := y^r$. If $z = 1$ then we are done!
- Assume $z \neq 1$. $\forall b \in \mathcal{B}$, $b = \sigma^r(b) = z^{-1}bz$.
- $\mathcal{B}[z]$ is commutative semisimple with automorphism σ via conjugation by y.

The Algorithm: Almost

- Given $\mathcal{A}$ and (its center) $\mathcal{C}$. Compute a *maximal* commutative semisimple $\mathcal{B} \leq \mathcal{A}$.
- $\mathcal{B}$ is a free $\mathcal{C}$-module of rank n.
- Using our commutative algorithm: find a semiregular $\mathcal{C}$-automorphism σ of $\mathcal{B}$.
- Compute a $y \in \mathcal{A}$ s.t. $\forall b \in \mathcal{B}$, $\sigma(b) = y^{-1}by$ (guaranteed by Skolem-Noether).
- We can replace σ and y by an appropriate power s.t. $\mathrm{ord}(\sigma) = r$ and $\mathrm{ord}(y)$ is an r-power.
- Put $z := y^r$. If $z = 1$ then we are done!
- Assume $z \neq 1$. $\forall b \in \mathcal{B}$, $b = \sigma^r(b) = z^{-1}bz$.
- $\mathcal{B}[z]$ is commutative semisimple with automorphism σ via conjugation by y.

The Algorithm: Almost

- Given $\mathcal{A}$ and (its center) $\mathcal{C}$. Compute a *maximal* commutative semisimple $\mathcal{B} \leq \mathcal{A}$.
- $\mathcal{B}$ is a free $\mathcal{C}$-module of rank n.
- Using our commutative algorithm: find a semiregular $\mathcal{C}$-automorphism σ of $\mathcal{B}$.
- Compute a $y \in \mathcal{A}$ s.t. $\forall b \in \mathcal{B}$, $\sigma(b) = y^{-1}by$ (guaranteed by Skolem-Noether).
- We can replace σ and y by an appropriate power s.t. $\mathrm{ord}(\sigma) = r$ and $\mathrm{ord}(y)$ is an r-power.
- Put $z := y^r$. If $z = 1$ then we are done!
- Assume $z \neq 1$. $\forall b \in \mathcal{B}$, $b = \sigma^r(b) = z^{-1}bz$.
- $\mathcal{B}[z]$ is commutative semisimple with automorphism σ via conjugation by y.

The Algorithm: Almost

- Given $\mathcal{A}$ and (its center) $\mathcal{C}$. Compute a *maximal* commutative semisimple $\mathcal{B} \leq \mathcal{A}$.
- $\mathcal{B}$ is a free $\mathcal{C}$-module of rank n.
- Using our commutative algorithm: find a semiregular $\mathcal{C}$-automorphism σ of $\mathcal{B}$.
- Compute a $y \in \mathcal{A}$ s.t. $\forall b \in \mathcal{B}$, $\sigma(b) = y^{-1}by$ (guaranteed by Skolem-Noether).
- We can replace σ and y by an appropriate power s.t. $\mathrm{ord}(\sigma) = r$ and $\mathrm{ord}(y)$ is an r-power.
- Put $z := y^r$. If $z = 1$ then we are done!
- Assume $z \neq 1$. $\forall b \in \mathcal{B}$, $b = \sigma^r(b) = z^{-1}bz$.
- $\mathcal{B}[z]$ is commutative semisimple with automorphism σ via conjugation by y.

The Algorithm: Almost

- Given $\mathcal{A}$ and (its center) $\mathcal{C}$. Compute a *maximal* commutative semisimple $\mathcal{B} \leq \mathcal{A}$.
- $\mathcal{B}$ is a free $\mathcal{C}$-module of rank n.
- Using our commutative algorithm: find a semiregular $\mathcal{C}$-automorphism σ of $\mathcal{B}$.
- Compute a $y \in \mathcal{A}$ s.t. $\forall b \in \mathcal{B}$, $\sigma(b) = y^{-1}by$ (guaranteed by Skolem-Noether).
- We can replace σ and y by an appropriate power s.t. $\mathrm{ord}(\sigma) = r$ and $\mathrm{ord}(y)$ is an r-power.
- Put $z := y^r$. If $z = 1$ then we are done!
- Assume $z \neq 1$. $\forall b \in \mathcal{B}$, $b = \sigma^r(b) = z^{-1}bz$.
- $\mathcal{B}[z]$ is commutative semisimple with automorphism σ via conjugation by y.

The Algorithm: Almost

- Given $\mathcal{A}$ and (its center) $\mathcal{C}$. Compute a *maximal* commutative semisimple $\mathcal{B} \leq \mathcal{A}$.
- $\mathcal{B}$ is a free $\mathcal{C}$-module of rank n.
- Using our commutative algorithm: find a semiregular $\mathcal{C}$-automorphism σ of $\mathcal{B}$.
- Compute a $y \in \mathcal{A}$ s.t. $\forall b \in \mathcal{B}$, $\sigma(b) = y^{-1}by$ (guaranteed by Skolem-Noether).
- We can replace σ and y by an appropriate power s.t. $\mathrm{ord}(\sigma) = r$ and $\mathrm{ord}(y)$ is an r-power.
- Put $z := y^r$. If $z = 1$ then we are done!
- Assume $z \neq 1$. $\forall b \in \mathcal{B}$, $b = \sigma^r(b) = z^{-1}bz$.
- $\mathcal{B}[z]$ is commutative semisimple with automorphism σ via conjugation by y.

The Algorithm: Almost

- Given $\mathcal{A}$ and (its center) $\mathcal{C}$. Compute a *maximal* commutative semisimple $\mathcal{B} \leq \mathcal{A}$.
- $\mathcal{B}$ is a free $\mathcal{C}$-module of rank n.
- Using our commutative algorithm: find a semiregular $\mathcal{C}$-automorphism σ of $\mathcal{B}$.
- Compute a $y \in \mathcal{A}$ s.t. $\forall b \in \mathcal{B}$, $\sigma(b) = y^{-1}by$ (guaranteed by Skolem-Noether).
- We can replace σ and y by an appropriate power s.t. $\mathrm{ord}(\sigma) = r$ and $\mathrm{ord}(y)$ is an r-power.
- Put $z := y^r$. If $z = 1$ then we are done!
- Assume $z \neq 1$. $\forall b \in \mathcal{B}$, $b = \sigma^r(b) = z^{-1}bz$.
- $\mathcal{B}[z]$ is commutative semisimple with automorphism σ via conjugation by y.

The Algorithm: Almost

- Given $\mathcal{A}$ and (its center) $\mathcal{C}$. Compute a *maximal* commutative semisimple $\mathcal{B} \leq \mathcal{A}$.
- $\mathcal{B}$ is a free $\mathcal{C}$-module of rank n.
- Using our commutative algorithm: find a semiregular $\mathcal{C}$-automorphism σ of $\mathcal{B}$.
- Compute a $y \in \mathcal{A}$ s.t. $\forall b \in \mathcal{B}$, $\sigma(b) = y^{-1}by$ (guaranteed by Skolem-Noether).
- We can replace σ and y by an appropriate power s.t. $\mathrm{ord}(\sigma) = r$ and $\mathrm{ord}(y)$ is an r-power.
- Put $z := y^r$. If $z = 1$ then we are done!
- Assume $z \neq 1$. $\forall b \in \mathcal{B}$, $b = \sigma^r(b) = z^{-1}bz$.
- $\mathcal{B}[z]$ is commutative semisimple with automorphism σ via conjugation by y.

The Proof Details: Cyclic Algebra

- So $\mathcal{B}[z]$ has ζ_r and an automorphism σ of order r.
- Compute a Lagrange resolvent $x \in \mathcal{B}[z]^*_r$, i.e. $\sigma(x) = \zeta_r x$.
- Consider $\mathcal{C}' := \mathcal{B}[z]_\sigma$ and $\mathcal{A}' := \mathcal{C}'[x, y]$.
- We can assume $\mathcal{A}'$ to be a free $\mathcal{C}'$-module. Its generators satisfy: $xy = \zeta_r yx$ and $x^r, y^r \in \mathcal{C}'$.
- $\mathcal{A}'$ is called cyclic algebra. It is a generalization of quaternions.
- With some work we find a zero divisor in it. Done!

The Proof Details: Cyclic Algebra

- So $\mathcal{B}[z]$ has ζ_r and an automorphism σ of order r.
- Compute a Lagrange resolvent $x \in \mathcal{B}[z]_r^*$, i.e. $\sigma(x) = \zeta_r x$.
- Consider $\mathcal{C}' := \mathcal{B}[z]_\sigma$ and $\mathcal{A}' := \mathcal{C}'[x, y]$.
- We can assume $\mathcal{A}'$ to be a free $\mathcal{C}'$-module. Its generators satisfy: $xy = \zeta_r yx$ and $x^r, y^r \in \mathcal{C}'$.
- $\mathcal{A}'$ is called cyclic algebra. It is a generalization of quaternions.
- With some work we find a zero divisor in it. Done!

The Proof Details: Cyclic Algebra

- So $\mathcal{B}[z]$ has ζ_r and an automorphism σ of order r.
- Compute a Lagrange resolvent $x \in \mathcal{B}[z]^*_r$, i.e. $\sigma(x) = \zeta_r x$.
- Consider $\mathcal{C}' := \mathcal{B}[z]_\sigma$ and $\mathcal{A}' := \mathcal{C}'[x, y]$.
- We can assume $\mathcal{A}'$ to be a free $\mathcal{C}'$-module. Its generators satisfy: $xy = \zeta_r yx$ and $x^r, y^r \in \mathcal{C}'$.
- $\mathcal{A}'$ is called cyclic algebra. It is a generalization of quaternions.
- With some work we find a zero divisor in it. Done!

The Proof Details: Cyclic Algebra

- So $\mathcal{B}[z]$ has ζ_r and an automorphism σ of order r.
- Compute a Lagrange resolvent $x \in \mathcal{B}[z]_r^*$, i.e. $\sigma(x) = \zeta_r x$.
- Consider $\mathcal{C}' := \mathcal{B}[z]_\sigma$ and $\mathcal{A}' := \mathcal{C}'[x, y]$.
- We can assume $\mathcal{A}'$ to be a free $\mathcal{C}'$-module. Its generators satisfy: $xy = \zeta_r yx$ and $x^r, y^r \in \mathcal{C}'$.
- $\mathcal{A}'$ is called cyclic algebra. It is a generalization of quaternions.
- With some work we find a zero divisor in it. Done!

The Proof Details: Cyclic Algebra

- So $\mathcal{B}[z]$ has ζ_r and an automorphism σ of order r.
- Compute a Lagrange resolvent $x \in \mathcal{B}[z]_r^*$, i.e. $\sigma(x) = \zeta_r x$.
- Consider $\mathcal{C}' := \mathcal{B}[z]_\sigma$ and $\mathcal{A}' := \mathcal{C}'[x, y]$.
- We can assume $\mathcal{A}'$ to be a free $\mathcal{C}'$-module. Its generators satisfy: $xy = \zeta_r yx$ and $x^r, y^r \in \mathcal{C}'$.
- $\mathcal{A}'$ is called cyclic algebra. It is a generalization of quaternions.
- With some work we find a zero divisor in it. Done!

The Proof Details: Cyclic Algebra

- So $\mathcal{B}[z]$ has ζ_r and an automorphism σ of order r.
- Compute a Lagrange resolvent $x \in \mathcal{B}[z]_r^*$, i.e. $\sigma(x) = \zeta_r x$.
- Consider $\mathcal{C}' := \mathcal{B}[z]_\sigma$ and $\mathcal{A}' := \mathcal{C}'[x, y]$.
- We can assume $\mathcal{A}'$ to be a free $\mathcal{C}'$-module. Its generators satisfy: $xy = \zeta_r yx$ and $x^r, y^r \in \mathcal{C}'$.
- $\mathcal{A}'$ is called cyclic algebra. It is a generalization of quaternions.
- With some work we find a zero divisor in it. Done!

The Proof Details: Cyclic Algebra

- So $\mathcal{B}[z]$ has ζ_r and an automorphism σ of order r.
- Compute a Lagrange resolvent $x \in \mathcal{B}[z]^*_r$, i.e. $\sigma(x) = \zeta_r x$.
- Consider $\mathcal{C}' := \mathcal{B}[z]_\sigma$ and $\mathcal{A}' := \mathcal{C}'[x, y]$.
- We can assume $\mathcal{A}'$ to be a free $\mathcal{C}'$-module. Its generators satisfy: $xy = \zeta_r yx$ and $x^r, y^r \in \mathcal{C}'$.
- $\mathcal{A}'$ is called cyclic algebra. It is a generalization of quaternions.
- With some work we find a zero divisor in it. Done!

Conclusion

- We developed a computational version of *Galois theory* for finite semisimple algebras.
- This gave us GRH free ways to compute *semiregular automorphisms* in the commutative case.
- And GRH free ways to compute *zero divisors* in the noncommutative case.
- In some cases we *factor polynomials* too!
- Can we extend the methods to solve polynomial factoring ?

Thank You!

Conclusion

- We developed a computational version of *Galois theory* for finite semisimple algebras.
- This gave us GRH free ways to compute *semiregular automorphisms* in the commutative case.
- And GRH free ways to compute *zero divisors* in the noncommutative case.
- In some cases we *factor polynomials* too!
- Can we extend the methods to solve polynomial factoring ?

Thank You!

Conclusion

- We developed a computational version of *Galois theory* for finite semisimple algebras.
- This gave us GRH free ways to compute *semiregular automorphisms* in the commutative case.
- And GRH free ways to compute *zero divisors* in the noncommutative case.
- In some cases we *factor polynomials* too!
- Can we extend the methods to solve polynomial factoring ?

Thank You!

Conclusion

- We developed a computational version of *Galois theory* for finite semisimple algebras.
- This gave us GRH free ways to compute *semiregular automorphisms* in the commutative case.
- And GRH free ways to compute *zero divisors* in the noncommutative case.
- In some cases we *factor polynomials* too!
- Can we extend the methods to solve polynomial factoring ?

Thank You!

Conclusion

- We developed a computational version of *Galois theory* for finite semisimple algebras.
- This gave us GRH free ways to compute *semiregular automorphisms* in the commutative case.
- And GRH free ways to compute *zero divisors* in the noncommutative case.
- In some cases we *factor polynomials* too!
- Can we extend the methods to solve polynomial factoring ?

Thank You!

Conclusion

- We developed a computational version of *Galois theory* for finite semisimple algebras.
- This gave us GRH free ways to compute *semiregular automorphisms* in the commutative case.
- And GRH free ways to compute *zero divisors* in the noncommutative case.
- In some cases we *factor polynomials* too!
- Can we extend the methods to solve polynomial factoring ?

Thank You!