Generating Subfields

joint with Marc van Hoeij, Andrew Novocin

Jürgen Klüners

Universität Paderborn

Number Theory Conference, Bordeaux, 14th January 2013

The subfield problem

Situation

Let K/k be a finite separable field extension of degree n.
Assume $K = k(\alpha)$ with minimal polynomial $f \in k[x]$ of α.

Goal

Compute all intermediate fields $k \subseteq L = k(\beta) \subseteq K = k(\alpha)$ (in polynomial time).

Algorithmic: Compute $\beta = \sum_{i=0}^{n-1} b_i \alpha^i$ for each subfield.

Example

$\mathbf{Gal}(K/k) \cong C_2^m \Rightarrow$ there are about $2^{\log(n)}$ subfields ($n = 2^m$).

Galois theory – subfields

Definition

$\emptyset \neq \Delta \subseteq \Omega$ is called block, if $\Delta^\tau \cap \Delta \in \{\emptyset, \Delta\}$ for all $\tau \in G$.

$$
\begin{array}{ccccc}
\mathbb{Q}(\alpha_1,\ldots,\alpha_n) & & \{\mathrm{id}\} & & \\
| & & | & & \\
\mathbb{Q}(\alpha_1) & & G_{\alpha_1} & & \{\alpha_1\} \\
| & d & | & & | \\
\mathbb{Q}(\beta) & & U & & U\alpha_1 = \{\alpha_{i_1},\ldots,\alpha_{i_d}\} = \Delta_1 \\
| & m & | & & | \\
\mathbb{Q} & & G & & \{\alpha_1,\ldots,\alpha_n\} = \Omega
\end{array}
$$

Lemma

$\mathbf{Gal}(f)$ *has block* Δ *of size* d*.*
Then $\mathbf{Gal}(f) \leq S_d \wr S_m$

Galois group computation is much more difficult!

Applications I

Example (Use a CAS to solve this system of equations:)

$$a^2 - 2ab + b^2 - 8 = 0, \quad a^2b^2 - (a^2 + 2a + 5)b + a^3 - 3a + 3 = 0$$

Result: $a = \alpha, \quad b =$

$$\frac{-17\alpha^7}{1809} + \frac{61\alpha^6}{3618} + \frac{371\alpha^5}{1809} - \frac{1757\alpha^4}{3618} - \frac{563\alpha^3}{603} + \frac{6013\alpha^2}{3618} + \frac{3184\alpha}{1809} + \frac{7175}{3618}$$

where α is a solution of

$$x^8 - 20x^6 + 16x^5 + 98x^4 + 32x^3 - 12x^2 - 208x - 191 = 0.$$

Simpler Solution:

$$a = \sqrt{3} + \sqrt[4]{2} - \sqrt{2}, \quad b = \sqrt{3} + \sqrt[4]{2} + \sqrt{2}$$

To find it we first need subfields of $\mathbb{Q}(\alpha)$.

Applications II

Bostan and Kauers [Proc AMS 2010] gave an algebraic expression for the generating function for Gessel walks, using two minpoly's with a combined size of 172 Kb. By computing subfields, this expression could be reduced to just 300 bytes, a 99.8% reduction. The idea is:

When $\mathrm{char}(k) = 0$, then a tower of extensions

$$k \subseteq k(\alpha_1) \subseteq k(\alpha_2) \subseteq k(\alpha_3) = K$$

can be given by a single extension $K = k(\alpha)$.

In general, the **primitive element theorem** will produce an α with a minpoly $f(x)$ of large size. Thus we can expect to reduce expression sizes using the reverse process (computing subfields).

The Subfield Polynomial

Situation

Let K/k be a finite separable field extension of degree n.
Assume $K = k(\alpha)$ with minimal polynomial $f \in k[x]$ of α.

Definition

We call the minpoly h of α over L the **subfield-polynomial** of L.

- L is generated (as a field) by the coefficients of h.
- L is generated (as a vector space) by the coefficients of f/h.

Naive algorithm

A subfield polynomial is also a factor of f in $k(\alpha)[x]$. So we could find all subfields by trying out every factor of f in $k(\alpha)[x]$.

Factors of f

Let $f = (x - \alpha) \cdot f_2 \cdots f_r$ be the factorization of f in $k(\alpha)[x]$.

Finding Subfields, Exponential Complexity:

For each of the 2^r monic factors of f in $k(\alpha)[x]$, compute the field generated by the coefficients of that factor.

Finding Subfields, Polynomial Complexity:

We perform a computation for each polynomial $f_2, f_3, \ldots, f_r$.

Problems:

1. These $f_2, f_3, \ldots$ are not subfield-polynomials (i.e. we do not get a subfield by simply looking at their coefficients).
2. We do not get all subfields in this way.

The principal subfields

Factorization step, $K \subseteq \tilde{K}$

$f = (x - \alpha) \cdot f_2 \cdots f_r \in \tilde{K}[x]$ complete factorization.

Elements of K are of the form $g(\alpha)$, where $g \in K[x]$ of degree $< n$.

1. $\tilde{K}_i := \tilde{K}[x]/(f_i)$
2. $\Phi_i : K \to \tilde{K}_i, g(\alpha) \mapsto g(x) \mod f_i$.
3. $L_i := \ker(\Phi_i - id) = \{g(\alpha) \in K \mid g(x) \equiv g(\alpha) \mod f_i\}$.

Translates into k-linear equations for the coefficients of g.

The principal subfields theorem

The set $\{L_2, \ldots, L_r\}$ is independent of the choice of $\tilde{K}$.

L_i is the field corresponding to the minimal block containing $\{\alpha, \phi_i(\alpha)\}$.

The intersection theorem

The subfield polynomial f_L of L is the minimal polynomial of α over L.

Lemma

Let L_1, L_2 be two subfields of K/k. Then $L_1 \subseteq L_2 \Leftrightarrow f_{L_2} \mid f_{L_1}$.

This easily proves

Theorem

$k \subseteq L \subseteq K \Rightarrow L = \bigcap_{i \in I} L_i$ *for some* $I \subseteq \{2, \ldots, r\}$.

Is $\{L_2, \ldots, L_r\}$ a minimal set with that property?

The generating subfields

Definition

- A set S of subfields is called **generating set**, if every subfield can be written as an intersection $\bigcap T$, where $T \subseteq S$.
- A subfield $k \subseteq L \subseteq K$ is called generating if one of the following equivalent conditions hold:
  1. $\bigcap_{L \subsetneq L' \subseteq K} L' \neq L$.
  2. There is precisely one $\tilde{L} \subseteq K$ such that L is a maximal subfield of $\tilde{L}$.

Theorem

S is a generating set $\Leftrightarrow$ every generating subfield is in S.

$$L \in S \text{ generating } \Leftrightarrow L \neq \bigcap \{L' \in S \mid L \subsetneq L'\}.$$

Complexity

We can compute the generating subfields of K/k when we are able to

1. factor polynomials in $K[x]$.
2. do linear algebra over k.

Theorem

Let K/k be a finite extension of number fields. Then there is a polynomial time algorithm (in the degree and logarithmic size of the coefficients) which computes the generating subfields of K/k.

Intersections of generating subfields

Given: K/k, generating set $S = \{L_1, \ldots, L_r\}$, $K \notin S$.

1. Print K.
2. Call NextSubfields$(S, K, (0, \ldots, 0), 0)$.

function NextSubfields(S, L, e, s)

for all i with $e_i = 0$ and $s < i \leq r$ **do**

1. $M := L \cap L_i$.
2. Compute $\tilde{e} := (\tilde{e}_1, \ldots, \tilde{e}_r)$, where $\tilde{e}_j = 1 \Leftrightarrow M \subseteq L_j$.
3. **if** $\tilde{e}_j \leq e_j$ for all $1 \leq j < i$ then
   1. Print M.
   2. Call NextSubfields$(S, M, \tilde{e}, i)$.

Invariant: s minimal with $L = \bigcap\{L_i \mid 1 \leq i \leq s, e_i = 1\}$.

Running time

Let m be the number of subfields and $S = \{L_1, \ldots, L_r\}$.
There are exactly m calls to NextSubfields.

Theorem

The intersection algorithm computes all subfields by computing at most mr intersections and at most mr^2 inclusion tests.

Theorem

Let K/k be an extension of number fields. Then all subfields can be computed in polynomial time in the degree, the size of the coefficients, and the number of subfields.

Polynomial time does not imply efficient in practice!

Summary

Let $K = k(\alpha)$ be separable of degree n, f minpoly of α.

- Factor $f = (x - \alpha) \cdot f_2 \cdots f_r \in K[x]$.
- Solve $(r - 1)$ linear systems of equations.

yields set S of generating subfields.

- All subfields are intersections of those in S.
- Number of intersections to compute is linear in the output.
- Very easy to implement (if we can factor in $K[x]$ and do linear algebra in k).

Improvements for implementations

Bottle neck

In the number field case: The factorization of $f \in K[x]$.

Idea

Replace K by a larger field $\tilde{K}$, e.g. $\tilde{K} = \mathbb{Q}_p$, where factoring is easier.

Example

Assume $k = \mathbb{Q}$ and choose a prime p such that
$f \equiv (x - a_1) \cdots (x - a_n) \bmod p$. Then Hensel lifting gives factorization:

$$f \equiv \prod_{i=1}^{n} (x - \alpha_i) \bmod p^k \text{ for } k \in \mathbb{N}.$$

Factoring is cheap, but how to do linear algebra with approximations?

The LLL algorithm

Let $\beta = \sum_{j=0}^{n-1} c_i \alpha^i$ be in the kernel of $\Phi_i - id$, i.e.

$$\sum_{j=1}^{n-1} c_i(\alpha_1^j - \alpha_i^j) = 0.$$

$$B := \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ \alpha_1 - \alpha_i & \dots & \alpha_1^{n-1} - \alpha_i^{n-1} & p^k \end{pmatrix}$$

The columns of B generate a lattice which need to be LLL-reduced.

Some remarks and questions

- Use LLL with removals (like in factoring).
- Better basis: $1/f'(\alpha), \alpha/f'(\alpha), \ldots, \alpha^{n-1}/f'(\alpha)$.
- Some work: Find good bound for Gram-Schmidt length bound.
- Can compute examples in higher degree which were impossible before (In worst case exponential search algorithm).
- Special algorithm to compute all quadratic subfields.
- Maximal subfields are certainly generating subfields.

Question

Is there an efficient algorithm to compute all minimal subfields?
In group theory: All maximal subgroups containing the point stabilizer?

The p-adic case

$K = \mathbb{Q}_p(\alpha)$,

where $k = \mathbb{Q}_p$, $f \in \mathbb{Q}_p[x]$ irreducible, and be α a root of f.

Goal: Compute all subfields of K.
Assume (by Krasner's lemma) that $f \in \mathbb{Z}[x] \Rightarrow$ our input is exact.

$f = (x - \alpha) \cdot f_2 \cdots f_r$,

but we can compute f_i only modulo p^k.

How to solve correctly the linear system of equations, if the input is only given by approximations? Same problem for intersections.

First implementation done in a bachelor thesis under my supervision.

The database of number fields (with Gunter Malle)

- http://galoisdb.math.uni-paderborn.de
- Database of number fields up to degree 23 (for the public 19)
- Covers all groups in that range except $L_2(16):2$, M_{23}, 11 groups in degree 21, 5 groups in degree 22.

Minimal discriminants

- All minimal discriminants up to degree 7.
- All minimal discriminants for imprimitive fields in degree 8, e.g. work by Cohen, Diaz y Diaz, and Olivier (quartic subfield).
- Only partial results for imprimitive degree 9 and 10 fields.