Generating Subfields

Mark van Hoeij
(joint work with Jürgen Klüners)

Let K/k be a finite separable field extension of degree n. We describe an algorithm that computes all subfields of K that contain k. We assume that a primitive element α of K/k is given as well as its minimal polynomial $f \in k[x]$. The main result is that all subfields can be presented as intersections of a small number of subfields, and that those subfields can be calculated efficiently. The concepts of principal and generating subfields are introduced.

1. The main theorem

Let $\tilde{K}$ be a field containing K and $f = f_1 \cdots f_r$ be the factorization of f over $\tilde{K}$ where the $f_i \in \tilde{K}[x]$ are irreducible and monic, and $f_1 = x - \alpha$. We define the fields $\tilde{K}_i := \tilde{K}[x]/(f_i)$ for $1 \leq i \leq r$. We denote elements of K as $g(\alpha)$ where g is a polynomial of degree $< n$, and define for $1 \leq i \leq r$ the embedding

$$\phi_i : K \to \tilde{K}_i, \qquad g(\alpha) \mapsto g(x) \bmod f_i.$$

Note that ϕ_1 is just the identity map $id : K \to \tilde{K}$. We define for $1 \leq i \leq r$:

$$L_i := \operatorname{Ker}(\phi_i - id) = \{g(\alpha) \in K \mid g(x) \equiv g(\alpha) \bmod f_i\}.$$

The L_i are closed under multiplication, and hence fields, since $\phi_i(ab) = \phi_i(a)\phi_i(b) = ab$ for all $a, b \in L_i$.

Theorem 1. *If L is a subfield of K/k then L is the intersection of L_i, $i \in I$ for some $I \subseteq \{1, \ldots, r\}$.*

Proof. Let f_L be the minimal polynomial of α over L. Then f_L divides f since $k \subseteq L$, and $f_L = \prod_{i \in I} f_i$ for some $I \subseteq \{1, \ldots, r\}$ because $L \subseteq \tilde{K}$. We will prove

$$L = \{g(\alpha) \in K \mid g(x) \equiv g(\alpha) \bmod f_L\} = \bigcap_{i \in I} L_i.$$

If $g(\alpha) \in L$ then $h(x) := g(x) - g(\alpha) \in L[x]$ is divisible by $x - \alpha$ in $K[x]$. The set of polynomials in $L[x]$ divisible by $x - \alpha$ is (f_L) by definition of f_L. Then $h(x) \equiv 0 \bmod f_L$ and hence $g(x) \equiv g(\alpha) \bmod f_L$. Conversely, $g(x) \bmod f_L$ is in $L[x]$ (mod f_L) because division by f_L can only introduce coefficients in L. So if $g(x) \equiv g(\alpha) \bmod f_L$ then $g(\alpha) \in K \cap L[x] = L$.

By separability and the Chinese remainder theorem, one has $g(x) \equiv g(\alpha)$ mod f_L if and only if $g(x) \equiv g(\alpha) \bmod f_i$ (i.e. $g(\alpha) \in L_i$) for every $i \in I$. □

We can choose for $\tilde{K}$ any field that contains K (the set $S := \{L_1, \ldots, L_r\}$ is independent of this choice). The most convenient choice is to take $\tilde{K} = K$, but in some situations it might be better to let $\tilde{K}$ be some completion of K (this would save time on the factorization of f over $\tilde{K}$, but it complicates computing the Ker in the definition of L_i since this would then have to be done with LLL techniques instead of linear algebra over k. So if one has very efficient factoring code [3] then taking $\tilde{K} = K$ might still be the best choice).

Definition 1. *We call the fields $L_1, \dots, L_r$ the* principal subfields *of K/k. A set S of subfields of K/k is called a* generating set *of K/k if every subfield of K/k can be written as $\bigcap T$ for some $T \subseteq S$. Here $\bigcap T$ denotes the intersection of all $L \in T$, and $\bigcap \emptyset$ refers to K. A subfield L of K/k is called a* generating subfield *if it satisfies the following equivalent conditions*

(1) *The intersection of all fields L' with $L \subsetneq L' \subseteq K$ is not equal to L.*
(2) *There is precisely one field $L \subsetneq \tilde{L} \subseteq K$ for which there is no field between L and $\tilde{L}$ (and not equal to L or $\tilde{L}$).*

The field $\tilde{L}$ in condition (2) is called *the field right above L*. It is clear that $\tilde{L}$ is the intersection in condition (1), so the two conditions are equivalent.

The field K is a principal subfield but not a generating subfield. A maximal subfield of K/k is a generating subfield as well. Theorem 1 says that the principal subfields form a generating set. By condition (1), a generating subfield can not be obtained by intersecting larger subfields, and must therefore be an element of every generating set. In particular, a generating subfield is also a principal subfield.

If S is a generating set, and we remove every $L \in S$ for which $\bigcap\{L' \in S | L \subsetneq L'\}$ equals L, then what remains is a generating set that contains only generating subfields. It follows that

Proposition 1. *S is a generating set if and only if every generating subfield is in S.*

Suppose that K/k is a finite separable field extension and that one has polynomial time algorithms for factoring over K and for linear algebra over k (for example when $k = \mathbb{Q}$). Then applying Theorem 1 with $\tilde{K} = K$ yields a generating set S with $r \leq n$ elements in polynomial time. We may want to minimize r by removing all elements of S that are not generating subfields. Then $r \leq n - 1$. In principle there are 2^r subsets of S to be considered, which may be substantially more than the number of subfields. So we design the algorithm in Section 2 in such a way that it finds each subfield only once. This way, when S is given, the cost of computing all subfields is proportional to the number of subfields.

2. Intersections

In this section we describe an algorithm to compute all subfields of K/k by intersecting elements of a generating set $S = \{L_1, \dots, L_r\}$. The complexity is proportional to the number of subfields of K/k. Unfortunately there exist families of examples where this number is more than polynomial in n.

To each subfield L of K/k we associate a tuple $e = (e_1, \dots, e_r) \in \{0, 1\}^r$, where $e_i = 1$ if and only if $L \subseteq L_i$.

Algorithm AllSubfields
Input: A generating set $S = \{L_1, \ldots, L_r\}$ for K/k.
Output: All subfields of K/k.

(1) Let $e := (e_1, \ldots, e_r)$ where $e_1 = 1$ if $L_i = K$ and $e_i = 0$ otherwise.
(2) ListSubfields $:= [K]$.
(3) Call NextSubfields$(S, K, e, 0)$.
(4) Return ListSubfields.

The following function returns no output but appends elements to ListSubfields, which is used as a global variable. The input consists of a generating set, a subfield L, its associated tuple $e = (e_1, \ldots, e_r)$, and the smallest integer $0 \leq s \leq r$ for which $L = \bigcap\{L_i \mid 1 \leq i \leq s,\ e_i = 1\}$.

Algorithm NextSubfields
Input: S, L, e, s.

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

(1) Let $M := L \cap L_i$.
(2) Let $\tilde{e}$ be the associated tuple of M.
(3) **If** $\tilde{e}_j \leq e_j$ for all $1 \leq j < i$ **then** append M to ListSubfields and call NextSubfields$(S, M, \tilde{e}, i)$.

Subfields that are isomorphic but not identical are considered to be different in this text. Let m be the number of subfields of K/k. Since S is a generating set, all subfields occur as intersections of $L_1, \ldots, L_r$. The condition in Step (3) in Algorithm NextSubfields holds if and only if M has not already been computed before. So each subfield will be placed in ListSubfields precisely once, and the total number of calls to Algorithm NextSubfields equals m. For each call, the number of i's with $e_i = 0$ and $s < i \leq r$ is bounded by r, so the total number of intersections calculated in Step (1) is $\leq rm$. Step (2) involves testing which L_j contain M. Bounding the number of j's by r, the number of subset tests is $\leq r^2 m$.

Theorem 2. *Given a generating set for K/k with r elements, Algorithm AllSubfields returns all subfields by computing at most rm intersections and at most $r^2 m$ subset tests, where m is the number of subfields of K/k.*

Thus the cost of computing all subfields is bounded by a polynomial times the number of subfields.

References

[1] Preliminary implementation: `http://www.math.fsu.edu/~hoeij/papers/subfields`
[2] J. Klüners, M. Pohst, *On Computing Subfields*, J. Symb. Comput., **24** (1997), 385–397.
[3] K. Belabas, *A relative van Hoeij algorithm over number fields*, J. Symb. Comput., **37** (2004), 641–668.