Computation of the Triangular Representation of a Splitting Field

SAGE DAYS 10

Guénaël Renault

1: INRIA SALSA Project / LIP6 / University Paris 06, France

October 10–15, 2008, LORIA Nancy, France

Part I

Introduction

The Splitting Field of a Polynomial

Let $f \in \mathbb{Z}[x]$ be a monic irreducible polynomial with degree n and $\underline{\alpha} = \{\alpha_1, \ldots, \alpha_n\}$ a set of its roots.

Aim

Compute a representation of $\mathbb{Q}_f = \mathbb{Q}(\underline{\alpha})$ the Splitting Field of f.

This corresponds to the normal closure of the number field defined by the polynomial f.

Representations of The Splitting Field of a Polynomial

Representation of $\mathbb{Q}_f$: as a simple extension of degree $N = |G|$ (the Galois groupe of f is G)

$\Rightarrow$ Representation of the roots needs polynomials of degree N

Representation of $\mathbb{Q}_f$: as a tower of extensions defined by the quotient algebra

$$\mathbb{Q}[x_1, \ldots, x_n]/\mathcal{I}$$

where $\mathcal{I}$ is the splitting ideal defined by

The kernel of the valuation map in $\underline{\alpha}$

$$\mathcal{I} = \{R \in \mathbb{Q}[x_1, \ldots, x_n] \,|\, R(\underline{\alpha}) = 0\}$$

$\Rightarrow$ Recursive definition of the roots

(Note: $\mathcal{I}$ depends on the numbering of the roots $\underline{\alpha}$)

Representations of The Splitting Field of a Polynomial

Representation of $\mathbb{Q}_f$ as a tower of extensions

$$
\begin{tikzcd}
\mathbb{Q}(\alpha_1,\ldots,\alpha_n) \arrow[d, dash, "g_n = x_n^{d_n} + r_n(x_1,\ldots,x_n)"'] \\
\mathbb{Q}(\alpha_1,\ldots,\alpha_{n-1}) \arrow[d, dash, "g_{n-1} = x_{n-1}^{d_{n-1}} + r_{n-1}(x_1,\ldots,x_{n-1})"'] \\
\mathbb{Q}(\alpha_1,\ldots,\alpha_{n-2}) \arrow[d, dash, dotted] \\
\mathbb{Q}(\alpha_1) \arrow[d, dash, "g_1 = f(x_1)"'] \\
\mathbb{Q}
\end{tikzcd}
$$

Computations in this Quotient Algebra

The ideal $\mathcal{I}$ is generated by the following triangular set $\mathcal{T}$

$$
\begin{array}{rcll}
g_1(x_1) & = & x_1^{d_1} + r_1(x_1) & \deg_{x_1}(r_1) < d_1 \\
g_2(x_1, x_2) & = & x_2^{d_2} + r_2(x_1, x_2) & \deg_{x_2}(r_2) < d_2 \\
\ldots & & & \\
g_n(x_1, \ldots, x_n) & = & x_n^{d_n} + r(x_1, \ldots, x_n) & \deg_{x_n}(r_n) < d_n
\end{array}
$$

$$g_i(\alpha_1, \ldots, \alpha_{i-1}, x_i)$$

minimal polynomial of α_i over $\mathbb{Q}(\alpha_1, \ldots, \alpha_{i-1})$.

Gröbner basis (LEX $x_1 < x_2 < \ldots < x_n$) $\Rightarrow$ computations $\mathbb{Q}[x_1, \ldots, x_n]/\mathcal{I}$.

The Galois Group in this Representation

The $\mathbb{Q}$-automorphism group of $\mathbb{Q}_f$ can be represented by a subgroup G_f of S_n, the Galois group of f:

$$\begin{array}{ccc} \mathbb{Q}_f = \mathbb{Q}(\underline{\alpha}) & \longrightarrow & \mathbb{Q}_f = \mathbb{Q}(\underline{\alpha}) \\ \alpha_i & \longmapsto & \alpha_j \end{array}$$

The permutation group G_f stabilizes the ideal $\mathcal{I}$:

$$G_f = \{\sigma \in S_n \,|\, \forall R \in \mathcal{I}, \sigma \cdot R := R(x_{\sigma(1)}, \ldots, x_{\sigma(n)}) \in \mathcal{I}\}$$

(Note: G_f depends on the numbering of the roots $\underline{\alpha}$)

Computation of the Set $\mathcal{T}$

Direct methods

- Successive factorizations (Kronecker-Tchebotarev method)
- Resolvents computations (Arnaudiès, Aubry, Ducos, Valibouze ...)

$\Rightarrow$We can compute G_f from $\mathcal{T}$

Driven methods

$\Rightarrow$Very efficient implementation for the computation of the G_f action over $\underline{\alpha}$ (Magma, Kash).

Problematic

How to use the knowledge of G_f in order to efficiently compute $\mathcal{T}$?

Computation of the Set $\mathcal{T}$

Driven methods

$\Rightarrow$Interpolation method, the action of G_f over p-adic approximations of $\underline{\alpha}$ is known [Yokoyama 97][Lederer 05]: generic

$\Rightarrow$[R., Yokoyama ANTS'06]: Interpolation based on linear algebra with a careful treatment on reducing computational difficulty (computation scheme).

$\Rightarrow$ [R., Yokoyama ISSAC'08]: Linear algebra $\rightarrow$ Lagrange Formulae and multi-modular strategy.

Part II

Computation Scheme

The generic shape of g_i's and $\mathcal{T}$

From the knowledge of G_f we obtain:

$$
\begin{tikzcd}
\mathbb{Q}(\alpha_1,\dots,\alpha_n) \arrow[d, dotted, dash] \arrow[d, dash, bend right=60, "\mathrm{Stab}_{G_f}[1,\dots,i]"'] \arrow[dd, dash, bend left=60, "\mathrm{Stab}_{G_f}[1,\dots,i-1]"] \\
\mathbb{Q}(\alpha_1,\dots,\alpha_i) \arrow[d, dash, "g_i=x_i^{d_i}+r_i(x_1,\dots,x_i)"'] \\
\mathbb{Q}(\alpha_1,\dots,\alpha_{i-1}) \arrow[d, dotted, dash] \\
\mathbb{Q}(\alpha_1) \arrow[d, dash, "g_1=f(x_1)"'] \\
\mathbb{Q}
\end{tikzcd}
$$

The generic shape of g_i's and $\mathcal{T}$

From the knowledge of G_f we obtain:

$$d_i = |\operatorname{Stab}_{G_f}([1,\ldots,i-1])|/|\operatorname{Stab}_{G_f}([1,\ldots,i])|\,.$$

$$\Downarrow$$

$$g_i = x_i^{d_i} + \sum_{0\leqslant k_j<d_j} c\, x_1^{k_1} x_2^{k_2}\cdots x_i^{k_i}$$

With this generic shape, there are $d_1 d_2 \cdots d_i$ indeterminate coefficients to compute for identifying g_i ([Yokoyama 97], [Lederer 05]).

$\mathcal{T}$ contains n polynomials with $\simeq |G_f|$ indeterminate coefficients

The principle of the computation scheme

$\Rightarrow$[R., Yokoyama ANTS'06] [R. ISSAC'06]

Definition

Be given a permutation group G, a computation scheme consists of a pre-computed data that guides the computation of the splitting field of a polynomial with Galois group G.

- reducing the number of indeterminates to compute
- reducing the number of polynomials to compute

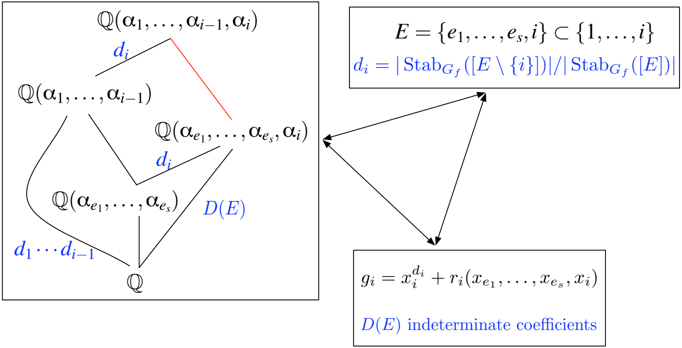
$\Rightarrow c(G)$ will denote the number of coefficients to compute in $\mathcal{T}$

Sparse shape of g_i

i-relation

$E = \{e_1 < \ldots < e_s < i\} \subset \{1, \ldots, i\}$

$$\exists r_i \in \mathbb{Q}[x_{e_1}, \ldots, x_{e_s}, x_i] : \quad \alpha_i^{d_i} + r_i(\underline{\alpha}) = 0 \text{ and } \deg_{x_i}(r_i) < d_i$$



i-relations with minimal $D(E) \Rightarrow$ minimal number of coefficients for g_i.

Avoiding some computations

Techniques

From a polynomial $g \in \mathcal{T}$ already computed it is possible to deduce a new one by using the knowledge of G_f:

- By action of G_f over g (Transporter technique)
- By *divided differences* of g (generalized Cauchy moduls)

Avoiding some computations : (i,j)-transporters

$E_i = \{e_1 < e_2 < \cdots < e_s = i\}$ is an i-relation and $j \in \{i+1,\ldots,n\}$.

Definition

$\sigma \in G_f$ is a (i,j)-transporter if $d_i = d_j$ and

$$\sigma(i) = j \text{ with } j = \max(\{\sigma(e) \;:\; e \in E_i\})$$

$$d_i = d_j = d \qquad \begin{cases} g_1 = x_1^{d_1} + \ldots \\ \vdots \\ g_i(X_{E_i}) = x_i^d + r(X_{E_i}) \\ \vdots \\ g_j = x_j^d + \ldots = \sigma . g_i \\ \vdots \end{cases}$$

σ

Avoiding some computations : **Cauchy moduls**

Let $\mathcal{O} = \{i_1 = i < i_2 < \cdots < i_{d_i}\}$ be the orbit of i under the action of $\mathrm{Stab}_{G_f}([1, \ldots, i-1])$.

Definition

The generalized Cauchy moduls of g_i are

$$
\begin{aligned}
c_1(g_i)(\ldots, x_{i_1}) &= g_i \\
c_2(g_i)(\ldots, x_{i_2}) &= \frac{c_1(g_i)(x_{i_2}) - c_1(g_i)(x_{i_1})}{(x_{i_2} - x_{i_1})} \\
&\vdots \\
c_{d_i}(g_i)(\ldots, x_{i_{d_i}}) &= \frac{c_{d_i-1}(g_i)(x_{i_{d_i}}) - c_{d_i-1}(g_i)(x_{i_{d_i}-1})}{(x_{i_{d_i}} - x_{i_{d_i-1}})}
\end{aligned}
$$

$c_j(g_i) \in \mathbb{Q}[x_1, \ldots, x_{i_j}] \cap \mathcal{I}$ monic in x_{i_j} and $\deg_{i_j}(c_j(g_i)) = d_i - j + 1$.
$c_j(g_i)(\underline{\alpha}, x_{i_j})$ is a univariate polynomial which vanishes on α_{i_j}.

Avoiding some computations : **Cauchy moduls**

$c_j(g_i) \in \mathbb{Q}[x_1, \ldots, x_{i_j}] \cap \mathcal{I}$ monic in x_{i_j} and $\deg_{i_j}(c_j(g_i)) = d_i - j + 1$.
$c_j(g_i)(\underline{\alpha}, x_{i_j})$ is a univariate polynomial which vanishes on α_{i_j}.

$i, j \in \mathcal{O}$

g_j is a divided difference of g_i.

$$\begin{cases} g_1 = x_1^{d_1} + \ldots \\ \vdots \\ g_i = x_i^{d_i} + \ldots \\ \vdots \\ g_j = x_j^{d_j} + \ldots \\ \vdots \end{cases}$$

Computation Scheme, Conclusion

Conclusion

Given G_f we can obtain a sparse shape for each polynomial g_i or a technique to obtain it without computation:

1: Compute d_i.

2: Search for generalized Cauchy moduls.

3: Search for a transporter.

4: If necessary, compute an i-relation E_i with minimal $D(E_i)$.

We denote by $c(G_f) = \sum D(E_i)$ the total number of indeterminate coefficients of polynomials in $\mathcal{T}$ we have to compute.

- The integer $c(G_f)$ is not an invariant for a conjugacy class.
- A representative with minimal *c-size* can be pre-computed and stored with its attached computation scheme.

Computation Scheme, example

Example : $G_2 \simeq 8T_{44} \simeq [2^4]S_4$, $|G_2| = 384$, imprimitive

$$G_2 = \langle (2,1), (8,6,4,1)(7,5,3,2), (8,1)(7,2) \rangle$$

Transporters

Cauchy

$$\begin{cases} g_1 = x_1^8 + \dots \\ g_2 = x_2^1 + \dots \\ g_3 = x_3^6 + \dots \\ g_4 = x_4^1 + \dots \\ g_5 = x_5^4 + \dots \\ g_6 = x_6^1 + \dots \\ g_7 = x_7^2 + \dots \\ g_8 = x_8^1 + \dots \end{cases}$$

$$\mathrm{E}_2 = \{2, 1\} \longrightarrow \mathrm{D}(\mathrm{E}_2)$$

$$\mathrm{c}(\mathrm{G}_2) = 8$$

Computation Scheme, example

Example : $G_2 \simeq 8T_{44} \simeq [2^4]S_4$, $|G_2| = 384$, imprimitive

$$G_2 = \langle (2,1), (8,6,4,1)(7,5,3,2), (8,1)(7,2) \rangle$$

Generic

1264 coefficients to compute

$$\left\{ \begin{array}{ll} g_1 = x_1^8 + \dots & 8 \\ g_2 = x_2^1 + \dots & 8 \\ g_3 = x_3^6 + \dots & 8\text{x}6 \\ g_4 = x_4^1 + \dots & 8\text{x}6 \\ g_5 = x_5^4 + \dots & 8\text{x}6\text{x}4 \\ g_6 = x_6^1 + \dots & 8\text{x}6\text{x}4 \\ g_7 = x_7^2 + \dots & 8\text{x}6\text{x}4\text{x}2 \\ g_8 = x_8^1 + \dots & 8\text{x}6\text{x}4\text{x}2 \end{array} \right.$$

Part III

Modular method for computing $\mathcal{T}$

Computation of a candidate: inputs

$\Rightarrow$From the knowledge of G_f we know a computation scheme, thus a subset

$$\mathcal{S} := \{g_{i_1}, \ldots, g_{i_k}\} \subset \mathcal{T}$$

of polynomials to compute and techniques for obtaining the others.

$\Rightarrow$To g in $\mathcal{S}$ corresponds an *i*-relation $E = \{e_1 < e_2 < \cdots < e_s = i\}$:

$$g = x_i^{d_i} + r(x_{e_1}, x_{e_2}, \ldots, x_i)$$

D(E) indeterminate coefficients to compute

Computation of a candidate: interpolation

From the action of G_f over $\underline{\alpha} \mod p^k$ ([Yokoyama 97], [Geissler, Klüners 00]) we can reconstruct $g \mod p^k$ by interpolation.

[R., Yokoyama ANTS'06]:

- $g(\underline{\beta}) = 0 \mod p^k, \forall \underline{\beta} \in G_f \cdot \underline{\alpha} \Rightarrow D(E)$ linear equations

$$\begin{pmatrix} & & \\ & D(E)^2 & \\ & & \end{pmatrix}$$

$$D(E) = d_{e_1} d_{e_2} \cdots d_i$$

Computation of a candidate: interpolation

From the action of G_f over $\underline{\alpha} \mod p^k$ ([Yokoyama 97], [Geissler, Klüners 00]) we can reconstruct $g \mod p^k$ by interpolation.

[R., Yokoyama ISSAC'08]:

- We can directly apply [Dahan, Schost 04] on sub-triangular set,
- and the formula can be established by Galois theory

$$g = \sum_{\sigma \in G_f // \mathrm{Stab}_{G_f}(E_i \backslash \{i\})} \left(\prod_{j \in E_i \backslash \{i\}} \prod_{\beta \in B(\sigma, j, E_i)} \frac{x_j - \beta}{\alpha_{\sigma(j)} - \beta} \right) \prod_{\beta \in B(\sigma, i, E_i)} \frac{x_i - \beta}{\alpha_{\sigma(i)} - \beta}$$

Correctness test

$\Rightarrow$After rational reconstruction, how to check the result ?

Theoretical Bounds: ([Lederer 05] for a generic shape of ideal $\mathcal{I}$).

$$d(E_i)\binom{d_1-1}{k_1}\nu^{d_1-1-k_1}\dots\binom{d_s}{k_s}\nu^{d_s-k_s}\mathbb{B}.$$

where ν and $\mathbb{B}$ are bounds computed from numerical app. roots of f

Normal Form Computation: Let h_i be the rational reconstruction of g_i mod p^k. Assume that $g_1,\dots,g_{i-1}$ are already computed.

Theorem. We have the following equivalence

$$h_i = g_i \Leftrightarrow NF_{\{g_1,\dots,g_{i-1},h_i\}}(\text{CauchyMod}_i(f)) = 0\,.$$

First comparisons

Complexity:
Interpolation based on lin. algebra $c(G)^\omega \rightarrow$ Lagrange formulae $c(G)^2$.

Experiments: Magma 2.14-13 (1.5GHz Intel Pentium 4, GNU/Linux), $k = 10$, f splits completely modulo p. All timings in seconds.

group	gen.	$c(G)$	Lagrange	NF	Total	Magma	Lederer
$7T_6$	3611	1260	47.5	3.04	52.5	$>$	1508.3
$8T_{32}$	624	$96+96$	0.55	0.14	0.72	33.5	12.5
$8T_{42}$	1008	$24+24$	0.05	0.02	0.1	17.9	20.08
$8T_{47}$	1008	24	0.03	0.0	0.5	422.3	238.3
$9T_{25}$	828	$27+324$	3.41	0.33	3.77	106.1	67.9
$9T_{27}$	3096	504	7.98	105.49	116.3	$>$	397.3
$9T_{31}$	2178	18	0.01	0.03	0.5	$>$	403.3
$9T_{32}$	9648	$1512+1512$	142.17	752.4	905.4	$>>$	1967.1

$(>, >>)$: we wait at least $(600, 2000)$ seconds

Part IV

Conclusion

SAGE possibilities

- KASH/KANT : Galois action over *p*-adic approximations of the roots $\underline{\alpha}$
- GAP : Computation Scheme
- Singular : Multivariate polynomials and normal forms computations

$\Rightarrow$This algorithm could be easily implemented in SAGE.