Computation Schemes for Splitting Fields of Polynomials

ISSAC'09

Sébastien Orange[1], Guénaël Renault[2] and Kazuhiro Yokoyama[3]

1: Université du Havre, France
2: UPMC, INRIA/LIP6 SALSA Project, France
3: Department of Mathematics, Rikkyo University, Japan

July, 2009, Seoul, Korea

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.

Representation of $\mathbb{Q}_f$:

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

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

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

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

The Splitting Field of a Polynomial

The splitting ideal $\mathcal{I}$ is generated by the following triangular Gröbner basis $\mathcal{T}$ (LEX $x_1 < x_2 < \ldots < x_n$)

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

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

$$\begin{cases} g_1(x_1) = x_1^{d_1} + r_1(x_1) \\ g_2(\alpha_1, x_2) = x_2^{d_2} + r_2(\alpha_1, x_2) \\ \vdots \\ g_n(\alpha_1, \alpha_2, \ldots, x_n) = x_n^{d_n} + r(\alpha_1, \alpha_2, \ldots, x_n) \end{cases}$$

$\mathbb{Q}$

$\mathbb{Q}(\alpha_1)$

$\mathbb{Q}(\alpha_1, \alpha_2)$

$\vdots$

$\mathbb{Q}(\alpha_1, \alpha_2, \ldots, \alpha_{n-1})$

$\mathbb{Q}_f = \mathbb{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n)$

The Galois Group of a Polynomial

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 $\mathcal{I}$:

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

The variety of $\mathcal{I}$ is defined by G_f action:

$$V(\mathcal{I}) = G_f \cdot (\alpha_1, \ldots, \alpha_n) = \{(\alpha_{\sigma(1)}, \ldots, \alpha_{\sigma(n)}) \,|\, \sigma \in G_f\}$$

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

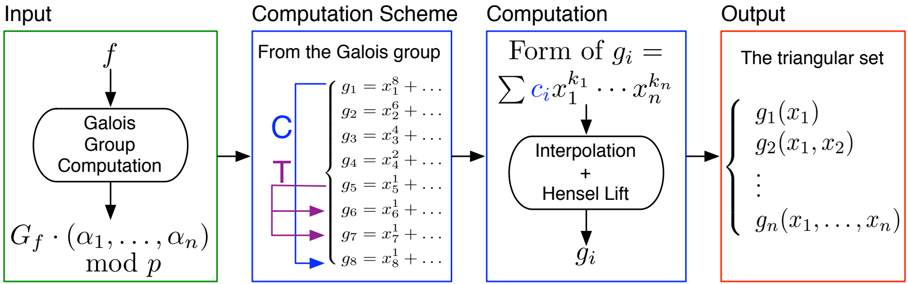
Related works

How to use some knowledges about the Galois action in order to compute efficiently the splitting field?

- Yokoyama, A modular method for computing the Galois groups of polynomials. MEGA 1996
- Fernandez-Ferreiros, Gomez-Molleda, Gonzalez-Vega, Partial solvability by radicals, ISSAC 2002.
- Lederer, M., Explicit constructions in splitting fields of polynomials. 2004
- R., Yokoyama, A modular method for computing the splitting field of a polynomial. ANTS 2006
- Diaz-Toca, Dynamic Galois Theory and Gröbner Basis, ACA 2008
- Valibouze, Sur les relations entre les racines d'un polynôme, Acta Arithmetica 2008.
- R., Yokoyama, Multi-modular algorithm for computing the splitting field of a polynomial, ISSAC 2008

Computation of the set $\mathcal{T}$

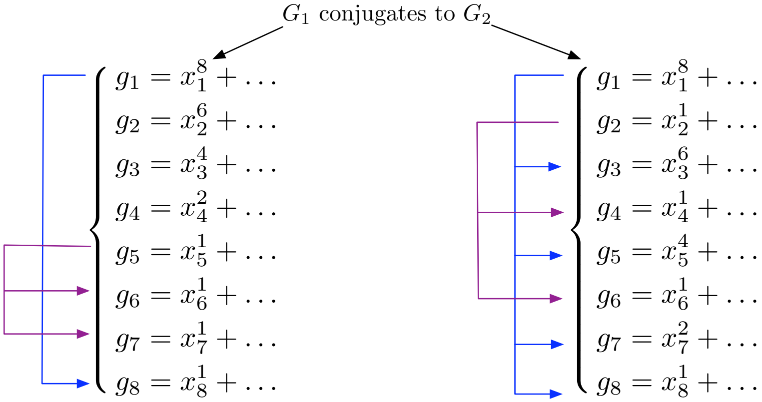
[R., Yokoyama ANTS'06][R., Yokoyama ISSAC'08]: Interpolation with a careful treatment on reducing computational difficulty (introduction of the computation schemes).



$\Rightarrow$The total efficiency of the computation relies on the computation scheme !

Computation Scheme: Problematic

Computation scheme is not an invariant of the conjugacy class of G_f!



632 coefficients to compute

8 coefficients to compute

$\Rightarrow$How to compute a conjugate of G_f with the best computation scheme?

Computation Scheme: Problematic

$\Rightarrow$How to compute a conjugate of G_f with the best computation scheme?

[R., Yokoyama ANTS'06]: brute force inspection of all the $|S_n : N_{S_n}(G_f)|$ ($\sim n!$ when G_f small) different conjugates of G_f.

- Combinatorial problem when $|G|$ is moderate ($|S_n : N_{S_n}(G_f)| >> |G|$), inefficient for $n > 7$
- Use of a data base to store the good conjugates

New contribution: Efficient algorithm for this computation.

- Based on the study of the orbits of G_f
- Theoretical studies for families of permutation groups
- We do not need of a data anymore for the computation of $\mathcal{I}$

Part II

Computation Scheme: Definition

The principle of the computation scheme

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

Be given a permutation group G, a computation scheme consists of a data that guides the computation of the splitting field of a polynomial with Galois group G by indeterminate coefficients method.

- reducing the number of polynomials to compute
- reducing the number of indeterminate coefficients to compute

$c(G)$ will denote the number of coefficients to compute in $\mathcal{T}$ by applying the corresponding computation scheme.

Shape of g_i's and $\mathcal{T}$

From the knowledge of G we obtain:

Fields	Galois Group	Orbits
$\mathbb{Q}$	G	$\{1,\ldots,n\}$
$d_1 = n$		
$\mathbb{Q}(\alpha_1)$	$\mathrm{Stab}_G(\{1\})$	$\{1\},\{i_1=2,\ldots,i_{d_2}\},\ldots$
d_2		
$\mathbb{Q}(\alpha_1,\alpha_2)$	$\mathrm{Stab}_G(\{1,2\})$	$\{1\},\{2\},\{i_1=3,\ldots,i_{d_3}\},\ldots$
$\vdots$	$\vdots$	$\vdots$

$$d_i = |\,\mathrm{Stab}_G(\{1,\ldots,i-1\})|/|\,\mathrm{Stab}_G(\{1,\ldots,i\})|\,.$$

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

Reducing the number of polynomials to compute: Cauchy modules

By inspecting the corresponding orbit of a polynomial g_i we may deduce another polynomial by generalized Cauchy module (divided difference) computation.

$$\begin{cases} g_1 = x_1^{d_1} + \dots \\ \vdots \\ g_i = x_i^{d_i} + \dots \\ \vdots \\ g_\ell = x_\ell^{d_\ell} + \dots \\ \vdots \end{cases}$$

Cauchy ($g_i \rightarrow g_\ell$)

Corresponding orbit ($g_i \rightarrow$) $\{j_1 = i < j_2 < \dots < j_k < \dots < j_{d_i}\}$

Let $\ell = j_k$

If $d_\ell = d_i - k + 1$ the Cauchy technique can be applied.

$\Rightarrow g_i(x_i, \alpha_{i-1}, \dots, \alpha_1)$ vanishes on α_ℓ for $\ell \in \{j_1 = i, j_2, \dots, j_{d_i}\}$.

Reducing the number of polynomials to compute: Transporters

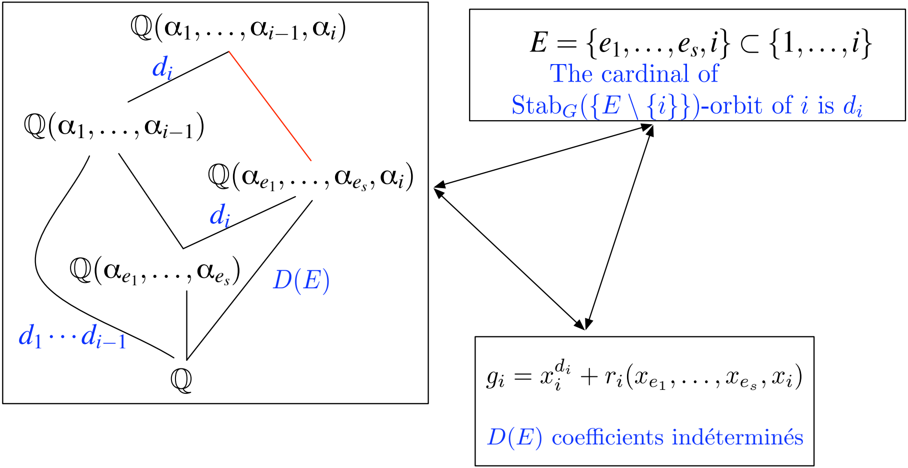
As G will be the stabilizer of the ideal generated by the set under construction, we can use its action.

$$\sigma \quad \begin{cases} g_1 = x_1^{d_1} + \dots \\ \vdots \\ g_i(X_{E_i}) = x_i^{d} + r(X_{E_i}) \\ \vdots \\ g_j = x_j^{d} + \dots = \sigma.f_i \\ \vdots \end{cases}$$

If $d_j = d_i = d$ and $\exists \sigma \in G$ s.t. $\sigma(i) = j$ and $\text{Max}(\sigma(E_i)) = j$ then f_j is deduced freely from f_i.

Reducing the number of coefficients to compute: *i*-relations

Generically g_i depends on $x_1, \ldots, x_1 \Rightarrow d_1 d_2 \cdots d_i$ indeterminate coefficients.



Computation Scheme: Definition

Definition

The computation scheme of the permutation group G is defined by the following data:

1. the degree d_i of the greatest variable in each polynomial in $\mathcal{T}$;
2. mathematical objects (shape) computed by Cauchy and Transporters techniques;
3. the minimal i-relation of each polynomial in $\mathcal{T}$ that can not be obtained by the preceding techniques.

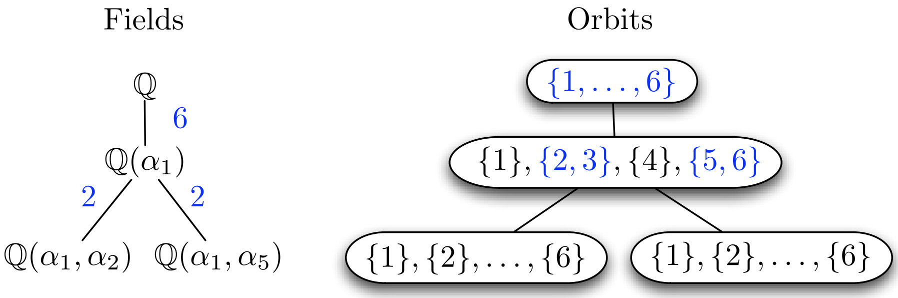
$\Rightarrow$Mainly depends on the orbits of the successive stabilizers of G.

Part III

Fast Computation of Computation Schemes: Orbits Tree

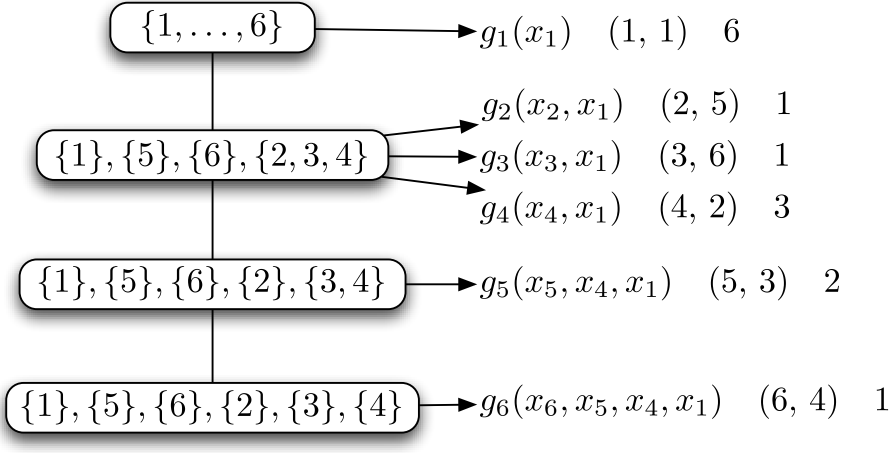
Tower of subfields and orbits tree

⇒We do not consider the linear factors ⇒*non redundant bases* of G.

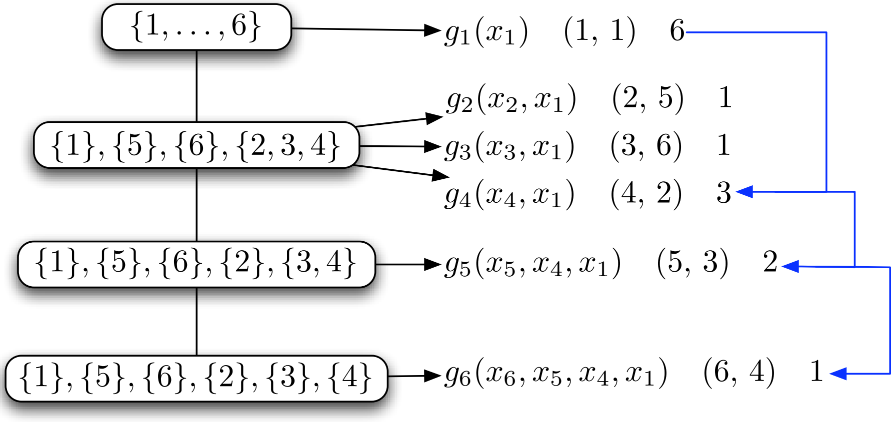


⇒We do not need to inspect the $|S_6 : N_{S_6}(G)| = 60$ different conjugates of G but only 2 branches of the orbit tree !

From a Branch to a Computation Scheme



From a Branch to a Computation Scheme



- Linear relations first $\Rightarrow$ best gain with the Cauchy technique
- We obtain sparse *i*-relation, but may be not the sparsest ones.

From the orbits tree to the best Computation Scheme

Sieving the orbits tree

In the same way, we can inspect the orbits tree for applying transporter technique and finding the sparsest *i*-relations.

Theoretical cost

The theoretical complexity is not so good : $poly(|G|)$, but

- The total complexity of the computation of the splitting field is not dominated by this step.
- For moderate size groups this complexity is $<< |S_n : N_{S_n}(G)|$.
- In practice, the algorithm is very efficient!

Cutting branches

Theoretical Tricks

By using group properties we can cut some branches in the tree (primitivity, transitivity, solvability, etc.)

- Alternate, Symmetric grps: hight transitivity $\Rightarrow$Cauchy technique.
- Cyclic groups: CS can be easily deduced without any computation
- Dihedral groups ([R. ISSAC'06]): idem
- Wreath products (This work): idem
- We can recursively use this results for cutting branches during the tree analysis.

Experimental results

Comp. Schemes Timings (Magma 2.14, 32 bits, Intel 2.5GHz)

For almost all the groups G of degree $\leqslant 15$ and $|G| \leqslant 10000$, the timings are too small (average < 1 second) to be really measured! Only few examples gave "*long*" timings (< 5 seconds). They appear when orbits tree has a large number of branches (< 750).

Splitting Fields Timings (Magma 2.14, 32 bits, Intel 2.5GHz)

group	$\|G\|$	Galois Grp	Comp. Schm.	Interpol+Lift	Magma	Lederer
$7T_6$	2520	0.06	0.00	52.5	$>$	1508.3
$8T_{32}$	96	0.16	0.00	0.72	33.5	12.5
$8T_{42}$	288	0.1	0.00	0.18	17.9	20.08
$8T_{47}$	1152	0.07	0.00	0.5	422.3	238.3
$9T_{25}$	324	0.42	0.01	4.07	106.1	67.9
$9T_{27}$	504	0.82	0.00	116.3	$>$	397.3
$9T_{31}$	1296	0.32	0.01	0.5	$>$	403.3
$9T_{32}$	1512	0.78	0.00	753.2	$>>$	1967.1

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

Conclusion

- Fill the gap between Galois group computation and the splitting field computation without data basis.
- Better knowledge for the use of the symmetries (extrem case) in the computation of Gröbner bases.