数式処理 *J.JSSAC (2002)*
Vol. 9, No. 1, pp. 53 - 79

論文

Radical Representation of Polynomial Roots

Hirokazu ANAI*
Secure Computing Laboratory
Fujitsu Laboratories Ltd.

Kazuhiro YOKOYAMA†
Graduate School of Mathematics,
Kyushu University

(RECEIVED 2001/12/13 REVISED 2002/6/6)

Abstract

We consider a fundamental question of Galois theory: how to express the roots of an irreducible polynomial f, $deg(f) > 4$ in terms of elements of the ground field by rational operations and radicals. In general, expressing the roots of f in terms of radicals is impossible when $\deg(f) > 4$. By Galois theory, however, we can test whether f is solvable by checking solvability of its Galois group. We will give a practical method for constructing a radical expression of the roots of f, when f is solvable, and report its experiment on a real computer.

1 Introduction

Our purpose is to express the roots of a polynomial $f(x)$ over the field $\mathbb{Q}$ of rational numbers in terms of radicals. For this purpose we present a *systematic* method, consisting of the following three parts derived quite naturally by Galois theory: (a) *construction of the Galois group G_f of f as a permutation group*, (b) *determining solvability of G_f and construction of its composition series*, (c) *expressing the roots in terms of radicals for solvable cases*. Without loss of generality, we assume that $f(x)$ is irreducible over $\mathbb{Q}$. This work shall provide a first springboard to further discussion and research along this direction.

In order to construct the radical representation of roots of f, it is necessary to obtain the Galois group G_f as a permutation group on all roots. As for (a), we employ the *direct* computation of the Galois group G_f proposed in Anai *et al.* [1], where "direct computation" means to obtain a concrete representation of G_f as a permutation group on the roots of

*anai@jp.fujitsu.com
†yokoyama@math.kyushu-u.ac.jp

f. We note that we can also employ the technique *p-adic approach* described in Yokoyama [26] which seems work very efficiently. For (b), we do not propose new algorithms since this part is not dominant in the whole procedure. Here we can use existing algorithms for group theory, see [21], [3] and [12]. Thus the first two parts have been solved and the remaining problem is how to execute (c) which we focus on.

Remark 1
For the Galois group computations there is an approach using classification tables of all transitive subgroups in the symmetric groups (see the recent survey in Hulpke [14]). For the solvability of a polynomial, there is a special algorithm by Landau & Miller [15] and its improvement by Yokoyama et al. [28]. Since these methods do not satisfy our requirement as they are, we do not employ them. However, to improve the efficiency of our proposed method, techniques used in their approach are useful. In particular, the direct method employed for (a) can be improved by p-adic technique in [26].

We shall deal with a *solvable* polynomial f only, since we assume that (a), (b) are resolved. So, the Galois group G_f is solvable and *a composition series* is already computed. By the Galois correspondence, there exists a subfield tower where every successive extension is a cyclic extension. Once a primitive element is expressed by radicals for each extension in the subfield tower, we obtain radical representations of roots of f. We pay special attention to the followings:

(c-1) Subfield computation: Methods in rather general setting are known, see Lazard & Valibouze [16], Dixon [8]. For our purpose, however, we utilize the speciality of the problem: We already know a composition series of G_f as permutation groups on the roots and a fixed primitive element β of the splitting field K_f. We present a method to obtain a corresponding subfield by using the polynomial associated with the *orbit* of a subgroup containing β. Several authors has presented some more specified methods to construct an intermediate field aiming at efficiency (see [13][17]).

(c-2) Radical representation of cyclic extensions: We provide an abstract procedure based on the *Lagrange resolvent* and devise concrete algorithms in order to make the abstract procedure practicable based on *elimination ideal computation* which can be efficiently executed by *basis-conversion* in Gröbner basis algorithms. (See Remark 2 for the reason why we introduce these notions.)

(c-3) Combining (c-1) and (c-2): To obtain radical representation of a root of $f(x)$, we have to combine algorithms presented in (c-1) and (c-2) adequately. This combination is achieved through the representation of the splitting field.

Since exact complexity analysis has not been done for the direct method employed for (a), we do not discuss complexity of our proposed method. Instead, efficiency of these methods is examined by experiments. We have implemented all algorithms mentioned

in this paper on a computer algebra system *Risa/Asir* [18] and computed a number of examples. Their performance and detailed results are listed in appendices. This shows that computing radical representation is practicable for polynomials whose splitting fields and Galois groups are computed by the direct method.

Remark 2

A different approach for computation of a radical representation is shown in Sturmfels [22] based on invariant theory of finite groups. In his approach, the Galois group G_f is also computed as a permutation group on the roots beforehand and each root is assigned to an indeterminate. (We denote them by X.) After computing invariants of G_f in the polynomial ring $\mathbb{Q}[X]$, the splitting field is computed by decomposition of the computed ideal, where factorization of polynomials is required. Then, a sequence of radical representation of elements invariant by subgroups in the composition series is computed, from which a sequence of radical representations of primitive elements of subfields is obtained by substitution.

Under an assumption that G_f is already known, his approach is applicable, and for another polynomial whose Galois group is permutationally isomorphic to G_f, we can use the same sequence of radical representations of invariant elements. So it might be an efficient complete procedure by combining with table-based methods in Remark 1. Comparison between our approach and Sturmfels' approach in both theory *and* practice *should be our next work.*

This paper is organized as follows. In §2 we sketch the mathematical basis. In §3 we give an outline of the whole procedure, and in §4 we present a procedure for finding subfields. In §5 we present procedures for radical representations of cyclic extension fields. In §6 we give a precise discussion on the whole procedure. In §7, we report our experiment on actual computation. Finally, in §8, we give our conclusion. In Appendix A, we list bench-marks for the method and in Appendix B, we list some results.

2 Mathematical Background

Here, we provide necessary notions and definitions of mathematical basis for radical representation of roots of polynomials. See van der Waerden [25].

2.1 What is radical representation

Let K be the field $\mathbb{Q}$ of rational numbers or its extension. For an algebraic element α over K, if α is expressed in terms of the elements of the ground field K by rational operations and radicals, we call the expression *a radical representation of* α over K. For example, $(1+\sqrt{-3})/2$ is a radical representation of a primitive 6-th root of unity. But, $\sqrt[6]{1}$

also gives a radical representation. (See [25].) From this, we recognize that there are two types for radical representation. We list up points.

⟨1⟩ An algebraic element is defined as a root of an irreducible polynomial over the ground field K. This polynomial is called the *minimal polynomial* of the element. Two algebraic elements are algebraic conjugate over K if and only if their minimal polynomials coincide.

⟨2⟩ In general, $\sqrt[m]{a}$ is not a single valued function. So the radical representation may present other elements if choice of the radicals appearing in it are changed.

⟨3⟩ It is very preferable that a radical representation of an algebraic element is to present its algebraic conjugate for any choice of the radicals appearing in the representation. (If a radical appears twice or more, it is assigned the same value for each time.)

If a radical representation satisfies the property described in ⟨3⟩, we call it *a strong radical representation*. Then $(1+\sqrt{-3})/2$ is a strong radical representation, but $\sqrt[6]{1}$ is not.

⟨4⟩ For an irreducible polynomial $f(x)$ over $\mathbb{Q}$, every its root has a strong radical representation if and only if the Galois group of f is solvable. By using the action of its Galois group, if one root of f has a *strong* radical representation, every root is also represented by the representation.

Hence, our target here is to compute a *strong radical representation* of *one root* for a given irreducible polynomial with solvable Galois group.

2.2 Galois theory

We briefly survey how a root of a solvable polynomial is represented in terms of radicals. Fix a monic irreducible polynomial $f(x)$ over $\mathbb{Q}$. Then, its Galois group G_f is solvable if and only if the following subgroup tower, called a *composition series*, exists:

$$G_f = G_0 \supset G_1 \supset \cdots \supset G_{r-1} \supset G_r = 1,$$

where G_i is a normal subgroup of G_{i-1} and G_{i-1}/G_i is a cyclic group with prime order p_i for each i. We assume that such a tower exists. By the Galois correspondence, there is a subfield tower

$$\mathbb{Q} = K_0 \subset K_1 \subset \cdots \subset K_{r-1} \subset K_r = K_f,$$

where K_f is the splitting field of f and K_i is the field consisting of all elements in K_f fixed by G_i. Then K_i is a cyclic extension of K_{i-1} and its extension degree $[K_i : K_{i-1}]$ is equal to p_i. Since K_i/K_{i-1} is a finite extension, there is a primitive element β_i in K_i over K_{i-1} such that $K_i = K_{i-1}(\beta_i)$.

If all primitive elements β_i are expressed by radicals, then all roots of f are also expressed by radicals. The solvability of G_f, therefore, means that all primitive elements

in the subfield tower can be expressed by radicals. The following describes the details for each extension K_i/K_{i-1}. Here, we denote a primitive p_i-th root of unity by ζ_{p_i}.

Let $L_{i-1} = K_{i-1}(\zeta_{p_i})$ and $L_i = K_i(\zeta_{p_i})$. Then, there exists a primitive element β_i of L_i such that $\beta_i^{p_i}$ belongs to L_{i-1}. That is, β_i is the p_i-th root of an element $\beta_i^{p_i}$.
Since every p-th root of unity for a prime p can be represented by radicals over $\mathbb{Q}$, once β_i is represented by radicals, β_{i+1} is also represented by radicals. Consequently, the primitive element of K_r is represented by radicals and all roots of f over K_r are represented by radicals. Thus, the problem of radical representation of polynomial roots is reduced to the problem of radical representation of cyclic extension fields.

2.3 Radical representations of cyclic extensions

To express a cyclic extension by radicals, we give a standard method, found in textbooks, based on *Lagrange resolvent*. Consider a cyclic extension $K(\beta)/K$ appearing the subfield tower. Let $n = [K(\beta) : K]$ and σ a generator of its Galois group. For radical representation of β over K, what we need is an element γ in $K(\beta)$ such that γ is also a primitive element and γ^n belongs to K. The Lagrange resolvent gives an "efficiently computable" primitive element. For simplicity, we assume that the ground field K contains a primitive n-th root of unity. Then the following well-known proposition holds:

Proposition 1
There is an element a in K such that $x^n - a$ is K-irreducible and $K(\gamma) = K(\beta)$ for any its root γ. Moreover, for a non-zero element γ in $K(\beta)$, $K(\gamma) = K(\beta)$ and $\gamma^n \in K$ if and only if $\sigma(\gamma) = \gamma\zeta$ for some primitive n-th root ζ of unity.

For the given primitive element β, we form the Lagrange resolvent

$$u(\beta, \zeta) = \beta_0 + \zeta\beta_1 + \cdots + \zeta^{n-1}\beta_{n-1}, \tag{1}$$

where ζ is an n-th root of unity (not necessarily primitive) and $\beta_\nu = \sigma^\nu(\beta)$. Then,

$$\sigma(u(\beta, \zeta)) = \beta_1 + \zeta\beta_2 + \cdots + \zeta^{n-2}\beta_{n-1} + \zeta^{n-1}\beta_0 = \zeta^{-1}u(\beta, \zeta). \tag{2}$$

Then $\sigma(u(\beta, \zeta)^n) = u(\beta, \zeta)^n$ and so $u(\beta, \zeta)^n \in K$. Moreover, if there is a primitive n-th root of unity ζ such that $u(\beta, \zeta) \neq 0$, then $u(\beta, \zeta)$ is a primitive element of $K(\beta)/K$. Because, ζ^{-1} is also a primitive n-th root of unity. And there *exists* such a primitive n-th root ζ when n is prime. (See Remark 3.) Thus, in this case, there is a polynomial $P(x)$ over K such that $P(u(\beta, \zeta)) = \beta$. From this, we obtain a radical representation of β.

Another popular representation utilizes Lagrange resolvents $u(\beta, \zeta)$ for all n-th roots ζ. By multiplying ζ^{-r} to Equation (1) and computing summation for all n-th root of unity ζ, we have

$$\sum_{\zeta} u(\beta, \zeta) = n\beta. \tag{3}$$

Thus, if we know every $u(\beta,\zeta)$, we can express β as the sum of $u(\beta,\zeta)$'s, all of which are n-th roots of elements in K.

Remark 3

(1) *Since the automorphisms, identity,* $\sigma,\ldots,\sigma^{n-1}$, *are* linearly independent, *for each n-th root* ζ *of unity, there is an element* γ *in* $K(\beta)$ *such that* $u(\gamma,\zeta)\neq 0$.

(2) *For the fixed primitive element* β, *Equation* (3) *implies that there is an n-th root of unity such that* $u(\beta,\zeta)\neq 0$. *Thus, if n is prime, there is a primitive n-th root* ζ *of unity such that* $u(\beta,\zeta)\neq 0$. *Because every* non-trivial *n-th roots of unity is primitive.*

2.4 Finding necessary algebraic relations by Gröbner basis techniques

In our method, every algebraic number is represented as an element of a certain residue class ring obtained by factoring a polynomial ring $\mathbb{Q}[x_1,\ldots,x_r]$ by its ideal $\mathcal{I}$, where certain algebraic numbers, say $\gamma_1,\ldots,\gamma_r$, are assigned to variables $x_1,\ldots,x_r$ respectively. Then, every arithmetic operation is executed over the residue class ring $\mathbb{Q}[x_1,\ldots,x_r]/\mathcal{I}$. For arithmetics over the residue class ring, the techniques of Gröbner basis and its applications are very useful. We give a brief explanation on a useful technique *elimination by basis-conversion*. (See Buchberger[5], Cox *et al.*[7] and Becker and Weispfenning[4].)

Assume that algebraic numbers $\gamma_1,\ldots,\gamma_r$ are given. We assign variables $x_1,\ldots,x_r$ to these. Letting $\mathcal{I}$ be the maximal ideal consisting of all polynomials having $(\gamma_1,\ldots,\gamma_r)$ as their zero, we have $\mathbb{Q}(\gamma_1,\ldots,\gamma_r)\equiv\mathbb{Q}[X]/\mathcal{I}$, where $X=\{x_1,\ldots,x_r\}$. Conversely, when a maximal ideal $\mathcal{I}$ is given, algebraic numbers $\gamma_1,\ldots,\gamma_r$ which satisfy all polynomials in $\mathcal{I}$ are determined up to their conjugates by the Galois group of the Galois closure $\bar{\mathbb{Q}}$.

Since $\mathcal{I}$ is maximal, with respect to the lexicographic order $x_1\prec\ldots\prec x_r$, its reduced Gröbner basis becomes $\{f_1(x_1),f_2(x_1,x_2),\ldots,f_r(x_1,\ldots,x_r)\}$, where each f_i is a polynomial in $x_1,\ldots,x_i$ over $\mathbb{Q}$ and it is monic with respect to x_i. Then $f_i(\gamma_1,\ldots,\gamma_{i-1},x_i)$ is the minimal polynomial of γ_i over $\mathbb{Q}(\gamma_1,\ldots,\gamma_{i-1})$. Thus, by changing the variable order, we can compute the *minimal* algebraic relation among specified algebraic numbers. This technique is related to *elimination ideal computation* and *it can be executed efficiently by basis-conversion technique.* In the basis-conversion technique, once one has a Gröbner basis of the ideal $\mathcal{I}$ with respect to some order, one can compute another Gröbner basis of $\mathcal{I}$ with respect to one's desired order merely by solving a system of linear equations (see [11]).

Remark 4

For finding "certain" minimal algebraic relations among specified algebraic numbers, it might seem inadequate to introduce the notion "elimination ideal computation" and use the full computation of Gröbner bases. When we compute algbraic relations at each step in the computation of radical representation, in many cases, we can guess their shapes,

i.e. we know all possible terms which may appear in the relation. And all such terms are expressed by vectors over $\mathbb{Q}$*, since the linear basis of the residue class ring* $\mathbb{Q}[X]/\mathcal{I}$ *is computed by the Gröbner basis of* $\mathcal{I}$*. (In our setting, we are given a Gröbner basis of* $\mathcal{I}$ *with respect to a certain order* $<$*.) Thus, we can compute necessary algebraic relations by solving systems of linear equations derived from vector representation of possible terms. This is the "principle" of basis-conversion technique, and to simply the description of our algorithm, we use the notion of* elimination ideal computation *and* basis-conversion*. Becasue, we can avoid unnecessary computation by stopping the basis-conversion when we obtain every necessary elements.*

For example, if we already know that an algebraic number γ_{i_0} can be expressed as a polynomial P in $\gamma_{i_1}, \ldots, \gamma_{i_s}$ over $\mathbb{Q}$, that is, $x_{i_0} - P(x_{i_1}, \ldots, x_{i_s})$ belongs to $\mathcal{I}$. With respect to the lexicographical order $x_{i_1} \prec \ldots \prec x_{i_s} \prec x_{i_0} \prec \ldots$, the reduced Gröbner basis $\mathcal{G}$ contains $g_1(x_{i_1}), \ldots, g_s(x_{i_1}, \ldots, x_{i_s}), g_{s+1}(x_{i_1}, \ldots, x_{i_s}, x_{i_0})$. Since $x_{i_0} - P(x_{i_1}, \ldots, x_{i_s})$ is reduced to 0 by M-reduction with respect to $\mathcal{G}$, $x_{i_0} - P(x_{i_1}, \ldots, x_{i_s})$ is reduced by g_{s+1}. This implies that g_{s+1} is linear with respect to x_{i_0}, i.e. $g_{s+1} = x_{i_0} - Q(x_{i_1}, \ldots, x_{i_s})$ for some polynomial Q. This is a polynomial expression of γ_{i_0} in $\gamma_{i_1}, \ldots, \gamma_{i_s}$.

To extract algebraic relations between fixed elements $\gamma_{i_0}, \gamma_{i_1}, \ldots, \gamma_{i_s}$, we may use another order. In many cases, employing the following block order improves the efficiency.

Let $\prec_1$, $\prec_2$ be admissible orders on the set T_1 of terms generated by $U = \{x_{i_0}, \ldots, x_{i_s}\}$ and that T_2 of terms generated by $X \setminus U$, respectively. For two terms $t = t_1 t_2$, $t' = t'_1 t'_2$ such that t_i, t'_i belongs to T_i, $t \prec t'$ is defined by $(t_2 \prec_2 t'_2)$ or $(t_2 = t'_2$ and $t_1 \prec_1 t'_1)$. We write $U \prec\prec X \setminus U$. The following gives a theoretical base for the argument above.

Proposition 2
Let $\prec$ *be the above block order and* $\mathcal{G}$ *a Gröbner basis of* $\mathcal{I}$ *with respect to* $\prec$*. Then,* $\mathcal{G} \cap \mathbb{Q}[U]$ *is a Gröbner basis of* $\mathcal{I} \cap \mathbb{Q}[U]$*.*

For an ideal $\mathcal{I}$ of $\mathbb{Q}[X]$, the ideal $\mathcal{I} \cap \mathbb{Q}[U]$ is called the *elimination ideal* of $\mathcal{I}$. Thus, Proposition 2 gives the concrete procedure for computation of elimination ideal. For general notions and algorithms concerning with Gröbener basis, see text books [7] and [4].

3 Outline of the Whole Procedure

Here, we give an outline of the whole procedure. We consider a monic irreducible polynomial $f(x)$ of degree n in the following situation:

Context: We have already computed the splitting field K_f of f and its Galois group G_f. Then K_f is represented by $\mathbb{Q}[y_1, \ldots, y_n]/\mathcal{J}$, where $\mathcal{J}$ is a maximal ideal called *the defining ideal of* K_f. Each root α_i of $f(x)$, $i = 1, \ldots, n$, is represented by a variable y_i, and G_f is represented as a permutation group on $y_1, \ldots, y_n$. Actually G_f is obtained by a set of

generators called a *strong generating set*. The defining ideal $\mathcal{J}$ is generated by the *defining polynomials* $f_1(y_1) = f(y_1)$, $f_2(y_1, y_2), \ldots, f_n(y_1, \ldots, y_n)$ such that each f_i is monic and irreducible with respect to y_i over the extension field $\mathbb{Q}[y_1, \ldots, y_{i-1}]/Id(f_1, \ldots, f_{i-1})$, where $Id(A)$ denotes the ideal generated by A. The set $\{f_1, \ldots, f_n\}$ forms a (reduced) Gröbner basis of $\mathcal{J}$ with respect to the lexicographic order $y_1 \prec y_2 \prec \ldots \prec y_n$. A primitive element β of K_f over $\mathbb{Q}$ is also computed as a linear sum of roots, i.e. $\beta = a_1\alpha_1 + \cdots + a_n\alpha_n$ for $a_i \in \mathbb{Q}$. By [1] there is an integer ℓ, called *the length*, such that $K_f = \mathbb{Q}(\alpha_1, \ldots, \alpha_\ell)$. Then, for a primitive element β, we can set $a_{\ell+1} = \cdots = a_n = 0$.

An outline of the flow of the whole computation is as follows:

General Procedure of Radical Representation

Input: $f(x)$, G, K_f, β. (*i.e.* the output of the direct method in [1])
Output: radical representation of the roots of f.
Restriction: f is solvable. (irreducible, monic)

1. Find the subgroup tower (composition series) of G.
2. Find the corresponding subfield tower.
3. Compute the radical representation of each cyclic extension.

3.1 Subgroups and solvability

We comment on the computation of subgroups of G_f briefly. Here, we employ the existing method using strong generators, see Butler [6], Furst *et al* [12]. Now, a strong generating set $S = \{s_1, \ldots, s_t\}$ of G_f is given and each element s_i in S is expressed as a permutation on the roots. The commutator subgroup $[G_f, G_f]$ of G_f is the *normal closure* of a group generated by commutators $[s_i, s_j] = s_i^{-1}s_j^{-1}s_is_j$ for $1 \leq i, j \leq t$:

$$[G_f, G_f] = \langle [s_i, s_j] \mid 1 \leq i, j \leq t \rangle^G.$$

The strong generating set S_1 of $[G_f, G_f]$ is computed by the normal closure computation (see also [12]). Thus, we obtain the following subgroup tower by repeating computation of commutator subgroups: ($G \rhd G'$ implies that G' is normal in G.)

$$G_f = G_0 \rhd G_1 = [G_0, G_0] \rhd G_2 = [G_1, G_1] \rhd \cdots \rhd G_r = [G_{r-1}, G_{r-1}]$$

Once the sequence of commutator subgroups is obtained, we can also determine the solvability of G_f immediately by checking whether $G_r = 1$ or not. The sequence of commutator subgroups is a *normal chain* and each G_{i-1}/G_i is abelian. From the normal chain, a composition series is computed easily. As the computation of subgroups is not a dominant step in the whole procedure, we do not consider a new method. We should examine which method is suited for practical computation.

3.2 Subfields

We present a method for constructing a subfield tower. From now on, we only consider the solvable case. Thus we already know a composition series $G_0 = G_f, G_1, \ldots, G_r = 1$.

Let B_i be the orbit of G_i containing the fixed primitive element β of K_f over $\mathbb{Q}$, i.e. $B_i = \{\sigma(\beta) \mid \sigma \in G_i\}$. Since β is a primitive element, B_0 is the set of all conjugates of β in K_f and G acts on B_0 regularly. From this, we have $|B_i| = |G_i| = [K_f : K_i]$.

Definition 3
For a finite set B in K_f, we define the polynomial f_B corresponding to B *by*

$$f_B = \prod_{b \in B} (x - b).$$

We call the field obtained by adjoining all coefficients of f_B to $\mathbb{Q}$ the field corresponding to B *and denote it by K_B. Then K_B is a subfield of K_f.*

Lemma 4
For each orbit B_i, the subfield K_{B_i} coincides with K_i.

Proof First we show $K_i \supseteq K_{B_i}$. Since each coefficient of f_{B_i} is a symmetric function in elements in B_i, it is fixed by G_i. By the Galois correspondence, this implies $K_i \supseteq K_{B_i}$. Next we show $K_{B_i} = K_i$. As β belongs to B_i, $f_{B_i}(\beta) = 0$. Since $\deg(f_{B_i}) = |B_i| = |G_i|$, $[K_f : K_{B_i}] \leq |G_i| = [K_f : K_i]$. Since $K_{B_i} \subset K_i$, we obtain $[K_f : K_{B_i}] = [K_f : K_i]$ and hence $K_i = K_{B_i}$. ∎

Lemma 4 corresponds to Lemma 1 in [16]. Since $[K_{i-1} : K_i]$ is prime, K_i is maximal in K_{i-1}. By this fact and [27], we have:

Corollary 5
(1) *Among all coefficients of f_{B_i}, there is a primitive element of K_i over K_{i-1}. Moreover, any coefficient of f_{B_i} not belonging to K_{i-1} is a primitive element of K_i over K_{i-1}.*
(2) *There is a primitive element of K_i over $\mathbb{Q}$ among $\mathbb{Q}$-linear sums of coefficients of f_{B_i}, in particular, among the $f_{B_i}(a)$ for distinct $(|B_i| - 1)|K_i : \mathbb{Q}|$ elements a of $\mathbb{Q}$.*
(3) *Assume that β_{i-1} is a primitive element of K_{i-1} over $\mathbb{Q}$ and β' is that of K_i over K_{i-1}. Then, there is a primitive element of K_i over $\mathbb{Q}$ among $\beta_{i-1} + a\beta'$ for $|K_i : \mathbb{Q}|$ distinct elements a of $\mathbb{Q}$.*

Definition 6
We call a primitive element of K_i over K_{i-1} a relative *primitive element. And, we call a primitive element of K_i over K_{i-1}* an absolutely *primitive element, if it is also a primitive element of K_i over $\mathbb{Q}$.*

Now $f_{B_r} = x - \beta$ and β is the last primitive element of K_r over K_{r-1}. Meanwhile, for the last relative primitive element of K_r over K_{r-1}, we can choose some root α_i.

Lemma 7

Among all roots $\alpha_1, \ldots, \alpha_n$, there is a relative primitive element of K_r. (In particular, among roots $\alpha_1, \ldots, \alpha_\ell$, there is a relative primitive element of K_r.) If G_{r-1} is a normal subgroup of G_0, then every root α_i is a relative primitive element of K_r.

Proof Since K_r is the splitting field of f, any proper subfields do not contain all roots of f. From this, there is a root, say α_i, of f which does not belong to K_{r-1}, and since there is no proper subfield between K_{r-1} and K_r, $K_{r-1}(\alpha_i) = K_r$. Moreover if $G_0 \rhd G_{r-1}$, then K_{r-1} is a Galois extension over $\mathbb{Q}$. From this, if K_{r-1} contains some root α_i, then K_{r-1} contains every conjugate of α_i and this implies a contradiction. ∎

Definition 8

There is a case where one root of f belongs o a proper subfield K_i, $i < r$. We call this contractible *case and call such a root a* contractible root *and such a subfield a* contracting subfield. *Of course, the contractibility of each root depends on the choice of a composition series of the Galois group.*

Since β is a linear sum of roots and G_i is represented as a concrete permutation group on the roots, elements of B_i are computed as polynomials in $y_1, \ldots, y_n$ modulo $\mathcal{J}$ and coefficients of f_{B_i} are also computed as polynomials in $y_1, \ldots, y_n$ modulo $\mathcal{J}$.

Now, we present an abstract procedure for computing subfields by *relative* primitive elements. We assume that $K_1, \ldots, K_i$ are already computed, that is, primitive elements $\beta_1, \ldots, \beta_i$ are already computed. Then, we compute a primitive element β_{i+1} of K_{i+1}. By Corollary 5 (1) or Lemma 7, we choose β_{i+1} from all coefficients of $f_{B_{i+1}}$ or roots of f. Finding such an element is reduced to determining a certain algebraic relation between $\beta_1, \ldots, \beta_i$ and each candidate. Determination of algebraic relations is easily computed by Gröbner basis algorithms in §2.4. We will give a further discussion later.

Procedure Next Relative Primitive Element

Input: G_{i+1}, β, $\beta_1, \ldots, \beta_i$.
Output: β_{i+1}, $m_{i+1}(\beta_1, \ldots, \beta_i, x)$.

1. Compute the polynomial $f_{B_{i+1}}$ corresponding to B_{i+1}.
2. Find a coefficient c not belonging to K_i among all coefficients of $f_{B_{i+1}}$.
   (When $i = r-1$, we can replace the steps 1, 2 with the following step 1′.
   1′. Find a root c not belonging to K_{r-1} among all roots of $f(x)$.)
3. Find the minimal polynomial of c over K_i as an algebraic relation between $\beta_1, \ldots, \beta_i$ and c with smallest exponent d $(=p_{i+1})$ such that

$$c^d + a_{d-1}(\beta_1, \ldots, \beta_i)c^{d-1} + \cdots + a_0(\beta_1, \ldots, \beta_i) = 0,$$

   where $a_0, \ldots, a_{d-1}$ are polynomials in $\beta_1, \ldots, \beta_i$ over $\mathbb{Q}$.
4. Return c as β_{i+1} and $x^d + a_{d-1}x^{d-1} + \cdots + a_0$ as $m_{i+1}(\beta_1, \ldots, \beta_i, x)$.

Remark 5

We can also compute a sequence of absolutely primitive elements of K_i's by using facts in Corollary 5 and algorithms in [27]. In this paper, we use relative primitive elements, because they are superior to absolutely primitive elements in terms of practical computation. See §7.2.

3.3 Radical representations of cyclic extension

By §2.2, we have the following abstract procedure for radical representations of cyclic extensions with prime extension degree:

Procedure Radical Representation of Cyclic Extensions

Input: the minimal polynomial $m(x)$ of a primitive element γ of a cyclic extension field L with prime extension degree p over K, and the minimal polynomial $g(y)$ of a primitive p-th root of unity ζ over K.
Output: a radical representation of γ over K. (γ, ζ are assigned to x, y, respectively.)
Assumption: a primitive p-th root of unity ζ is represented by radicals and it gives a non-zero Lagrange resolvent.

1. Construct a non-zero Lagrange resolvent $u(x, y)$ and let $h(x, y, z) = z - u(x, y)$.
2. Represent x as a polynomial $P(y, z)$ from $m(x)$, $g(y)$, $h(x, y, z)$.
3. Compute z^p and reduce it by $m(x)$ and $g(y)$ to a polynomial R in $K[y]$.
4. Replace z in $P(y, z)$ with $\sqrt[n]{R}$ and replace y in $P(y, z)$ with its radical representation.
5. Return P.

Since $[K(\zeta) : K] < [K(\gamma) : K]$, $[K(\gamma) : K]$ and $[K(\zeta) : K]$ are mutually prime. Therefore, $K(\gamma)$ and $K(\zeta)$ are linearly disjoint over K and so $K(\gamma, \zeta) \cong K[x, y]/Id(f(x), g(y))$.

To execute the above algorithm efficiently, the most expensive part is to express x as a polynomial in y and z. We obtain $P(y, z)$ by computing algebraic relation among x, y and z from $m(x), g(y)$ and $h(x, y, z)$ as described in §6.

3.4 The radical representation of a primitive root of unity

In computing radical representations, a primitive p-th root of unity must be expressed by radicals beforehand. A radical representation of a primitive n-th root of unity ζ_n for a positive integer n is also obtained by the method using Lagrange resolvent, see [2]. The extension field obtained by adjoining ζ_n is an abelian extension with degree smaller than n and so if we know the radical representation of a primitive m-th root ζ_m of unity for any positive integer m smaller than n, we can represent ζ_n by radicals. Thus, radical representation of ζ_n is reduced to the case of smaller degree. Moreover, suppose that $n = p_1^{e_1} \cdots p_r^{e_r}$. Then, ζ_n is expressed by $\zeta_n = \zeta_{p_1^{e_1}} \zeta_{p_2^{e_2}} \cdots \zeta_{p_r^{e_r}}$ and so if the radical

representation of each primitive $p_i^{e_i}$-th root of unity $\zeta_{p_i^{e_i}}$ is known, we are able to represent ζ_n in terms of radicals. By repeating these reductions, the radical representation of a primitive n-th root of unity is reduced to the case of smaller prime.

As for *strong radical representation*, we note the following. By the theory of cyclotomic fields, if n, m are mutually prime, then $\mathbb{Q}(\zeta_n)$ and $\mathbb{Q}(\zeta_m)$ are linearly disjoint over $\mathbb{Q}$ and $\zeta_n\zeta_m$ is a primitive nm-th root of unity. From this fact, we have the following.

Lemma 9
Let n, m be mutually prime positive integers. The product of strong radical representations of ζ_n and ζ_m gives a strong radical representation of ζ_{nm}.

4 Subfield Computation

Now, we present a concrete method for constructing the subfield tower discussed roughly in §3.2. Assume the context of §3.

Remark 6
We can omit the redundant *variables $y_{\ell+1}, \ldots, y_n$, where ℓ is the length of the representation of K_f. Because each root $\alpha_{\ell+i}$, $1 \leq i$, is expressed by a polynomial in $y_1, \ldots, y_\ell$ over $\mathbb{Q}$ (see [1]).*

From now on, we denote $y_1, \ldots, y_n$ or $y_1, \ldots, y_\ell$ by $y_1, \ldots, y_t$ and set $Y = \{y_1, \ldots, y_t\}$. If we use the above improvement, then $t = \ell$ and $\mathcal{J} = Id(f_1, \ldots, f_\ell)$, and otherwise, $t = n$ and $\mathcal{J} = Id(f_1, \ldots, f_n)$.

4.1 Finding subfields

Our target is the subfield tower $K_0, \ldots, K_r$ corresponding to the composition series $G_0, \ldots, G_r$. First, for each i, $1 \leq i \leq r-1$, we compute the orbit B_i of G_i containing the fixed primitive element β, and its corresponding polynomial f_{B_i}. By Corollary 5 (1), we have only to find a coefficient of f_{B_i} not belonging to K_{i-1} as a primitive element of K_i. Also, by Lemma 7, we can find a primitive element of K_r over K_{r-1} among roots $\alpha_1, \ldots, \alpha_t$. Procedure Next Relative Primitive Element gives an abstract procedure for it. Thus, we concentrate on how to find a desired element, a coefficient of f_{B_i} or a root α_j, under the following setting.

Setting: Assume that relative primitive elements $\beta_1, \cdots, \beta_{i-1}$ are already computed, where each β_j, $j = 1, \ldots, i-1$, is represented by a polynomial in Y over $\mathbb{Q}$. The minimal polynomial $m_j(\beta_1, \ldots, \beta_{i-1}, x)$ of β_j over K_{j-1} is also computed for $j = 1, \ldots, i-1$.

Choose a candidate γ, a coefficient of f_{B_i} or a root α_j, if $i = r$, for some j. Then, we can determine whether γ is a primitive element of K_i over K_{i-1} as follows. By *elimination ideal*

computation, we can compute *algebraic dependency*. Let $u_1, \ldots, u_{i-1}, v$ be new variables assigned to $\beta_1, \ldots, \beta_{i-1}, \gamma$, respectively. That is,

$$u_j - \beta_j(Y) = 0 \text{ for } j = 1, \cdots, i-1 \text{ and } v - \gamma(Y) = 0.$$

We set $U_{i-1} = \{u_1, \ldots, u_{i-1}\}$. Let $\bar{\mathcal{J}} = Id(u_1 - \beta_1, \ldots, u_{i-1} - \beta_{i-1}, v - \gamma, f_1, \ldots, f_t)$ in $\mathbb{Q}[Y, U_{i-1}, v]$. Then, $\bar{\mathcal{J}}$ is a maximal ideal such that $\mathbb{Q}[Y, U_{i-1}, v]/\bar{\mathcal{J}} \cong K_f$. And, $\{v - \gamma, u_{i-1} - \beta_{i-1}, \ldots, u_1 - \beta_1, f_t, \ldots, f_1\}$ is a Gröbner basis of $\bar{\mathcal{J}}$ with respect to the lexicographical ordering $Y \prec U_{i-1} \prec v$. Here we also denote by Y the order $y_1 \prec \ldots \prec y_t$ and so on.

Theorem 10
(1) *The ideal $\bar{\mathcal{J}} \cap \mathbb{Q}[U_{i-1}]$ coincides with the maximal ideal generated by minimal polynomials $m_1(u_1), \ldots, m_{i-1}(U_{i-1})$ over $\mathbb{Q}$.*
(2) *The ideal $\bar{\mathcal{J}} \cap \mathbb{Q}[U_{i-1}, v]$ coincides with the ideal generated by $m_1, \ldots, m_{i-1}$ and the minimal polynomial of γ over K_{i-1}.*

Proof We note that since $\bar{\mathcal{J}}$ is a maximal ideal, every elimination ideal becomes a maximal ideal of each corresponding polynomial ring.
(1) Since $\bar{\mathcal{J}} \cap \mathbb{Q}[U_{i-1}]$ is a maximal ideal, it contains every algebraic relation among $\beta_1, \ldots, \beta_{i-1}$ and its residue class ring is isomorphic to K_{i-1}, therefore, $m_1, \ldots, m_{i-1}$ belong to $\bar{\mathcal{J}} \cap \mathbb{Q}[U_{i-1}]$. But, since each m_j is irreducible over K_{j-1}, $\{m_1, \ldots, m_{i-1}\}$ generates a maximal ideal $\dot{\mathcal{J}}$ such that $K_{i-1} \equiv \mathbb{Q}[U_{i-1}]/\dot{\mathcal{J}}$. From this, we have $\bar{\mathcal{J}} \cap \mathbb{Q}[U_{i-1}] = \dot{\mathcal{J}}$.
(2) By using the similar argument as above, we can prove (2). ∎

Corollary 11
Let GB be the reduced Gröbner basis of $\bar{\mathcal{J}}$ with respect to a block order $\{U_{i-1} \prec v\} \prec\prec Y$. Then,
(1) *GB contains $m_j(U_j)$ for each j, $1 \leq j \leq i-1$.*
(2) *GB contains a polynomial $h(U_{i-1}, v)$ which coincides with the minimal polynomial of γ over K_{i-1}.*

Changing from the lexicographic order to the block order, we can compute the minimal polynomial of γ over K_{i-1} by Gröbner basis algorithms or equivalenty by solving a system of linear equations (see §2.4), and moreover, we can determine whether γ belongs to K_{i-1} by testing whether the minimal polynomial of γ is linear with respect to v.

Corollary 12
Use the same notation as in Corollary 11. Then, γ is a primitive element of K_i over K_{i-1} if and only if there is an element in GB which is a polynomial in U_{i-1}, v over $\mathbb{Q}$ with nonlinear v-term.

4.2 Finding contractible roots

As a by-product of finding relative primitive elements, we can find a contractible root. Corollary 14 shows that we can check whether there is a contractible root immediately.

Lemma 13
If some K_i is a contracting subfield for a root α_j, $1 \leq j \leq t$, then there is a polynomial $y_j - P(U_i)$ in the ideal $\bar{\mathcal{J}}' = Id(u_1 - \beta_1, \ldots, u_i - \beta_i, f_1, \ldots, f_t)$ of $\mathbb{Q}[Y, U_i]$.

Corollary 14
Let GB be a Gröbner basis of $\bar{\mathcal{J}}'$ with respect to a block order $U_i \prec\prec Y$. Then, K_i is a contracting subfield for some α_j if and only if there is a polynomial in U_i, y_j linear with respect to y_j in GB.

Remark 7
For radical representation, it suffices to compute one radical representation of a contractible root α_j in its contracting subfield K_i. This gives further improvement to the total efficiency. Moreover, since we use some cyclotomic field $\mathbb{Q}(\zeta)$ as a ground field (See §6), there is a case where a root of f does not belong to K_i but it belongs to $K_i(\zeta)$. This case is also termed contractible.

5 Radical Representation of Cyclic Extensions

Here we present concrete methods for the radical representation of *general* cyclic extensions with prime degree under the following setting:

Setting: Let L be a cyclic extension of the ground field K with prime extension degree p, and G its Galois group, where K is either $\mathbb{Q}$ or its finite extension. A primitive element β of L/K, and all its conjugates over K are given. This implies that the minimal polynomial $m(x)$ of β over K is given and all conjugates of β are expressed as polynomials in β over K by the identification $L = K(\beta) \cong K[x]/Id(f(x))$, where x is assigned to β.

The algorithm consists of two parts: construction of a non-zero Lagrange resolvent and construction of an expression of the fixed primitive element as a polynomial in the Lagrange resolvent and the fixed primitive p-th root of unity.

5.1 Finding a non-zero resolvent

We seek a primitive p-th root ζ of unity which gives non-zero Lagrange resolvent $u(\beta, \zeta)$. As mentioned in Remark 3 (2), there exists such a primitive p-th root ζ. Let $g_0(y)$ be the minimal polynomial of a primitive p-th root of unity over $\mathbb{Q}$. Then, $g_0(y)$ is also the minimal polynomial of *any* primitive p-th root of unity. In order to represent the *composite field* L'

of L and the cyclotomic field K' generated by all p-th roots of unity, we factorize g_0 into its irreducible factors $g_1, \ldots, g_s$ over $L = K(\beta)$.

Lemma 15

(1) *For each primitive p-th root ζ of unity , $K' = K(\zeta)$ and $L' = L(\zeta)$.*

(2) *Every irreducible factor g_i is also an irreducible polynomial over K and has the same degree.*

Proof Since (1) is clear, we have only to show (2). Since $\deg(g_i) = [L' : L]$, every irreducible factor has the same degree. We show that g_i's are polynomials over K. If K contains a primitive p-th root, then g_i's are polynomials over K. Thus, we assume that K does not contain any primitive p-th root. Since $[K' : K] < p$, K' and L are linearly disjoint and so $[L' : L] = [K' : K]$. Thus each g_i is a polynomial over K. ∎

By Lemma 15, each $R_i = K[x, y]/Id(f(x), g_i(y))$, $i \geq 1$, is isomorphic to L', where we assign a variable y to each primitive p-th root. By Remark 3 (2), we have:

Lemma 16

(1) *There is some R_i such that $u(x, y) \neq 0$ in R_i. In particular, if g_0 is irreducible over K, $u(\beta, \zeta) \neq 0$ for any primitive p-th root ζ of unity.*

(2) *If $u(\beta, \zeta) = 0$ for a fixed primitive p-th root ζ of unity, then there is a positive integer s less than p such that $u(\beta, \zeta^s) \neq 0$, where ζ^s is also a primitive p-th root of unity.*

We fix a primitive p-th root ζ_p. By Lemma 16 one of the following holds.

(a) $g(y) = g_i(y)$ for some i, $L' = R_i \cong K[x, y]/Id(f(x), g(y))$ and a variable z is assigned to the non-zero Lagrange resolvent $u(x, y)$ in L',

(b) $g(y) = g_i(y)$ for some i, $L' = R_i$ and a variable z is assigned to the non-zero Lagrange resolvent $u(x, y^s)$ in L' for some s.

Then, $u(\beta, \zeta_p)^p$ or $u(\beta, \zeta_p^s)^p$ is easily computed in R_i, and it is expressed as a polynomial $H(\zeta_p)(= H(y))$ in $\zeta_p(= y)$ over K. The *minimal* polynomial $h(x, y, z)$ of z over L' is $z - u(x, y)$ or $z - u(x, y^s)$, and the minimal polynomial of z over $K(\zeta_p)$ is $z^p - H(y)$. And

$$L' = R_i \cong K[x, y, z]/Id(f(x), g(y), h(x, y, z)) \cong K[y, z]/Id(g(y), z^p - H(y)).$$

5.2 Expressing a primitive element by radical

Now we come to consider expressing the fixed primitive element as a polynomial in the Lagrange resolvent and the fixed primitive p-th root of unity. The target problem is *"how to obtain the polynomial $x - P(y, z)$ from $m(x), g(y)$ and $h(x, y, z)$"*. Here, we present a method based on *elimination ideal computation*. We assume that a primitive p-th root of unity is already found, and use the same notations as in §5.1. Then,

$$K(\beta, \zeta_p) \cong K[x, y, z]/Id(m(x), g(y), h(x, y, z)).$$

The polynomials $m(x), g(y)$ and $h(x, y, z)$ form the reduced Gröbner basis of the ideal $Id(m(x), g(y), h(x, y, z))$ with respect to the lexicographic ordering $x \prec y \prec z$ as they are. Since x can be expressed as a polynomial in y and z, we have the following by §2.4.

Theorem 17

There exists an element $x - P(y, z)$ in the Gröbner basis of the ideal $Id(f(x), g(y), h(x, y, z))$ with respect to a block order $\{y, z\} \prec\prec x$.

Remark 8

On the Gröbner basis computation, we noted the following:
(1) *In this method, we may stop the procedure of Gröbner basis computation when a polynomial $r(x, y, z)$ which is linear with respect to x appears in the procedure. Then, the efficiency of this method shall be improved, though the shape of the result differs from that obtained through complete computation.*
(2) *With respect to the order $y \prec z \prec x$, $g(y)$ and $z^p - H(y)$ appear in the Gröbner basis of $Id(f(x), g(y), h(x, y, z))$. Thus, for the computation of Gröbner basis we use the set $\{m(y), g(y), z^p - H(y), h(x, y, z)\}$ as the input. (In this case, the whole computation becomes very similar to the procedure described in Remark 9.)*

Remark 9

We can obtain a radical representation by GCD of $m(x)$ and $h(x, y, z)$ over the field $K(\zeta_p, u(\beta, \zeta_p)) \cong K[y, z]/Id(g(y), z^p - H(y))$. Since $m(x), g(y), h(x, y, z)$ generate a maximal ideal, $m(x)$ and $h(x, y, z)$ have a common factor. As $m(x)$ is square-free, the common factor is a linear factor. We can compute GCD by Euclid's algorithm using pseudo division. However, in general, this method gives the result like $B(y, z)x - C(y, z)$, where B, C are polynomials in y, z over K. (The pseudo-GCD computation corresponds to that of S-polynomial *in the Gröbner basis computation.) And* intermediate coefficient swell *occurring in the pseudo-GCD becomes a serious issue.*

5.3 On strong representation

In our setting, a primitive element β' of K is represented by radicals over $\mathbb{Q}$, and ζ_p is also represented by radicals over $\mathbb{Q}$. By the method in §5.2, we compute a strong radical representation of β over K, that is, an expression in terms of radicals of polynomials in β' and ζ_p, and finally by substituting β' and ζ_p with their strong radical representations over $\mathbb{Q}$, we obtain a radical representation of β over $\mathbb{Q}$. From $\langle 4 \rangle$ in §2.1, we have:

Lemma 18

Assume that $\mathbb{Q}(\beta')(= K)$ and $\mathbb{Q}(\zeta_p)$ are linearly disjoint. (This is equivalent to the condition that $g_0 = g$.) Then, the radical representation of β becomes a strong representation for any pair of strong radical representations of ζ_p over $\mathbb{Q}$ and β' over $\mathbb{Q}$.

When $\mathbb{Q}(\beta')$ and $\mathbb{Q}(\zeta_p)$ are not linearly disjoint, there may be some terms in radical representations of β' and ζ_p such that their evaluations depend on each other. In this case, additional procedures are required. We will discuss this in the next section.

6 Radical Representation of Polynomial Roots

We assume the setting in §4 and consider how we compute a radical representation of each primitive element β_i. Since our final aim is to obtain one radical representation of a root of $f(x)$, it suffices to compute radical representation of each primitive element β_i for $i = 1, \ldots, r'$, if there is a contractible root in $K_{r'}$. So, first we will give a concrete procedure for the *general* case, where there is no contractible root, and then we will comment the *contractible* case. Here, we assume that every primitive element β_i is already computed. At each step i, since primitive element β_{i+1} corresponds to cyclic extension, we apply methods in §5. To compute a strong radical representation, we must replace the ground field $\mathbb{Q}$ with a certain cyclotomic field.

6.1 Cyclotomic field corresponding to a polynomial

First, we define the following cyclotomic field determined by the input polynomial f.

Definition 19
Let $p_i = [K_i : K_{i-1}]$ for $i = 1, \ldots, r$ and let $q_1, \ldots, q_s$ be distinct odd primes among $p_1, \ldots, p_r$. Then $K_f(\zeta_{q_1}, \ldots, \zeta_{q_s})$ is a field over which every arithmetic operation for radical representation can be done, and we call it the extended splitting field *of f and denote it by L_f. Moreover, we call $\mathbb{Q}(\zeta_{q_1}, \ldots, \zeta_{q_s})$ the* cyclotomic field corresponding to f *and denote it by C_f.*

A strong radical representation of each ζ_{q_i} can be computed efficiently by taking advantage of the special properties of primitive roots of unity. (See [2].) Thus, we assume that strong radical representations of ζ_{q_i}'s are already computed. By Lemma 9, there is no restriction on the choice of radical representations of ζ_{q_i}'s.

We assign new variables $z_1, \ldots, z_s$ to $\zeta_{q_1}, \ldots, \zeta_{q_s}$. Set $Z = \{z_1, \ldots, z_s\}$. Since $\mathbb{Q}(\zeta_{q_i})$ is linearly disjoint to $\mathbb{Q}(\zeta_{q_1}, \ldots, \zeta_{q_{i-1}}, \zeta_{q_{i+1}}, \ldots, \zeta_{q_s})$ for $i = 1, \ldots, s$, C_f is expressed as

$$C_f \cong \mathbb{Q}[Z]/Id(g_{0,1}(z_1), \ldots, g_{0,s}(z_s)),$$

where each $g_{0,i}$ is the minimal polynomial of ζ_{q_i} over $\mathbb{Q}$ and $g_{0,i} = (z_i^{q_i} - 1)/(z_i - 1)$. Moreover, L_f is expressed as

$$L_f \cong \mathbb{Q}[Y, Z]/Id(f_1, \ldots, f_t, g_1, \ldots, g_s),$$

where $Y = \{y_1, \ldots, y_t\}$ and each $g_i \in \mathbb{Q}[Y, z_1, \ldots, z_i]$ is the minimal polynomial of ζ_{q_i} over $K_f(\zeta_{q_1}, \ldots, \zeta_{q_{i-1}}) = \mathbb{Q}[Y, z_1, \ldots, z_{i-1}]/Id(f_1, \ldots, f_t, g_1, \ldots, g_{i-1})$. Each polynomial g_i can be obtained by factoring $g_{0,i}$ over $K_f(\zeta_{q_1}, \ldots, \zeta_{q_{i-1}})$.

6.2 Radical representation of relative primitive elements

We describe the computation at the i-th step, $1 \leq i \leq r$. Let $L_j = K_j(\zeta_{q_1}, \ldots, \zeta_{q_s})$ for $1 \leq j \leq r$, and $L_0 = C_f$, and assign new variables $u_1, \ldots, u_i$ to $\beta_1, \ldots, \beta_i$, that is, $u_j - \beta_j(Y) = 0$ for $j = 1, \ldots, i$. Set $U_j = \{u_1, \ldots, u_j\}$ for $j \leq r$. Since $[K_i : K_{i-1}]$ is a prime p_i, $[L_i : L_{i-1}] = p_i$ or $L_i = L_{i-1}$ holds. In more detail, we have

Lemma 20

(1) *$L_r = L_f$ is a Galois extension over $\mathbb{Q}$ and so over L_0.*
(2) *The Galois group of L_r/L_i coincides with the stabilizer of L_0 in G_i.*
(3) *The element β_i is a primitive element of L_i over L_{i-1}.*
(4) *If $L_{i-1} \neq L_i$, then the Galois group of L_i/L_{i-1} is isomorphic to the Galois group of K_i/K_{i-1} and so the factor group G_{i-1}/G_i.*

We can determine whether $L_i = L_{i-1}$ by *elimination ideal computation*. With respect to a block order $Z \prec\prec U_i \prec\prec Y$, the reduced Gröbner basis, say GB_i, of the ideal $\mathcal{J}_i = Id(u_1 - \beta_1, \ldots, u_i - \beta_i, g_1, \ldots, g_s, f_1, \ldots, f_t)$ has the following property. (For the proof, see §4.)

Lemma 21

In GB_i there is a polynomial P_j in variables U_j, Z which is monic with respect to u_j for $j = 1, \ldots, i$. Moreover, $GB_i \cap \mathbb{Q}[U_i, Z] = \{g_{0,1}, \ldots, g_{0,s}, P_1, \ldots, P_i\}$ and

$$L_i \cong \mathbb{Q}[U_i, Z]/Id(g_{0,1}, \ldots, g_{0,s}, P_1, \ldots, P_i).$$

Then, $L_i = L_{i-1}$ if and only if P_i is linear with respect to u_i. And if P_i is linear with respect to u_i, then u_i, *i.e.* the algebraic number β_i, is expressed as a polynomial in $\beta_1, \ldots, \beta_{i-1}$ and $\zeta_{q_1}, \ldots, \zeta_{q_s}$. From this expression, a radical representation of β_i is obtained. Therefore we have only to consider the case where $[L_i : L_{i-1}] = p_i$ holds.

By Lemma 20, if $[L_i : L_{i-1}] = p_i$, the set of all conjugates of β_i in K_i/K_{i-1} coincides with that in L_i/L_{i-1}. Thus, the Lagrange resolvent is computed easily and represented by a polynomial $V_i(Z, Y)$. We assign a variable v_i to $V_i(Z, Y)$, i.e. $v_i - V_i(Z, Y) = 0$.

Consider the ideal $\tilde{\mathcal{J}}_i$ generated by its Gröbner basis $GB_i^* = \{u_1 - \beta_1, \ldots, u_i - \beta_i, v_i - V_i, f_1, \ldots, f_t, g_1, \ldots, g_s\}$ in the polynomial ring $\mathbb{Q}[Y, Z, U_i, v_i]$ with respect to the lexicographic order $\{Y \prec Z \prec U_i \prec v_i\}$. The ideal $\tilde{\mathcal{J}}_i$ is a maximal ideal whose residue class ring is equivalent to L_f. By changing the order to a block order $\{Z \prec\prec \{U_{i-1} \prec v_i \prec u_i\} \prec\prec Y\}$, we can obtain the polynomial expression of u_i in $u_1, \ldots, u_{i-1}, v, z_1, \ldots, z_s$. Moreover, the

minimal polynomial of v_i over L_{i-1} is of form $v_i^{p_i} - Q_j(U_{i-1}, Z)$ and Q_j is computed directly from the normal form of $V_j^{p_j}$ with respect to GB_i^*. (See §5.)

Thus, at this step, we have a strong radical representation of the primitive element β_i by the previous ones $\beta_{i-1}, \ldots, \beta_1$ and the fixed roots of unity. At the final step $i = r$, we have a strong radical representation of the final primitive element β_r from which strong radical representations of roots are derived. In more detail, by substituting each β_i with its strong radical representation by $\beta_{i-1}, \ldots, \beta_1$ and the fixed primitive p_i-th root from $i = r$ to 1 repeatedly, we have a strong radical representation of β_r over C_f, i.e. a representation of β_r in terms of radicals over $\mathbb{Q}[Z]/Id(g_{0,1}, \ldots, g_{0,s})$. Since each z_i is defined by an arbitrary root of $g_{0,i}$, we can use any primitive p_i-th root of unity as the value of ζ_{q_i}. Thus, by using the already computed strong radical representation for each ζ_{q_i}, we finally obtained a strong radical representation of β_r.

We give another direct description of the whole procedure which is equivalent to the above. New variables $v_1, \ldots, v_r$ correspond to algebraic numbers $\gamma_1, \ldots, \gamma_s$ such that $L_i = L_{i-1}(\gamma_i)$ and γ^{p_i} belongs to L_{i-1} for each i, $1 \leq i \leq s$. We set $V = \{v_1, \ldots, v_r\}$. Then, we express every β_i as a polynomial in $\zeta_{q_1}, \ldots, \zeta_{q_s}$ and $\gamma_1, \ldots, \gamma_s$ as follows:

Let $\mathcal{I}$ be an ideal in $\mathbb{Q}[Y, Z, U, V]$ generated by its Gröbner basis $\{f_1, \ldots, f_t, g_1, \ldots, g_s, u_1 - \beta_1, \ldots, u_r - \beta_r, v_1 - V_1, \ldots, v_r - V_r\}$ with respect to the lexicographic order $\{Y \prec Z \prec U \prec V\}$. Then its Gröbner basis GB with respect to a block order $\{\{Z \prec V\} \prec\prec \{Y \prec U\}\}$ has the following property.

Lemma 22
For each j, $1 \leq j \leq r$, there is a polynomial R_j in variables V_{j-1}, Z such that $v_j^{p_j} - R_j$ belongs to GB, where $V_j = \{v_1, \ldots, v_j\}$. Moreover, $GB \cap \mathbb{Q}[V, Z] = \{g_{0,1}, \ldots, g_{0,s}, v_1^{p_1} - R_1, \ldots, v_r^{p_r} - R_r\}$ and

$$L_f \cong \mathbb{Q}[V, Z]/Id(g_{0,1}, \ldots, g_{0,s}, v_1^{p_1} - R_1, \ldots, v_r^{p_r} - R_r).$$

Each v_j, i.e. an algebraic number γ_j, is expressed by a p_i-th root of R_j, and from the polynomial R_r we can compute a strong radical representation of a root of f.

Contractible Case As a by-product in the step for constructing the subfield tower, we recognize the case where some root, say α_i, belongs to a proper subfield K_j. In this case, we can replace the extended splitting field L_f with $K_f(\zeta_{q_1}, \ldots, \zeta_{q_{s'}})$, where $q_1, \ldots, q_{s'}$ are all distinct primes among $p_1, \ldots, p_j$. By the same procedure as in the general case, we have radical representations of the primitive elements $\beta_1, \ldots, \beta_j$. Although β_j is not a root of f, a contractible root α_i is expressed as a polynomial in $\beta_1, \ldots, \beta_j$. From this, we have a radical representation of a root of f.

Moreover, there is a case where any root of $f(x)$ does not belong to any proper subfield K_j, but some root α_i, $1 \leq i \leq t$, belongs to some proper subfield $L_j = K_j(\zeta_{q_1}, \ldots, \zeta_{q_{s'}})$.

This case is checked by whether a polynomial in y_i, U_j, Z linear with respect to y_i appears in the Gröbner basis or not. See Remark 7.

6.3 Improvements and comments

When the Galois group K_f is large, it is rather difficult to factorize $g_{0,i}$'s. To execute their factorizations efficiently, we propose the following improvements.

[Improvement A] Before factoring $g_{0,i}$ over $K_f(\zeta_{q_1}, \ldots, \zeta_{q_{i-1}})$, we factorize $g_{0,i}$ over a certain proper subfield $K_j = \mathbb{Q}[y_1, \ldots, y_j]/\ Id(f_1, \ldots, f_i)$ for $j < t$ or $K_f(\zeta_{q_1}, \ldots, \zeta_{q_{j'}})$ for $j' < i-1$. Then it suffices to factorize any factor of the above factorization instead of $g_{0,i}$ itself over K_f. This improvement works very well when there are strong algebraic relations between K_f and $\mathbb{Q}(\zeta_{q_i})$, in particular, when K_f contains $\mathbb{Q}(\zeta_{q_i})$.

[Improvement B] In factorization of $g_{0,i}$ over K_f, we can replace the field K_f with a smaller subfield K'. Let $N = [K_f : \mathbb{Q}]$. Since $[K_f(\zeta_{q_i}) : K_f] = [\mathbb{Q}(\zeta_{q_i}) : \mathbb{Q}(\zeta_{q_i}) \cap K_f]$ and $[\mathbb{Q}(\zeta_{q_i}) \cap K_f : \mathbb{Q}]$ is a common divisor of N and $q_i - 1$, $[\mathbb{Q}(\zeta_{q_i}) \cap K_f : \mathbb{Q}]$ is a divisor of $GCD(N, q_i - 1)$ and $\deg(g_{0,i}) \geq (q_i - 1)/GCD(N, q_i - 1)$. So we first compute a divisor M of $q_i - 1$ such that $GCD(N, q_i - 1)$ divides M and $GCD(N, (q_i - 1)/M) = 1$. Next, by using the knowledge on the Galois group of $\mathbb{Q}(\zeta_{q_i})/\mathbb{Q}$, we compute a primitive element θ of a subfield K' of $\mathbb{Q}(\zeta_{q_i})$ with extension degree M. (See §4.) Then, the minimal polynomial of ζ_{q_i} over $K_f(\theta)$ coincides with the minimal polynomial of ζ_{q_i} over $\mathbb{Q}(\theta)$. Thus, instead of factoring $g_{0,i}$ over K_f, we factorize $g_{0,i}$ over $\mathbb{Q}(\theta)$ and factorize, over K_f, the minimal polynomial of θ over $\mathbb{Q}$ and then we obtain the representation of $K_f(\zeta_{q_i})$.

Finally, in this section we discuss two alternatives for the concrete procedures.

(1) Let ζ_q be a primitive q-th root of unity, where $q = \prod_{i=1}^{s} q_i$. Since ζ_q^{q/q_i} is a primitive q_i-th root of unity for each i, we can remove $z_1, \ldots, z_s$ by assigning a new variable z to ζ_q. This succeeds in reducing the number of variables, however, it becomes very hard to factorize the minimal polynomial of ζ_q over K_f. This fact can be seen in our experiment.
(2) We can give another procedure which seems quite natural with respect to the construction. Each p_i-th root is added to the subfield at each step, i.e. the following $\bar{L}_i$'s are constructed: $\bar{L}_0 = \mathbb{Q}(\zeta_{p_1}), \ldots, \bar{L}_i = \bar{L}_{i-1}(\beta_i, \zeta_{p_{i+1}}) = \mathbb{Q}(\beta_1, \ldots, \beta_i, \zeta_1, \ldots, \zeta_{p_{i+1}})$. We set $\bar{L}_r = L_r$. At each i-th step, we have to compute the minimal polynomial $\bar{m}_i$ of β_i over $\bar{L}_{i-1}$ and that $\bar{g}_{0,i}$ of ζ_{p_i} over $\bar{L}_{i-1}(\beta_i)$. These are also obtained by algebraic factorization. $\bar{m}_i$ is an irreducible factor of P_i over L_{i-1} and $\bar{g}_{0,i}$ is also an irreducible factor of $g_{0,i}$ over L_{i-1}. In the computational point of view, the factorization of P_i over L_{i-1} is reduced to those of $\zeta_{p_1}, \ldots, \zeta_{p_{i-1}}$ over K_i by basis-conversion techniques and vice versa. Thus we factorize $g_{0,i}$ over several proper subfields of L_f. This method is a variant of the procedure proposed in §6.2 with Improvement A.

7 Experiment on Computers

Here, we report our experiments on computation of radical representation for actual examples. In the previous sections, we showed the whole procedure consists of three parts; (1) computing a composition series (2) computing the corresponding subfield tower, and (3) computing radical representations. We implemented all three parts on Risa/Asir and tested efficiency of proposed methods for a number of examples. Timing data is shown in Appendix A. We note the splitting fields and Galois groups were computed by direct method in [1]. The experiment shows that radical representation is executable for polynomials whose splitting fields and Galois groups are computed by the direct method.

7.1 Algebraic factorization

Since we do not have an *effective* criterion for strong representation except Lemma 18, we have to factorize the minimal polynomials $g_{0,1}, \cdots, g_{0,s}$ over K_f or its extensions. For these factorizations, we can use an improved algebraic factorization using *non-square-free norm* proposed in [1], and its further extension proposed by Noro & Yokoyama [20] based on Encarnacion's algorithm [9] for factorization over simple extension fields. Since these two improvements work complementary, we can combine these effectively. By the former improvement, we sometimes catch an intermediate decomposition of a given polynomial, and by the latter improvement, we reduce unnecessary combinations of candidates for irreducible factors of the polynomial.

Timing data for algebraic factorization (Table 1 in Appendix A) was, in principle, obtained by the method using non-square-free norm. We remark that it took about 1 hour to execute algebraic factorization for Example (19) and we could not obtain the result for Example (26) within 1 hour. We applied the algorithm in [20] for Example (19), by which the computing time decreased to 1/20. For Example (26) we obtained the result in about 200 seconds by Improvement A. However, the part of factorizations of $g_{0,i}$'s is still time-consuming compared with basis-conversion parts.

The *improvements for algebraic factorization* proposed in §6.3 are based on the above algebraic factoring algorithms. The effects of each improvement depend on the cases. We comment on Improvement A briefly: If $g_{0,i}$ is factorized into its proper factors with smaller degree, the total efficiency should be much improved. Moreover, if the degree of irreducible factors of f over K_i is prime to $[K_f : K_i]$, each irreducible factor of f over K_i is also irreducible over K_f, that is, we do not have to factorize each factor over K_f. As for Improvement B, we did not apply it in the experiment. This seems much suited for the case where M is considerably small compared with N and $q_i - 1$.

7.2 Relative and absolutely primitive elements

From [2] and Table 2 in Appendix A, we can point out the following. (i) Relative primitive elements can be obtained more easily than absolutely ones. (ii) Coefficients of the minimal polynomials of relative primitive elements tend to be much smaller than those of absolutely ones. (iii) Although a method using absolutely primitive elements has a smaller number of variables, the computation of a polynomial expression of β_i is much more time-consuming. From these points, we conclude that methods using relative primitive elements are superior to methods using absolutely ones. This behavior is very similar to that of algebraic factorization, see [1].

7.3 Elimination ideal computation by basis-conversion techniques

As mensioned in Remark 2, there is a slight difference between finding necessary algebraic relations and elimination ideal computation by basis-conversion. But, as they have the same mathematical basis and computational behavior, we focus on elimination ideal computation by basis-conversion and we discuss which variant of basis-conversion is efficient for the radical representation. Since each ideal appearing in the problem is given by its Gröbner basis with respect to an admissible order, change-of-ordering algorithms are much more efficient than direct application of Gröbner basis computation. (Timing data in appendix was obtained by using the change-of-ordering algorithm in Risa/Asir.) There are several studies about change-of-ordering algorithms, e.g. Faugère *et al.* [11], Faugère [10]. Among those, we used one described in Noro & Yokoyama [19] as the most suitable one for the problem in our settings. Because the algorithm in [19] employs modular techniques to avoid intermediate coefficient growth.

As reported in [2], since the degree of each minimal polynomial in examples is a small prime, direct Gröbner basis computation with respect to a lexicographical order by an algorithm based on trace-lifting, see Traverso [23], also worked well in the experiment. However, even in these cases, the change-of-ordering algorithm employed here can compute much more efficiently. Efficiency will be much improved by stopping the basis-conversion when we obtain every necessary elements in the basis.

7.4 Radical representation of primitive roots of unity

We give a comment on radical representation of a primitive p-th root ζ_p of unity for a prime p. Since the Galois group of $\mathbb{Q}(\zeta_p)$ over $\mathbb{Q}$ is a cyclic group of order $p-1$, we can apply the general method for it, that is, we first compute a subfield tower, and so on. Moreover, we can also apply methods in §5 directly to it. By [2] it is proved that the method derived from Equation (3) is the most efficient, if a radical representation of

a primitive n-th root of unity for each n smaller than p is already known. Since radical representations of primitive roots of unity are used as basic items for radical representation for roots of general polynomials, these expressions must be filed as data. Hence, the shape of their radical expressions is more important than the timing of their computation. This should be analyzed in our future works.

8 Concluding remarks

By the direct method in [1] or the p-adic approach in [26], exact permutation representations of Galois groups are computed efficiently. Aiming at efficient computation of radical representation from the outputs of the direct method, we gave further discussion on the subject and proposed a concrete method. We examined the efficiency of the method by experiments with a number of examples. The experiment shows that radical representation is executable for polynomials whose splitting fields and Galois groups are computed by the direct method. That is, combination of the presented methods with the direct method and efficient group theoretical methods for composition series, gives a practical procedure for the radical representation of roots of polynomials. Since the presented method follows the well-known abstract procedure for radical representation, we may call the method a *direct* method for radical representation.

To improve the efficiency of the direct method, the technique of p-adic approach seems very useful. Because, as we can guess the shape of necessary algebraic relations, we can apply modular techniques, where we can compute the results in the finite field $GF(p)$ for some prime and lift them by Hensel procedure. This will be done in our further study.

Finally, we list up additional problems for further study: (1) By the presented direct method, radical representations of several polynomials were obtained. However, their expressions are too complicated to recognize what these expressions imply. (See Appendix B.) Thus, the problem to simplify expressions arises, or we have to consider what are preferable expressions (cf. [29]). (2) By using arguments in [15], it seems possible to give a bound of coefficients of radical representations in terms of the magnitude of coefficients and the degree of the input polynomial. If we have adequate bounds, we can apply modular techniques very efficiently.

Acknowledgment

The authors are grateful to Prof. J. McKay for reading a draft and giving us helpful comments. We are also indebted to Prof. M. Noro for his support on the experiment.

References

[1] Anai, H., Noro M., Yokoyama K. Computation of the splitting fields and the Galois groups of polynomials. *Progress in Mathematics* **143**, 29-50, 1996.

[2] Anai, H., Yokoyama K. Radical Representation of Polynomial Roots. ISIS Research Report 94-13E, 1994.

[3] Atkinson, M. D. An Algorithm for Finding the Blocks of a Permutation Group. *Math. Comp.* **29**, 911-913, 1975.

[4] Becker, T., Weispfenning, V. *Gröbner Bases*. Springer-Verlag, GTM 141, 1993.

[5] Buchberger, B. *Gröbner bases: an algorithmic method in polynomial ideal theory. Multidimensional System Theory*, Reidel Publ. Comp., pp. 184-232, 1985.

[6] Butler, G. *Fundamental Algorithms for Permutation Groups*. Lect. Notes in Comp. Sc., **559**, Springer-Verlag, 1991.

[7] Cox, D.A., Little, J.B., O'Shea, D.B. *Ideals, Varieties, and Algorithms*. Springer-Verlag, 1991.

[8] Dixon, J. Computing subfields in algebraic number fields. *J. Austral. Math. Soc. (Series A)* **49**, 434-448, 1990.

[9] Encarnacion, M. Faster algorithms for reconstructing rationals, computing polynomial gcds, and factoring polynomials. Ph.D. Thesis, RISC-Linz, 1995.

[10] Faugère, J.C. Résolution des sytèmes d'équations algébriques. Thesis Univ. Paris VI, 1994.

[11] Faugère, J.C., Gianni, P., Lazard, D., Mora, T. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. *J. Symb. Comp.* **16**, 329-344, 1993.

[12] Furst, M., Hopcroft, J., Luks, E. Polynomial-Time Algorithm for Permutation Groups. *Proc. Twenty-first Annu. IEEE Sympos. Found. Comput. Sci. 1980*, pp. 36-41, 1980.

[13] Hanrot, G., Morain, F. Solvability by radicals from an algorithmic point of view. In Proceedings of 2001 International Symposium on Symbolic and Algebraic Computation. ACM Press, pp.175-182, 2001.

[14] Hulpke, A. Techniques for the computation of Galois groups. *Algorithmic Algebra and Number Theory*, Springer-Verlag, pp.65-77, 1999.

[15] Landau, S., Miller, G. L. Solvability by radicals is in polynomial time. *J. Comput. System Sci.* **30**, 179-208, 1985.

[16] Lazard, D., Valibouze, A. Computing subfields: Reverse of the primitive element problem. *Proceedings of Effective Methods in Algebraic Geometry 1992.* Birkhäuser, pp. 163-176, 1992.

[17] Motoyoshi, F. (1997). Simplified representation of successive algebraic extension fields. RIMS (Research Institute for Mathematical Sciences, Kyoto University) Kokyuroku, No.986, pp.78-82. (in Japanese)

[18] Noro, M., Takeshima, T. Risa/Asir – a computer algebra system. *Proceedings of International Symposium on Symbolic and Algebraic Computation 1992*. New York: ACM Press, pp. 387-396, 1992.

[19] Noro, M., Yokoyama, K. A modular method to compute rational univariate representation of zero-dimensional ideals. *J. Sym. Comp.* **28**, 243-263, 1999.

[20] Noro, M., Yokoyama, K. Factoring polynomials over algebraic extension fields. *Josai Information Science Researches* **9**, 11-33, 1998.

[21] Sims, C. C. Computational methods in the study of permutation groups. *Computational Problems in Abstract Algebra*, Pergamon, Elmsford, pp. 169-183, 1970.

[22] Sturmfels, B. *Algorithms in invariant theory*. Texts and Monographs in Symbolic Computation, Springer-Verlag, 1993.

[23] Traverso, C. Gröbner trace algorithms. *International Symposium on Symbolic and Algebraic Computation 1988*. Lect. Notes in Comp. Sc. **358**, pp. 125-138, 1988.

[24] Valibouze, A. Computation of the Galois Groups of the Resolvent Factors for the Direct and Inverse Galois Problems. *AAECC-11* Lect. Notes in Comp. Sc. **948**, pp. 456-468, 1995.

[25] van der Waerden, B. L. *Algebra I.* Springer-Verlag, 1991.

[26] Yokoyama, K. A Modular Method for Computing the Galois Groups of Polynomials. *J. Pure and Applied Algebra* **117/118**, 617-636, 1997.

[27] Yokoyama, K., Noro, M., Takeshima, T. Computing Primitive Elements of Extension Fields. *J. Sym. Comp.* **8**, 553-380, 1989.

[28] Yokoyama, K., Noro, M., Takeshima, T., On determining the solvability of polynomials. *Proceedings of ISSAC '90*, ACM Press, pp. 127-134, 1990.

[29] Zippel, R. Simplification of Expressions Involving Radicals. *J. Sym. Comp.* **1**, 189-210, 1985.

Appendix A Timing data

Timings, given in seconds, were measured on a SUN4-20/61, where garbage collection time is excluded. As samples, we took the following 10 solvable polynomials. These but Example $(4')$ are quoted from Anai *et al.* (1994a) and the same indices are assigned to these polynomials. Example $(4')$ is given by Professor G. Fee in his *Computer Challenge Problems* [1] whose Galois group is isomorphic to Example (4) in Anai *et al.* (1994a).

$(4')$	$x^5 - 5x^3 + 5x - 5$	(17)	$x^6 + x^4 - 8$
(10)	$x^6 + 9x^4 - 4x^2 - 4$	(19)	$x^6 + x^4 - x^2 + 5x - 5$
(11)	$x^6 + x^3 + 7$	(24)	$x^7 + 7x^3 + 7x^2 + 7x - 1$
(12)	$x^6 - 3x^4 + 1$	(25)	$x^7 - 14x^5 + 56x^3 - 56x + 22$
(15)	$x^6 - 2x^3 - 2$	(26)	$x^7 - 2$

Table 1 shows the timings to compute *relative* primitive elements and their strong radical representations, and Table 2 shows the comparison of the method with relative primitive elements and the method with absolutely ones.

Here, we use the following abbreviations: In Table 1, $|G|$ denotes the order of the Galois group, D denotes the extension degree of each subfield over its previous subfield, S denotes the timing for computing each subfield, R denotes the timing for a strong radical representation of each primitive element, and af denotes the timing for computing C_f. A contractible root was found at the step marked by "▷". In this case we do not have to proceed further. And "*" means that contraction occurs at the extension. Moreover " [1] " means that we used the extension proposed by Noro & Yokoyama (1996), and "[2] " means that we used Improvement A. (See §6.3.) In Table 2, $|G|, D, S$ and R are the same as in Table 1. S^a denotes the timing to construct absolutely primitive elements and R^a denotes the timing for a strong radical representation of each absolutely primitive element.

Appendix B Result of $(4')$

SUBFIELDS : For Example $(4')$, K_f is obtained by adding two roots a, b of $(4')$ to $\mathbb{Q}$:

[1] 14 problems to challenge computer algebra systems are proposed in *Computer Challenge Problems* which was posted to Internet News Group: *sci.math.symbolic* on 7 Jun, 1994.

	$\|G\|$	D	S	R
$(4')$	20	2	1.48	* 0.07
F20		2	1.54	1.70
		5	0.67	3.36
af				7.15
(10)	12	3	0.04	0.12
A4		2	0.03	0.07
		2	0.02	0.06
af				0.50
(11)	18	3	0.12	0.16
3.S3		2	0.11	* 0.10
		3	0.06	0.28
af				1.73
(12)	24	3	0.16	0.20
2.A4		2	0.06	0.07
		2	▷ 0.07	0.07
		2	(0.06)	(0.08)
af				3.57
(15)	36	2	0.77	0.07
3^2.2^2		3	0.28	* 0.06
		2	0.08	0.23
		3	0.05	0.21
af				10.59

	$\|G\|$	D	S	R
(17)	48	2	1.22	0.10
2.S4		3	0.14	0.18
		2	0.10	0.13
		2	▷ 0.09	0.13
		2	(0.11)	(0.17)
af				61.16
(19)	72	2	23.74	0.15
3^2.D4		2	4.91	0.33
		2	7.67	3.03
		3	3.69	4.13
		3	7.89	7.33
af				[1] *171.00*
(24)	14	2	0.46	* 1.32
D7		7	0.35	14.81
af				14.70
(25)	21	3	1.46	* 2.19
F21		7	1.18	19.93
af				40.5
(26)	42	2	0.71	* 0.22
F42		3	0.19	* 0.50
		7	0.14	13.63
af				[2] *207.20*

Table 1: Time for radical representation.

	D	S	S^a	R	R^a
$(4')$	2	1.48	1.25	0.07	0.50
	2	1.54	1.29	1.70	1.62
	5	0.67	0.33	3.36	17.53
(10)	3	0.04	0.04	0.12	0.27
	2	0.03	0.03	0.07	0.03
	2	0.02	0.02	0.06	0.04
(11)	3	0.12	0.08	0.16	0.08
	2	0.11	0.07	0.10	0.07
	3	0.06	0.07	0.28	0.28
(12)	3	0.16	0.13	0.20	0.26
	2	0.06	0.05	0.07	0.04
	2	0.07	0.03	0.07	0.04
	2	0.06	0.09	0.08	0.12
(15)	2	0.77	0.68	0.07	0.04
	3	0.28	0.21	0.06	0.10
	2	0.08	0.06	0.23	0.49
	3	0.05	0.04	0.21	0.28

	D	S	S^a	R	R^a
(17)	2	1.22	1.12	0.10	0.08
	3	0.14	0.09	0.18	0.25
	2	0.10	0.04	0.13	0.06
	2	0.09	0.13	0.13	0.21
	2	0.11	4.24	0.17	1.52
(19)	2	23.74	24.06	0.15	0.13
	2	4.91	5.00	0.33	0.29
	2	7.67	20.44	3.03	8.52
	3	3.69	36.45	4.13	24.45
	3	7.89	0.72	7.33	7.13
(24)	2	0.46	0.51	1.32	0.59
	7	0.35	0.52	14.81	237.85
(25)	3	1.46	1.47	2.19	34.76
	7	1.18	1.58	19.93	364.88
(26)	2	0.71	0.79	0.22	0.36
	3	0.19	0.22	0.50	0.51
	7	0.14	0.07	13.63	17.67

Table 2: Comparison of relative primitive elements and absolutely ones

$$K_f \cong \mathbb{Q}[a,b]/Id(a^5 - 5a^3 + 5a - 5, a^4 + ba^3 + (b^2 - 5)a^2 + (b^3 - 5b)a + b^4 - 5b^2 + 5).$$

There are two subfields corresponding to the composition series of the Galois group between $\mathbb{Q}$ and K_f. A primitive element β_i of K_i over $K_{i-1} = \mathbb{Q}(\beta_1, \cdots, \beta_{i-1})$ and its minimal polynomial m_i over K_{i-1} for each i $(i = 1, 2, 3)$ are as follows:

$$\begin{aligned}
\beta_1 &= \tfrac{1}{27}((20b^3 - 50b^2 - 60b + 100)a^3 + (-50b^3 + 20b^2 + 150b - 40)a^2 + (-60b^3 + 150b^2 + \\
&\qquad 180b - 300)a + 100b^3 - 40b^2 - 300b - 550), \\
\beta_2 &= 10ba^4 + (-5b^2 + 20)a^3 + (15b^3 - 75b)a^2 + (25b^2 - 70)a - 20b^3 + 70b + 25, \\
\beta_3 &= -a + b,
\end{aligned}$$

$$\begin{aligned}
m_1 &= u_1^2 + 50u_1 + 500, \\
m_2 &= 2u_2^2 - 1155u_1 - 15750, \\
m_3 &= -2u_3^5 - u_1u_3^3 + (5u_1 + 50)u_3 + 2u_2.
\end{aligned}$$

Radical Representations : For each i, u_i and v_i are assigned to β_i and the corresponding Lagrange resolvent, respectively. z_1 is assigned to the fixed primitive 5-th root of unity, whose minimal polynomial over K_f is

$$\begin{aligned}
g_1(z_1; a, b) = 21z_1^2 + ((2b^3 - 5b^2 - 6b + 10)a^3 + (-5b^3 + 2b^2 + 15b - 4)a^2 + \\
(-6b^3 + 15b^2 + 18b - 30)a + 10b^3 - 4b^2 - 30b + 8)z_1 + 21.
\end{aligned}$$

Then the results are as follows:

$$\begin{aligned}
u_1 &= 10z_1^3 + 10z_1^2 - 20, \\
u_2 &= \tfrac{1}{2}v_2, \\
u_3 &= \tfrac{1}{31250}(((3z_1^3 + 3z_1^2 + 5)u2 + 50z_1^3 + 50z_1 1 + 25)v^4 + 6250v), \\
v_2 &= \sqrt{23100z_1^3 + 23100z_1^2 - 14700}, \\
v_3 &= \sqrt[5]{\tfrac{1}{2}(3125v_2 - 78125z_1^3 - 234375z_1^2 - 312500z_1 - 156250)},
\end{aligned}$$

$$\begin{aligned}
a &= \tfrac{1}{312500}(((7v_2 - 500)v_1^4 - 25000v_1)z_1^3 + ((-11v_2 - 250)v_1^4 + 12500v_1)z_1^2 + ((11v_2 - \\
&\qquad 250)v_1^4 - 12500v_1)z_1 + (-7v_2 - 500)v_1^4 - 37500v_1), \\
b &= \tfrac{1}{156250}(11v_2v_1^4 - 12500v_1)z_1^3 + ((2v_2 - 125)v_1^4 + 6250v_1)z_1^2 + ((11/2v_2 + 125)v_1^4 \\
&\qquad -6250v_1)z_1 + (9v_2 - 125)v_1^4 + 12500v_1.
\end{aligned}$$