A Speed-Up of the Algorithm for Computing Comprehensive Gröbner Systems

Katsusuke Nabeshima
Research Institute for Symbolic Computation (RISC-Linz),
Johannes Kepler University Linz, A-4040, Linz, Austria
Katsusuke.Nabeshima@risc.uni-linz.ac.at

Abstract

We introduce a new algorithm for computing comprehensive Gröbner systems. There exists the Suzuki-Sato algorithm for computing comprehensive Gröbner systems. The Suzuki-Sato algorithm often creates overmuch cells of the parameter space for comprehensive Gröbner systems. Therefore the computation becomes heavy. However, by using inequations ("not equal zero"), we can obtain different cells. In many cases, this number of cells of parameter space is smaller than that of Suzuki-Sato's. Therefore, our new algorithm is more efficient than Suzuki-Sato's one, and outputs a nice comprehensive Gröbner system. Our new algorithm has been implemented in the computer algebra system Risa/Asir. We compare the runtime of our implementation with the Suzuki-Sato algorithm and find our algorithm superior in many cases.

1 Introduction

Comprehensive Gröbner bases and comprehensive Gröbner systems for parametric ideals were introduced, constructed and studied by Weispfenning in 1992 [12]. Since then comprehensive Gröbner bases and systems have been studied by several researchers and implemented in several computer algebra systems [2, 6, 5, 10, 11, 13]. Roughly speaking, a comprehensive Gröbner system is a parametric Gröbner basis with cells of the parameter space for a parametric polynomial ideal. If we take a cell $\mathbb{P}$ and its set of parametric polynomials G from a comprehensive Gröbner systems for a parametric polynomial ideal I, then $\sigma(G)$ constitutes a Gröbner basis of the ideal generated by $\sigma(I)$ under the specialization σ with respect to the cell $\mathbb{P}$ of the parameters. This article describes a new algorithm for computing comprehensive Gröbner systems. There exists the Suzuki-Sato algorithm [11] for computing comprehensive Gröbner systems. In many cases, this algorithm is faster than other existing algorithms. In this paper, we improve the Suzuki-Sato algorithm by using inequations ("$\neq 0$") and Gröbner bases computation in a polynomial ring over a polynomial ring. If we compute a Gröbner basis for an ideal in a polynomial ring over a polynomial ring, then the Gröbner basis computed often has the special property $\diamondsuit 1$ (see section 4.1) which doesn't hold in a polynomial ring over a field. This property makes overmuch cells of the parameter space. Thus, the computation of comprehensive Gröbner systems becomes expensive. However, by using inequations ("$\neq 0$"), we can avoid this behavior. Therefore, we can compute a comprehensive Gröbner system much faster, and have a nice comprehensive Gröbner system. We implemented our new algorithm in the computer algebra system Risa/Asir [9]. Through our computation experiment, we checked that in many cases, our program runs more efficient than the Suzuki-Sato algorithm. Especially, if the number of parameters is greater than the number of variables, our algorithm is much more efficient than Suzuki-Sato's one. Moreover, the outputs of our program are much nicer than the Suzuki-Sato algorithm. That is, the number of cells of the outputs is smaller than Suzuki-Sato's outputs.

2 Notations

Let $\bar{A} := \{A_1, \dots, A_m\}$ and $\bar{X} := \{X_1, \dots, X_n\}$ be finite sets of variables such that $\bar{A} \cap \bar{X} = \emptyset$. K and L denote fields such that L is an extension of K. $\mathrm{pp}(\bar{X})$, $\mathrm{pp}(\bar{A})$ and $\mathrm{pp}(\bar{A}, \bar{X})$ denote the sets of power products of $\bar{X}$, $\bar{A}$ and $\bar{A} \cup \bar{X}$, respectively. $\mathbb{N}$, $\mathbb{Q}$ and $\mathbb{C}$ denote as the set of natural numbers with 0, the field of rational numbers and the field of complex numbers, respectively. In this paper, we define $K[\bar{A}, \bar{X}]$ as a polynomial ring over a field K and $K[\bar{A}][\bar{X}] := (K[\bar{A}])[\bar{X}]$ as a polynomial ring over a polynomial ring

$K[\bar{A}]$ (coefficients in a polynomial ring). Let f and g be non-zero polynomials in $K[\bar{A},\bar{X}]$ (or $K[\bar{A}][\bar{X}]$) and $\succ$ be an arbitrary term order on the set of power products $\mathrm{pp}(\bar{A},\bar{X})$ (or $\mathrm{pp}(\bar{X})$). If polynomials f and g are in $K[\bar{A}][\bar{X}]$, then we use the subscript $\bar{A}$ as follows:

• The **support of** f (written : $\mathrm{supp}(f)$ (or $\mathrm{supp}_{\bar{A}}(f)$)) is the set of power products of f that appear with a non-zero coefficient.

• The biggest power product of $\mathrm{supp}(f)$ (or $\mathrm{supp}_{\bar{A}}(f)$) w.r.t. $\succ$ is denoted by $\mathrm{lpp}(f)$ (or $\mathrm{lpp}_{\bar{A}}(f)$) and is called the **leading power product of** g **w.r.t.** $\succ$.

• The coefficient corresponding to $\mathrm{lpp}(f)$ (or $\mathrm{lpp}_{\bar{A}}(f)$) is called the **leading coefficient of** f **w.r.t.** $\succ$ which is defined by $\mathrm{lc}(f)$ (or $\mathrm{lc}_{\bar{A}}(f)$).

• The product $\mathrm{lc}(f)\,\mathrm{lpp}(f)$ is called the **leading monomial of** f **w.r.t.** $\succ$ which is defined by $\mathrm{lm}(f)$ (or $\mathrm{lm}_{\bar{A}}(f)$).

• The **set of monomials** of f is denoted by $\mathrm{Mono}(f)$ (or $\mathrm{Mono}_{\bar{A}}(f)$).

Let F be a subset of $K[\bar{A}][\bar{X}]$. We define $\mathrm{lc}(F) := \{\mathrm{lc}(f) : f \in F\}$, $\mathrm{lc}_{\bar{A}}(F) := \{\mathrm{lc}_{\bar{A}}(f) : f \in F\}$, $\mathrm{lpp}(f) := \{\mathrm{lpp}(f) : f \in F\}$ and $\mathrm{lpp}_{\bar{A}}(F) := \{\mathrm{lpp}_{\bar{A}}(f) : f \in F\}$.

Example 2.1. Let a, b, x, y be variables and $f = 2ax^2y + bx^2y + 3x + by + 1, g = abx^2 + 2xy + by + 2$ be polynomials. If we consider polynomials f and g as members of $\mathbb{Q}[a, b, x, y]$ with a block order $\succ$ such that $x \succ_{lex} y \gg a \succ_{lex} b$ where $\succ_{lex}$ is the lexicographic order, then : $\mathrm{supp}(f) = \{ax^2y, bx^2y, x, y, 1\}$, $\mathrm{lpp}(f) = ax^2y$, $\mathrm{lc}(f) = 2$, $\mathrm{lm}(f) = 2ax^2y$, $\mathrm{Mono}(f) = \{2ax^2y, bx^2y, 3x, by, 1\}$. If we consider polynomials f, g as members of $\mathbb{Q}[a, b][x, y]$ with the lexicographic order $x \succ_{lex} y$, then : $\mathrm{supp}_{\{a,b\}}(f) = \left\{x^2y, x, y, 1\right\}$, $\mathrm{lpp}_{\{a,b\}}(f) = x^2y$, $\mathrm{lc}_{\{a,b\}}(f) = 2a + b$, $\mathrm{lm}_{\{a,b\}}(f) = (2a + b)x^2y$, $\mathrm{Mono}_{\{a,b\}}(f) = \left\{(2a + b)x^2y, 3x, by, 1\right\}$.

In this paper, angle brackets $\langle \cdot \rangle$ are defined as follows: let $f_1, \ldots, f_l \in R$ where R is a commutative ring with identity. Then, $\langle f_1, \ldots, f_l\rangle := \{\sum_{i=1}^{s} h_i f_i : h_1, \ldots, h_s \in R\}$.

3 Suzuki-Sato's Algorithm

In this section we review the theory of stability of Gröbner bases and Suzuki-Sato's algorithm for computing comprehensive Gröbner systems. First, we describe the stability of Gröbner bases under specialization. This is the key theory to construct the algorithm for computing comprehensive Gröbner systems.

3.1 Stability of ideals

Here we describe the stability of Gröbner bases under specialization in $K[\bar{A}][\bar{X}]$ (see [4]). The following is a definition of Gröbner bases in $K[\bar{A}][\bar{X}]$.

Definition 3.1. Let I be an ideal in $K[\bar{A}][\bar{X}]$ and $\succ$ a term order on $\mathrm{pp}(\bar{X})$. A subset $G \subset K[\bar{A}][\bar{X}]$ is called a **Gröbner basis** of I w.r.t. $\succ$ if $\mathrm{lm}_{\bar{A}}(I)$ is generated by $\mathrm{lm}_{\bar{A}}(G)$.

By using a block order with $\bar{X} \gg \bar{A}$, we can easily compute a Gröbner basis for an ideal in $K[\bar{A}][\bar{X}]$ (see [1, 7, 11]). (In [7], Gröbner bases in $K[\bar{A}][\bar{X}]$ have been studied.)

Algorithm 3.2. GröbnerBasis$(F, \succ)$

Input F: a finite subset of $K[\bar{A}][\bar{X}]$, $\succ$: a term order on $\mathrm{pp}(\bar{X})$,
Output G: a Gröbner basis of $\langle F\rangle$ w.r.t. $\succ$ in $K[\bar{A}][\bar{X}]$.
(1) Consider F as a subset of $K[\bar{A},\bar{X}]$. (Clearly, $K[\bar{A}][\bar{X}]$ is isomorphic to $K[\bar{A},\bar{X}]$.)
(2) Compute the reduced Gröbner basis G for $\langle F\rangle$ w.r.t. a block order with $\bar{X} \gg \bar{A}$ in $K[\bar{A},\bar{X}]$.
(3) Consider G as a subset of $K[\bar{A}][\bar{X}]$. Then, G is a Gröbner basis for $\langle F\rangle$ w.r.t. $\succ$ in $K[\bar{A}][\bar{X}]$.

Remark: We don't need to compute reduced Gröbner bases in $K[\bar{A},\bar{X}]$. It suffices to compute a (normal) Gröbner basis. However, in algorithms Suzuki-Sato, NEW and the proof of Theorem 4.7, we need the properties of reduced Gröbner bases in $K[\bar{A},\bar{X}]$. Therefore, reduced Gröbner bases computation was built in the algorithm GröbnerBasis. (See [11].)

Every ring homomorphism $\sigma : K[\bar{A}] \to L$ extends naturally to a homomorphism $\sigma : K[\bar{A}][\bar{X}] \to L[\bar{X}]$. The image under σ of an ideal $I \subseteq K[\bar{A}][\bar{X}]$ generates the extension $\sigma(I) := \{\sigma(f) : f \in I\} \subseteq L[\bar{X}]$.

Definition 3.3. We call an ideal $I \subseteq K[\bar{A}][\bar{X}]$ **stable** under the ring homomorphism σ and a term order $\succ$ if it satisfies $\sigma(\mathrm{lm}_{\bar{A}}(I)) = \mathrm{lm}(\sigma(I))$ where $\sigma(\mathrm{lm}_{\bar{A}}(I)) := \{\sigma(\mathrm{lm}_{\bar{A}}(f)) : f \in I\}$ and $\mathrm{lm}(\sigma(I)) := \{\mathrm{lm}(f) : f \in \sigma(I)\}$.

In several papers [1, 3, 4], the stability of ideals under specialization was studied. The following theorem is the key theorem for constructing the Suzuki-Sato algorithm (also our new algorithm) for computing comprehensive Gröbner systems.

Theorem 3.4 (Kalkbrener (1997) [4]). Let σ be a ring homomorphism from $K[\bar{A}]$ to L, I an ideal in $K[\bar{A}][\bar{X}]$ and $G = \{g_1, \dots, g_s\}$ a Gröbner basis of I w.r.t. a term order $\succ$. We assume that the g_i's are ordered in such a way that there exists an $r \in \{1, \dots, s\}$ with $\sigma(\mathrm{lc}_{\bar{A}}(g_i)) \neq 0$ for $i \in \{1, \dots, r\}$ and $\sigma(\mathrm{lc}_{\bar{A}}(g_i)) = 0$ for $i \in \{r+1, \dots, s\}$. Then the following three conditions are equivalent.
(1) I is stable under σ and $\succ$.
(2) $\{\sigma(g_1), \dots, \sigma(g_r)\}$ is a Gröbner basis of $\sigma(I)$ w.r.t. the term order $\succ$.
(3) For every $i \in \{r+1, \dots, s\}$, $\sigma(g_i)$ is reducible to 0 modulo $\{\sigma(g_1), \dots, \sigma(g_r)\}$ in $L[\bar{X}]$.

3.2 Suzuki-Sato's algorithm

Here, we introduce Suzuki-Sato's algorithm [11] and definitions of comprehensive Gröbner systems. For arbitrary $\bar{a} \in L^m$, we can define the canonical specialization homomorphism $\sigma_{\bar{a}} : K[\bar{A}] \to L$ induce by $\bar{a}$, and we can naturally extend it to $\sigma_{\bar{a}} : K[\bar{A}][\bar{X}] \to L[\bar{X}]$. For $f_1, \dots, f_k \in K[\bar{A}]$, $\mathbb{V}(f_1, \dots, f_k) \subseteq L^m$ denotes the affine variety of $f_1, \dots, f_k$, i.e.,$\mathbb{V}(f_1, \dots, f_k) = \{\bar{a} \in L^m : f_1(\bar{a}) = \cdots = f_k(\bar{a}) = 0\}$.

Definition 3.5. Let F be a subset of $K[\bar{A}][\bar{X}]$, $\mathcal{A}_1, .., \mathcal{A}_l$ algebraically constructible subsets of L^m and $G_1, \dots, G_l$ subsets of $K[\bar{A}][\bar{X}]$. Let $\mathcal{S}$ be a subset of L^m such that $\mathcal{S} \subseteq \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_l$. A finite set $\mathcal{G} = \{(\mathcal{A}_1, G_1), \dots, (\mathcal{A}_l, G_l)\}$ of pairs is called a **comprehensive Gröbner system** on $\mathcal{S}$ for F if $\sigma_{\bar{a}}(G_i)$ is a Gröbner basis of the ideal $\langle \sigma_{\bar{a}}(F) \rangle$ in $L[\bar{X}]$ for each $i = 1, \dots, l$ and $\bar{a} \in \mathcal{A}_i$. Each $(\mathcal{A}_i, G_i)$ is called a **segment** of $\mathcal{G}$. We simply say $\mathcal{G}$ is a comprehensive Gröbner system for F if $\mathcal{S} = L^m$.

In this paper, we use an algebraically constructible set that has a form $\mathbb{V}(f_1, .., f_k) \setminus \mathbb{V}(g_1, .., g_l) \subseteq L^m$ where $f_1, .., f_k,\ g_1, .., g_l \in K[\bar{A}]$. In this paper, we assume the algorithm LCM which outputs the least common multiple. The next two lemmas are the direct consequences of Theorem 3.4.

Lemma 3.6. Let F be a subset of $K[\bar{A}][\bar{X}]$. Let G be a Gröbner basis for $\langle F \rangle$ in $K[\bar{A}][\bar{X}]$ w.r.t. a term order $\succ$. Suppose that $\{h_1, \dots, h_s\} := \{\mathrm{lc}_{\bar{A}}(g) : g \in G\}$ and $h := \mathsf{LCM}(h_1, \dots, h_s)$. Then, for any $\bar{a} \in L^m \setminus \mathbb{V}(h)$, $\sigma_{\bar{a}}(G)$ is a Gröbner basis for $\langle \sigma_{\bar{a}}(F) \rangle$ w.r.t. $\succ$ in $L[\bar{X}]$.

Lemma 3.7. Let F be a subset of $K[\bar{A}][\bar{X}]$ and S a subset of $K[\bar{A}]$. Let G be a Gröbner basis for $\langle F \cup S \rangle$ in $K[\bar{A}][\bar{X}]$ w.r.t. a term order $\succ$. Suppose that $B := \{b : b \in \langle S \rangle, b \in G\}$, $\{h_1, \dots, h_s\} := \{\mathrm{lc}_{\bar{A}}(g) : g \in G \setminus B\}$ and $h := \mathsf{LCM}(h_1, \dots, h_s)$. Then, for any $\bar{a} \in \mathbb{V}(S) \setminus \mathbb{V}(h)$, $\sigma_{\bar{a}}(G)$ is a Gröbner basis for $\langle \sigma_{\bar{a}}(F) \rangle$ w.r.t. $\succ$ in $L[\bar{X}]$. Actually, we have $\sigma_{\bar{a}}(G) = \sigma_{\bar{a}}(G \setminus B)$.

Proof. If we take $g \in G \setminus B$, then for all $\bar{a} \in \mathbb{V}(S) \setminus \mathbb{V}(h)$ we have $\sigma_{\bar{a}}(\mathrm{lc}_{\bar{A}}(g)) \neq 0$. If we take $g \in G \cap B$, then we have $\sigma_{\bar{a}}(g) = 0$ and $\sigma_{\bar{a}}(\mathrm{lc}_{\bar{A}}(g)) = 0$. Of course, $\langle 0 \rangle$ is stable. Therefore, G is stable under the specialization $\sigma_{\bar{a}}$. By Theorem 3.4, $\sigma_{\bar{a}}(G) = \sigma_{\bar{a}}(G \setminus B)$ is a Gröbner basis for $\langle \sigma_{\bar{a}}(F) \rangle$. □

By Lemma 3.6 and Lemma 3.7, we can construct an algorithm for computing comprehensive Gröbner systems [11].

Algorithm 3.8. $\mathsf{Suzuki\text{-}Sato}(F, \succ)$ [11]

Input F: a finite subset of $K[\bar{A}][\bar{X}]$, $\succ$: a term order on $\mathrm{pp}(\bar{X})$,
Output G: a comprehensive Gröbner system for $\langle F \rangle$ w.r.t. $\succ$ on L^m.
begin $G \leftarrow \mathsf{CGSMain}(F, \emptyset, \succ)$;
$\mathsf{return}(G)$
end

Algorithm 3.9. $\mathsf{CGSMain}(F, Z, \succ)$

Input F: a finite subset of $K[\bar{A}][\bar{X}]$, Z : a finite set of polynomials in $K[\bar{A}]$, $\succ$: a term order on $\mathrm{pp}(\bar{X})$,
Output H: a comprehensive Gröbner system for $\langle F \rangle$ on $\mathbb{V}(Z)$.
begin
  $G \leftarrow \mathsf{GröbnerBasis}(F, \succ)$
if $1 \in G$ **then** $H \leftarrow \{(Z, \{1\}, \{1\})\}$
else

$G' \leftarrow G \backslash \{g : g \in G \cap K[\bar{A}], g \in \langle Z \rangle\}$; $S \leftarrow \{h_1, .., h_l\} := \{\text{lc}_{\bar{A}}(f) : f \in G'\}$ $(**)$
if $S \neq \emptyset$ **then**
$h \leftarrow \mathsf{LCM}(h_1, \ldots, h_l)$;
$H \leftarrow \{(Z, \{h\}, G')\} \cup \mathsf{CGSMain}(G \cup \{h_1\}, Z \cup \{h_1\}, \succ) \cup \mathsf{CGSMain}(G \cup \{h_l\}, Z \cup \{h_l\}, \succ)$
else
$H \leftarrow \{(Z, \{1\}, G')\}$
end-if
end-if
return(H)
end

Remark : We can apply a lot of optimization techniques to obtain small and nice outputs comprehensive Gröbner systems. For instance; we can check all cells of the output (condition of parameters) after the algorithm terminates, and in $(**)$, we can factorize all elements into irreducible factors.

4 A New Algorithm

This section is the main part of this paper. In this section, we introduce a new algorithm for computing comprehensive Gröbner systems. First, we motivate the new algorithm in order to facilitate the understanding of the algorithm.

4.1 Motivation

Let F be a subset of $K[\bar{A}][\bar{X}]$. Then, by the algorithm GröbnerBasis we can compute a Gröbner basis $G = \{g_1, .., g_l\}$ for $\langle F \rangle$ w.r.t. $\succ$ in $K[\bar{A}][\bar{X}]$. The Gröbner basis G in $K[\bar{A}][\bar{X}]$ often (not always) has the following property (because the coefficient domain is the polynomial ring $K[\bar{A}]$)

($\diamondsuit 1$)

$$g_i, g_j \in G \text{ such that } \text{lpp}_{\bar{A}}(g_i) \,|\, \text{lpp}_{\bar{A}}(g_j) \text{ and } g_i \neq g_j.$$

If we consider a reduced Gröbner basis for a given ideal in $K[\bar{X}]$ (a polynomial ring over a field), then the reduced Gröbner basis doesn't hold this property. Actually, when we compute a comprehensive Gröbner system for a given ideal, this property often makes a lot of small and unnecessary cells of the parameter space. Hence, our strategy for computing comprehensive Gröbner systems is **"avoiding this property by using inequations ($\neq 0$)"**. Before describing our algorithm, we consider the Suzuki-Sato algorithm and our idea.

The first step of Suzuki-Sato$(F, \succ)$ works as Figure 1. That is, we have to consider l cases $\text{lc}_{\bar{A}}(g_1) = 0, \ldots, \text{lc}_{\bar{A}}(g_l) = 0$ for computing a comprehensive Gröbner system for $\langle F \rangle$.

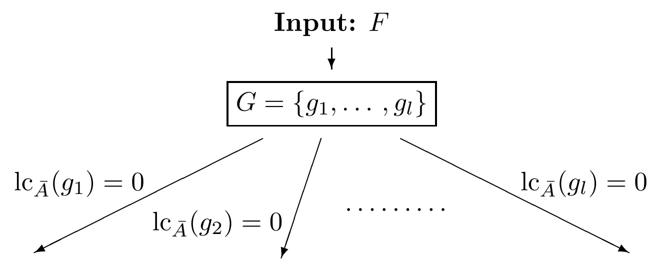

Figure 1

The Suzuki-Sato algorithm doesn't apply inequations (" $\neq 0$") for computing comprehensive Gröbner systems. In this point, this algorithm is extremely simple. However, as we said the above, Gröbner bases in $K[\bar{A}][\bar{X}]$ have the special property ($\diamondsuit 1$). Hence, the Suzuki-Sato algorithm provides overmuch cells of the parameter space. This condition (overmuch cells) is not nice when we compute a comprehensive Gröbner systems. Probably, by using inequations (" $\neq 0$"), we can obtain the number of cells which are smaller than Suzuki-Sato's outputs. This means that we may compute a comprehensive Gröbner systems more efficient than the Suzuki-Sato algorithm. (In the next subsection, we will discuss about this theory.)

When and how do we use inequations?

If there exists $g_i \in G$ such that $\mathrm{lpp}_{\bar{A}}(g_i) = 1$, then we don't need to consider l cases. We need to consider only one case (branch) $\mathrm{lc}_{\bar{A}}(g_i) = 0$, because for $\bar{a} \in L^m \backslash \mathbb{V}(\mathrm{lc}_{\bar{A}}(g_i))$, the Gröbner basis of $\sigma_{\bar{a}}(F)$ is $\{1\}$. That is, if $\mathrm{lc}_{\bar{A}}(g_i) \neq 0$, then we can decide one segment without computing the cases $\mathrm{lc}_{\bar{A}}(g_j) = 0$ for $1 \leq j \neq i \leq l$. Therefore, we can remove the cases left by the inequations $\mathrm{lc}_{\bar{A}}(g_i) \neq 0$. Suppose that for $p \in G$, $G_p := \{g \in G \backslash \{p\} : \mathrm{lpp}_{\bar{A}}(p) | \mathrm{lpp}_{\bar{A}}(g)\}$. If G_p isn't an empty-set, for $\bar{a} \in L^m \backslash \mathbb{V}(\mathrm{lc}_{\bar{A}}(p))$, all elements of $\mathrm{lpp}_{\bar{A}}(G_p)$ can be reduced by $\sigma_{\bar{a}}(\mathrm{lpp}_{\bar{A}}(p))$. Therefore, we don't need to consider the cases $\mathrm{lc}_{\bar{A}}(g_{pi}) = 0$ where $g_{pi} \in G_p$. That is, the set $\mathrm{lc}_{\bar{A}}(G_p)$ can be removed by $\mathrm{lc}_{\bar{A}}(p) \neq 0$. If so, we can construct a new algorithm for computing comprehensive Gröbner systems which is more efficient than the Suzuki-Sato one. If G_p is an empty-set, then we can follow the Suzuki-Sato algorithm. This is our main strategy for computing comprehensive Gröbner systems.
Now we have a specific example for computing a comprehensive Gröbner system. Let $F = \{ax^3, bx^2, cx\} \in \mathbb{Q}[a, b, c][x]$ where a, b, c are parameters and x is a variable. First, we consider the Suzuki-Sato algorithm. The Suzuki-Sato algorithm works as Figure 2.

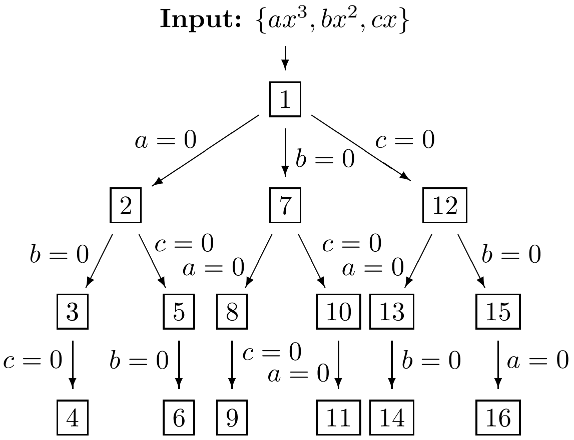

Figure 2

A comprehensive Gröbner basis for $\langle F \rangle$ is $\left\{ \boxed{1}, \boxed{2}, \ldots, \boxed{16} \right\}$. Of course, we can use several optimization techniques for getting small and nice comprehensive Gröbner systems. However, basically, the Suzuki-Sato algorithm works as Figure 2. The algorithm (with several techniques) has been implemented in the computer algebra system Risa/Asir. The program outputs the following as a comprehensive Gröbner systems for $\langle F \rangle$.

$$\begin{cases} \{x\}, & \text{if } a = 0, cb \neq 0, \\ \{x\}, & \text{if } a = b = 0, c \neq 0 \\ \{x\} & \text{if } b = 0, ac \neq 0 \\ \{x^2\} & \text{if } a = c = 0, b \neq 0 \\ \{\emptyset\} & \text{if } a = b = c = 0, \\ \{x^3\} & \text{if } c = b = 0, a \neq 0, \\ \{x^2\}, & \text{if } c = 0, ab \neq 0 \\ \{x\} & \text{if } abc \neq 0. \end{cases}$$

The program outputs 8 segments.
Next we try to apply our idea which uses inequations "$\neq 0$". First, we compute a Gröbner basis $S1$ for $\langle F \rangle$ in $\mathbb{Q}[a, b, c][x]$. Then, $S1 = \{ax^3, bx^2, cx\}$, and we know that $\mathrm{lpp}_{\{a,b,c\}}(cx) = x$, $\mathrm{lpp}_{\{a,b,c\}}(bx^2) = x^2$, $\mathrm{lpp}_{\{a,b,c\}}(ax^3) = x^3$. Clearly, $x|x^2$ and $x|x^3$. Hence, if $\mathrm{lc}_{\{a,b,c\}}(cx) = c \neq 0$, then a Gröbner basis for $\langle F \rangle$ is $\{x\}$. Since $L^3 \backslash \mathbb{V}(c)$ $(c \neq 0)$ cannot cover the whole space L^3, next we have to consider the case $c = 0$. If $c = 0$, then we have $S2 = \{ax^3, bx^2\}$ which is a Gröbner basis for $\langle S2 \rangle$ (itself). Clearly, $\mathrm{lpp}_{\{a,b,c\}}(bx^2) | \mathrm{lpp}_{\{a,b,c\}}(ax^3)$. Hence, if $c = 0$ and $b \neq 0$, then a Gröbner basis for $\langle F \rangle$ is $\{x^2\}$. Finally, we have to consider the cases $a \neq 0$ and $a = 0$. Therefore, our idea works as Figure 3. Our idea returns

the following comprehensive Gröbner system for $\langle F \rangle$.

$$\begin{cases} \boxed{1} = [\{x\}, & \text{if } c \neq 0], \\ \boxed{2} = [\{x^2\}, & \text{if } c = 0, b \neq 0], \\ \boxed{3} = [\{x^3\} & \text{if } b = c = 0, a \neq 0], \\ \boxed{4} = [\{\emptyset\} & \text{if } a = b = c = 0]. \end{cases}$$

This comprehensive Gröbner system has 4 segments.
As we saw, our computation process Figure 3 is simpler than Suzuki-Sato's one Figure 2. Furthermore, the output of our approach has only 4 segments which is smaller than Suzuki-Sato's one. Therefore, our approach is much more efficient than the Suzuki-Sato algorithm.

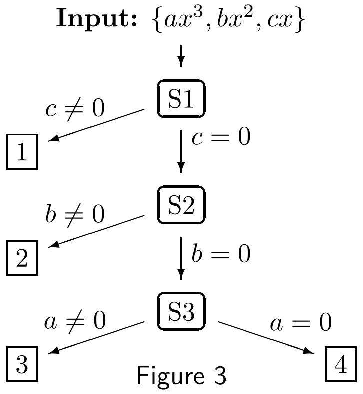

Figure 3

In the next subsection, we will describe our approach strictly, and give a new algorithm for computing comprehensive Gröbner systems.

4.2 A New Algorithm

In this subsection, we give a new algorithm for computing comprehensive Gröbner systems. The following theorem is the main idea for constructing the new algorithm.

Theorem 4.1. Let F be a subset of $K[\bar{A}][\bar{X}]$, $H = \{g, g_1, \ldots, g_l\}$ a Gröbner basis for $\langle F \rangle$ w.r.t. $\succ$. Select g from H, and set $r := \frac{1}{\mathrm{lc}_{\bar{A}}(g)}$ (r is a new variable) and $g' := \mathrm{lpp}_{\bar{A}}(g) + r \cdot (g - \mathrm{lm}_{\bar{A}}(g))$. Suppose that $H' := (H \backslash \{g\}) \cup \{g'\} = \{g', g_1, \ldots, g_l\} \subseteq K[r, \bar{A}][\bar{X}]$, and G' is a Gröbner basis of H' w.r.t. $\succ$ in $K[r, \bar{A}][\bar{X}]$. Furthermore, $G := \{f \in K[\bar{A}][\bar{X}] : f \neq 0, f = \mathrm{lc}_{\bar{A}}(g)^k \cdot \sigma_{r=\frac{1}{\mathrm{lc}_{\bar{A}}(g)}}(q), \deg_r(q) = k \in \mathbb{N}, q \in G'\}$ and $\{h_1, \ldots, h_e\} := \{\mathrm{lc}_{\bar{A}}(f) \in K[\bar{A}] : f \in G\}$. Then, for any $\bar{a} \in L^m \backslash (\mathbb{V}(\mathrm{lc}_{\bar{A}}(g)) \cup \mathbb{V}(h))$, $\sigma_{\bar{a}}(G)$ is a Gröbner basis for $\langle \sigma_{\bar{a}}(F) \rangle$ w.r.t. $\succ$ in $L[\bar{X}]$ where $h = \mathsf{LCM}(h_1, \cdots, h_e)$. ($\sigma_{r=\frac{1}{\mathrm{lc}_{\bar{A}}(g)}}(q)$ means substituting $\frac{1}{\mathrm{lc}_{\bar{A}}(g)}$ for the variable r of q.)

Proof. For all $\bar{a} = (a_1, \ldots, a_m) \in L^m \backslash (\mathbb{V}(\mathrm{lc}_{\bar{A}}(g)) \cup \mathbb{V}(h))$, we have $\sigma_{\bar{a}}(\mathrm{lc}_{\bar{A}}(g)) \neq 0$, and $\sigma_{\bar{a}}(r) \neq 0$. Set $\bar{b} := (a_1, \ldots, a_n, \frac{1}{\sigma_{\bar{a}}(\mathrm{lc}(g))}) \in L^{m+1}$. By the definition of G', for all $p \in G'$, we have $\sigma_{\bar{b}}(\mathrm{lm}_{\bar{A}}(p)) \neq 0$. Hence, by Theorem 3.6, $\sigma_{\bar{b}}(G')$ is a Gröbner basis for $\langle \sigma_{\bar{b}}(H') \rangle$ w.r.t. $\succ$. Actually, $\langle \sigma_{\bar{b}}(G') \rangle = \langle \sigma_{\bar{a}}(G) \rangle$. Therefore, $\sigma_{\bar{a}}(G)$ is a Gröbner basis for $\langle \sigma_{\bar{b}}(H') \rangle$ w.r.t. $\succ$, too. Since $\sigma_{\bar{a}}(g_i) = \sigma_{\bar{b}}(g_i)$ and $\langle \sigma_{\bar{a}}(g) \rangle = \langle \sigma_{\bar{b}}(g') \rangle$ for $1 \leq i \leq l$, we have $\langle \sigma_{\bar{b}}(H') \rangle = \langle \sigma_{\bar{a}}(H) \rangle$. As $\sigma_{\bar{a}}$ is a ring homomorphism, obviously we have $\langle \sigma_{\bar{a}}(H) \rangle = \langle \sigma_{\bar{a}}(F) \rangle$. Therefore, $\sigma_{\bar{a}}(G)$ is a Gröbner basis for $\langle \sigma_{\bar{a}}(F) \rangle$ w.r.t. $\succ$. $\square$

The following corollary are the direct consequence of Theorem 4.1. Actually, after applying Theorem 4.1, we need the following corollary for computing comprehensive Gröbner systems.

Corollary 4.2. Let F be a subset of $K[\bar{A}][\bar{X}]$, Z_1, Z_2 subsets of $K[\bar{A}]$ such that $\langle Z_1 \rangle \not\subset \langle Z_2 \rangle$, and $H = \{g, g_1, \ldots, g_l\}$ a Gröbner basis for $\langle F \cup Z_1 \rangle$ w.r.t. a term order $\succ$. Select $g \in H$ such that $g \notin Z_1$, and set $r := \frac{1}{\mathrm{lc}_{\bar{A}}(g)}$ and $g' := \mathrm{lpp}_{\bar{A}}(g) + r \cdot (g - \mathrm{lm}_{\bar{A}}(g))$. Suppose that $H' := (H \backslash \{g\}) \cup \{g'\} = \{g', g_1, \ldots, g_l\} \subset K[r, \bar{A}][\bar{X}]$, and G' is a Gröbner basis of H' w.r.t. $\succ$ in $K[r, \bar{A}][\bar{X}]$. Furthermore, $G := \{f \in K[\bar{A}][\bar{X}] : f \neq 0, f = \mathrm{lc}_{\bar{A}}(g)^k \cdot \sigma_{r=\frac{1}{\mathrm{lc}_{\bar{A}}(g)}}(q), \deg_r(q) = k \in \mathbb{N}, q \in G'\}$ and $\{h_1, \ldots, h_e\} := \{\mathrm{lc}_{\bar{A}}(f) \in K[\bar{A}] : f \in G\}$. Then, for any $\bar{a} \in \mathbb{V}(Z_1) \backslash (\mathbb{V}(Z_2) \cup \mathbb{V}(\mathrm{lc}_{\bar{A}}(g)) \cup \mathbb{V}(h))$, $\sigma_{\bar{a}}(G)$ is a Gröbner basis for $\langle \sigma_{\bar{a}}(F) \rangle$ w.r.t. $\succ$ in $L[\bar{X}]$ where $h = \mathsf{LCM}(h_1, \cdots, h_e)$.

Now, we can construct a new algorithm by using Lemma 3.6, Lemma 3.7, Theorem 4.1 and Corollary 4.2. Before describing the algorithm, we give one example for computing a comprehensive Gröbner system by using our strategy.

Example 4.3. Let $F = \{xy+x, ax^2+y+2, bxy+y\}$ be a subset of $\mathbb{Q}[a,b][x,y]$, a, b parameters and x, y variables. We have the lexicographic order $\succ$ such that $x \succ y$. Let's compute a comprehensive Gröbner system for $\langle F \rangle$ w.r.t. $\succ$.

(1) First, we compute a Gröbner basis of $\langle F \rangle$ in $\mathbb{Q}[a,b][x,y]$ by the algorithm GröbnerBasisB. Then, GröbnerBasisB$(F, \succ)$ outputs $\{a+b^2, y+1, bx+1, ax-b\}$. Clearly, for $\alpha \in \mathbb{C}^2 \setminus \mathbb{V}(a+b^2)$, $\{1\}$ is the Gröbner basis for $\langle \sigma_\alpha(F) \rangle$ w.r.t. $\succ$. That is, one of segments of a comprehensive Gröbner system for $\langle F \rangle$ is $(\mathbb{C}^2 \setminus \mathbb{V}(a+b^2), \{1\})$.

(2) Next, we have to consider the case $\{a+b^2 = 0\}$. By Lemma 3.7, we can obtain one segment $(\mathbb{V}(a+b^2) \setminus \mathbb{V}(ab), \{y+1, bx+1, ax-b\})$. However, this procedure is the same as Suzuki-Sato's one. As we are considering a new algorithm by using Theorem 4.1, we do not apply this procedure. Since we use Theorem 4.1, we have to select one polynomial from $\{y+1, bx+1, ax-b\}$. Now we have a question
"**Which polynomial had we better select to compute a comprehensive Gröbner system efficiently?**"
This answer is very important for our new algorithm. In this example, we know that $\mathrm{lpp}_{\{a,b\}}(bx+1)$ divides $\mathrm{lpp}_{\{a,b\}}(ax-b)$, and $\mathrm{lpp}_{\{a,b\}}(ax-b)$ divides $\mathrm{lpp}_{\{a,b\}}(bx+1)$. Hence, if we select $bx+1$ (or $ax-b$), then $ax-b$ (or $bx+1$) can be reduced by $\mathrm{lpp}_{\{a,b\}}(bx+1)$ (or $\mathrm{lpp}_{\{a,b\}}(ax-b)$). Therefore, we had better select one of $bx+1, ax-b$. Let's select $ax-b$. In order to follow Theorem 4.1, we replace $ax-b$ as $x-br$ where r is a new variable and $r := \frac{1}{a}$.

(3) Now we are considering the case $\{a+b^2 = 0, a \neq 0\}$. We compute a Gröbner basis for $\langle a+b^2, y+1, bx+1, x-br \rangle$ in $\mathbb{Q}[a,b,r][x,y]$ w.r.t. $\succ$. Then the algorithm GröbnerBasisB outputs $\{-ar+1, a+b^2, y+1, x-br\}$. Since we are considering the case $\{a+b^2 = 0, a \neq 0\}$ (and $r = \frac{1}{a}$), we do not need $-ar+1, -a-b^2$ from the set and we can replace $x-br$ as $ax-b$. By Theorem 4.1, for $\alpha \in \mathbb{V}(a+b^2) \setminus \mathbb{V}(a)$, $\{ax-b, y+1\}$ is a Gröbner basis for $\langle \sigma_\alpha(F) \rangle$ in $\mathbb{C}[x,y]$. That is, one of the segments is $(\mathbb{V}(a+b^2) \setminus \mathbb{V}(a), \{ax-b, y+1\})$.

(4) Finally, we have to consider the case $\{a+b^2 = 0, a = 0\}$. We can simplify the case into $\{a = 0, b = 0\}$. In this case, clearly, the Gröbner basis is $\{1\}$. Therefore, a comprehensive Gröbner system for $\langle F \rangle$ w.r.t. $\succ$ is $\{(\mathbb{C}^2 \setminus \mathbb{V}(a+b^2), \{1\}), (\mathbb{V}(a+b^2) \setminus \mathbb{V}(a), \{ax-b, y+1\}), (\mathbb{V}(a,b), \{1\})\}$. That is,

$$\begin{cases} \{1\}, & \text{if } a+b^2 \neq 0, \\ \{ax-b, y+1\}, & \text{if } a+b^2 = 0, a \neq 0, \\ \{1\} & \text{if } a = b = 0. \end{cases}$$

Let G be a Gröbner basis for an ideal I in $K[\bar{A}][\bar{X}]$. Suppose that

$$E := \{f \in G : \exists g \in G \backslash \{f\} s.t. \, \mathrm{lpp}_{\bar{A}}(f) | \, \mathrm{lpp}_{\bar{A}}(g)\}.$$

When we apply Theorem 4.1 for computing comprehensive Gröbner systems, we have to select one polynomial from G. Then, we have the following question.
Which polynomial should we select in order to compute comprehensive Gröbner systems efficiently?
In Example 4.3 (2), we selected $ax-b$, because $\mathrm{lpp}_{\{a,b\}}(ax-b)$ divides $\mathrm{lpp}_{\{a,b\}}(bx+1)$. In this case, we have $E = \{ax-b, bx+1\}$. If E is an empty set, then in Theorem 4.3, we have always $\mathrm{lpp}(\sigma_{\bar{a}}(H)) = \mathrm{lpp}(\sigma_{\bar{a}}(H'))$ for any $\bar{a} \in L^m \backslash (\mathbb{V}(\mathrm{lc}_{\bar{A}}(g)) \cup \mathbb{V}(h)) = L^m \setminus \mathbb{V}(h)$. (By the theorem, clearly $\sigma_{\bar{a}}(H)$ and $\sigma_{\bar{a}}(H')$ is a Gröbner basis for $\langle F \rangle$ w.r.t. $\succ$ in $L[\bar{X}]$.) Namely, in this case, we should not apply Theorem 4.3, because we have $\mathrm{lc}_{\bar{A}}(H) \backslash K = (\mathrm{lc}_{\bar{A}}(H') \cup \{\mathrm{lc}_{\bar{A}}(g)\}) \backslash K$. That is, the cells of the parameter space can not be changed by the selected polynomial. In this case, we apply Suzuki-Sato's approach. **If E is not an empty set, then our approach (Theorem 4.1) works powerfully for computing comprehensive Gröbner systems.** In fact, it is often happened that E is not an empty set. Our answer of the question is that "selecting one element from E". In the new algorithm which is the following, like normal strategy of Gröbner bases computation we select one polynomial from E which have the **lowest** leading power product in $\mathrm{lpp}_{\bar{A}}(E)$ w.r.t. a term order.
In the new algorithm NEW, we assume the algorithm factorize. The algorithm factorize(h) outputs a set

of all irreducible factors of h in $K[\bar{A}]$ where $h \in K[\bar{A}]$.
In the remark of the algorithm NEW, we describe why we input a natural number U in the algorithm NEW.

Algorithm 4.4. $\mathsf{NEW}(F, U, \succ)$

Input F: a finite subset of $K[\bar{A}][\bar{X}]$, $\succ$: a term order on $\mathrm{pp}(\bar{X})$, U: a natural number ($< \infty$),
Output G: a comprehensive Gröbner system for $\langle F \rangle$ w.r.t. $\succ$ on L^m.
begin
  $G \leftarrow \mathsf{NewCGSMain}(F, \emptyset, \emptyset, 1, 0, \succ, U)$
  $\mathsf{return}(G)$
end

Algorithm 4.5. $\mathsf{NewCGSMain}(F, L_1, L_2, D, N, \succ, U)$

Input F : a finite subset of $K[r, \bar{A}][\bar{X}]$,
L_1 : a finite set of polynomials in $K[\bar{A}]$ "$(= 0)$"
L_2 : a finite set of polynomials in $K[\bar{A}]$ "$(\neq 0)$",
D: a polynomial in $K[\bar{A}]$, U: a natural number ($< \infty$),
$\succ$: a term order on $\mathrm{pp}(\bar{X})$, N: a natural number ($< U$),
Output H: a comprehensive Gröbner system for $\langle F \rangle$ w.r.t. $\succ$ on $\mathbb{V}(L_1) \backslash \mathbb{V}(L_2)$.
begin
$G \leftarrow \mathsf{Gr\ddot{o}bnerBasisB}(F \cup L_1, \succ)$ in $K[r, \bar{A}][\bar{X}]$
$G^* \leftarrow \mathsf{Transform}(G, D)$
$G_1 \leftarrow G^* \backslash \{g : g \in G^* \cap K[\bar{A}], g \in \langle L_1 \rangle\}$
$E \leftarrow \{f \in G_1 : \exists g \in G_1 \backslash \{f\}$ s.t. $\mathrm{lpp}_{\bar{A}}(f) | \mathrm{lpp}_{\bar{A}}(g)\}$
if $E \neq \emptyset$ and $N \leq U$ **then**
  Select q from E s.t. $\mathrm{lpp}_{\bar{A}}(q)$ is the lowest element in $\mathrm{lpp}_{\bar{A}}(E)$ w.r.t. $\succ$ ($r := \mathrm{lc}_{\bar{A}}(q)^{-1}$, i.e., r is the new variable.)
  $q^* \leftarrow \mathrm{lpp}_{\bar{A}}(q) + r \cdot (q - \mathrm{lm}_{\bar{A}}(q))$ (i.e., $\mathrm{lc}_{\bar{A}}(q^*) = 1$)
  $F^* \leftarrow (G_1 \backslash \{q\}) \cup \{q^*\}$
  $\{t_1, \ldots, t_k\} \leftarrow \mathsf{factorize}(\mathrm{lc}_{\bar{A}}(q))$
  $t \leftarrow t_1 \cdot t_2 \cdots t_k$
   if $\mathbb{V}(L_1) \backslash (\mathbb{V}(t) \cup \bigcup_{s \in L_2} \mathbb{V}(s)) \neq \emptyset$ **then** (♣1)
   $N \leftarrow N + 1$
   $H_1 \leftarrow \mathsf{NewCGSMain}(F^*, L_1, L_2 \cup \{t\}, \mathrm{lc}_{\bar{A}}(q), N, \succ, U)$
   end-if
$H_2 \leftarrow \mathsf{NewCGSMain}(G_1, L_1 \cup \{t_1\}, L_2, \emptyset, 0, \succ, U) \cup \cdots \cup \mathsf{NewCGSMain}(G_1, L_1 \cup \{t_k\}, L_2, \emptyset, 0, \succ, U)$
  $H \leftarrow H_1 \cup H_2$
else
$S \leftarrow \{h_1, .., h_l\} := \{f : \mathbb{V}(f) \not\subset \bigcup_{s \in L_2} \mathbb{V}(s), f \in \mathsf{factorize}(\mathrm{lc}_{\bar{A}}(g)), \mathrm{lc}_{\bar{A}}(g) \notin K, g \in G_1\}$ (♣2)
  $h \leftarrow \mathsf{LCM}(h_1, \ldots, h_l)$
  $H \leftarrow \{(L_1, \{h\}, G_1)\}$
   if $S \neq \emptyset$ **then**
    while $S \neq \emptyset$ **do**
     Select p from S; $S \leftarrow S \backslash \{p\}$
     $H \leftarrow H \cup \mathsf{NewCGSMain}(G_1, L_1 \cup \{p\}, L_2, \emptyset, 0, \succ, U)$
    end-while
   else
     $H \leftarrow \{(L_1, L_2, G_1)\}$
   end-if
end-if
$\mathsf{return}(H)$
end

Algorithm 4.6. $\mathsf{Transform}(F, D)$

Input F : a finite subset of $K[r, \bar{A}][\bar{X}]$,
  D: a polynomial in $K[\bar{A}]$

Output G: a finite subset of $K[\bar{A}][\bar{X}]$
begin
if $D = \emptyset$ **then** return(F) **end-if**
$G \leftarrow \emptyset$
while $F \neq \emptyset$ **do**
  Select f from F; $F \leftarrow F \backslash \{f\}$
  $N \leftarrow \deg_r(f)$
  $L \leftarrow \mathrm{Mono}(f)$; $H \leftarrow 0$
    while $L \neq \emptyset$ **do**
    Select p from L ; $L \leftarrow L\backslash\{p\}$; $N_1 \leftarrow N - \deg_r(p)$
    $p_1 \leftarrow$ Substitute $\frac{1}{\mathrm{lc}_{\bar{A}}(p)}$ for the variable r of p
    $q_1 \leftarrow p_1 \cdot D^{N_1}$; $H \leftarrow H + q_1$
    end-while
  if $H \neq 0$ **then** $G \leftarrow G \cup \{H\}$ **end-if**
end-while
return(G)
end

Remark : In (♣1) and (♣2), we applied the notation $\bigcup$ (union) for the algorithm. Since obviously $\mathbb{V}(h_1) \cup \mathbb{V}(h_2) = \mathbb{V}(\mathsf{LCM}(h_1, h_2))$ where $h_1, h_2 \in K[\bar{A}]$, we can apply the notation $\mathbb{V}(\mathsf{LCM}(s_1, \dots, s_l))$ instead of $\bigcup_{s \in L_2} \mathbb{V}(s)$ where $L_2 = \{s_1, \dots, s_l\}$. As we used the notation "∪ (union)" in Theorem 4.1, we followed Theorem 4.1 in the algorithm.
In Theorem 4.1 and Corollary 4.2, we need to transform a set F as follows;
(1) computing a Gröbner basis H for $\langle F \rangle$, (line 1)
(2) transforming H into H' by the new variable r, (line 7)
(3) computing a Gröbner basis G' for $\langle H' \rangle$, (line 1)
(4) transforming G' into G by Transform. (line 2)
If we do not use the natural number U in the algorithm, then by these transformations, we **rarely** obtain the infinite loop from 1 to 14 (recurrently) on a path of a tree structure. When we compute a Gröbner basis in $K[\bar{A}][\bar{X}]$, we apply the algorithm GröbnerBasisB. Since the algorithm GröbnerBasisB which uses a block order, regards parameters as variables, if we iterate the procedure from 1 to 14, then we rarely see that line 1 always outputs the same Gröbner basis. In order to avoiding this infinite loop, we introduced the natural number U. This is very technical step for always terminating the algorithm. We can apply many optimization techniques to obtain small and nice outputs comprehensive Gröbner systems (like the Suzuki-Sato algorithm [11]). Theoretically, like Algorithm 3.9, we do not need factorize in order to compute comprehensive Gröbner systems. However, since factorize is very effective as one of the optimization techniques to obtain small and nice outputs, we add the algorithm factorize to the algorithm. (We can also compute a radical ideal of $\langle L_1 \rangle$ to get nice outputs, however the computation is often expensive.)

Theorem 4.7. The algorithm $\mathsf{NEW}(F, U, \succ)$ terminates. The output forms a comprehensive Gröbner system for $\langle F \rangle$ on L^m.

Proof. First we show the termination. It suffices to show the termination of $\mathsf{NewCGSMain}(F, L_1, L_2, D, N, \succ, U)$. The key part is line 5 of Algorithm 4.5.

$(*1)$ If $E = \emptyset$ and $N \leq U$, then we have to consider lines 18–29 where is Suzuki-Sato's approach. In this case, the algorithm provides one segment (see line 19).
$(*2)$ If $E \neq \emptyset$ and $N \leq U$, then we have to consider lines 6–16. In this case, the algorithm does not provide any segment.

NewCGSMain is a recurrence algorithm and makes the tree structure. Take an arbitrary path of the tree structure. We prove that the algorithm executes lines 17–28 $(*1)$ and lines 6–15 $(*2)$ a finite number of times in the path. By the same reason of the proof of Suzuki-Sato [11], the algorithm executes $(*1)$ a finite number of times (see [11]). We need to prove that the algorithm executes $(*2)$ a finite number of times. As we said in the remark, if we don't have the number U and N, then the algorithm does not always terminate. However, the algorithm has U which is a finite number, and thus the algorithm executes $(*2)$ at most U times. Hence, this algorithm terminates. Next we have to show the correctness. This proof is almost same as the proof of the Suzuki-Sato algorithm. We remark that in this proof we need Theorem 4.1 and Corollary 4.2. In line 13 and 15, the algorithm compute the cases $t = \mathsf{LCM}(t_1, \dots, t_k) \neq 0$

and $t_1 = 0, \ldots, t_k = 0$, i.e, $\bigcup_{i=1}^{k} \mathbb{V}(t_i) \cup (L^m \setminus \mathbb{V}(t)) = L^m$. By this fact and the proof of Suzuki-Sato [11], the output of the algorithm covers the whole parameter space. $\square$

The algorithm NEW has been implemented in the computer algebra system Risa/Asir. In the following examples, we give outputs of the program. Note that in the program the natural number U of the algorithm NEW is fixed $U = 5$.

Example 4.8. Let $F = \{ax^2 + by^2, cx^2 + y^2, 2ax - 2cy\}$ be a subset of $\mathbb{Q}[a, b, c][x, y]$, a, b, c parameters and x, y variables. We have the lexicographic order $\succ$ such that $x \succ y$. Then, the program outputs a comprehensive Gröbner system for $\langle F \rangle$ w.r.t. $\succ$ as follows.

```
[a,b,c]==0,  [[1]]!=0,
[y^2].

[b^2+c,a-c*b,b*a+c^2]==0,  [[a]]!=0,
[a*x-c*y].

[a,c]==0,  [[b^2+c]]!=0,
[y^2].

[a,c*b]==0,  [[c],[b^2+c]]!=0,
[c*x^2,y].

[a-c*b]==0,  [[a],[b^2+c]]!=0,
[a*x-c*y,y^2].

[a]==0,  [[c],[a-c*b]]!=0,
[c*x^2,y].

[0]==0,  [[a],[a-c*b]]!=0,
[a*x-c*y,y^2].
```

This meaning is the following;

$$\begin{cases} \{y^2\}, & \text{if } \mathbb{V}(a, b, c), \\ \{ax - cy\}, & \text{if } \mathbb{V}(b^2 + c, a - cb, ba + c^2) \setminus \mathbb{V}(a), \\ \{y^2\}, & \text{if } \mathbb{V}(a, c) \setminus \mathbb{V}(b^2 + c), \\ \{cx^2, y\} & \text{if } \mathbb{V}(a, cb) \setminus (\mathbb{V}(c) \cup \mathbb{V}(b^2 + c)), \\ \{ax - cy, y^2\} & \text{if } \mathbb{V}(a - cb) \setminus (\mathbb{V}(a) \cup \mathbb{V}(b^2 + c)), \\ \{cx^2, y\} & \text{if } \mathbb{V}(a) \setminus (\mathbb{V}(c) \cup \mathbb{V}(a - cb)), \\ \{ax - cy, y^2\} & \text{if } \mathbb{C}^3 \setminus (\mathbb{V}(a) \cup \mathbb{V}(a - cb)). \end{cases}$$

This output has 7 segments. (Note that $\mathbb{V}(h_1) \cup \mathbb{V}(h_2) = \mathbb{V}(\mathsf{LCM}(h_1, h_2))$ where $h_1, h_2 \in K[\bar{A}]$.) By the way, the program of the Suzuki-Sato algorithm outputs 17 segments.

5 Benchmark tests and improvements

The algorithms Suzuki-Sato and NEW which contain several optimization techniques, have been implemented in Risa/Asir by the author. These programs are including in the package PGB [8]. In this section, we compare both programs Suzuki-Sato and NEW, and notice both problems. Moreover, in the second part of this section, we improve our algorithm NEW. Note that the natural number U of the algorithm NEW is fixed $U = 5$. (We used a PC [CPU: Pentium M 1.73 GHZ, Memory 512 MB RAM, OS: Windows XP].)

Let a, b, c, d be parameters, x, y, z, w variables and $\succ$ the lexicographic order such that $x \succ y \succ z \succ w$. We have the following subsets of $\mathbb{C}[a, b, c, d][x, y, z, w]$;
$F_1 = \{ax^4y + xy^2 + bx, x^3 + 2xy, bx^2 + x^2y\}$,
$F_2 = \{ax^2y^3 + by + y, x^2y^2 + xy + 2, ax^2 + by + 2\}$,
$F_3 = \{ax^4 + cx^2 + b, bx^3 + x^2 + 2, cx^2 + dx\}$,
$F_4 = \{ax^3y + cxy^2, x^4y + 3dy, cx^2 + bxy, x^2y^2 + ax^2, x^5 + y^5\}$.
The following table includes timing date of the programs in each problems.

Problem	Algorithm	Segments	time (sec.)
F_1	Suzuki-Sato	7	0.079
	NEW	4	0.031
F_2	Suzuki-Sato	4	0.047
	NEW	6	0.093
F_3	Suzuki-Sato	31	2.421
	NEW	22	2.203
F_4	Suzuki-Sato	39	1.391
	NEW	15	0.234

In the table above, we can see that our algorithm NEW runs faster than the algorithm Suzuki-Sato in the problems F_1, F_3, F_4. Furthermore, the numbers of segments are smaller than Suzuki-Sato's outputs. We remark that NEW does not always run faster than Suzuki-Sato. See the problem F_2. However, in many cases, NEW runs faster than Suzuki-Sato. Especially, if the number of parameters is greater than the number of variable, i.e., $|\bar{A}| > |\bar{X}|$, then NEW is much more efficient than Suzuki-Sato for computing comprehensive Gröbner systems. Next, we consider more difficult problems.
$F_5 = \{ax^2y + bx + y^3, ax^2y + bxy, y^2 + bx^2y + cxy\}$,
$F_6 = \{x^4 + ax^3 + bx^2 + cx + d, 4x^3 + 3ax^2 + 2bx + c\}$,
$F_7 = \{x^3 - a, y^4 - b, x + y - az\}$,
$F_8 = \{ax^2 + by, cw^2 + z, (x - z)^2 + (y - w)^2, 2dxw - 2by\}$.

Problem	Algorithm	Segments	time (sec.)
F_5	Suzuki-Sato	14	0.219
	NEW	6	0.109
F_6	Suzuki-Sato	875	92.88
	NEW	17	0.312
F_7	Suzuki-Sato	7	0.282
	NEW	$--$	> 30 m
F_8	Suzuki-Sato	$--$	> 30 m
	NEW	$--$	> 30 m

In the problems F_5 and F_6, NEW runs faster than Suzuki-Sato. Why does the program NEW run faster? Because NEW compute segments whose number is smaller than that of Suzuki-Sato's. Look at "**Segments**" of the table. In the problem F_7, NEW cannot return between 30 minutes. **Why?** Because we need much time for computing a Gröbner basis in $\mathbb{C}[r, a, b][x, y, z]$ (the algorithm GröbnerBasisB) where r is the new variable. In Algorithm 4.5 line 6, we have to make the new variable r. This is a dangerous step when we compute a Gröbner basis in polynomial rings. The problems of the algorithms Suzuki-Sato and NEW, are the following.

- Suzuki-Sato creates overmuch segments.
- NEW (sometimes) needs expensive Gröbner bases computations (however, the number of segments is not big).

Now we improve the algorithm NEW. Look at line 6 of Algorithm 4.5. In the line, we must select one polynomial. Actually, in the problem F_8, the program NEW selected a bad polynomial. Therefore, the program could not return. We should select a good polynomial from E for computing a comprehensive Gröbner system. In Algorithm 4.5, we define the following set as E

$$E := \{f \in G : \exists g \in G \backslash \{f\} \text{ s.t. } \operatorname{lpp}_{\bar{A}}(f) | \operatorname{lpp}_{\bar{A}}(g)\}.$$

In fact, in the problem F_8, the program NEW selects a polynomial f from E which has 12 monomials, i.e., the cardinality of $\mathrm{Mono}_{\{a,b\}}(f)$ is 12. Since this polynomial f is very big (and we multiply f by the new variable r), the Gröbner bases computation become very expensive. The author has computed a lot of comprehensive Gröbner systems by the program. By these computational experiments, the author was noticed that "**we should not select a big polynomial from E.**" That is, in concerning speed, we need to consider **how many monomials the selected polynomial has**. Now, in Algorithm 4.5, we can replace E to the following set

$$E_s := \{f \in G : \sharp(\mathrm{Mono}_{\bar{A}}(f)) \leq s, \exists g \in G \backslash \{f\} \\ \text{ s.t. } \operatorname{lpp}_{\bar{A}}(f) | \operatorname{lpp}_{\bar{A}}(g)\}$$

where $s \in \mathbb{N}$ and $\sharp(\mathrm{Mono}_{\bar{A}}(f))$ is the cardinality of the set $\mathrm{Mono}_{\bar{A}}(f)$. Clearly, we have $E_s \subseteq E$. In this case, we rename the algorithm NEW as NEW[s].

Problem	**Algorithm**	**Segments**	**time (sec.)**
F_6	NEW[1]	621	91.39
	NEW[2]	53	1.141
	NEW[3]	17	0.359
F_7	Suzuki-Sato	7	0.328
	NEW[1]	7	0.375
	NEW[2]	7	0.375
F_8	Suzuki-Sato	$--$	> 30 m
	NEW[1]	458	133.2

Remark: If we apply $s = 1$ for NEW[s], then we don't need the new variable r. However, like the problem F_6, sometimes the algorithm creates a lot of segments. In our algorithm, selecting a good polynomial from E (or E_s) is very important to compute a comprehensive Gröbner system, efficiently. The method of selecting a (good) polynomial from E (or choosing s) for computing comprehensive Gröbner systems efficiently, is a problem.

6 Comparisons

The anonymous referees suggested to compare our implementations[1] [8] with other implementations, namely the Maple-implementation[2] by Manubens and Montes [5] (Maple 9.5), and the Reduce-implementation[3] by Dolzmann and Sturm [2] (Reduce 3.8). All conditions and problems are the same as the previous section.

In this paper, we introduced a new algorithm for computing comprehensive Gröbner systems. In many cases, our algorithm is more efficient than the Suzuki-Sato algorithm. That is, our algorithm creates smaller outputs, and runs faster than the Suzuki-Sato algorithm. In general, if the number of parameters is greater than the number of variables, then the Suzuki-Sato algorithm runs slower than other existing algorithm. This is because the Suzuki-Sato algorithm creates overmuch segments, and regards parameters as variables in the computation. (Look at problems F_3 and F_6 which have only one variable.) However, in this case, our algorithm still works well. This is the main advantage of our algorithm compared to the Suzuki-Sato algorithm. If the number of parameters is smaller than the number of variables, then the Suzuki-Sato algorithm is very fast. In this case, as our algorithm is based on the Suzuki-Sato algorithm, it is clear that our algorithm is also very fast. See the list of benchmark tests [11].

[1] `http://www.risc.uni-linz.ac.at/people/knabeshi/pgb/` or `http://www.risc.uni-linz.ac.at/research/compalg/software/`

[2] `http://www-ma2.upc.edu/~montes/`

[3] `http://students.fim.uni-passau.de/~reduce/cgb/`

Pro.	Algorithm	System	Seg.	time (sec.)
F_1	NEW	Risa/Asir	4	0.031
	Suzuki-Sato	Risa/Asir	7	0.079
	Montes	Maple	4	2.703
	Weispfenning	Reduce	10	0.032
F_2	NEW	Risa/Asir	6	0.093
	Suzuki-Sato	Risa/Asir	4	0.047
	Montes	Maple	5	6.890
	Weispfenning	Reduce	15	0.344
F_3	NEW	Risa/Asir	22	2.203
	Suzuki-Sato	Risa/Asir	31	2.421
	Montes	Maple	14	13.422
	Weispfenning	Reduce	28	0.140
F_4	NEW	Risa/Asir	15	0.234
	Suzuki-Sato	Risa/Asir	39	1.391
	Montes	Maple	18	22.359
	Weispfenning	Reduce	21	0.940
F_5	NEW	Risa/Asir	6	0.109
	Suzuki-Sato	Risa/Asir	14	0.219
	Montes	Maple	9	78.610
	Weispfenning	Reduce	8	0.079
F_6	NEW	Risa/Asir	17	0.312
	Suzuki-Sato	Risa/Asir	875	92.88
	Montes	Maple	8	36.797
	Weispfenning	Reduce	6	0.110
F_7	NEW[3]	Risa/Asir	7	0.375
	Suzuki-Sato	Risa/Asir	7	0.328
	Montes	Maple	–	> 30m
	Weispfenning	Reduce	–	> 30m
F_8	NEW[1]	Risa/Asir	458	133.2
	Suzuki-Sato	Risa/Asir	–	> 30m
	Montes	Maple	–	> 30m
	Weispfenning	Reduce	–	> 30m

Acknowledgments

This work has been supported by the Austrian Science Foundation (FWF), project P16357-N04. The author thanks Prof. Franz Winkler for stimulating discussions and suggestions.

References

[1] Becker, T. On Gröbner bases under specialization. *Applicable Algebra in Engineering, Communication and Computing*, 5:1–8, 1994.

[2] Dolzmann, A. and Sturm, T. Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin*, 31(2):2–9, 1997.

[3] Gianni, P. Properties of Gröbner bases under specializations. In Davenport, J., editor, *EUROCAL'87*, pages 293–297. ACM Press, 1987.

[4] Kalkbrener, M. On the Stability of Gröbner Bases Under Specializations. *Journal of Symbolic Computation*, 24:51–58, 1997.

[5] Manubens, M. and Montes, A. Improving DISPGB algorithm using the discriminant ideal. *Journal of Symbolic Computation*, 41:1245–1263, 2006.

[6] Montes, A. A new algorithm for discussing Gröbner basis with parameters. *Journal of Symbolic Computation*, 33/1-2:183–208, 2002.

[7] Nabeshima, K. Reduced Gröbner bases in polynomial rings over a polynomial ring. In Wang, D. and Zheng, Z., editors, *International Conference on Mathematical Aspects of Computer and Information Sciences*, pages 15–32, 2006.

[8] Nabeshima, K. PGB: A Package for Computing Parametric Gröbner Bases and Related Objects. 2007. preprint.

[9] Noro, M. and Takeshima, T. Risa/Asir- A Computer Algebra System. In Wang, P., editor, *International Symposium on Symbolic and Algebraic Computation*, pages 387–396. ACM-Press, 1992. `http://www.math.kobe-u.ac.jp/Asir/asir.html`.

[10] Suzuki, A. and Sato, Y. An alternative approach to Comprehensive Gröbner bases. *Journal of Symbolic Computation*, 36/3-4:649–667, 2003.

[11] Suzuki, A. and Sato, Y. A Simple Algorithm to compute Comprehensive Gröbner Bases using Gröbner bases. In Dumas, J-G., editor, *International Symposium on Symbolic and Algebraic Computation*, pages 326–331. ACM Press, 2006.

[12] Weispfenning, V. Comprehensive Gröbner bases. *Journal of Symbolic Computation*, 14/1:1–29, 1992.

[13] Weispfenning, V. Canonical Comprehensive Gröbner bases. In Mora, T., editor, *International Symposium on Symbolic and Algebraic Computation*, pages 270–278. ACM Press, 2002.