Journal of Symbolic Computation 52 (2013) 124–142

Contents lists available at SciVerse ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

An efficient method for computing comprehensive Gröbner bases ☆,☆☆

Deepak Kapur [a], Yao Sun [b,c], Dingkang Wang [c]

[a] *Dept. of Computer Science, University of New Mexico, Albuquerque, NM, USA*
[b] *SKLOIS, Institute of Information Engineering, CAS, Beijing 100093, China*
[c] *KLMM, Academy of Mathematics and Systems Science, CAS, Beijing 100190, China*

ARTICLE INFO

Article history:
Received 20 October 2011
Accepted 3 May 2012
Available online 6 August 2012

Keywords:
Gröbner basis
Comprehensive Gröbner basis
Comprehensive Gröbner system
Stability condition

ABSTRACT

A new approach is proposed for computing a comprehensive Gröbner basis of a parameterized polynomial system. The key new idea is not to simplify a polynomial under various specialization of its parameters, but rather keep track in the polynomial, of the power products whose coefficients vanish; this is achieved by partitioning the polynomial into two parts—*nonzero* part and *zero* part for the specialization under consideration. During the computation of a comprehensive Gröbner system, for a particular branch corresponding to a specialization of parameter values, nonzero parts of the polynomials dictate the computation, i.e., computing S-polynomials as well as for simplifying a polynomial with respect to other polynomials; but the manipulations on the whole polynomials (including their zero parts) are also performed. Once a comprehensive Gröbner system is generated, both nonzero and zero parts of the polynomials are collected from every branch and the result is a *faithful* comprehensive Gröbner basis, to mean that every polynomial in a comprehensive Gröbner basis belongs to the ideal of the original parameterized polynomial system. This technique should be applicable to all algorithms for computing a comprehensive Gröbner system, thus producing both a comprehensive Gröbner system as well as a faithful comprehensive Gröbner basis of a parameterized polynomial system simultaneously. To propose specific algorithms for computing comprehensive Gröbner bases, a more generalized theorem is presented to give a more generalized stable condition for parametric polynomial systems. Combined

☆ This paper is an expanded version of the paper entitled "Computing comprehensive Gröbner systems and comprehensive Gröbner bases simultaneously" that is presented at ISSAC'2011 (Kapur et al., 2011).

☆☆ The first author is supported by the National Science Foundation award CCF-0729097 and the last two authors are supported by NKBRPC 2011CB302400, NSFC 10971217, 60970152 and 61121062.

E-mail addresses: kapur@cs.unm.edu (D. Kapur), sunyao@iie.ac.cn (Y. Sun), dwang@mmrc.iss.ac.cn (D.K. Wang).

0747-7171/$ – see front matter
http://dx.doi.org/10.1016/j.jsc.2012.05.015

with the new approach, the new theorem leads to two efficient algorithms for computing comprehensive Gröbner bases. The timings on a collection of examples demonstrate that both these two new algorithms for computing comprehensive Gröbner bases have better performance than other existing algorithms.

1. Introduction

The concept of a comprehensive Gröbner basis was introduced by Weispfenning (1992) as a special basis of a parametric polynomial system such that for every possible specialization of its parameters, the basis obtained from the comprehensive Gröbner basis serves as a Gröbner basis of the ideal generated by the specialization of the parametric polynomial system (see also Kapur, 1995, where a related concept of a parametric Gröbner basis is introduced).

Generally, a faithful comprehensive Gröbner basis for a given polynomial set F is harder to compute than a comprehensive Gröbner system of F. The difficulty of computing a faithful comprehensive Gröbner basis of F is that all the polynomials in this comprehensive Gröbner basis should be faithful polynomials, i.e., these polynomials should belong to the ideal $\langle F\rangle$, while the polynomials in a comprehensive Gröbner system of F are not necessarily faithful polynomials. Therefore, the algorithms for computing comprehensive Gröbner systems usually have better performance than those for comprehensive Gröbner bases. Consequently, a mainstream approach for computing a comprehensive Gröbner basis is to compute a faithful comprehensive Gröbner system, i.e. all the polynomials appearing in this comprehensive Gröbner system are faithful polynomials. In Weispfenning (1992), Weispfenning gave a method of preserving all polynomials faithful during computations of a comprehensive Gröbner system. However, Weispfenning's method, which colors different parts of the polynomials appearing in the computation, is complicated and not efficient.

We present in this paper, an efficient method to keep track of faithful polynomials during the computations, such that a comprehensive Gröbner basis of a parametric polynomial system can be constructed more efficiently. The key idea is to split a polynomial into two parts—*nonzero* part and *zero* part for the specialization under consideration. The proposed idea can be used in all algorithms for computing comprehensive Gröbner systems, including Weispfenning's (Weispfenning, 1992), Kapur's (Kapur, 1995), Montes' (Montes, 2002; Manubens and Montes, 2006, 2009; Montes and Wibmer, 2010), Wang's (Chen et al., 2005), Suzuki–Sato's (Suzuki and Sato, 2006), Nabeshima's (Nabeshima, 2007) as well as our recently proposed algorithm (Kapur et al., 2010).

To illustrate the key idea, let us consider Example 8.4 from Weispfenning (2003) where Weispfenning defined the concept of a canonical comprehensive Gröbner basis of a parametric polynomial system to mimic the concept of a reduced Gröbner basis of a polynomial system determined by the associated ideal and term order. Suppose there are two polynomials $f, g \in k[u, v][x, y]$:

$$f = y + ux + v, \qquad g = uy + x + v.$$

Further, suppose we are interested in computing Gröbner basis with the lexicographic order induced by $y \succ x$.

Clearly, f can be used to simplify g, resulting in

$$h = g - uf = (1 - u^2)x - uv + v.$$

In fact, g can be deleted without any loss of generality. Based on the specialization of u and v, the leading power product of h is either x or 1.

For the branch where $1 - u^2 \neq 0$, the nonzero part of h is $(1 - u^2)x + (-uv + v)$. Since both f and h have noncomparable leading power products, $\{f, h\}$ constitutes a Gröbner basis for this branch for those specializations satisfying $1 - u^2 \neq 0$.

For the branch, where $1 - u^2 = 0$ and $-uv + v \neq 0$ for all those specializations of u and v, the nonzero part of h is $-uv + v$ and the zero part of h is $(1 - u^2)x$. For this branch, a Gröbner basis is $\{h\}$, since the leading power product of the nonzero part of h is 1, which reduces every other power

product. If h is simplified using the specializations of u and v, the Gröbner basis would have been $\{1\}$. However, such a Gröbner basis is not *faithful*, since 1 is not in $\langle f, g\rangle$. But to maintain the faithfulness, we keep h instead.

Finally, for the branch where $1-u^2=0$ and $-uv+v=0$, h vanishes completely. And, the nonzero part of f is itself, since the leading coefficient of f is 1. A Gröbner basis for this branch is $\{f\}$; if the specialization of u and v had been used to simplify f, we have $\{y+x+v\}$ as a Gröbner basis.

Using the proposed algorithm, a comprehensive Gröbner system consists of three branches: a branch corresponding to specializations satisfying $1-u^2\neq 0$, for which $\{f, h\}$ is a Gröbner basis for a 0-dimensional specialization; another branch, corresponding to the specialization satisfying $1-u^2=0$, $-uv+v\neq 0$ (which can be further simplified to $u+1=0$, $v\neq 0$), for which $\{h\}$ is a Gröbner basis for the ideal generated by 1; the last branch corresponds to the specialization $1-u^2=0$ and $-uv+v=0$, for which $\{f\}$ is a Gröbner basis for the one-dimensional ideal.

The key difference between the output of this algorithm and other algorithms including our algorithm in Kapur et al. (2010), is that a Gröbner basis in every branch in a comprehensive Gröbner system is a subset of the original ideal, and hence contributes to a comprehensive Gröbner basis.

A faithful comprehensive Gröbner basis for the above system can be easily constructed by taking the union of Gröbner bases along all the branches; for every possible specialization, there is exactly one branch generating a Gröbner basis for the specialized ideal; furthermore, by construction, all the polynomials are in the ideal of the original system. For the above example, a comprehensive Gröbner basis is $\{f, h\}$.[1]

Based on the ideas illustrated for the above example, we propose in this paper, an efficient method of computing a faithful comprehensive Gröbner basis. The key idea is that, during computations of a comprehensive Gröbner system, the zero part of a polynomial under a specialization is also kept in a tuple representation so as to recover the original polynomial when needed. Specifically, when computing a comprehensive Gröbner system of the set $F\subset k[U][X]$, we use a tuple $(q,\bar{q})\in (k[U][X])^2$ to replace each polynomial $p=q+\bar{q}$ in the computation, with the following properties: (i) $p\in\langle F\rangle$, and (ii) $\bar{q}$ is 0 under the specialization of parameters being considered. When a comprehensive Gröbner system of F is obtained, then for each tuple $(g,\bar{g})$ in this comprehensive Gröbner system, we recover the faithful polynomial $g+\bar{g}$; this way, a comprehensive Gröbner basis of F is obtained simultaneously with the comprehensive Gröbner system.

An important feature of this new method is that it can be implemented in any algorithm for computing comprehensive Gröbner systems to obtain faithful comprehensive Gröbner bases simultaneously. Consequently, the algorithm can exploit the efficiency of existing Gröbner bases algorithms. In particular, it uses the efficient algorithm proposed in Kapur et al. (2010) for computing a comprehensive Gröbner system which has the following properties:

- Branches (segments) generated are disjoint and constitute a partition of the parameter space.
- Consistency of parametric constraints is tested, so empty segments along branches are not generated.
- The leading coefficients of polynomials in each branch are not zero under the specializations associated with the branch.

To illustrate how this approach works, we first propose a more generalized stable condition for parametric polynomial systems, and then present two efficient algorithms for computing comprehensive Gröbner bases based on the new stable condition. This new proposed stable condition is a substantially improved version of Kalkbrener's stable condition (Kalkbrener, 1997), and it completes the theory of computing comprehensive Gröbner systems that are presented in Kapur et al. (2010).

Comprehensive Gröbner basis and Gröbner system constructions have been found useful in many engineering applications which can be modeled using parameterized polynomial systems; see Donald et al. (1992), Gao et al. (2003), Montes (2002) for examples of some applications. These constructions

[1] An interested reader would notice that this result is different from the one reported in Weispfenning (2003). In fact, the canonical comprehensive Gröbner basis reported there for the same order is a proper superset of the above result, suggesting that after all, the definition in Weispfenning (2003) does not quite capture the notion of minimality and hence, canonicity.

have also been found useful for automated geometry theorem proving (Chen et al., 2005) and automated geometry theorem discovery (Montes and Recio, 2007), as well as more recently, for computing loop invariants in program analysis (Kapur, 2006). Solving parametric polynomial systems has also been investigated by Chou and Gao (1992) and Chen et al. (2007) using the characteristic set construction, as well as by Wibmer (2007) using Gröbner cover.

This paper is an expanded version of the paper entitled "Computing comprehensive Gröbner systems and comprehensive Gröbner bases simultaneously" that is presented at ISSAC'2011 (Kapur et al., 2011). The paper is organized as follows. We give some notations and definitions in Section 2. The new technique mentioned above is described in Section 3. We propose a new stable condition for parametric polynomial systems and two new algorithms for computing comprehensive Gröbner bases in Section 4. A simple example illustrates one of the proposed algorithms in Section 5. Empirical data and comparison with other existing algorithms are presented in Section 6. Concluding remarks follow in Section 7.

2. Notations and definitions

Let k be a field, R be the polynomial ring $k[U]$ in the parameters $U = \{u_1, \ldots, u_m\}$, and $R[X]$ be the polynomial ring over the parameter ring R in the variables $X = \{x_1, \ldots, x_n\}$ where $X \cap U = \emptyset$, i.e. X and U are disjoint sets.

Let $PP(X)$, $PP(U)$ and $PP(U, X)$ be the sets of power products of X, U and $X \cup U$ respectively. The order $\prec_{X,U}$ is an admissible block term order on $PP(U, X)$ where $U \ll X$. The orders $\prec_X$ and $\prec_U$ are the restrictions of $\prec_{X,U}$ on $PP(X)$ and $PP(U)$ respectively.

For a polynomial $f \in R[X] = k[U][X]$, the leading power product, leading coefficient and leading monomial of f w.r.t. the order $\prec_X$ are denoted by $\mathrm{lpp}_X(f)$, $\mathrm{lc}_X(f)$ and $\mathrm{lm}_X(f)$ respectively. Note that $\mathrm{lpp}_X(f) \in PP(X)$. Since f can also be regarded as an element of $k[U, X]$, in this case, the leading power product, leading coefficient and leading monomial of f w.r.t. the order $\prec_{X,U}$ are denoted by $\mathrm{lpp}_{X,U}(f)$, $\mathrm{lc}_{X,U}(f)$ and $\mathrm{lm}_{X,U}(f)$ respectively. For f, we always have $\mathrm{lm}_X(f) = \mathrm{lc}_X(f)\,\mathrm{lpp}_X(f)$ and $\mathrm{lm}_{X,U}(f) = \mathrm{lc}_{X,U}(f)\,\mathrm{lpp}_{X,U}(f)$.

Given a field L, a specialization of R is a homomorphism $\sigma : R \longrightarrow L$. In this paper, we always assume L to be an algebraically closed field containing k and we only consider the specializations induced by the elements in L^m. That is, for $\bar{a} \in L^m$, the induced specialization $\sigma_{\bar{a}}$ is defined as follows:

$$\sigma_{\bar{a}} : f \longrightarrow f(\bar{a}),$$

where $f \in R$. Every specialization $\sigma : R \longrightarrow L$ extends canonically to a specialization $\sigma : R[X] \longrightarrow L[X]$ by applying σ coefficient-wise.

For a parametric polynomial system, the comprehensive Gröbner system and comprehensive Gröbner basis are defined below.

Definition 2.1 *(CGS).* Let F be a subset of $R[X]$, $A_1, \ldots, A_l$ be algebraically constructible subsets of L^m, $G_1, \ldots, G_l$ be subsets of $R[X]$, and S be a subset of L^m such that $S \subseteq A_1 \cup \cdots \cup A_l$. A finite set $\mathcal{G} = \{(A_1, G_1), \ldots, (A_l, G_l)\}$ is called a **comprehensive Gröbner system** on S for F, if $\sigma_{\bar{a}}(G_i)$ is a Gröbner basis for the ideal $\langle \sigma_{\bar{a}}(F) \rangle$ in $L[X]$ for any $\bar{a} \in A_i$ and $i = 1, \ldots, l$. Each (A_i, G_i) is called a branch of $\mathcal{G}$. If $S = L^m$, then $\mathcal{G}$ is simply called a comprehensive Gröbner system for F.

A comprehensive Gröbner system $\mathcal{G} = \{(A_1, G_1), \ldots, (A_l, G_l)\}$ for F is called **faithful**, if in addition, every element of G_i is also in $\langle F \rangle$.

For a set $F \subset R = k[U]$, the variety defined by F in L^m is denoted by $V(F)$. In this paper, the constructible set A_i always has the form: $A_i = V(E_i) \setminus V(N_i)$, where E_i, N_i are subsets of $k[U]$. Particularly, we call E_i and N_i equality constraints and disequality constraints respectively. Clearly, if the set $A_i = V(E_i) \setminus V(N_i)$ is empty, the branch (A_i, G_i) is redundant.

Definition 2.2 *(CGB).* Let F be a subset of $R[X]$ and S be a subset of L^m. A finite subset G in $R[X]$ is called a **comprehensive Gröbner basis** on S for F, if $\sigma_{\bar{a}}(G)$ is a Gröbner basis for the ideal $\langle \sigma_{\bar{a}}(F) \rangle$ in $L[X]$ for any $\bar{a} \in S$. If $S = L^m$, then G is simply called a comprehensive Gröbner basis for F.

A comprehensive Gröbner basis G for F is called **faithful**, if in addition, every element of G is also in $\langle F\rangle$.

A typical approach to compute a comprehensive Gröbner basis on S for F is to compute a *faithful* comprehensive Gröbner system $\mathcal{G} = \{(A_1, G_1), \ldots, (A_l, G_l)\}$ on S for $F \subset R[X]$ first. Since $\mathcal{G}$ is faithful, we have $G_i \subset \langle F\rangle$ for $i = 1, \ldots, l$. Then the set $G_1 \cup \cdots \cup G_l$ is a comprehensive Gröbner basis on S for F. However, almost all known algorithms for computing comprehensive Gröbner systems output non-faithful comprehensive Gröbner systems, since polynomials get simplified based on parameter specialization. So the main challenge for computing a comprehensive Gröbner basis is to retrieve the terms that are simplified by parameter specializations. In the next section, we propose a new approach for this purpose.

3. A polynomial as a tuple under parameter specialization

As mentioned in the introduction and illustrated using an example, the key new idea in our approach is to keep track of polynomials in $\langle F\rangle$ while computing various Gröbner bases under different parameter specializations. If some terms in these polynomials vanish due to specialization of parameters during the computation of a comprehensive Gröbner system, this information can be kept by splitting the polynomial into the *nonzero* part and the *zero* part under the specialization.

A polynomial $p \in \langle F\rangle$ is replaced by a tuple $(q, \bar{q})$ along a branch of a comprehensive Gröbner system computation for a specialization of parameters from a constructible set A_i, such that

(i) $p = q + \bar{q}$, and further,
(ii) $\sigma(\bar{q})$ is 0 for every parameter specialization σ from A_i.

We call $(q, \bar{q})$ an **admissible tuple representation** of p in the ideal $\langle F\rangle$ w.r.t. A_i.

For an admissible tuple representation $(q, \bar{q})$ of p, the polynomial $\bar{q}$ corresponds to the *zero* part of p, because $\bar{q}$ is 0 under the specializations from A_i. The polynomial q will appear in the branches of comprehensive Gröbner systems, and it is supposed to be the *nonzero* part of p, although we do not include the condition "$\sigma(\mathrm{lc}_X(q)) \neq 0$ for any specialization from A_i" in the above definition. The additional condition is excluded because this property is not preserved under addition. However, as the reader would observe that, all existing algorithms for computing comprehensive Gröbner systems aim to make $\mathrm{lc}_X(q)$ nonzero under the specializations from A_i.

Since general polynomials are replaced by admissible tuple representations, we next show how to manipulate admissible tuple representations in practical computations.

3.1. Basic operations

Let us see the addition and multiplication of admissible tuple representation of polynomials from an ideal. Given admissible tuple representations $(p, \bar{p})$ and $(q, \bar{q})$ of $p + \bar{p}$ and $q + \bar{q}$ in $\langle F\rangle$, w.r.t. A_i, the sum of $(p, \bar{p})$ and $(q, \bar{q})$ is $(p + q, \bar{p} + \bar{q})$, which is also an admissible tuple representation of $p + q + \bar{p} + \bar{q}$ in the ideal $\langle F\rangle$ w.r.t. A_i. Furthermore, given a polynomial $r \in R[X]$, the product of $(p, \bar{p})$ and r is $(r \cdot p, r \cdot \bar{p})$, also an admissible tuple representation of $r \cdot p + r \cdot \bar{p}$ in the ideal $\langle F\rangle$ w.r.t. A_i.

Next, let us now consider other operations on admissible tuple representations while computing a comprehensive Gröbner system. In Gröbner basis computations, there are two crucial steps: S-polynomial construction from a pair of distinct polynomials and simplification of a polynomial by another polynomial. In addition, we also need to simplify polynomials according to equality and disequality constraints on parameters.

For a constructible set A_i and two admissible tuple representations $\mathbf{p} = (p, \bar{p})$, $\mathbf{q} = (q, \bar{q})$ of $p + \bar{p}$ and $q + \bar{q}$ in an ideal $\langle F\rangle$ w.r.t. A_i, respectively, assuming both $\mathrm{lc}_X(p)$ and $\mathrm{lc}_X(q)$ are *nonzero* w.r.t. A_i, their **S-polynomial** is defined to be

$$\mathrm{lc}_X(q)t_p \cdot \mathbf{p} - \mathrm{lc}_X(p)t_q \cdot \mathbf{q} = \big(\mathrm{lc}_X(q)t_p p - \mathrm{lc}_X(p)t_q q, \mathrm{lc}_X(q)t_p \bar{p} - \mathrm{lc}_X(p)t_q \bar{q}\big),$$

where $t_p = \frac{\mathrm{lcm}(\mathrm{lpp}_X(p),\mathrm{lpp}_X(q))}{\mathrm{lpp}_X(p)}$ and $t_q = \frac{\mathrm{lcm}(\mathrm{lpp}_X(p),\mathrm{lpp}_X(q))}{\mathrm{lpp}_X(q)}$. Clearly, the S-polynomial of $\mathbf{p}$ and $\mathbf{q}$ is also an admissible tuple representation of the polynomial $\mathrm{lc}_X(q)t_p(p+\bar{p}) - \mathrm{lc}_X(p)t_q(q+\bar{q})$ in $\langle F\rangle$ w.r.t. A_i.

Similarly, along a branch corresponding to a constructible set A_i, assuming $\mathrm{lpp}_X(g)$ divides $\mathrm{lpp}_X(f)$ and $\mathrm{lc}_X(g)$ is *nonzero* w.r.t. A_i, the result of reducing (simplifying) $\mathbf{f} = (f, \bar{f})$ by $\mathbf{g} = (g, \bar{g})$ is

$$\mathrm{lc}_X(g)\cdot\mathbf{f} - \mathrm{lc}_X(f)t\cdot\mathbf{g} = \big(\mathrm{lc}_X(g)f - \mathrm{lc}_X(f)tg, \mathrm{lc}_X(g)\bar{f} - \mathrm{lc}_X(f)t\bar{g}\big),$$

where $t = \frac{\mathrm{lpp}_X(f)}{\mathrm{lpp}_X(g)}$. The result is an admissible tuple representation of the simplified polynomial in the ideal of $\langle F\rangle$.

Next, let us consider simplifying an admissible tuple representation by the equality and disequality constraints. Let $\mathbf{p} = (p, \bar{p})$ be an admissible tuple representation of $p + \bar{p}$ in $\langle F\rangle$ w.r.t. $A_i = V(E_i) \setminus V(N_i)$, with $E_i, N_i \subset k[U]$. We next simplify $\mathbf{p}$ by A_i in the following way. If $\mathrm{lc}_X(p)$ is *zero* w.r.t. A_i, then the tuple $(p, \bar{p})$ is simplified to $(q, \bar{q})$ by moving all terms in p that vanish into $\bar{q}$ such that $p + \bar{p} = q + \bar{q}$ and the leading coefficient of q is not always zero for the specializations from A_i and $\bar{q}$ is 0 w.r.t. A_i. Note that $(q, \bar{q})$ is admissible tuple representation of $q + \bar{q} = p + \bar{p}$ in $\langle F\rangle$ w.r.t. A_i.

For an admissible tuple representation $(p, \bar{p})$ of the polynomial $p + \bar{p}$ in $\langle F\rangle$ w.r.t. A_i, we usually require $\mathrm{lc}_X(p)$ to be *nonzero* w.r.t. A_i during computations along a branch for A_i. This condition can be achieved by expanding the constructible set A_i, and this method has been used in many algorithms including Kapur (1995), Montes (2002), Chen et al. (2005). For example, let $h = \mathrm{lc}_X(p) \in k[U]$, and h is not always zero under the specializations from $A_i = V(E_i) \setminus V(N_i)$, where $E_i, N_i \subset k[U]$. In order to make $\mathrm{lc}_X(p)$ *nonzero*, we set $A_i' = V(E_i) \setminus V(h \times N_i)$, where $h \times N_i = \{hn \mid n \in N_i\}$. Then $\mathrm{lc}_X(p)$ will be nonzero w.r.t. A_i'. In case $h = 0$, we can continue to consider the tuple $(p, \bar{p})$ w.r.t. the constructible set $A_i'' = V(E_i \cup \{h\}) \setminus V(N_i)$: that is, simplify $(p, \bar{p})$ to $(q, \bar{q})$ by A_i'' first, and then repeat the above discussions if $\mathrm{lc}_X(q)$ is not always zero w.r.t. A_i''.

In algorithms for computing a comprehensive Gröbner system from F, if we use the above admissible tuple representation of polynomials in F and perform the above S-polynomial and reduction as defined above on tuples, then for each branch, we get a finite set of admissible tuples such that their first components constitute a Gröbner basis of F under the parameter specialization belonging to A_i. Furthermore, these constructions produce tuples such that the polynomials corresponding to them, obtained by adding the two components of the tuple, are in the ideal $\langle F\rangle$. In this way, a faithful Gröbner basis is generated for every branch corresponding to A_i.

The operations in this section are only basic operations for admissible tuple representations. To manipulate tuples more efficient in practical implementations, we next introduce another two kinds of operations: module operations and polynomial operations.

3.2. Module operations

Another way to manipulate admissible tuple representations of polynomials is to consider tuples as elements in a module and use corresponding module operations. Below we discuss this; for terminologies on "module" computations, an interested reader can refer to Chapter 5 of Cox et al. (2005).

Let F be a subset of $R[X]$ and A_i be a constructible set. Then

$$\mathbf{M}(F, A_i) = \big\{(p, \bar{p}) \;\big|\; p + \bar{p} \in \langle F\rangle \text{ and } \sigma_{\bar{a}}(\bar{p}) = 0 \text{ for all } \bar{a} \in A_i\big\}$$

is the set of all admissible tuple representations of polynomials from $\langle F\rangle$ w.r.t. A_i. Clearly, $\mathbf{M}(F, A_i) \subset (R[X])^2$ is an $R[X]$-module with the following operations:

1. for $\mathbf{p} = (p, \bar{p}), \mathbf{q} = (q, \bar{q}) \in \mathbf{M}(F, A_i)$, $\mathbf{p} + \mathbf{q} := (p + q, \bar{p} + \bar{q}) \in \mathbf{M}(F, A_i)$, and
2. for $\mathbf{p} = (p, \bar{p}) \in \mathbf{M}(F, A_i)$ and $r \in R[X]$, $r \cdot \mathbf{p} := (r \cdot p, r \cdot \bar{p}) \in \mathbf{M}(F, A_i)$.

Since $\mathbf{M}(F, A_i)$ is a module, we can use general definitions of the *S-polynomial* and *reduction* in a module. To make these definitions consistent with those defined on tuples in the last subsection, it suffices to extend the term order defined on $R[X]$ to the free $R[X]$-module $(R[X])^2$ in a POT (position over term) fashion with $(1, 0) \succ (0, 1)$. More precisely, let $\mathbf{e}_1 = (1, 0)$, $\mathbf{e}_2 = (0, 1)$; then for $i, j = 1, 2$,

$$x^\alpha \mathbf{e}_i \succ x^\beta \mathbf{e}_j \quad \text{iff} \quad \begin{cases} i < j, \\ \text{or} \\ i = j \text{ and } x^\alpha \succ x^\beta. \end{cases}$$

An important operation for computing a comprehensive Gröbner system is simplifying $(p, \bar{p}) \in \mathbf{M}(F, A_i)$ w.r.t. A_i. As mentioned earlier, we can simplify $(p, \bar{p})$ to $(q, \bar{q})$ by moving all terms in p that vanish into $\bar{q}$ such that $p + \bar{p} = q + \bar{q}$ and the leading coefficient of q is not always zero for the specializations from A_i and $\bar{q}$ is 0 w.r.t. A_i. This simplification can also be expressed using module operations. Assume $A_i = V(E_i) \setminus V(N_i)$ where $E_i, N_i \subset R$ and $\langle E_i \rangle$ is radical. Then simplifying $(p, \bar{p})$ w.r.t. A_i is equivalent to reducing $(p, \bar{p})$ by the set $\{(e, -e) \mid e \in E_i\} \subset \mathbf{M}(F, A_i)$. For example, let $F = \{ax^2 + bx + a + 1\} \subset \mathbb{Q}[a, b][x]$, $A_i = V(E_i) = V(\{a, b-1\})$ and $\mathbf{p} = (ax^2 + bx + a + 1, 0) \in \mathbf{M}(F, A_i)$. Then $\mathbf{p} = (ax^2 + bx + a + 1, 0)$ can be reduced to $(x + 1, ax^2 + bx - x + a)$ as follows:

$$\left(ax^2 + bx + a + 1, 0\right) - \left(x^2 + 1\right) \cdot (a, -a) - x \cdot (b - 1, 1 - b) = \left(x + 1, ax^2 + bx - x + a\right).$$

Notice that the result is also an element in $\mathbf{M}(F, A_i)$, since $(a, -a), (b - 1, 1 - b) \in \mathbf{M}(F, A_i)$.

3.3. Polynomial operations

Another way to represent a tuple of polynomials is by a single polynomial using an extra variable such that this extra variable separates the two parts of the tuple. All tuple operations can then be implemented efficiently on the corresponding polynomials. We think that this might be the motivation behind the trick used in Suzuki and Sato (2006).

The key idea is that for an admissible tuple representation $(p, \bar{p})$ of $p + \bar{p}$ in $\langle F \rangle$ w.r.t. a constructible set A_i, we use the polynomial $py + \bar{p}$ in the polynomial ring $R[X, y]$ to represent $(p, \bar{p})$, where y is a new variable different from parameters U and unknowns X. In this case, the set

$$M(F, A_i) = \left\{ py + \bar{p} \;\middle|\; p + \bar{p} \in \langle F \rangle \text{ and } \sigma_{\bar{a}}(\bar{p}) = 0 \text{ for all } \bar{a} \in A_i \right\}$$

is isomorphic to the set of all admissible tuple representations of polynomials from $\langle F \rangle$ w.r.t. A_i.

$M(F, A_i)$ is still an $R[X]$-module and is not substantially different from the module defined in the last subsection. A polynomial $py + \bar{p}$ from $M(F, A_i)$ corresponds to an admissible tuple representation of the polynomial $p + \bar{p}$ in $\langle F \rangle$ w.r.t. A_i.

Operations on the elements in $M(F, A_i)$ can be done by the following polynomial operations:

1. for $f = py + \bar{p}, g = qy + \bar{q} \in M(F, A_i)$, $f + g := (p + q)y + \bar{p} + \bar{q}$, and
2. for $f = py + \bar{p} \in M(F, A_i)$ and $r \in R[X]$, $rf := rpy + r\bar{p}$.

The S-polynomial of f and g in $M(F, A_i)$ is defined to be spoly(f, g), where spoly(f, g) is the general S-polynomial defined in a polynomial ring. The reduction of elements in $M(F, A_i)$ is no different from the general reduction defined in a polynomial ring. To make the definitions of S-polynomial and reduction consistent with the corresponding definitions of admissible tuple representations, we require the term order on $R[X, y]$ to be a block order with $X \ll y$.

Simplification of a polynomial $py + \bar{p}$ w.r.t. $A_i = V(E_i) \setminus V(N_i)$, where $E_i, N_i \subset R$, is similar as the module case. That is, simplifying $py + \bar{p}$ w.r.t. A_i is equivalent to reducing $py + \bar{p}$ by the set $\{ey - e \mid e \in E_i\}$.

Note that since we require $X \ll y$, the polynomials, whose degrees are greater than 2 in y, will *not* be generated by the above operations.

3.4. Duplication of manipulations done on the first components

As the reader might have noticed, it suffices to perform various Gröbner basis operations only on the first component of the tuple representation of a polynomial from the input ideal to generate a comprehensive Gröbner system. However, to compute a comprehensive Gröbner basis consisting of faithful polynomials from the input ideal, the same operations have to be done on the second component also, even though computations on the second components do not affect the overall computation of a Gröbner basis along a branch under a specialization.

In this way, another method of handling admissible tuple representations appears. That is, recording the manipulations done on the first components first, and then act these manipulations to the

second components afterward. Specifically, let $(p_1, \bar{p}_1), \ldots, (p_s, \bar{p}_s)$ be several admissible tuple representations of polynomials in $\langle F\rangle$ w.r.t. A_i, and p be a polynomial computed from $p_1, \ldots, p_s$, i.e. there exist $q_1, \ldots, q_s$ in $R[X]$, such that: $p = q_1p_1 + \cdots + q_sp_s$. Then the tuple $(p, q_1\bar{p}_1 + \cdots + q_s\bar{p}_s)$ is an admissible tuple representation of some polynomial in $\langle F\rangle$ w.r.t. A_i.

In specific algorithms for computing comprehensive Gröbner systems for the polynomials $\{p_1, \ldots, p_s\} \subset R[X]$, the Gröbner basis for the ideal generated by $\{p_1, \ldots, p_s\}$ is usually needed. So a Gröbner basis implementation that also provides information about how the elements of a Gröbner basis can be obtained from the input basis (i.e., the representation of each element of a Gröbner basis in terms of the input basis), can be used to derive the required information about the second components; hence, in this way, the faithful polynomial corresponding to the first component can be generated.

General methods of getting information about Gröbner basis elements in terms of the input basis are very expensive. However, using the results from Sun and Wang (2011) as well as Sun et al. (2012), if one uses the F5 algorithm or other signature-based algorithms to compute a Gröbner basis, then the desired information can be constructed from the outputs of these algorithms within polynomial time.

4. New algorithms for computing comprehensive Gröbner bases

The algorithm proposed in Kapur et al. (2010) is an efficient algorithm for computing comprehensive Gröbner systems. In this section, we first present a stability condition for parametric polynomial systems. We use this stability condition later to compute comprehensive Gröbner bases.

Theorem 4.1. *Let G be a Gröbner basis for an ideal $\langle F\rangle \subset R[X]$ w.r.t. an admissible order $\prec$ in X and σ be a specialization from R to L. Let G_m be a subset of G such that $\sigma(\mathrm{lc}_X(g)) \neq 0$ for any $g \in G_m$. Let G_{redund} be the set $\{g \in G \setminus G_m \mid$ there exists $g' \in G_m$ such that $\mathrm{lpp}_X(g')$ divides $\mathrm{lpp}_X(g)\}$, and $G_0 = G \setminus (G_m \cup G_{redund})$. Then the following two conditions are equivalent.*

(1) *$\sigma(G_m) = \{\sigma(g) \mid g \in G_m\}$ is a Gröbner basis for $\langle\sigma(F)\rangle$ in $L[X]$ w.r.t. $\prec$.*
(2) *For every $g \in G_0$, the polynomial $\sigma(g)$ is reducible to 0 modulo $\sigma(G_m)$.*

It should be noted that the above G_m does not necessarily contain *all* the g's that satisfies $\sigma(\mathrm{lc}_X(g)) \neq 0$ in G.

Proof. The proof for (1) $\Rightarrow$ (2) is trivial, since $\sigma(g) \in \langle\sigma(F)\rangle$ for any $g \in G_0$. Next, we prove (2) $\Rightarrow$ (1).

We first **claim** $\sigma(\mathrm{lc}_X(g)) = 0$ for every $g \in G_0$. If $\sigma(\mathrm{lc}_X(g)) \neq 0$, then by (2), there exists $g' \in G_m$ such that $\mathrm{lpp}_X(g')$ divides $\mathrm{lpp}_X(g)$. This means g must be in the set G_m or G_{redund} by definition, which is a contradiction with $g \in G_0 = G \setminus (G_m \cup G_{redund})$.

Next, to prove $\sigma(G_m)$ is a Gröbner basis for $\langle\sigma(F)\rangle$, it suffices to show that for any $f \in \langle F\rangle$, there exists $g \in G_m$ such that $\mathrm{lpp}_X(\sigma(g))$ divides $\mathrm{lpp}_X(\sigma(f))$. Similarly to the proof for Theorem 3.1 in Kalkbrener (1997), we do the proof by induction on $\prec$.

Induction basis: Consider the case $\mathrm{lpp}_X(f) = 1$. Since G is a Gröbner basis for $\langle F\rangle$, there must exist some $g \in G$ such that $\mathrm{lpp}_X(g)$ divides $\mathrm{lpp}_X(f) = 1$, i.e. $\mathrm{lpp}_X(g) = 1$.

If there exists $g \in G_m \cup G_{redund}$ such that $\mathrm{lpp}_X(g) = 1$, then there must exist $g' \in G_m$ such that $\mathrm{lpp}_X(g') = 1$ by the definition of G_{redund}. In this case, $\mathrm{lpp}_X(\sigma(g'))$ divides $\mathrm{lpp}_X(\sigma(f))$ no matter $f = 0$ or not.

Otherwise, the set $H = \{g \in G \mid \mathrm{lpp}_X(g) = 1\}$ is a subset of G_0. Assume $H = \{h_1, \ldots, h_s\}$. Then there exist $c_1, \ldots, c_s \in R$ such that $\mathrm{lc}_X(f) = c_1\mathrm{lc}_X(h_1) + \cdots + c_s\mathrm{lc}_X(h_s)$. Since $\mathrm{lpp}_X(f) = 1$, we have $\mathrm{lc}_X(f) = f$. Then we must have $\sigma(f) = 0$, because $\sigma(\mathrm{lc}_X(h_i)) = 0$ by the above claim.

Induction step: Consider the case $\mathrm{lpp}_X(f) \succ 1$. We assume for any $f' \in \langle F\rangle$ with $\mathrm{lpp}_X(f') \prec \mathrm{lpp}_X(f)$, there always exists $g \in G_m$ such that $\mathrm{lpp}_X(\sigma(g))$ divides $\mathrm{lpp}_X(\sigma(f'))$.

If there exists $g \in G_m \cup G_{redund}$ such that $\mathrm{lpp}_X(g)$ divides $\mathrm{lpp}_X(f)$, then there must exist $g' \in G_m$ such that $\mathrm{lpp}_X(g')$ divides $\mathrm{lpp}_X(f)$ by the definition of G_{redund}. Next we discuss two cases depending on whether $\sigma(\mathrm{lc}_X(f))$ is 0 or not.

1. If $\sigma(\mathrm{lc}_X(f)) \neq 0$, then $\mathrm{lpp}_X(\sigma(f)) = \mathrm{lpp}_X(f)$, and hence, $\mathrm{lpp}_X(\sigma(g'))$ divides $\mathrm{lpp}_X(\sigma(f))$.
2. If $\sigma(\mathrm{lc}_X(f)) = 0$, then the polynomial

$$f' = \mathrm{lc}_X(g')f - \mathrm{lc}_X(f)\big(\mathrm{lpp}_X(f)/\mathrm{lpp}_X(g')\big)g'$$

is in $\langle F\rangle$. Moreover, we have $\mathrm{lpp}_X(f') \prec \mathrm{lpp}_X(f)$ and $\sigma(f') = \sigma(\mathrm{lc}_X(g'))\sigma(f)$. By induction assumption, there exists $g'' \in G_m$ such that $\mathrm{lpp}_X(\sigma(g''))$ divides $\mathrm{lpp}_X(\sigma(f')) = \mathrm{lpp}_X(\sigma(f))$.

Otherwise, the set $H = \{g \in G \mid \mathrm{lpp}_X(g) \text{ divides } \mathrm{lpp}_X(f)\}$ is a subset of G_0. Assume $H = \{h_1, \ldots, h_s\}$. Then there exist $c_1, \ldots, c_s \in R$ such that $\mathrm{lc}_X(f) = c_1\mathrm{lc}_X(h_1) + \cdots + c_s\mathrm{lc}_X(h_s)$. Note that the claim indicates $\sigma(\mathrm{lc}_X(h_i)) = 0$, and hence $\sigma(\mathrm{lc}_X(f)) = 0$. By (2), for each $h_i \in H$, the polynomial $\sigma(h_i)$ is reducible to 0 modulo $\sigma(G_m)$. Similar to the proof of Theorem 3.1 in Kalkbrener (1997), there exist an $\bar{h}_i \in \langle F\rangle$ and a $b_i \in R \setminus \ker(\sigma)$ with $\sigma(b_i)\sigma(h_i) = \sigma(\bar{h}_i)$ and $\mathrm{lpp}_X(h_i) \succ \mathrm{lpp}_X(\sigma(h_i)) = \mathrm{lpp}_X(\bar{h}_i)$. Consider the polynomial

$$f' = bf - (b/b_1)c_1\big(\mathrm{lpp}_X(f)/\mathrm{lpp}_X(h_1)\big)(b_1h_1 - \bar{h}_1) - \cdots - (b/b_s)c_s\big(\mathrm{lpp}_X(f)/\mathrm{lpp}_X(h_s)\big)(b_sh_s - \bar{h}_s),$$

where $b = b_1b_2\cdots b_s$. Obviously, we have $f' \in \langle F\rangle$ and $\mathrm{lpp}_X(f') \prec \mathrm{lpp}_X(f)$. Note that $\sigma(b_ih_i - \bar{h}_i) = 0$ and $\sigma(b) \neq 0$. So we also have $\sigma(f') = \sigma(b)\sigma(f)$. By induction assumption, there exists $g'' \in G_m$ such that $\mathrm{lpp}_X(\sigma(g''))$ divides $\mathrm{lpp}_X(\sigma(f')) = \mathrm{lpp}_X(\sigma(f))$. □

The reader would notice that in the statement of Theorem 4.1, G_m is not required to be minimal, i.e., for any two distinct $g_1, g_2 \in G_m$, $\mathrm{lpp}_X(g_1)$ could be a multiple of $\mathrm{lpp}_X(g_2)$, since we want to establish below Theorem 3.1 in Kalkbrener (1997) immediately follow from the above theorem. A set G_m in Theorem 4.1 can be selected to require that for any two distinct $g_1, g_2 \in G_m$, neither $\mathrm{lpp}_X(g_1)$ is a multiple of $\mathrm{lpp}_X(g_2)$ nor $\mathrm{lpp}_X(g_2)$ is a multiple of $\mathrm{lpp}_X(g_1)$ in order to obtain a minimal Gröbner basis for $\langle\sigma(F)\rangle$. Further, in Theorem 4.1, we have that $\sigma(\mathrm{lc}_X(g)) \neq 0$ for any $g \in G_m$, and $\sigma(\mathrm{lc}_X(g)) = 0$ for any $g \in G_0$, which is proved as a claim in the proof of Theorem 4.1. However, for $g \in G_{redund}$, there is no condition on $\sigma(\mathrm{lc}_X(g))$ being 0 or not.

Corollary 4.2. *(See Kalkbrener, 1997.) Let $G = \{g_1, \ldots, g_s\}$ be a Gröbner basis for the ideal $\langle F\rangle \subset R[X]$ w.r.t. an admissible order $\prec$ in X, and σ be a specialization from R to L. We assume that the g_i's are ordered in such a way that there exists an $r \in \{0, \ldots, s\}$ with $\sigma(\mathrm{lc}_X(g_i)) \neq 0$ for $i \in \{1, \ldots, r\}$ and $\sigma(\mathrm{lc}_X(g_i)) = 0$ for $i \in \{r+1, \ldots, s\}$. Then the following two conditions are equivalent.*

(1) *$\{\sigma(g_1), \ldots, \sigma(g_r)\}$ is a Gröbner basis for $\langle\sigma(F)\rangle$ w.r.t. $\prec$.*
(2) *For every $i \in \{r+1, \ldots, s\}$, the polynomial $\sigma(g_i)$ is reducible to 0 modulo $\{\sigma(g_1), \ldots, \sigma(g_r)\}$.*

Proof. The proof for (1) ⇒ (2) is trivial. To prove (2) ⇒ (1), let $G_m = \{g_1, \ldots, g_r\}$, $G_{redund} = \{g \in G \setminus G_m \mid \text{there exists } g' \in G_m \text{ such that } \mathrm{lpp}_X(g') \text{ divides } \mathrm{lpp}_X(g)\}$, and $G_0 = G \setminus (G_m \cup G_{redund})$. Clearly, the set G_0 is a subset of $\{g_{r+1}, \ldots, g_s\}$. So by (2), for any $g \in G_0$, the polynomial $\sigma(g)$ is reducible to 0 modulo $\sigma(G_m) = \{\sigma(g_1), \ldots, \sigma(g_r)\}$. Theorem 4.1 shows $\sigma(G_m)$ is a Gröbner basis for $\langle\sigma(F)\rangle$. □

The above proof shows that Theorem 4.1 is a strict generalization of Kalkbrener's theorem (Corollary 4.2). The set G_{redund} is often not the empty set. Hence, the set G_m in Theorem 4.1 is usually a proper subset of the set $\{g_1, \ldots, g_r\}$ in Corollary 4.2. Moreover, the set G_0 in Theorem 4.1 is also usually a proper subset of $\{g_{r+1}, \ldots, g_s\}$ in Corollary 4.2, which means fewer polynomials need to be checked in Theorem 4.1.

We define below the concept of a minimal Dickson basis of a set of polynomials consisting only of those polynomials whose leading power products cannot be simplified by any polynomial in the set. We then prove two other corollaries of Theorem 4.1, which are useful for developing algorithms for comprehensive Gröbner bases in the next subsections.

Definition 4.3 *(Minimal Dickson basis).* Let $\prec$ be an admissible order in X. For a polynomial set G in $R[X]$, we say $F \subset R[X]$, denoted by MDBasis(G), is a minimal Dickson basis of G, if

1. F is a subset of G,
2. for every polynomial $g \in G$, there is some polynomial $f \in F$ such that $\mathrm{lpp}_X(g)$ is a multiple of $\mathrm{lpp}_X(f)$, i.e. $\langle \mathrm{lpp}_X(F)\rangle = \langle \mathrm{lpp}_X(G)\rangle$, and
3. for any two distinct $f_1, f_2 \in F$, neither $\mathrm{lpp}_X(f_1)$ is a multiple of $\mathrm{lpp}_X(f_2)$ nor $\mathrm{lpp}_X(f_2)$ is a multiple of $\mathrm{lpp}_X(f_1)$.

A minimal Dickson basis of a set may not be unique.

Corollary 4.4. *Let G be a Gröbner basis for the ideal $\langle F\rangle \subset R[X]$ w.r.t. an admissible order $\prec$ in X. Let G_0 be any subset of G, $G_r \subset R$ be the set of all the coefficients of G_0, and $G_m = \mathrm{MDBasis}(G \setminus G_0)$. If σ is a specialization from R to L such that*

1. *$\sigma(g) = 0$ for $g \in G_r$, and*
2. *$\sigma(h) \neq 0$, where $h = \prod_{g\in G_m} \mathrm{lc}_X(g) \in R$,*

then $\sigma(G_m)$ is a (minimal) Gröbner basis of $\langle \sigma(F)\rangle$ in $L[X]$ w.r.t. $\prec_X$.

Proof. Note that $\sigma(g) = 0$ for any $g \in G_0$, since all the coefficients of g are in G_r. Let $G_{redund} = G \setminus (G_m \cup G_0)$. The corollary holds by Theorem 4.1. □

By setting G_0 be the set $G \cap R$, the above corollary becomes Theorem 4.3 in Kapur et al. (2010).

Corollary 4.5. *(See Kapur, Sun and Wang, 2010.) Let G be a Gröbner basis for the ideal $\langle F\rangle \subset k[U, X]$ w.r.t. an admissible block order with $U \ll X$. Let $G_r = G \cap k[U]$ and $G_m = \mathrm{MDBasis}(G \setminus G_r)$. If σ is a specialization from $k[U]$ to L such that*

1. *$\sigma(g) = 0$ for $g \in G_r$, and*
2. *$\sigma(h) \neq 0$, where $h = \prod_{g\in G_m} \mathrm{lc}_X(g) \in k[U]$,*

then $\sigma(G_m)$ is a (minimal) Gröbner basis of $\langle \sigma(F)\rangle$ in $L[X]$ w.r.t. $\prec_X$.

Compared with Corollary 4.5 which is one of the main results in Kapur et al. (2010), Corollary 4.4 gives us more flexibilities for choosing the set G_0 to construct specializations. For example, let $F = \{ax - b, by - a, cx^2 - y, cy^2 - x\}$ be a subset of $\mathbb{Q}[a, b, c][x, y]$. As discussed in Section 5, $G = \{x^3 - y^3, cx^2 - y, ay^2 - bc, cy^2 - x, ax - b, bx - acy, a^2y - b^2c, by - a, a^6 - b^6, a^3c - b^3, b^3c - a^3, ac^2 - a, bc^2 - b\}$ is a Gröbner basis for the ideal generated by F over the ring $\mathbb{Q}[a, b, c, x, y]$ w.r.t. the block order $\prec_{X,U}$ with $\{a, b, c\} \ll \{x, y\}$, and within each block, $\prec_X$ and $\prec_U$ are graded reverse lexicographic orders with $y < x$ and $c < b < a$, respectively. G is also a Gröbner basis for the ideal $\langle F\rangle$ over the ring $\mathbb{Q}[a, b, c][x, y]$ w.r.t. $\prec_X$.

Using Corollary 4.4, we have (at least) the following *three* ways of constructing specializations, and at the same time, obtaining Gröbner bases for the ideal $\langle F\rangle$ after the specializations.

1. Let $G_0 = G \cap \mathbb{Q}[a, b, c]$ thus implying that the chosen specialization must make all polynomials in G_0 to be 0. Let $G_m = \{bx - acy, by - a\}$ be a minimal Dickson basis of $G \setminus G_0$. So both Corollaries 4.4 and 4.5 indicate that, the set $\sigma(G_m)$ is a Gröbner basis of $\langle \sigma(F)\rangle$ for any specialization deduced by points in $V(G_0) \setminus V(b)$.
2. We choose $G_0 = \{ay^2 - bc, ax - b, bx - acy, a^2y - b^2c, by - a, a^6 - b^6, a^3c - b^3, b^3c - a^3, ac^2 - a, bc^2 - b\}$ and a specialization which makes all these polynomials to be 0. Let $G_m = \{cx^2 - y, cy^2 - x\}$ be a minimal Dickson basis of $G \setminus G_0$. According to Corollary 4.4, the set $\sigma(G_m)$ is a Gröbner basis of $\langle \sigma(F)\rangle$ for any specialization deduced from $V(a, b) \setminus V(c)$.

3. We choose $G_0 = \emptyset$. Let $G_m = \{bc^2 - b\}$ be a minimal Dickson basis of G. Then Corollary 4.4 also shows $\sigma(G_m)$ is a Gröbner basis of $\langle F\rangle$ for any specialization deduced from $\mathbb{C}^3 \setminus V(bc^2 - b)$.

Using Corollary 4.4, we present two algorithms for computing comprehensive Gröbner bases using tuple representations of polynomials. The first algorithm uses module operations for computations, whereas the second algorithm uses the trick of introducing a new variable to represent a tuple of polynomials using a single polynomial.

4.1. Algorithm using module operations

The theorem below serves as a basis of the algorithm for computing a comprehensive Gröbner basis. The set E below refers to the set of equality constraints, and it is usually the empty set at the beginning.

Theorem 4.6. *Let F be a set of polynomials in $k[U, X]$, E be a subset of $k[U]$, and $\mathbf{M}$ be a $k[U, X]$-module generated by $\{(f, 0) \mid f \in F\} \cup \{(e, -e) \mid e \in E\}$. Suppose $\mathbf{G}$ is a Gröbner basis for the module $\mathbf{M}$ w.r.t. an order extended from $\prec_{X,U}$ in a position over term fashion with $(0, 1) \prec (1, 0)$, where $\prec_{X,U}$ is an admissible block order with $U \ll X$.*

Denote $G^{1st} = \{g \mid (g, \bar{g}) \in \mathbf{G}\}$, $G_r = G^{1st} \cap k[U]$ and $G_m = \mathrm{MDBasis}(G^{1st} \setminus G_r)$. $\mathbf{G}_m$ is a subset of $\mathbf{G}$ such that $\{(g, \bar{g}) \in \mathbf{G}_m \mid g \in G_m\}$. If σ is a specialization from $k[U]$ to L such that

1. *$\sigma(g) = 0$ for $g \in G_r$, and*
2. *$\sigma(h) \neq 0$, where $h = \prod_{g \in G_m} \mathrm{lc}_X(g) \in k[U]$,*

then

(1) for each $(g, \bar{g}) \in \mathbf{G}_m$, $g + \bar{g} \in \langle F\rangle$ and $\sigma(\bar{g}) = 0$, and
(2) $\{\sigma(g + \bar{g}) \mid (g, \bar{g}) \in \mathbf{G}_m\}$ is a Gröbner basis of $\langle\sigma(F)\rangle$ in $L[X]$ w.r.t. $\prec_X$.

That is, $\{(V(G_r) \setminus V(h), G_m)\}$ is comprehensive Gröbner system on $V(G_r) \setminus V(h)$ for F, and $\{g + \bar{g} \mid (g, \bar{g}) \in \mathbf{G}_m\}$ is a comprehensive Gröbner basis on $V(G_r) \setminus V(h)$ for F.

Proof. For (1), we first show $E \subset \langle G_r\rangle$. Since $\mathbf{G}$ is a Gröbner basis of the module $\mathbf{M}$ generated by $\{(f, 0) \mid f \in F\} \cup \{(e, -e) \mid e \in E\}$ w.r.t. an order extended from $\prec_{X,U}$ in a POT fashion with $(0, 1) \prec (1, 0)$, we next show G^{1st} is a Gröbner basis for the ideal $\langle F \cup E\rangle$ w.r.t. $\prec_{X,U}$. For any $h \in \langle F \cup E\rangle$, we have $h = \sum_{f \in F} p_f f + \sum_{e \in E} q_e e$ where $p_f, q_e \in k[U, X]$, so $(h, -(\sum_{e \in E} q_e e)) = \sum_{f \in F} p_f (f, 0) + \sum_{e \in E} q_e (e, -e) \in \mathbf{M}$. As $\mathbf{G}$ is a Gröbner basis for $\mathbf{M}$, there exists $(g, \bar{g}) \in \mathbf{G}$ such that $\mathrm{lpp}_X(g)$ divides $\mathrm{lpp}_X(h)$, which means G^{1st} is a Gröbner basis for the ideal $\langle F \cup E\rangle$. Besides, $G_r = G^{1st} \cap k[U] \subset \langle F \cup E\rangle$, so we have $E \subset \langle G_r\rangle \subset k[U]$ since $\prec_{X,U}$ is a block order with $U \ll X$.

Notice that $\mathbf{G}_m$ is a subset of the module $\mathbf{M}$; for each $(g, \bar{g}) \in \mathbf{G}_m$, we have

$$\begin{pmatrix} g \\ \bar{g} \end{pmatrix} = \sum_{f \in F} p_f \begin{pmatrix} f \\ 0 \end{pmatrix} + \sum_{e \in E} q_e \begin{pmatrix} e \\ -e \end{pmatrix},$$

where $p_f, q_e \in k[U, X]$. So $g + \bar{g} = (\sum_{f \in F} p_f f + \sum_{e \in E} q_e e) + \sum_{e \in E} q_e (-e) = \sum_{f \in F} p_f f \in \langle F\rangle$, and $\bar{g} = \sum_{e \in E} q_e (-e)$. Since $E \subset \langle G_r\rangle$, then $\sigma(\bar{g}) = 0$.

For (2), G^{1st} is a Gröbner basis for the ideal $\langle F \cup E\rangle$ w.r.t. $\prec_{X,U}$ as shown above, $G_r = G^{1st} \cap k[U]$ and $G_m = \mathrm{MDBasis}(G^{1st} \setminus G_r)$, so $\sigma(G_m) = \{\sigma(g + \bar{g}) \mid (g, \bar{g}) \in \mathbf{G}_m\}$ is a minimal Gröbner basis of $\langle\sigma(F)\rangle$ by Corollary 4.4. □

The above theorem does not require us to compute a whole Gröbner basis for the module $\mathbf{M}$. Instead, it suffices to compute $\mathbf{G} \subset \mathbf{M}$ such that $G^{1st} = \{g \mid (g, \bar{g}) \in \mathbf{G}\}$ is a Gröbner basis for the ideal $\langle F \cup E\rangle$.

The above theorem gives a direct way to compute a comprehensive Gröbner system and a comprehensive Gröbner basis simultaneously for $F \subset k[U, X]$ on $V(E) \setminus V(N)$. First, we compute a Gröbner basis $\mathbf{G}$ for the module generated by $\{(f, 0) \mid f \in F\} \cup \{(e, -e) \mid e \in E\}$ w.r.t. an order extended from $\prec_{X,U}$ in a position over term fashion with $(0, 1) \prec (1, 0)$, where $\prec_{X,U}$ is an admissible block order with $U \ll X$.

Theorem 4.6 shows that $\{(V(G_r) \setminus (V(h) \cup V(N)), G_m)\}$ is comprehensive Gröbner system on $V(G_r) \setminus (V(h) \cup V(N))$ for F, and $\{g + \bar{g} \mid (g, \bar{g}) \in \mathbf{G}_m\}$ is a comprehensive Gröbner basis on $V(G_r) \setminus (V(h) \cup V(N))$ for F, where $h = \prod_{g \in G_m} \mathrm{lc}_X(g) \in k[U]$.

The ideals $\langle G_r \rangle$ and $\langle E \rangle$ may not be identical in general, so we need to consider the difference of these two ideals. That is, if $V(E) \setminus (V(G_r) \cup V(N))$ is not empty, then for any specialization σ from $k[U]$ to L deduced by a point in $V(E) \setminus (V(G_r) \cup V(N))$, we must have $\langle \sigma(F) \rangle = \langle 1 \rangle$. The set G_r is a comprehensive Gröbner basis on $V(E) \setminus (V(G_r) \cup V(N))$ for F.

The constructible set $V(E) \setminus V(N)$ has been divided into disjoint three parts:

$$V(E) \setminus V(N) = \big(V(E) \setminus \big(V(G_r) \cup V(N)\big)\big) \cup \big(V(G_r) \setminus V(h)\big) \cup \big(V(G_r, h) \setminus V(N)\big).$$

We can set $E' = G_r \cup \{h\}$ and use Theorem 4.6 to recursively compute a comprehensive Gröbner system and a comprehensive Gröbner basis on $V(E') \setminus V(N)$ for F. Finally, collecting all results computed in the above three steps, we can get a comprehensive Gröbner system as well as a comprehensive Gröbner basis simultaneously for F on $V(E) \setminus V(N)$.

We give below a detailed algorithm for computing a comprehensive Gröbner basis on $V(E) \setminus V(N)$ for $F \subset k[U, X]$. The correctness of the new algorithm is a direct result of the above theorem. The core of the algorithm below is an efficient algorithm for computing a comprehensive Gröbner system proposed in Kapur et al. (2010). Its termination can be proved in a same way as in Kapur et al. (2010). Proposition 4.7 below is a proof of termination the algorithm.

In order to keep the presentation simple, we have deliberately avoided to include tricks and optimizations such as factoring h in the description below. All the tricks suggested in Kapur et al. (2010) can be used here as well. In fact, our implementation incorporates fully these optimizations.

Algorithm CGB-Module(E, N, F)
Input: (E, N, F): E, N, finite subsets of $k[U]$; F, a finite subset of $k[U, X]$.
Output: a comprehensive Gröbner basis of the set F on $V(E) \setminus V(N)$.

1. $\mathcal{CGS}$:= CGSMainMod(E, N, F), where $\mathcal{CGS}$ is a finite set of 3-tuples $(E_i, N_i, \mathbf{G}_i)$ such that $\{(V(E_i) \setminus V(N_i), G_i^{1st})\}$, where $G_i^{1st} = \{g \mid (g, \bar{g}) \in \mathbf{G}_i\}$, constitutes a comprehensive Gröbner system on $V(E) \setminus V(N)$ for F, and for each $(g, \bar{g}) \in \mathbf{G}_i$, $g + \bar{g} \in \langle F \rangle$ and $\sigma(\bar{g})$ is 0 for every parameter specialization σ from $V(E_i) \setminus V(N_i)$.
2. Return $\{g + \bar{g} \mid (g, \bar{g}) \in \mathbf{G}_i \text{ for all } i\}$.

Below we assume that all Gröbner basis computations are done in $(k[U, X])^2$ using the order extended by $\prec_{X,U}$ in a POT fashion with $(1, 0) \succ (0, 1)$.

Algorithm CGSMainMod(E, N, F)
Input: (E, N, F): E, N, finite subsets of $k[U]$; F, a finite subset of $k[U, X]$.
Output: $\mathcal{CGS}$: a finite set of 3-tuples $(E_i, N_i, \mathbf{G}_i)$ such that $\{(V(E_i) \setminus V(N_i), G_i^{1st})\}$, where $G_i^{1st} = \{g \mid (g, \bar{g}) \in \mathbf{G}_i\}$, constitutes a comprehensive Gröbner system on $V(E) \setminus V(N)$ for F, and for each $(g, \bar{g}) \in \mathbf{G}_i$, $g + \bar{g} \in \langle F \rangle$ and $\sigma(\bar{g})$ is 0 for every parameter specialization σ from $V(E_i) \setminus V(N_i)$.

1. If inconsistent(E, N), then return $\emptyset$.
2. Otherwise, $\mathbf{G}_0$:= ReducedGröbnerBasis$(\{(f, 0) \mid f \in F\} \cup \{(e, -e) \mid e \in E\})$.
3. $\mathbf{G} := \mathbf{G}_0 \setminus \{(g, \bar{g}) \in \mathbf{G}_0 \mid g = 0\}$ and $G^{1st} := \{g \mid (g, \bar{g}) \in \mathbf{G}\}$.
4. If there exists $(g, \bar{g}) \in \mathbf{G}$ such that $g = 1$, then return $\{(E, N, \{(g, \bar{g})\})\}$.
5. Let $\mathbf{G}_r := \{(g, \bar{g}) \in \mathbf{G} \mid g \in k[U]\}$ and $G_r := \{g \mid (g, \bar{g}) \in \mathbf{G}_r\}$.
6. If inconsistent$(E, G_r \times N)$, then $\mathcal{CGS} := \emptyset$, else $\mathcal{CGS} := \{(E, G_r \times N, \mathbf{G}_r)\}$.
7. If inconsistent(G_r, N), then return $\mathcal{CGS}$.

8. Otherwise, let $G_m := \text{MDBasis}(G^{1st} \setminus G_r)$ and $\mathbf{G}_m := \{(g, \bar{g}) \in \mathbf{G} \setminus \mathbf{G}_r \mid g \in G_m\}$.
9. If consistent$(G_r, N \times \{h\})$, then $\mathcal{CGS} := \mathcal{CGS} \cup \{(G_r, N \times \{h\}, \mathbf{G}_m)\}$, where $h = \text{lcm}\{h_1, \ldots, h_k\}$ and $\{h_1, \ldots, h_k\} = \{\text{lc}_X(g) \mid g \in G_m\}$.
10. Return $\mathcal{CGS} \cup \bigcup_{h_i \in [h_1, \ldots, h_k]} \text{CGSMainMod}(G_r \cup \{h_i\}, N \times \{h_1 h_2 \cdots h_{i-1}\}, \{g + \bar{g} \mid (g, \bar{g}) \in \mathbf{G} \setminus \mathbf{G}_r\})$.

In the above algorithm, $A \times B = \{fg \mid f \in A,\ g \in B\}$. Also, for the case $i = 1$, $N \times \{h_1 h_2 \cdots h_{i-1}\} = N$, and inconsistent$(E, N)$ returns true if $V(E) \setminus V(N)$ is empty, false otherwise. In Kapur et al. (2010), we have discussed various heuristics for performing the inconsistency check (inconsistent(E, N)). Besides, please note that in each recursive call of the function CGSMainMod(E, N, F), the constructible set $S = V(E) \setminus V(N)$ is partitioned into the following three disjoint parts:

$$S = S_1 \cup S_2 \cup S_3,$$

where $S_1 = V(E) \setminus (V(G_r) \cup V(N))$, $S_2 = V(G_r) \setminus (V(N) \cup V(\text{lcm}(h_1, \ldots, h_k)))$, and $S_3 = \bigcup_i V(G_r, h_i) \setminus (V(N) \cup V(\text{lcm}(h_1, \ldots, h_{i-1})))$.

Proposition 4.7. *The algorithm CGB-Module terminates after finitely many steps.*

Proof. We use König's Lemma to prove the termination. It suffices to show that (1) in each recursive call of the algorithm CGSMainMod, only finitely many branches are created, and (2) along each branch, the algorithm terminates after finitely many steps.

For (1), by algorithm CGSMainMod, at step 10, since the number of polynomials in G_m is finite, only finitely many branches are created. For (2), since $h_i = \text{lc}_X(g) \in k[U]$ for some $g \in G^{1st}$, G^{1st} is a *reduced* Gröbner basis for $\langle F \cup E \rangle \subset k[U, X]$, and $G_r = G^{1st} \cap k[U]$, then $h_i \notin \langle G_r \rangle \subset k[U]$, as otherwise, the polynomial $g \in G^{1st}$ can be simplified further by G_r. In the next recursive call of CGSMainMod, the ideal generated by the input set $E' = G_r \cup \{h_i\}$ is strictly larger than $\langle E \rangle$ from the previous call of CGSMainMod. So each branch must terminate after finitely many steps. □

4.2. Algorithm using polynomial operations

We first give the theorem on which the algorithm in this subsection is based. Recall that the set E also refers to the set of equality constraints.

Theorem 4.8. *Let F be a set of polynomials in $k[U, X]$, E be a subset of $k[U]$, and I be an ideal over $k[U, X, y]$ generated by $\{fy \mid f \in F\} \cup \{ey - e \mid e \in E\}$. Suppose G is a Gröbner basis for the ideal I w.r.t. an admissible block order with $U \ll X \ll y$.*

Denote $G^{1st} = \{g \mid gy + \bar{g} \in G\}$, $G_r = (G^{1st} \cap k[U]) \cup E$ and $G_m = \text{MDBasis}(G^{1st} \setminus G_r)$. $G_{m,y}$ is a subset of G such that $\{gy + \bar{g} \in G \mid g \in G_m\}$. If σ is a specialization from $k[U]$ to L such that

1. *$\sigma(g) = 0$ for $g \in G_r$, and*
2. *$\sigma(h) \neq 0$, where $h = \prod_{g \in G_m} \text{lc}_X(g) \in k[U]$,*

then

(1) *for each $gy + \bar{g} \in G$, $g + \bar{g} \in \langle F \rangle$ and $\sigma(\bar{g}) = 0$, and*
(2) *$\{\sigma(g + \bar{g}) \mid gy + \bar{g} \in G_{m,y}\}$ is a Gröbner basis of $\langle \sigma(F) \rangle$ in $L[X]$ w.r.t. $\prec_X$.*

That is, $\{(V(G_r) \setminus V(h), G_m)\}$ is comprehensive Gröbner system on $V(G_r) \setminus V(h)$ for F, and $\{g + \bar{g} \mid gy + \bar{g} \in G_{m,y}\}$ is a comprehensive Gröbner basis on $V(G_r) \setminus V(h)$ for F.

Proof. According to the block order with $U \ll X \ll y$, it is easy to see that every polynomial in G has at most degree 1 in y.

For (1), since each $gy + \bar{g} \in G$ is in the ideal generated by $\{fy \mid f \in F\} \cup \{ey - e \mid e \in E\}$, we have $gy + \bar{g} = \sum_{f \in F} p_f(fy) + \sum_{e \in E} q_e(ey - e)$ where $p_f, q_e \in k[U, X]$. Setting $y = 1$ in the equa-

tion, we then have $g+\bar{g}=\sum_{f\in F} p_f f$, which means $g+\bar{g}\in\langle F\rangle$. When setting $y=0$, we get $\bar{g}=-\sum_{e\in E} q_e e\in\langle E\rangle$. Since $E\subset G_r$ as defined, we have $\sigma(\bar{g})=0$.

For (2), let $G_{U,X}=G\cap k[U,X]$. We first show that $G^{1st}\subset\langle F\cup E\rangle$ and $G^{1st}\cup G_{U,X}$ is a Gröbner basis for $\langle F\cup E\rangle$. From (1), for any $gy+\bar{g}\in G^{1st}$, we have $g+\bar{g}\in\langle F\rangle$ and $\bar{g}\in\langle E\rangle$. So for each $gy+\bar{g}\in G^{1st}$, the relation $g\in\langle F\cup E\rangle$ holds directly, and then $G^{1st}\subset\langle F\cup E\rangle$. Next, for any $h\in\langle F\cup E\rangle$, we have $h=\sum_{f\in F} p_f f+\sum_{e\in E} q_e e$ where $p_f,q_e\in k[U,X]$, so the polynomial $hy-(\sum_{e\in E} q_e e)=\sum_{f\in F} p_f(fy)+\sum_{e\in E} q_e(ey-e)$ is in the ideal generated by $\{fy\mid f\in F\}\cup\{ey-e\mid e\in E\}$. Note that G is a Gröbner basis for this ideal by hypothesis. Then there exists $h_0\in G$ such that $\mathrm{lpp}_X(h_0)$ divides $\mathrm{lpp}(hy-(\sum_{e\in E} q_e e))=\mathrm{lpp}_X(h)y$. Assume $h_0=gy+\bar{g}$. If $g\neq 0$, then we have $g\in G^{1st}$ and $\mathrm{lpp}_X(g)$ divides $\mathrm{lpp}_X(h)$; if $g=0$, then we have $h_0=\bar{g}\in G_{U,X}$ and $\mathrm{lpp}_X(\bar{g})$ divides $\mathrm{lpp}_X(h)$. As a result, $G^{1st}\cup G_{U,X}$ is a Gröbner basis for $\langle F\cup E\rangle$.

Let $G'=G^{1st}\cup G_{U,X}\cup E$ and $G_0=G_r\cup G_{U,X}=(G^{1st}\cap k[U])\cup G_{U,X}\cup E$. Then G' is a Gröbner basis for $\langle F\cup E\rangle$. If the specialization σ satisfies 1 and 2, then it is easy to check $\sigma(\mathrm{lc}_X(g))\neq 0$ for any g in G_m and $\sigma(g')=0$ for any $g'\in G_0$. Then by Corollary 4.4, the set $\sigma(G_m)$ is a Gröbner basis for $\langle\sigma(F)\rangle$. Using (1), we have $\{\sigma(g+\bar{g})\mid gy+\bar{g}\in G_{m,y}\}$ is a Gröbner basis of $\langle\sigma(F)\rangle$ in $L[X]$ w.r.t. $\prec_X$. □

Based on the above theorem, we give below another algorithm for computing a comprehensive Gröbner basis on $V(E)\setminus V(N)$ for $F\subset k[U,X]$.

Algorithm CGB-Polynomial(E,N,F)
Input: (E,N,F): E, N, finite subsets of $k[U]$; F, a finite subset of $k[U,X]$.
Output: a comprehensive Gröbner basis of the set F on $V(E)\setminus V(N)$.

1. $\mathcal{CGS}:=$ CGSMainPoly(E,N,F), where $\mathcal{CGS}$ is a finite set of 3-tuples (E_i,N_i,G_i) such that $\{(V(E_i)\setminus V(N_i),G_i^{1st})\}$, where $G_i^{1st}=\{g\mid gy+\bar{g}\in G_i\}$, constitutes a comprehensive Gröbner system on $V(E)\setminus V(N)$ for F, and for each $gy+\bar{g}\in G_i$, $g+\bar{g}\in\langle F\rangle$ and $\sigma(\bar{g})$ is 0 for every parameter specialization σ from $V(E_i)\setminus V(N_i)$.
2. Return $\{g+\bar{g}\mid gy+\bar{g}\in G_i$ for all $i\}$.

Below we assume that all Gröbner basis computations are done in $k[U,X,y]$ using the block order with $U\ll X\ll y$.

Algorithm CGSMainPoly(E,N,F)
Input: (E,N,F): E, N, finite subsets of $k[U]$; F, a finite subset of $k[U,X]$.
Output: $\mathcal{CGS}$: a finite set of 3-tuples (E_i,N_i,G_i) such that $\{(V(E_i)\setminus V(N_i),G_i^{1st})\}$, where $G_i^{1st}=\{g\mid gy+\bar{g}\in G_i\}$, constitutes a comprehensive Gröbner system on $V(E)\setminus V(N)$ for F, and for each $gy+\bar{g}\in G_i$, $g+\bar{g}\in\langle F\rangle$ and $\sigma(\bar{g})$ is 0 for every parameter specialization σ from $V(E_i)\setminus V(N_i)$.

1. If inconsistent(E,N), then return $\emptyset$.
2. Otherwise, $G_0:=$ ReducedGröbnerBasis$(\{fy\mid f\in F\}\cup\{ey-e\mid e\in E\})$.
3. $G:=G_0\setminus(G_0\cap k[U,X])$ and $G^{1st}:=\{g\mid gy+\bar{g}\in G\}$.
4. If there exists $gy+\bar{g}\in G$ such that $g=1$, then return $\{(E,N,\{gy+\bar{g}\})\}$.
5. Let $G_{r,y}:=\{gy+\bar{g}\in G\mid g\in k[U]\}$ and $G_r:=\{g\mid gy+\bar{g}\in G_{r,y}\}\cup E$.
6. If inconsistent$(E,G_r\times N)$, then $\mathcal{CGS}:=\emptyset$, else $\mathcal{CGS}:=\{(E,G_r\times N,G_{r,y})\}$.
7. If inconsistent(G_r,N), then return $\mathcal{CGS}$.
8. Otherwise, let $G_m:=$ MDBasis$(G^{1st}\setminus G_r)$ and $G_{m,y}:=\{gy+\bar{g}\in G\setminus G_{r,y}\mid g\in G_m\}$.
9. If consistent$(G_r,N\times\{h\})$, then $\mathcal{CGS}:=\mathcal{CGS}\cup\{(G_r,N\times\{h\},G_{m,y})\}$, where $h=\mathrm{lcm}\{h_1,\ldots,h_k\}$ and $\{h_1,\ldots,h_k\}=\{\mathrm{lc}_X(g)\mid g\in G_m\}$.
10. Return $\mathcal{CGS}\cup\bigcup_{h\in[h_1,\ldots,h_k]}$ CGSMainPoly$(G_r\cup\{h_i\},N\times\{h_1h_2\cdots h_{i-1}\},\{g+\bar{g}\mid gy+\bar{g}\in G\setminus G_{r,y}\})$.

The correctness and termination of the above algorithm can be established in a way similar to those of CGB-Module.

5. A simple example

Note that the algorithms CGB-Module and CGB-Polynomial have the same theoretical base. So we only illustrate the algorithm CGB-Module by using the same example discussed in Kapur et al. (2010) primarily to help an interested reader to see the differences between the algorithm in Kapur et al. (2010) and the new algorithm of this paper. The discussion here is however self-contained.

Example 5.1. Let $F = \{ax - b, by - a, cx^2 - y, cy^2 - x\} \subset \mathbb{Q}[a, b, c][x, y]$, with the block order $\prec_{X,U}$, $\{a, b, c\} \ll \{x, y\}$; within each block, $\prec_X$ and $\prec_U$ are graded reverse lexicographic orders with $y < x$ and $c < b < a$, respectively.

At the beginning, $F^{(1)} = F = \{ax - b, by - a, cx^2 - y, cy^2 - x\}$, $E^{(1)} = \emptyset$ and $N^{(1)} = \{1\}$. We compute a comprehensive Gröbner system for $\{(f, 0) \mid f \in F^{(1)}\} \in (\mathbb{Q}[a, b, c][x, y])^2$ using the tuple representation, so that along every branch, for every polynomial in a Gröbner basis, we can also extract the original polynomial from the input ideal generated by $F^{(1)}$ to maintain the faithfulness of the output. In the following procedure, checking whether a constructible set, such as $V(E) \setminus V(N)$, is empty is done by using methods given in Kapur et al. (2010).

(1) The set $V(E^{(1)}) \setminus V(N^{(1)}) = V(0) \setminus V(1) = \mathbb{C}^3$ is not empty. The reduced Gröbner basis of the $\mathbb{Q}[a, b, c][x, y]$-module $\langle (f, 0) \mid f \in F^{(1)} \rangle \subset (\mathbb{Q}[a, b, c][x, y])^2$ w.r.t. the order extended by $\prec_{X,U}$ in POT fashion, is

$$\begin{aligned}\mathbf{G}_0^{(1)} = \big\{ &\big(x^3 - y^3, 0\big), \big(cx^2 - y, 0\big), \big(ay^2 - bc, 0\big), \big(cy^2 - x, 0\big),\\ &(ax - b, 0), (bx - acy, 0), \big(a^2y - b^2c, 0\big), (by - a, 0), \big(a^6 - b^6, 0\big),\\ &\big(a^3c - b^3, 0\big), \big(b^3c - a^3, 0\big), \big(ac^2 - a, 0\big), \big(bc^2 - b, 0\big)\big\}.\end{aligned}$$

Let $\mathbf{G}^{(1)} := \mathbf{G}_0^{(1)} \setminus \{(g, \bar{g}) \in \mathbf{G}_0^{(1)} \mid g = 0\} = \mathbf{G}_0^{(1)}$ and $G^{1st(1)}$ be the set $\{g \mid (g, \bar{g}) \in \mathbf{G}^{(1)}\}$. Next denote $\mathbf{G}_r^{(1)} := \{(g, \bar{g}) \in \mathbf{G}^{(1)} \mid g \in \mathbb{Q}[a, b, c]\} = \{(a^6 - b^6, 0), (a^3c - b^3, 0), (b^3c - a^3, 0), (ac^2 - a, 0), (bc^2 - b, 0)\}$, and correspondingly, $G_r^{(1)} := \{g \mid (g, \bar{g}) \in \mathbf{G}_r^{(1)}\}$. Boldfaced symbols such as $\mathbf{G}$ and $\mathbf{G}_r$ are used for sets of vectors in $(\mathbb{Q}[a, b, c][x, y])^2$, while regular symbols such as G^{1st} and G_r, denote polynomials sets in $\mathbb{Q}[a, b, c][x, y]$ that are constructed by the first components of the corresponding vector sets.

The constructible set $(V(E^{(1)}) \setminus V(G_r^{(1)})) \setminus V(N^{(1)}) = \mathbb{C}^3 \setminus V(G_r^{(1)})$ is not empty. This implies that $(\mathbb{C}^3 \setminus V(G_r^{(1)}), \mathbf{G}_r)$ is a trivial branch of the comprehensive Gröbner system for $F^{(1)}$.

(2) Since $G^{1st(1)} \setminus G_r^{(1)} = \{x^3 - y^3, cx^2 - y, ay^2 - bc, cy^2 - x, ax - b, bx - acy, a^2y - b^2c, by - a\}$, then let $G_m^{(1)} := \text{MDBasis}(G^{1st(1)} \setminus G_r^{(1)}) = \{bx - acy, by - a\}$ and $\mathbf{G}_m^{(1)} := \{(bx - acy, 0), (by - a, 0)\}$. Further, $h^{(1)} := \text{lcm}\{\text{lc}_X(bx - acy), \text{lc}_X(by - a)\} = b$. This gives us another branch of comprehensive system for $F^{(1)}$ corresponding to the case when all polynomials in $G_r^{(1)}$ are 0 and $b \neq 0$ by Theorem 4.6: $(V(G_r^{(1)}) \setminus V(b), \mathbf{G}_m)$. Note that $V(G_r^{(1)}) \setminus V(b)$ is not empty.

(3) The next branch to consider is when $b = 0$. The Gröbner basis of $G_r^{(1)} \cup \{b\}$ is $\{a^3, ac^2 - a, b\}$, which is the input $E^{(2)}$ in the recursive call of CGSMainMod, with the other input being $N^{(2)} = \{1\}$ and $F^{(2)} = \{g + \bar{g} \mid (g, \bar{g}) \in \mathbf{G}^{(1)} \setminus \mathbf{G}_r^{(1)}\}$.

Since $V(E^{(2)}) \setminus V(N^{(2)})$ is not empty, we can compute the reduced Gröbner basis for $\{(f, 0) \mid f \in F^{(2)}\} \cup \{(a^3, -a^3), (ac^2 - a, -ac^2 + a), (b, -b)\}$. By removing the tuples whose first component is 0, we get $\mathbf{G}^{(2)} = \{(x^3 - y^3, 0), (cx^2 - y, 0), (cy^2 - x, 0), (a, -by), (b, -b)\}$ of which $\mathbf{G}_r^{(2)} = \{(a, -by), (b, -b)\}$. Similarly, denote $G^{1st(2)} = \{g \mid (g, \bar{g}) \in \mathbf{G}^{(2)}\}$ and $G_r^{(2)} = \{g \mid (g, \bar{g}) \in \mathbf{G}_r^{(2)}\}$. We can check the set $V(E^{(2)}) \setminus V(G_r^{(2)})$ is empty, so no element in $\mathbf{G}_r^{(2)}$ contributes to the comprehensive Gröbner system.

Next, $G_m^{(2)} = \{cx^2 - y, cy^2 - x\}$, $\mathbf{G}_m^{(2)} = \{(cx^2 - y, 0), (cy^2 - x, 0)\}$ and $h^{(2)} = \text{lcm}(\text{lc}_X(cx^2 - y), \text{lc}_X(cy^2 - x)) = c$. This results in another branch: $(V(G_r^{(2)}) \setminus V(c), \mathbf{G}_m^{(2)})$.

(4) For the case when $h^{(2)} = c = 0$, the set $E^{(3)} = \{a, b, c\}$ which is the Gröbner basis of $G_r^{(2)} \cup \{c\}$. Now $N^{(3)} = \{1\}$ and $F^{(3)} = \{x^3 - y^3, cx^2 - y, cy^2 - x\}$. Computing the reduced Gröbner basis for $\{(f, 0) \mid f \in F^{(3)}\} \cup \{(a, -a), (b, -b), (c, -c)\}$ and removing the tuples whose first component is 0,

we get $\mathbf{G}^{(3)} = \{(x, -cy^2), (y, -cx^2), (a, -a), (b, -b), (c, -c)\}$. Then, $\mathbf{G}_r^{(3)} = \{(a, -a), (b, -b), (c, -c)\}$, $G_m^{(3)} = \{x, y\}$ and $\mathbf{G}_m^{(3)} = \{(x, -cy^2), (y, -cx^2)\}$. Further, $h^{(3)} = \mathrm{lcm}(\mathrm{lc}_X(x), \mathrm{lc}_X(y)) = 1$. Similarly, denote $G^{(3)}$ and $G_r^{(3)}$ as before. This gives the last branch: $(V(G_r^{(3)}), \mathbf{G}_m^{(3)})$.

Since $h^{(3)} = 1$, no more branches are generated and the algorithm terminates. Thus, we obtain a comprehensive Gröbner system for F:

$$\begin{cases} \{(a^6 - b^6, 0), (a^3c - b^3, 0), & \text{if } a^6 - b^6 \neq 0 \text{ or } a^3c - b^3 \neq 0 \\ \quad (b^3c - a^3, 0), (ac^2 - a, 0), & \quad \text{or } b^3c - a^3 \neq 0 \text{ or } ac^2 - a \neq 0 \\ \quad (bc^2 - b, 0)\}, & \quad \text{or } bc^2 - b \neq 0, \\ \{(bx - acy, 0), (by - a, 0)\}, & \text{if } a^6 - b^6 = a^3c - b^3 \\ & \quad = b^3c - a^3 = ac^2 - a \\ & \quad = bc^2 - b = 0 \text{ and } b \neq 0, \\ \{(cx^2 - y, 0), (cy^2 - x, 0)\} & \text{if } a = b = 0 \text{ and } c \neq 0, \\ \{(x, -cy^2), (y, -cx^2)\} & \text{if } a = b = c = 0. \end{cases}$$

An interested reader would observe comparing the above output with the output from Kapur et al. (2010) that except for the last branch, the outputs are the same. In Kapur et al. (2010), the last branch for the case when $a = b = c = 0$, the Gröbner basis is: $\{x, y\}$, whereas in the above the Gröbner basis is: $\{x - cy^2, y - cx^2\}$, when the tuple representation is replaced by the corresponding polynomials from the ideal of F. $x - cy^2$ is the faithful polynomial from the ideal of F corresponding to the output element x in Kapur et al. (2010); similarly, $y - cx^2$ is the faithful polynomial corresponding to y.

A comprehensive Gröbner basis of F, after removing the duplicate ones, can be obtained directly from the above comprehensive Gröbner system. That is $\{a^6 - b^6, a^3c - b^3, b^3c - a^3, ac^2 - a, bc^2 - b, bx - acy, by - a, cx^2 - y, cy^2 - x\}$.

6. Implementation and comparative performance

Both the algorithms CGB-Module and CGB-Polynomial have been implemented on the computer algebra system *Singular* (Decker et al., 2012).[2] The implementation has been experimented on a number of examples from different application domains including geometry theorem proving and computer vision, and it has been compared with implementations of other algorithms. Since the proposed algorithm uses the new technique and basic module/polynomial operations, it is efficient and can compute comprehensive Gröbner basis for most problems in a few seconds.

Table 1 shows a comparison of our implementations on Singular with other existing algorithms for computing comprehensive Gröbner bases, including: Suzuki–Sato algorithm implemented by Nabeshima in Risa/Asir (package PGB, ver20090915) and the function "cgb" for computing comprehensive Gröbner bases in Reduce (package RedLog). The versions of Singular, Risa/Asir and Reduce are ver3-1-2, ver20090715 and free CSL version, respectively.

The implementation has been tried on many examples. Many of these examples could be solved very quickly. To generate complex examples, we modified problems F1, F2, F3, F4, F5, F6 and F8 in Nabeshima (2007), and labeled them as S1–S7. The polynomials for these problems are given below: We have also been successful in solving the famous P3P problem for pose-estimation from computer vision, which is investigated by Gao et al. (2003) using the characteristic set method; see the polynomial system below.

S1: $F = \{ax^4y + xy^2 + bx, x^3 + 2xy + cy, x^2y + bx^2\}$, $X = \{x, y\}$, $U = \{a, b, c\}$;
S2: $F = \{ax^2y^3 + ay + by, x^2y^2 + xy + 2x, ax^2 + by + 2x\}$, $X = \{x, y\}$, $U = \{a, b, c\}$;
S3: $F = \{ax^4 + cx^2 + y, bx^3 + x^2 + y + 2, cx^2 + dx + y\}$, $X = \{x, y\}$, $U = \{a, b, c, d\}$;
S4: $F = \{ax^3y + cxz^2, x^4y + 3dy + z, cx^2 + bxy, x^2y^2 + x^2, x^5 + y^5\}$, $X = \{x, y, z\}$, $U = \{a, b, c, d\}$;

[2] Implementation codes on Singular are available at http://www.mmrc.iss.ac.cn/~dwang/.

Table 1
Timings.

Exa.	Algorithm	Time (s)	Exa.	Algorithm	Time (s)
S1	New-mod(S)	1.155	S5	New-mod(S)	0.890
	New-poly(S)	1.191		New-poly(S)	1.388
	cgb(R)	24.960		cgb(R)	2.395
	SuzukiSato(A)	>1 h		SuzukiSato(A)	>1 h
S2	New-mod(S)	0.530	S6	New-mod(S)	0.738
	New-poly(S)	0.716		New-poly(S)	0.842
	cgb(R)	134.513		cgb(R)	9.496
	SuzukiSato(A)	1.56		SuzukiSato(A)	error
S3	New-mod(S)	0.451	S7	New-mod(S)	18.303
	New-poly(S)	1.099		New-poly(S)	15.217
	cgb(R)	309.334		cgb(R)	>1 h
	SuzukiSato(A)	error		SuzukiSato(A)	>1 h
S4	New-mod(S)	2.633	P3P	New-mod(S)	19.565
	New-poly(S)	3.109		New-poly(S)	14.564
	cgb(R)	54.871		cgb(R)	>1 h
	SuzukiSato(A)	>1 h		SuzukiSato(A)	>1 h

S5: $F = \{y^3 + bx, ax^2y + bxy + cx, y^2 + bx^2y + cxy\}$, $X = \{x, y\}$, $U = \{a, b, c\}$;
S6: $F = \{dx^4 + ax^3 + bx^2 + cx + d, 4bx^3 + 3ax^2 + 2bx + c\}$, $X = \{x\}$, $U = \{a, b, c, d\}$;
S7: $F = \{ax^2 + byz, cw^2 + by + z, (x-z)^2 + (y-w)^2, 2dxw - 2byz\}$, $X = \{x, y, z, w\}$, $U = \{a, b, c, d\}$;
P3P: $F = \{(1-a)y^2 - ax^2 - py + arxy + 1, (1-b)x^2 - by^2 - qx + brxy + 1\}$, $X = \{x, y\}$, $U = \{p, q, r, a, b\}$.

For all these examples, the term orders used on X are graded reverse lexicographic orders.

In Table 1, entries labeled with New-mod(S) and New-poly(S) are the algorithms CGB-Module and CGB-Polynomial implemented on Singular; (R) and (A) stand for Reduce and Risa/Asir, respectively. The label "*error*" is included if an implementation ran out of memory or broke down. The timings were obtained by running the implementations on Core i5 2.8 GHz with 12 GB memory running 64-bit Windows 7.

As is evident from Table 1, the proposed algorithms have better performance in contrast to other algorithms discussed in the literature for two possible reasons. Firstly, the proposed approach is simpler and easier to implement leading to considerable savings in computational performance. Secondly, the algorithm for computing comprehensive Gröbner systems proposed in Kapur et al. (2010) is exploited in our implementation. As shown in Kapur et al. (2010), this algorithm is one of the most efficient algorithms at present, which also speeds up the implementations for computing comprehensive Gröbner bases. The reader would notice that the algorithm CGB-Module usually performs better than the algorithm CGB-Polynomial on simple examples. However, for more complicated examples such as S7 and P3P, the algorithm CGB-Polynomial has a better performance. An interesting topic for further investigation is a better identification of classes of problems for which these two algorithms are the most efficient.

Since the core of our approach is the use of our algorithm for computing comprehensive Gröbner systems proposed in Kapur et al. (2010), we compared our implementation with Montes' implementation *cgsdr* from Singular library grobcov.lib. Both implementations are implemented on Singular. In Table 2, timings for the above examples on the computer (Core i5 2.8 GHz, 12 GB memory, 64-bit Windows 7) are given.

In Table 2, *Kapur–Sun–Wang* refers to the algorithm we proposed in Kapur et al. (2010), and *Montes* means the implementation *cgsdr* from Singular library grobcov.lib.

7. Concluding remarks

We have adapted the algorithm proposed in Kapur et al. (2010) for computing a comprehensive Gröbner system of a parameterized polynomial system F such that the new algorithms not only pro-

Table 2
Timings.

Exa.	Algorithm	Time (s)	Exa.	Algorithm	Time (s)
S1	Kapur–Sun–Wang	0.240	S5	Kapur–Sun–Wang	1.357
	Montes	0.714		Montes	7.036
S2	Kapur–Sun–Wang	0.521	S6	Kapur–Sun–Wang	0.990
	Montes	1.546		Montes	67.780
S3	Kapur–Sun–Wang	0.045	S7	Kapur–Sun–Wang	1.672
	Montes	>1 h		Montes	>1 h
S4	Kapur–Sun–Wang	0.120	P3P	Kapur–Sun–Wang	2.309
	Montes	11.966		Montes	2.811

duce comprehensive Gröbner systems of F but they also generate comprehensive Gröbner bases of F. The main idea is to use polynomials from the ideal generated by F during the computation along various branches corresponding to constructible sets specializing parameters in the algorithm in Kapur et al. (2010). Polynomials from $\langle F \rangle$ are represented as tuples, with the first component corresponding to its nonzero part under the specialization and the second component being zero under the specialization. The key steps of a Gröbner basis computation including reduction of a polynomial by another polynomial and S-polynomial construction, are performed on these tuple representations; these steps can be done by computing Gröbner bases of a module or a larger ideal.

The new algorithms produce comprehensive Gröbner systems, in which each branch is a finite set of tuples along a constructible set (which is specified by a finite set of equalities over parameters and a finite set of disequalities over parameters), with the properties (i) the constructible sets constitute a partition over the set of parameter specializations under consideration, and (ii) for every parameter specialization in the constructible set of the branch, the second component of every tuple is 0 under the specialization and the leading coefficient of the first component in every tuple is nonzero under the specialization, and most importantly, (iii) the sum of the first component and the second component in the tuple is in the ideal generated by the input F. For generating a comprehensive Gröbner system, the second component of these tuples do not give any useful information and can hence be discarded. Using these second components however, a comprehensive Gröbner basis is the union over every branch of the set of polynomials obtained by adding the two components of each tuple. Such a comprehensive Gröbner basis is faithful since all the polynomials in the basis are also in the ideal; thus, both a comprehensive Gröbner system as well as a comprehensive Gröbner basis of a parametric polynomial system are simultaneously generated. Further, no branches with empty segments (inconsistent set of parametric polynomial constraints) are generated, and branches are disjoint and the associated Gröbner basis for every branch has a fixed set of leading power products, also implying that there is at most one branch for the parametric constraints for which the Gröbner basis of a polynomial system is $\{1\}$.

The algorithm is faster in practice than known existing algorithms primarily because it does not need to use primary decomposition of parametric constraints as well as it generates fewer branches.

The above construction can be used to adapt all known algorithms for computing a comprehensive Gröbner system. We believe that various optimization criteria to discard redundant computations can also be integrated in the proposed algorithm. As a result, future advances to make comprehensive Gröbner systems more efficient can be directly exploited in the proposed approach and algorithms.

A theoretical contribution of this paper is a more generalized stable condition for parametric polynomial systems which serves as the base of the proposed algorithms for computing comprehensive Gröbner bases. We also believe this new stable condition can lead to more interesting results.

Using insights discussed above, we are investigating the design of a new algorithm for computing a minimal comprehensive Gröbner basis of a parametric polynomial systems, which will be reported in a forthcoming paper. Using this notion, we are able to define a canonical minimal comprehensive Gröbner basis, unlike the notion in Weispfenning (2003), where a canonical comprehensive Gröbner basis is defined but it does not have the property of being minimal.

Acknowledgements

We thank the referees for their many constructive suggestions which helped in highlighting the key contributions of the proposed approach. We particularly thank Professor Antonio Montes for his helpful discussions and encouragement.

References

Chen, C., Golubitsky, O., Lemaire, F., Moreno Maza, M., Pan, W., 2007. Comprehensive triangular decomposition. In: Proc. CASC'07. In: Lecture Notes in Comput. Sci., vol. 4770. Springer, Berlin, pp. 73–101.

Chen, X.F., Li, P., Lin, L., Wang, D.K., 2005. Proving geometric theorems by partitioned-parametric Gröbner bases. In: Proc. Automated Deduction in Geometry (ADG) 2004. In: Lecture Notes in Comput. Sci., vol. 3763. Springer, Berlin, pp. 34–43.

Cox, D., Little, J., O'Shea, D., 2005. Using Algebraic Geometry, second ed. Springer, New York.

Decker, W., Greuel, G.-M., Pfister, G., Schönemann, H., 2012. SINGULAR 3-1-4 – A computer algebra system for polynomial computations. http://www.singular.uni-kl.de.

Donald, B., Kapur, D., Mundy, J.L. (Eds.), 1992. Symbolic and Numerical Computation for Artificial Intelligence. Computational Mathematics and Applications. Academic Press Ltd., London.

Gao, X.S., Chou, S.C., 1992. Solving parametric algebraic systems. In: Proc. ISSAC'1992. ACM Press, New York, pp. 335–341.

Gao, X.S., Hou, X.R., Tang, J.L., Chen, H.F., 2003. Complete solution classification for the perspective-three-point problem. IEEE Trans. PAMI 25 (8), 930–943.

Kalkbrener, K., 1997. On the stability of Gröbner bases under specialization. J. Symbolic Comput. 24 (1), 51–58.

Kapur, D., 1995. An approach for solving systems of parametric polynomial equations. In: Saraswat, Van Hentenryck (Eds.), Principles and Practice of Constraint Programming. MIT Press, Cambridge.

Kapur, D., 2006. A quantifier-elimination based heuristic for automatically generating inductive assertions for programs. J. Syst. Sci. Complex. 19 (3), 307–330.

Kapur, D., Sun, Y., Wang, D.K., 2010. A new algorithm for computing comprehensive Gröbner systems. In: Proc. ISSAC'2010. ACM Press, New York, pp. 29–36.

Kapur, D., Sun, Y., Wang, D.K., 2011. Computing comprehensive Gröbner systems and comprehensive Gröbner bases simultaneously. In: Proc. ISSAC'2011. ACM Press, New York, pp. 193–200.

Manubens, M., Montes, A., 2006. Improving DISPGB algorithm using the discriminant ideal. J. Symbolic Comput. 41 (11), 1245–1263.

Manubens, M., Montes, A., 2009. Minimal canonical comprehensive Gröbner system. J. Symbolic Comput. 44 (5), 463–478.

Montes, A., 2002. A new algorithm for discussing Gröbner basis with parameters. J. Symbolic Comput. 33 (1–2), 183–208.

Montes, A., Recio, T., 2007. Automatic discovery of geometry theorems using minimal canonical comprehensive Gröbner systems. In: Proc. Automated Deduction in Geometry (ADG) 2006. In: Lecture Notes in Artificial Intelligence, vol. 4869. Springer, Berlin, Heidelberg, pp. 113–138.

Montes, A., Wibmer, M., 2010. Gröbner bases for polynomial systems with parameters. J. Symbolic Comput. 45 (12), 1391–1425.

Nabeshima, K., 2007. A speed-up of the algorithm for computing comprehensive Gröbner systems. In: Proc. ISSAC'2007. ACM Press, New York, pp. 299–306.

Sun, Y., Wang, D.K., 2011. Solving detachability problem for the polynomial ring by signature-based Gröbner basis algorithms. Preprint, arXiv:1108.1301 [cs.SC].

Sun, Y., Wang, D.K., Ma, X.D., Zhang, Y., 2012. A signature-based algorithm for computing Gröbner bases in solvable polynomial algebras. In: Proc. ISSAC'2012. ACM Press, New York, pp. 351–358.

Suzuki, A., Sato, Y., 2006. A simple algorithm to compute comprehensive Gröbner bases using Gröbner bases. In: Proc. ISSAC'2006. ACM Press, New York, pp. 326–331.

Weispfenning, V., 1992. Comprehensive Gröbner bases. J. Symbolic Comput. 14 (1), 1–29.

Weispfenning, V., 2003. Canonical comprehensive Gröbner bases. J. Symbolic Comput. 36 (3–4), 669–683.

Wibmer, M., 2007. Gröbner bases for families of affine or projective schemes. J. Symbolic Comput. 42 (8), 803–834.