数式処理 *Bulletin of JSSAC (2008)*
Vol. 15, No. 1, pp. 3 - 24

奨励賞論文

パラメトリック・グレブナ基底計算のテクニック

鍋島克輔*

科学技術振興機構 (JST) / 東京大学大学院 情報理工学系研究科

(RECEIVED 2008/3/21)

概　要

In 2006 the Suzuki-Sato algorithm for computing comprehensive Gröbner systems, was introduced. In this paper we improve the Suzuki-Sato algorithm by new efficient techniques. The original Suzuki-Sato algorithm often creates overmuch cells of the parameter space for comprehensive Gröbner systems. Therefore the computation becomes heavy. However, by using our techniques, 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.

1　はじめに

本稿は，係数にパラメータを持つ多項式イデアルのパラメトリック・グレブナ基底の計算のテクニックについて考える．パラメトリック・グレブナ基底は主に “包括的グレブナ基底” と “包括的グレブナ基底系” の二種類ある．それぞれ，Weispfenning [Wei92] により紹介され，その後，数々の研究 [DS97, Mon02, MM06, SS03, SS06, Wei03] がなされると共にいくつかの計算機代数システムに実装されている．

本稿では 2006 年に Suzuki, Sato [SS06] によって発表された包括的グレブナ基底系の計算アルゴリズムをより効率のよいものに改良するためのテクニックについて考える．

グレブナ基底の有用性はさまざまな分野で広く知られており今だ多くの研究者が研究をしている [Buc65, BW98]．もちろん，数々の問題においてパラメータの付いた多項式が現れる場合もありこれらの問題を解くにはパラメトリック・グレブナ基底を計算する必要がある．このことよりパラメトリック・グレブナ基底はそれ自体が重要な研究のテーマでもあり，問題を解くための重要な道具でもある．本稿ではパラメトリック・グレブナ基底として “包括的グレブナ基底系” を考える．包括的グレブナ基底系とは，雑に言うとパラメータ空間のセルと各セルに伴うパラメータ付きの多項式の集合からなる集合である．パラメータを持つ多項式イデアルを I とする．もし，ある項順序に関しての I の包括的グレブナ基底からセルとして $\mathbb{P}$ を取りそれに伴うパラ

*Katsusuke_Nabeshima@ipc.i.u-tokyo.ac.jp

メータ付きの多項式を G とすると，このときパラメータの取る値を $\mathbb{P}$ から任意の値を取り G に代入したものはいつでもそのパラメータの値をイデアル I に代入した物のイデアルのグレブナ基底になっている．このようなものが包括的グレブナ基底系である．

包括的グレブナ基底系の計算アルゴリズムとして Suzuki-Sato アルゴリズムがある．一般的な包括的グレブナ基底系の計算手順は木構造をなし，各枝の分枝は多項式の先頭項がゼロになるかならないかで決められる．しかしながら，Suzuki-Sato アルゴリズムはグレブナ基底の安定性の理論を用い，各枝の分枝として多項式の先頭項がゼロになる場合のみを用いている．本稿では Suzuki-Sato アルゴリズムでなす木構造の分枝に先頭項がゼロにならない場合 (“$\neq 0$”) の枝を作ると言う操作を加えて Suzuki-Sato アルゴリズムを改良する．この操作により全体として小さな木構造を作ることができ計算の効率化をはかると共により良い出力も得ることができる．

本稿の計画として 2 章で本稿で扱う記号の紹介をし，3 章で Suzuki-Sato アルゴリズムを見る．4 章では，メインとなるテクニックと新しいアルゴリズムについて述べ，5 章ではいくつかの計算の戦略とベンチマークについて述べる．最後に 6 章においてこの論文をまとめる．

2 記号

本稿において以下の記号を固定する．K を体とし L を K の拡大体とする．$\bar{X} = \{X_1, \ldots, X_n\}$, $\bar{A} = \{A_1, \ldots, A_m\}$ を変数とし $\bar{X} \cap \bar{A} = \emptyset$ とする．$\mathrm{pp}(\bar{X})$，$\mathrm{pp}(\bar{A})$, $\mathrm{pp}(\bar{A}, \bar{X})$ を順に $\bar{X}$ の項 (power product) の集合，$\bar{A}$ の項の集合，$\bar{X} \cup \bar{A}$ の項の集合とする．$\mathbb{N}$ を自然数の集合（0 を含む），$\mathbb{Q}$ を有理数体，$\mathbb{C}$ を複素数体とする．$K[\bar{A}][\bar{X}] := (K[\bar{A}])[\bar{X}]$ を多項式環 $K[\bar{A}]$ を係数ドメインとする多項式環とし，いま多項式 f を 0 でない $K[\bar{A}, \bar{X}]$ (または $K[\bar{A}][\bar{X}]$) の元とする．ここで $\mathrm{pp}(\bar{A}, \bar{X})$ (または $\mathrm{pp}(\bar{X})$) 上の任意の項順序を $\succ$ としたとき以下を定義する．(もし，f が $K[\bar{A}][\bar{X}]$ の元のとき $K[\bar{A}, \bar{X}]$ の元との混乱をさせるため添え字 $\bar{X}$ を付ける．) f の**先頭項**を $\mathrm{lpp}(f)$ (または $\mathrm{lpp}_{\bar{X}}(f)$) (leading power product)，**先頭係数**を $\mathrm{lc}(f)$ (または $\mathrm{lc}_{\bar{X}}(f)$) (leading coefficient)，**先頭単項**を $\mathrm{lm}(f) := \mathrm{lc}(f)\,\mathrm{lm}(f)$ (または $\mathrm{lm}_{\bar{X}}(f)$) (leading monomial) と書く．多項式 f の**単項の集合**を $\mathrm{Mono}(f)$ (または $\mathrm{Mono}_{\bar{X}}(f)$) と書く．もし，$\mathrm{lpp}(f) = A_1^{\alpha_1} \cdots A_m^{\alpha_m} X_1^{\beta_1} \cdots X_n^{\beta_n} \in \mathrm{pp}(\bar{A}, \bar{X})$ なら**次数**として $\deg_{\{\bar{A},\bar{X}\}}(f) := (\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_n) \in \mathbb{N}^{m+n}$，$\deg_{X_1}(f) := \beta_1 \in \mathbb{N}$ と表す．F を $K[\bar{A}, \bar{X}]$ (または $K[\bar{A}][\bar{X}]$) の多項式の集合とする．そのとき，$\mathrm{lc}(F) := \{\mathrm{lc}(f) : f \in F\}$ (または $\mathrm{lc}_{\bar{X}}(F) := \{\mathrm{lc}_{\bar{X}}(f) : f \in F\}$), $\mathrm{lpp}(f) := \{\mathrm{lpp}(f) : f \in F\}$ (または $\mathrm{lpp}_{\bar{X}}(F) := \{\mathrm{lpp}_{\bar{X}}(f) : f \in F\}$) とする．

例 1

a, b, x, y を変数とし，$f = 2ax^2y + bx^2y + 3x + by + 1$ を多項式とする．もし f を $\mathbb{Q}[x, y, a, b]$ の元と見ると，辞書式順序 $x \succ y \succ a \succ y$ に関して以下となる．$\mathrm{lpp}(f) = ax^2y$, $\mathrm{lc}(f) = 2$, $\mathrm{lm}(f) = 2ax^2y$, $\mathrm{Mono}(f) = \{2ax^2y, bx^2y, 3x, by, 1\}$．次に，$f$ を $\mathbb{Q}[a, b][x, y]$ の元と見ると，辞書式順序 $x \succ y$ に関して以下となる．$\mathrm{lpp}_{\{x,y\}}(f) = x^2y$, $\mathrm{lc}_{\{x,y\}}(f) = 2a + b$, $\mathrm{lm}_{\{x,y\}}(f) = (2a + b)x^2y$, $\mathrm{Mono}_{\{x,y\}}(f) = \left\{(2a + b)x^2y, 3x, by, 1\right\}$.

本稿では $\mathbb{V}$ とアングル・ブラケット $\langle \cdot \rangle$ を次のように定義する．$f_1, \ldots, f_k \in K[\bar{A}]$ において，$\mathbb{V}(f_1, \ldots, f_k) \subseteq L^m$ を $f_1, \ldots, f_k$ のアフェイン代数多様体と定義する．すなわち，$\mathbb{V}(f_1, \ldots, f_k) =$

$\{\bar{a} \in L^m : f_1(\bar{a}) = \cdots = f_k(\bar{a}) = 0\}$. R を単位元を持つ可換環とする． そのとき，$f_1, \ldots, f_k \in R$ において $\langle f_1, \ldots, f_k \rangle := \{\sum_{i=1}^{s} h_i f_i : h_1, \ldots, h_k \in R\}$.

3 Suzuki-Sato のアプローチ

本章では 2006 年に Suzuki と Sato[SS06] によって発表されたパラメトリック・グレブナ基底計算アルゴリズムを紹介する．この Suzuki-Sato のアルゴリズムはグレブナ基底の安定性の理論を用いて構成されているので，まずグレブナ基底の安定性について述べ，次に Suzuki-Sato のアルゴリズムを紹介する．

3.1 グレブナ基底の安定性

ここでは $K[\bar{A}][\bar{X}]$ 上におけるイデアルのグレブナ基底の安定性について見る．グレブナ基底の安定性については多くの研究者がその性質を研究し論文 [Bec94, Gia87, Kal97, FGT01, Sat05] などで見ることができる．ここでは主に Kalkbrener [Kal97] の結果について述べる．

まず，$K[\bar{A}][\bar{X}]$ 上におけるイデアルのグレブナ基底の定義からこの章を始める．

定義 2 (グレブナ基底)
$I \subseteq K[\bar{A}][\bar{X}]$ をイデアル，G を $K[\bar{A}][\bar{X}]$ の部分集合，$\succ$ を $\mathrm{pp}(\bar{X})$ 上の単項順序とする．もし，G が $\mathrm{lm}_{\bar{X}}(I) = \langle \mathrm{lm}_{\bar{X}}(G) \rangle$ を満たすならば，G を I の $\succ$ に関するグレブナ基底と呼ぶ．

多項式環 $K[\bar{A}][\bar{X}]$ と $K[\bar{A}, \bar{X}]$ が同型ということと，ブロック項順序 $\bar{X} \gg \bar{A}$ を使うことでこのグレブナ基底は簡単に計算できることが次のようによく知られている．

アルゴリズム 3 (GröbnerBasis($F, \succ$))
Input F:$K[\bar{A}][\bar{X}]$ の有限集合， $\succ$: $\mathrm{pp}(\bar{X})$ 上の項順序,
Output G: $\langle F \rangle$ の $\succ$ に関するグレブナ基底.

(1) F を $K[\bar{A}, \bar{X}]$ の集合と考える．(明らかに $K[\bar{A}][\bar{X}]$ は $K[\bar{A}, \bar{X}]$ と同型である．)
(2) $\bar{X}$ が $\bar{A}$ より大 ($\bar{X} \gg \bar{A}$) となるブロック・オーダーにおいて $\langle F \rangle$ の簡約グレブナ基底 (reduced Gröbner basis) G を $K[\bar{A}, \bar{X}]$ 上で計算する．($\mathrm{pp}(\bar{X})$ 上の項順序は $\succ$ で $\mathrm{pp}(\bar{A})$ 上の項順序は任意でよい．)
(3) G を $K[\bar{A}][\bar{X}]$ の集合と見る．そのとき，G は $\langle F \rangle$ の $\succ$ に関するグレブナ基底である．

注意: $K[\bar{A}][\bar{X}]$ 上のグレブナ基底を得るためには，(2) において簡約グレブナ基底を計算する必要はなく，ノーマルなグレブナ基底を求める計算で十分である．しかしながら，Suzuki-Sato のアルゴリズムと新しいアプローチにおいてアルゴリズムの停止性を証明するために簡約グレブナ基底の性質が必要になる．そのため，ここでは (2) で簡約グレブナ基底を計算することにする．

特化準同型写像 (specialization homomorphism) $\sigma : K[\bar{A}] \to L$ を定義し，更にこれを $\sigma : K[\bar{A}][\bar{X}] \to L[\bar{X}]$ へと自然に拡張する．イデアル $I \subseteq K[\bar{A}][\bar{X}]$ の σ による像を $\sigma(I) := \{\sigma(f) : f \in I\} \subseteq L[\bar{X}]$ とする．

定義 4 (安定)
I を $K[\bar{A}][\bar{X}]$ のイデアルとし，$\sigma : K[\bar{A}] \to L$ を特化準同型写像，$\succ$ を $\mathrm{pp}(\bar{X})$ の項順序とする．その時，イデアル I が $\sigma(\mathrm{lm}_{\bar{X}}(I)) = \mathrm{lm}(\sigma(I))$ を満たすならば，I は σ と $\succ$ において**安定**と言う．($\sigma(\mathrm{lm}_{\bar{X}}(I)) := \{\sigma(\mathrm{lm}_{\bar{X}}(f)) : f \in I\}$, $\mathrm{lm}(\sigma(I)) := \{\mathrm{lm}(f) : f \in \sigma(I)\}$．)

次の定理はパラメトリック・グレブナ基底を求める Suzuki-Sato のアルゴリズムの鍵となる重要な定理である．

定理 5 (Kalkbrener (1997) [Kal97])
I を $K[\bar{A}][\bar{X}]$ のイデアル，$\sigma : K[\bar{A}] \to L$ を特化準同型写像，$\succ$ を $\mathrm{pp}(\bar{X})$ の項順序とし，$G = \{g_1, \ldots, g_s\}$ を I の $\succ$ に関してのグレブナ基底とする．G の各要素 g_i は順序付けられており次のようなると仮定する．$r \in \{1, \ldots, s\}$ が存在し
• 各 $i \in \{1, \ldots, r\}$ で $\sigma(\mathrm{lc}_{\bar{X}}(g_i)) \neq 0$ となる．
• 各 $i \in \{r+1, \ldots, s\}$ で $\sigma(\mathrm{lc}_{\bar{X}}(g_i)) = 0$ となる．
このとき次の 3 つは同値である．

(1) I は σ と $\succ$ において安定である．
(2) $\{\sigma(g_1), \ldots, \sigma(g_r)\}$ は $\succ$ に関して $\sigma(I)$ のグレブナ基底である．
(3) 各 $i \in \{r+1, \ldots, s\}$ で，$\sigma(g_i)$ は $\{\sigma(g_1), \ldots, \sigma(g_r)\}$ によって 0 へ簡約される．

3.2 Suzuki-Sato アルゴリズム

2006 年に Suzuki-Sato によってパラメトリック・グレブナ基底計算アルゴリズム [SS06] が発表された．このアルゴリズムは体を係数ドメインとする多項式環上で一般的なグレブナ基底を再帰的に計算することによって構成されており，他のパラメトリック・グレブナ基底計算アルゴリズムと比べると簡単な構造となっている．このアルゴリズムには前章で見たグレブナ基底の安定性の理論が使われている．

アルゴリズムの詳細を見る前にまず以下の記号を固定する．各 $\bar{a} \in L^m$ に対して特化準同型写像 (specialization homomorphism) $\sigma_{\bar{a}} : K[\bar{A}] \to L$ を各 A_i への a_i の代入により自然に定義し，更にこれを $\sigma_{\bar{a}} : K[\bar{A}][\bar{X}] \to L[\bar{X}]$ へと自然に拡張する．

パラメトリック・グレブナ基底として次の包括的グレブナ基底系を定義する．実際，本論文ではこの包括的グレブナ基底系 (comprehensive Gröbner system) を計算するための効率的で新しいテクニックを提唱する．

定義 6 (包括的グレブナ基底系)
F を $K[\bar{X}][\bar{A}]$ の有限部分集合，$S \subseteq L^m$ を代数構成的集合，$\succ$ を項順序とする．ペアの有限集合 $\mathcal{G} = \{(S_1, G_1), \ldots, (S_l, G_l)\}$ が $\langle F \rangle$ の $\succ$ に関する S 上の**包括的グレブナ基底系 (comprehensive Gröbner system)** であるとは以下を満たす時に言う．

1. $S_1, \ldots, S_l$ は L^m の構成的部分集合，$G_1, \ldots, G_l$ は $K[\bar{X}][\bar{A}]$ の有限部分集合であり，
2. $S_1 \cup \cdots \cup S_l \supseteq S$ を満たし，

3. 各 $i = 1, \ldots, l$ と各 $\bar{a} \in S_i$ に対して $\sigma_{\bar{a}}(G_i)$ が $\langle \sigma_{\bar{a}}(F) \rangle$ の $L[\bar{X}]$ 上での $\succ$ に関するグレブナ基底をなす.

特に L_m 上 $\langle F \rangle$ の包括的グレブナ基底系 (comprehensive Gröbner system) を単に $\langle F \rangle$ の $\succ$ に関する包括的グレブナ基底系と呼ぶ. また, 各代数構成的集合 S_i をパラメータ空間 L^m の**セル (cell)** と呼び, 各ペア (S_i, G_i) を**断片**と言う.

例 7

パラメターを a, b, 変数を x, y とする. $F = \{ax^2y + y, bx^2y^2 + ax + y\} \subset \mathbb{Q}[a, b][x, y]$ から生成させれるイデアルの辞書式順序 $x \succ y$ に関する包括的グレブナ基底系は

$$\left\{\left(\mathbb{Q}^2 \backslash \mathbb{V}(a, b), \{a^2x - by^2 + ay, -b^2y^5 + 2bay^4 - a^2y^3 - a^3y\}\right), \left(\mathbb{V}(a, b), \{y\}\right)\right\}$$

である. これは次を意味する.

$$\begin{cases} \text{もし } a \neq 0, b \neq 0 \text{ ならば}, & \{a^2x - by^2 + ay, -b^2y^5 + 2bay^4 - a^2y^3 - a^3y\}, \\ \text{もし } a = b = 0 \text{ ならば}, & \{y\}. \end{cases}$$

本論文において, 代数構成的集合をアフェイン多様体を用いて $\mathbb{V}(f_1, .., f_k) \backslash \mathbb{V}(g_1, .., g_l) \subseteq L^m$ の形で表す. ここで, $f_1, .., f_k, g_1, .., g_l \in K[\bar{A}]$ である. また, 本論文において LCM (Least Common Multiple) は多項式の最小公倍元を求めるものと仮定する.

次の定理は定理 5 から直接に従い, Suzuki-Sato の包括的グレブナ基底系を求めるアルゴリズムを構成するための主たる定理である. どのようなときにグレブナ基底が安定しているかを考えることがミソである.

定理 8

$F \subset K[\bar{A}][\bar{X}]$, $S \subset K[\bar{A}]$ とし, $G \subset K[\bar{A}][\bar{X}]$ をある項順序 $\succ$ に関しての $\langle F \cup S \rangle \subseteq K[\bar{A}][\bar{X}]$ のグレブナ基底とする. $B := \{b : b \in \langle S \rangle, b \in G\}$, $\{h_1, \ldots, h_s\} := \{\mathrm{lc}(g) : g \in G \backslash B\}$ とし $h := \mathsf{LCM}(h_1, \ldots, h_s)$ とする. そのとき, 任意の $\bar{a} \in \mathbb{V}(S) \backslash \mathbb{V}(h)$ において $\sigma_{\bar{a}}(G)$ は $L[\bar{X}]$ で項順序 $\succ$ において $\langle \sigma_{\bar{a}}(F) \rangle$ のグレブナ基底である. (このとき実際 $\sigma_{\bar{a}}(G) = \sigma_{\bar{a}}(G \backslash B)$ である.)

この定理より, 包括的グレブナ基底系を求めるアルゴリズムを以下のように構成することができる. [SS06].

アルゴリズム 9 (Suzuki-Sato($F, \succ$)**[SS06]**)

Input F: $K[\bar{A}][\bar{X}]$ の有限部分集合, $\succ$:pp($\bar{X}$) の項順序,

Output G: L^m 上の $\langle F \rangle$ の $\succ$ に関する包括的グレブナ基底系,

begin

$G \leftarrow \mathsf{CGSMain}(F, \emptyset, \succ)$; return($G$)

end

アルゴリズム 10 (CGSMain($F, Z, \succ$))

Input F:$K[\bar{A}][\bar{X}]$ の有限部分集合, Z : $K[\bar{A}]$ の有限集合, $\succ$: pp($\bar{X}$) 上の項順序,

Output H: $\mathbb{V}(Z)$ 上の $\langle F \rangle$ の $\succ$ に関する包括的グレブナ基底系.

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

注意: 多くの最適化のテクニックを使って良い形の包括的グレブナ基底系を求めることができ，それらのテクニックでこのアルゴリズムを改良することは可能である．例えば $(**)$ において全ての元を既約元に分解することもその一つである．ここでは詳しく述べない．（参照 [SS06].）

4 新しいアプローチ

ここでは，新しいテクニックを使って Suzuki-Sato アルゴリズムを改良する．まず，新しいテクニックを紹介する前に，モチベーションから話を始める．

4.1 モチベーション

項順序を固定し，F を $K[\bar{A}][\bar{X}]$ の部分集合とする．その時，アルゴリズム GröbnerBasis より $K[\bar{A}][\bar{X}]$ 上で $\langle F \rangle$ のグレブナ基底 $G = \{g_1, .., g_l\}$ を計算することができる．このとき，多項式環の係数ドメインが多項式環なので，このグレブナ基底 G はよく次のような性質を持つ．

$(\diamond 1)$

$g_i \neq g_j$ なる $g_i, g_j \in G$ では，$\mathrm{lpp}_{\bar{X}}(g_i) | \mathrm{lpp}_{\bar{X}}(g_j)$ となることがある．

もし，体を係数ドメインとする多項式環で簡約グレブナ基底を考えるならば，このようなことは起こらない．実際，特化準同型 $K[\bar{A}][\bar{X}] \to L[\bar{X}]$ を考えるので，$K[\bar{A}][\bar{X}]$ 上のグレブナ基底と $L[\bar{X}]$ 上のグレブナ基底のギャップを埋めることが新しいテクニックでの戦略である．すなわち，上の性質をできるだけ避け，$L[\bar{X}]$ 上のグレブナ基底の性質に近づけることである．これにより，包括的グレブナ基底系のパラメータ空間のセルの数，不必要なセルを小さくすることができる．

新しいテクニックを述べる前に，Suzuki-Sato アルゴリズムの振る舞いと新しいテクニックのアイデアについて考える．

Suzuki-Sato アルゴリズムの第一ステップは次の Figure 1 のようになる．つまり，$\langle F \rangle$ の包括的グレブナ基底系を計算するためにまずは l 個の場合 $\mathrm{lc}_{\bar{X}}(g_1) = 0, \ldots, \mathrm{lc}_{\bar{X}}(g_l) = 0$ を考えなければならない．

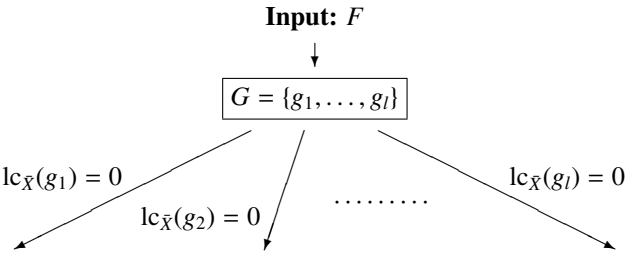

Figure 1:

このステップではグレブナ基底 G の各先頭係数がゼロの場合を考える．第一ステップのみならずすべてのステップにおいてこの手順を繰り返す．ここで，Suzuki-Sato アルゴリズムはゼロ (= 0) の場合は考察するが，先頭係数がゼロでない (≠ 0) は考察しない．各ステップの枝はゼロ (= 0) の場合のみである．だから，このアルゴリズムは簡単な構造である．しかしながら，上の性質 (◇1) を避けるために枝としてゼロでない (≠ 0) を考えると，オリジナルの Suzuki-Sato アルゴリズムから出力される木より小さな木を得ることができる．すなわち，枝としてゼロでない (≠ 0) を使うと効率的に計算できる．さて，どのように，そしてどのような場合，“枝としてゼロでない (≠ 0)” を使うか？

まず，簡単なものとして，もし集合 G の元として $\mathrm{lpp}_{\bar{X}}(g_i) = 1$ なる g_i が存在したときを考える．そのとき，各 l 個の多項式の先頭係数が 0 の場合の l 個の場合 (枝) を考える必要はない．なぜならば，$\mathrm{lc}_{\bar{X}}(g_i) \neq 0$ のとき，1 はすべての多項式の項 (先頭項とそれ以外のすべての項) を割るから，明らかにグレブナ基底は $\{1\}$ になる．よって他の多項式の係数が 0 になろうが 0 でなかろうが関係ない．この場合は 1 個の場合 $\mathrm{lc}_{\bar{X}}(g_i) = 0$ を考えるだけでよい．すなわち，$\mathrm{lc}_{\bar{X}}(g_i) \neq 0$ の場合 (枝) を考えることによって多くの場合 (枝) が削除されることが分かる．この考えを一般的なものへと拡張する．ここで，各 $p \in G$ において，集合 $G_p := \{g \in G \backslash \{p\} : \mathrm{lpp}_{\bar{X}}(p) | \mathrm{lpp}_{\bar{X}}(g)\}$ を定義する．もし G_p が空集合なら，各 $\bar{a} \in L^m \backslash \mathbb{V}(\mathrm{lc}_{\bar{X}}(p))$ において，すべての $\mathrm{lpp}_{\bar{X}}(G_p)$ の元は $\sigma_{\bar{a}}(\mathrm{lpp}_{\bar{A}}(p))$ によって簡約 (リダクション) される．したがって，$g_{pi} \in G_p$ で $\mathrm{lc}_{\bar{X}}(g_{pi}) = 0$ の場合は考える必要はない．すなわち，構成される木の枝の数が少なくすることができ，このテクニックを使うことで Suzuki-Sato アルゴリズムをより効率よく改良できると想像できる．もし，G_p が空集合なら，そのときはこのテクニックは使えないので Suzuki-Sato アルゴリズムの手順に従う．ここでこのアイデアについで具体的な例を見る．a, b, c をパラメータとし x を変数とする．多項式の集合 $F = \{ax^3, bx^2, cx\} \in \mathbb{Q}[a, b, c][x]$ の包括的グレブナ基底系を求める．まず，この場合 Suzuki-Sato アルゴリズムがどのように働くかを Figure 2 で見る．

Suzuki-Sato アルゴリズムは Figure 2 の木構造の手順をとる．この図での各ボックスは $\langle F \rangle$ の包括的グレブナ基底系の断片である．したがって, この図のすべてのボックスの集合 $\left\{\boxed{1}, \boxed{2}, \ldots, \boxed{16}\right\}$ が $\langle F \rangle$ の包括的グレブナ基底系である．もちろん，より良い包括的グレブナ基底系を得るために色々な最適化のテクニックを使うことができる．しかしながら，基本的には Suzuki-Sato アルゴリズムは Figure 2 のように動く．いくつかの最適化のテクニックが組み込まれた Suzuki-Sato

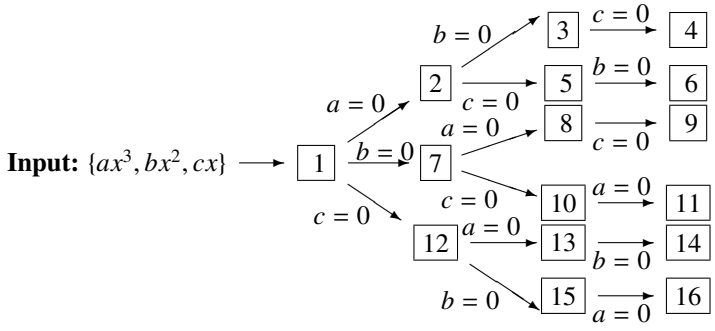

Figure 2:

アルゴリズムは数式処理ソフト Risa/Asir[NT92] 上で実装されており (参照 [SS06])，そのプログラムは $\langle F \rangle$ の包括的グレブナ基底系として次を出力する．

$$\left\{ \begin{array}{l} (\mathbb{V}(a)\backslash\mathbb{V}(cb), \{x\}), (\mathbb{V}(a,b)\backslash\mathbb{V}(c), \{x\}), (\mathbb{V}(b)\backslash\mathbb{V}(ac), \{x\}), (\mathbb{V}(a,c)\backslash\mathbb{V}(b), \{x^2\}), \\ (\mathbb{V}(a,b,c), \{\emptyset\}), (\mathbb{V}(c,b)\backslash\mathbb{V}(a), \{x^3\}), (\mathbb{V}(c)\backslash\mathbb{V}(ab), \{x^2\}), (\mathbb{C}^3\backslash\mathbb{V}(abc), \{x\}) \end{array} \right\}.$$

この出力は 8 個の断片を持つ．

次に新しいアイデアを使ってこの問題を考えてみる．i.e.，枝として “$\neq 0$” を持った木を構成する．まず，$\mathbb{Q}[a,b,c][x]$ 上で $\langle F \rangle$ のグレブナ基底 $S1$ を計算する．このとき，$S1 = \{ax^3, bx^2, cx\}$ である．ここで各先頭項は $\mathrm{lpp}_{\{a,b,c\}}(cx) = x$，$\mathrm{lpp}_{\{a,b,c\}}(bx^2) = x^2$，$\mathrm{lpp}_{\{a,b,c\}}(ax^3) = x^3$ より，明らかに $x|x^2$ そして $x|x^3$ である．もし，$\mathrm{lc}_{\{a,b,c\}}(cx) = c \neq 0$ なら，そのとき明らかに $\langle F \rangle$ のグレブナ基底は $\{x\}$ である．このとき生成される包括的グレブナ基底系の断片のパラメータ空間は $L^3\backslash\mathbb{V}(c)$ であり，このパラメータ空間は全パラメータ空間 L^3 を覆ってないので次に $c = 0$ の場合 (枝) を考えなくてはいけない．$c = 0$ の場合，$S2 = \{ax^3, bx^2\}$ のグレブナ基底として $\mathbb{Q}[a,b,c][x]$ 上では $S2$ 自身を得る．このとき，明らかに $\mathrm{lpp}_{\{a,b,c\}}(bx^2) | \mathrm{lpp}_{\{a,b,c\}}(ax^3)$ より，$c = 0$ そして $b \neq 0$ の場合，$\langle F \rangle$ のグレブナ基底は $\{x^2\}$ である．最後に $c = 0, b = 0, a \neq 0$ と $c = b = a = 0$ の場合を考える新しいアイデアでの手順を終了し包括的グレブナ基底系を出力する．このときの手順によって得られた木構造は Figure 3 である．

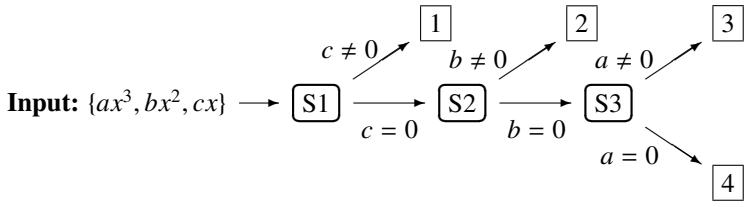

Figure 3:

この新しいテクニックによって得られた $\langle F\rangle$ の包括的グレブナ基底系は次のようになる．

$$\left\{ \begin{array}{l} \boxed{1} = (\mathbb{C}^3\backslash\mathbb{V}(c), \{x\}), \boxed{2} = (\mathbb{V}(c)\backslash\mathbb{V}(b), \{x^2\}), \\ \boxed{3} = (\mathbb{V}(b,c)\backslash\mathbb{V}(a), \{x^3\}), \boxed{4} = (\mathbb{V}(a,b,c), \{\emptyset\}) \end{array} \right\}.$$

この包括的グレブナ基底系は 4 個の断片を持つ．
前に見たように，新しいアイデアでのプロセス Figure 3 は Suzuki-Sato アルゴリズムのプロセス Figure 2 よりも簡単なプロセスとなっていることが分かる．また，さらにそれぞれの出力を比べると新しいアイデアから得られた包括的グレブナ基底系の断片の数は 4 個で Suzuki-Sato アルゴリズムにより得られた断片の数 8 個より少なくより良い形となっている．このことから，新しいアイデアは Suzuki-Sato アルゴリズムよりもっと効率的であり出力の形も良くなると思われる．

次の節ではこの新しいアイデアについて厳密に述べ，このアイデアを使った包括的グレブナ基底系を計算するための新しいアルゴリズムを与える．

4.2 新しいアルゴリズム

本節では包括的グレブナ基底系を計算するための新しいアルゴリズムを与える．新しいアルゴリズムを構成するための主アイデアとなる定理が次である．

定理 11
F を $K[\bar{A}][\bar{X}]$ の部分集合とし，$H = \{g, g_1, \ldots, g_l\}$ を項順序 $\succ$ に関しての $\langle F\rangle$ のグレブナ基底とする．g を H から選び，$r := \frac{1}{\mathrm{lc}_{\bar{X}}(g)}$ と r をセットし (r を新しい変数と見る)，$g' := \mathrm{lpp}_{\bar{X}}(g) + r\cdot(g - \mathrm{lm}_{\bar{X}}(g))$ とする．今，$H' := (H\backslash\{g\})\cup\{g'\} = \{g', g_1, \ldots, g_l\} \subseteq K[r, \bar{A}][\bar{X}]$ と定義し，$K[r, \bar{A}][\bar{X}]$ 上でその H' から生成されるイデアルの $\succ$ に関するグレブナ基底を G' とする．さらに，$G := \{f \in K[\bar{A}][\bar{X}] : f \neq 0, f = \mathrm{lc}_{\bar{X}}(g)^k \cdot \sigma_{r=\frac{1}{\mathrm{lc}_{\bar{X}}(g)}}(q), \deg_r(q) = k \in \mathbb{N}, q \in G'\}$ とし，$\{h_1, \ldots, h_e\} := \{\mathrm{lc}_{\bar{X}}(f) \in K[\bar{A}] : f \in G\}$ とする．
その時，各 $\bar{a} \in L^m\backslash(\mathbb{V}(\mathrm{lc}_{\bar{X}}(g)) \cup \mathbb{V}(h))$ で，$\sigma_{\bar{a}}(G)$ は $\langle\sigma_{\bar{a}}(F)\rangle$ の $\succ$ に関するグレブナ基底である．ここで，$h = \mathsf{LCM}(h_1, \cdots, h_e)$ とし，$\sigma_{r=\frac{1}{\mathrm{lc}_{\bar{X}}(g)}}(q)$ は変数 r に $\frac{1}{\mathrm{lc}_{\bar{X}}(g)}$ を代入することを意味する．

証明　各 $\bar{a} = (a_1, \ldots, a_m) \in L^m\backslash(\mathbb{V}(\mathrm{lc}_{\bar{X}}(g)) \cup \mathbb{V}(h))$ で，$\sigma_{\bar{a}}(\mathrm{lc}_{\bar{A}}(g)) \neq 0$，$\sigma_{\bar{a}}(r) \neq 0$ となる．いま $\bar{b} := (a_1, \ldots, a_m, \frac{1}{\sigma_{\bar{a}}(\mathrm{lc}(g))}) \in L^{m+1}$ を考える．G' の定義より，各 $p \in G'$ において，$\sigma_{\bar{b}}(\mathrm{lm}_{\bar{X}}(p)) \neq 0$ である．よって，定理 8 より，$\sigma_{\bar{b}}(G')$ は $\succ$ に関して $\langle\sigma_{\bar{b}}(H')\rangle$ のグレブナ基底である．実際，$\langle\sigma_{\bar{b}}(G')\rangle = \langle\sigma_{\bar{a}}(G)\rangle$ である．したがって，$\sigma_{\bar{a}}(G)$ も $\succ$ に関して $\langle\sigma_{\bar{b}}(H')\rangle$ のグレブナ基底である．$\sigma_{\bar{a}}(g_i) = \sigma_{\bar{b}}(g_i)$ と各 $1 \leq i \leq l$ において $\langle\sigma_{\bar{a}}(g)\rangle = \langle\sigma_{\bar{b}}(g')\rangle$ なので，$\langle\sigma_{\bar{b}}(H')\rangle = \langle\sigma_{\bar{a}}(H)\rangle$ が成り立つ．$\sigma_{\bar{a}}$ は環準同型写像なので，明らかに $\langle\sigma_{\bar{a}}(H)\rangle = \langle\sigma_{\bar{a}}(F)\rangle$ となる．以上より，$\sigma_{\bar{a}}(G)$ は $\succ$ に関して $\langle\sigma_{\bar{a}}(F)\rangle$ のグレブナ基底である．∎

次の系は定理 11 から直接従う結果であり，定理 11 の一般化である．この系を使うことで包括的グレブナ基底系を計算する新しいアルゴリズムを Suzuki-Sato アルゴリズムと同様に再帰的構造となるように構成できる．

系 12

F を $K[\bar{A}][\bar{X}]$ の部分集合とし，Z_1，Z_2 を $K[\bar{A}]$ の部分集合で $\langle Z_1\rangle \not\subset \langle Z_2\rangle$ とする．$H = \{g, g_1, \ldots, g_l\}$ を項順序 $\succ$ に関しての $\langle F \cup Z_1\rangle$ のグレブナ基底とする．g を H から選び，$r := \frac{1}{\mathrm{lc}_{\bar{X}}(g)}$ と r をセットし (r を新しい変数と見る)，$g' := \mathrm{lpp}_{\bar{X}}(g) + r \cdot (g - \mathrm{lm}_{\bar{X}}(g))$ とする．今，$H' := (H \backslash \{g\}) \cup \{g'\} = \{g', g_1, \ldots, g_l\} \subseteq K[r, \bar{A}][\bar{X}]$ と定義し，$K[r, \bar{A}][\bar{X}]$ 上でその H' から生成されるイデアルの $\succ$ に関するグレブナ基底を G' とする．さらに，$G := \{f \in K[\bar{A}][\bar{X}] : f \neq 0, f = \mathrm{lc}_{\bar{X}}(g)^k \cdot \sigma_{r=\frac{1}{\mathrm{lc}_{\bar{X}}(g)}}(q), \deg_r(q) = k \in \mathbb{N}, q \in G'\}$ とし，$\{h_1, \ldots, h_e\} := \{\mathrm{lc}_{\bar{X}}(f) \in K[\bar{A}] : f \in G\}$ とする．

その時，各 $\bar{a} \in \mathbb{V}(Z_1) \backslash (\mathbb{V}(Z_2) \cup \mathbb{V}(\mathrm{lc}_{\bar{X}}(g)) \cup \mathbb{V}(h))$ で，$\sigma_{\bar{a}}(G)$ は $\langle \sigma_{\bar{a}}(F)\rangle$ の $\succ$ に関するグレブナ基底である．ここで，$h = \mathsf{LCM}(h_1, \cdots, h_e)$ である．

ここで重要なことは，新しい変数 r を係数ドメインの多項式環上に導入したことである．これが，アイデアの一つで鍵である．この変数 r を使うことにで，多項式 g を**モニック**にする操作をしたとしても r によって g の先頭係数の情報はキープされる．これら定理 11，系 12 を使うことによって包括的グレブナ基底系計算が “どのようになるか？”，“どのように効率的か？” を次の例で見る．

例 13

$F = \{xy + x, ax^2 + y + 2, bxy + y\}$ を $\mathbb{Q}[a, b][x, y]$ 部分集合とし，a, b をパラメータ，x, y を変数とすし，$\succ$ を $x \succ y$ となる辞書式順序とする．このとき $\succ$ に関しての $\langle F\rangle$ の包括的グレブナ基底系を求める．

(1) まず最初にアルゴリズム GröbnerBasis によって $\succ$ に関しての $\langle F\rangle$ のグレブナ基底を計算する．ここでアルゴリズム GröbnerBasis$(F, \succ)$ はグレブナ基底として $\{a+b^2, y+1, bx+1, ax-b\}$ を出力する．このとき明らかに，各 $\alpha \in \mathbb{C}^2 \backslash \mathbb{V}(a + b^2)$ において，$\{1\}$ は $\succ$ に関して $\langle \sigma_\alpha(F)\rangle$ のグレブナ基底である．したがって，ここで包括的グレブナ基底系の一つの断片 $(\mathbb{C}^2 \backslash \mathbb{V}(a + b^2), \{1\})$ を得る．

(2) 次に，$\{a + b^2 = 0\}$ の場合を考えなければいけない．ここで，定理 8 より断片 $(\mathbb{V}(a + b^2) \backslash \mathbb{V}(ab), \{y + 1, bx + 1, ax - b\})$ を得ることができる．しかし，この手順は Suzuki-Sato アルゴリズムと同じである．今，定理 11 と系 12 を使っての包括的グレブナ基底系の計算方法を考察しているのでここでは定理 8 を使った手順を適用しないようにする．定理 11 と系 12 を使うので，まず集合 $\{y + 1, bx + 1, ax - b\}$ から多項式を一つ選ばなければいけない．さて，

包括的グレブナ基底系を “効率” よく計算するためにはどの多項式を選べばよいか？

この問題は効率化そして計算速度と言う面で重要である．ここで $\mathrm{lpp}_{\{x,y\}}(bx+1)$ は $\mathrm{lpp}_{\{x,y\}}(ax-b)$ を割り，そして $\mathrm{lpp}_{\{x,y\}}(ax-b)$ は $\mathrm{lpp}_{\{x,y\}}(bx+1)$ を割ることが分かる．選ばれた多項式はモニックにされ，そしてまたグレブナ基底を計算することからモニックにされた多項式が $\mathbb{Q}[a, b][x, y]$ 上で他の多項式をリダクションすることができれば先頭項の集合の元の数を減らすことができると思われる．なので，ここでは $ax - b$ もしくは $bx + 1$ を選んだ方が効率的に計算できると思われる．ここでは $ax - b$ を選ぶとする．定理 11 に従うとまず $ax - b$ をモニックの多項式 $x - br$

へと変形する．ここで r は新しい変数で実際 $r := \frac{1}{a}$ である．すなわち，ここでは $a \neq 0$ を仮定している．

(3) 今，$\{a + b^2 = 0, a \neq 0\}$ の場合を考えている．$\succ$ に関しての $\langle a + b^2, y + 1, bx + 1, x - br\rangle$ のグレブナ基底を $\mathbb{Q}[a, b, r][x, y]$ 上で計算する．ここで，アルゴリズム GröbnerBasis は $\{-ar + 1, a + b^2, y + 1, x - br\}$ を出力する．今は $\{a + b^2 = 0, a \neq 0\}$ で $r = \frac{1}{a}$ の場合を考えているので $-ar+1, a+b^2$ は必要ない．よって定理 11 より，各 $\alpha \in \mathbb{V}(a+b^2)\backslash\mathbb{V}(a)$ において $\sigma_\alpha(\{ax-b, y+1\})$ は $\mathbb{C}[x, y]$ 上で $\langle\sigma_\alpha(F)\rangle$ のグレブナ基底である．すなわち，包括的グレブナ基底系の一つのセルは $(\mathbb{V}(a + b^2)\backslash\mathbb{V}(a), \{ax - b, y + 1\})$ である．

(4) 最後に，$\{a+b^2 = 0, a = 0\}$ の場合を考える．この場合明らかにグレブナ基底は $\{1\}$ である．したがって，$\succ$ に関して $\langle F\rangle$ の包括的グレブナ基底は $\{(\mathbb{C}^2\backslash\mathbb{V}(a+b^2), \{1\}), (\mathbb{V}(a+b^2)\backslash\mathbb{V}(a), \{ax - b, y + 1\}), (\mathbb{V}(a, b), \{1\})\}$ である．これは以下を意味する．

$$\left\{ \begin{array}{l} (\mathbb{C}^3\backslash\mathbb{V}(a + b^2), \{1\}), (\mathbb{V}(a + b^2)\backslash\mathbb{V}(a), \{ax - b, y + 1\}), \\ (\mathbb{V}(a, b), \{1\}) \end{array} \right\}.$$

この操作を図示したものが Figure 4 である．

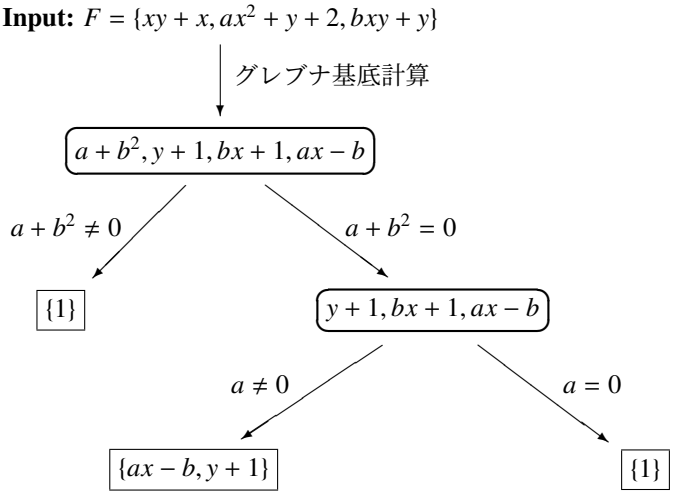

Figure 4:

G を $K[\bar{A}][\bar{X}]$ 上のイデアルのグレブナ基底とし，集合 E を以下のように定義する．

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

包括的グレブナ基底系の計算のため定理 11 を適用するならば，G から一つの多項式を選択する必要がある．そこで，次の問題がある．

効率的に計算するためにはどのような多項式を選ぶべきか？

例 13 (2) において，$\mathrm{lpp}_{\{x,y\}}(ax-b)$ は $\mathrm{lpp}_{\{x,y\}}(bx+1)$ を割るので $ax-b$ を選んだ．この例の場合 $E=\{ax-b, bx+1\}$ となる．もし E が空集合なら，そのとき定理 11 では，各 $\bar{a}\in L^m\backslash(\mathbb{V}(\mathrm{lc}_{\bar{X}}(g))\cup\mathbb{V}(h))=L^m\backslash\mathbb{V}(h)$ においていつも $\mathrm{lpp}(\sigma_{\bar{a}}(H))=\mathrm{lpp}(\sigma_{\bar{a}}(H'))$ となる．(この場合，明らかに $\sigma_{\bar{a}}(H)$ と $\sigma_{\bar{a}}(H')$ は $L[\bar{X}]$ 上で $\succ$ に関して $\langle F\rangle$ のグレブナ基底である．) この場合では，$\mathbb{V}(\mathrm{lc}_{\bar{X}}(H))=\mathbb{V}(\mathrm{lc}_{\bar{X}}(H')\cup\{\mathrm{lc}_{\bar{X}}(g)\})$ となるので，定理 11 を適用すべきでない．すなわち，パラメータ空間のセルの数は任意に選ばれた多項式によって変化しないので，定理 11 を適用すると余分な計算が増えるのでこの場合は Suzuki-Sato の定理 8 を適用したほうが良い．**もし E が空集合でなければ新しいアプローチとして定理 11 を適用する．このときこの定理はパワフルに働く．**このことより，問題の答えは **"E から一つの元を選ぶこと"** と言うことができる．(実際，E が空集合でない場合はよく起こる．) 新しいアルゴリズムでは一般的なグレブナ基底計算の normal strategy のように，集合 E を求めたあとその中で定められた項順序に関して最小の先頭項を持つものを選ぶ．(ここでの戦略はいろいろ考えられるがここでは述べず，次の章で考えるようにする．)

アルゴリズムの後に書く**注意**においてなぜアルゴリズム NEW において自然数 U が必要なのかを述べる．

アルゴリズム 14 (NEW($F, U, \succ$))
Input F: $K[\bar{A}][\bar{X}]$ の有限集合，$\succ$: $\mathrm{pp}(\bar{X})$ の項順序，U: 自然数 ($<\infty$),
Output G: $\succ$ に関する L^m 上 $\langle F\rangle$ の包括的グレブナ基底系.
begin
  $G\leftarrow$ NewCGSMain($F, \emptyset, \emptyset, 1, 0, \succ, U$)
  return(G)
end

アルゴリズム 15 (NewCGSMain ($F, L_1, L_2, D, N, \succ, U$))
Input F: $K[r,\bar{A}][\bar{X}]$ の有限集合，　L_1: $K[\bar{A}]$ の有限集合 (= 0)，L_2: $K[\bar{A}]$ の有限集合 ($\neq 0$),
D: $K[\bar{A}]$ の多項式，　U: 自然数 ($<\infty$)，　$\succ$: $\mathrm{pp}(\bar{X})$ の項順序，　N: 自然数 ($<U$),
Output H: $\succ$ に関する $\mathbb{V}(L_1)\backslash\mathbb{V}(L_2)$ 上 $\langle F\rangle$ の包括的グレブナ基底系.
begin
1: $G\leftarrow$ GröbnerBasisB($F\cup L_1, \succ$) in $K[r,\bar{A}][\bar{X}]$
2: $G^*\leftarrow$ Transform(G, D)
3: $G_1\leftarrow G^*\backslash\{g : g\in G^*\cap K[\bar{A}], g\in\langle L_1\rangle\}$
4: $E\leftarrow\{f\in G_1 : \exists g\in G_1\backslash\{f\}$ s.t. $\mathrm{lpp}_{\bar{X}}(f)|\,\mathrm{lpp}_{\bar{X}}(g)\}$
5: **if** $E\neq\emptyset$ and $N\leq U$ **then**
6: E から q を取る．このとき，$\mathrm{lpp}_{\bar{X}}(q)$ は $\succ$ に関して $\mathrm{lpp}_{\bar{X}}(E)$ で最少である．
    ($r:=\mathrm{lc}_{\bar{X}}(q)^{-1}$, i.e., r は新しい変数)
7: $q^*\leftarrow\mathrm{lpp}_{\bar{X}}(q)+r\cdot(q-\mathrm{lm}_{\bar{X}}(q))$ (i.e., $\mathrm{lc}_{\bar{X}}(q^*)=1$)
8: $F^*\leftarrow(G_1\backslash\{q\})\cup\{q^*\}$
9: $\{t_1,\ldots,t_k\}\leftarrow$ factorize($\mathrm{lc}_{\bar{X}}(q)$)
10: $t\leftarrow t_1\cdot t_2\cdots t_k$

11: **if** $\mathbb{V}(L_1)\backslash(\mathbb{V}(t) \cup \bigcup_{s\in L_2} \mathbb{V}(s)) \neq \emptyset$ **then** (♣1)
12: $N \leftarrow N + 1$
13: $H_1 \leftarrow \mathsf{NewCGSMain}(F^*, L_1, L_2 \cup \{t\}, \mathrm{lc}_{\bar{X}}(q), N, \succ, U)$
14: **end-if**
15:$H_2 \leftarrow \mathsf{NewCGSMain}(G_1, L_1 \cup \{t_1\}, L_2, \emptyset, 0, \succ, U) \cup \cdots$
$\cdots \cup \mathsf{NewCGSMain}(G_1, L_1 \cup \{t_k\}, L_2, \emptyset, 0, \succ, U)$
16: $H \leftarrow H_1 \cup H_2$
17: **else**
18: $S \leftarrow \{h_1, .., h_l\} := \{f : \mathbb{V}(f) \not\subset \bigcup_{s\in L_2} \mathbb{V}(s), f \in \mathsf{factorize}(\mathrm{lc}_{\bar{X}}(g)), \mathrm{lc}_{\bar{X}}(g) \notin K, g \in G_1\}$ (♣2)
19: $h \leftarrow \mathsf{LCM}(h_1, \ldots, h_l)$
20: $H \leftarrow \{(L_1, \{h\}, G_1)\}$
21: **if** $S \neq \emptyset$ **then**
22: **while** $S \neq \emptyset$ **do**
23: S から p を取る; $S \leftarrow S\backslash\{p\}$
24: $H \leftarrow H \cup \mathsf{NewCGSMain}(G_1, L_1 \cup \{p\}, L_2, \emptyset, 0, \succ, U)$
25: **end-while**
26: **else**
27: $H \leftarrow \{(L_1, L_2, G_1)\}$
28: **end-if**
29: **end-if**
30: return(H)
end

アルゴリズム 16 (Transform(F, D))
Input F :$K[r, \bar{A}][\bar{X}]$ の有限部分集合, D:$K[\bar{A}]$ の多項式,
Output $K[\bar{A}][\bar{X}]$ の有限部分集合.
• F に, $r = \frac{1}{D}$ として代入し変数 r を消去する. そしてすべてが多項式の形を取るように分母を払う.

注意 : アルゴリズムの (♣1) と (♣2) において, 記号 $\bigcup$ を使っている. $h_1, h_2 \in K[\bar{A}]$ で, 明らかに $\mathbb{V}(h_1) \cup \mathbb{V}(h_2) = \mathbb{V}(\mathsf{LCM}(h_1, h_2))$ であるので, $L_2 = \{s_1, \ldots, s_l\}$ としたとき, 記号 $\mathbb{V}(\mathsf{LCM}(s_1, \ldots, s_l))$ は $\bigcup_{s\in L_2} \mathbb{V}(s)$ の変わりに使うことができる. ここでは定理 11 において記号 "∪" を使ったのでこの記号を使った.
定理 11 と系 12 において, 集合 F を次のように変形する必要がある.
(1) $\langle F\rangle$ のグレブナ基底 H を計算する. (F から H への変形) (1 行目)
(2) 新しい変数 r によって H を H' へ変形する. (7 行目)
(3) $\langle H'\rangle$ のグレブナ基底 G' を計算する. (H' から G' への変形) (1 行目)
(4) G' を G へ変形する. (2 行目)
このアルゴリズムでは "$\neq 0$" の場合の枝の深さを自然数 U で制限している. 多くの場合この制

限は必要ない．しかしながら，(1) から (4) の式の変形を何度も繰り返していると**まれ**に同じものが出現し無限ループ（1 行から 14 行）に陥る場合が存在する．この無限ループを解消するために自然数 U を導入した．定理 8 と定理 11 と系 12 によって U が有限な自然数であれば何であっても出力は包括的グレブナ基底系である．詳しいことは定理 17 の証明で議論する．

論文 [SS06] においてより良い包括的グレブナ基底を計算するためにいろいろな最適化のテクニックが述べられているように，アルゴリズム NEW でも同じようなテクニックを使うことができる．これらのテクニックについては [SS06] を参照.

定理 17
アルゴリズム NEW$(F, U, \succ)$ は終了し，$\succ$ に関して $\langle F\rangle$ の包括的グレブナ基底系を出力する．

証明　まず最初にアルゴリズムの停止性についてみる．アルゴリズム NewCGSMain$(F, L_1, L_2, D, N, \succ, U)$ の停止性を証明する．アルゴリズム 15 での鍵となる部分は 5 行目である．

(∗1) もし $E = \emptyset$ で $N \leq U$ ならば，18–29 行目を考える．実際，この場合は Suzuki-Sato のアプローチである．この場合，アルゴリズムは 1 個の断片を生成する．（19 行目を見よ．）
(∗2) もし $E \neq \emptyset$ で $N \leq U$ ならば，6–16 行目を考える必要がある．この場合，アルゴリズムは何も生成しない．

アルゴリズム NewCGSMain は再帰的なアルゴリズムで木構造を作る．この木構造から任意のパスをとる．このパスが有限な深さを持つことを証明すればアルゴリズムが停止することが証明される．すなわち，17–28 行 (∗1) と 6–15 行 (∗2) がこのパスで有限回実行されることが言えれば良い．Suzuki-Sato アルゴリズムと同じ理由により (∗1) は有限回実行される（参照 [SS06]）．したがって，(∗2) が有限回実行されることを証明する必要がある．アルゴリズムの (注意) でも少し述べたように，もし自然数 U がなければ (∗2) の過程で何度も変形をするので無限ループに陥る可能性があり．しかし，U は有限な数より多くても U 回 (∗2) を “連続” で実行するだけで止まる．止まれば，次の実行は (∗1) となる．(∗1) は有限回しか実行されないことよりこのアルゴリズムは停止する．

次に，このアルゴリズムは包括的グレブナ基底系を出力することを証明しなければならない．この証明はほぼ Suzuki-Sato アルゴリズム [SS06] と同じである．注意する点は，新しいアルゴリズムの核となる定理 11 と系 12 を使っているところである．13 と 15 行目において，このアルゴリズムは $t = \mathrm{LCM}(t_1, \ldots, t_k) \neq 0$ と $t_1 = 0, \ldots, t_k = 0$ の場合を計算している．すなわち，$\bigcup_{i=1}^{k} \mathbb{V}(t_i) \cup (L^m \backslash \mathbb{V}(t)) = L^m$ である．この事実と Suzuki-Sato アルゴリズム (の証明) により，出力はいつも全てのパラメータ空間を覆う包括的グレブナ基底系である． ∎

著者はアルゴリズム NEW を計算機代数システム Risa/Asir に実装した．次の例で，このプログラムがどのような出力をするかを見る．ここで，NEW の入力として必要となる自然数 U を 5 とする．

例 18
$F = \{ax^2 + by^2, cx^2 + y^2, 2ax - 2cy\}$ を $\mathbb{Q}[a, b, c][x, y]$ の部分集合，a, b, c をパラメータ，x, y を変

数とする．ここで項順序として $x \succ y$ となる辞書式順序 $\succ$ を考える．このとき，$\langle F \rangle$ の包括的グレブナ基底系としてプログラムは次を出力する．

```
[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].
```

これは次を意味する．

$$\left\{ \begin{array}{l} (\mathbb{V}(a,b,c),\{y^2\}),(\mathbb{V}(b^2-+c,a-cb,ba+c^2)\backslash\mathbb{V}(a),\{ax-cy\}), \\ (\mathbb{V}(a,c)\backslash\mathbb{V}(b^2+c),\{y^2\}),(\mathbb{V}(a,cb)\backslash(\mathbb{V}(c)\cup\mathbb{V}(b^2-c)),\{cx^2,y\}), \\ (\mathbb{V}(a-cb)\backslash(\mathbb{V}(a)\cup\mathbb{V}(b^2-c)),\{ax-cy,y^2\}),(\mathbb{V}(a)\backslash(\mathbb{V}(c)\cup\mathbb{V}(a-cb)), \\ \{cx^2,y\}),(\mathbb{C}^3\backslash(\mathbb{V}(a)\cup\mathbb{V}(a-cb)),\{ax-cy,y^2\}) \end{array} \right\}.$$

この出力は7個の断片を持つ．著者はSuzuki-Satoアルゴリズムも計算機代数システムRisa/Asirに実装し，この例題を計算した．そのとき17個の断片を出力した．

4.3 他のテクニック

ここでは，より良い包括的グレブナ基底系を計算するためにいくつかの最適化のテクニックについて考える．Suzuki-Satoアルゴリズム，新しく紹介したアルゴリズムの両方の基礎となる理論はグレブナ基底の安定性である．すなわち，どのような場合にグレブナ基底が安定するかを考えればよい．これを考えたとき以下のような場合にも安定性を言うことができる．以下のような場合は特別であると思われるが，著者の経験上，計算を何度かしてみると案外見ることがある．

補題 19

F を $K[\bar{A}][\bar{X}]$ の部分集合，$\succ$ を $\mathrm{pp}(\bar{X})$ 上の項順序，G を $\succ$ に関しての $\langle F \rangle$ のグレブナ基底とする．ここで，$g_1,\ldots,g_l \in G$ において $\mathrm{Mono}_{\bar{X}}(g_i) \subset \langle \mathrm{lpp}_{\bar{X}}(G\backslash\{g_1,\ldots,g_l\})\rangle$ であるとする $(1 \le i \le l)$．また，$\{h_1,\ldots,h_s\} := \{\mathrm{lc}_{\bar{X}}(f) \in K[\bar{A}] : f \in G\backslash\{g_1,\ldots,g_l\}\}$ で $h := \mathrm{LCM}(h_1,\ldots,h_s)$ とする．
そのとき，任意の $\bar{a} \in L^m\backslash\mathbb{V}(h)$ において，

(1) $\sigma_{\bar{a}}$ と $\succ$ に関して，$\langle G \rangle$ は安定である，
(2) $L[\bar{X}]$ 上で $\sigma_{\bar{a}}(G\backslash\{g_1,\ldots,g_l\})$ は $\succ$ に関して $\langle \sigma_{\bar{a}}(F)\rangle$ のグレブナ基底である．

証明
仮定より $\mathrm{Mono}_{\bar{X}}(g_i) \subset \langle \mathrm{lpp}_{\bar{X}}(G\backslash\{g_1,\ldots,g_l\})\rangle$ であるので，$\mathrm{lm}_{\bar{X}}(\sigma_{\bar{a}}(g_i)) \in \langle \mathrm{lpp}_{\bar{X}}(G\backslash\{g_1,\ldots,g_l\})\rangle$ を得る．したがって，$\langle \sigma_{\bar{a}}(\mathrm{lm}_{\bar{X}}(G))\rangle = \langle \sigma_{\bar{a}}(\mathrm{lm}_{\bar{X}}(G\backslash\{g_1,\ldots,g_l\}))\rangle$．ここで，$\sigma_{\bar{a}}(g_i)$ は $\sigma_{\bar{a}}(G\backslash\{g_1,\ldots,g_l\})$ によって0に簡約される．定理5より，G は $\sigma_{\bar{a}}$ に関して安定であり，$\sigma_{\bar{a}}(G\backslash\{g_1,\ldots,g_l\})$ は $\succ$ に関して $L[\bar{X}]$ 上で $\langle \sigma_{\bar{a}}(F)\rangle$ のグレブナ基底である． ∎

この補題の簡単な例はもうすでに例 13 において見た．もし，$\mathrm{lpp}_{\bar{X}}(p) = 1$ となる $p \in G$ が存在したならば（記号は上の補題と同じ），その時，包括的グレブナ基底系の他の断片を計算するため $G\backslash\{p\}$ についての係数のチェック等を考える必要はない．なぜならば 1 はすべてを割るからである．もっと一般的な補題 19 の例を次で見る．

例 20

a, b をパラメータとして，x, y, z を変数とする．また，$F = \{axz + bxz + a, bz + a, (a^2 + a)xy\}$ を $\mathbb{Q}[a, b][x, y, z]$ の部分集合とする．アルゴリズム GröbnerBasis によって $\langle F\rangle$ の辞書式順序 $x \succ y \succ z$ のグレブナ基底 G を計算することができる．このグレブナ基底 G は以下である．

$$G = \left\{ \begin{array}{l} g_1 = bz + a,\ g_2 = (-a^2 - a)y,\ g_3 = (-a^2 - ab)x + ab, \\ g_4 = (az - a)x + a,\ g_5 = (b - 1)axy - aby \end{array} \right\}.$$

このとき，$g_5 \in \langle \mathrm{lpp}_{\{x,y\}}(g_1), \ldots, \mathrm{lpp}_{\{x,y\}}(g_4)\rangle$ である．補題 19 より，任意の $\alpha \in C^2\backslash\mathbb{V}(ab(a+1)(a+b))$ において，$\{\sigma_\alpha(g_1), \ldots, \sigma_\alpha(g_4)\}$ は $\mathbb{C}[x, y, z]$ 上で $\langle\sigma_\alpha(F)\rangle$ のグレブナ基底である．ここで，補題 19 を使うことで g_5 を除くことができる．したがって，$\{\mathrm{lc}_{\{x,y\}}(g_5) = b - 1 = 0\}$ の場合を考える必要がないので，包括的グレブナ基底系の断片の数を抑えることができる．

この補題のテクニックと似ているものに $K[\bar{A}][\bar{X}]$ 上での syzygy を用いた特別なリダクションを使うテクニックがある．このテクニックがどのように役に立つかを見るためにまず次の例を見る．

$\mathbb{Q}[a, b][x, y, z]$ の多項式の集合として $F = \{f_1 = ax + 1, f_2 = (b + 1)y, f_3 = az + bz + z\}$ を考える．今，$\succ$ を $\mathrm{pp}(x, y, z)$ 上の辞書式順序で $x \succ y \succ z$ とする．. $\langle F\rangle$ のグレブナ基底を計算するためアルゴリズム GröbnerBasis よりまず F を $\mathbb{Q}[a, b, x, y, z]$ の集合として考える．次に，ブロック項順序 $\succ_{\{x,y,z\},\{a,b\}}$ よりグレブナ基底を $\mathbb{Q}[a, b, x, y, z]$ 上で計算する．ここで，$\succ_{glex}$ を $\mathrm{pp}(a, b)$ 上の全次数辞書式順序とし $a \succ_{glex} b$ とする．その時，$\mathbb{Q}[a, b, x, y, z]$ でのブロック項順序 $\succ_{\{x,y,z\},\{a,b\}}$ に関して $\langle F\rangle$ のグレブナ基底は

$$G = \{g_1 = (a + b + 1)z, g_2 = (b + 1)y, g_3 = yz, g_4 = ax + 1, g_5 = (b + 1)xz - z\}$$

である．G は $\mathbb{Q}[a, b, x, y, z]$ 上で簡約グレブナ基底なので，$\forall g \in G$ は $G\backslash\{g\}$ によって簡約されない．G は $\mathbb{Q}[a, b, x, y, z]$ 上で簡約グレブナ基底なので，$\forall g \in G$ は $G\backslash\{g\}$ によって簡約されない．しかしながら，g_5 を見ると $\mathbb{Q}[a, b][x, y, z]$ 上で $\mathrm{lm}_{\{x,y,z\}}(g_5) \in \langle \mathrm{lm}_{\{x,y,z\}}(G\backslash\{g_5\})\rangle$ である．つまり，g_5 は $\mathbb{Q}[a, b][x, y, z]$ 上では $g_5 = x \cdot g_1 - z \cdot g_4$ と書かれる．すなわち，g_5 は $\mathbb{Q}[a, b][x, y, z]$ 上では，g_1 と g_4 によってゼロに簡約され，g_5 は冗長な多項式であることが分かる．**しかし，$G\backslash\{g_5\}$ はアルゴリズム GröbnerBasis によっては計算されない．**(参照 [Nab06]) そこで，次のようなリダクションを定義する．

定義 21 (リダクション [AL94])

2 つの多項式 f, h とゼロでない多項式の集合 $F = \{f_1, \ldots, f_s\} \subset K[\bar{A}][\bar{X}]$ が与えられたとき，f は F によって h へ**リダクション (reduction)** されるとは，$\mathrm{lc}_{\bar{X}}(f) = c_1 \mathrm{lc}_{\bar{X}}(f_1) + \cdots + c_s \mathrm{lc}_{\bar{X}}(f_s)$

となる $c_1, \ldots, c_s \in K[\bar{A}]$ と $\mathrm{lpp}_{\bar{X}}(f) = D_i \,\mathrm{lpp}_{\bar{X}}(f_i)$ となる $D_1, \ldots, D_s$ が存在するとき，$h = f - (c_1 D_1 f_1 + \cdots + c_s D_s f_s)$ となる．

このリダクションを使うことで上での g_5 は g_1 と g_4 によってゼロとなる．アルゴリズム GröbnerBasis の実行後は，不必要な元が存在する場合がある．この不必要な元を削除するためにこの特別なリダクションを使うことでより良い包括的グレブナ基底を計算することが可能である．

4.4 モノイデアル

ここでは単項から生成されるイデアルのパラメトリック・グレブナ基底について考える．まず，包括的グレブナ基底の定義を次で与える．

定義 22 (包括的グレブナ基底)
$\mathrm{pp}(\bar{X})$ 上の項順序を固定し，と G を $K[\bar{A}][\bar{X}]$ の部分集合とする．任意の $\bar{a} \in L^m$ において，もし $G \subset \langle F \rangle$ で $\sigma_{\bar{a}}(G)$ が $L[\bar{X}]$ 上で $\langle \sigma_{\bar{a}}(F) \rangle$ のグレブナ基底であるならば，G を $\langle F \rangle$ の**包括的グレブナ基底 (CGB)** と言う．

単項のみから生成されるイデアルのパラメトリック・グレブナ基底は一般の場合に比べれば簡単である．グレブナ基底の定義と安定性の定義から単項のみから生成されるパラメトリック・イデアルはいつもそれ自身がそれ自身の包括的グレブナ基底である．しかしながら，パラメータに値を代入してもそれは簡約グレブナ基底ではない．すべてのパラメータの場合において簡約グレブナ基底になるようなものを得るには，そのようになる包括的グレブナ基底系を計算する必要がある．この単項の場合では定理 11 はもっと簡単になる．まず，選ぶ多項式はすべて単項であることから変数 r は必要ない．また，いつも安定であることより選んだ単項の係数を 1 とし割ることのできる単項を割るだけでよく，グレブナ基底の計算は必要ない．この割り算を繰り返すことで簡約グレブナ基底になる包括的グレブナ基底系の断片を得ることをできる．アルゴリズム NEW の手順のようにすればすべてのパラメータの場合で簡約グレブナ基底となる包括的グレブナ基底系を簡単に計算することができる．（参照 4.1 章モチベーション）

5 戦略と実験

本章では，計算機代数システム Risa/Asir 上に実装された 2 つのアルゴリズム Suzuki-Sato, NEW を比較する．アルゴリズム NEW は定理 11 を適用しているのでグレブナ基底計算後そのグレブナ基底から 1 つ多項式を選ばなければならない．簡単のためにこの多項式を次のように定義する．

定義 23 (消去多項式)
$I \subseteq K[\bar{A}][\bar{X}]$ をイデアル，$\succ$ を $\mathrm{pp}(\bar{X})$ 上の項順序とする．さらに，G を $\succ$ に関する I のグレブナ基底とし，$E := \{f \in G : \mathrm{lpp}_{\bar{X}}(f) | \mathrm{lpp}_{\bar{X}}(g)$ となるような $g \in G$ が存在 $\}$ とする．このとき任意の $f \in E$ を G の**消去多項式**と言う．

この消去多項式の選択方法はいくつか考えられる．この選択方法によって出力される包括的グレブナ基底系，計算速度は大きく変わってくるので選択方法は重要である．アルゴリズム NEW

の “6 行目” では次の戦略 1 を適用した．この 6 行目で消去多項式の選択の戦略を変える事ができる．まず，次の戦略 1 を持つアルゴリズム NEW と Suzuki-Sato アルゴリズムを比較する．ここで比較に使った計算機は PC [CPU:Pentium M 1.73 GHz, Memory 1024 MB RAM, OS: Windows XP] ある．また，アルゴリズム NEW での自然数 U を 5 と固定した．

• **戦略 1** 項順序 $\succ$ に関して最小の消去多項式を E から選ぶ.

以下，a,b,c,d をパラメータとし，x,y,z,w を変数，項順序 $\succ$ を $x \succ y \succ z \succ w$ なる辞書式順序と固定する．次の $\mathbb{C}[a,b,c,d][x,y,z,w]$ の部分集合を考える．$F_1 = \{ax^4y + xy^2 + bx, x^3 + 2xy, bx^2 + x^2y\}$, $F_2 = \{ax^2 + by^2, cx^2 + y^2, 2ax - 2cy\}$, $F_3 = \{ax^4 + cx^2 + b, bx^3 + x^2 + 2, cx^2 + dx\}$, $F_4 = \{ax^3y + cxy^2, x^4y + 3dy, cx^2 + bxy, x^2y^2 + ax^2, x^5 + y^5\}$.

Problem	アルゴリズム	断片の数	CPU タイム (sec.)
F_1	Suzuki-Sato	7	0.079
	NEW	4	0.031
F_2	Suzuki-Sato	17	0.235
	NEW	7	0.078
F_3	Suzuki-Sato	31	2.421
	NEW	22	2.203
F_4	Suzuki-Sato	39	1.391
	NEW	15	0.234

上の図では NEW はオリジナルの Suzuki-Sato より効率的であることがわかる．次にもう少し難しいものについて見てみる．$F_5 = \{ax^2y + bx + y^3, ax^2y + bxy, y^2 + bx^2y + cxy\}$, $F_6 = \{x^4 + ax^3 + bx^2 + cx + d, 4x^3 + 3ax^2 + 2bx + c\}$, $F_7 = \{x^3 - a, y^4 - b, x + y - az\}$, $F_8 = \{ax^2 + by, cw^2 + z, (x - z)^2 + (y - w)^2, 2dxw - 2by\}$.

Problem	アルゴリズム	断片の数	CPU タイム (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

F_5 と F_6 において，NEW は Suzuki-Sato より速い．これは，NEW が生成する断片の数が Suzuki-Sato の生成する断片の数より小さいからである．F_7 においては，Suzuki-Sato ではすぐに計算結果を得ることができる．しかし, NEW では 30 分間以上計算機を走らせても計算結果を得ることはできなかった．これはなぜか？ NEW では r として新しい変数を導入して $\mathbb{C}[r,a,b][x,y,z]$

上でグレブナ基底の計算が必要である．このグレブナ基底の計算が重いことから 30 分間しても計算結果を得ることができなかった．グレブナ基底計算において変数を増やすことは計算量・計算速度の立場から見ると危険なことである．また，r として複雑なパラメータ入りの係数を取ると r を元に戻したときに，戻した多項式の集合は複雑な形となりその (r のない) グレブナ基底計算が重くなることもある．ここで，Suzuki-Sato と NEW の問題をまとめると以下である．

- Suzuki-Sato 数多くの断片を生成する．
- NEW は重いグレブナ基底計算が必要である．（しかし，断片の数は多くはない．）

ここで，消去多項式の選択方法を変えてみる．実際，F_7 におていプログラム NEW は悪い消去多項式を選択したことによって，プログラムは計算結果を 30 分以内に返さなかった．包括的グレブナ基底系の計算においては E からより良い消去多項式を選ぶことは重要なことである．
($E := \{f \in G : \exists g \in G \backslash \{f\} \text{ s.t. } \mathrm{lpp}_{\bar{X}}(f) | \mathrm{lpp}_{\bar{X}}(g)\}$)

実際，F_7 においては，NEW は 12 個の単項を持つ多項式を消去多項式として選択している．i.e.，集合 $\mathrm{Mono}_{\{x,y\}}(f)$ の要素数が 12 である．この多項式は大きく，そして変数 r を先頭項以外の 11 個の単項に掛けることよりその後のグレブナ基底計算が重くなると考えられる．この考察から著者は多くの計算実験をした結果，包括的グレブナ基底の計算時には **“我々は小さい多項式を消去多項式を選ぶべき”** ということに気がついた．すなわち，計算速度に関して**消去多項式の単項の数をカウントすること**を考察する必要がある．ここで，戦略 2 を紹介する．

• 戦略 2

> E の元の中で指定された数以下の項を持つ元を消去多項式の候補として選ぶ．次にそれらに戦略 1 を適用する．

ここで E_s として次のように定義する．
$E_s := \{f \in G : \sharp(\mathrm{Mono}_{\bar{X}}(f)) \leq s, \exists g \in G \backslash \{f\} \text{ s.t. } \mathrm{lpp}_{\bar{X}}(f) | \mathrm{lpp}_{\bar{X}}(g)\}$

$s \in \mathbb{N}$ で $\sharp(\mathrm{Mono}_{\bar{X}}(f))$ は集合 $\mathrm{Mono}_{\bar{X}}(f)$ の要素数である．明らかにこの場合 $E_s \subseteq E$ である．この E_s は消去多項式が s 個の単項を持つ集合である． 戦略 2 と戦略 1 を区別するため，戦略 1 を NEW と書き戦略 2 で E_s を持つものを NEW[s] と書くようにする．すると以下の図のようになる．戦略 2 を導入したことによって得られなかったものが得られるようになり，Suzuki-Sato よりもより良い結果が得られることが下の表より分かる．現在，計算速度と断片の数を考慮した最良の消去多項式の決定方法はオープンプロブレムである．

Problem	アルゴリズム	断片の数	CPU タイム (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

次に多くの人々が考える次の戦略について少し見る．

• **戦略 3**

E の各元の先頭項が $\mathrm{lpp}_{\bar{X}}(G)$ の元をいくつ割るかカウントし，その割る元の数が最大となる E の元を消去多項式とする．もし，最大数の数が同じなら $\succ$ に関して小さいものをとる．

この場合，問題によって良い場合と悪い場合があるがグレブナ基底計算が重くなる場合が多く見られる．基本的な振る舞いは戦略 1 と大きな違いはないように思われる．NEW のアルゴリズムは再帰的構造で小さいものから消去多項式を取っていくからこのような場合も計算するからだろうと考えられる．

戦略 1 と戦略 3 を見た場合，経験的に戦略 3 の場合が重いグレブナ基底計算が必要となる場合が多い．それは，戦略 3 は複雑な多項式の形の集合のままでのグレブナ基底計算が必要となる一方，戦略 1 は多項式の集合から小さい消去多項式を選び，複雑な多項式の形の集合をいくぶん複雑でない形，もしくはそんなに複雑にしない形に変形している（ここでの変形はグレブナ基底計算）のでその後のグレブナ基底計算が戦略 3 の場合よりも重くないと考えられる．

本アルゴリズムは Suzuki-Sato アルゴリズムの改良なので Suzuki-Sato 同様に計算時間はグレブナ基底計算に依存している．

他のパラメトリック・グレブナ基底計算のプログラム [DS97, DSS06, Mon02, MM06] と Risa/Asir 上の NEW(または NEW[s]) の実装と比較すると，パラメータ $\bar{A}$ の数が少ないときは Suzuki-Sato 同様に速い．パラメータの数が多いときも NEW(または NEW[s]) により Suzuki-Sato では得られなかった計算結果が得られるようになった．しかしながら, この場合は REDUCE[DS97] での実装，または MAPLE での実装 [Mon02, MM06] が速い場合も多々見られる．もちろん，Risa/Asir 上での本実装が速い場合も多々見られる．どの実装が速いかは問題に依存する．

6 まとめ

Suzuki-Sato アルゴリズムはパラメータ $\bar{A}$ の数が少ないときに高速に計算することができる．しかしながら多いときに生成させる包括的グレブナ基底系の断片の数が多くなることから計算速度は遅くなる．なぜならばパラメータの数が多いとパラメータ空間の分割の数が多くなることからその分割の数だけグレブナ基底の計算をする必要が出てくることが原因である．この分割数の増大は，包括的グレブナ基底系の計算時に $K[\bar{A}][\bar{X}]$ （多項式環を係数とする多項式環）上のグ

レブナ基底の計算が必要となり，そのグレブナ基底と $L[\bar{X}]$（体を係数とする多項式環）上の簡約グレブナ基底とのギャップが引き起こすものである．このギャップを埋めるべく，本稿では消去多項式を選択し新しい変数 r を導入して再度グレブナ基底を計算することでパラメータ空間の分割の数を抑えることに成功した．これにより，より良い包括的グレブナ基底系を得ることができ，また多くの場合で計算速度も向上している．しかしながら，新しいアルゴリズムでは重いグレブナ基底計算が必要となるときがある．このことより分割の数と計算時間を考慮した最良の消去多項式を選択することが重要である．

参 考 文 献

[AL94] William W. Adams and Philippe Loustaunau. *An Introduction to Gröbner Bases*. AMS-Providence, 1994.

[Bec94] Thomas Becker. On Gröbner bases under specialization. *Applicable Algebra in Engineering, Communication and Computing*, 5:1–8, 1994.

[Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*. Universität Innsbruck, Austria, 1965. Ph.D. Thesis.

[BW98] Bruno Buchberger and Franz Winkler Ed. *Gröbner Bases and Applications*. Cambridge University Press, 1998.

[DS97] Andreas Dolzmann and Thomas Sturm. Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin*, 31(2):2–9, 1997.

[DSS06] Andreas Dolzmann, Thomas Strurm, and Andreas Seidl. *Redlog User Mabual Version 3.06*. Universität Passau, 2006.

[FGT01] Elisabetta Fortuna, Patrizia Gianni, and Barry Trager. Degree reduction under specialization. *Journal of Pure and Applied Algebra*, 164(1-2):153–163, 2001.

[Gia87] Patrizia Gianni. Properties of Gröbner bases under specializations. In James H. Davenport, editor, *EUROCAL'87*, pages 293–297. ACM Press, 1987.

[Kal97] Michael Kalkbrener. On the Stability of Gröbner Bases Under Specializations. *Journal of Symbolic Computation*, 24:51–58, 1997.

[MM06] Montserrat Manubens and Antonio Montes. Improving DISPGB algorithm using the discriminant ideal. *Journal of Symbolic Computation*, 41:1245–1263, 2006.

[Mon02] Antonio Montes. A new algorithm for discussing Gröbner basis with parameters. *Journal of Symbolic Computation*, 33/1-2:183–208, 2002.

[Nab06] Katsusuke Nabeshima. Reduced Gröbner bases in polynomial rings over a polynomial ring. In Dongming Wang and Zhiming Zheng, editors, *International Conference on Mathematical Aspects of Computer and Information Sciences*, pages 15–32, 2006.

[NT92] Masayuki Noro and Taku Takeshima. Risa/Asir- A Computer Algebra System. In P. Wang, editor, *International Symposium on Symbolic and Algebraic Computation*, pages 387–396. AMC-Press, 1992.

[Sat05] Yosuke Sato. Stability of Gröbner basis and ACGB. In Andreas. Dlozmann, Andreas. Seidl, and Thomas Sturm, editors, *the A3L 2005, conference in Honor of the 60th Birthday of Volker Weispfenning*, pages 223–228. BOD Norderstedt, 2005.

[SS03] Akira Suzuki and Yosuke Sato. An alternative approach to Comprehensive Gröbner bases. *Journal of Symbolic Computation*, 36/3-4:649–667, 2003.

[SS06] Akira Suzuki and Yosuke Sato. A Simple Algorithm to compute Comprehensive Gröbner Bases using Gröbner bases. In *International Symposium on Symbolic and Algebraic Computation*, pages 326–331. ACM Press, 2006.

[Wei92] Volker Weispfenning. Comprehensive Gröbner bases. *Journal of Symbolic Computation*, 14/1:1–29, 1992.

[Wei03] Volker Weispfenning. Canonical Comprehensive Gröbner bases. *Journal of Symbolic Computation*, 36/3-4:669–683, 2003.