Non-compatible な加群項順序の下での signature based algorithm について

Signature based algorithm under non-compatible module term orderings

立教大学理学部　　野呂 正行 [*1]
Masayuki Noro
Faculty of Science
Rikkyo University

Abstract

The compatibility of a term ordering in a polynomial ring $R = K[X]$ and a module term ordering in R^m is a sufficient condition of the termination of the signature based algorithm. Experiments show that a combination of non-compatible term orderings may give good performance for computing Groebner bases with respect to some term orderings. In such cases, we can apply the notion of Hilbert function for guaranteeing the termination. The hilbert function can be computed by using a Groebner basis with respect to another term ordering and thus this algorithm is a kind of change of ordering. We implement this algorithm in Risa/Asir and we compare its performance with the usual Hilbert driven algorithm.

1　概要

Signature based algorithm の停止性を保証する十分条件として, 多項式環の項順序と加群項順序の compatibility がある. 実験の結果, non-compatible でない順序を用いてある種の項順序でのグレブナー基底が高速に計算できることがわかってきた. このような場合に停止性を保証するために, 既にわかっている別の順序でのグレブナー基底から計算した Hilbert 関数を用いることができる. これは, signature based algorithm を用いた change of ordering と言える. このアルゴリズムを Risa/Asir に実装し, 通常の Hilbert driven algorithm との性能比較を行う.

謝　　辞

この研究は JSPS 科研費基盤研究 (C) 21K03377 の助成をうけています。

This work was supported by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University.

2　Signature based algorithm

本節では, signature based algorithm (以下 sba) について, [9] の概要をまとめて述べる.

[*1] E-mail: noro@rikkyo.ac.jp

2.1 $\mathfrak{S}$-既約性と Signature based algorithm

K を体, $R = K[x_1, \ldots, x_n]$ とする. $I = \langle f_1, \ldots, f_\ell \rangle$ を R のイデアルとする. $<$ を R の項順序, $\prec$ を $R^\ell = Re_1 \oplus \cdots \oplus Re_\ell$ の加群項順序とする. $f \in R$ ($h \in R^\ell$) に対し, $<$ ($\prec$) に関する係数 1 の先頭単項式を $\mathrm{LM}(f)$ ($\mathrm{LM}(h)$) と書く.

定義 1 (signature, $\mathfrak{S}$(シグマ)-簡約, $\mathfrak{S}$-既約)

1. $f \in I$, $f \neq 0$ に対し, $S(f) = \min_\prec \{\mathrm{LM}(h) \mid h = \sum_{i=1}^{\ell} h_i e_i \in R^m, \sum_{i=1}^{\ell} h_i f_i = f\}$ を f の signature と呼ぶ.
2. $f \in I$ の先頭項を $g \in I$ で簡約する際, $S(f) \succ S(mg) \quad (m = \frac{\mathrm{LM}(f)}{\mathrm{LM}(g)})$ が成り立つとき, (regular) $\mathfrak{S}$-top-簡約 (略して $\mathfrak{S}$-簡約) と呼ぶ.
3. 1. で $S(f) \succeq S(mg)$ としたとき singular $\mathfrak{S}$-(top-) 簡約と呼ぶ.
4. f を regular $\mathfrak{S}$-簡約する $g \in I$ が存在しないとき f は $\mathfrak{S}$-既約という.

定義 2 ($\mathfrak{S}$-グレブナー基底)

$G \subset I$ が次を満たすとき G を I の $\mathfrak{S}$-グレブナー基底と呼ぶ.

任意の $\mathfrak{S}$-既約な $p \in I$ に対し $g \in G$, $m \in M$ が存在して $\mathrm{LM}(p) = m\mathrm{LM}(g)$, $S(p) = mS(g)$.

定義 3 (擬正則な S ペア)

1. $g, g' \in I$ の S 多項式が $cmg - c'm'g'$ $(c, c' \in K, m, m' \in M$ のとき, $mS(g) \neq m'S(g)$ なら $p = (g, g')$ を 擬正則な S ペアと呼ぶ.
2. $mS(g) \succ m'S(g')$ なら mg を, $mS(g) \prec m'S(g')$ なら $m'g'$ を主成分とよぶ.
3. 擬正則な S ペア p の主成分 mg に対する $mS(g)$ を S ペアの signature と呼び, $\hat{S}(p)$ と書く.

Algorithm 1 signature based algorithm

Input : $f_1, \ldots, f_\ell \in R$

Output : $I = \langle f_1, \ldots, f_\ell \rangle$ の $\mathfrak{S}$-グレブナー基底

```
G ← {f_1, ..., f_ℓ}; S(f_i) ← e_i (i = 1, ..., ℓ); S ← ∅
D ← G から作った 擬正則 S ペア全体
while D ≠ ∅ do
    s ← min{Ŝ(p) | p ∈ D}
    if s を割り切る S の元がない then
        mg ← {mg | g ∈ G, m ∈ M, mS(g) = s} 中で LM(mg) が最小の元
        if mg を主成分とする p ∈ D が存在する then
            r ← Spoly(p) を G で regular 簡約した余り
            if r ≠ 0 then
                S(r) ← s; D ← D∪ (G と r から作った 擬正則 S ペア); G ← G ∪ {r}
            else
                S ← S ∪ {s}
    D ← D \ {q ∈ D | S(q) = s}
return G
```

Algorithm 1 は $\mathfrak{G}$-既約元の signature s を加群項順序が小さい順にたどりながら, signature s 以下の元を 0 に singular $\mathfrak{G}$-簡約できる基底 G を求めていくアルゴリズムである. このアルゴリズムが停止すれば $\mathfrak{G}$-グレブナー基底を出力する. Buchberger アルゴリズムと同様, ゼロ簡約を排除するための判定基準が組み込まれている. この中で重要なものの一つが syzygy criterion である. これは, $LMSyz = \langle \mathrm{LM}(\mathrm{syz}(f_1, \ldots, f_\ell)) \rangle$ の元が signature になり得ないことを利用して, $\hat{S}(p)$ が $LMSyz$ に属する場合に p を捨てるという判定基準である. アルゴリズムにおいて, S はゼロ簡約された S 多項式に対する S ペアの signature からなる. regular S ペア, regular 簡約の性質よりこれは $LMSyz$ の元なので, S の元で割り切れる signature を持つペアは捨ててよいことになる.

2.2 Signature based algorithm の停止性

Algorithm 1 は, 任意の単項式順序, 加群単項式順序に対して実行できるが, 停止性は保証されない. 十分条件として次の compatibility がある.

定義 4
単項式順序 $<$, 加群単項式順序 $\prec$ が次を満たすとき, $\prec$ は $<$ と compatible であるという:
$t, s \in M$ が $t < s$ を満たすならば, $i = 1, \ldots, \ell$ に対し $te_i \prec se_i$.

定理 5 (有限 $\mathfrak{G}$-グレブナー基底)
加群単項式順序 $\prec$ が単項式順序 $<$ と compatible のときイデアル I の任意の $\mathfrak{G}$-グレブナー基底 G に対し, その有限部分集合で, $\mathfrak{G}$-グレブナー基底となるものがとれる. 特に, $\mathfrak{G}$-既約な有限個の元からなる $\mathfrak{G}$-グレブナー基底が存在する.

定理 6
加群単項式順序 $\prec$ が単項式順序 $<$ と compatible のときアルゴリズム 1 は停止して $\mathfrak{G}$-グレブナー基底を出力する.

加群単項式順序 $\prec$ が単項式順序 $<$ と compatible でない場合に, 実際にアルゴリズムが停止しない場合がある. [5] に実例が示されているが, ここではそれをより簡単にした例を示す.

例 1 (アルゴリズムが停止しない例:Arri-Perry の反例による)
$f_1 = x + z$, $f_2 = y + z$, $I = \langle f_1, f_2 \rangle \subset \mathbb{Q}[x, y, z] = R$ とする. $\prec$ を $x > y > z$ なる R の grevlex (全次数逆辞書式) 順序上の R^2 の Schreyer 順序, $<$ を $z > y > x$ なる R の glex 順序とする. 以下, 多項式 f とその signature を合わせて $(f, s(f))$ と書く. $f_1 = (\underline{z} + x, e_1)$, $f_2 = (\underline{z} + y, e_2)$ から $f_3 = (\underline{y} - x, e_1)$ が生成される. 以下, アルゴリズムに従って冗長ペアを省くと次に処理されるペアは (f_2, f_3) で, S 多項式は $ye_2 \succ ze_1$ より $y(z + y, e_2) - z(y - x, e_1) = (\underline{zx} + y^2, ye_2)$ である. これを簡約できる可能性があるのは f_2 のみだが, $xf_2 = (\underline{zx} + yz, xe_2)$ で $xe_2 \succ ye_2$ より f_2 で regular 簡約できない. よって $zx + y^2$ は $\mathfrak{G}$-既約であり $f_4 = (\underline{zx} + y^2, ye_2)$. 以下同様に $f_5 = (\underline{zyx} + y^3, y^2e_2), \ldots, f_k = (\underline{zy^{k-4}x} + y^{k-2}, y^{k-3}e_2)$ が $\mathfrak{G}$-既約となり, 無限に $\mathfrak{G}$-既約元が生成される.

加群単項式順序 $\prec$ が単項式順序 $<$ と compatible でも, sba が停止しにくい場合もある.

例 2 (glex 順序に関する sba の実行)
Katsura-7 を入力として単項式順序を glex (全次数辞書式), 加群単項式順序を glex 上の Schreyer 順序 で実行すると, 30 分待っても停止しないが, 単項式順序を grevlex, 加群単項式順序を glex 上の Schreyer 順序 で実行すると, 0.05sec で終了する. これは Singular の関数 `sba` で実行しても同様である. 一方で, 単

項式順序を glex, 加群単項式順序を grevlex 上の Schreyer 順序で実行すると, 0.4sec で停止する. これは non-compatible での実行である.

3 sba による change of ordering

例 1 により, grevlex 以外の項順序に関して, その上の Schreyer 順序を設定して $\mathfrak{S}$-グレブナー基底を計算することは効率の面から問題があることがわかる. ここで, 応用上必要となるのは一般には単なるグレブナー基底であることから, 加群項順序としては grevlex 上の Schreyer 順序を常に用いて sba を実行し, 計算途中で, 中間基底に対しグレブナー基底判定を行い, グレブナー基底を得た時点で実行を中断する方法を考える. グレブナー基底判定を行う方法としては, 他の項順序に関するグレブナー基底がわかっている場合に, Hilbert 関数を用いる方法が考えられる.

3.1 Hilbert 関数および Hilbert-Poincaré 級数

Hilbert 関数についての事実をいくつか述べる. 証明は例えば [4] を参照してほしい. $I \subset K[X] = K[x_1, \ldots, x_n]$ が重み $(w_1, \ldots, w_n)$ に関して斉次イデアルとする. $K[X]_d$, I_d をそれぞれ $K[X]$, I の d 次斉次成分, $M_d = K[X]_d/I_d$ とすると $M = K[X]/I \simeq \bigoplus_{d=0}^{\infty} M_d$ が成り立つ.

定義 7
$HF_M(d) = \dim_K M_d$ を M の *Hilbert* 関数, $HP_M(t) = \sum_{d=0}^{\infty} FH_M(d)t^d$ を *Hilbert-Poincaré* 級数と呼ぶ.

定理 8
$HP_M(t) = \frac{HN_M(t)}{(1-t^{w_1})\cdots(1-t^{w_n})}$ を満たす多項式 $HN_M(t)$ が存在する. $HN_M(t)$ を M の Hilbert numerator と呼ぶ [2]. G を任意の項順序 $<_0$ に関するグレブナー基底とするとき, $HP_M(t) = HP_{\langle \mathrm{LM}_{<_0}(G)\rangle}(t)$. 特に $HN_M(t) = HN_{\langle \mathrm{LM}_{<_0}(G)\rangle}(t)$.

定理 9
G を重み $(w_1, \ldots, w_n)$ に関して斉次多項式からなる I の部分集合とし, $N = K[X]/\langle \mathrm{LM}(G)\rangle$ とおくとき, G が項順序 $<$ に関する I のグレブナー基底 $\Leftrightarrow HN_M(t) = HN_{\langle \mathrm{LM}_{<}(G)\rangle}(t)$.

定理 8 により, 斉次イデアルの Hilbert numerator の計算は, 任意項順序に関するグレブナー基底の先頭項で生成される単項式イデアルの Hilbert numerator の計算に帰着される. また, 定理 9 により, グレブナー基底判定が, 中間基底 G の先頭項から計算される Hilbert numerator と, 別の項順序のもとで既に計算済みの $K[X]/I$ の Hilbert numerator の比較により行うことができる. 非斉次イデアルの場合, 斉次化を経由して計算することになる.

定義 10 (斉次化)
$f = \sum_\alpha c_\alpha x^\alpha \in R$, $f \neq 0$ に対し, f の, 重み $w = (w_1, \ldots, w_n)$ に関する斉次化 $f^h \in R[x_0] = K[x_0, \ldots, x_n]$ を, $\mathrm{tdeg}_w(x^\alpha) = w_1\alpha_1 + \cdots + w_n\alpha_n$, $\mathrm{tdeg}_w(f) = \max_\alpha \mathrm{tdeg}_w(x^\alpha)$ として

$$f^h = x_0^{\mathrm{tdeg}_w(f)} f(x_1/x_0^{w_1}, \ldots, x_n/x_0^{w_n})$$

で定義する. また, R の項順序 $<$ に対し, $<$ の斉次化 $<^h$ を, $R[x_0]$ の項順序で, 単項式 $t, s \in R$ に対し

$$x_0^i t <^h x_0^j s \Leftrightarrow$$
$$i + \mathrm{tdeg}_w(t) < j + \mathrm{tdeg}_w(s) \text{ または } (i + \mathrm{tdeg}_w(t) = j + \mathrm{tdeg}_w(s) \text{ かつ } t < s)$$

と定義する. 斉次多項式 $h \in R[x_0]$ に対し, h の**非斉次化**を $h|_{x_0=1} = h(1, x_1, \ldots, x_n)$ で定義する.

命題 11
$<$ が重み w 付き次数付き順序, すなわち, $\sum_i w_i\alpha_i < \sum_i w_i\beta_i$ ならば $x^\alpha < x^\beta$ が成り立つとき, $\mathrm{LM}_{<^h}(f^h) = \mathrm{LM}_<(f)$.

系 12
$<$ が重み w 付き次数付き順序とし, G をイデアル I の $<$ に関するグレブナー基底とすれば, $G^h = \{g^h \mid g \in G\}$ は $I^h = \langle f^h \mid f \in I, f \neq 0\rangle$ の $<^h$ に関するグレブナー基底である.

3.2 Hilbert-driven algorithm

Hilbert 関数を用いて 0 簡約を減らす方法として Traverso [1] による Hilbert-driven algorithm が知られている. 今回提案する Hilbert-driven sba と比較するため, ここで簡単にその原理を紹介しておく. 重み w に関する斉次イデアル I に対し $HN_M(t)$ $(M = K[X]/I)$ が与えられているとする. 項順序 $<$ に関する Buchberger アルゴリズムを w-次数の小さい順に実行するとき, w-斉次多項式からなる中間基底 G に対して $J = \langle G\rangle$, $N = K[X]/J$, J の d 次斉次部分 J_d, $N_d = K[X]_d/J_d$ とする. このとき $HP_N(t) = \frac{HN_N(t)}{(1-t^{w_1})\cdots(1-t^{w_n})}$ が $\mathrm{LM}_<(G)$ により計算できるが,

$$M_0 = N_0, \ldots, M_j = N_j \Leftrightarrow t^{j+1} \mid (HP_M(t) - HP_N(t)) \Leftrightarrow t^{j+1} \mid (HN_M(t) - HN_N(t))$$

より, 新たに中間基底を得るごとに $HN_N(t)$ を更新していき, $t^{j+1} \mid (HN_M(t) - HN_N(t))$ となった時点で, j 次のグレブナー基底の元が全て得られたことになるため残りの j 次の S ペアを全て捨てる, という方法で 0 簡約を減らすことができる. これが Hilbert-driven algorithm である. この方法は, d 次の基底が得られるまでは, 通常の剰余計算を行う必要があり, 0 簡約を全て排除することは一般にはできない. 特に, 係数体が無限体の場合にその計算コストが問題となるため, あらかじめ有限体上で簡約を行って剰余が 0 の場合にはもとの体上でも 0 とみなす trace アルゴリズムが有効である. この場合, 剰余が 0 でないものを 0 と判定する可能性があるが, これは d 次の S ペアを使い切った時点で $t^{j+1} \nmid (HN_M(t) - HN_N(t))$ となっていることで判定できる. この場合, 有限体を取り直して d 次の処理をやり直せばよい. この方法を modular Hilbert-driven algorithm と呼ぶ. この方法も [1] に概要が示唆されている.

3.3 Hilbert-driven signature based algorithm

Algorithm 2 Hilbert driven signature based algorithm

Input : イデアル I の, 重み w 付き grevlex 順序 $<_0$ に関するグレブナー基底 $G_0 = (g_1, \ldots, g_\ell)$
目的項順序 $<$

Output : I の $<$ に関するグレブナー基底

1: $G \leftarrow (g_1^h, \ldots, g_\ell^h)$
2: $\prec \leftarrow <_0^h$ 上の Schreyer 順序
3: $S(g_i^h) \leftarrow e_i \ (i = 1, \ldots, \ell)$
4: $S \leftarrow \{t_{ij} e_j \mid t_{ij} = \mathrm{LCM}(\mathrm{LM}(g_i), \mathrm{LM}(g_j)), 1 \leq i < j \leq \ell\}$
5: $D \leftarrow G$ から作った $(<_0^h, \prec)$ に関する擬正則 S ペア全体
6: $Q_0(t) \leftarrow HN_{K[X]/\langle \mathrm{LM}_{<_0}(g) \mid g \in G_0 \rangle}(t)$
7: $Q(t) \leftarrow HN_{K[X \cup \{x_0\}]/\langle \mathrm{LM}_{<^h}(g) \mid g \in G \rangle}(t)$
8: **while** $Q(t) \neq Q_0(t)$ **do**
9: $\quad s \leftarrow \min_\prec \{\hat{S}(p) \mid p \in D\}$
10: $\quad$ **if** s を割り切る S の元がない **then**
11: $\quad\quad R \leftarrow \{mg \mid g \in G, m \in M, mS(g) = s\}$
12: $\quad\quad mg \leftarrow R$ 中で $\mathrm{LM}(mg)$ が最小の元
13: $\quad\quad$ **if** mg を主成分とする $p \in D$ が存在する **then**
14: $\quad\quad\quad r \leftarrow \mathrm{Spoly}(p)$ を G で regular 簡約した余り
15: $\quad\quad\quad S(r) \leftarrow s$
16: $\quad\quad\quad D \leftarrow D \cup$ (G と r から作った $(<_0^h, \prec)$ に関する擬正則 S ペア)
17: $\quad\quad\quad G \leftarrow G \cup \{r\}$; $Q(t) \leftarrow HN_{K[X \cup \{x_0\}]/\langle \mathrm{LM}_{<^h}(g) \mid g \in G \rangle}(t)$
18: $\quad D \leftarrow D \setminus \{q \in D \mid S(q) = s\}$
19: **return** $G|_{a_0=1}$

定理 13
Algorithm 2 は停止して $<$ に関するグレブナー基底を出力する. アルゴリズムの 14 行目の剰余は常に 0 でない.

証明 $\langle g_1^h, \ldots, g_\ell^h \rangle$ の $<^h$ に関する簡約グレブナー基底 H に対し, $s_0 = \max\{S_\prec(h) \mid h \in H\}$ とする. $\prec$ は次数付き順序の上の Schreyer 順序より, s_0 以下の signature は有限個である. Algorithm 2 は, signature s を小さい順にたどりながら signature s 以下のイデアルの元を 0 に singular $\mathfrak{G}$-簡約できる基底を求めるアルゴリズムなので, 有限回ののちに s_0 に到達し, その時点での中間基底 G は $<^h$ に関するグレブナー基底となる. よって, この時点で $Q(t)$ は $Q_0(t)$ に等しくなり, アルゴリズムは停止する. G が $\langle g_1^h, \ldots, g_\ell^h \rangle$ の $<^h$ に関するグレブナー基底より, $G|_{x_0=1}$ は $\langle g_1^h|_{x_0=1}, \ldots, g_\ell^h|_{x_0=1} \rangle = I$ の $<$ に関するグレブナー基底となる. S は $\prec$ に関する $\mathrm{syz}(g_1, \ldots, g_\ell)$ のグレブナー基底の先頭加群単項式の集合より, S のどの元でも割り切れない s は signature であり, 従って 14 行目の剰余は 0 でない. ∎

Algorithm 2 は, $<_0$ に関するグレブナー基底を入力として, $<$ に関するグレブナー基底を計算するので, change of ordering アルゴリズムの一種である. 特筆すべき点は, 0 簡約が生じないことである. sba において, Hilbert-Poincaré 級数を用いてグレブナー基底を得た時点で実行を中断する方法は, [6] において既に提案されているが, 一般の入力イデアルに対するもので, Hilbert-Poincaré 級数は optional な入力として扱わ

	Hd sba		modular Hda				Hda
	full	top	full, nored	full, red	top, nored	top, red	top, nored
ahml	150	140	190	140	180	130	740
brocard	360	11	140	150	120	140	> 2h
butterfly	50	10	3.4	4.8	0.91	4.1	27
chou115_2	42	3.6	24	25	3.9	20	59
chou167	190	45	9.8	11	7.3	11	1300
chou238	710	370	13	15	8.1	16	340
chou302	11	5.9	2.2	5.3	1.5	13	23
chou393	220	36	110	120	30	110	190
chou393_2	90	32	14	23	16	20	1300
chou460	480	200	5.2	11	41	10	3000
chou94	18	5.4	7.4	13	3.1	8.8	7.1
cohn3	41	120	130	29	150	29	330
cyclic8	>2h	1000	>1h	8900	1600	8500	> 3h
czapor86c_2	69	15	1.1	1.7	0.62	1.3	0.73
gerhard2	450	1500	220	140	210	140	390
hairer2	4.3	1.5	3.2	3.5	2.3	3	49
hawes	320	14	370	300	24	300	310
rbpl	1600	>2h	1900	1500	1500	>2h	>10h
simson3	24	3.2	13	15	4.3	16	130

表 1: grevlex 順序から lex 順序への change of ordering

れている. アルゴリズムは non-compatible な項順序に対するものではなく, また, 0 簡約を完全に排除するものでもない.

4 計算機実験

4.1 Lex 順序グレブナー基底への change of ordering

0 次元イデアルについては FGLM アルゴリズムが効率よく change of ordering を実行できることが知られているが, 非 0 次元イデアルの場合には FGLM アルゴリズムは使えない. 表 4.1 に, 非 0 次元イデアルの lex 順序グレブナー基底を, Hilbert-driven sba (Hd sba), Hilbert-driven algorithm (Hda) で計算した結果を示す. Hda については modular および non-modular の結果を示す. 計算機実験は, Mac mini(2018) 上の Asir で行った. 計算時間の単位は秒である. 予備実験でグレブナー Walk も試したが, 計算が終わらない例が多数あったのでここでは用いないことにした. グレブナー基底に到達する時間を比較するため, 簡約グレブナー基底を得るための相互簡約は行わないこととする. 表において, full は先頭項以外も簡約, top は先頭項のみ簡約, modular Hda において, red は d 次の処理が終わったあとで, d 次の基底を相互簡約する, nored はその操作を行わない, を意味する.

	Hd sba		modular Hda			Hd sba		modular Hda	
	full	top	nored	red		full	top	nored	red
ahml	11	6.9	20	6.5	chou460	8.4	2.3	4.2	3.1
brocard	150	2.3	19	20	cohn3	23	25	64	15
chou167	9.1	2.7	1.0	1.2	cyclic8	570	420	1900	100
chou238	>1h	130	110	210	gerhard2	460	2800	250	140
chou302	19	1.2	0.98	27	hairer2	4.5	0.19	1.4	2.0
chou393	3.2	0.44	1.1	1.1	simson3	74	0.73	5.5	7.6

表 2: grevlex 順序から消去順序への change of ordering

4.2 消去イデアルへの change of ordering

イデアルの分解計算においては, lex 順序グレブナー基底の計算よりも, いくつかの変数を消去するための, 積順序による消去イデアル計算のほうが実用的である. イデアル I に対し $I \cap K[x_i, \ldots, x_n] \neq \langle 0 \rangle$, $I \cap K[x_{i+1}, \ldots, x_n] \neq \langle 0 \rangle$ となる i に対し, $K[x_1, \ldots, x_i]$, $K[x_{i+1}, \ldots, x_n]$ 上の grevlex 順序 $<_1$, $<_2$ の積順序によるグレブナー基底計算を, Hd sba および modular Hda で行った結果を表 4.3 に示す. 前節の結果から, modular Hda については top の場合のみ示す. 計算時間がどの方法でも 1 秒程度のものは除いてある.

4.3 実験結果について

lex 順序への change of ordering については, ここで用いた例に関して言えば, 多くの例で modular Hda の方が Hd sba より高速であり, 前者では top 簡約かつ各次数の相互簡約を行わない方が高速である. しかし, いくつかの例で sba が優っている. 消去順序への change of ordering については, 表からはどの方法が高速とは言えない. Hd sba, modular Hda それぞれにおいても, 実験を行った 2 つの設定でどちらが高速かは問題による. これらは, Hd sba と modular Hda の比較であるが, Hd sba が modular 計算を用いていないことを考えればその有用性は無視できない. 実際, 表 4.1 において non-modular Hda の結果と比較すれば, 大部分の例で Hd sba の方が高速である.

この実験では, 極小グレブナー基底を求めた段階で計算を終了させている. これは, 多くの例で, 簡約グレブナー基底を求めるための相互簡約計算に多大な時間がかかり, 比較が無意味になってしまうためである. 実際の応用においては極小グレブナー基底で十分な場合もあるが, 簡約グレブナー基底が必要な場合もあるため, 相互簡約をより高速化する方法の研究も必要である.

参 考 文 献

[1] C. Traverso, Hilbert Functions and The Buchberger Algorithm, J. Symb. Comp., 22, 355-376, 1997.

[2] M. Kreuzer, L. Robbiano, Computational Commutative Algebra 2, Springer, 2000.

[3] J.-C. Faugère, A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5), in Proc. ISSAC2002, 75-83, 2002.

[4] G-.M. Greuel, G. Pfister, A Singular Introduction to Commutative Algebra, Second Edition, Springer, 2007.

[5] A. Arri, J. Perry, The F5 criterion revised, J. Symb. Comp., 46, 1017-1029,2011. A revised version : https://arxiv.org/abs/1012.3664v6.

[6] Bruno Sim oes, An Hilbert-Driven Strategy for Signature-based Gröbner Basis Algorithms, Future Vision and Trends on Shapes, Geometry and Algebra, R. De Amicis and G. Conti (eds.), Springer, 13-38, 2014.

[7] 横山和弘, Signature-based アルゴリズムの正当性・停止性について, 計算機代数夏の学校 2019.

[8] W. Decker, G.-M. Greuel, G. Pfister, H. Schönemann, SINGULAR 4-2-0 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de, 2019.

[9] 野呂正行, 横山和弘, Risa/Asir における signature-based アルゴリズムの実装について, RIMS 講究録 No.2185, 139-148 (2021).

[10] T. Vaccon, T. Verron, K. Yokoyama, An affine tropical F_5 algorithm, J. Symb. Comp., 102, 132-152, 2021.