代数的閉体における限量子消去アルゴリズムについて

深作, 亮也
東京理科大学

井上, 秀太郎
東京理科大学

佐藤, 洋祐
東京理科大学

出版情報：COE Lecture Note. 49, pp.27-32, 2013-08-09. 九州大学マス・フォア・インダストリ研究所
バージョン：
権利関係：

代数的閉体における限量子消去アルゴリズムについて
(On QE Algorithms over algebraically closed field)

深作亮也
Ryoya Fukasaku *
井上秀太郎
Inoue Shutaro †
佐藤洋祐
Yosuke Sato ‡

東京理科大学
Tokyo University of Science

Abstract

Quantifier Elimination(QE) in the domain of an algebraically closed field is much simpler than that of a real closed field. We can construct a QE algorithm using only GCD computations of (parametric) unary polynomials. Though a more sophisticated QE algorithm using Gröbner bases computations is implemented in the computer algebra system Mathematica, it is basically based on GCD computations of (parametric) unary polynomials. We propose two algorithms, one is an improvement of the algorithm of Mathematica based on the result of [1], the another one is an algorithm based on computations of comprehensive Gröbner systems.

1 はじめに

複素数領域における限量子消去 (以下 QE と略記する) は実数領域における QE と比較すると、理論的には実装が容易である。1 変数多項式の GCD の計算を再帰的に繰り返すことでアルゴリズムを構成することができる。ただし、扱う 1 変数多項式は一般にパラメターを含んでいるので、擬似剰余演算が必要になり、計算を効率的に行うには様々な工夫が必要になる。数式処理システム Mathematica の組み込み関数 Reduce と Resolve で実装されている複素数領域における QE では、グレブナー基底の計算等を利用してパラメターを含んだ 1 変数多項式の GCD の計算をおこなうよう工夫されている [6]。

Comprehensive グレブナー基底系 (以下 CGS と略記する) を用いると、GCD 計算による再帰的アルゴリズムとは全く違った方法で複素数領域における QE が容易に実装できるが、この方法では新たな変数を導入する必要があり、これまで CGS の効率的アルゴリズムの実装がなかったこともあり、これまでこの方法による実装はなされていない。

*fukasaku@mi.kagu.tus.ac.jp
†sinoue@rs.kagu.tus.ac.jp
‡ysato@rs.kagu.tus.ac.jp

最近の一連の研究成果 [9, 7, 4, 5, 8] により、CGS 計算が実装され利用できるようになったことを踏まえ、われわれは CGS を用いる方法と、GCD 計算による再帰的アルゴリズムによる方法の改良版を数式処理システム Risa/Asir を用いて実装し、２つの方法について比較検証をおこなった。

複素数領域における限量子消去 (以下 QE と略記する) をおこなうには、以下の形の論理式から、限量子 $\exists X_1 \exists X_2 \ldots \exists X_n$ を消去した $Y_1, \ldots, Y_m$ のみの式が得られればよいので、以下ではこの形の論理式にたいする限量子消去アルゴリズムのみをあつかう。

$$\exists X_1 \exists X_2 \ldots \exists X_n (f_1(Y_1, \ldots, Y_m, X_1, \ldots, X_n) = 0 \wedge \cdots \wedge f_s(Y_1, \ldots, Y_m, X_1, \ldots, X_n) = 0 \wedge \\ g_1(Y_1, \ldots, Y_m, X_1, \ldots, X_n) \neq 0 \wedge \cdots \wedge g_t(Y_1, \ldots, Y_m, X_1, \ldots, X_n) \neq 0)$$

以下、２章で本論文で用いるバックグラウンドについて必要最低限の解説を与える。３章では GCD 計算による再帰的アルゴリズムによる方法について、われわれの改良版も含め述べる。４章では CGS に基づく方法について述べる。最後に、われわれの計算実験により得られた双方の問題点と今後の課題について報告する。

2　グレブナー基底の安定性と CGS

以下において K は任意の体、$\overline{K}$ をその代数閉包とする。$\bar{X}$ は n 個の変数 $X_1, X_2, \ldots, X_n$、$\bar{Y}$ はそれとは異なる m 個の変数 $Y_1, Y_2, \ldots, Y_m$ を表す。$m = 1, n = 1$ のときは単に X, Y と記すことにする。また、イデアル I の多様体を $\mathbb{V}(I)$ で表す。

まず、グレブナー基底の安定性に関する次の結果から述べる。

定理 1 ([1])
$K[\bar{Y}, X]$ の有限集合 F に対して G を X を辞書式に $\bar{Y}$ よりも大きくなるような項順序に関するイデアル $\langle F \rangle$ のグレブナー基底とするとき、任意の要素 $\bar{c} = c_1, c_2, \ldots, c_m \in \overline{K}^m$ に対して、$G(\bar{c}) = \{g(\bar{c}, X) : g \in G\}$ は $\overline{K}[X]$ において $F(\bar{c}) = \{f(\bar{c}, X)\}$ で生成されるイデアルのグレブナー基底になる。

次に CGS の定義を述べる。

定義 1
$\bar{X}$ の項順序 $>$ を一つ固定する。$K[\bar{Y}, \bar{X}]$ の有限部分集合 F に対して、以下をみたす順序対の有限集合 $\mathcal{G} = \{(G_1, P_1, Q_1), \ldots, (G_s, P_s, Q_s)\}$ をパラメーター $\bar{Y}$、主変数 $\bar{X}$ の $>$ に関する F の CGS とよぶ。ここで、各 G_i は $K[\bar{Y}, \bar{X}]$ の有限部分集合、各 P_i, Q_i は $K[\bar{Y}]$ の有限部分集合である。
(i)　$\cup_{i=1}^{s} \mathbb{V}(\langle P_i \rangle) - \mathbb{V}(\langle Q_i \rangle) = \overline{K}^m$、$(\mathbb{V}(\langle P_i \rangle) - \mathbb{V}(\langle Q_i \rangle)) \cap (\mathbb{V}(\langle P_j \rangle) - \mathbb{V}(\langle Q_j \rangle)) = \emptyset$ for $i \neq j$。
(ii) 任意の $\bar{c} \in \mathbb{V}(\langle P_i \rangle) - \mathbb{V}(\langle Q_i \rangle)$ にたいして、$G_i(\bar{c}, \bar{X}) = \{g(\bar{c}, \bar{X}) : g \in G_i\}$ は $\overline{K}[\bar{X}]$ において、
$\langle f(\bar{c}, \bar{X}) \rangle$ のグレブナー基底である。
さらに、各 $G_i(\bar{c}, \bar{X})$ が reduced(minimal) グレブナー基底であるであるとき (monic であることは仮定しない)、reduced(minimal)CGS とよぶ。

3　QE の再帰的アルゴリズム

論理式 $\exists X_1 \exists X_2 \ldots \exists X_n (f_1(\bar{Y}, \bar{X}) = 0 \wedge \cdots \wedge f_s(\bar{Y}, \bar{X}) = 0 \wedge g_1(\bar{Y}, \bar{X}) \neq 0 \wedge \cdots \wedge g_t(\bar{Y}, \bar{X}) \neq 0)$ は $g_1(\bar{Y}, \bar{X}) \cdots g_t(\bar{Y}, \bar{X}) = g(\bar{Y}, \bar{X})$ として以下の同値な式に変形できる。

$\exists X_1 \exists X_2 \ldots \exists X_n (f_1(\bar{Y}, \bar{X}) = 0 \wedge \cdots \wedge f_s(\bar{Y}, \bar{X}) = 0 \wedge g(\bar{Y}, \bar{X}) \neq 0)$
$\exists X_n (f_1(\bar{Y}, \bar{X}) = 0 \wedge \cdots \wedge f_s(\bar{Y}, \bar{X}) = 0 \wedge g(\bar{Y}, \bar{X}) \neq 0)$ の式から $\exists X_n$ を消去し、その式の $\vee\wedge$ 標準型からさらに $\exists X_{n-1}$ を消去し、これを繰り返すことで、すべての限量子が消去できる。
したがって、$n = 1$ の場合について、アルゴリズムを構成すれば、これを再帰的に繰り返すことですべての限量子が消去できる。数式処理システム Mathematica の組み込み関数 Reduce と Resolve で実装されている K が有理数体のときの QE アルゴリズムでは、基本的にこの方法が用いられている。
以下では、Mathematica におけるアルゴリズムの概要を述べ、次にわれわれによる改良を与える。

Mathematica の QE の概要
Input: $\exists X(f_1(\bar{Y}, X) = 0 \wedge \cdots \wedge f_s(\bar{Y}, X) = 0 \wedge g(\bar{Y}, X) \neq 0)$
Output: $\bar{Y}$ のみの多項式による論理式
Step1. 項順序 $X \gg \bar{Y}$ による $\langle f_1, \ldots, f_s \rangle$ のグレブナー基底
$G = \{g_1(\bar{Y}, X), \ldots, g_l(\bar{Y}, X), h_1(\bar{Y}), \ldots, h_t(\bar{Y})\}$ を計算する。
Step2. $h_1(\bar{Y}) = 0 \wedge \cdots \wedge h_t(\bar{Y}) = 0$ でなければ与式は偽になることに注意する。以下 $h_1(\bar{c}) = 0 \wedge \cdots \wedge h_t(\bar{c}) = 0$ をみたす $\bar{c} \in \mathbb{C}^m$ について考える。与式が真になるのは、$g(\bar{c}, X)$ が $\langle f_1(\bar{c}, X), \ldots, f_s(\bar{c}, X) \rangle$ の根基イデアルに属さないことと同値であることに注意する。各 g_i を X の多項式とみなして、その中の最小次数のもの g_i を一つ選び、その次数を d、その最大項の係数を $p(\bar{Y})$ とおく。$p(\bar{c}) \neq 0$ ならば、$\{g_i(\bar{c}, X)\}$ が $\langle f_1(\bar{c}, X), \ldots, f_s(\bar{c}, X) \rangle$ のグレブナー基底、すなわち $g_i(\bar{c}, X)$ が $f_1(\bar{c}, X), \ldots, f_s(\bar{c}, X)$ の GCD になることに注意すると、$g(\bar{c}, X)$ が $\langle f_1(\bar{c}, X), \ldots, f_s(\bar{c}, X) \rangle$ の根基イデアルに属さないことと $g(\bar{c}, X)^d$ を $g_i(\bar{c}, X)$ で割った余りが 0 でないことが同値になる。$g(\bar{Y}, X)^d$ の $g_i(\bar{Y}, X)$ による疑剰余の係数を $p_1(\bar{Y}), \ldots, p_r(\bar{Y})$ とすると、$p(\bar{c}) \neq 0$ ならば、与式は $p_1(\bar{c}) \neq 0 \vee \cdots \vee p_r(\bar{c}) \neq 0$ と同値になる。
$p(\bar{c}) = 0$ の場合は、$\exists X(f_1(\bar{Y}, X) = 0 \wedge \cdots \wedge f_s(\bar{Y}, X) = 0 \wedge p(\bar{Y}) = 0 \wedge g(\bar{Y}, X) \neq 0)$ を新たな入力として再帰的に計算をおこうなう。その出力を $\phi(\bar{Y})$ とすると、結局全体の出力は
$\phi(\bar{Y}) \vee (h_1(\bar{Y}) = 0 \wedge \cdots \wedge h_t(\bar{Y}) = 0 \wedge p(\bar{Y}) \neq 0) \wedge (p_1(\bar{c}) \neq 0 \vee \cdots \vee p_r(\bar{c}) \neq 0))$ となる。

われわれの改良
$p(\bar{c}) \neq 0$ ならば、$g_i(\bar{c}, X)$ が $f_1(\bar{c}, X), \ldots, f_s(\bar{c}, X)$ の GCD になることはグレブナー基底の安定性に関する [3, 2] 等の結果を使うと容易に示すことができるが、前章で述べた定理 1 を使うと、先頭項が 0 であるなしにかかわらず $\{g_1(\bar{c}, X), \ldots, g_l(\bar{c}, X)\}$ がグレブナー基底であるので、これらの中に 1 つでも 0 でないものがある場合は GCD が定まり、再帰計算をおこなう必要がない。$g_1(\bar{Y}, X), \ldots, g_l(\bar{Y}, X)$ の係数として現れる $K[\bar{Y}]$ の要素をすべてを並べて $r_1(\bar{Y}), \ldots, r_k(\bar{Y})$ とおくと、再帰計算が必要になるのは $\langle h_1(\bar{Y}), \ldots, h_t(\bar{Y}), r_1(\bar{Y}), \ldots, r_k(\bar{Y}) \rangle \neq \langle 1 \rangle$ の場合のみであるため、ほとんどの場合は再帰計算をおこなわずにすむ。

4 QE の CGS によるアルゴリズム

前章で述べたアルゴリズムでは、パラメターを含む 1 変数多項式の GCD の効率的な計算のためにグレブナー基底計算をおこなってはいるが、グレブナー基底計算の計算が本質的に必要なわけではない。これにたいし、CGS の計算を用いると限量子 $\exists X_1 \exists X_2 \ldots \exists X_n$ を一挙に消去できる。

CGS による QE アルゴリズム
Input: $\exists \bar{X}(f_1(\bar{Y}, \bar{X}) = 0 \wedge \cdots \wedge f_s(\bar{Y}, \bar{X}) = 0 \wedge g_1(\bar{Y}, \bar{X}) \neq 0 \wedge \cdots \wedge g_t(\bar{Y}, \bar{X}) \neq 0)$
Output: $\bar{Y}$ のみの多項式による論理式

新たな変数 $\bar{Z} = Z_1, \ldots, Z_t$ を用いて、
$\{f_1(\bar{Y}, \bar{X}), \ldots, f_s(\bar{Y}, \bar{X}), g_1(\bar{Y}, \bar{X})Z_1 - 1, \ldots, g_t(\bar{Y}, \bar{X})Z_t - 1\}$ のパラメーター $\bar{Y}$ 主変数 $\bar{X}, \bar{Z}$ の CGS $\mathcal{G} = \{(G_1, P_1, Q_1), \ldots, (G_r, P_r, Q_s)\}$ を計算する。
ここで、G_i が定数でない多項式、すなわち $\bar{X}$ のどれかの変数を含む多項式を少なくとも 1 つ持つものすべてを $G_1, \ldots, G_k$ とすると、出力する論理式は $\vee_{i=1}^{k} \phi_i$ となる。ここで各 ϕ_i は $P_i = \{p_1(\bar{Y}), \ldots, p_a(\bar{Y})\}, Q_i = \{q_1(\bar{Y}), \ldots, q_b(\bar{Y})\}$ にたいして、$\phi_i \equiv p_1(\bar{Y}) = 0 \wedge \cdots \wedge p_a(\bar{Y}) = 0 \wedge (q_1(\bar{Y}) \neq 0 \vee \cdots \vee q_b(\bar{Y}) \neq 0)$ で与えられる。

CGS の計算が利用できる場合は、この方法は前章の再起的方法と比べると実装は容易であるが、あらたな変数を導入するため、一般的に、この CGS の計算は再起的方法で用いるグレブナー基底の計算よりもかなり重たいグレブナー基底の計算を必要とする。

5 まとめ

本文では述べなかったが QE アルゴリズムの最重要なポイントの 1 つとして、得られた論理式の簡単化があげられる。例えば、$\exists x \exists y \exists z (x*y+a*x*z+y*z-1 = 0 \wedge x*y*z+x*z+x*y+a = 0 \wedge x*z+y*z-a*z-x-y-1 = 0)$ から限量子を除去するために、Mathematica の

```
Resolve[Exists[{x,y,z},x*y+a*x*z+y*z-1==0&&x*y*z+x*z+x*y+a==0&&x*z+y*z-a*z-x-y-1==0]]
```

を実行すると、以下のような出力がされる。

```
(2159 - 4829 a - 592 a^2 + 6293 a^3 - 3932 a^4 - 844 a^5 + 3494 a^6 -
     1783 a^7 - 308 a^8 + 632 a^9 - 260 a^10 + 35 a^11 !=
    0 && -1 + a + a^2 ==
    0) || (-55 + 608 a - 2250 a^2 + 1429 a^3 - 1233 a^4 + 1935 a^5 -
     178 a^6 - 575 a^7 + 216 a^8 + 78 a^9 - 80 a^10 + 15 a^11 !=
    0 && -1 + a + a^2 ==
    0) || (-1339 + 3981 a - 5010 a^2 - 150 a^3 + 4607 a^4 -
     3776 a^5 + 794 a^6 + 2220 a^7 - 2500 a^8 + 1268 a^9 - 350 a^10 +
     40 a^11 != 0 && -1 + a + a^2 ==
    0) || (1017 - 1640 a - 164 a^2 + 323 a^3 + 665 a^4 - 1557 a^5 +
     1060 a^6 + 779 a^7 - 1268 a^8 + 720 a^9 - 210 a^10 + 25 a^11 !=
    0 && -1 + a + a^2 ==
    0) || (-736 + 1666 a - 2634 a^2 + 1425 a^3 + 2505 a^4 -
     3144 a^5 + 444 a^6 + 1919 a^7 - 2060 a^8 + 990 a^9 - 250 a^10 +
     25 a^11 != 0 && -1 + a + a^2 == 0) || (2 + a - 4 a^2 + 2 a^3 !=
    0 && 1 - a + 5 a^2 - 6 a^3 + 2 a^4 == 0) || (a !=
    0 && -1 + a + a^2 != 0 &&
   1 - 5 a + 9 a^2 - 34 a^3 + 37 a^4 + 593 a^5 - 1814 a^6 - 558 a^7 +
     8218 a^8 - 8848 a^9 - 2449 a^10 + 7850 a^11 - 11542 a^12 +
     23516 a^13 - 5320 a^14 - 34315 a^15 + 23525 a^16 + 14094 a^17 -
     13659 a^18 - 1889 a^19 + 2819 a^20 - 62 a^21 - 192 a^22 +
     21 a^23 + a^24 != 0) || (a != 0 && -1 + a + a^2 != 0 &&
```

```
  2 - a - 23 a^2 - 309 a^3 + 2459 a^4 - 6333 a^5 + 5855 a^6 +
    2023 a^7 - 5443 a^8 - 3255 a^9 + 16135 a^10 - 27046 a^11 +
    11336 a^12 + 43457 a^13 - 59351 a^14 - 1657 a^15 + 37823 a^16 -
    10467 a^17 - 8711 a^18 + 3105 a^19 + 645 a^20 - 244 a^21 +
    4 a^22 != 0) || (a != 0 && -1 + a + a^2 != 0 &&
  3 - 24 a + 73 a^2 - 215 a^3 + 1220 a^4 - 4879 a^5 + 11776 a^6 -
    25023 a^7 + 64683 a^8 - 146092 a^9 + 225027 a^10 - 253592 a^11 +
    262404 a^12 - 218871 a^13 + 23516 a^14 + 175369 a^15 -
    142711 a^16 - 1717 a^17 + 40984 a^18 - 10560 a^19 - 2375 a^20 +
    1108 a^21 - 122 a^22 + 6 a^23 != 0) || (a != 0 && -1 + a + a^2 !=
    0 && -2 + 18 a - 68 a^2 + 238 a^3 - 1300 a^4 + 5866 a^5 -
    17520 a^6 + 38578 a^7 - 75524 a^8 + 143803 a^9 - 245133 a^10 +
    346252 a^11 - 396084 a^12 + 326258 a^13 - 94318 a^14 -
    161094 a^15 + 204788 a^16 - 61953 a^17 - 35949 a^18 +
    27510 a^19 - 3626 a^20 - 1245 a^21 + 657 a^22 - 200 a^23 +
    26 a^24 != 0) || (a != 0 && -1 + a + a^2 !=
   0 && -1 + 10 a - 25 a^2 - 23 a^3 + 98 a^4 + 303 a^5 - 1071 a^6 +
    258 a^7 + 1563 a^8 + 2283 a^9 - 15219 a^10 + 28567 a^11 -
    33638 a^12 + 46364 a^13 - 86553 a^14 + 103975 a^15 -
    33012 a^16 - 54793 a^17 + 52896 a^18 - 4243 a^19 - 11829 a^20 +
    4135 a^21 + 280 a^22 - 367 a^23 + 57 a^24 != 0) || (a !=
   0 && -1 + a + a^2 != 0 &&
  9 - 91 a + 435 a^2 - 1521 a^3 + 4807 a^4 - 13313 a^5 + 30577 a^6 -
    60336 a^7 + 105398 a^8 - 157341 a^9 + 191334 a^10 -
    175100 a^11 + 79427 a^12 + 65253 a^13 - 133039 a^14 +
    63412 a^15 + 23932 a^16 - 30149 a^17 + 3701 a^18 + 3572 a^19 -
    955 a^20 - 44 a^21 + 26 a^22 != 0) || (-1 + a + a^2 != 0 &&
  17 - 44 a + 36 a^2 - 12 a^3 + 4 a^4 != 0 &&
  a == 0) || (-1 + a + a^2 != 0 && 10 - 14 a + 3 a^2 + 2 a^3 != 0 &&
  a == 0) || (-1 + a + a^2 != 0 &&
  7 - 40 a + 48 a^2 - 20 a^3 + 2 a^4 != 0 &&
  a == 0) || (-1 + a + a^2 !=
   0 && -3 - 8 a + 18 a^2 - 12 a^3 + 8 a^4 != 0 && a == 0) ||
1 + a == 0
```

この出力は間違いではないが、実はこれを簡易化すると true になる。しかし Mathematica の Simplify や FullSimplify を使っても true は得られない。一方、$\{x*y+a*x*z+y*z-1, x*y*z+x*z+x*y+a, x*z+y*z-a*z-x-y-1\}$ の CGS を a をパラメーターとして計算すると、定数だけからなる G_i は存在しないので、これからただちに与式が true であることが得られる。われわれが使用した CGS の計算アルゴリズムは [8] のものを使っているため、他のアルゴリズムと比較すると CGS の個数が少なく抑えられている。このため、出力された論理式は一般的にある程度の簡易化がすでになされている。ただし、自由変数の個数が多い時は CGS の計算時間が増大するという欠点がある。計算時間をおさえつつ簡易化もある程度実現させるためには、再帰計算か CGS の計算のどちらか一方をおこなうのではなく、再帰計算の方法と CGS の計算の方法を融合させたアルゴリズムが有効であることが予想される。

参 考 文 献

[1] Fortuna,E., Gianni,P. and Trager,B. (2001). Degree reduction under specialization. J. Pure Appl. Algebra, 164, pp. 153-164, 2001.

[2] Gianni, P.(1987). Properties of Grb̈ner bases under specializations. Lecture Notes in Comput. Sci., 378. pp. 293297. 1987.

[3] Kalkbrener, M.(1987). Solving systems of algebraic equations by using Gröbner bases. Lecture Notes in Comput. Sci., 378. pp. 282-292. 1987.

[4] Kapur, D., Sun, Y., and Wang, D. (2010). A New Algorithm for Computing Comprehensive Gröbner Systems. In International Symposium on Symbolic and Algebraic Computation, pp. 29-36. ACM-Press, 2010.

[5] Kurata, Y. (2011). Improving Suzuki-Sato's CGS Algorithm by Using Stability of Gröbner Bases and Basic Manipulations for Efficient Implementation. Communications of JSSAC Vol 1. pp 39-66. 2011.

[6] Mathematica Tutorial 複素多項式系　QE(量限定子除去)

[7] Nabeshima, K. (2007). A Speed-Up of the Algorithm for Computing Comprehensive Gröbner Systems. International Symposium on Symbolic and Algebraic Computation, pp. 299-306. ACM-Press, 2007.

[8] Nabeshima, K. (2012). Stability Conditions of Monomial Bases and Comprehensive Gröbner systems. Lecture Notes in Computer Science, Vol.7442, pp.248–259, 2012.

[9] Suzuki,A. and Sato,Y. (2006). A Simple Algorithm to Compute Comprehensive Grbner Bases Using Gröbner Bases. International Symposium on Symbolic and Algebraic Computation, pp. 326-331. ACM-Press, 2006.