On QE Algorithms over Algebraically Closed Field

Extended Abstract

Ryoya Fukasaku, Shutaro Inoue and Yosuke Sato

1. Introduction

Quantifier Elimination(QE) in the domain of an algebraically closed field is much simpler than that of a real closed field at least from a theoretical point of view. Basically, we have two naive methods.

We can recursively eliminate a quantified variable step by step using only GCD computations of parametric unary polynomials. (See Chapter 1 of [2] for example.) This method, we call GCD-QE in this paper, is implemented in the computer algebra system Mathematica with a more sophisticated algorithm using Gröbner bases computations[8]. As far as we know, it is the most efficient existing implementation among others such as [4]. When the number of quantified variables is not small, however, the output often consists of very complicated form. As long as we use GCD-QE, we often encounter the blowup of the recursive steps.

The another method is based on the computation of a comprehensive Gröbner system(CGS for short), we call it CGS-QE in this paper. We can eliminate quantified variables simultaneously by computing only one CGS. With a series of resent results of [11, 9, 6, 7, 10] we now have practical implementations to compute CGS's. Among the algorithms introduced by them, the algorithm introduced in [10] often produces a CGS with a minimum number of segments, which enables us to obtain a concise form of the equivalent quantifier-free formula.

We implemented his algorithm on the computer algebra system Risa/Asir[1] to compare the above two methods. (We also implemented GCD-QE algorithm used in Mathematica with a slight improvement on Risa/Asir in order for the comparison to be fair.) According to our computation experiments, in most cases the output of CGS-QE algorithm is more concise than the output of GCD-QE algorithm. However, when we have many inequations in a given quantified formula, we sometimes have examples such that neither GCD-QE algorithm nor CGS-QE algorithm terminates. In order to handle such hard examples, we introduced a new algorithm which combines CGS-QE and GCD-QE algorithms, and implemented it on Risa/Asir. For many examples which are not handled by either GCD-QE or CGS-QE algorithm, we can successfully get the equivalent quantifier free formulas using our program.

Note that for QE in an algebraically closed field, it suffices to give an algorithm for the following basic form:

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

In this paper, we deal with only this basic formula. In section 2, we give a minimum description concerning stability of a Gröbner basis and CGS. In section 3, we describe the GCD-QE algorithm

M. Kosta and T. Sturm, eds. Proceedings of the MACIS 2013, Nanning, China, December 11-13, 2013.

implemented in Mathematica. In section 4, we describe the CGS-QE algorithm. In section 5, we introduce our new algorithm.

2. Stability of Gröbner Basis and CGS

We use the following notations. K denotes a field and $\overline{K}$ its algebraic closure. $K[\bar{Y}, \bar{X}]$ denotes a polynomial ring with variables $\bar{Y} = Y_1, \ldots, Y_m$ and $\bar{X} = X_1, \ldots, X_n$. σ denotes a homomorphism from $K[\bar{Y}]$ to $\overline{K}$, i.e. a specialization of $\bar{Y}$ with elements $c_1, \ldots, c_m$ of $\overline{K}$, it is also naturally extended to a homomorphism from $K[\bar{Y}, \bar{X}]$ to $\overline{K}[\bar{X}]$. $T(\bar{X})$ denotes the set of terms consisting of $\bar{X}$. An admissible term order on $T(\bar{Y}, \bar{X})$ such that each X_i is greater than any term in $T(\bar{Y})$ is denotes by $\bar{X} \gg \bar{Y}$.
We fix an admissible term order $>$ on $T(\bar{X})$, $LM(h)$, $LT(h)$ and $LC(h)$ denotes the leading monomial, the leading term and the leading coefficient respectively of $h \in K[\bar{Y}, \bar{X}]$ w.r.t. $>$ regarding $K[\bar{Y}, \bar{X}]$ as a polynomial ring $(K[\bar{Y}])[\bar{X}]$ over the coefficient ring $K[\bar{Y}]$. Note that $LM(h) = LC(h)LT(h)$.
For an ideal I of a polynomial ring over K, its variety in $\overline{K}$ is denoted by $\mathbb{V}(I)$.

We begin with the following result concerning stability of Gröbner basis, which is an easy consequence of Theorem 3.1 of [5] as observed in [6, 7].

Theorem 1

Let I be an ideal of $K[\bar{Y}, \bar{X}]$ and G be its Gröbner basis w.r.t. $>$ regarding $K[\bar{Y}, \bar{X}]$ as a polynomial ring $(K[\bar{Y}])[\bar{X}]$. let $G = \{g_1, \ldots, g_s, \ldots, g_t\}$ such that $G \cap K[\bar{Y}] = \{g_{s+1}, \ldots, g_t\}$ and $\sigma(g_{s+1}) = 0, \ldots, \sigma(g_t) = 0$. Let $\{LT(g_{n_1}), \ldots, LT(g_{n_l})\}$ be the minimal subset of $\{LT(g_1), \ldots, LT(g_s)\}$ concerning the order of divisibility, that is each term of $\{LT(g_1), \ldots, LT(g_s)\}$ is divisible by some term of $\{LT(g_{n_1}), \ldots, LT(g_{n_l})\}$ and any term of $\{LT(g_{n_1}), \ldots, LT(g_{n_l})\}$ is not divisible by others. If $\sigma(LM(g_{n_1})) \neq 0, \ldots, \sigma(LM(g_{n_l})) \neq 0$, then $G' = \{\sigma(g_{n_1}), \ldots, \sigma(g_{n_l})\}$ is a Gröbner basis of $\langle \sigma(I) \rangle$ w.r.t. $>$ regardless whether $\sigma(LM(g_i)) = 0$ or not for each $i \in \{1, \ldots, s\} - \{n_1, \ldots, n_l\}$.

Note that we can compute a Gröbner basis of $(K[\bar{Y}])[\bar{X}]$ using a term order of $T(\bar{Y}, \bar{X})$ which extends $>$ and satisfies $\bar{X} \gg \bar{Y}$. We next give a definition of CGS.

Definition 1

For a finite subset F of $K[\bar{Y}, \bar{X}]$, a finite subset $\mathcal{G} = \{(G_1, P_1, Q_1), \ldots, (G_s, P_s, Q_s)\}$ of triples which satisfies the following properties is called a CGS(comprehensive Gröbner system) of F with parameters $\bar{Y}$ and main variables $\bar{X}$ w.r.t. $>$. Where each G_i is a finite subset of $K[\bar{Y}, \bar{X}]$ and each P_i, Q_i is a finite subset of $K[\bar{Y}]$.
(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) For each $\bar{c} \in \mathbb{V}(\langle P_i \rangle) - \mathbb{V}(\langle Q_i \rangle)$, $G_i(\bar{c}, \bar{X}) = \{g(\bar{c}, \bar{X}) : g \in G_i\}$ is a Gröbner basis of $\langle f(\bar{c}, \bar{X}) \rangle$ in $\overline{K}[\bar{X}]$
w.r.t. $>$.
In addition, if each $G_i(\bar{c}, \bar{X})$ is a reduced(minimal) Gröbner basis, $\mathcal{G}$ is said to be reduced(minimal). (We do not require the polynomials to be monic.)

3. GCD-QE algorithm

We give a brief sketch of GCD-QE algorithm implemented in Mathematica packages Reduce and Resolve.
The basic formula

$$\exists X_1 \exists X_2 \ldots \exists X_n (f_1(\bar{Y}, \bar{X}) = 0 \wedge \cdots \wedge f_s(\bar{Y}, \bar{X}) = 0 \wedge g_1(\bar{Y}, \bar{X}) \neq 0 \wedge \cdots \wedge g_t(\bar{Y}, \bar{X}) \neq 0)$$

is equivalent to the following form with $g(\bar{Y}, \bar{X}) = g_1(\bar{Y}, \bar{X}) \cdots g_t(\bar{Y}, \bar{X})$.

$$\exists X_1 \exists X_2 \ldots \exists X_n (f_1(\bar{Y}, \bar{X}) = 0 \wedge \cdots \wedge f_s(\bar{Y}, \bar{X}) = 0 \wedge g(\bar{Y}, \bar{X}) \neq 0)$$

If we eliminate $\exists X_n$ from $\exists X_n (f_1(\bar{Y}, \bar{X}) = 0 \wedge \cdots \wedge f_s(\bar{Y}, \bar{X}) = 0 \wedge g(\bar{Y}, \bar{X}) \neq 0)$ and obtain an equivalent quantifier free formula, then by converting it to a $\vee\wedge$-canonical form we have basic formulas with quantifies $\exists X_1 \exists X_2 \ldots \exists X_{n-1}$. Therefore, as long as we give an algorithm for one quantifier, we

can recursively apply it to a general formula to eliminate all quantifiers. For the case $K = \mathbb{Q}$, in Mathematica packages Reduce and Resolve, this strategy is basically used with some sophisticated technique using Gröbner bases computations.

GCD-QE algorithm of Mathematica

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: The equivalent quantifier free formula

Step1. Compute the reduced Gröbner basis $G = \{g_1(\bar{Y}, X), \ldots, g_l(\bar{Y}, X), h_1(\bar{Y}), \ldots, h_t(\bar{Y})\}$ of $\langle f_1, \ldots, f_s\rangle$ w.r.t. a term order $X \gg \bar{Y}$.

Step2. Note that the given formula is false unless $h_1(\bar{Y}) = 0 \wedge \cdots \wedge h_t(\bar{Y}) = 0$. Let $\bar{c} \in \mathbb{C}^m$ be such that $h_1(\bar{c}) = 0 \wedge \cdots \wedge h_t(\bar{c}) = 0$. Note that for $\bar{Y} = \bar{c}$ the given formula is true if and only if $g(\bar{c}, X)$ does not belong to the radical ideal of $\langle f_1(\bar{c}, X), \ldots, f_s(\bar{c}, X)\rangle$. Considering each g_i as a unary polynomial of X, choose g_i which has the least degree. Let d be its degree and $p(\bar{Y})$ be its coefficient. If $p(\bar{c}) \neq 0$ then $\{g_i(\bar{c}, X)\}$ is a Gröbner basis of $\langle f_1(\bar{c}, X), \ldots, f_s(\bar{c}, X)\rangle$, which is an easy consequence of Theorem 1. In another word, $g_i(\bar{c}, X)$ is the GCD of $f_1(\bar{c}, X), \ldots, f_s(\bar{c}, X)$. Therefore, $g(\bar{c}, X)$ belongs to the radical ideal $\langle f_1(\bar{c}, X), \ldots, f_s(\bar{c}, X)\rangle$ if and only if the remainder of $g(\bar{c}, X)^d$ by $g_i(\bar{c}, X)$ is equal to 0. Compute the remainder of the pseudo division of $g(\bar{Y}, X)^d$ by $g_i(\bar{Y}, X)$, let $p_1(\bar{Y}), \ldots, p_r(\bar{Y})$ be its coefficients. When $p(\bar{c}) \neq 0$, the given formula for $\bar{Y} = \bar{c}$ is equivalent to $p_1(\bar{c}) \neq 0 \vee \cdots \vee p_r(\bar{c}) \neq 0$. When $p(\bar{c}) = 0$, we have to do another computation. For the new input formula $\exists X(f_1(\bar{Y}, X) = 0 \wedge \cdots \wedge f_s(\bar{Y}, X) = 0 \wedge p(\bar{Y}) = 0 \wedge g(\bar{Y}, X) \neq 0)$, proceed the above computation recursively and let $\phi(\bar{Y})$ be its output. Then the output for the original input is

$$\phi(\bar{Y}) \vee (h_1(\bar{Y}) = 0 \wedge \cdots \wedge h_t(\bar{Y}) = 0 \wedge p(\bar{Y}) \neq 0 \wedge (p_1(\bar{Y}) \neq 0 \vee \cdots \vee p_r(\bar{Y}) \neq 0)).$$

If we use the result of [3], $\{g_1(\bar{c}, X), \ldots, g_l(\bar{c}, X)\}$ is always a Gröbner basis. Hence, as long as it contains at least one non-zero polynomial, GCD is determined and we do not need any further recursive computation. Let $r_1(\bar{Y}), \ldots, r_k(\bar{Y})$ be an enumeration of all polynomials of $\mathbb{Q}[\bar{Y}]$ which appear as a coefficient of some polynomial among $g_1(\bar{Y}, X), \ldots, g_l(\bar{Y}, X)$. We need a further recursive computation only if $\langle r_1(\bar{Y}), \ldots, r_k(\bar{Y})\rangle \neq \langle 1\rangle$.

In our implementation of GCD-QE algorithm on Risa/Asir we use this strategy.

When we have many quantifiers, recursive use of this algorithm encounters a blowup of a search space. For the input formula,

$\exists x \exists y \exists z(x*y+a*x*z+y*z-1 = 0 \wedge x*y*z+x*z+x*y+a = 0 \wedge x*z+y*z-a*z-x-y-1 = 0)$

either of the following Mathematica inputs returns a complicated formula, although the given formula is always true. For checking it, we need further process of simplification.

```
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]]
```

```
Reduce[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],
       Complex]
```

For the input formula,

$\exists x \exists y \exists z(x*y+a*x*z+y*z-1 = 0 \wedge x*y*z+x*z+x*y+a = 0 \wedge x*z+y*z-b*z-x-y-1 = 0)$,

either of the above Mathematica programs or our GCD-QE implementation does not terminate. Note that the above examples contains no inequations. For such formulas, we have observed in practice that the CGS-QE algorithm is often more efficient.

4. CGS-QE algorithm

We can eliminate all quantifiers $\exists X_1 \exists X_2 \ldots \exists X_n$ simultaneously by computing only one CGS.

CGS-QE algorithm

Input: $\exists X_1 \exists X_2 \ldots \exists X_n(f_1(\bar{Y}, \bar{X}) = 0 \wedge \cdots \wedge f_s(\bar{Y}, \bar{X}) = 0 \wedge g_1(\bar{Y}, \bar{X}) \neq 0 \wedge \cdots \wedge g_t(\bar{Y}, \bar{X}) \neq 0)$

Output: The equivalent quantifier free formula

Let $\bar{Z} = Z_1, \ldots, Z_t$ be new variables. Compute a minimal CGS $\mathcal{G} = \{(G_1, P_1, Q_1), \ldots, (G_r, P_r, Q_r)\}$ of

$\{f_1(\bar{Y}, \bar{X}), \ldots, f_s(\bar{Y}, \bar{X}), g_1(\bar{Y}, \bar{X})Z_1 - 1, \ldots, g_t(\bar{Y}, \bar{X})Z_t - 1\}$ with parameters $\bar{Y}$ and main variables $\bar{X}, \bar{Z}$. We order G_i's, so that each $G_1, \ldots, G_k$ contains at least one polynomial including some main

variable, and each $G_{k+1}, \ldots, G_r$ contains only polynomials of parameters. When $k = r$, the output is true, otherwise the output formula is given by $\phi_1 \vee \cdots \vee \phi_k \vee \theta_{k+1} \vee \cdots \vee \theta_r$, where each ϕ_i and θ_j is given as follows. Let $P_i = \{p_1(\bar{Y}), \ldots, p_a(\bar{Y})\}, Q_i = \{q_1(\bar{Y}), \ldots, q_b(\bar{Y})\}$, then $\phi_i \equiv p_1(\bar{Y}) = 0 \wedge \cdots \wedge p_a(\bar{Y}) = 0 \wedge (q_1(\bar{Y}) \neq 0 \vee \cdots \vee q_b(\bar{Y}) \neq 0)$. For $j = k+1, \ldots, r$, let $P_j = \{p_1(\bar{Y}), \ldots, p_a(\bar{Y})\}, Q_j = \{q_1(\bar{Y}), \ldots, q_b(\bar{Y})\}$ and $G_j = \{h_1(\bar{Y}), \ldots, h_c(\bar{Y})\}$, then $\theta_j \equiv p_1(\bar{Y}) = 0 \wedge \cdots \wedge p_a(\bar{Y}) = 0 \wedge (q_1(\bar{Y}) \neq 0 \vee \cdots \vee q_b(\bar{Y}) \neq 0) \wedge h_1(\bar{Y}) = 0 \wedge \cdots \wedge h_c(\bar{Y}) = 0$.

According to our experiment, as long as this algorithm terminates, the output is more concise than GCD-QE algorithm. For the two examples of the previous section, our CGS program returns true within a second on a standard laptop computer. When we have many inequations in the given formula i.e. t is not small, however, we have to induce many new variables $\bar{Z}$. Though we can replace $g_1(\bar{Y}, \bar{X})Z_1 - 1, \ldots, g_t(\bar{Y}, \bar{X})Z_t - 1$ by a polynomial $g_1(\bar{Y}, \bar{X}) \cdots g_t(\bar{Y}, \bar{X})Z - 1$ with a single variable Z, we have to use a huge polynomial $g_1(\bar{Y}, \bar{X}) \cdots g_t(\bar{Y}, \bar{X})$. In such a case, CGS computation sometimes does not terminate.

5. Hybrid algorithm

In order to handle hard cases, we introduce a new algorithm which is a modification of CGS-QE algorithm partially using GCD-QE algorithm, we call it a hybrid algorithm.
Each of the practical algorithms of CGS introduced in [9, 6, 7, 10] is a modification of Suzuki-Sato's CGS algorithm [11]. In those algorithms, we incrementally divide parametric spaces, and proceed a Gröbner basis computation for each space in parallel. According to our experiments, when the CGS computation does not terminate, in many cases there are only a few Gröbner bases computations which do not terminate. For a quantifier elimination, we do not actually need a CGS. For a divided parametric space, if the Gröbner bases computation does not terminate, we can quit it and consider the original formula with the additional condition used for the divided parametric space. In the CGS algorithm of [10], the divided parametric space is given in a form of $\mathbb{V}(P) - \mathbb{V}(Q)$ for finite subsets P and Q of $K[\bar{Y}]$. In this parametric space, the original formula is equivalent to the following form:

$$\exists X_1 \exists X_2 \ldots \exists X_n (f_1(\bar{Y}, \bar{X}) = 0 \wedge \cdots \wedge f_s(\bar{Y}, \bar{X}) = 0 \wedge g_1(\bar{Y}, \bar{X}) \neq 0 \wedge \cdots \wedge g_t(\bar{Y}, \bar{X}) \neq 0 \\ \wedge p_1(\bar{Y}) = 0 \wedge \cdots \wedge p_a(\bar{Y}) = 0)$$

where $P = \{p_1(\bar{Y}) \ldots, p_a(\bar{Y})\}$. In our hybrid algorithm, we proceed GCD-QE algorithm to handle it. Since we have new extra conditions $p_1(\bar{Y}) = 0 \wedge \cdots \wedge p_a(\bar{Y}) = 0$, there is a much better chance that the computation terminates than the CGD-QE computation for the original formula. This rather simple idea leads us to a drastic improvement as described in the introduction.

6. Conclusion and Remarks

In the implementation of our hybrid algorithm, whenever we obtain a divided parametric space, we proceed both computations of CGS and GCD-QE in parallel, we adopt the computation which terminates first and quit the other one. We use only equations by P since the procedure of the next GCD-QE algorithm will be more complicated if we use inequations by Q. We might have a better implementation if we use inequations. For the quantifier elimination of a basic formula with recursive applications of GCD-QE algorithm, we also have two choices for each step. We, however, do not adopt CGS-QE algorithm for such a case in the hybrid algorithm, since the implementation becomes extremely complicated.

The characteristic set methods such as [12, 13, 14] are altarnatives to GCD-QE algorithm. They would be more efficient than GCD-QE algorithm, though we have not made an implementation yet.

References

[1] A computer algebra system Risa/Asir. http://www.math.kobe-u.ac.jp/Asir/asir.html

[2] Basu,S., Pollack,R. and Roy,M. Algorithms in Real Algebraic Geometry. Algorithms and Computation in Mathematics Volum 10, Springer.

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

[4] Harrison,J. Complex Quantifier Elimination in HOL.(2001). In Richard J. Boulton and Paul B.Jackson,editors,TPHOLs Supplemental Proceedings, pages 159174. Division of Informatics, University of Edinburgh, 2001. Published as Informatics Report Series EDI-INF-RR-0046. Available on the Web at http://www.informatics.ed.ac.uk/publications/report/0046.html.

[5] Kalkbrener, M. On the Stability of Gröbner Bases Under Specializations. *J. Symbolic Computation*. Vol. 24/1, pp. 51–58. 1997.

[6] 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.

[7] 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.

[8] Mathematica Tutorial: tutorial/ComplexPolynomialSystems

[9] 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.

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

[11] 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.

[12] Gao, X., Wang, D., 2003. Zero decomposition theorems for counting the number of solutions forparametric equation systems. In: Proc. ASCM 2003. pp. 129144

[13] Wang, D.,2004. The projection property of regular systems and its application to solving parametric polynomial systems. In: Dolzmann, A., Seidl, A., Sturm, T. (Eds.), Algorithmic Algebra and Logic. Herstellung und Verlag, Norderstedt, pp. 269274.

[14] Chen, C., Golubitsky, O., Lemaire, F., Moreno Maza, M., Pan, W., 2007. Comprehensive Triangular Decomposition. Vol. 4770 of LNCS. Springer Verlag, pp. 73101.

Ryoya Fukasaku, Shutaro Inoue and Yosuke Sato
Tokyo University of Science