On QE Algorithms over algebraically closed field

Ryoya Fukasaku, Shutaro Inoue and Yosuke Sato

Tokyo University of Science

MACIS2013

Contents of my talks

What is Quantifier Elimination over algebraically closed field(ACF QE) ?

ACF QE is computing the equivalent formula eliminating quantifier from a first-order formula over ACF.

Example

Input

$\exists x \in \mathbb{C}(x - a_1 = 0 \wedge x - a_2 = 0 \wedge x - a_1 a_2 \neq 0)$

Output

$a_1 = a_2 \wedge -a_2 + a_2^2 \neq 0$

What is Quantifier Elimination over algebraically closed field(ACF QE) ?

ACF QE is computing the equivalent formula eliminating quantifier from a first-order formula over ACF.

Example

Input

$\exists x \in \mathbb{C}(x - a_1 = 0 \wedge x - a_2 = 0 \wedge x - a_1 a_2 \neq 0)$

Output

$a_1 = a_2 \wedge -a_2 + a_2^2 \neq 0$

Basic formulas

K : field, $\overline{K}$: the algebraic closure of K,
$\overline{A}$: free variables $A_1, \ldots, A_n$, $\overline{X}$: quantified variables $X_1, \ldots, X_m$,
$f_1, \ldots, f_r, g_1, \ldots, g_s \in K[\overline{A}, \overline{X}]$

Basic formula

$\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$

Basic formula

$\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \ldots g_s \neq 0)$

The above formulas are equivalent.
General ACF QE can return by QE the above basic formulas.

Basic formulas

K : field, $\overline{K}$: the algebraic closure of K,
$\overline{A}$: free variables $A_1, \ldots, A_n$, $\overline{X}$: quantified variables $X_1, \ldots, X_m$,
$f_1, \ldots, f_r, g_1, \ldots, g_s \in K[\overline{A}, \overline{X}]$

Basic formula

$\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$

Basic formula

$\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \ldots g_s \neq 0)$

The above formulas are equivalent.
General ACF QE can return by QE the above basic formulas.

Existing algorithms

The existing algorithms of ACF QE :

1. Method based on greatest common divisor (GCD-QE)
2. Method based on comprehensive Groebner system (CGS-QE)
3. Method based on characteristic sets and regular chains(CSRC-QE)

The problems of the existing algorithms :

- Computation speed
  The computation speed of GCD-QE and CGS-QE is slow.
- Representation of output result
  The representation of output result of GCD-QE and CSRC-QE is generally complicated.

Existing algorithms

The existing algorithms of ACF QE :

1. Method based on greatest common divisor (GCD-QE)
2. Method based on comprehensive Groebner system (CGS-QE)
3. Method based on characteristic sets and regular chains(CSRC-QE)

The problems of the existing algorithms :

- Computation speed
  The computation speed of GCD-QE and CGS-QE is slow.
- Representation of output result
  The representation of output result of GCD-QE and CSRC-QE is generally complicated.

Merits of each existing algorithms

- GCD-QE
  The computation of ACF QE for one segment is often fast.

- CGS-QE
  By recent research we can compute a CGS with almost a minimum number of segments, which give us a very simple QE formula.

- CSRC-QE
  The computation of ACF QE is often fast.

Demerits of each existing algorithms

- GCD-QE
  We generally have a huge number of segments which makes the QE formula complicated, further we also need long computation time.

- CGS-QE
  We have to use new variables for inequations, which sometimes makes computation very heavy.

- CSRC-QE
  The representation of output result is generally complicated.

Demerits of each existing algorithms

Example

$\exists(x,y,z)\in\mathbb{C}^3$
$axz+xy+yz=0\wedge axyz+axy+axz+1=0\wedge axz-az+yz-x-y=0$

- Output of GCD-QE :
  $a^2-3a+3=0\vee a\neq 0\vee(16a^8-144a^7+504a^6-864a^5+729a^4-135a^3-324a^2+405a-81=0\wedge -a^2+3a-3\neq 0)$
- Output of CGS-QE :
  $a\neq 0$
- Output of CSRC-QE :
  $a(a+1)(a^2-3a+3)\neq 0\vee a^2-3a+3=0\vee a+1=0$

The output of CGS-QE is the most simple of 3 outputs.

Demerits of each existing algorithms

Example

$\exists(x,y,z) \in \mathbb{C}^3$
$axz+xy+yz=0 \wedge axyz+axy+axz+1=0 \wedge axz-az+yz-x-y=0$

- Output of GCD-QE :
  $a^2-3a+3=0 \vee a \neq 0 \vee (16a^8-144a^7+504a^6-864a^5+729a^4-135a^3-324a^2+405a-81=0 \wedge -a^2+3a-3 \neq 0)$
- Output of CGS-QE :
  $a \neq 0$
- Output of CSRC-QE :
  $a(a+1)(a^2-3a+3) \neq 0 \vee a^2-3a+3=0 \vee a+1=0$

The output of CGS-QE is the most simple of 3 outputs.

Today Main Theme

Today we introduce Hybrid-QE for the following point.

- Return fast.
- Return simple output.

Definitions

$m \in \mathbb{N}$, $S_1, \ldots, S_t \subseteq \overline{K}^m$

Definition

$\{S_1, \ldots, S_t\}$ is a partition of $\overline{K}^m$.
$:\Leftrightarrow (\forall i, j (i \neq j \Rightarrow S_i \cap S_j = \emptyset)) \wedge (S_1 \cup \ldots \cup S_t = \overline{K}^m)$

- We call each S_i segment.
- S_i is represented by a set which subtracts a variety from a variety.
- We identify S_i with its defining formula.

$F, G_1, \ldots, G_t$: finite subsets of $K[\overline{A}, \overline{X}]$, $\{S_1, \ldots, S_t\}$: a partition of $\overline{K}^m$

Definition

$\{(S_1, G_1), \ldots, (S_t, G_t)\}$ is a CGS of $\langle F \rangle$.
$:\Leftrightarrow \forall \overline{a} \in S_i \; G_i(\overline{a})$ is a Groebner basis(GB) of $\langle F(\overline{a}) \rangle$ for each i
, where $F(\overline{a}) = \{f(\overline{a}, \overline{X}) : f \in F\} \subset \overline{K}[\overline{X}]$.

Definitions

$m \in \mathbb{N}$, $S_1, \ldots, S_t \subseteq \overline{K}^m$

Definition

$\{S_1, \ldots, S_t\}$ is a partition of $\overline{K}^m$.
$:\Leftrightarrow (\forall i, j(i \neq j \Rightarrow S_i \cap S_j = \emptyset)) \wedge (S_1 \cup \ldots \cup S_t = \overline{K}^m)$

- We call each S_i segment.
- S_i is represented by a set which subtracts a variety from a variety.
- We identify S_i with its defining formula.

$F, G_1, \ldots, G_t$: finite subsets of $K[\overline{A}, \overline{X}]$, $\{S_1, \ldots, S_t\}$: a partition of $\overline{K}^m$

Definition

$\{(S_1, G_1), \ldots, (S_t, G_t)\}$ is a CGS of $\langle F \rangle$.
$:\Leftrightarrow \forall \overline{a} \in S_i$ $G_i(\overline{a})$ is a Groebner basis(GB) of $\langle F(\overline{a}) \rangle$ for each i
, where $F(\overline{a}) = \{f(\overline{a}, \overline{X}) : f \in F\} \subset \overline{K}[\overline{X}]$.

Definitions

$m \in \mathbb{N}$, $S_1, \ldots, S_t \subseteq \overline{K}^m$

Definition

$\{S_1, \ldots, S_t\}$ is a partition of $\overline{K}^m$.
$:\Leftrightarrow (\forall i, j (i \neq j \Rightarrow S_i \cap S_j = \emptyset)) \wedge (S_1 \cup \ldots \cup S_t = \overline{K}^m)$

- We call each S_i segment.
- S_i is represented by a set which subtracts a variety from a variety.
- We identify S_i with its defining formula.

$F, G_1, \ldots, G_t$: finite subsets of $K[\overline{A}, \overline{X}]$, $\{S_1, \ldots, S_t\}$: a partition of $\overline{K}^m$

Definition

$\{(S_1, G_1), \ldots, (S_t, G_t)\}$ is a CGS of $\langle F \rangle$.
$:\Leftrightarrow \forall \overline{a} \in S_i$ $G_i(\overline{a})$ is a Groebner basis(GB) of $\langle F(\overline{a}) \rangle$ for each i
, where $F(\overline{a}) = \{f(\overline{a}, \overline{X}) : f \in F\} \subset \overline{K}[\overline{X}]$.

Lemma

$f_1, \ldots, f_r, g_1, \ldots, g_s \in K[\overline{A}, \overline{X}]$

Lemma

The following formulas is equivalent.

- $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$
- $\neg(\forall \overline{X} \in \overline{K}^n (g_1 \ldots g_s \in \sqrt{\langle f_1, \ldots, f_r \rangle}))$
- $\exists (\overline{Z}, \overline{X}) \in \overline{K}^{s+n} (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge 1 - Z_1 g_1 = 0 \wedge \ldots \wedge 1 - Z_s g_s = 0)$

GCD-QE algorithm

X : a quantified variable, $f_1, \ldots, f_r, g \in K[\overline{A}, X]$

Basic formula

$\exists X \in \overline{K}(f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g \neq 0)$

1. Compute parametric GCD's s.t.
   $g_i(\overline{a}, X) :=$GCD$(f_1(\overline{a}, X), \ldots, f_r(\overline{a}, X))$
   for $\overline{a} \in S_i$, where $\{S_1, \ldots, S_t\}$ is a partition of $\overline{K}^m$
2. Refine each segment to S_i' s.t. $\neg(\forall \overline{a} \in S_i \forall X \in \overline{K}(g \in \sqrt{\langle g_i(\overline{a}, X)\rangle}))$;
3. Return $\cup S_i'$;

We can eliminate many quantified variables by recursive application.

GCD-QE algorithm

X : a quantified variable, $f_1, \ldots, f_r, g \in K[\overline{A}, X]$

Basic formula

$\exists X \in \overline{K}(f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g \neq 0)$

1. Compute parametric GCD's s.t.
   $g_i(\overline{a}, X) :=$GCD$(f_1(\overline{a}, X), \ldots, f_r(\overline{a}, X))$
   for $\overline{a} \in S_i$, where $\{S_1, \ldots, S_t\}$ is a partition of $\overline{K}^m$
2. Refine each segment to S_i' s.t. $\neg(\forall \overline{a} \in S_i \forall X \in \overline{K}(g \in \sqrt{\langle g_i(\overline{a}, X)\rangle}))$;
3. Return $\cup S_i'$;

We can eliminate many quantified variables by recursive application.

GCD-QE algorithm

X : a quantified variable, $f_1, \ldots, f_r, g \in K[\overline{A}, X]$

Basic formula

$\exists X \in \overline{K}(f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g \neq 0)$

1. Compute parametric GCD's s.t.
   $g_i(\overline{a}, X) :=$GCD$(f_1(\overline{a}, X), \ldots, f_r(\overline{a}, X))$
   for $\overline{a} \in S_i$, where $\{S_1, \ldots, S_t\}$ is a partition of $\overline{K}^m$
2. Refine each segment to S_i' s.t. $\neg(\forall \overline{a} \in S_i \forall X \in \overline{K}(g \in \sqrt{\langle g_i(\overline{a}, X)\rangle}))$;
3. Return $\cup S_i'$;

We can eliminate many quantified variables by recursive application.

CGS-QE algorithm

$\overline{X}$: quantified variables, $f_1, \ldots, f_r, g_1, \ldots, g_s \in K[\overline{A}, \overline{X}]$

Basic formula

$\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$

1. Introduce new variables $Z_1, \ldots, Z_s$;
2. Let $I = \langle f_1, \ldots, f_r, 1 - Z_1 g_1, \ldots, 1 - Z_s g_s \rangle$, $R = \emptyset$;
3. Compute a CGS $\mathcal{G}$ of I w.r.t. graded reverse lexicographic order(GRL);
4. For $(S_i, G_i) \in \mathcal{G}$,
   if $G_i(\overline{a})$ doesn't contain non-zero constant for $\overline{a} \in S_i$, then $R = R \cup S_i$;
5. Return R;

CGS-QE algorithm

$\overline{X}$: quantified variables, $f_1, \ldots, f_r, g_1, \ldots, g_s \in K[\overline{A}, \overline{X}]$

Basic formula

$\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$

1. Introduce new variables $Z_1, \ldots, Z_s$;
2. Let $I = \langle f_1, \ldots, f_r, 1 - Z_1 g_1, \ldots, 1 - Z_s g_s \rangle$, $R = \emptyset$;
3. Compute a CGS $\mathcal{G}$ of I w.r.t. graded reverse lexicographic order(GRL);
4. For $(S_i, G_i) \in \mathcal{G}$,
   if $G_i(\overline{a})$ doesn't contain non-zero constant for $\overline{a} \in S_i$, then $R = R \cup S_i$;
5. Return R;

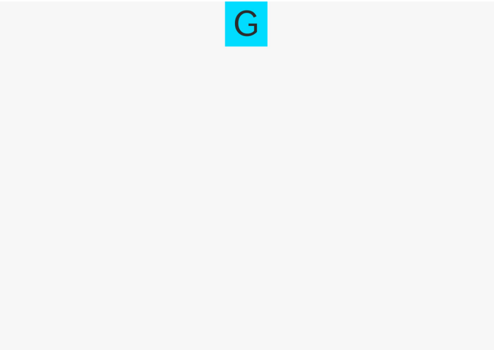
Suzuki-Sato's CGS original algorithm

input : finite $F \subset K[\overline{A}, \overline{X}]$,
a term order $<_{\overline{X}}$ on $T(\overline{X})$,
the term order $<_{\overline{A},\overline{X}}$ on $T(\overline{A}, \overline{X})$ s.t. $\overline{A} \ll \overline{X}$ which extends $<_{\overline{X}}$;
output : CGS of $\langle F \rangle$ w.r.t. $<_{\overline{X}}$;

Suzuki-Sato's CGS original algorithm

input : finite $F \subset K[\overline{A}, \overline{X}]$,
a term order $<_{\overline{X}}$ on $T(\overline{X})$,
the term order $<_{\overline{A},\overline{X}}$ on $T(\overline{A}, \overline{X})$ s.t. $\overline{A} \ll \overline{X}$ which extends $<_{\overline{X}}$;
output : CGS of $\langle F \rangle$ w.r.t. $<_{\overline{X}}$;

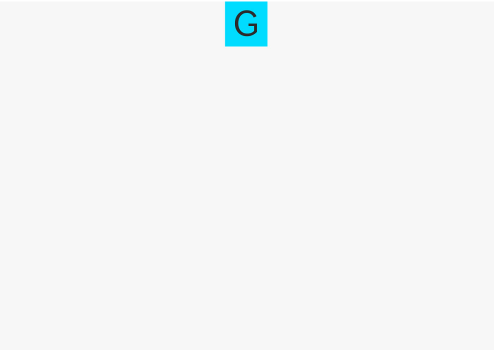
Compute a reduced GB G of $\langle F \rangle$ w.r.t. $<_{\overline{A},\overline{X}}$ in $K[\overline{A}, \overline{X}]$.



Suzuki-Sato's CGS original algorithm

input : finite $F \subset K[\overline{A}, \overline{X}]$,
a term order $<_{\overline{X}}$ on $T(\overline{X})$,
the term order $<_{\overline{A},\overline{X}}$ on $T(\overline{A}, \overline{X})$ s.t. $\overline{A} \ll \overline{X}$ which extends $<_{\overline{X}}$;
output : CGS of $\langle F \rangle$ w.r.t. $<_{\overline{X}}$;

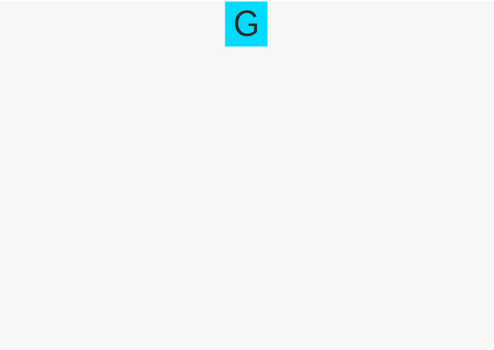
Let $\{h_1, \ldots, h_s\} = \{hc(g) \in K[\overline{A}] : g \in G \setminus K[\overline{A}]\}$.



Suzuki-Sato's CGS original algorithm

input : finite $F \subset K[\overline{A}, \overline{X}]$,
a term order $<_{\overline{X}}$ on $T(\overline{X})$,
the term order $<_{\overline{A},\overline{X}}$ on $T(\overline{A}, \overline{X})$ s.t. $\overline{A} \ll \overline{X}$ which extends $<_{\overline{X}}$;
output : CGS of $\langle F \rangle$ w.r.t. $<_{\overline{X}}$;

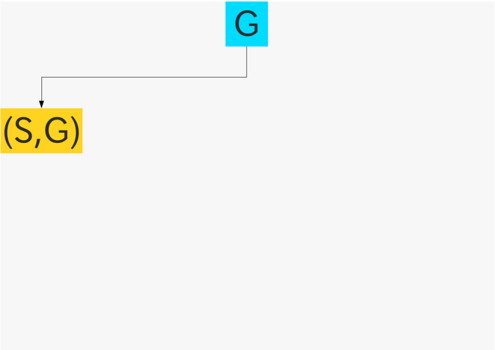
Let $S = \mathbf{V}(G \cap K[\overline{A}]) \setminus \mathbf{V}(\mathrm{LCM}(h_1, \ldots, h_s))$.



Suzuki-Sato's CGS original algorithm

input : finite $F \subset K[\overline{A}, \overline{X}]$,
a term order $<_{\overline{X}}$ on $T(\overline{X})$,
the term order $<_{\overline{A},\overline{X}}$ on $T(\overline{A}, \overline{X})$ s.t. $\overline{A} \ll \overline{X}$ which extends $<_{\overline{X}}$;
output : CGS of $\langle F \rangle$ w.r.t. $<_{\overline{X}}$;

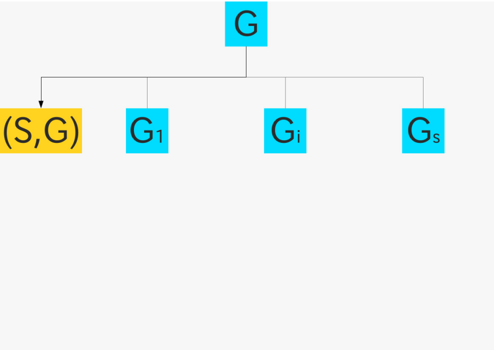
$G(\overline{a})$ is a GB w.r.t. $<_{\overline{X}}$ for $\overline{a} \in S$.



Suzuki-Sato's CGS original algorithm

input : finite $F \subset K[\overline{A}, \overline{X}]$,
a term order $<_{\overline{X}}$ on $T(\overline{X})$,
the term order $<_{\overline{A},\overline{X}}$ on $T(\overline{A}, \overline{X})$ s.t. $\overline{A} \ll \overline{X}$ which extends $<_{\overline{X}}$;
output : CGS of $\langle F \rangle$ w.r.t. $<_{\overline{X}}$;

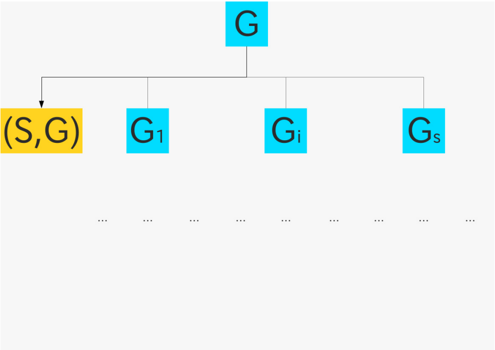
Compute a reduced GB G_i of $\langle F, h_i \rangle$ w.r.t. $<_{\overline{A},\overline{X}}$ in $K[\overline{A}, \overline{X}]$ for each i.



Suzuki-Sato's CGS original algorithm

input : finite $F \subset K[\overline{A}, \overline{X}]$,
a term order $<_{\overline{X}}$ on $T(\overline{X})$,
the term order $<_{\overline{A},\overline{X}}$ on $T(\overline{A}, \overline{X})$ s.t. $\overline{A} \ll \overline{X}$ which extends $<_{\overline{X}}$;
output : CGS of $\langle F \rangle$ w.r.t. $<_{\overline{X}}$;

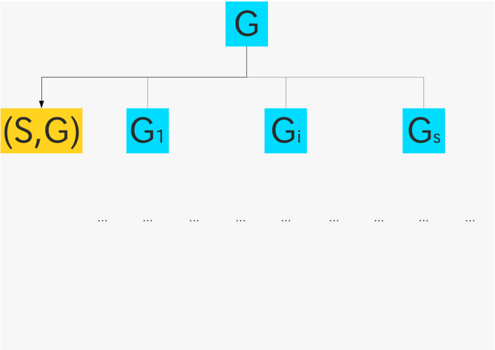
The following is computed similarly.



Suzuki-Sato's CGS original algorithm

input : finite $F \subset K[\overline{A}, \overline{X}]$,
a term order $<_{\overline{X}}$ on $T(\overline{X})$,
the term order $<_{\overline{A},\overline{X}}$ on $T(\overline{A}, \overline{X})$ s.t. $\overline{A} \ll \overline{X}$ which extends $<_{\overline{X}}$;
output : CGS of $\langle F \rangle$ w.r.t. $<_{\overline{X}}$;

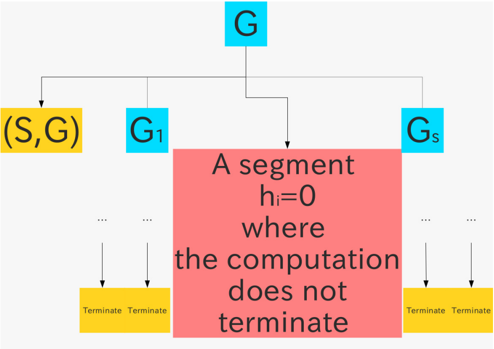
This algorithm is structure like pyramid.



Suzuki-Sato's CGS original algorithm

When the whole algorithm does not terminate,
we sometimes have only a few segments
where the computation does not terminate.

The following is an example that we have a segment $h_i = 0$ where the computation does not terminate.



Algorithm

input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$;
output : ACF QE formula;

Algorithm

input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \land \ldots \land f_r = 0 \land g_1 \neq 0 \land \ldots \land g_s \neq 0)$;
output : ACF QE formula;

Introduce new variables $Z_1, \ldots, Z_s$.

input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$;
output : ACF QE formula;

Let $F = \{f_1, \ldots, f_r, 1 - Z_1 g_1, \ldots, 1 - Z_s g_s\}$.

Algorithm

input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$;
output : ACF QE formula;

Let $<$ be GRL in $T(\overline{Z}, \overline{X})$.

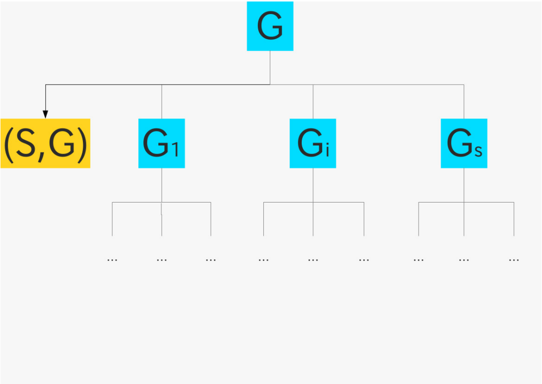
Algorithm

input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$;
output : ACF QE formula;

Let $<'$ be an order in $T(\overline{Z}, \overline{X}, \overline{A})$ satisfying $\overline{A} \ll \{\overline{Z}, \overline{X}\}$ and extending $<$.

input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$;
output : ACF QE formula;

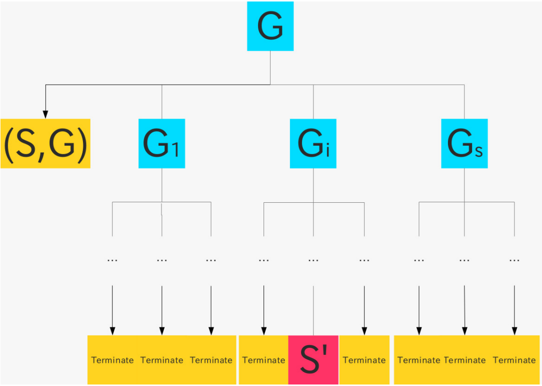
Apply Suzuki-Sato's CGS algorithm to F,$<$,$<'$.



Algorithm

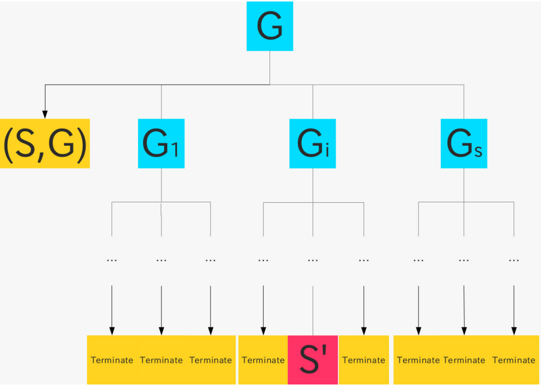
input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$;
output : ACF QE formula;

Let S' be a segment where the computation does not terminate.



input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$;
output : ACF QE formula;

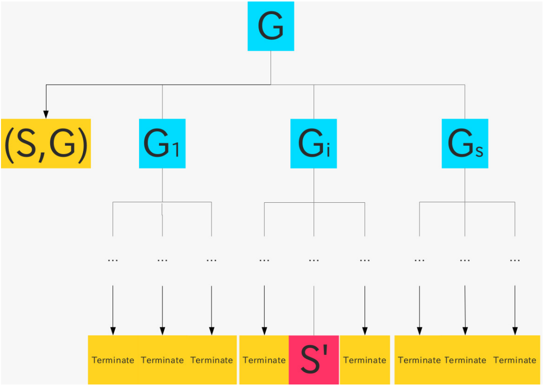
Let $Q = \exists \overline{X} \in \overline{K}^n (S' \wedge f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \ldots g_s \neq 0)$.



Algorithm

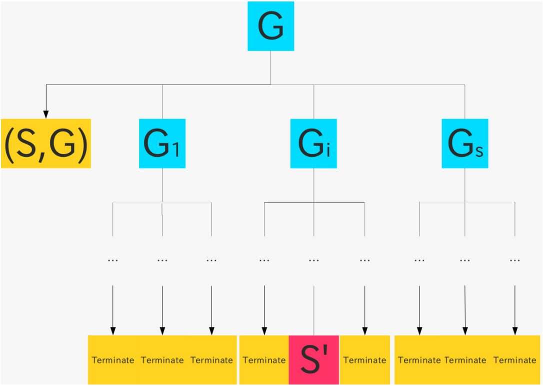
input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$;
output : ACF QE formula;

Apply GCD-QE to Q.



input : $\exists \overline{X} \in \overline{K}^n (f_1 = 0 \wedge \ldots \wedge f_r = 0 \wedge g_1 \neq 0 \wedge \ldots \wedge g_s \neq 0)$;
output : ACF QE formula;

For the other segments, follow CGS-QE.



Algorithm

Merits :

- Minimum number of segments
  We can have the partition which is almost a minimum number of segments by using CGS-QE.

- Segments where the computation of CGS does not terminate
  GCD-QE does not introduce new variables.
  Even when GCD-QE does not terminate on whole space, it often terminates on a segment.

Example

Example

$f_1 := AX + 2, f_2 := X + BY - AY + 1, g := AX + 1$
$\exists(X, Y) \in \mathbb{C}^2(f_1 = 0 \wedge f_2 = 0 \wedge g \neq 0)$

Of course the computation of applying any existing algorithms to the above example terminates, but we apply Hybrid-QE.

- We introduce a new variable Z.
- Let $F = \{f_1, f_2, 1 - Zg\}$.
- Let $<_{Z,X,Y,A,B}$ be block order which is GRL on $T(Z, X, Y)$ s.t. $Z > X > Y$ and GRL on $T(A, B)$ s.t. $A > B$ and satisfying $\{Z, X, Y\} \gg \{A, B\}$.

Example

$f_1 := AX + 2, f_2 := X + BY - AY + 1, g := AX + 1$
$\exists (X, Y) \in \mathbb{C}^2 (f_1 = 0 \wedge f_2 = 0 \wedge g \neq 0)$

Of course the computation of applying any existing algorithms to the above example terminates, but we apply Hybrid-QE.

- We introduce a new variable Z.
- Let $F = \{f_1, f_2, 1 - Zg\}$.
- Let $<_{Z,X,Y,A,B}$ be block order which is GRL on $T(Z, X, Y)$ s.t. $Z > X > Y$ and GRL on $T(A, B)$ s.t. $A > B$ and satisfying $\{Z, X, Y\} \gg \{A, B\}$.

Example

Example

$f_1 := AX + 2, f_2 := X + BY - AY + 1, g := AX + 1$
$\exists (X, Y) \in \mathbb{C}^2 (f_1 = 0 \wedge f_2 = 0 \wedge g \neq 0)$

Of course the computation of applying any existing algorithms to the above example terminates, but we apply Hybrid-QE.

- We introduce a new variable Z.
- Let $F = \{f_1, f_2, 1 - Zg\}$.
- Let $<_{Z,X,Y,A,B}$ be block order which is GRL on $T(Z, X, Y)$ s.t. $Z > X > Y$ and GRL on $T(A, B)$ s.t. $A > B$ and satisfying $\{Z, X, Y\} \gg \{A, B\}$.

Example

$f_1 := AX + 2, f_2 := X + BY - AY + 1, g := AX + 1$
$\exists (X, Y) \in \mathbb{C}^2 (f_1 = 0 \wedge f_2 = 0 \wedge g \neq 0)$

Of course the computation of applying any existing algorithms to the above example terminates, but we apply Hybrid-QE.

- We introduce a new variable Z.
- Let $F = \{f_1, f_2, 1 - Zg\}$.
- Let $<_{Z,X,Y,A,B}$ be block order which is GRL on $T(Z, X, Y)$ s.t. $Z > X > Y$ and GRL on $T(A, B)$ s.t. $A > B$ and satisfying $\{Z, X, Y\} \gg \{A, B\}$.

Example

- Compute a reduce GB G of $\langle F \rangle$ w.r.t.$<_{Z,X,Y,A,B}$ in $\mathbb{Q}[Z, X, Y, A, B]$.
  - $G = \{A^2Y - BAY - A + 2, X - AY + BY + 1, Z + 1\}$
  - The set consisting of the head coefficients of $G = \{A^2 - BA, 1, 1\}$
  - LCM($A^2 - BA, 1, 1$)= $A^2 - BA$
  - $G(a, b)$ is a GB for $(a, b) \in \mathbb{C}^2 \setminus \mathbf{V}(A^2 - BA)$.
  - $G(a, b) \cap (\mathbb{Q} \setminus \{0\}) = \emptyset$ for $(a, b) \in \mathbb{C}^2 \setminus \mathbf{V}(A^2 - BA)$
- If the computation on the segment $\mathbf{V}(A^2 - BA)$ does not terminate, then we apply the following :
  - let $Q = \exists(X, Y) \in \mathbb{C}^2(A^2 - BA = 0 \wedge f_1 = 0 \wedge f_2 = 0 \wedge g \neq 0)$,
  - apply GCD-QE to Q,

    where the return is the segment $\mathbf{V}(-A + 2, B - 2)$.
- Return $(\mathbb{C}^2 \setminus \mathbf{V}(A^2 - BA)) \cup \mathbf{V}(-A + 2, B - 2)$

Example

- Compute a reduce GB G of $\langle F \rangle$ w.r.t.$<_{Z,X,Y,A,B}$ in $\mathbb{Q}[Z,X,Y,A,B]$.
  - $G = \{A^2Y - BAY - A + 2, X - AY + BY + 1, Z + 1\}$
  - The set consisting of the head coefficients of $G = \{A^2 - BA, 1, 1\}$
  - LCM($A^2 - BA, 1, 1$)= $A^2 - BA$
  - $G(a,b)$ is a GB for $(a,b) \in \mathbb{C}^2 \setminus \mathbf{V}(A^2 - BA)$.
  - $G(a,b) \cap (\mathbb{Q} \setminus \{0\}) = \emptyset$ for $(a,b) \in \mathbb{C}^2 \setminus \mathbf{V}(A^2 - BA)$
- If the computation on the segment $\mathbf{V}(A^2 - BA)$ does not terminate, then we apply the following :
  - let $Q = \exists (X,Y) \in \mathbb{C}^2 (A^2 - BA = 0 \land f_1 = 0 \land f_2 = 0 \land g \neq 0)$,
  - apply GCD-QE to Q,

    where the return is the segment $\mathbf{V}(-A + 2, B - 2)$.
- Return $(\mathbb{C}^2 \setminus \mathbf{V}(A^2 - BA)) \cup \mathbf{V}(-A + 2, B - 2)$

Example

- Compute a reduce GB G of $\langle F \rangle$ w.r.t.$<_{Z,X,Y,A,B}$ in $\mathbb{Q}[Z,X,Y,A,B]$.
    - $G = \{A^2Y - BAY - A + 2, X - AY + BY + 1, Z + 1\}$
    - The set consisting of the head coefficients of $G = \{A^2 - BA, 1, 1\}$
    - LCM($A^2 - BA, 1, 1$)= $A^2 - BA$
    - $G(a,b)$ is a GB for $(a,b) \in \mathbb{C}^2 \setminus \mathbf{V}(A^2 - BA)$.
    - $G(a,b) \cap (\mathbb{Q} \setminus \{0\}) = \emptyset$ for $(a,b) \in \mathbb{C}^2 \setminus \mathbf{V}(A^2 - BA)$
- If the computation on the segment $\mathbf{V}(A^2 - BA)$ does not terminate, then we apply the following :
    - let $Q = \exists (X,Y) \in \mathbb{C}^2 (A^2 - BA = 0 \wedge f_1 = 0 \wedge f_2 = 0 \wedge g \neq 0)$,
    - apply GCD-QE to Q,
      where the return is the segment $\mathbf{V}(-A + 2, B - 2)$.
- Return $(\mathbb{C}^2 \setminus \mathbf{V}(A^2 - BA)) \cup \mathbf{V}(-A + 2, B - 2)$

Computation experiment

We applied ACF QE to a number of experiments.
We checked about 100 examples that CGS-QE does not terminate, but Hybrid-QE terminates, neither of the other algorithms terminates.

One example

$f_1 := (AX+BY)^{26}-1, f_2 := (AXY+BX+CY)^{26}-B, g := AX+BY$
$\exists (X,Y) \in \mathbb{C}^2 (f_1 = 0 \wedge f_2 = 0 \wedge g \neq 0)$

system	program	time
Mathematica	Reduce	$>$ 1 hour
Mathematica	Resolve	$>$ 1 hour
Maple	Projection	$>$ 1 hour
risa/asir	our implementation of GCD-QE	$>$ 1 hour
risa/asir	our implementation of CGS-QE	out of memory
risa/asir	our implementation of Hybrid-QE	139.7 sec.

Computation experiment

We applied ACF QE to a number of experiments.
We checked about 100 examples that CGS-QE does not terminate, but Hybrid-QE terminates, neither of the other algorithms terminates.

One example

$f_1 := (AX + BY)^{26} - 1, f_2 := (AXY + BX + CY)^{26} - B, g := AX + BY$
$\exists (X, Y) \in \mathbb{C}^2 (f_1 = 0 \wedge f_2 = 0 \wedge g \neq 0)$

system	program	time
Mathematica	Reduce	$>$ 1 hour
Mathematica	Resolve	$>$ 1 hour
Maple	Projection	$>$ 1 hour
risa/asir	our implementation of GCD-QE	$>$ 1 hour
risa/asir	our implementation of CGS-QE	out of memory
risa/asir	our implementation of Hybrid-QE	139.7 sec.

Computation experiment

The following computation terminates by only using CSRC-QE and Hybrid-QE.

One example

$f_1 := AXZ + BX - 1, f_2 := (BX + CY)^{14} - 1, g := AX + BZ$
$\exists(X, Y, Z) \in \mathbb{C}^3(f_1 = 0 \wedge f_2 = 0 \wedge g \neq 0)$

Output :

- CSRC-QE
  $(ABC \neq 0) \vee (C(A^2 + B^3) \neq 0) \vee (C = 0 \wedge AB(A^{12} + 7A^4B^{12} - 14A^2B^{15} + 7B^{18}) \neq 0) \vee (C = 0 \wedge AB(A^{12} - 2A^{10}B^3 + 4A^8B^6 - 8A^6B^9 + 9A^4B^{12} - 4A^2B^{15} + B^{18}) \neq 0) \vee (C = 0 \wedge AB(A^2 + 2B^3) \neq 0) \vee (C = 0 \wedge AB \neq 0) \vee (A = 0 \wedge C = 0 \wedge B \neq 0) \vee (A^{12} - 2A^{10}B^3 + 4A^8B^6 - 8A^6B^9 + 9A^4B^{12} - 4A^2B^{15} + B^{18} = 0 \wedge C = 0 \wedge AB(47A^{10} - 284A^8B^3 + 568A^6B^6 - 519A^4B^9 + 214A^2B^{12} - 47B^{15})(94A^{10} + 117A^8B^3 - 783A^6B^6 + 1017A^4B^9 - 468A^2B^{12} + 117B^{15})(3844755A^{10} - 9231137A^8B^3 + 7214722A^6B^6 - 403976A^4B^9 - 832313A^2B^{12} + 474788B^{15}) \neq 0) \vee (A^{12} + 7A^4B^{12} - 14A^2B^{15} + 7B^{18} = 0 \wedge C = 0 \wedge AB(42701A^{10} - 346432A^8B^3 + 896904A^6B^6 - 1411539A^4B^9 + 1193297A^2B^{12} - 396739B^{15})(69310A^{10} - 221942A^8B^3 + 412158A^6B^6 - 411014A^4B^9 + 176253A^2B^{12} - 17157B^{15})(2186507864386A^{10} - 2706446731217A^8B^3 - 61230596433A^6B^6 + 10476412105940A^4B^9 - 16403396742588A^2B^{12} + 7291066799632B^{15}) \neq 0)$
- Hybrid-QE
  $(C = 0 \wedge AB \neq 0) \vee (B = 0 \wedge AC \neq 0) \vee (A = 0 \wedge C = 0 \wedge B \neq 0) \vee (A = 0 \wedge BC \neq 0) \vee (ABC \neq 0)$

Conclusions

- Hybrid-QE

  We proposed hybrid-QE by using GCD-QE and CGS-QE.

- Experiments

  The output of Hybrid-QE is simple.

  There are many examples which only Hybrid-QE terminates.

Future works

- Parallel computation

  We have many possibilities of parallel computation.

- CSRC-QE

  For Hybrid-QE we may apply CSRC-QE instead of GCD-QE.

Thank you for your kind attention!

謝謝!