Generic Hermitian Quantifier Elimination

Andreas Dolzmann[1] and Lorenz A. Gilch[2]

[1] University of Passau, Passau, Germany
dolzmann@uni-passau.de,
WWW home page: http://www.fmi.uni-passau.de/~dolzmann/
[2] Graz University of Technology, Graz, Austria
gilch@TUGraz.at,
WWW home page: http://people.freenet.de/lorenzg/

Abstract. We present a new method for generic quantifier elimination that uses an extension of Hermitian quantifier elimination. By means of sample computations we show that this generic Hermitian quantifier elimination is, for instance, an important method for automated theorem proving in geometry.

1 Introduction

Ever since quantifier elimination by CAD has been implemented by Collins [1], quantifier elimination has become more and more important. This was reinforced especially by the development and implementation of the partial CAD [2] by Hong and later Brown [3]. Beside the approach used there, quantifier elimination by virtual substitution was published by Weispfenning [4, 5] and further developed and implemented by the first author together with Sturm in REDLOG [6]. The latter method can only be applied to degree restricted formulas. Although both methods are implemented highly efficiently, none is superior to the other one. Moreover sometimes the methods fail solving problems which seem to be solvable using quantifier elimination. Therefore it is necessary to develop and implement further quantifier elimination algorithms.

The quantifier elimination by real root counting was published by Weispfenning in 1998 [7], although he had already published in 1993 a technical report describing this method. The algorithm was first implemented by the first author in 1994 as a diploma thesis [8] in the computer algebra system MAS. Numerous new optimizations were developed by the authors. They were implemented by the second author [9] in a complete reimplementation of the method in the package REDLOG [6] of the computer algebra system REDUCE . The improved version of this quantifier elimination is called Hermitian quantifier elimination. The name "Hermitian" quantifier elimination was chosen to acknowledge the influence of Hermite's work in the area of real root counting.

Hermitian quantifier elimination has been proved to be a powerful tool for particular classes of elimination problems. In [10] the first author has used it for the automatic solution of a real algebraic implicitization problem by quantifier elimination. For this automatic solution he has used all three quantifier elimination methods, namely quantifier elimination by virtual substitution, Hermitian quantifier elimination, and quantifier elimination by partial cylindrical algebraic decomposition as well as the simplification methods described in [11].

The definition, development and implementation of new paradigms related to quantifier elimination algorithms have been very successful in the past. Extended quantifier elimination provides not only an equivalent formula but sample solutions. It can be applied e.g. in the area of generalized constraint solving yielding optimal solutions [12].

Generic quantifier elimination was introduced for the virtual substitution method by the first author together with Sturm and Weispfenning [13]. Let φ be an input formula with quantified variables $x_1, \dots, x_n$ and parameters $u_1, \dots, u_m$. Recall that regular quantifier elimination computes from φ a quantifier-free formula φ' such that for all real values $c_1, \dots, c_m$ for the parameters $u_1, \dots, u_m$ both φ and φ' are equivalent, i.e we have $\varphi(c_1, \dots, c_m) \longleftrightarrow \varphi'(c_1, \dots, c_m)$. In the case of generic quantifier elimination we compute additionally a conjunction Θ of non-trivial negated-equations, such that

$$\Theta \longrightarrow (\varphi \longleftrightarrow \varphi').$$

In other words, Θ restricts the parameter space. Note that Θ cannot become inconsistent, and moreover, the complement of the set described by Θ has a lower dimension than the complete parameter space. Thus it restricts our parameter space only slightly.

The idea behind the generic quantifier elimination is to add assumptions to Θ whenever this may either speed up the computation or may cause the algorithm to produce a shorter result formula φ'. The paradigm of generic quantifier elimination was introduced in [13] in the area of automated geometry proving. The key idea here is to express a geometric theorem as a quantified formula and then verify it by quantifier elimination. Regular quantifier elimination may fail due to lack of resources or if the theorem does not hold. In the latter case it may be false only for some degenerated situations, as for empty triangles or rectangles instead of arbitrary triangles. Generic quantifier elimination is in this area superior to the regular one for two reasons: The computations are in general much faster and the assumptions made in Θ may exclude degenerated situations in which the theorem is false. In the above cited paper, which is based on quantifier elimination by virtual substitution, it was heuristically shown that for this generic quantifier elimination in fact Θ contains mostly non-degeneracy conditions.

Meanwhile, using a generic projection operator, the concept of generic quantifier elimination was also successfully applied to quantifier elimination by partial cylindrical algebraic decomposition [14]. Seidl and Sturm study the general applicability of generic quantifier elimination in contrast to the regular one. As for regular quantifier elimination by cylindrical algebraic decomposition, this approach is successful mostly for problems containing only a few variables. This restricts the applicability of the generic projection operator to the area of automated theorem proving.

In this note we introduce a generic variant of Hermitian quantifier elimination and apply it in the area of automated theorem proving. This generic quantifier elimination is not degree restricted as the method based on virtual substitution, and it can handle in general, more variables than the method based on cylindrical algebraic decomposition. Hermitian quantifier elimination is, however, well suited for formulas containing many equations as in the case of automated geometric theorem proving. Nevertheless the generic Hermitian quantifier elimination is well suited in many other application areas, e.g., in physical applications in which a equation between two different values is always of no meaning.

The plan of the paper is as follows: In the next Section 2 we sketch the Hermitian quantifier elimination algorithm. In Section 3 we discuss three parts of the algorithm where the concept of generic quantifier elimination can be successfully applied. After describing the generic algorithm we show in Section 4 the scope of our method by computation examples. In the final Section 5 we conclude and summarize our results.

2 The Basic Algorithm

We want to eliminate the quantifiers from an arbitrary first-order formula in the language of ordered rings. In our discussion we restrict our attention to the main parts of the Hermitian quantifier elimination with some improvements. Given an arbitrary first-order formula, we first compute an equivalent prenex normal form of the form

$$\mathsf{Q}_n x_{n1} \cdots \mathsf{Q}_n x_{nm_n} \cdots \mathsf{Q}_2 x_{21} \cdots \mathsf{Q}_2 x_{2m_2} \mathsf{Q}_1 x_{11} \cdots \mathsf{Q}_1 x_{1m_1}(\psi), \quad \mathsf{Q}_i \in \{\exists, \forall\},$$

with $\mathsf{Q}_{i-1} \neq \mathsf{Q}_i$ for $i \in \{2, \dots, n\}$ and ψ quantifier-free.

Our elimination algorithm eliminates the quantifier blocks, block by block, beginning with the innermost one, i.e., we compute first a quantifier-free equivalent of

$$\mathsf{Q}_1 x_{11} \cdots \mathsf{Q}_1 x_{1m_1}(\psi).$$

Using the equivalence

$$\forall x_1 \dots \forall x_n(\psi) \longleftrightarrow \neg \exists x_1 \dots \exists x_n(\neg\psi),$$

we can obviously restrict our discussion to the case of one existential quantifier block, i.e. $\mathsf{Q}_1 = \exists$. We can furthermore assume without lost of generality (for short w.l.o.g.) that ψ contains only atomic formulas of the form $t = 0$, $t > 0$ and $t \neq 0$ and that ψ is in disjunctive normal form. By applying the equivalence

$$\exists x_1 \dots \exists x_n \left(\bigvee_{i=1}^{k} \psi_i \right) \longleftrightarrow \bigvee_{i=1}^{k} \exists x_1 \dots \exists x_n(\psi_i)$$

we assume in the following that ψ is a conjunction of atomic formulas of the above form.

2.1 Preparation

We assume that our input formula has the following form

$$\exists x_1 \ldots \exists x_n \left(\bigwedge_{\hat{g} \in \hat{G}} (g = 0) \wedge \bigwedge_{h \in H} (h > 0) \wedge \bigwedge_{f \in F} (f \neq 0) \right),$$

where $\hat{G}$, H, and F are finite sets of polynomials in $\mathbb{Q}[u_1, \ldots, u_m][x_1, \ldots, x_n]$. We can obviously evaluate each variable-free atomic formula to a truth value making itself superfluous or the whole conjunction contradictive. Thus we can w.l.o.g. assume that each polynomial is an element of $\mathbb{Q}[u_1, \ldots, u_m][x_1, \ldots, x_n] \setminus \mathbb{Q}$.

For a polynomial $g \in \mathbb{Q}[u_1, \ldots, u_m][x_1, \ldots, x_n]$ and $(c_1, \ldots, c_m) \in \mathbb{R}^m$ we denote by $g(c_1, \ldots, c_m)$ the polynomial in $\mathbb{R}[x_1, \ldots, x_n]$ constructed from g by plugging in the c_i for u_i with $i \in \{1, \ldots, m\}$. We extend this notation in the natural manner to sets of polynomials.

If the set $\hat{G}$ is empty, we proceed with our quantifier elimination as described in Section 2.3. If $\hat{G}$ is not empty, we compute a Gröbner system [15] w.r.t. an arbitrary but fixed term order. This term order is also fixed for all subsequent computations in the following paragraphs.

The concept of Gröbner systems generalizes the concept of Gröbner bases to the parametric case. With the term "parametric case" we describe situations in which the coefficient of the polynomials are given parametric as polynomials in some variables, e.g. $mx + b$ is a univariate polynomial in x with the parametric coefficients m and b.

A Gröbner system S is a finite set of pairs (γ, G), called branches of the Gröbner system. Each branch consists of a quantifier-free formula γ in the u_1, $\ldots$, u_m and a finite set of polynomials $\mathbb{Q}[u_1, \ldots, u_m][x_1, \ldots, x_n]$. For each $c \in \mathbb{R}^m$ there is one branch (γ, G) such that $\gamma(c)$ holds, we have $\mathrm{Id}\big(G(c)\big) = \mathrm{Id}\big(\hat{G}(c)\big)$, and $G(c)$ is a Gröbner basis. In fact, all computations used for our algorithm can be performed parametrically using G.

Note, that for every (γ, G) and $c \in \mathbb{R}^m$ with $\gamma(c)$ we have that $\hat{G}(c)$, $G(c)$ and $\mathrm{Id}\big(G(c)\big)$ have the same zeroes. By switching from

$$\bigwedge_{g \in \hat{G}} (g = 0) \quad \text{to} \quad \bigvee_{(\gamma, G) \in S} \gamma \wedge \bigwedge_{g \in G} (g = 0)$$

and interchanging the disjunction with the existential quantifier block it suffices to eliminate the quantifiers from

$$\gamma \wedge \exists x_1 \cdots \exists x_n \left(\bigwedge_{g \in G} g = 0 \wedge \bigwedge_{h \in H} h > 0 \bigwedge_{f \in F} f \neq 0 \right).$$

Let d be the dimension of $\mathrm{Id}\big(G(c)\big)$ with $c \in \mathbb{R}^m$ and $\gamma(c)$. Note that this dimension is uniquely determined by γ. According to the dimension d we proceed as follows: If the ideal is zero dimensional, i.e., $d = 0$, we eliminate the complete block of existential quantifiers as described in the next Section 2.2. If the dimension is -1, i.e., the ideal is actually the entire polynomial ring, and thus there is obviously no zero of G, because 1 is member of the ideal. Our elimination result in this case is simply *false*. If the dimension is n, which is the number of main variables, we have to reformulate the problem and call our quantifier elimination recursively as described in Section 2.3. If, finally, the dimension is between 1 and $n-1$ then we eliminate the quantifier block with two recursive calls of our Hermitian quantifier elimination as described in Section 2.4.

2.2 The zero-dimensional case

We want to eliminate the quantifiers from

$$\gamma \wedge \exists x_1 \cdots \exists x_n \big(\bigwedge_{g \in G} g = 0 \wedge \bigwedge_{h \in H} h > 0 \bigwedge_{f \in F} f \neq 0 \big),$$

where for each c in $\mathbb{R}^m$ with $\gamma(c)$ we have that $G(c)$ is Gröbner basis of the zero-dimensional ideal $\mathrm{Id}\big(G(c)\big)$. For this we use a method originally developed for counting the real zeroes of $\mathrm{Id}\big(G(c)\big)$ w.r.t. the side conditions generated by H and F.

The result we use was found independently by Pedersen, Roy, Szpirglas [16] and Becker, Wörmann [17] generalizing a result of Hermite for the bivariate case. It was adapted to the parametric case including several side conditions by Weispfenning [7] and further extended by the first author [8].

For a moment, assume that $H = \{h_1, \ldots, h_r\}$ and $F = \{f_1, \ldots, f_s\}$. Let $E = \{1, 2\}^r$. For $e \in E$ define h^e by

$$h^e = \prod_{i=1}^{r} h_i^{e_i} \cdot \prod_{i=1}^{s} f_i^2.$$

For a univariate polynomial q define $Z_+(q)$ as the number of positive zeroes and $Z_-(q)$ as the number of negative zeroes, respectively, both counted with multiplicities.

Consider $R = \mathbb{Q}(u_1, \ldots, u_m)[x_1, \ldots, x_n]$ and let be $I = \mathrm{Id}(G)$ and $B = \{v_1, \ldots, v_b\}$ the reduced terms of G. Then $R(\underline{c})/I(\underline{c})$ is a $\mathbb{Q}$-algebra with basis $B(\underline{c})$ for each $\underline{c}$ with $\gamma(c)$. Note that each element in R can also be viewed as an element of R/I. For $q \in R$, the map

$$m_q : R/I \to R/I, \quad \text{defined by} \quad m_q(p) = q \cdot p$$

is linear. Using this definition we define for a polynomial $p \in R$ the $b \times b$ matrix $Q_p = (q_{ij})$ by

$$q_{ij} = \mathrm{trace}(m_{v_i v_j p}).$$

Finally let $\chi(Q_p)$ be the characteristic polynomial of Q_p.

Then we have for each $c \in \mathbb{R}^m$ with $\gamma(c)$, that

$$\left| \left\{ a \in \mathbb{R}^n \;\middle|\; \bigwedge_{g \in G} g(c)(a) = 0 \wedge \bigwedge_{h \in H} h(c)(a) > 0 \wedge \bigwedge_{f \in F} f(c)(a) \neq 0 \right\} \right|$$

equals $Z_+(\chi) - Z_-(\chi)$, where $\chi = \prod_{e\in E} \chi(Q_{h^e})$.

While the computations used so far are all uniform in $c \in \mathbb{R}^m$ with $\gamma(c)$, we cannot uniformly count Z_+ or Z_- for χ. Note that $\chi(c)$ is of real type, i.e., there are no zeroes in $\mathbb{C} \setminus \mathbb{R}$. For those polynomials we can compute the number of positive and negative zeroes using Descartes rule of signs. It states that the positive real zeroes of a polynomial $\chi(c)$ of real type are exactly the number of sign changes in the list of coefficients of $\chi(c)$ ignoring 0. By considering $\chi(-c)$ one can also compute the negative zeroes using Descartes rule of signs.

Let $\chi = \sum_{i=0}^{l} a_i/b_i y^i$, where $a_i, b_i \in \mathbb{Q}[u_1, \dots, u_m]$. For $\delta \in \{<, =, >\}^l$ let φ_δ be the formula

$$a_1 b_1 \ \delta_1 \ 0 \wedge \cdots \wedge a_l b_l \ \delta_l \ 0.$$

Using Descartes rule of signs we can now uniformly count the number Z_+^δ of positive and the number Z_-^δ of negative zeroes of $\chi(c)$ for all $c \in \mathbb{R}^m$ with $\gamma(c)$ and $\varphi_\delta(c)$.

Finally define φ by

$$\bigvee_{\delta\in\{<,=,>\}^l} \Big\{ \varphi_\delta \ \Big| \ Z_+^\delta - Z_-^\delta = 0 \Big\}.$$

Our formula φ states that the polynomial χ has exactly the same number of positive as of negative real zeroes. A quantifier-free formula with this property is called *type formula* for the polynomial χ. Recall from our discussion above that in this situation G has no zeroes which satisfy the given side conditions. Thus our final elimination result is $\gamma \wedge \neg\varphi$.

2.3 Constructing equations

We enter this case of the Hermitian quantifier elimination if the input formula does not contain any equation or the dimension of $\mathrm{Id}\big(G(c)\big)$ is n, i.e., $I(G) = \{0\}$ for $\underline{c}$ with $\gamma(\underline{c})$. In other words we consider the input formula

$$\exists x_1 \cdots \exists x_n \Big(\bigwedge_{h\in H} h > 0 \wedge \bigwedge_{f\in F} f \neq 0 \Big).$$

In this case, we can eliminate one quantifier, say x_n, and the other quantifiers of the considered block are eliminated by a recursive call of the Hermitian quantifier elimination.

Let h have a representation of the form $\sum_{k=0}^{d_h} a_{h,k} x_n^k$ where each $a_{h,k}$ is a polynomial in $\mathbb{Q}[u_1, \dots, u_m, x_1, \dots, x_{n-1}]$ with $a_{h,d_h} \neq 0$. Assume for a moment that $H = \{h_1, \dots, h_r\}$ and let $D = \times_{k=1}^{r} \{0, \dots, d_{h_k}\}$. For $\delta \in D$ we denote by δ_h the s-th element of δ such that $h_s = h$. Define $P = \{ (h_i, h_j) \mid 1 \leq i < j \leq r \} \subseteq H^2$.

For $h \in H$ and $d \in \{0, \dots, d_h\}$ let Γ_d^h be the following formula:

$$\bigwedge_{k=d+1}^{d_h} a_{h,k} = 0 \wedge a_{h,d} \neq 0.$$

For fixed $h \in H$ the formulas Γ_d^h build a complete disjunctive case distinction of the degree of h under specifications of the parameters. Notice that it does not matter at this point, if Γ_d^h is equivalent to *false*. Let $\varrho_d(h) = \sum_{k=0}^d a_{h,k} x_n^k$. We have the equivalence

$$\exists x_n \Big(\bigwedge_{h \in H} h > 0 \wedge \bigwedge_{f \in F} f \neq 0 \Big) \longleftrightarrow \bigvee_{\delta \in D} \exists x_n \Big(\bigwedge_{h \in H} \Gamma_{\delta_h}^h \wedge \varrho_{\delta_h}(h) > 0 \wedge \bigwedge_{f \in F} f \neq 0 \Big).$$

For each δ we then in turn transform the formulas

$$\exists x_n \Big(\bigwedge_{h \in H} \Gamma_{\delta_h}^h \wedge \varrho_{\delta_h}(h) > 0 \wedge \bigwedge_{f \in F} f \neq 0 \Big)$$

separately to

$$\begin{aligned}
&\bigwedge_{f \in F} \bigvee_{i=0}^{d_f} a_{f,i} \neq 0 \wedge \\
&\Big(\Big(\bigwedge_{h \in H} \Gamma_{\delta_h}^h \wedge a_{h,\delta_h} > 0 \Big) \vee \\
&\Big(\bigwedge_{h \in H} \Gamma_{\delta_h}^h \wedge a_{h,\delta_h} < 0 \vee (\mathrm{Even}(\delta_h) \wedge a_{h,\delta_h} > 0) \Big) \vee \\
&\bigvee_{p \in Q} \exists x_n \Big(\bigwedge_{h \in H} \Gamma_{\delta_h}^h \wedge \varrho_{\delta_h}(h) > 0 \wedge \frac{\partial p}{\partial x_n} = 0 \Big) \vee \\
&\bigvee_{(p,q) \in P} \exists x_n \Big(\bigwedge_{h \in H} \Gamma_{\delta_h}^h \wedge \varrho_{\delta_h}(h) > 0 \wedge (p - q) = 0 \Big) \Big),
\end{aligned}$$

where $Q = \{\, h \in H \mid \delta_h \geq 2 \,\}$. The used predicate $\mathrm{Even}(n)$ is true if and only if n is even. Thus we have shown how to trace back this case to the case with at least one existing equation in the input formula.

Let φ' denote the complete transformed input formula. Then we apply the Hermitian quantifier elimination to each quantified constituent and obtain, by eliminating x_n, a quantifier-free equivalent ψ'. Finally we apply the Hermitian quantifier elimination again recursively to

$$\exists x_1 \cdots \exists x_{n-1}(\psi')$$

obtaining a quantifier-free equivalent ψ. The final result of the elimination step is then

$$\gamma \wedge \psi.$$

2.4 Partial elimination

We enter this case of the Hermitian quantifier elimination if the dimension of G is d with $d \in \{1, \ldots, n-1\}$. We compute a maximal strongly independent set Ξ [18]. Let w.l.o.g. be $\Xi = \{x_1, \ldots, x_k\}$. Then we apply recursively the Hermitian quantifier elimination to

$$\exists x_{k+1} \cdots \exists x_n \left(\bigwedge_{g \in G} g = 0 \wedge \bigwedge_{h \in H} h > 0 \wedge \bigwedge_{f \in F} f \neq 0 \right)$$

and obtain a quantifier-free formula ψ'. Then we apply our quantifier elimination procedure again recursively to

$$\exists x_1 \cdots \exists x_k(\psi')$$

yielding ψ. Our quantifier-free result is then $\gamma \wedge \psi$. This concludes the description of the Hermitian quantifier-elimination.

3 Generic Hermitian Quantifier Elimination

In this section we discuss our modifications to the algorithm for obtaining a generic quantifier elimination. As already mentioned in the introduction, a generic quantifier elimination computes for a first-order formula φ a quantifier-free formula φ' and a conjunction Θ of negated equations in the parameters u_1, …, u_m such that

$$\Theta \longrightarrow (\varphi \longleftrightarrow \varphi').$$

Θ is called a theory. Recall from our discussion in the previous section that our quantifier elimination algorithm is recursive. In each recursive call we consider variables originally bound by quantifiers as additional parameters. Obviously we are not allowed to add assumptions about these additional parameters to Θ. To guarantee this restriction we denote by v_1, …, v_m the set of parameters of the input formula. In the discussion below we will always test whether an assumption is valid by checking whether it contains only variables from $\{v_1, \ldots, v_m\}$.

3.1 Generic Gröbner Systems

Our first and most prominent modification to the pure elimination algorithm is to compute in the preparation phase a generic Gröbner system instead of a regular one.

Let $<$ be a term order and let $p = c_1 t_1 + \cdots + c_d t_d$ be a polynomial in $\mathbb{Q}[u_1, \ldots, u_m][x_1, \ldots, x_n]$, where c_1, …, $c_d \in \mathbb{Q}[u_1, \ldots, u_m]$ and $t_d > \cdots > t_1$ terms. Then the head term of p is c_d. For a given $c \in \mathbb{R}^m$ this may or may not be true for the polynomial $p(c)$. It depends on whether $c_d(c) \neq 0$ or not. During the construction of a Gröbner system we systematically construct a case distinction about some parameters of the occurring polynomials. In each case of this case distinction the head term of all polynomials is uniformly determined.

A generic Gröbner system allows us to exclude some cases by adding assumptions to Θ. In particular if c_d contains only parameters from $\{v_1, \ldots, v_m\}$ we add

$c_d \neq 0$ to Θ and assume in the following computation steps that the head term of p is t_d.

We denote by $\vdash$ a suitable heuristic to decide an implication: If $\gamma \vdash \alpha$, then we have the validity of $\gamma \longrightarrow \alpha$. Note that the construction of a Gröbner system requires that this heuristic can actually decide some implications.

The first algorithm extends a partial Gröbner system by an additional polynomial. Note that we assume that the theory Θ to be computed is globally available.

We use the following notations: $\mathrm{HC}(f)$ is the head or leading coefficient of f w.r.t. our fixed term order, $\mathrm{Red}(f)$ is the polynomial up to the head monomial, $\mathrm{Var}(f)$ is the set of variables actually occurring in f.

Algorithm 1 (extend) *Input: A partial system S, a branch (γ, G), and two polynomials h and h'. Output: An extended partial system.*

1 **if** $h' = 0$ **then**
2   return $S \cup \{(\gamma, G)\}$
3 **else if** $\mathrm{Var}\big(\mathrm{HC}(h')\big) \subseteq \{v_1, \dots, v_m\}$ **then**
4     $\Theta := \Theta \wedge (\mathrm{HC}(h') \neq 0)$
5   return $S \cup \big\{\big(\gamma,\, G \cup \{h\}\big)\big\}$
6 **else if** $\gamma \wedge \Theta \vdash \mathrm{HC}(h') \neq 0$ **then**
7   return $S \cup \big\{\big(\gamma,\, G \cup \{h\}\big)\big\}$
8 **else if** $\gamma \wedge \Theta \vdash \mathrm{HC}(h') = 0$ **then**
9   return extend$(S,\, (\gamma, G),\, h,\, \mathrm{Red}(h'))$
10 **else**
11     $S' := \big\{\big(\gamma \wedge \mathrm{HC}(h') \neq 0,\, G \cup \{h\}\big)\big\}$
12   return extend$(S',\, (\gamma \wedge \mathrm{HC}(h) = 0, G),\, h,\, \mathrm{Red}(h'))$
13 **fi**

This algorithm differs from the algorithm for regular Gröbner systems by accessing the theory Θ and by the lines 3 and 4 for generating new assumptions.

For computing a Gröbner system we start with computing an initial partial system S by calling the following algorithm Initialize with input $\hat{G}$.

Algorithm 2 (Initialize) *Input: A finite set H of polynomials. Output: A partial system.*

begin
    $S := \{(\mathrm{true}, \emptyset)\}$
    for each $h \in H$ **do**
        for each $(\gamma, G) \in S$ **do**
            $S := S \setminus \{(\gamma, G)\}$
            S := extend$(S,\, (\gamma, G),\, h,\, h)$
        od
    od
end

For computing the Gröbner system from the partial system we proceed as follows: We select a branch (γ, G) of S, compute $S' = S \setminus \{(\gamma, G)\}$. Then we select g_1, g_2 from G such that the normal form h of the S-polynomial of g_1, g_2 is not 0. Finally we extend S' by (γ, G), h and h. This process is repeated until the normal form of all S-polynomials is 0.

As mentioned above the generic variant of the Gröbner system computation allows us to drop branches. Recall from the presentation of our quantifier elimination algorithm that we have to perform for each branch a separate quantifier elimination. If we are on the top-level of our quantifier-elimination algorithm we actually compute a Gröbner system containing one single branch, because the condition on line 3 of the algorithm "extend" is tautological in this situation. This reduces, in general, both the computation time and the size of the output formula dramatically. As a rule, observed from our sample computations, we compute only a few assumptions which can often be easily interpreted.

3.2 Generic Equation Construction

In Section 2.3 we have discussed how to construct an equation from a set of ordering relations. In this section we adapt this to the generic case.

Recall that we generate a complete case distinction about the highest coefficient of each $h \in H$. The size of this case distinction can be reduced by making appropriate assumptions as shown below.

For $h \in H$ let

$$n_h = \max\bigl(\{-1\} \cup \{\, i \in \{0, \ldots, d_h\} \mid \mathrm{Var}(a_{h,i}) \subseteq \{v_1, \ldots, v_m\} \,\}\bigr).$$

For all n_h with $h \in H$ and $n_h \geq 0$ we add the assumption $a_{h,n_h} \neq 0$ to our theory Θ. Let finally $D' = \times_{k=1}^{r} \{\max(0, n_h), \ldots, d_h\}$. Then we can proceed with the transformation described in Section 2.3 using D' instead of D. Note that $D' \subseteq D$ and often $D' \subsetneq D$.

3.3 Generic Type Formula Computation

In this section we discuss an approach to computing generic type formulas.

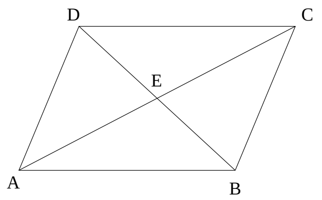
The type formula construction presented in Section 2.2 is a primitive version of the method used in our highly optimized Hermitian quantifier elimination. We actually compute a type formula τ_d for a polynomial $p = \sum_{i=0}^{d} c_i y^i$ of degree d recursively:

$$\tau_d(c_d, \ldots, c_0) \equiv (c_0 = 0) \wedge \tau_{d-1} \vee \tau'_d(c_d, \ldots, c_0).$$

The recursion basis are the simple type formulas up to the degree 3. The definition of τ'_d is similar to the definition of τ_d, but assumes a non-vanishing constant coefficient which implies the absence of the zero 0. The formula τ'_d is actually a disjunctive normal form. Each constituent has the following schema

$$c_{k_1} \; \varrho_{k_1} \; 0 \wedge \cdots \wedge c_{k_l} \; \varrho_{k_l} \; 0,$$

where $\{k_1, \ldots, k_l\} \subseteq \{1, \ldots, d\}$ and $\varrho_{k_j} \in \{<, >\}$.



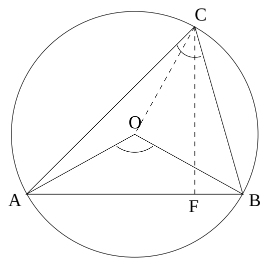


Fig. 1. Example 1 (left) and Example 2 (right)

For our generic type formula computation we cannot make use of our assumption for computing τ_d'. If $\mathrm{Var}(c_0) \subseteq \{v_1, \ldots, v_m\}$ we can however avoid the recursion by adding $c_0 \neq 0$ to Θ. This reduces the size of the output formula dramatically and if it occurs in the recursions it reduces the computation time, too. Our test computations have, however, shown that, in general, the assumptions made here are very complex and cannot be easily interpreted. For our application of automated theorem proving we have thus not analyzed this method further. This does not mean that generic type formulas are an irrelevant optimization for other application areas of generic quantifier elimination.

4 Examples

In this section we will apply our elimination algorithm to some automatic proofs of geometric theorems. We have implemented the algorithm in REDLOG 3.0, which is part of the current version 3.8 of the computer algebra system REDUCE.

For the geometric theorem proving we proceed here as described in [13]. Our examples will show the meaning of the assumptions which are created during the computations. In most cases, these assumptions can be interpreted as (necessary) non-degeneracy conditions, so they have a powerful geometric interpretation. Note that the constructed assumptions may not be a complete list of non-degeneracy conditions for the particular example. We will also show that generic Hermitian quantifier elimination will speed up the elimination procedure and will create smaller solution formulas than the regular Hermitian quantifier elimination. We explain in detail how to express a geometric theorem as a first-order formula by means of our first example.

Example 1 Given a parallelogram $ABCD$, let E be the intersection point of its diagonals. Then E is the midpoint of the diagonals (see Figure 1). This example was taken from [19]. By a suitable motion in $\mathbb{R}^2$ we can assume w.l.o.g.

$$A = (0,0), \quad B = (u_1, 0), \quad C = (u_2, u_3), \quad D = (x_2, x_1), \quad E = (x_4, x_3).$$

We now can describe the necessary properties to our statement by the following equations:

$$\begin{array}{ll} h_1 \equiv u_1x_1 - u_1u_3 = 0 & AB\|DC \\ h_2 \equiv u_3x_2 - (u_2 - u_1)x_1 = 0 & DA\|CB \\ h_3 \equiv x_1x_4 - (x_2 - u_1)x_3 - u_1x_1 = 0 & E \in BD \\ h_4 \equiv u_3x_4 - u_2x_3 = 0 & E \in AC \\ g \equiv 2u_2x_4 + 2u_3x_3 - u_3^2 - u_2^2 = 0 & \mathrm{Length}(AE) = \mathrm{Length}(CE) \end{array}$$

The theorem can then be formulated as $\forall x_1 \forall x_2 \forall x_3 \forall x_4 (h_1 \wedge h_2 \wedge h_3 \wedge h_4 \rightarrow g)$. The application of our elimination algorithm leads in 30 ms to the result

$$\Theta \equiv u_1 \neq 0 \wedge u_3 \neq 0, \quad \varphi' \equiv \mathrm{true}.$$

The theory Θ states that $ABCD$ is a proper parallelogram, i.e., opposite edges do not collapse. Non-generic Hermitian quantifier elimination yields in 170 ms a quantifier-free formula consisting of 24 atomic formulas.

Example 2 Let O be the center of the circumcircle of a triangle ABC. If O does not lie outside of ABC, then $\angle ACB = \angle AOB/2$. See Figure 1. This example was taken from [20].

W.l.o.g. we can assume the following coordinates

$$A = (-u_1, 0), \quad B = (u_1, 0), \quad C = (u_2, u_3), \quad O = (0, x_1), \quad F = (0, u_3).$$

We express the theorem as follows:

$$\begin{aligned} &\forall r \forall x_1 \forall t_1 \forall t_2 \forall t \forall t' \big(r^2 = u_1^2 + x_1^2 \wedge r^2 = u_2^2 + (u_3 - x_1)^2 \wedge \\ &\quad u_3t_1 = u_1 + u_2 \wedge u_3t_2 = u_1 - u_2 \wedge (1 - t_1t_2)t = t_1 + t_2 \wedge \\ &\qquad x_1t' = u_1 \rightarrow t = t'\big). \end{aligned}$$

Generic Hermitian quantifier elimination on this formula leads in 10 ms to the result

$$\Theta \equiv u_1^2 - u_2^2 - u_3^2 \neq 0 \wedge u_3 \neq 0, \quad \varphi \equiv \mathit{true}.$$

We now take a closer look at Θ. $u_3 \neq 0$ ensures that not all points of the triangle lie on the x-axis. This is a necessary non-degeneracy condition. The assumption $u_1^2 - u_2^2 - u_3^2 \neq 0$ prevents that the midpoint of the circumcircle lies on the edge AB. We have proved this theorem if the constructed non-degeneracy assumptions hold. Actually the theorem holds for $u_3 \neq 0$, i.e., the second assumption is superfluous.

Example 3 (Feuerbach's Theorem) *The nine point circle of a triangle is tangent to the incircle and to each of the excircles of the triangle.* See [19] for a formulation of this problem. We get the following elimination result:

$$\begin{aligned} \Theta \equiv\ & u_1u_3 + u_2^2 \neq 0 \wedge u_1 + 2u_2 + u_3 \neq 0 \wedge u_1 + u_2 \neq 0 \wedge u_1 - u_2 \neq 0 \\ & \wedge u_1 - 2u_2 + u_3 \neq 0 \wedge u_1 - u_3 \neq 0 \wedge u_1 \neq 0 \wedge u_2 + u_3 \neq 0 \wedge u_2 - u_3 \neq 0 \wedge \\ & u_2 \neq 0 \wedge u_3 \neq 0, \\ \varphi \equiv\ & u_1 - u_3 \neq 0. \end{aligned}$$

φ is obviously equivalent to *true* under the assumption of Θ. While we receive this result in 350 ms, regular Hermitian quantifier elimination cannot eliminate the quantifiers using 128MB.

Example 4 (M. Paterson's problem) *Erect three similar isosceles triangles A_1BC, AB_1C, and ABC_1 on the sides of a triangle ABC. Then AA_1, BB_1 and CC_1 are concurrent. How does the point of concurrency moves as the areas of the three similar triangles are varied between 0 and ∞.* This example is actually an example for theorem finding and not only theorem proving. See Chou [19] for a description of this problem. We get the following elimination result:

$$\begin{aligned}
&\Theta \equiv u_1u_2 - u_2x - u_3y \neq 0 \wedge u_2 - x \neq 0 \wedge u_2 \neq 0 \wedge u_3 - y \neq 0 \wedge y \neq 0,\\
&\varphi' \equiv u_1^2u_2y + u_1^2u_3x - 2u_1^2xy + u_1u_2^2y - 2u_1u_2u_3x + 2u_1u_2xy - u_1u_3^2y\\
&\qquad -u_1u_3x^2 + u_1u_3y^2 - 2u_2^2xy + 2u_2u_3x^2 - 2u_2u_3y^2 + 2u_3^2xy = 0 \vee u_1 = 0.
\end{aligned}$$

The result is obtained in 60 ms and describes a geometric locus. If one uses non-generic Hermitian quantifier elimination for eliminating, the result is obtained in 2,8 seconds and consists of 295 atomic formulas.

5 Conclusions

We have presented a generic quantifier elimination method based on Hermitian quantifier elimination. For this purpose we have analyzed where making assumptions on parameters may support the algorithm: We compute generic Gröbner systems instead of regular ones reducing the practical complexity of our algorithm in all cases. In the special case that no equations occur in the input, we have additionally reduced the number of recursions needed.

By example computations we have shown that our generic Hermitian quantifier elimination can be successfully used for automatic theorem proving and theorem finding. In all examples the results are considerably shorter and the computation times are much faster than for regular Hermitian quantifier elimination.

References

1. Collins, G.E.: Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In Brakhage, H., ed.: Automata Theory and Formal Languages. 2nd GI Conference. Volume 33 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, New York (1975) 134–183
2. Collins, G.E., Hong, H.: Partial cylindrical algebraic decomposition for quantifier elimination. Journal of Symbolic Computation **12** (1991) 299–328
3. Brown, C.W.: Simplification of truth-invariant cylindrical algebraic decompositions. In Gloor, O., ed.: Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation (ISSAC 98), Rostock, Germany, ACM, ACM Press, New York (1998) 295–301

4. Weispfenning, V.: The complexity of linear problems in fields. Journal of Symbolic Computation **5** (1988) 3–27
5. Loos, R., Weispfenning, V.: Applying linear quantifier elimination. THE Computer Journal **36** (1993) 450–462 Special issue on computational quantifier elimination.
6. Dolzmann, A., Sturm, T.: Redlog: Computer algebra meets computer logic. ACM SIGSAM Bulletin **31** (1997) 2–9
7. Weispfenning, V.: A new approach to quantifier elimination for real algebra. In Caviness, B., Johnson, J., eds.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and Monographs in Symbolic Computation. Springer, Wien, New York (1998) 376–392
8. Dolzmann, A.: Reelle Quantorenelimination durch parametrisches Zählen von Nullstellen. Diploma thesis, Universität Passau, D-94030 Passau, Germany (1994)
9. Gilch, L.A.: Effiziente Hermitesche Quantorenelimination. Diploma thesis, Universität Passau, D-94030 Passau, Germany (2003)
10. Dolzmann, A.: Solving geometric problems with real quantifier elimination. In Gao, X.S., Wang, D., Yang, L., eds.: Automated Deduction in Geometry. Volume 1669 of Lecture Notes in Artificial Intelligence (Subseries of LNCS). Springer-Verlag, Berlin Heidelberg (1999) 14–29
11. Dolzmann, A., Sturm, T.: Simplification of quantifier-free formulae over ordered fields. Journal of Symbolic Computation **24** (1997) 209–231
12. Weispfenning, V.: Applying quantifier elimination to problems in simulation and optimization. Technical Report MIP-9607, FMI, Universität Passau, D-94030 Passau, Germany (1996) To appear in the Journal of Symbolic Computation.
13. Dolzmann, A., Sturm, T., Weispfenning, V.: A new approach for automatic theorem proving in real geometry. Journal of Automated Reasoning **21** (1998) 357–380
14. Seidl, A., Sturm, T.: A generic projection operator for partial cylindrical algebraic decomposition. In Sendra, R., ed.: Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation (ISSAC 03), Philadelphia, Pennsylvania. ACM Press, New York, NY (2003) 240–247
15. Weispfenning, V.: Comprehensive Gröbner bases. Journal of Symbolic Computation **14** (1992) 1–29
16. Pedersen, P., Roy, M.F., Szpirglas, A.: Counting real zeroes in the multivariate case. In Eysette, F., Galigo, A., eds.: Computational Algebraic Geometry. Volume 109 of Progress in Mathematics. Birkhäuser, Boston, Basel; Berlin (1993) 203–224 Proceedings of the MEGA 92.
17. Becker, E., Wörmann, T.: On the trace formula for quadratic forms. In Jacob, W.B., Lam, T.Y., Robson, R.O., eds.: Recent Advances in Real Algebraic Geometry and Quadratic Forms. Volume 155 of Contemporary Mathematics., American Mathematical Society, American Mathematical Society, Providence, Rhode Island (1994) 271–291 Proceedings of the RAGSQUAD Year, Berkeley, 1990–1991.
18. Kredel, H., Weispfenning, V.: Computing dimension and independent sets for polynomial ideals. Journal of Symbolic Computation **6** (1988) 231–247 Computational aspects of commutative algebra.
19. Chou, S.C.: Mechanical Geometry Theorem Proving. Mathematics and its applications. D. Reidel Publishing Company, Dordrecht, Boston, Lancaster, Tokyo (1988)
20. Sturm, T.: Real Quantifier Elimination in Geometry. Doctoral dissertation, Department of Mathematics and Computer Science. University of Passau, Germany, D-94030 Passau, Germany (1999)