Local Quantifier Elimination

Andreas Dolzmann and Volker Weispfenning
Department of Mathematics and Computer Science
University of Passau, Germany
{dolzmann,weispfen}@uni-passau.de
http://www.fmi.uni-passau.de/~{dolzmann,weispfen}/

MIP-0003
February 29, 2000

Abstract

We introduce local quantifier elimination as a new variant of real quantifier elimination. Given a first-order formula and a real point we compute a quantifier-free formula which is not only for the given point equivalent to the input formula but also for all points in a semi-algebraic set containing the specified point. The description of this semi-algebraic set is explicitly computed in the form of a conjunction of atomic formulas. Local quantifier elimination is in its application area, superior to both regular and generic quantifier elimination due to faster running times and shorter results.

1 Introduction

In the last 25 years real quantifier elimination has turned out to be an important mathematical tool. In 1975 Collins introduced the quantifier elimination by cylindrical algebraic decomposition (CAD), cf. [3] which was implemented by Arnon in 1981. This was the first quantifier elimination algorithm ever implemented.

Since then many new quantifier elimination algorithms and their optimizations were developed, and many problems from science, engineering, and also in economics, namely in operations research, were formulated as quantifier elimination problems. The research on the algorithms and on the applications have mutually inspired each other. The success of the research was made possible due to the existence of efficiently implemented quantifier elimination procedures. Though there are many approaches for quantifier elimination only two implemented quantifier elimination algorithms are widely in use. Both implementations are based on mathematical methods which are carefully studied and refined for an efficient implementation.

Intensive research on the quantifier elimination by CAD, cf. [16, 10, 11], results in quantifier elimination by partial cylindrical algebraic decomposition, cf. [4], which was implemented in Hong's QEPCAD program. The most recent improvements were done by Brown, cf. [1].

In 1988 Weispfenning introduced a quantifier elimination procedure for linear formulas [17]. It was first implemented by Burhenne, cf. [2]. The method was refined by the authors together with Loos and Sturm, cf. [15, 6, 19] to the quantifier elimination by virtual substitution. It was carefully analyzed, adapted for the implementation, and eventually implemented by the first author and Sturm. Their implementation is as a part of REDLOG, cf. [5], available in the commercial supported and distributed computer algebra system REDUCE 3.7. The current implemented version is still restricted to formulas in which the quantified variables occur with "low" degrees, though the method can in principle extended to an unrestricted quantifier elimination procedure, cf. [18, 19].

For a more detailed overview of the applications areas, the history of quantifier elimination, and the most important implemented algorithms, see [9].

In this note we study local quantifier elimination: Given a first-order formula $\varphi(u_1, \dots, u_m, v_1, \dots, v_n)$ in the language of ordered rings and a point $(a_1, \dots, a_n) \in \mathbb{R}^n$ we compute a quantifier-free formula $\varphi^*(\underline{u}, \underline{v})$ and a conjunction $\Theta(\underline{v})$ of atomic formulas such that

$$\Theta \longrightarrow (\varphi \longleftrightarrow \varphi^*) \quad \text{and} \quad \Theta(a_1, \dots, a_n).$$

In other words we compute from φ a quantifier-free formula φ^* and a semi-algebraic set $S \subseteq \mathbb{R}^n$ containing $\underline{a}$, such that for all $\underline{r} \in \mathbb{R}^m$, and $\underline{s} \in S$ we have that $\varphi(r, s)$ and $\varphi^*(r, s)$ are equivalent.

We call the variables $v_1, \dots, v_n$ *local parameters*, and $\underline{a}$ a *suggested point* for the local variables. Terms and formulas containing only variables from the set $\{v_1, \dots, v_m\}$ are also called *local.* The semi-algebraic set

$$S = \{\, \underline{x} \in \mathbb{R}^n \mid \Theta(\underline{x}) \,\}$$

is called *range* of local quantifier elimination applied to φ.

Local quantifier elimination is motivated by the well-known fact that the computation of the output formula φ^* in regular quantifier elimination may be very big and require long computation time. The goal is to improve this situation significantly by restricting the required equivalence to an interesting range of local parameters suggested by a point in the local parameter space.

The scope of local quantifier elimination is between regular quantifier elimination applied to $\varphi(\underline{u}, \underline{v})$ and applied to $\varphi(\underline{u}, \underline{a})$. If $m = 0$ the latter case is actually a decision problem, whereas the former one is a quantifier elimination problem. Both of these special cases can be viewed as local quantifier elimination by setting either $\Theta \equiv \mathrm{true}$ or setting $\Theta \equiv \bigwedge_{i=1}^{n} v_i = a_i$, respectively.

We are, however, not interested in these degenerated cases. Though the range will be restricted by our method, the range will as a rule not be restricted to the pure trivial case. Instead it will in almost all cases be an infinite semi-algebraic set containing the suggested point, and most frequently a neighborhood of this point. In comparison to regular quantifier elimination the output formula will be significantly smaller and will be computed faster. The local quantifier elimination procedure, presented here, is based on the quantifier elimination by virtual substitution. Constraints for Θ are generated, whenever they support the algorithm. We have implemented our method using REDLOG, cf. [5].

The concept of local quantifier elimination is closely related to the generic quantifier elimination: Generic quantifier elimination computes to an input formula $\varphi(u_1, \dots, u_m)$ a formula $\varphi^*(u_1, \dots, u_m)$ and a theory $\Theta(u_1, \dots, u_m)$ such that

$$\Theta \longrightarrow (\varphi \longleftrightarrow \varphi^*),$$

where Θ contains, in contrast to the local quantifier elimination, only negated equations. As a consequence the theory Θ holds for almost all parameter values. So the range of generic quantifier elimination applied to φ is usually larger than the range of local quantifier elimination. On the other hand the corresponding output formula will be much bigger in generic quantifier elimination. The resulting trade-off between the size of the range and the size of the output formula varies with the type of input formula.

The authors together with Sturm has introduced this concept and applied it successfully in the area of automated theorem proving in real geometry, cf. [8]. The first author and Sturm have implemented generic quantifier elimination based on the virtual substitution method in REDLOG. Ideas similar to generic quantifier elimination, as defined here, were presented by Hong on the IMACS ACA 1995, cf. [12]. His "generic" quantifier elimination is based on the quantifier elimination by partial CAD.

The plan of this note is as follows: Section 2 sketches the quantifier elimination by virtual substitution. In Section 3 we describe how to adapt quantifier elimination by virtual substitution to a local quantifier elimination algorithm. In Section 4 we give an explicit series of examples, for which local quantifier elimination has a better complexity than regular quantifier elimination. Section 5 introduces our test implementation and REDLOG. In Section 6 we give some computation examples. Finally, in Section 7 we summarize the results of our work.

2 Quantifier Elimination by Virtual Substitution

We consider *polynomial equations* $f = 0$, *weak polynomial inequalities* $f \geq 0$, $f \leq 0$, and *strict polynomial inequalities* $f > 0$, $f < 0$, $f \neq 0$, where f is a multivariate polynomial with rational coefficients. In order to distinguish inequalities of the form $f \neq 0$ from order inequalities, they are also called *disequations.* A *quantifier-free formula* ψ is a boolean combination of such equations and inequalities obtained by applying the logical operators "$\wedge$," which stands for "and," and "$\vee$," which stands for "or." We call ψ of degree d in a variable x if all polynomials occurring in ψ have an x-degree of at most d. The x_i-degree of $f \in \mathbb{R}[x_1, \ldots, x_n]$ is the degree of the univariate polynomial $f \in \mathbb{R}[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n][x_i]$.

Suppose now that ψ is quadratic, i.e. of degree 2, in some variable x, and denote $\exists x(\psi(x, u_1, \ldots, u_n))$ by $\varphi(u_1, \ldots, u_n)$. Each u_i is either quantified further outside or a parameter. In the former case it will be eliminated by iterating the procedure described here. The algorithm given in [19] computes from φ a quantifier-free formula $\varphi^*(u_1, \ldots, u_n)$ not containing x such that over the field of the reals we have the equivalence

$$\varphi(u_1, \ldots, u_n) \longleftrightarrow \varphi^*(u_1, \ldots, u_n).$$

In other words, for arbitrary values $a_1, \dots, a_n \in \mathbb{R}$ of the u_i, the assertion $\varphi^*(a_1, \dots, a_n)$ holds in $\mathbb{R}$ iff there exists $b \in \mathbb{R}$ such that $\psi(b, a_1, \dots, a_n)$ holds in $\mathbb{R}$. This is referred to as *quantifier elimination*.

The elimination of a universal quantifier can be reduced to that of an existential quantifier using the equivalence

$$\forall x \psi \longleftrightarrow \neg \exists x \neg \psi,$$

where "$\neg$" denotes logical negation. In our case, this works because the inner negation can be moved inside ψ using de Morgan's laws, and can finally be encoded by modifying the contained equations and inequalities. For sketching the elimination method of [19], we may thus restrict to the elimination of an existential quantifier.

The idea for the construction of φ^* from φ is as follows: We fix real values a_i for the variables u_i. Then all polynomials occurring in ψ become linear or quadratic univariate polynomials in x with real coefficients. So the set

$$M_\varphi = \{\, b \in \mathbb{R} \mid \psi(b, a_1, \dots, a_n) \,\}$$

of all real values b of x satisfying ψ is a finite union of closed, open, and half-open intervals on the real line. The endpoints of these intervals are among $\pm\infty$ together with the real zeros of the linear and quadratic polynomials occurring in ψ. Candidate terms $\alpha_1, \dots, \alpha_m$ for the zeros can be computed uniformly in $u_1, \dots, u_n$ by the solution formulas for linear and quadratic equations.

The candidate terms are represented by improper terms, i.e., terms that may contain symbols not belonging to the language of ordered rings. In addition, we compute to each candidate term a guard, i.e. a quantifier-free formula, stating that the term is valid and actually represents a solution of the respective equation. For instance, consider the atomic formula $c_1 x + c_0 \leq 0$. Then we generate the candidate term $\frac{-c_0}{c_1}$ and the guard $c_1 \neq 0$. Note that the language of ordered rings does not contain the division. For each form of a candidate term, however, we have specified a virtual substitution, denoted by "$/\!/$," i.e., we can compute to a formula φ a variable x, a term t, and a guard γ a formula ψ, such that $\gamma \wedge \varphi[x /\!/ t]$ and $\gamma \wedge \psi$ are equivalent in an obvious sense, cf. [19]. For example:

$$(b_1 x + b_0 > 0)\Big[x /\!/ \frac{-c_0}{c_1}\Big] \equiv -b_1 c_0 c_1 - b_0 c_1^2 > 0.$$

If all inequalities in ψ are weak, then all the intervals constituting M_φ will, into each direction, be either unbounded or closed. In the latter case, such an interval will contain its real endpoint. Thus M_φ is non-empty iff the substitution of $\pm\infty$ or of one of the candidate solutions α_j for x satisfies ψ.

If ψ happens to contain also strict inequalities, we need to add to our candidates for points in M_φ expressions of the form $\alpha \pm \varepsilon$, where α is candidate solution for some left-hand side polynomial occurring in a strict inequality. The symbol ε stands for a positive infinitesimal number. Again the substitution of these expressions into a polynomial equation or inequality can be rewritten as a quantifier-free formula.

By disjunctively substituting all candidates into ψ we obtain a quantifier-free formula φ^* equivalent to $\exists x \psi$ over the reals. In most cases it is not necessary to substitute all candidate terms. Instead of substituting each candidate solution, we compute a subset E of the set C of all candidate terms, such that

$$\exists x(\psi) \longleftrightarrow \bigvee_{t \in E} \gamma_t \wedge \psi(x /\!/ t),$$

where γ_t denotes the guard belonging to t. Each E satisfying the above condition is called elimination set for φ wrt. x.

Besides many other selection strategies for selecting small elimination set, we sketch only one approach which is particularly important for the local quantifier elimination. Recall that we test the non-emptiness of M_φ by testing, whether an interval boundary, possibly infinitesimal shifted belongs to M_φ. It is easy to see, that is sufficient to consider only either all lower bounds or all upper bounds of the intervals. For some of the candidate terms, we can actually decide, if they represents a lower bound or upper bound of an interval. Note that this property depends on the atomic formula from which the candidate term is computed.

The atomic formula $2x + c_0 > 0$, for instance, provides the lower bound $c_0/2$. This decision involves the sign of the coefficient of x and the type of the order relation. We can compute the boundary of an atomic formula whenever we are able to compute the sign of the highest coefficient c. Provided its sign is independent of the interpretation of the parameters, we can compute it by applying a decision procedure to $c < 0$, $c > 0$, and $c = 0$. In the REDLOG implementation we use some heuristics based on our simplification algorithm, cf. [6]. In order to obtain an elimination set from the set of all candidate solutions we can remove all candidate solutions that are lower bounds or upper bounds, respectively.

For practical applications this method, of course, has to be refined by a careful selection of a smaller number of candidate solutions and by a combination with powerful simplification techniques for quantifier-free formulas, cf. [6] for details.

Recall that the well-known solution formula for quadratic equations $c_2x^2 + c_1x + c_0 = 0$ requires $c_2 \neq 0$. In our situation c_2 is a term in $u_1, \ldots, u_n$, so $c_2 \neq 0$ can, in general, not be decided uniformly but depends on the interpretation of the u_i. Thus a quadratic polynomial $c_2x^2 + c_1x + c_0$ does not only deliver two square-root expressions α_1 and α_2 as candidate solutions but also $\alpha_3 = -c_0/c_1$, which in turn requires $c_1 \neq 0$. Let t_1, t_2, and t_3 be the candidate points for M_φ obtained from α_1, α_2, and α_3, respectively, by possibly adding or subtracting ε. With the substitution of the t_i into ψ, it is necessary to add the conditions on the non-vanishing of a and b. Formally, we obtain

$$\begin{gathered}(a \neq 0 \land \Delta \geq 0 \land (\psi[x/\!/t_1] \lor \psi[x/\!/t_2])) \lor \\ (a = 0 \land b \neq 0 \land \psi[x/\!/t_3]),\end{gathered}$$

where Δ denotes the discriminant of the equation $c_2x^2 + c_1x + c_0 = 0$. If, however, c_2 is a rational constant, then the case distinction is superfluous. In particular, if c_2 is non-zero, the second case can be dropped.

As indicated above, dramatic improvements of the general procedure sketched up to now can be obtained by reducing the number of test candidates for M_φ depending on the structure of the formula ψ, cf. [15, 19]. One simple instance for such an improvement is the following natural extension of *Gauss elimination*: Suppose ψ is of the form

$$bx + c = 0 \land \psi_1,$$

where at least one of the coefficient terms b, c is a rational non-zero constant. Then we know that under any interpretation of the u_i the equation is *non-trivial*, i.e. different from $0 = 0$. Hence the only test candidate required in the construction of φ^* is $-c/b$, substituted, of course, with the condition $b \neq 0$. No additional test candidates arising from equations or inequalities in the remainder ψ_1 of ψ need to be considered. This idea can easily be extended to a quadratic equation instead of a linear one, taking into account again the discriminant.

We have seen that it is convenient to be able to compute the sign of coefficients. To support decisions about signs, the elimination procedure may, more generally, allow as additional input a *theory* $\vartheta(u_1, \ldots, u_n)$. This is a conjunction of polynomial equations and inequalities in the parameters that may serve as a global hypothesis for the equivalence between $\exists x\psi$ and φ^*. In other words, the equivalence is asserted only for those real values of the u_i that satisfy ϑ. Then better elimination set computation, simpler substitution, and Gauss elimination can also be performed if the required coefficient conditions are part of the theory or can be automatically inferred from it.

Successive elimination of several existential and universal quantifiers by the method is possible as long as after each elimination step the degree of the next variable to be eliminated is at most 2 in the quantifier-free formula resulting from previous eliminations. Note that the elimination of an innermost variable in general increases the degree of the outer variables in the elimination result compared to the original matrix formula ψ. The REDLOG implementation of the quantifier elimination contains heuristics to cope with equations and inequalities with an x-degree greater than 2. A systematic extension of the method to arbitrary degrees has been sketched in [19]. The cubic case has been worked out in detail in [18].

3 Local Quantifier Elimination by Substitution

In the previous section we have discussed how to compute to a formula $\exists x \varphi(x, \underline{u})$ a quantifier-free formula $\varphi^*(\underline{u})$. In this section we present our local quantifier elimination algorithm which is derived from the quantifier elimination described in the previous section.

In contrast to regular quantifier elimination we distinguish between non-local parameters $u_1, \dots, u_m$ and local parameters $v_1, \dots, v_n$. For the latter we specify in the input of the local quantifier elimination the suggested point $(a_1, \dots, a_n) \in \mathbb{R}^n$. Recall that local quantifier elimination computes for φ and $\underline{a}$ a quantifier-free formula φ^* and a theory Θ, which is a conjunction of atomic formulas, such that

$$\Theta \longrightarrow (\varphi \longleftrightarrow \varphi^*) \quad \text{and} \quad \Theta(a_1, \dots, a_n).$$

Note that the condition $\Theta(\underline{a})$ guarantees, that Θ cannot become inconsistent. The constraints contained in Θ are generated accordingly the following scheme: Let t be a local term, $\underline{a} \in \mathbb{R}^n$, and define

$$R(s) = \begin{cases} > & \text{for } s = 1 \\ = & \text{for } s = 0 \\ < & \text{for } s = -1 \end{cases}.$$

We can automatically evaluate $t(\underline{a})$ and compute the sign s of $t(\underline{a})$. Defining

$$\theta_{\underline{a}}(t) \equiv t \; R(s) \; 0,$$

it follows obviously that $\theta_{\underline{a}}(t)(\underline{a})$ holds.

In the following subsections we discuss how to make use of a and Θ for speeding up the computation and for obtaining smaller output formulas. We consider modifications of three phases of the quantifier elimination: The computation of all candidate solutions, the virtual substitution of some candidate solutions, and the simplification of the result formula.

3.1 Local Computation of Candidate Solutions

Both computation time and output size of the quantifier elimination by virtual substitution depend heavily on the size of the computed elimination set. For an input formula of size $|\varphi|$ and an elimination set with size $|E|$ we compute, roughly speaking, an output formula of size $|\varphi| \cdot |E|$. Our major goal is thus to reduce the size of E. Recall, that we compute for each atomic formula in the input formula a candidate point and an appropriate guard. From the set of all candidate solutions we select then an elimination set.

To begin with, consider an atomic formula $\alpha \equiv c_1 x + c_0 \ \varrho \ 0$ where ϱ is one of the order relations and suppose that c_1 is a local term. Adding the constraint $\theta_{\underline{a}}(c_1)$ conjunctively to the theory Θ, ensures that the sign of c_1 is constant and known on the range of the local quantifier elimination. Hence we can decide whether α yields an upper, or a lower bound, or whether α is equivalent to $c_0 \ \varrho \ 0$. In the latter case we do not need to generate a candidate solution at all. The knowledge about the type of bound will support the elimination set computation. If c_1 is not local, i.e. it contains variables besides the parameters $\underline{v}$ we proceed as for regular quantifier elimination.

Constraints of the form $t > 0$ and $t < 0$, respectively are valid not only for the suggested point $\underline{a}$, but also for all points in a neighborhood of $\underline{a}$. This is obviously false for equation constraints. We have therefore introduced *restricted* local quantifier elimination, which assumes only strict order relations in the theory Θ. If $\theta_{\underline{a}}(c_1)$ is an equation constraint, we do not add it to Θ and we proceed as for regular quantifier elimination.

Next we consider a quadratic atomic formula $c_2 x^2 + c_1 x + c_0 \ \varrho \ 0$. Recall from Section 2 that in this case our regular quantifier elimination procedure generates first a condition that c_2 does not vanish together with at most two candidate solutions belonging to the roots of $c_2 x^2 + c_1 x + c_0$. Furthermore it generates a condition, that c_2 vanishes together with the candidate solution for the atomic formula $c_1 x + c_0 \ \varrho \ 0$.

If the highest coefficient c_2 is local we add $\theta_{\underline{a}}(c_2)$ to Θ. We do not consider the linear case provided that $\operatorname{sign}(c_2) \neq 0$ and we do not consider the pure quadratic case provided that $\operatorname{sign}(c_2) = 0$. If c_2 is not local we consider both the quadratic and the linear case separately. The linear case is discussed above. So we only have to clarify how to proceed in the pure quadratic case, i.e. we assume $c_2 \neq 0$. We start with the computation of the discriminant $\Delta = -4c_2c_0 + c_1^2$ and we check if it is local. If it is not local we proceed as usual. Otherwise we add the constraint $\theta_{\underline{a}}(\Delta)$ to Θ. If $\operatorname{sign}(\Delta) < 0$, which means that $c_2x^2 + c_1x + c_0$ does not have a real zero, we compute no candidate solution. If $\operatorname{sign}(\Delta) > 0$ we can drop the appropriate guard in front of the substitution result. If $\operatorname{sign}(\Delta) = 0$ we have to consider only the candidate solution $\frac{-c_1}{2c_2}$. Note that in this case, we either cannot determine the sign of c_2 or it is definitely different from 0. The restricted local quantifier elimination would, analogously to the linear case, not generate any equation constraints for Θ. In case that a local term evaluates to 0 the restricted quantifier elimination proceeds as the regular quantifier elimination. Thus, we have to code certain sign conditions into the output formula such that the output is, in general, not so short as in the unrestricted case.

In Section 2 we have introduced an extension of the Gauss elimination. It is applicable if a top-level equation is guaranteed to be non-trivial. For deciding $c_2x^2 + c_1x + c_0 = 0$ to be non-trivial, we check, starting with c_2, successively if $\operatorname{sign}(c_i(\underline{a})) \neq 0$. If $\operatorname{sign}(c_j(\underline{a})) \neq 0$ for one j we add $\theta_{\underline{a}}(c_j)$ to Θ and apply the Gauss elimination as usual.

Finally, we summarize the number of constraints added to Θ, for the elimination of $\exists x(\psi)$. We compute to each atomic formula in ψ at most two candidate solutions. Recall, that during the computation of a candidate solution we add at most two constraints to Θ. This means, that the size of Θ is in $O(n)$, where n is the number of atomic formula in φ. Of course, we simplify the obtained theory Θ at the very end of the quantifier elimination by applying our simplification algorithm.

3.2 Local Simplification

Simplification has been turned out to be crucial for a successful application of quantifier elimination. This suggests to modify the simplification algorithm in order to take advantage of the additional features of local quantifier elimination.

Only two steps of our simplifier are affected by the modifications for a local simplification. For a detailed discussion about simplification of quantifier free formulas, and a description of how, when, and where these two steps are applied in the simplification algorithm, see [6, 7].

We adapt the following two steps to our framework of local quantifier elimination:

- The potential simplification of an atomic formula to a truth value. For example our simplifier recognizes $x^2 + y^2 \geq 0$ to be equivalent to true.

- The potential simplification of a conjunction of two atomic formulas to a single atomic formula. For example we simplify $3x - 2 >= 0 \wedge 4x - 3 >= 0$ to $3x - 2 >= 0$. Our simplifier can combine only two atomic formulas of the form $t + q_1 \ \varrho_1 \ 0$ and $t + q_2 \ \varrho_2 \ 0$, where t is a term and q_1, q_2 are rational numbers. Note that not all conjunctions matching this condition, can be simplified.

In the next subsections we sketch how to improve these two steps in the framework of local quantifier elimination. We denote again with $\underline{a}$ the suggested point for the local parameters.

Consider a local atomic formula $t \ \varrho \ 0$, where ϱ is an arbitrary relation. Adding $\theta_{\underline{a}}(t)$ to Θ, the atomic formula is for all points in the range equivalent either to true or to false. The respective truth value can be computed using the sign of $t(\underline{a})$. We can, hence, replace the atomic formula by true or false, respectively. This supports the simplification in particular in the case of "true" in a disjunction or "false" in a conjunction. In this case a not necessarily local subformula can be replaced by one truth value.

Using this simplification one is faced with a larger growth of Θ than in the elimination phase. A larger Θ means in general a smaller range. Simplifying the result formula with the above sketched method can, in the worst case, add for each atomic formula in the input formula a new constraint to Θ. In our case the input formula is actually the output formula of the local quantifier elimination. This implies that in this case the size of Θ is in $O(n^2)$ instead of $O(n)$, where n is the number of atomic formula of the elimination input φ. The output formula, however, will be much shorter using this simplification.

A naive extension of our simplifier would apply this simplification for atomic formulas to each atomic formula contained in the input formula. Besides the disadvantage of the growth of Θ one has to deal with the problem of adding obviously unnecessary constraints to Θ. Consider, e.g. the subformula $v_1 > 0 \wedge v_2 > 0$ of a complex formula and suppose that $(a_1, a_2) = (1, 1)$. Then the simplifier would make the assumptions $v_1 > 0$ and $v_2 > 0$ but it is sufficient to assume either $v_1 > 0$ or $v_2 > 0$. This situation cannot be resolved uniformly, because the decision which constraint is more suitable depends of the given application. Adding further constraints to Θ means in general to further restrict the range. For a first implementation we suggest the heuristic to add successively some constraints until no more atomic formulas can be simplified this way. This avoids adding unnecessary constraints to Θ but cannot exclude that Θ is restricted much more than necessary, because this depends heavily on the added constraints and not on the number of the constraints in Θ.

Next we discuss how to combine two atomic formulas. For the general simplifier this was only possible for terms that differ only in the absolute summand. This concept can be easily extended in the framework of the local quantifier elimination. Here we can combine two atomic formulas of the form $t + p_1\ \varrho_1\ 0$ and $t + p_2\ \varrho_2\ 0$, where t is a term, that does not contain a local summand and both p_1 and p_2 are local terms. Adding $\theta_{\underline{a}}(p_1 - p_2)$ to Θ one can decide whether $p_1 < p_2$, $p_1 > p_2$, or $p_1 = p_2$. Using this information, we can straightforwardly generalized the techniques for simplifying conjunctions of our simplifier. As an example consider the formula $3x - v_1 - 1 >= 0 \wedge 4x - 2v_2 - 1 >= 0$ and suppose $(a_1, a_2) = (1, 1)$. Then we add the constraint $4v_1 + 4 < 6v_2 + 3$ and simplify the formula to $4x - 2v_2 - 1 >= 0$. As for the simplification of atomic formulas this simplification technique can add for each atomic formula in the input a new constraint to Θ.

Again it is easy to see, how to define a restricted local simplifier that does not assume any equation constraints.

3.3 Local Virtual Substitution

We can take advantage of the possibility to decide the sign of a local term for obtaining better substitution results. Here we can both consider the assumptions already made and assume new constraints. The former case does not increase the size of Θ. In the latter case we may increase the size of Θ and are faced with problems similar to those discussed in the description of local simplification.

Consider a candidate solution $\frac{-c_0}{c_1}$ and an atomic formula $b_1 x + b_0 > 0$. The candidate solution is necessary and valid only in the case $c_1 \neq 0$. Rewriting the substitution result without an denominator yields the formula $-b_1 c_0 c_1 - b_0 c_1^2 > 0$ guarded with the condition $c_1 \neq 0$. If we can strengthen the condition to $c_1 > 0$ or $c_1 < 0$, respectively, as in the case of the local quantifier elimination, the result can be simplified to $-b_1 c_1 - b_0 > 0$ or $-b_1 c_1 - b_0 < 0$, respectively. Note that for quotients obtained as candidate solutions of a linear equation Θ contains already one of the constraint $c_1 > 0$ and $c_1 < 0$. This observation can be easily generalized for the substitution into atomic formulas containing polynomials of an arbitrary x-degree and for the substitution of other terms containing denominators.

As discussed in the previous subsection local atomic formulas can alway be evaluated to true or false. Such atomic formulas are generated systematically for resolving the substitution of improper test points in atomic formulas. Consider for example the substitution of ∞ for x into an atomic formula $ax + b > 0$, where a and b are local. We obtain the result $a > 0 \vee a = 0 \wedge b > 0$. Both $a > 0$ and $b > 0$ are local and can by enlarging Θ appropriately simplified.

3.4 Local vs. Generic Quantifier Elimination

In this subsection we discuss the relation between generic quantifier elimination, cf. [8] and local quantifier elimination. Even if the specification of both variants of quantifier elimination are different, they are closely related to each other. Both variants generates automatically assumptions over terms in a specified subset of all parameters. These assumptions are collected conjunctively in a theory Θ. The equivalence between input and output of both local and generic quantifier elimination is restricted to the semi-algebraic set represented by Θ. The main different between the local and the generic quantifier elimination is the form of and the requirements on the theory. The generic quantifier elimination assumes only disequations. This implies on one hand that the equivalence holds for almost all parameter values. On the other hand it guarantees that Θ cannot become inconsistent. In case of the local quantifier elimination we restrict the range possibly to a zero-dimensional set. The restricted local quantifier elimination, however, does not assume equation constraints and thus the range has the full dimension of the local parameter space. The suggested point guarantees that the range contains an interesting part of the parameter space. In both variants Θ cannot become inconsistent, due to the requirement that Θ holds for the suggested point.

A second different between the generic and the local quantifier elimination is more technical. Generic quantifier elimination may generate constraints only for the elements of the actually chosen elimination set. The local quantifier elimination, in contrast, may generate constraint for each candidate solution. Though this implies that local quantifier elimination assumes more constraints, the number of generated constraints is for both variants linear in the number of atomic formulas in the input. This changes if one allows to generate constraints for the result simplification or for the substitution of an elimination term into an atomic formula as described above. Note that analogous techniques are possible for the generic quantifier elimination, too.

4 A Complexity Remark

In general local quantifier elimination has the same complexity than the regular quantifier elimination. However, in this section we give a series of examples with increasing number n of atomic formulas, such that the output formula of regular quantifier elimination as well as the generic quantifier elimination is exponential in n, whereas the output of the local quantifier elimination is only polynomial in n for an arbitrary suggested point.

We consider a parametric box in real n-space, which is given by the following formula:

$$\exists x_1 \cdots \exists x_n (\bigwedge_{i=1}^{n} a_i x_i \leq 1 \wedge b_i(x_i - 1) \leq 1).$$

We consider the following slightly modified input

$$\exists x_1 \cdots \exists x_n (\bigwedge_{i=1}^{n} a_i x_i \leq 1 \wedge b_i(x_i - 1) \leq 1 \wedge \Pi),$$

where Π is a quantifier-free formula in the parameters a_i and b_i.

For the elimination of x_n, regular quantifier elimination computes the following three candidate points:

$$-\infty, \quad \frac{1}{a_n}, \quad \frac{1+b_n}{b_n},$$

together with guards $a_n \neq 0$ and $b_n \neq 0$, respectively, for the latter two test points. Only all three points together form an elimination set. With other words, we cannot drop any candidate solution. The virtual substitution of the terms of the elimination set for x_i into the input formula results in the following formula:

$$\begin{aligned}
&\exists x_1 \cdots \exists x_{n-1} \Big(\\
&\quad \bigwedge_{i=1}^{n-1} a_i x_i \leq 1 \wedge b_i(x_i - 1) \leq 1 \wedge \\
&\qquad a_n \geq 0 \wedge b_n \geq 0 \wedge \Pi[x_n // \infty] \vee \\
&\quad \bigwedge_{i=1}^{n-1} a_i x_i \leq 1 \wedge b_i(x_i - 1) \leq 1 \wedge \\
&\qquad a_n \neq 0 \wedge b_n(a_n - a_n^2) \leq a_n^2 \wedge \Pi[x_n // \frac{1}{a_n}] \vee \\
&\quad \bigwedge_{i=1}^{n-1} a_i x_i \leq 1 \wedge b_i(x_i - 1) \leq 1 \wedge \\
&\qquad b_n \neq 0 \wedge a_n(1 + b_n - b_n^2) \leq b_n^2 \wedge \Pi[x_n // \frac{1+b_n}{b_n}] \Big).
\end{aligned}$$

Even the generic quantifier elimination cannot drop one of the three candidate solutions. It may, however, add the conditions $a_n \neq 0$ and $b_n \neq 0$ to the theory and remove these atomic formulas from the substitution result.

For the elimination of the next quantifiers we interchange them with the top-level disjunction. Thus we get 3 subproblems; each of them can be eliminated independently. Each of the subproblems has again the form of the considered input. Thus for the elimination of each quantifier we alway get 3 test points and we can always interchange the resulting disjunction and the remaining quantifiers. During the elimination we obtain a computation tree of depth n such that each node has exactly 3 successors. Altogether, this proves that the number of atomic formulas in the output of both regular and generic quantifier elimination is in $O(3^n)$, in other words it is exponential in n.

Next we consider the same elimination problem as an input of the local quantifier elimination procedure together with an arbitrary suggested point

$$(a_1, \dots, a_n, b_1, \dots, b_n) = (\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n).$$

We will discuss all possible cases: To begin with, we start with the worst case for our local quantifier elimination, namely $\mathrm{sign}(\alpha_n) = -\mathrm{sign}(\beta_n)$, where wlog. we may assume that $\mathrm{sign}(\alpha_n) = 1$. We consider, as above, the elimination of "$\exists x_n$." The local quantifier elimination adds then $\theta_{(\underline{\alpha},\underline{\beta})}(a_n)$ and $\theta_{(\underline{\alpha},\underline{\beta})}(b_n)$ to Θ. This restricts the range such that $a_n x_n \leq 1$ is an upper bound, whereas $b_i(x_i - 1) \leq 1$ is a lower bound. Due to the fact, that the constraints occur conjunctively on the top-level and that Θ guarantees $a_n \neq 0$, one can easily see, that, e.g. $\{\frac{1}{a_n}\}$ is an elimination set. The substitution result is

$$\exists x_1 \cdots \exists x_{n-1} \Big(\bigwedge_{i=1}^{n-1} a_i x_i \leq 1 \wedge b_i(x_i - 1) \leq 1 \wedge$$
$$b_n(a_n - a_n^2) \leq a_n^2 \wedge \Pi[x_n /\!/ \frac{1}{a_n}] \Big).$$

In all other cases either our elimination procedure computes either $\{\infty\}$ or $\{-\infty\}$ as an elimination set.

In all cases the substitution result matches the form of the input formula. Thus we can simply iterate the local quantifier elimination and we obtain in each iteration an elimination set containing only one element. Altogether we obtain a result formula, which is linear in n and a theory containing $2n$ constraints.

5 Implementation in REDLOG

For demonstrating the scope of local quantifier elimination we have implemented the most relevant ideas. This test implementation is implemented on top of REDLOG. REDLOG stands for "REDUCE logic" system. It provides an extension of the computer algebra system REDUCE to a computer logic system implementing symbolic algorithms on first-order formulas w.r.t. temporarily fixed first-order languages and theories. Underlying theories currently available are algebraic closed fields, real closed fields, and discretely valued fields. We use, of course, the context of real closed fields.

REDLOG origins from the research on quantifier elimination and simplification procedures. Successfully applying such methods to both academic and real-world problems, the authors have developed a large collection of formula-manipulating tools, many of which are meanwhile interesting in their own right:

- Numerous tools for comfortably inputing, decomposing, and analyzing formulas.
- Several techniques for the logical and algebraic simplification of formulas.
- Various normal form computations.
- Quantifier elimination computes quantifier-free equivalents for given first-order formulas.
- Generic quantifier elimination makes non-degeneracy assumptions on the parameters, which considerably speeds up the elimination.
- Extended (generic) quantifier elimination provides additional information such as satisfying sample points for existentially quantified formulas.
- Linear optimization techniques based on quantifier elimination.

For more information on REDLOG including example computations and a discussion about the design goals of REDLOG, cf. [5]. REDLOG is contained in REDUCE 3.7. Source code, documentation, references, and example computations are freely available on the WWW.[1]

Our implementation uses large parts of the context independent implemented quantifier elimination by virtual substitution. We have modified the procedures implementing the candidate solution computation and the substitution of quotients. The ideas regarding local simplification and local virtual substitution have not been implemented.

[1] `http://www.fmi.uni-passau.de/~redlog/`

6 Application Examples

We present some example computations. All computations have been performed on a SUN ULTRA 1 computer with 140 Mhz and a heap space of 32 MByte for REDUCE.

6.1 Generic Quadratic Equation

In the first toy example we demonstrate the behavior of the implemented local quantifier elimination in contrast to the regular quantifier elimination and to the generic quantifier elimination. We consider the input formula $\varphi \equiv \exists x(v_2x^2 + v_1x + v_0 > 0)$ and the point $\underline{a} = (1, 1, 1)$.

Local quantifier elimination computes the result $v_0 > 0$ together with the theory $4v_0v_2 - v_1^2 > 0$. Quantifier elimination of $\exists x(v_2x^2 + v_1x + v_0 > 0)[v_0/1, v_1/1, v_2/1]$ yields the result true, whereas regular quantifier elimination of $\exists x(v_2x^2 + v_1x + v_0 > 0)$ produces the result

$$\begin{aligned} v_2 > 0 \lor 2v_0v_1v_2 - v_1^3 > 0 \land v_1 \neq 0 \land v_2 = 0 \lor \\ \lor v_2 = 0 \land (v_1 > 0 \lor v_0 > 0 \land v_1 = 0) \lor \\ 4v_0v_2 - v_1^2 < 0 \land v_2 < 0. \end{aligned}$$

Generic quantifier elimination computes the result $4v_0v_2 - v_1^2 < 0 \lor v_2 \geq 0$ together with the theory $v_2 \neq 0$. The computation time is in all cases smaller than the smallest measurable time of 10 ms.

6.2 Generic Polygon

We consider the input formula

$$\exists x \exists y \Big(\bigwedge_{i=1}^{n} a_ix + b_iy \leq c_i \Big),$$

which describes, whether a convex polygon is non-empty.

We fix $n = 3$ and specify the a_i and b_j as local parameters, suggesting the point

$$(a_1, a_2, a_3, b_1, b_2, b_3) = (1, -3, 5, -7, 11, -13).$$

Local quantifier elimination produces in 50 ms the output formula

$$a_1b_2^2c_3 - a_1b_2b_3c_2 - a_2b_1b_2c_3 + a_2b_2b_3c_1 + a_3b_1b_2c_2 - a_3b_2^2c_1 \leq 0,$$

together with the theory

$$a_1 > 0 \wedge a_2 < 0 \wedge a_3 > 0 \wedge b_1 < 0 \wedge b_2 > 0 \wedge b_3 < 0 \wedge$$
$$a_1 b_2 - a_2 b_1 < 0 \wedge a_2 b_3 - a_3 b_2 < 0.$$

Regular quantifier elimination computes in 350 ms an output formula containing 78 atomic formula. Generic quantifier elimination computes in 450 ms the same output together with true as theory. Substituting the suggested point in the input the regular quantifier elimination computes in 10 ms the equivalent formula $8c_1 + 11c_2 + 5c_3 \geq 0$.

For $n = 10$ and the suggested point

$$(a_1, \ldots, a_n, b_1, \ldots, b_n) = (p_1, -p_2, p_3, \ldots, p_{2n}),$$

where p_i is the i-th prime number we obtain the following results: The local quantifier elimination computes in 2.6 s a theory containing 55 atomic formulas and an output containing 160 atomic formulas. The regular quantifier elimination computes in 26 s a formula containing 1520 atomic formulas. This formula is also computed in 33.5 s by the generic quantifier elimination together with true as theory. Fixing the point allows us to compute in 190 ms a result with 160 atomic formulas.

6.3 Kahan's Problem

Kahan's problem, cf. [13] is one of the most well-known benchmark problems for quantifier elimination procedures:

> *The problem concerns four variables a, b, c, d to be interpreted as center (c, d) and principal semiaxes a, b of an ellipse*
>
> $$E : (\frac{x-c}{a})^2 + (\frac{y-d}{b})^2 - 1 = 0.$$
>
> *We wish to know when E lies inside the unit disk*
>
> $$D : x^2 + y^2 \leq 1.$$

An optimal solution was computed by Lazard, cf. [14].

We consider here the special case $d = 0$, and suggest the point

$$a = 1/2, \quad b = 1/2, \quad c = 1/2.$$

Local quantifier elimination computes in 160 ms the result

$$a^2 + 2ac + c^2 - 1 \leq 0 \wedge a^2 - 2ac + c^2 - 1 \leq 0,$$

together with the theory

$$a^2 - b^2 = 0 \wedge a^2 > 0 \wedge b^2 c > 0 \wedge b^2 \neq 0.$$

Note that the theory does not imply the result formula. In other words, Θ does not restrict the problem to a trivial special case.

Regular quantifier elimination computes in 910 ms a formula containing 59 atomic formulas, and if we fix the suggested point we yield in 10 ms the result true. Generic quantifier elimination computes a formula containing 35 atomic formulas together with the theory $a + b \neq 0 \wedge a - b \neq 0 \wedge a \neq 0$.

7 Conclusions

We have introduced local quantifier elimination as a variant of real quantifier elimination. For local quantifier elimination we allow us to assume arbitrary order and equation constraints on local terms. As expected, this leads theoretically and practically to shorter output formulas than those produced by both regular and generic quantifier elimination. One consequence of the shorter (intermediate) results is the considerable speed-up of the elimination process. The suggested point for the local parameters guarantees that the range of the elimination is not empty and includes at least one point on which the user is interested in. Our concept of the restricted local quantifier elimination, guarantees furthermore that the range contains actually a neighborhood of the suggested point and has therefore the same dimension as the local parameter space. The theoretically expected improvements of local quantifier elimination in contrast to the regular quantifier elimination were exceeded by the results of our test implementation.

Wherever it suffices to restrict the equivalence of output and input to a neighborhood of the suggested the concept of local quantifier elimination is superior both to regular quantifier elimination and the generic quantifier elimination.

References

[1] Brown, C. W. Simplification of truth-invariant cylindrical algebraic decompositions. In *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation (ISSAC 98)* (Rostock, Germany, Aug 1998), O. Gloor, Ed., ACM, ACM Press, New York, pp. 295–301.

[2] Burhenne, K.-D. Implementierung eines Algorithmus zur Quantorenelimination für lineare reelle Probleme. Diploma thesis, Universität Passau, D-94030 Passau, Germany, Dec. 1990.

[3] Collins, G. E. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In *Automata Theory and Formal Languages. 2nd GI Conference* (1975), H. Brakhage, Ed., vol. 33 of *Lecture Notes in Computer Science*, Gesellschaft für Informatik, Springer-Verlag, Berlin, Heidelberg, New York, pp. 134–183.

[4] Collins, G. E., and Hong, H. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation 12*, 3 (Sept. 1991), 299–328.

[5] Dolzmann, A., and Sturm, T. Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin 31*, 2 (June 1997), 2–9.

[6] Dolzmann, A., and Sturm, T. Simplification of quantifier-free formulae over ordered fields. *Journal of Symbolic Computation 24*, 2 (Aug. 1997), 209–231.

[7] Dolzmann, A., and Sturm, T. P-adic constraint solving. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation (ISSAC 99), Vancouver, BC*, S. Dooley, Ed. ACM Press, New York, NY, July 1999, pp. 151–158.

[8] Dolzmann, A., Sturm, T., and Weispfenning, V. A new approach for automatic theorem proving in real geometry. *Journal of Automated Reasoning 21*, 3 (1998), 357–380.

[9] Dolzmann, A., Sturm, T., and Weispfenning, V. Real quantifier elimination in practice. In *Algorithmic Algebra and Number Theory*, B. H. Matzat, G.-M. Greuel, and G. Hiss, Eds. Springer, Berlin, 1998, pp. 221–247.

[10] Hong, H. An improvement of the projection operator in cylindrical algebraic decomposition. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 90)* (Tokyo, Japan, Aug. 1990), S. Watanabe and M. Nagata, Eds., ACM, ACM Press, New York, 1990, pp. 261–264.

[11] Hong, H. Simple solution formula construction in cylindrical algebraic decomposition based quantifier elimination. In *Proceedings of the*

International Symposium on Symbolic and Algebraic Computation (ISSAC 92) (Berkeley, CA, July 1992), P. S. Wang, Ed., ACM, ACM Press, New York, 1992, pp. 177–188.

[12] Hong, H. Generic quantifier elimination. Presented on the IMACS ACA 1995 in Albuquerque, NM, USA, 1995.

[13] Kahan, W. An ellipse problem. *ACM SIGSAM Bullitin 9*, 3 (Aug. 1975), 11. SIGSAM Probem #9.

[14] Lazard, D. Quantifier elimination: Optimal solution for two classical examples. *Journal of Symbolic Computation 5*, 1&2 (Feb. 1988), 261–266.

[15] Loos, R., and Weispfenning, V. Applying linear quantifier elimination. *The Computer Journal 36*, 5 (1993), 450–462. Special issue on computational quantifier elimination.

[16] McCallum, S. An improved projection operation for cylindrical algebraic decomposition of three-dimensional space. *Journal of Symbolic Computation 5*, 1–2 (Feb.–Apr. 1988), 141–161.

[17] Weispfenning, V. The complexity of linear problems in fields. *Journal of Symbolic Computation 5*, 1–2 (Feb.–Apr. 1988), 3–27.

[18] Weispfenning, V. Quantifier elimination for real algebra—the cubic case. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 94)* (Oxford, England, July 1994), ACM Press, New York, 1994, pp. 258–263.

[19] Weispfenning, V. Quantifier elimination for real algebra—the quadratic case and beyond. *Applicable Algebra in Engineering Communication and Computing 8*, 2 (Feb. 1997), 85–101.