数理解析研究所講究録 1335 巻 2003 年 91-96

Linear QE Algorithms and their Implementation on Maple

屋並仁史

YANAMI HITOSHI

(株) 富士通研究所

FUJITSU LABORATORIES LTD.*

Abstract

We show some known linear QE algorithms by virtual substitution, which were firstly proposed by Weispfenning in 1988. After that we present our experimental results based on our implementation of these algorithms on Maple 6.

1 What is QE?

A *quantifier elimination* (QE) procedure takes an arbitrary (first-order) formula as input and returns an quantifier-free formula equivalent to the input.

Examples

Assume that all variables are real.

(1) $\forall x(x^2+bx+c>0) \overset{QE}{\longmapsto} b^2-4c<0$

(2) $\exists x(x^2+bx+c>0) \overset{QE}{\longmapsto}$ 'True'

(3) $\exists x \exists y(y>2x+3 \wedge x>0 \wedge y<s) \overset{QE}{\longmapsto} s>3$

2 Preliminaries

We will explain linear QE algorithms based on the virtual substitution method. First we define terminology related to real quantifier elimination. Let V be an infinite set of variables and X a subset of V. We will use elements of X to represent quantified variables. Other symbols that can be used are: the relations $R = \{=, \neq, \leq, <\}$, the quantifiers $\exists x$ and $\forall x$ with $x \in X$, and the logical operators $T = \{\vee, \wedge, \neg\}$. Terms, atomic formulas, and formulas are constructed from these stuff.

Definition 1

Let V, X, R, and T be as above. A term is simply a polynomial $t \in \mathbb{Q}[V]$. An atomic formula is of the form $t_1 \, \rho \, t_2$, where t_1 and t_2 are terms and $\rho \in R$. In particular, an atomic formula of the form $t_1 \, \rho \, 0$ is said to be standard. Every atomic formula is equivalent to a standard one. A formula is defined as a Boolean combination of atomic formulas by operators in T preceded by a sequence of quantifiers.

*yanami@flab.fujitsu.co.jp

Terms:	$5,\ 3x-y,\ 7xy-2yz+s+3t$
Atomic formulas:	$3x-y<0,\ 7xy-2yz+1\le st-y$
Formulas:	$\forall x(x^2+bx+c>0),\ \exists x\exists y(7xy-1<s-y\ \wedge\ t+2ux^2y+3y^3<0)$

Quantifier elimination is a procedure that, for an input formula, returns a quantifier-free formula equivalent to the input. We can interchange the quantifiers $\forall x$ and $\exists x$ in a formula according to the following equivalence:

$$\forall x f(x) \Longleftrightarrow \neg(\exists x \neg f(x))\ .$$

Thus for a given formula we have an equivalent one of the form

$$(\neg)\exists x_1 \ldots (\neg)\exists x_n \varphi\ ,$$

where φ is quantifier-free and $(\neg)$ represents a possible negation operation. It is easy to eliminate the negation '$\neg$' in $\neg f$ with f quantifier-free; use De Morgan's laws and rewrite each atomic subformula. This procedure is not an essential part of quantifier elimination. In addition to that, a practical problem is mostly given in an *existential formula,* i.e., a formula of the form

$$\exists x_1 \cdots \exists x_n \varphi\ ,$$

where φ is quantifier-free. We assume from now on that the input is an existential formula. Thus our main purpose is to eliminate the existential quantifier $\exists x$ in $\exists x\varphi$ with φ quantifier-free.

We deal in the present paper with a class of formulas—the linear formulas—to which QE algorithms by virtual substitution are applicable. Here we give some more definitions.

Definition 2
A term is called linear if it can be written in the form

$$a_0 + a_1x_1 + \cdots + a_nx_n\ ,$$

where $x_1, \ldots, x_n$ are variables in X and $a_0, \ldots, a_n$ terms containing no variables in X. An atomic formula is called linear if it is, when expressed in its standard form, of the form

$$t\ \rho\ 0,\ \rho \in R$$

with t a linear term. A formula is called linear if every atomic subformula in it is linear.

Remark 1
Linearity is measured by the total degree with respect to X. Let $x, y \in X$. Then $xy + x + 1$ is linear with respect to either x or y, but it is not a linear term.

3 QE by Virtual Substitution

Quantifier elimination by virtual substitution was firstly proposed by Weispfenning in 1988. Following his paper [1], we explain the linear QE algorithm by virtual substitution.

Definition 3
Let φ be a quantifier-free formula, $x \in X$ a quantified variable, and S a finite set of terms, where each term $t \in S$ does not contain x. Then S is called an elimination set for $\exists x\varphi$ if the equivalence

$$\exists x\varphi \Longleftrightarrow \bigvee_{t\in S} \varphi(x//t)$$

holds, where $\varphi(x//t)$ is the formula obtained by a modified substitution.[1] *Elements of S are called test terms.*

Linear formulas are easy to treat in the sense that for a given linear formula $\exists x\varphi$ we can find an equivalent quantifier-free formula that is again linear, which enables us to eliminate all the quantifiers; eliminate them one by one from inside. The next lemma shows how we can take an elimination set for a linear formula.

Lemma 4 (Weispfenning [1])
Let φ be a linear quantifier-free formula, $x \in X$ a quantified variable in φ, and $\Psi = \{a_i x - b_i \ \rho_i \ 0 | i \in I, \rho_i \in \{=, \neq, \leq, <\}\}$ the set of atomic subformulas in φ. Then

$$S = \{\frac{b_i}{a_i}, \frac{b_i}{a_i} \pm 1 | i \in I\} \cup \{\frac{1}{2}(\frac{b_i}{a_i} + \frac{b_j}{a_j}) | i, j \in I, i \neq j\}$$

is an elimination set for $\exists x\varphi$, where S is regarded as a set of linear terms.

By using the above lemma, we can eliminate all the quantifiers in a linear existential formula. Now we can show an algorithm for QE procedure.

Procedure:	QE_Lin
Input:	An existential formula of the form $\exists x_n \ldots \exists x_1 \varphi$ with φ quantifier-free
Output:	A quantifier-free formula equivalent to the input

```
QE_Lin := proc(quantifier::list, qfreepart)

    n := nops(quantifier); # number of quantifiers
    qfreeformula[1] := qfreepart;

    for i from 1 to n do
        atom[i] := collect_atom(qfreeformula[i]);
        elim_set[i] := elimination_set(atom[i], quantifier[i]);

        qfreeformula[i+1] := substitute(qfreeformula[i], elim_set[i], quantifier[i]);
    end do;

    return qfreeformula[n+1];

end proc;
```

[1] There is a procedure assigning the expression $\varphi(x/t)$ obtained from φ by substituting t for x a formula equivalent to it. We denote the resulting formula by $\varphi(x//t)$.

4 Smaller Elimination Sets

The elimination set in Lemma 4 contains redundant test terms. Loos and Weispfenning [2] presented two types of elimination sets smaller than in Lemma 4. Using smaller elimination sets helps the number of atomic formulas not to grow too fast during the QE procedure, which contributes to increasing the algorithm's efficiency.

4.1 Optimization 1

Let $\exists x\varphi$ be a linear formula. Take an atomic formula in $\exists x\varphi$, say, $ax - b\ \rho\ 0$. When ρ is either $=$ or $\leq$, the test terms $\frac{b}{a} \pm 1$ are redundant; when ρ is either $\neq$ or $<$, the test term $\frac{b}{a}$ is of no use. So by treating $\{=, \leq\}$ and $\{<, \neq\}$ separately we can reduce the size of the elimination set.

Lemma 5 (Loos and Weispfenning [2])
Let φ, x and Ψ be as in Lemma 4. Partition Ψ into Ψ_1 and Ψ_2:

$$\Psi_1 = \{a_i x - b_i\ \rho_i\ 0 | i \in I_1, \rho_i \in \{=, \leq\}\} ,$$
$$\Psi_2 = \{a_i x - b_i\ \rho_i\ 0 | i \in I_2, \rho_i \in \{\neq, <\}\} .$$

Then the following set

$$S = \{\frac{b_i}{a_i} | i \in I_1\} \cup \{\frac{b_i}{a_i} \pm 1 | i \in I_2\} \cup \{\frac{1}{2}(\frac{b_i}{a_i} + \frac{b_j}{a_j}) | i, j \in I_2, i \neq j\}$$

is an elimination set for $\exists x\varphi$.

4.2 Optimization 2

Further investigation has been made in Loos and Weispfenning [2]. In Lemma 5 we need the set $\{\frac{1}{2}(\frac{b_i}{a_i} + \frac{b_j}{a_j}) | i, j \in I_2, i \neq j\}$ as test terms in case $\frac{b_j}{a_j}$ lies between $\frac{b_i}{a_i} - 1$ and $\frac{b_i}{a_i} + 1$ for some i and j with $i \neq j$. We can remove these test terms from the elimination set by changing the constant 1 in $\frac{b_i}{a_i} \pm 1$ into a value smaller than $\min_{i \neq j}\{|\frac{b_i}{a_i} - \frac{b_j}{a_j}|\}$. It is, however, impossible to determine a suitable real value, since $\frac{b_i}{a_i}$ might contain a variable. A new symbol ε, which behaves like a positive infinitesimal number in the hyperreal numbers, helps the situation.

We can also reduce the size of the second set $\{\frac{b_i}{a_i} \pm 1 | i \in I_2\}$ of test terms. Remove the 'points' $\{\frac{b_i}{a_i} | i \in I_2\}$ from $\mathbb{R}^1$ and one obtain a finite set of disjoint open segments. We only need to take one test term a segment. Introduce another symbol ∞, an analogue of infinity, and the terms $\{\frac{b_i}{a_i} - \varepsilon | i \in I_2\}$ as well as ∞ satisfy the demand. Note that these test terms work regardless of values assigned to the variables. Summarizing the above observation leads us to the following lemma.

Lemma 6 (Loos and Weispfenning [2])
Let φ, x and Ψ_i be as in Optimization 1. Then the following set

$$S = \{\frac{b_i}{a_i} | i \in I_1\} \cup \{\frac{b_i}{a_i} - \varepsilon, \infty | i \in I_2\}$$

is an elimination set for $\exists x\varphi$.

After substitution using the elimination set in Lemma 6, we obtain a representation including the new symbols ε and ∞. It should be equivalently rewritten in a formula before proceeding the elimination of the next quantifier. The following are examples of rewriting rules for atomic formulas:

$$a\infty - b \leq 0 \overset{def}{\Longleftrightarrow} (a = 0 \wedge 0 \leq b) \vee (a < 0) \ ,$$

$$a(p - \varepsilon) - b < 0 \overset{def}{\Longleftrightarrow} (a > 0 \wedge ap \leq b) \vee (a \leq 0 \wedge ap < b) \ .$$

See Loos and Weispfenning [2] for other rewriting rules. We note that in Lemma 6 the size of the elimination set is reduced to linear in the number of atomic formulas.

5 Experimental Results

We have implemented QE algorithms by using three types of elimination sets shown in the previous section. We have used the following six example problems to compare data.

Example Problems

```
problems := [
[[x], x<=2 and x>5], # 1
[[x,y], y>=x-1 and y>-x-2 and y<-3], # 2
[[x,y], y>=x and x>=0 and y<=0], # 3
[[x,y], y<2*x+2 and y<=-3*x+12 and y>(1/3)*x+5], # 4
[[x,y], y>2*x+3 and x>0 and y<s], # 5
[[x,y], y>=100/79*x+65/67 and y<=100/79*x+66/67 and
        y>-74/7*x-41/3 and y<-74/7*x-40/3], # 6
];
```

Table 1 shows the size of the elimination set at the last elimination stage; Table 2 shows the number of atomic formulas in the output quantifier-free formula. The subprocedure `elimination_set` in the `QE_Lin` procedure is implemented according to the elimination set in Lemmas 4, 5, or 6. Note that no simplification is implemented. Tables 1 and 2 clearly show that both types of optimization take effect, especially when there are not-equal ($\neq$) or less-than ($<$) relations in an input formula.

6 Future Work

We have presented some known linear QE algorithms based on virtual substitution and shown our experimental results on Maple implementation to see how elimination sets and formulas grow. In the present paper we have made our first and primitive implementation. Our research group has just set about developing a Maple toolbox for solving real constraints. The toolbox is named SyNRAC, which stands for a Symbolic-Numeric toolbox for Real Algebraic Constraints. (We gave this name after the conference.)

There are many things to do to improve our QE implementation. We have not implemented simplification algorithm so far. During a quantifier elimination procedure, the number of atomic subformulas are growing, so implementing simplification algorithms—it takes a quantifier-free formula as input and

Problem No.	1	2	3	4	5	6
No optimization	7	20	5	116	12	420
Separating $\{=, \leq\}$ from $\{\neq, <\}$	3	9	1	13	9	13
Using ∞ and ε	3	5	2	5	3	6

Table 1: The Size of Elimination Sets

Problem No.	1	2	3	4	5	6
No optimization	3	16	11	24	27	37
Separating $\{=, \leq\}$ from $\{\neq, <\}$	3	11	3	10	27	7
Using ∞ and ε	3	6	3	5	3	7

Table 2: The Number of Atomic Formulas

returns an equivalent quantifier-free formula that is simpler than the input—are of significant importance. We refer to [3] for various simplification methods. Though the meaning of 'simple' formulas varies, the number of atomic formulas in a formula is considered as a typical indicator of measuring simplicity. It is urgent for us to implement simplification algorithms.

Acknowledgements

The author thanks Volker Weispfenning for calling attention to another paper of his on a linear QE algorithm that can make elimination sets still smaller.

References

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

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

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