1991-02-08-A

Research

Institute for

Symbolic

Computation

LINZ

Johannes Kepler University, A-4040 Linz, Austria (Europe)

Publications / Reports

Speeding-up Quantifier Elimination by Gröbner Bases

Bruno BUCHBERGER, Hoon HONG

Technical Report

RISC-Linz Series no. 91-06.0

February 8, 1991

Sponsored by: Austrian Ministery of Science, proj. no. 613.523/3-27a/89 ("Gröbner Bases").

Speeding-up Quantifier Elimination by Gröbner Bases*

(Preliminary Report)

Bruno Buchberger Hoon Hong

February 8, 1991

Research Institute for Symbolic Computation
Johannes Kepler University
A-4040 Linz, Austria

Abstract

In this paper we investigate the possibility of using the first author's Gröbner bases algorithm for speeding up the CAD-based quantifier elimination algorithm (QE) which was discovered by Collins and improved by the second author. In particular, we use the Gröbner bases algorithm to preprocess input formulas of QE into equivalent, but often "triangularized", formulas. Then the improved QE algorithm utilizes such structure to complete quantifier elimination with partially built CAD's. Preliminary experiments show that this method often gives a significant speed-up, though sometimes it results in a slow-down. Further study is needed for explaining when and why the use of Gröbner bases is not helpful. Then based on the resulting understanding, we could be able to adapt the Göbner bases algorithm to our use.

1 Introduction

In this paper we investigate the possibility of using the first author's Gröbner bases algorithm [3] for speeding up the CAD-based quantifier elimination algorithm (QE) which was discovered by Collins [7] and improved by the second author [8, 5].

In particular, we use the Gröbner bases algorithm to preprocess input formulas of QE into equivalent, but often "triangularized", formulas. Then Hong's improved QE algorithm utilizes such structure to complete quantifier elimination with partially built CAD's.

*Sponsored by the Gröbner Bases Project of the Austrian Ministry for Science and Research.

Preliminary experiments show that this method often gives significant a speed-up, though sometimes it results in a slow-down. Further study is needed for explaining when and why the use of Gröbner bases is not helpful. Then based on the resulting understanding, we could be able to adapt the Göbner bases algorithm to our use.

We assume that the reader is familiar with Buchberger's Gröbner bases method, Collins' QE method, and Hong's improved QE method. For detailed expositions of these methods, see [4], [7, 2], and [8, 5] respectively.

The plan of the paper is as following: In Section 2 we elaborate the idea of using the Gröbner bases algorithm for preprocessing input formulas of quantifier elimination. In Section 3 we describe the hardware and the programs used in our experiments. In Section 4 we present preliminary experimental results.

2 Idea

As mentioned in the introduction, for a certain type of QE problems, we can use the Gröbner bases algorithm to preprocess the input formulas into forms more suitable to the QE algorithm.

Specifically consider an input formula containing a conjunction of equations:

$$F_1 = 0 \wedge \cdots \wedge F_n = 0.$$

Let $G = \{G_1, \ldots, G_m\}$ be a Gröbner basis of $F = \{F_1, \ldots, F_n\}$. Then the polynomials in G have the same common complex roots, and thus the same common real roots, as the polynomials in F since by definition the ideal generated by G is the same as the ideal generated by F. Therefore the conjunction above is equivalent to, and thus can be replaced by, the conjunction

$$G_1 = 0 \wedge \cdots \wedge G_m = 0.$$

Now if G has been produced from F using pure lexicographical ordering of power products, then it is "triangularized" (see Lemma 6.8 of [4]). Using the triangularized polynomials has two benefits:

- The number of projection polynomials may be reduced. For example, if we have two polynomial equations and the Gröbner basis contains only one polynomial of positive degree in the main variable, then the projection will not contain the resultant of two polynomials. This reduction in the number of projection polynomials would in itself reduce the time required for stack constructions.

- The equations of the polynomials of zero degree in the main variable may function as "constraints" which Hong's improved QE algorithm can utilize to complete quantifier elimination with a partially built CAD.

Offsetting these potential savings is the extra cost of Gröbner basis computation. Also sometimes Gröbner bases have more polynomials, and often with much higher degrees or larger coefficients or more terms than the original polynomials. Our experiments in the next section show that in spite of these extra costs the use of Gröbner bases often gives a significant overall speed-up, though sometimes it does result in a slow-down.

3 Test Environment

In this section we describe the programs and the hardware used in our experiments.

Both GB algorithm and QE algorithm have been implemented on top of the computer algebra system SAC-2 developed by Collins [6]. Specifically

GB Program: We used the implementation by Gebauer. Gebauer implemented four versions of programs with different features. We used the version with the following features:

- selection of S-polynomials by ordering,
- partial reduction of S-polynomials,
- application of the deletion rule, and
- application of the product criterion.

In all the experiments, we used pure lexicographic ordering of power products.

QE Program: We used the implementation by Hong [8]. This program allows the user to choose a projection operator and a cell-choice strategy. In all the experiments, we chose Lazard's projection operator [10] and the cell-choice strategy (TC,LD,HL) which stands for Trivial Conversion of sample points, Least Degree of minimal polynomial, and Highest Level. For the details, see [8].

The computer algebra system SAC-2 and both implementations of GB and QE algorithms were originally written in the programming language ALDES developed by Loos [11]. Since a compiler for ALDES is not available, we translated all the programs into C language.

All the experiments were carried out on a DEC station 5000 with 32 megabytes main memory running ULTRIX operating system. Out of 32 megabytes, 6 megabytes were used for list processing. The timings were obtained by using ULTRIX operating system's 60 Hz cpu-clock which is accurate up to $100/60 \approx 17$ milliseconds. Therefore when the clock reported 0 milliseconds, we took it as 8 milliseconds.

Problem	QE alone	GB and QE					Speedup
Intersect A	351200	1150	=	350	+	800	305.4
Intersect B	2065550	1183	=	583	+	600	1746.1
Random A	14583	1167	=	50	+	1117	12.5
Random B	170533	1283	=	83	+	1200	132.9
Ellipse A	171616	69658	=	8	+	69650	2.5
Ellipse B	?	?	=	100	+	?	?
Solotar A	11500	834	=	717	+	117	13.7
Solotar B	51900	555720	=	554720	+	1000	0.1
Collision A	47817	?	=	984	+	?	?
Collision B	?	?	=	6134	+	?	?

Table 1: Summary of Experiments

4 Preliminary Experimental Results

In this section we present the results of our preliminary experiments. We tested five QE problems with two different orderings of variables for each problem, obtaining ten sets of statistics. In the following ten sub-sections, we describe each experimental result in detail. But first we give in Table 1 a summary of the ten experiments with respect to the total times. All the timings are in milliseconds. Several experiments were aborted after 40 minutes of computation, and are indicated by the symbol '?'. For the experiments with the combined method (the column for GB and QE), we give three timings as:

$$T = T_{gb} + T_{qe},$$

where T_{gb} is the time taken for Gröbner basis computation, T_{qe} is the time taken for quantifier elimination of the preprocessed input formula, and T is the total time: $T_{gb} + T_{qe}$.

From the table we see that the use of GB was helpful for the following problems: Intersect A, Intersect B, Random A, Random B, Ellipse A, and Solotareff A.

For Solotareff B, the QE with the preprocessed input ran about 50 times faster than the QE with the original input, but the GB took very long, resulting in net slowdown.

For Collision A, the GB produced large polynomials which immediately blow up the projection phase of QE. For Collision B, the GB produced very large polynomials which make QE hopeless.

For Ellipse B, the GB produced a triangularized system, but its univariate polynomial is of degree 6, which makes stack construction over the cells defined by the roots of the polynomial very expensive.

Note that the ordering of variables makes significant difference for the computing time of QE. For Solotareff's problem, the ordering of variables also makes significant difference for the computing time of GB.

In the following ten sub-sections, we give the exact descriptions of the input formulas, the Gröbner bases, the output formulas, and the tables containing the statistics. In all the following tables, T_{total} is the total time in milliseconds, T_{proj} is the time taken for the projection phase of quantifier elimination, T_{stack} is the time taken for the stack construction phase of quantifier elimination, T_{simp} is the time taken for the formula simplification phase of quantifier elimination, N_{proj} is the number of projection polynomials produced, and N_{stack} is the number of stacks constructed.

4.1 Intersection A: $x < y < z$

	QE alone	GB and QE					Ratio
T_{total}	351200	1150	$=$	350	$+$	800	305.4
N_{proj}	18					8	2.3
T_{proj}	984					300	3.3
N_{stack}	531					7	75.9
T_{stack}	320613					283	1132.9
T_{simp}	6967					8	870.9

Table 2: Intersection: $x < y < z$

4.1.1 Input formula for QE

```
Order of variables: x,y,z.

(E z) [   x^2 - 1/2 y^2 - 1/2 z^2 = 0
       /\ x z + z y - 2 x = 0
       /\ z^2 - y = 0 ]
```

The three polynomials define three surfaces: the first is a cone spreading to both directions on the x-axis, the second is a plane twisted around the z-axis, and the third is a parabolic surface extended on the x-axis. Thus in this problem, we are interested in finding a condition for a point (x, y) above/on/below which the three surfaces intersect.

4.1.2 Gröbner basis

```
G1 = x^5 + 18 x^4 - 21 x^3 + 4 x^2 - 2 x
G2 = y + 5/19 x^4 + 88/19 x^3 - 144/19 x^2 + 32/19 x
```

```
G3 = z x - 7/38 x^4 - 127/38 x^3 + 59/19 x^2 - 11/19 x
G4 = z^2 + 5/19 x^4 + 88/19 x^3 - 144/19 x^2 + 32/19 x
```

Note that the Gröbner basis is triangularized in the sense that $G1 \in Q[x]$, $G2 \in Q[x,y]$, and $G3, G4 \in Q[x,y,z]$. Note also that we have the leading monomials x^5, y, and z^2, and thus the Gröbner basis has only finitely many (possibly no) common real zeros. We can also compute those common zeros by successively eliminating variables starting from x. Therefore we could have solved this problem without using the QE algorithm, but by simply reasoning with the common real zeros.

4.1.3 Output formulas

For the original input the QE algorithm produces the following quantifier free formula:

```
y^2 + y - 2 x^2 = 0 /\ y^3 + 2 x y^2 + x^2 y - 4 x^2 = 0
```

For the preprocessed input, the QE algorithm produces the following quantifier free formula:

```
   19 y + 5 x^4 + 88 x^3 - 144 x^2 + 32 x = 0
/\ [ x^3 + 19 x^2 - 2 x + 2 = 0 \/ x - 1 = 0 \/ x = 0 ]
```

Both formulas are equivalent. In fact we proved the equivalence by entering the following sentence to our QE system:

$$(\forall x)(\forall y)F_1 \Longleftrightarrow F_2$$

where F_1 is the output formula from the original input, and F_2 is the output formula from the preprocessed input. The QE program proved the sentence in 8 seconds. For each of the other problems following, we carried out the same test to check the equivalence of its two output formulas.

Note that we can easily simplify the second output formula by evaluating its first polynomial on $x = 1$ and $x = 0$, obtaining:

```
    [x = 0  /\  y = 0]
\/  [x = 1  /\  y = 1]
\/  [x^3 + 19 x^2 - 2 x + 2 = 0 /\
     19 y + 5 x^4 + 88 x^3 - 144 x^2 + 32 x = 0]
```

We can still make further simplification by computing a Göbner basis of the last two polynomials, obtaining:

```
    [x = 0  /\  y = 0]
\/  [x = 1  /\  y = 1]
\/  [x^3 + 19 x^2 - 2 x + 2 = 0 /\ 19 y - x^2 + 8 x + 14 = 0]
```

	QE alone	GB and QE					Ratio
T_{total}	2065550	1183	=	583	+	600	1746.1
N_{proj}	16					7	2.3
T_{proj}	1167					200	5.8
N_{stack}	389					7	55.6
T_{stack}	1930058					217	8894.3
T_{simp}	3683					8	460.4

Table 3: Intersection: $y < x < z$

4.2 Intersection B: $y < x < z$

4.2.1 Input formula for QE

```
Order of variables: y,x,z.

(E z) [   x^2 - 1/2 y^2 - 1/2 z^2 = 0
       /\ x z + z y - 2 x = 0
       /\ z^2 - y = 0 ]
```

This is the same problem as the last one, except that we switched the order of two free variables. This does not change the geometric or logical meaning of the formula, but it does affect the computation carried out by the QE and the GB algorithms.

4.2.2 Gröbner basis

```
G1 = y^5 - 26 y^4 - 15 y^3 + 24 y^2 + 16 y
G2 = x - 3/64 y^4 + 41/32 y^3 - 59/64 y^2 - 21/16 y
G3 = z y - 1/8 y^4 + 27/8 y^3 - 5/4 y^2 - 3 y
G4 = z^2 - y
```

4.2.3 Output formulas

For the original input, the QE algorithm produces the following quantifier free formula:

```
2 x^2 - y^2 - y = 0 /\ y x^2 - 4 x^2 + 2 y^2 x + y^3 = 0
```

For the preprocessed input, the QE algorithm produces the following quantifier free formula:

```
64 x - 3 y^4 + 82 y^3 - 59 y^2 - 84 y = 0
/\ [ y^3 - 25 y^2 - 40 y - 16 = 0 \/ y - 1 = 0 \/ y = 0 ]
```

We can simplify this output formula further by evaluating the first polynomial on $y = 1$ and $y = 0$, obtaining:

```
    [x = 0  /\  y = 0]
\/  [x = 1  /\  y = 1]
\/  [y^3 - 25 y^2 - 40 y - 16 = 0 /\
     64 x - 3 y^4 + 82 y^3 - 59 y^2 - 84 y = 0]
```

We can still make further simplification by computing a Göbner basis of the last two polynomials, obtaining:

```
    [x = 0  /\  y = 0]
\/  [x = 1  /\  y = 1]
\/  [y^3 - 25 y^2 - 40 y - 16 = 0 /\ 16 x - y^2 + 37 y + 28 = 0]
```

4.3 Random A: $x < y < z$

	QE alone	GB and QE	Ratio
T_{total}	14583	1167 = 50 + 1117	12.5
N_{proj}	19	5	3.8
T_{proj}	2050	117	17.5
N_{stack}	55	4	13.8
T_{stack}	11666	684	17.1
T_{simp}	8	8	1.0

Table 4: Random: $x < y < z$

4.3.1 Input formula for QE

```
Order of variables: x,y,z

(E x)(A y)(E z) [    4 x^2 + x y^2 - z + 1/4 = 0
                  /\ 2 x + y^2 z + 1/2 = 0
                  /\ x^2 z - 1/2 x - y^2 = 0 ]
```

The three polynomials are from Buchberger's survey paper on Gröbner bases [4]. We formed the sentence arbitrary using the polynomials.

4.3.2 Gröbner basis

```
G1 = x^7 + 29/4 x^6 - 17/16 x^4 - 11/8 x^3 + 1/32 x^2
     + 15/16 x + 1/4
G2 = y^2 + 112/2745 x^6 - 84/305 x^5 - 1264/305 x^4
```

```
     - 13/549 x^3 + 84/305 x^2 + 1772/2745 x + 2/2745
G3 = z - 1568/2745 x^6 - 1264/305 x^5 + 6/305 x^4 + 182/549 x^3
     - 2047/610 x^2 - 103/2745 x - 2857/10980
```

Note that the Göbner basis has the leading monomials x^7, y^2, and z, and thus it has only finitely many common real zeros. Therefore we could have concluded already, without using the QE algorithm, that the input formula is false since the variable y is universally quantified.

4.3.3 Output formulas

For both the original and the preprocessed input sentences, the QE algorithm reports that the sentences are false.

4.4 Random B: $z < y < x$

	QE alone	GB and QE					Ratio
T_{total}	170533	1283	=	83	+	1200	132.9
N_{proj}	30					5	6.0
T_{proj}	9416					117	80.5
N_{stack}	107					4	26.8
T_{stack}	149498					783	190.9
T_{simp}	8					8	1.0

Table 5: Random: $z < y < x$

4.4.1 Input formula for QE

```
Order of variables: z,y,x

(E z)(A y)(E x) [    4 x^2 + x y^2 - z + 1/4 = 0
                  /\ 2 x + y^2 z + 1/2 = 0
                  /\ x^2 z - 1/2 x - y^2 = 0 ]
```

This is the same sentence as the last one, except that we reversed the order of the variables.

4.4.2 Gröbner basis

```
G1 = z^7 - 1/2 z^6 + 1/16 z^5 + 13/4 z^4 + 75/16 z^3
     - 171/8 z^2  + 133/8 z - 15/4
G2 = y^2 - 19188/497 z^6 + 318/497 z^5 - 4197/1988 z^4
     - 251555/1988 z^3 - 481837/1988 z^2 + 1407741/1988 z
     - 297833/994
G3 = x + 4638/497 z^6 - 75/497 z^5 + 2111/3976 z^4
     + 61031/1988 z^3 + 232833/3976 z^2 - 85042/497 z
     + 144407/1988
```

4.4.3 Output formulas

For both the original and the preprocessed input sentences, the QE algorithm reports that the sentences are false.

4.5 Ellipse A: $a < b < c < x < y$

	QE alone	GB and QE	Ratio
T_{total}	171616	69658 = 8 + 69650	2.5
N_{proj}	40	37	1.1
T_{proj}	5700	2417	2.4
N_{stack}	1814	365	5.0
T_{stack}	146700	56300	2.6
T_{simp}	5983	17	351.9

Table 6: Ellipse: $a < b < c < x < y$

4.5.1 Input formula for QE

```
Order of variables: a,b,c,x,y

(E x)(E y)[   x^2 + y^2 - 1 = 0
           /\ b^2 (x - c)^2 + a^2 y^2 - a^2 b^2 = 0
           /\ a > 0
           /\ a < 1
           /\ b > 0
           /\ b < 1
           /\ c >= 0
           /\ c < 1   ]
```

The first two polynomial equations respectively define a unit circle centered at the origin and an ellipse of semi-axes a and b centered at $(c, 0)$. The problem is to find a condition for a, b, and c such that the circle and the ellipse intersect. In order to reduce the complexity of the problem, we also restricted a, b, and c to the ranges as defined by the six inequalities in the formula. This problem was motivated by a similar problem proposed by Kahan [9], where one is interested in a condition that the ellipse is inside the circle.

4.5.2 Gröbner basis

```
G1 = x^2 b^2 - x^2 a^2 - 2 x c b^2 + c^2 b^2 - b^2 a^2 + a^2
G2 = y^2 + x^2 - 1
```

In this case, the Gröbner basis computation is trivial rewriting: eliminating y^2 from the second polynomial equation of the input formula by using the first polynomial equation.

4.5.3 Output formulas

For both the original and the preprocessed input formulas, the QE algorithm produces the following quantifier free formula:

```
[  c + a - 1 >= 0
\/ [ b^2 - a >= 0 /\ b^2 c^2 + b^4 - a^2 b^2 - b^2 + a^2 >= 0 ] ]
/\ a > 0 /\ a - 1 < 0 /\ b > 0 /\ b - 1 < 0 /\
   c >= 0 /\ c - 1 < 0
```

4.6 Ellipse B: $a < b < c < y < x$

	QE alone	GB and QE	Ratio
T_{total}	?	? = 100 + ?	?
N_{proj}	117	?	?
T_{proj}	156817	?	?
N_{stack}	?	?	?
T_{stack}	?	?	?
T_{simp}	?	?	?

Table 7: Ellipse: $a < b < c < y < x$

4.6.1 Input formula for QE

```
Order of variables: a,b,c,y,x

(E y)(E x)[   x^2 + y^2 - 1 = 0
           /\ b^2 (x - c)^2 + a^2 y^2 - a^2 b^2 = 0
           /\ a > 0
           /\ a < 1
           /\ b > 0
           /\ b < 1
           /\ c >= 0
           /\ c < 1   ]
```

This is the same formula as the last one, except that we switched the order of two bound variables.

4.6.2 Gröbner basis

```
G1 = y^4 b^4 - 2 y^4 b^2 a^2 + y^4 a^4 + 2 y^2 c^2 b^4
     + 2 y^2 c^2 b^2 a^2 + 2 y^2 b^4 a^2 - 2 y^2 b^4
     - 2 y^2 b^2 a^4 + 2 y^2 b^2 a^2 + c^4 b^4 - 2 c^2 b^4 a^2
     - 2 c^2 b^4 + b^4 a^4 - 2 b^4 a^2 + b^4
G2 = x y^2 c a^2 + 1/2 y^4 b^2 - 1/2 y^4 a^2 + y^2 c^2 b^2
     + 1/2 y^2 c^2 a^2 + y^2 b^2 a^2 - y^2 b^2 - 1/2 y^2 a^4
     + 1/2 y^2 a^2 + 1/2 c^4 b^2 - c^2 b^2 a^2 - c^2 b^2
     + 1/2 b^2 a^4 - b^2 a^2 + 1/2 b^2
G3 = x y^2 b^2 - x y^2 a^2 + x b^2 a^2 - x b^2 - 3/2 y^2 c b^2
     - 1/2 y^2 c a^2 - 1/2 c^3 b^2 + 1/2 c b^2 a^2 + 3/2 c b^2
G4 = x c b^2 + 1/2 y^2 b^2 - 1/2 y^2 a^2 - 1/2 c^2 b^2
     + 1/2 b^2 a^2 - 1/2 b^2
G5 = x^2 + y^2 - 1
```

4.6.3 Output formulas

For both the original and the preprocessed input formulas, the QE algorithm was aborted after 40 minutes computation.

4.7 Solotareff A: $a < b < x < y$

	QE alone	GB and QE					Ratio
T_{total}	11500	834	=	717	+	117	13.7
N_{proj}	28					35	0.8
T_{proj}	367					283	1.2
N_{stack}	86					4	21.5
T_{stack}	7033					250	28.1
T_{simp}	3783					17	222.5

Table 8: Solotareff: $a < b < x < y$

4.7.1 Input formula for QE

```
Order of variables: a,b,x,y

(E x)(E y)[    3 x^2 - 2 x - a = 0
          /\  x^3 - x^2 - a x - 2 b + a - 2 = 0
/\  3 y^2 - 2 y - a = 0
/\  y^3 - y^2 - a y - a + 2 = 0
/\  1 <= 4 a
/\  4 a <= 7
/\ -3 <= 4 b
/\  4 b <= 3
/\ -1 <= x
/\  x <= 0
/\  0 <= y
/\  y <= 1  ]
```

This formula arises as a subcase of Solotareff's first problem for cubic polynomials [1]. In general, Solotareff's first problem is to find a polynomial of degree $n-2$ which best "approximates" a given polynomial of degree n within a given range.

4.7.2 Gröbner basis

```
G1 = a^3 - 11 a^2 + 35 a - 25
G2 = b^2 - 2/3 b a + 56/27 b - 1/3 a^2 + 16/27 a + 4/27
G3 = x + 81/1024 b a^2 - 459/512 b a + 3141/1024 b
     + 93/1024 a^2 - 527/512 a + 2241/1024
G4 = y - 3/32 a^2 + 17/16 a - 63/32
```

4.7.3 Output formulas

For the original input, the QE algorithm produces the following quantifier free formula:

```
a - 1 = 0 /\ 27 b^2 - 18 a b + 56 b - a^3 + 2 a^2 - 19 a + 29 = 0
/\ 4 a - 1 >= 0 /\ 4 a - 7 <= 0 /\ 4 b + 3 >= 0 /\ 4 b - 3 <= 0
```

For the preprocessed input, the QE algorithm produces the following quantifier free formula:

```
4 a - 1 >= 0 /\ 4 a - 7 <= 0 /\ 4 b + 3 >= 0 /\ 4 b - 3 <= 0
/\ [ a - 1 = 0 \/ a - 5 = 0 ]
/\ [ 27 b + 9 a + 2 = 0 \/ b - a + 2 = 0 ]
```

Both formulas can be easily simplified to

$$a = 1 \wedge 27b + 11 = 0.$$

4.8 Solotareff B: $b < a < x < y$

	QE alone	GB and QE	Ratio
T_{total}	51900	555720 = 554720 + 1000	0.1
N_{proj}	42	21	2.0
T_{proj}	500	600	0.8
N_{stack}	164	4	41.0
T_{stack}	14151	200	70.8
T_{simp}	36100	8	4512.5

Table 9: Solotareff: $b < a < x < y$

4.8.1 Input formula for QE

```
Order of variables: b,a,x,y

(E x)(E y)[    3 x^2 - 2 x - a = 0
          /\  x^3 - x^2 - a x - 2 b + a - 2 = 0
          /\  3 y^2 - 2 y - a = 0
          /\  y^3 - y^2 - a y - a + 2 = 0
          /\  1 <= 4 a
          /\  4 a <= 7
          /\ -3 <= 4 b
          /\  4 b <= 3
          /\ -1 <= x
          /\  x <= 0
          /\  0 <= y
          /\  y <= 1  ]
```

This is the same formula as the last one, except that we switched the order of two free variables.

4.8.2 Gröbner basis

```
G1 = b^6 - 10/9 b^5 - 2915/243 b^4 - 6724/19683 b^3
     + 830093/19683 b^2 + 286982/6561 b + 24299/2187
G2 = a - 537286851/1733427200 b^5 + 1274021541/1733427200 b^4
     + 2556627057/866713600 b^3 - 3694037643/866713600 b^2
     - 3091197667/346685440 b - 6499394527/1733427200
G3 = x + 1227687759/27734835200 b^5 - 897078969/27734835200 b^4
     - 8587423413/13867417600 b^3 - 1285658313/13867417600 b^2
     + 16457320943/5546967040 b + 42073006043/27734835200
G4 = y + 2835001539/13867417600 b^5 - 7132522149/13867417600 b^4
     - 12643778673/6933708800 b^3 + 19121535627/6933708800 b^2
     + 15435163363/2773483520 b + 9745294303/13867417600
```

4.8.3 Output formulas

For the original input, the QE algorithm produces the following quantifier free formula:

```
a - 1 = 0 /\ a^3 - 2 a^2 + 18 b a + 19 a - 27 b^2 - 56 b - 29 = 0
/\ 4 a - 1 >= 0 /\ 4 a - 7 <= 0 /\ 4 b + 3 >= 0 /\ 4 b - 3 <= 0
```

For the preprocessed input, the QE algorithm produces the following quantifier free formula:

```
4 a - 1 >= 0 /\ 4 a - 7 <= 0 /\ 4 b + 3 >= 0 /\ 4 b - 3 <= 0 /\
[ 27 b + 11 = 0 \/ b + 1 = 0 \/ b - 3 = 0 \/ 27 b + 47 = 0 ] /\
1733427200 a - 537286851 b^5 + 1274021541 b^4 + 5113254114 b^3
- 7388075286 b^2 - 15455988335 b - 6499394527 = 0
```

Both formulas can be easily simplified to

$$a = 1 \wedge 27b + 11 = 0.$$

4.9 Collision A: $a < t < x < y$

	QE alone	GB and QE	Ratio
T_{total}	47817	? = 984 + ?	?
N_{proj}	37	?	?
T_{proj}	4800	?	?
N_{stack}	671	?	?
T_{stack}	40249	?	?
T_{simp}	17	?	?

Table 10: Collision: $a < t < x < y$

4.9.1 Input formula for QE

```
Order of variables: a,t,x,y

(E t)(E x)(E y)[   1/4 (x - t)^2 + (y - 10)^2  - 1 = 0
                /\ 1/4 (x - a t)^2 + (y - a t)^2 - 1  = 0
                /\ t > 0
                /\ a > 0 ]
```

In this formula, the variable t stands for time. The first polynomial equation defines an ellipse of semi-axes 2 and 1, initially centered at $(0, 10)$, and moving horizontally with speed 1. The second polynomial equation defines another ellipse of the same size, initially centered at the origin, and moving with the velocity (a, a). Thus the problem is to find a condition on a such that the two ellipses collide or at least touch.

4.9.2 Gröbner basis

```
G1 = x^2 t^2 a^2 - 2/5 x^2 t^2 a + 1/5 x^2 t^2 - 16 x^2 t a
     + 80 x^2 - x t^3 a^3 - 3/5 x t^3 a^2 + 1/5 x t^3 a
     - 1/5 x t^3 + 16 x t^2 a^2 + 16 x t^2 a - 80 x t a
     - 80 x t + 5/4 t^4 a^4 + 3/10 t^4 a^2 + 1/20 t^4
     - 40 t^3 a^3 - 8 t^3 a + 2584/5 t^2 a^2 + 40 t^2
     - 3136 t a + 7680
G2 = y x a - y x + 1/2 y t - 5 y a - 1/8 x^2 t a^2
     + 1/20 x^2 t a - 1/40 x^2 t + x^2 a + 1/8 x t^2 a^3
     + 3/40 x t^2 a^2 - 1/40 x t^2 a + 1/40 x t^2
     - 11/8 x t a^2 - 5/8 x t a - 5 x a + 5 x - 5/32 t^3 a^4
     - 3/80 t^3 a^2 - 1/160 t^3 + 55/16 t^2 a^3 + 5/16 t^2 a
     - 271/10 t a^2 - 5/2 t + 121 a
G3 = y x t - 10 y x - 1/2 y t^2 + 50 y + 5/4 x^2 t a
     - 1/4 x^2 t - 10 x^2 - 5/4 x t^2 a^2 - 11/8 x t^2 a
     + 1/8 x t^2 + 55/4 x t a + 25/4 x t + 50 x
     + 25/16 t^3 a^3 + 11/16 t^3 a - 275/8 t^2 a^2
     - 25/8 t^2 + 271 t a - 1210
G4 = y t a - 10 y + 1/4 x t a - 1/4 x t - 5/8 t^2 a^2
     + 1/8 t^2 + 50
G5 = y^2 - 20 y + 1/4 x^2 - 1/2 x t + 1/4 t^2 + 99
```

4.9.3 Output formulas

For the original input formula, the QE algorithm produces the following quantifier free formula:

```
5 a^2 - 12 a + 6 <= 0 /\ a > 0
```

From this we obtain the range of a:

$$\frac{6-\sqrt{6}}{5} \leq a \leq \frac{6+\sqrt{6}}{5}.$$

For the preprocessed input formula, the QE algorithm was aborted after 40 minutes of computation.

4.10 Collision B: $a < y < x < t$

	QE alone	GB and QE	Ratio
T_{total}	?	? = 6134 + ?	?
N_{proj}	?	?	?
T_{proj}	?	?	?
N_{stack}	?	?	?
T_{stack}	?	?	?
T_{simp}	?	?	?

Table 11: Collision: $a < y < x < t$

4.10.1 Input formula for QE

```
Order of variables: a,y,x,t

(E y)(E x)(E t)[   1/4 (x - t)^2 + (y - 10)^2  - 1 = 0
                /\ 1/4 (x - a t)^2 + (y - a t)^2 - 1  = 0
                /\ t > 0
                /\ a > 0 ]
```

This is the same formula as the last one, except that we reversed the order of the bound variables.

4.10.2 Gröbner basis

```
G1 = x^4 a^4 - 4/5 x^4 a^3 + 14/25 x^4 a^2 - 4/25 x^4 a
     + 1/25 x^4 - 16/5 x^3 y a^3 + 32/25 x^3 y a^2
     - 16/25 x^3 y a + 8 x^2 y^2 a^4 - 16/5 x^2 y^2 a^3
     + 16/5 x^2 y^2 a^2 - 16/25 x^2 y^2 a + 8/25 x^2 y^2
     - 160 x^2 y a^4 + 64 x^2 y a^3 + 96/5 x^2 y a^2
     + 792 x^2 a^4 - 1584/5 x^2 a^3 - 2416/25 x^2 a^2
     + 16/25 x^2 a - 8/25 x^2 - 64/5 x y^3 a^3
     + 128/25 x y^3 a^2 - 64/25 x y^3 a + 256 x y^2 a^3
     - 512/5 x y^2 a^2 - 6336/5 x y a^3 + 12672/25 x y a^2
     + 64/25 x y a + 16 y^4 a^4 + 96/25 y^4 a^2 + 16/25 y^4
     - 640 y^3 a^4 - 384/5 y^3 a^2 + 9568 y^2 a^4
     + 9664/25 y^2 a^2 - 32/25 y^2 - 63360 y a^4 - 128 y a^2
     + 156816 a^4 + 3168/5 a^2 + 16/25
G2 = t y^2 a^5 - 2/5 t y^2 a^4 + 2/25 t y^2 a^2 - 1/25 t y^2 a
     - 20 t y a^5 + 8 t y a^4 - 4/5 t y a^3 + 99 t a^5
     - 198/5 t a^4 + 104/25 t a^3 - 2/25 t a^2 + 1/125 t a
     + 1/8 x^3 a^5 - 1/8 x^3 a^4 + 9/100 x^3 a^3 - 17/500 x^3 a^2
     + 9/1000 x^3 a - 1/1000 x^3 - 3/10 x^2 y a^4
     + 4/25 x^2 y a^3 - 7/125 x^2 y a^2 + 1/250 x^2 y
     + 1/2 x y^2 a^5 - 1/2 x y^2 a^4 + 9/25 x y^2 a^3
     - 17/125 x y^2 a^2 + 9/250 x y^2 a - 1/250 x y^2
     - 10 x y a^5 + 10 x y a^4 - 6/5 x y a^3 - 2/25 x y a^2
     + 99/2 x a^5 - 99/2 x a^4 + 141/25 x a^3 + 67/125 x a^2
     - 9/250 x a + 1/250 x - 6/5 y^3 a^4 + 16/25 y^3 a^3
     - 28/125 y^3 a^2 + 2/125 y^3 + 24 y^2 a^4 - 64/5 y^2 a^3
     + 72/25 y^2 a^2 - 594/5 y a^4 + 1584/25 y a^3
     - 1772/125 y a^2 - 2/125 y
G3 = t x y a - 125/4 t y^2 a^4 + 25/4 t y^2 a^3 + 5/4 t y^2 a^2
     - 9/4 t y^2 a + 625 t y a^4 - 125 t y a^3 - 12375/4 t a^4
     + 2475/4 t a^3 - 25/4 t a^2 + 5/4 t a - 125/32 x^3 a^4
     + 25/8 x^3 a^3 - 35/16 x^3 a^2 + 5/8 x^3 a - 5/32 x^3
     + 75/8 x^2 y a^3 - 25/8 x^2 y a^2 + 5/8 x^2 y a + 1/8 x^2 y
     - 125/8 x y^2 a^4 + 25/2 x y^2 a^3 - 35/4 x y^2 a^2
     + 5/2 x y^2 a - 5/8 x y^2 + 625/2 x y a^4 - 250 x y a^3
     - 25/2 x y a^2 - 12375/8 x a^4 + 2475/2 x a^3 + 285/4 x a^2
     - 5/2 x a + 5/8 x + 75/2 y^3 a^3 - 25/2 y^3 a^2 + 5/2 y^3 a
     + 1/2 y^3 - 750 y^2 a^3 + 250 y^2 a^2 + 7425/2 y a^3
     - 2475/2 y a^2 - 5/2 y a - 1/2 y
G4 = t x^2 - 375/2 t y^2 a^4 + 25/2 t y^2 a^3 - 25/2 t y^2 a^2
     - 25/2 t y^2 a + 4 t y^2 + 3750 t y a^4 - 250 t y a^3
     + 400 t y a^2 - 37125/2 t a^4 + 2475/2 t a^3 - 4035/2 t a^2
     + 5/2 t a - 4 t - 375/16 x^3 a^4 + 125/8 x^3 a^3
     - 55/4 x^3 a^2 + 31/8 x^3 a - 37/16 x^3 + 225/4 x^2 y a^3
     - 45/4 x^2 y a^2 + 47/4 x^2 y a + 5/4 x^2 y
     - 375/4 x y^2 a^4 + 125/2 x y^2 a^3 - 55 x y^2 a^2
     + 31/2 x y^2 a - 37/4 x y^2 + 1875 x y a^4 - 1250 x y a^3
     - 25 x y a^2 - 160 x y a - 37125/4 x a^4 + 12375/2 x a^3
     + 180 x a^2 + 1569/2 x a + 37/4 x + 225 y^3 a^3
     - 45 y^3 a^2 + 47 y^3 a + 5 y^3 - 4500 y^2 a^3
     + 900 y^2 a^2 - 640 y^2 a + 22275 y a^3 - 4455 y a^2
     + 3153 y a - 5 y
G5 = t x a^2 - 1/5 t x a - 4/5 t y a - 1/2 x^2 a^2 + 1/10 x^2
     - 2 y^2 a^2 + 2/5 y^2 + 40 y a^2 - 198 a^2 - 2/5
G6 = t^2 - 2 t x + x^2 + 4 y^2 - 80 y + 396
```

4.10.3 Output formulas

For both the original and the preprocessed input formulas, the QE algorithm was aborted after 40 minutes of computation.

5 Conclusion

In this paper we investigated the possibility of using the Gröbner bases algorithm for speeding up the CAD-based quantifier elimination algorithm. In particular, we used the Gröbner bases algorithm to preprocess input formulas of quantifier elimination. Preliminary experiments showed that this method often gives a significant speed-up (ranging 2 through 1700 times), though sometimes it results in a slow-down (about 10 times).

We plan to carry out further research in order to explain when and why Gröbner bases computation does not help quantifier elimination. Based on the resulting understanding, we could be able to specialize the Gröbner bases algorithm for the use as a preprocessor to quantifier elimination algorithm.

References

[1] N. I. Achieser. *Theory of Approximation.* Frederick Ungar Publishing Co., New York, 1956.

[2] D. S. Arnon, G. E. Collins, and S. McCallum. Cylindrical algebraic decomposition i: The basic algorithm. *SIAM J. Comp.*, 13:865–877, 1984.

[3] B. Buchberger. *An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-Dimensional Polynomial Ideal.* PhD thesis, Universitat Innsbruck, Institut fur Mathematik, 1965. German.

[4] B. Buchberger. Groebner bases: An algorithmic method in polynomial ideal theory. In N. K. Bose, editor, *Recent Trends in Multidimensional Systems Theory*, chapter 6. D. Riedel Publ. Comp., 1983.

[5] G. E. Collins and H. Hong. Partial CAD construction in quantifier elimination. Technical Report OSU-CISRC-10/89 TR45, Computer Science Dept, The Ohio State University, 1989. To appear in *Journal of Symbolic Computation.*

[6] G. E. Collins and R. G. K. Loos. SAC-2 system documentation. Technical report. In Europe available from: R. G. K. Loos, Universität Tübingen, Informatik, D-7400 Tübingen, W-Germany. In the U.S. available from: G. E. Collins, Ohio State University, Computer Science, Columbus, OH 43210, U.S.A.

[7] George E. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In *Lecture Notes In Computer Science*, pages 134–183. Springer-Verlag, Berlin, 1975. Vol. 33.

[8] H. Hong. *Improvements in CAD–based Quantifier Elimination.* PhD thesis, The Ohio State University, 1990.

[9] W. Kahan. Problem no. 9: An ellipse problem. *SIGSAM Bulletin of the Assoc. Comp. Mach.*, 9(35):11, 1975.

[10] D. Lazard. An improved projection for cylindrical algebraic decomposition. Unpublished manuscript, 1990.

[11] R. G. K. Loos. The algorithm description language ALDES (Report). *ACM SIGSAM Bull.*, 10(1):15–39, 1976.