A HOL Decision Procedure for Elementary Real Algebra

John Harrison*

University of Cambridge Computer Laboratory

Abstract. The elementary theory of real algebra, including multiplication, is decidable. More precisely, there is an algorithm to eliminate quantifiers which does not introduce new free variables or new constants other than rational numbers. Therefore if a closed term of elementary real algebra involves no constants other than the rational numbers, its truth or falsity can be determined automatically. Quite a number of interesting algebraic and geometric problems can be expressed in this decidable subset. In this paper we describe a HOL implementation of a quantifier-elimination procedure and give some preliminary results.

1 Introduction

The formulas of elementary (first-order) real arithmetic are those built up, using the usual logical operations and quantification over the real numbers, from atomic formulas involving the usual ordering relations, equality, and the operations of addition, subtraction, negation and multiplication, which are in turn built from real variables and rational constants. A quantifier elimination algorithm is one which takes a term and returns one provably equivalent to it but involving fewer quantifiers. A simple example of such an equivalence (though not usually thought of by schoolchildren as quantifier elimination!) is the test for the existence of a root of a quadratic equation:

$$\vdash (\exists x.\ ax^2 + bx + c = 0) \equiv b^2 - 4ac \geq 0$$

(Strictly speaking this does not fall within our subset but we can make it so by rewriting x^2 to xx.)

It was first proved by Tarski [6] that such a quantifier elimination procedure exists for the elementary theory of reals. His method was rather complicated and inefficient, and better algorithms were developed by Seidenberg [5] and Cohen [2] among others. More recently, researchers in computer algebra have been investigating the use of Collins' method of Cylindrical Algebraic Decomposition [3] to give more efficient quantifier elimination procedures. Even these algorithms tend to be doubly exponential in the number of quantifiers to be eliminated, i.e. of the form:

$$O(2^{2^{kn}})$$

---

* Supported by SERC Grant number 92700205

where n is the number of quantifiers and k is some constant. Very recent work [7] has improved this to 'only' being doubly exponential in the number of *alternations* of quantifiers.

The current version of HOL includes a library providing a (definitionally constructed) theory of real numbers, and this, combined with HOL's programmability, make it a viable system in which to implement such a decision procedure. Some initial steps towards this goal are covered in this paper. We aim to produce a HOL derived rule, i.e. ensure that each step of the reduction is justified by an actual HOL theorem. We broadly follow the algorithm given in Kreisel and Krivine's book [4]. We have made a few simplifications and corrections, which are noted where significant.

2 Preliminary simplification

In this section we note how the general problem can be reduced to one particularly simple case by exploiting a few logical identities.

2.1 Arbitrary real closed fields

For the sake of completeness (no pun intended) we should note that none of the following depends on the fact that the underlying field is the real numbers. In fact any so-called *real closed* field will do, although our work in HOL does not make this generalization.

A field is said to be *formally real* if whenever a sum of squares is zero, all the numbers in the sum are zero (equivalently, -1 is not expressible as a sum of squares). A *real closed* field is one which is formally real, but has no formally real proper algebraic extension (an algebraic extension results from adjoining to the field the roots of polynomial equations).

That the reals are real closed is easily seen. In any real closed field one can prove certain key properties (e.g. every polynomial of odd degree has a root) which are enough to make the procedure we describe work [8].

2.2 Adding some eliminable extensions

We can extend the permissible subset to include a few other operations which can be eliminated in terms of those in the strict subset. Terms raised to the power of a numeral can be rewritten in terms of repeated multiplication:

$$x^n = xx \dots x$$

The `abs` function can simply be rewritten with its definition:

$$\vdash abs(x) = (0 \leq x) \rightarrow x \mid -x$$

Conditional expressions can be pulled up through terms using the theorems:

$$\vdash f(b \to x \mid y) = b \to f(x) \mid f(y)$$

and

$$\vdash (b \to f \mid g)x = b \to f(x) \mid g(x)$$

until they reach the 'atomic formula' level (of course HOL really has no syntactic notion of formulas vs. terms, so it would be more accurate to say 'boolean term level'). At this level they can be eliminated using the following theorem:

$$\vdash b \to x \mid y = (b \wedge x) \vee (\neg b \wedge y)$$

2.3 Eliminating superfluous relations and functions

The ordering relations are heavily redundant, and we can reduce them all to $>$ alone by rewriting with:

$$\begin{aligned} &\vdash x \geq y = \neg y > x \\ &\vdash x \leq y = \neg x > y \\ &\vdash x < y = y > x \end{aligned}$$

It is convenient to get rid of subtraction and negation, by treating -1 as a constant in the remainder of the elimination procedure:

$$\begin{gathered} \vdash x - y = x + (-1)y \\ \vdash -x = (-1)x \end{gathered}$$

Finally we can get rid of boolean equality (shown here as logical equivalence for emphasis) and implication:

$$\begin{aligned} &\vdash x \equiv y = (x \Longrightarrow y) \wedge (y \Longrightarrow x) \\ &\vdash x \Longrightarrow y = \neg x \vee y \end{aligned}$$

2.4 Normalizing polynomial relations

We can make the right-hand sides of equations and inequalities zero by rewriting with:

$$\begin{gathered} \vdash x > y = x + (-1)y > 0 \\ \vdash (x = y) = (x + (-1)y = 0) \end{gathered}$$

Now we can distribute multiplication over addition:

$$\vdash x(y+z) = xy + xz$$
$$\vdash (x+y)z = xz + yz$$

So now the 'atomic formulas' are all either equations or inequalities with $>$, in either case with 0 on the right and a polynomial (in general, in several variables) on the left. The only logical operations are conjunction, disjunction and negation.

2.5 Reducing to simple existential elimination, pushing in negations

Clearly it is sufficient to be able to eliminate the innermost quantifier, since we can then apply the procedure repeatedly (we get a logical equivalence at each stage). Furthermore, universal quantifiers can be turned into existential ones via

$$\vdash \forall x.\ P[x] = \neg\exists x.\ \neg P[x]$$

Hence we only need to eliminate a single existential quantifier from a formula transformed according to the above sections. We can push negations down a term using the De Morgan laws:

$$\vdash \neg\neg P = P$$
$$\vdash \neg(P \wedge Q) = \neg P \vee \neg Q$$
$$\vdash \neg(P \vee Q) = \neg P \wedge \neg Q$$

Furthermore, when the negations, if any, reach the 'atomic formula' level, they can be eliminated completely, since:

$$\vdash \neg(p(x) = 0) = p(x) > 0 \vee (-1)p(x) > 0$$
$$\vdash \neg p(x) > 0 = (p(x) = 0) \vee (-1)p(x) > 0$$

Now we can distribute conjunction over disjunction:

$$\vdash x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$
$$\vdash (x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$$

and can push the existential quantifier down through the disjuncts:

$$\vdash (\exists x.P[x] \vee Q[x]) = (\exists x.P[x]) \vee (\exists x.Q[x])$$

Therefore our problem is reduced to eliminating a single existential quantifier whose body is a conjunction of atomic formulas of the form $p(x) = 0$ or $p(x) > 0$, where x is the quantified variable and $p(x)$ is a polynomial in x with, in general, coefficients composed of other variables.

2.6 Reduction to bounded existential quantifier

For this step we use the following theorem:

$$\vdash (\exists y.\ P[y]) = (\exists u.\ 0 < u < 1 \land (\exists x.\ -1 < x < 1 \land P(u^{-1}x))$$

To see the truth of this, consider left-to-right and right-to-left implications, and pick witnesses for the antecedents. The right-to-left implication is trivial: set $y = u^{-1}x$. For the other direction, choose $u = 1/(|y|+2)$ and $x = y/(|y|+2)$.

Now it is sufficient to have a procedure which will eliminate an existential quantifier restricted to a finite open interval, i.e.

$$\exists x.\ a < x < b \land P[x]$$

and produce a formula which still corresponds to our rules. We can transform a general existentially quantified goal using the above theorem, then eliminate the inner existential quantifier treating u^{-1} as just another variable or constant. But since the result of this stage is built up from atomic formulas of the form $p(u^{-1}) = 0$ and $p(u^{-1}) > 0$, where the p's are polynomials, and since we have in context that $0 < u$, we can multiply these equations and inequalities through by any necessary power of u giving a logically equivalent form. Then we have another bounded existential quantifier in the chosen subset:

$$\exists u.\ 0 < u < 1 \land R[u]$$

to which our procedure can be applied.

3 Degree reduction

By the *degree* of a variable x in a polynomial $p(x)$ we shall mean the highest power of x occurring in some monomial in $p(x)$. For example in $x^3 + 3xy^2 + 8$, the variables x, y and z have degrees 3, 2 and 0 respectively. Note that this might more accurately be called the 'formal degree' since we do not exclude the possibility that the coefficient of the relevant monomial might be zero for some or all values of the other variables.

We need to deal with terms of the following form:

$$p_1(x) = 0 \land \ldots \land p_k(x) = 0 \land q_1(x) > 0 \land \ldots \land q_l(x) > 0$$

which occur in the body of our existential terms (together with a bound on the quantified variable x, but we can forget about that for the present stage). It is a crucial observation that we can reduce such a term to a logically equivalent disjunction of similar terms, but each disjunct having the following properties:

- There is at most one equation (i.e. k = 0 or k = 1 in the above scheme)

- The degree of x in the equation, if any, is no more than the lowest degree of x in any of the original equations whose leading coefficient is known to be nonzero[2].
- If there is an equation, then the degree of x in all the inequalities is strictly lower than its degree in the equation.

To see this, observe that first we can repeatedly case-split over whether the leading coefficients of the lowest-degreed equation are zero. This gives a disjunction of similar terms, where either the equations all disappear completely (by this colourful phrase we really mean that x is no longer free in them, and hence we are at liberty to pull them outside the quantifier), in which case we are finished, or we have a situation where the leading coefficient of the lowest-degreed equation is nonzero.

3.1 Degree reduction of other equations

If the lowest-degreed equation, say $p_1(x) = 0$ has degree n, then we have that $p_1(x)$ is $ax^n + r_1(x)$ for some a which by the above remarks we can assume to be nonzero. But now if there is any other equation, say $p_2(x)$, it must have the form $bx^m + r_2(x)$ where, since $p_1(x)$ was chosen to be of least degree, $m \geq n$. However the following, since $a \neq 0$, is easily seen to be a logical equivalence.

$$\vdash (p_1(x) = 0) \wedge (p_2(x) = 0) = (p_1(x) = 0) \wedge (bx^{m-n}p_1(x) - ap_2(x) = 0)$$

But now we have reduced the degree of the latter equation[3]. If $p_1(x)$ still has lower degree, we can repeat the procedure; if the second equation disappears then we can attack the next equation, and finally if the second equation now has lower degree, we can case-split on that one's leading coefficients and start eliminating using it. Eventually we have at most one equation with x free.

3.2 Degree reduction of inequalities

If we have one equation $p_1(x)$ of the form $ax^n + r_1(x)$ left, then we may again suppose that $a \neq 0$. Now if the polynomial on the left of an inequality $q_1(x)$, say, is of the form $bx^m + s_1(x)$ with $m \geq n$, we can reduce its degree using the following:

$$\vdash (p_1(x) = 0) \wedge q_1(x) > 0 = (p_1(x) = 0) \wedge (a^2 q_1(x) + (-1)abp_1(x) > 0)$$

---

[2] Kreisel and Krivine assert that the degree can be made no greater than the lowest degree in the original equations. This is false if they mean 'formal degree' since an equation all of whose coefficients are zero cannot be used for elimination.

[3] On the other reading of 'degree' the corresponding statement of Kreisel and Krivine that if a is zero then deleting it from $p_1(x)$ reduces its degree is false. So they lose one way or the other; I have not been able to determine for certain which meaning they do attach to 'degree'.

which is again easily seen to be true (observe that $0 < a^2$ because $a \neq 0$). It is convenient to finish off the elimination step by case-splitting over the leading coefficients in the inequalities, so we may thereafter assume them to be nonzero.

4 The main part of the algorithm

We shall find it convenient to consider formulas of the following three kinds:

- I: $\exists x.\ a < x < b \wedge (p(x) = 0) \wedge q_1(x) > 0 \wedge \ldots \wedge q_l(x) > 0$
- II: $\exists x.\ a < x < b \wedge q_1(x) > 0 \wedge \ldots \wedge q_l(x) > 0$
- III: $\forall x.\ a < x < b \Longrightarrow q_1(x) > 0 \wedge \ldots \wedge q_l(x) > 0$

We only want to eliminate quantifiers from the first two kinds (in the former, we allow the case where there are no inequalities, i.e. $l = 0$), but the third form arises as an intermediate result and it is useful to consider it separately. The basic idea of the algorithm is to transform between these forms, making polynomials which occurred in inequalities in one form appear in equations in another. We can then apply the degree-reduction step dealt with above and therefore get a term with a lower overall degree. This step is then repeated. To make the notion of decreasing degree rigorous, we will, following Kreisel and Krivine, make the following formal definition of the degree of a variable in a term:

- The degree of x in $p(x) = 0$ is the degree of x in $p(x)$
- The degree of x in $q(x) > 0$ is *one greater than* the degree of x in $q(x)$
- The degree of x in a non-atomic formula is the highest degree of x in any of its atoms.

It is clear that if the body of a quantifier has zero degree in the quantified variable, the elimination of the quantifier is trivial, the following being the case:

$$\vdash (\exists x.\ A) = A$$

Therefore we will have a terminating algorithm for quantifier elimination if we can show that quantifier elimination from a formula in any of these classes can be reduced to the consideration of finitely many other such formulas with *strictly lower degree*. Sometimes we will need to transform a formula several times to make this true. Note also that the subproblems of lower-degree eliminations are not all independent. In fact the elimination of one quantifier may result in the production of several nested quantifiers, but the elimination of each one of these always involves a formula of strictly smaller degree. Let us consider the cases in reverse order.

4.1 Reduction of Type III Terms

Observe that if c lies between a and b, the following is true:

$$\vdash (\forall x.\, a < x < b \Longrightarrow \bigwedge_{i=1}^{l} q_i(x) > 0) = (\bigwedge_{i=1}^{l} q_i(c) > 0 \wedge \neg \exists x.\, a < x < b \wedge \bigvee_{i=1}^{l} q_i(x) = 0)$$

This relies on the Intermediate Value Theorem: if a polynomial is positive at one value of x but not at another, it must attain zero at some intermediate value of x. In particular we may choose $c = 2^{-1}(a + b)$, the midpoint of the interval [4]. Now the (bounded) existential quantifier can be pushed through the disjuncts and we are reduced to several type I terms of lower degree. We have introduced the constant 2^{-1} but this is unproblematic since we can do rational arithmetic when quantifier elimination is finished.

4.2 Reduction of Type II Terms

The key observation here is that, again as a consequence of the continuity of polynomials, the set of points at which a polynomial (and by induction any finite set of polynomials as considered here) is positive is *open* in the topological sense. This means that given any point in the set, there is some nontrivial surrounding region which is also contained entirely in the set:

$$open(S) = \forall x \in S.\ \exists \delta > 0.\ \forall x'.\ |x' - x| < \delta \Longrightarrow x' \in S$$

This means that if a set of polynomials are all positive at a point, as a Type II term asserts, then they are positive in some interval surrounding that point (and conversely, obviously). The idea behind the reduction step is that we can choose this interval to be as large as possible, subject to its being within the bound associated with the quantifier. One possibility is that the polynomials are positive throughout the interval. But then we have a Type III term which, as we have already seen, can be reduced to one of lower degree. Otherwise one end is, or both ends are, properly contained within the quantifier bounds, and it is easy to see that some polynomial in the set must be zero at any such points (otherwise, by the open set property, the interval could be made larger). This has allowed us to turn an inequality into an equation as we wanted. Consider the formal statement of the reduction theorem:

$$\begin{aligned}
\vdash\ & (\exists x.\ a < x < b \wedge \textstyle\bigwedge_{i=1}^{l} q_i(x) > 0) \\
& \equiv \exists a'\ b'.\ a \leq a' < b' \leq b \wedge \\
& \qquad (\forall x.\ a' < x < b' \Longrightarrow \textstyle\bigwedge_{i=1}^{l} q_i(x) > 0) \wedge \\
& \qquad (a' = a \vee \textstyle\bigvee_{i=1}^{l} q_i(a') = 0) \wedge \\
& \qquad (b' = b \vee \textstyle\bigvee_{i=1}^{l} q_i(b') = 0)
\end{aligned}$$

---

[4] The first edition of Kreisel and Krivine made the mistake of thinking $\bigwedge_{i=1}^{l} q_i(a) \geq 0$ adequate. The second edition has a rather complicated alternative, roughly, that the first nonzero derivative at a is positive. Apart from being simpler, our solution generalizes to infinite intervals, as noted in the later section on optimizations.

We have already seen that Type III terms can be reduced to terms of lower degree, so we can eliminate the inner quantifier over x. We can then distribute over the disjunction and get in general many different clauses. The point is that even if the elimination of x has resulted in higher-degree terms in a' and/or b', they can be reduced to the levels of a simple equality or one of the q_j's using the range-reduction step. (We may assume that each of the q_j's has nonzero leading coefficient because of the previous elimination step.) Thus a' and b' only need be eliminated from lower-degree terms.

4.3 Reduction of Type I terms

The reduction of these terms has the same idea behind it as in the previous section, but the details are slightly more complicated since we need to consider the derivative of the polynomial appearing in the equation. Suppose that we have a point x where $p(x) = 0$ and all the $q_i(x)$ are positive. There are three possibilities for the derivative $p'(x)$. The simplest is that it is zero. In this case we can perform an elimination and get a term of lower degree (note that the leading coefficient of the derivative is known to be nonzero, because it is a positive integer multiple of the leading coefficient of the original polynomial). The more complicated cases arise where the derivative is either strictly positive or strictly negative at x. We shall consider the former; the latter can be treated in essentially the same way.

The derivative is also a polynomial, and is continuous, and is positive at x. We can therefore consider the larger group of inequalities resulting from its addition to the original ones. All these polynomials are strictly positive at x, and so as last time we can find a maximum interval in which this is true. Furthermore, since $p'(x)$ is positive throughout the interval, and $p(x)$ is zero somewhere within it, then $p(x)$ must be strictly negative at the left-hand end, and strictly positive at the right-hand end. Conversely, if this is the case, then there must be a point within the interval at which the polynomial $p(x)$ cuts the x-axis (again using the Intermediate Value Theorem). This allows a similar reduction to the above.

5 The HOL Implementation

When implementing any sort of derived rule in HOL, it is desirable to move as much as possible of the inference from 'run time' to the production of a few proforma theorems which can then be instantiated efficiently. To this end, we have defined encodings of common syntactic patterns in HOL which make it easier to state proforma theorems of sufficient generality.

5.1 Syntactic encodings

Firstly, we define a constant `poly` which takes a list as an argument and gives a polynomial in one variable with that list of coefficients, where the head of the list corresponds to the constant term, and the last element of the list to the term of highest degree.

```
|- (poly [] x = &0) /\
   (poly (CONS h t) x = h + (x * poly t x))
```

This immediately has the benefit of letting us prove quite general theorems such as 'every polynomial is differentiable' and 'every polynomial is continuous'. Differentiability of polynomials can be defined as a simple recursive function on the list of coefficients. The operations of addition, negation and constant multiplication can likewise be defined in an easy way in terms of the list of coefficients. For example, addition can be executed by rewriting with the following clauses:

```
|- (poly_add [] m = m) /\
   (poly_add l [] = l) /\
   (poly_add (CONS h1 t1) (CONS h2 t2) =
        (CONS (h1 + h2) (poly_add t1 t2)))
```

and we have the theorems

```
|- !l x. --(poly l x) = poly (poly_neg l) x

|- !l x c. c * (poly l x) = poly (poly_cmul c l) x

|- !l m x. poly l x + poly m x = poly (poly_add l m) x
```

The reduction theorems have a recurring theme of 'for all polynomials in a finite list' or 'for some polynomial in a finite list'. Accordingly, we make the following general definitions:

```
|- (FORALL P [] = T) /\
   (FORALL P (CONS (h:*) t) = P h /\ FORALL P t)

|- (EXISTS P [] = F) /\
   (EXISTS P (CONS (h:*) t) = P h \/ EXISTS P t)
```

Now we need only the following extra definitions:

```
|- POS x l = poly l x > &0

|- ZERO x l = (poly l x = &0)
```

and we are in a position to state the reduction theorems actually at the HOL object level.

5.2 HOL versions of reduction theorems

The theorem justifying reduction to bounded existential quantifiers is:

```
|- !P. (?x. P x) = (?u. &0 < u /\ u < &1 /\
                        (?x. --(&1) < x /\ x < &1 /\ P(x / u)))
```

The reduction theorem for Type III terms takes the following form:

```
|- !l a b. a < b /\
           (!x. a < x /\ x < b ==> FORALL (POS x) l)
        =  a < b /\
           FORALL (POS (inv(&2) * (a + b))) l /\
           (~?x. a < x /\ x < b /\ EXISTS (ZERO x) l)
```

Note that this one, unlike the other two, demands the extra context of $a < b$ to be included, otherwise the left-to-right implication fails. The reduction theorem for Type II terms is:

```
|- !l a b. (?x. a < x /\ x < b /\ FORALL (POS x) l)
         = (?a' b'. a <= a' /\ a' < b' /\ b' <= b /\
                    (!z. a' < z /\ z < b' ==> FORALL (POS z) l) /\
                    ((a' = a) \/ EXISTS (ZERO a') l) /\
                    ((b' = b) \/ EXISTS (ZERO b') l))
```

And the reduction theorem for Type I terms is the following:

```
|- !l r a b.
      (?x. a < x /\ x < b /\ FORALL(POS x) l /\ (poly r x = &0)) =

      (?x. a < x /\ x < b /\ FORALL(POS x)l /\
           (poly r x = & 0) /\ (poly(poly_diff r)x = & 0)) \/

      (?a' b'. a <= a' /\ a' < b' /\ b' <= b /\
           ((a' = a) \/ EXISTS (ZERO a') (CONS (poly_diff r) l)) /\
           ((b' = b) \/ EXISTS (ZERO b') (CONS (poly_diff r) l)) /\
           ((poly (poly_neg r) a' > &0 /\ poly r b' > &0 /\
              !z. a' < z /\ z < b' ==>
                FORALL (POS z) (CONS (poly_diff r) l)) \/
            (poly r a' > &0 /\ poly (poly_neg r) b' > &0 /\
              !z. a' < z /\ z < b' ==>
                FORALL (POS z) (CONS (poly_neg (poly_diff r)) l))))
```

5.3 Implementation of the algorithm

Although the above theorems have all been proved, and a naive degree-reduction function coded, the system has only been used semi-manually so far. Each step can generate a large number of disjuncts and it is currently intolerably slow for anything beyond one bounded existential quantifier. However we believe there are grounds for optimism that the system can be made more practical by modification of the algorithm and more careful coding (quite apart from the many other interesting algorithms which merit exploring).

Allowing infinite intervals One unfortunate feature of the algorithm as currently implemented is that to eliminate one unbounded existential quantifier, we generate two bounded ones (as well as the hassle of multiplying out the terms to force the intermediate result back into the subset). However it seems to us that the basic algorithm works in a broadly unchanged manner if we allow $a = -\infty$ and/or $b = +\infty$ in the bounds. Here we are speaking informally of infinity – it might be easy enough to formalize an extended real line in HOL, but we are really anticipating splitting each reduction theorem into four separate theorems according as none, either one or both ends of the existential quantifier are bounded.

Note that another advantage of our use of the midpoint in the reduction of Type I terms is that it can easily be extended to give a simple expression for a nominated point of an interval even if it is semi-infinite or infinite. The various cases are:

$$\begin{array}{rcl} -\infty < & 0 & < +\infty \\ -\infty < & b-1 & < b \\ a < & a+1 & < +\infty \\ a < & \frac{a+b}{2} & < b \end{array}$$

The appropriate reduction theorems are still true, mutatis mutandis. For example, to see that the right-to-left implication of the Type I reduction theorem remains true for an interval of the form $(-\infty, +\infty)$, we need to check that a polynomial whose derivative is positive *everywhere* possesses a root *somewhere* (this is untrue for plenty of non-polynomial functions, e.g. e^x).

Actually a stronger result is true: if the derivative of a polynomial is nonzero everywhere then the polynomial has a root. Indeed, every polynomial of odd (actual) degree has a root, so either the antecedent of this conditional statement is trivially false, or the consequent trivially true.

Ordering of disjuncts Many of the terms split into large numbers of disjuncts. The implementation already exploits the 'laziness' assumption that if one disjunct evaluates to truth, the others need not be considered. This is mostly of benefit when the term contains no free variables and hence the result of quantifier elimination is evaluable. However one possibility which has not been exploited is to use straightforward heuristics to decide the order in which the disjuncts are attacked. Clearly it is generally a good idea to start with the 'simple' ones, e.g. those of lowest degree or containing fewer variables.

Treating simple cases separately Certain terms can be eliminated in a simple ad-hoc way without proceeding all the way to degree zero. In fact we could stop at degree 1 since after an elimination step only a single equation remains which can be eliminated very easily by observing:

$$\vdash (\exists x.\ ax + b = 0) = \neg a = 0 \vee b = 0$$

More generally, it might be advantageous to delegate some terms, e.g. linear terms lying within the Presburger subset, to more efficient decision procedures. Work is already in progress on extending the HOL Presburger decision procedure [1] to the reals.

Optimizing the implementation As detailed above, some pains were taken to move the burden of inference onto the production of a few proforma theorems, in order to make the algorithm as far as possible a matter of instantiation and rewriting. However the part of the algorithm doing elimination has been coded rather naively and could probably be speeded up by a moderate factor with careful coding.

6 Conclusion

Though clearly not yet ready for use as a general-purpose tool, we believe this sort of algorithm is a very promising avenue of research. In the future we hope to implement some of the optimizations detailed above in the hope that the procedure can be made much more efficient.

7 Acknowledgements

My thanks go to Konrad Slind, who first pointed out to me the decidability of the elementary theory of reals, and with whom I have had many stimulating discussions on this and other matters. James Davenport explained to me the status of various algorithms and got me started on a literature search. Finally, my supervisor Mike Gordon has never failed to be encouraging and supportive.

References

1. R. J. Boulton, *The HOL arith library*, in HOL System Libraries reference manual, distributed with HOL version 2.01.
2. P. J. Cohen, *Decision Procedures for Real and p-adic Fields*, Communications in Pure and Applied Mathematics vol. 22, pp. 131-151, 1969.
3. G. E. Collins, *Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition*, Second GI Conference on Automata Theory and Formal Languages, Springer Lecture Notes in Computer Science vol. 33, pp. 134-183, 1975.
4. G. Kreisel, J. L. Krivine, *Elements of Mathematical Logic*, 2nd revised printing, North-Holland 1971.
5. A. Seidenberg, *A New Decision Method for Elementary Algebra*, Annals of Mathematics vol. 60, pp. 365-374, 1954.
6. A. Tarski, *A Decision Method for Elementary Algebra and Geometry*, University of California Press 1951.
7. N. J. Vorobjov (Jr.), *Deciding Consistency of Systems of Polynomial in Exponent Inequalities in Subexponential Time*, Proceedings of the MEGA-90 Symposium on Effective Methods in Algebraic Geometry, Progress in Mathematics vol. 94, Birkhäuser 1991.
8. B. L. van der Waerden, *Algebra, vol. I*, 7th edition, Springer-Verlag 1966.

This article was processed using the LaTeX macro package with LLNCS style