THE DECISION METHOD FOR REAL ALGEBRA: IS IT PRACTICAL?

Joel Friedman

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND
MEMORANDUM
RM-4436-PR
JANUARY 1968

THE DECISION METHOD FOR
REAL ALGEBRA: IS IT PRACTICAL?
Joel Friedman

This research is sponsored by the United States Air Force under Project RAND—Con-
tact No. AF 49(638)-1700—monitored by the Directorate of Operational Requirements
and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF.
Views or conclusions contained in this Memorandum should not be interpreted as
representing the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT
Distribution of this document is unlimited.
This Memorandum deals with the practicality of Tarski's decision procedure for the mathematical theory of Real Algebra. Since many important problems and theorems can be stated in the theory, it would be desirable to realize this procedure by writing a computer program which would function as a general problem-solver and theorem-prover for Real Algebra.

However, this Memorandum attempts to show the general impracticality of Tarski's procedure, even when improved by simplifications. Thus it is intended to prevent wasted effort in realizing the procedure and to motivate further research on a more practical decision procedure.
SUMMARY

Tarski's decision procedure is presented in a modified form. It is reduced to three basic methods. Each method processes a basic kind of formula of Real Algebra. The efficiency of each of these methods is discussed. Finally, an alternative to the third method, invented by Hamblin, is presented. Its efficiency is also discussed.

For each of these methods, only a partial measure of efficiency is offered. This partial measure is itself not determined exactly, but rather to within certain bounds. The general impracticality of both the third method and its alternative is then inferred from this partial measure of efficiency. Hence the general impracticality of the whole decision procedure follows.
# CONTENTS

PREFACE ................................................................. iii
SUMMARY ................................................................. v

Section

1. INTRODUCTION ..................................................... 1
2. PRELIMINARIES ..................................................... 4
3. TARSKI'S TDP (WITH MODIFICATIONS) ...................... 7
   3.1. Method A ..................................................
   3.2. Method B ..................................................
   3.3. Method C ..................................................
4. HAMBLIN'S METHOD C' .......................................... 34
5. CONCLUSION ....................................................... 42
6. APPENDIX A ....................................................... 43
7. REFERENCES ....................................................... 44
THE DECISION METHOD FOR REAL ALGEBRA:
IS IT PRACTICAL?

1. INTRODUCTION

Real Algebra is often referred to as Elementary Algebra, but more precisely, it is the theory of real closed fields.* The theory can be axiomatized in first-order logic, though it is not finitely axiomatizable. However, a second-order theory corresponding to this first-order theory is finitely axiomatizable: simply add to the axioms for an ordered field the axiom "Every set of real numbers bounded from above has a least upper bound." The language of set theory is not available for Real Algebra, however, and therefore the latter axiom must be expressed by means of an axiom schema which generates an infinite number of first-order axioms. Such a schema will be given later.

The amazing fact about Real Algebra, which is a rather rich mathematical theory, is that it is decidable. This means that there exists a general method such that given any sentence of Real Algebra, this method determines its truth or falsity in a finite number of steps. Tarski first proved, in [6], the decidability of Real Algebra by constructing such a method. Actually he did much more. Given any formula with free variables (parameters), Tarski's method finds an equivalent formula which has no quantifiers and has at most the

*For a precise definition of "real closed field" see [7], p. 225.
free variables of the original formula. We shall refer to any such method of quantifier elimination for formulas of Real Algebra as a TDP (Tarski Decision Procedure).

Various simplifications of Tarski's TDP have been offered—for example, Collins in [2] and Hamblin in [3]. Also, alternative proofs to Tarski's result have been offered—for example, Seidenberg in [5] and Cohen in [1]. In each of their proofs, a TDP is sketched differing considerably from Tarski's. A. Robinson in [4] offered an alternative proof, although he did not construct a TDP, but rather gave a model-theoretic proof of the decidability of Real Algebra. This is done by showing that Real Algebra is complete, which means that given any formula $F$ of Real Algebra, either $F$ or the negation of $F$ is provable from the axioms. It is easy to see that if a theory is axiomatizable and complete, then it is decidable. For, given any formula $F$, we simply begin generating proofs. After a finite amount of steps, either a proof of $F$ or a proof of the negation of $F$ will be generated since one or the other of these proofs exists by completeness. Furthermore, by the nature of proof, we will be able to decide whether any given proof is a proof of $F$ or its negation.

An important distinction must be made between the simplicity of a proof and the simplicity of a TDP. For example, Robinson's proof is by far the simplest (given that we already know a sufficient amount of model theory).
It does not bother itself with constructing a TDP, which is always rather complicated, since so many cases must be considered corresponding to the different kinds of formulas. Also, Cohen's proof is simpler than Tarski's, though quite possibly the actual TDP sketched by Cohen is as complicated. One measure of the simplicity of a TDP is the degree of complexity required to construct a computer program for it.

Another very important distinction to consider is that between the simplicity of a TDP and its efficiency, which can be measured by the amount of time it takes to process certain standard formulas of Real Algebra. Concerning this distinction, Seidenberg's TDP is a good example of one that is very inefficient though relatively simple. Along with Collins, we believe that Tarski's TDP, or rather, Hamblin's modification of it, is the most efficient currently given in the literature—even more efficient than Cohen's TDP; but this admittedly is speculation. Indeed, more work needs to be done to make Cohen's TDP explicit, and there are indications that it can be considerably improved both as regards simplicity and efficiency.

In this paper, we describe Tarski's TDP and some important modifications of it, and compare its efficiency with that of Hamblin's modification. The comparison will show that Hamblin's is considerably less inefficient, although neither of these TDP's are actually efficient; in fact, both are, by and large, quite impractical.
2. PRELIMINARIES

The language of Real Algebra consists of the following primitive symbols:

A. Logical symbols
   1. variables: a, b, ..., y, z (with or without subscripts).
   2. logical constants: \( \rightarrow, \&, \lor, \sim, E, A \). These are read "implies," "and," "or," "not," "there exists," and "for all," respectively.

B. Algebraic constants
   1. 0, 1, \(-1\), +, \(\cdot\), \(+\), =.

(For convenience, we shall sometimes deviate from this symbolism.)

The terms and formulas of this language may be recursively defined, but here a few examples will be sufficient.

Example 1.

\[(\exists x)(\exists y)(x^2 + y^2 < 1).\]

(There exists a point in the unit circle.)

Example 2.

\[(\forall a_3, a_2, a_1, a_0)[a_3 = 0 \lor (\exists x)(a_3x^3 + a_2x^2 + a_1x + a_0 = 0)].\]

(Every polynomial of degree 3 has a root.)

Example 3.

\[(\exists x)(5x^2 + 4x + 10 = 0 \& 7x^3 + 5x + 3 > 0 \& 9x^2 + 5x + 1 > 0).\]

(No translation necessary.)
The axioms of Real Algebra consist of the axioms for first-order logic, together with the axioms for an ordered field, together with all the axioms generated by the following axiom schema:

\[(Ey)(Ax)(Fx \rightarrow x \leq y) \rightarrow \]

\[(Ey)[(Ax)(Fx \rightarrow x \leq y) \& (Az)((Ax)(Fx \rightarrow x \leq z) \rightarrow y \leq z)]\]

where "F" ranges over every formula of the language. Thus, there are an infinite number of axioms generated.

It should be noted that no TDP uses these axioms directly. Rather the axioms may be used to justify the correctness of a TDP, which as we stated before, uses the method of quantifier elimination. Given a formula, for example, \((Ex)(x^2 + bx + c = 0)\), a TDP processes this formula by finding an equivalent formula which doesn't contain "(Ex)" nor the variable \(x\), but at most the variables \(b\) and \(c\). One such equivalent formula is \(b^2 - 4c \geq 0\). In general, given a formula of the form \((Ex)F(a_1, \ldots, a_n, x)\), where \(F\) is quantifier-free, a TDP will find an equivalent formula \(H(a_1, \ldots, a_n)\) which contains at most the variables \(a_1, \ldots, a_n\). Moreover, \(H\) will not contain any quantifiers. Now given any arbitrary formula of Real Algebra, the process of eliminating its quantifiers can be reduced to iteration of the above process of existential elimination. This can be shown by elementary logical considerations, which will not be dealt
with here. Now, each time a quantifier is eliminated, together with its corresponding variable, no new variables are introduced and at least one variable is lost. Thus, if a formula with no free variables is given, a TDP will find an equivalent formula with no variables whatever but only constants. Consider, for example, "1 < 5 v (3 = 2 & 9 > 7 \cdot 5 + 4)." Each such formula is obviously decidable, and a TDP will output T or F, depending on its truth or falsity.
3. TARKI'S TDP (WITH MODIFICATIONS)

Consider again a formula of the form \((\exists x)F(a_1, \ldots, a_n, x)\), where \(F\) is quantifier-free. It can be shown that the elimination of "\((\exists x)\)" from this formula is reducible to the elimination of "\((\exists x)\)" from formulas of the form:

(a) \((\exists x)P(x) = 0,\)

(b) \((\exists x)(Q_1(x) > 0 \& \ldots \& Q_r(x) > 0),\)

(c) \((\exists x)(P(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0).\)

For each of the forms (a), (b), and (c), a different method of elimination is used. However, the method for (b) reduces to that for (c). The method used for (a) is due essentially to Tarski, but has been simplified by Collins and Hamblin. The reduction of (b) to (c) is again due to Tarski, but a simplification of this reduction will be given.\(^*\) There are two alternative methods for (c); the first is Tarski's, the second is Hamblin's. In this section we will give the first method for (c).

Here it should be noted that Tarski did not actually process formulas of the forms (a), (b), (c). Rather, he processed formulas of the forms \((E_k x)F_1, (E_k x)F_2,\) and \((E_k x)F_3\) (conjoined or disjoined with conditions on the coefficients), where \(F_1, F_2,\) and \(F_3\) correspond to the matrices in the forms (a), (b) and (c). "\((E_k x)\)" should

\(^*\)This simplification is due to Stephen Cook.
be read, "There are exactly $k$ $x$'s such that." Thus, for example, Tarski processed formulas of the form $(\forall_k x) P(x) = 0$ instead of $(\exists x) P(x) = 0$. Nevertheless, it is convenient to adapt his methods to forms (a), (b), and (c). We shall call these methods (in their general outline), Methods A, B, and C, respectively.

Before describing Method A, we need some definitions.

**Def. 1.** $\text{Deg } P = m$

(The degree of the polynomial $P$, where $P = a_m x^m + \ldots + a_0$)

Note that we are treating polynomials as expressions, so that the degree is not affected even when $a_m$ is the constant $0$.

**Def. 2.**

$$Rd(P) = \begin{cases} a_{m-1}x^{m-1} + \ldots + a_0, & \text{if } \text{Deg } P > 0, \\ 0, & \text{if } \text{Deg } P = 0. \end{cases}$$

(The reduction of the polynomial $P$)

**Def. 3.**

$$\text{REM}(P, Q) = \begin{cases} 0, & \text{if } n = 0, \\ P, & \text{if } m < n, \\ Rd(b_n P - a_m Q), & \text{if } m = n > 0, \\ \text{REM}(Rd(b_n P - a_m x^{m-n} Q), Q), & \text{if } m > n > 0. \end{cases}$$

(The integral remainder after division of the polynomial $P$ by the polynomial $Q$, where $\text{Deg } P = m$, $\text{Deg } Q = n$, $P = a_m x^m + \ldots + a_0$, and $Q = b_n x^n + \ldots + b_0$)

**Def. 4.** $P' = ma_m x^{m-1} + \ldots + a_1$

(The formal derivative of the polynomial $P$)
Def. 5.

\[ \pi(n) = \begin{cases} 
0, & \text{if } n \text{ is even}, \\
1, & \text{if } n \text{ is odd}.
\end{cases} \]

(The characteristic function of the set of odd numbers evaluated at the integer \( n \))

Def. 6. \( \Pi \) = the product of the polynomials \( P \) and \( Q \).

3.1. Method A

It is an important fact that there exists a predicate \( G \) (discovered by Tarski but modified by Collins and Hamblin\(^*\)) such that given \( \text{Deg } P = m \), \( \text{Deg } Q = n \), \( a_m > 0 \), and \( c \) is an integer, then, subject to further conditions given below, the following equivalences (1)–(8) hold:

1. \( G^c(P, Q) \equiv \begin{cases} 
    b_n = 0 & \& G^c(P, RdQ) \\
    \bigvee \big[ b_n > 0 & \& G^{c-\pi(m+n)}(Q, -\text{REM}(P, Q)) \\
    b_n < 0 & \& G^{c+\pi(m+n)}(-Q, \text{REM}(P, -Q)) \big] 
\end{cases} \]

 Unless \( mQ \neq 0 \) or \( cQ \neq 0 \), and \( |c| \leq \min(m, n + \pi(m + n)) \).

2. \( G^{>c}(P, Q) \equiv \begin{cases} 
    b_n = 0 & \& G^{>c}(P, RdQ) \\
    \bigvee \big[ b_n > 0 & \& G^{>c-\pi(m+n)}(Q, -\text{REM}(P, Q)) \\
    b_n < 0 & \& G^{>c+\pi(m+n)}(-Q, \text{REM}(P, -Q)) \big] 
\end{cases} \]

 Unless \( Q \neq 0 \) and \( -\min(m, n + \pi(m + n)) \leq c < \min(m, n + \pi(m + n)) \).

\(^*\)We shall use Hamblin's equivalences. Also, see Appendix A.
(3) \( G^0(a, Q) = T, \ G^0(P, 0) = T, \ G^c(P, 0) = F, \) for \( c \neq 0 \)

and \( a > 0; \ G^c(P, Q) = F, \) for \( |c| > \min(m, n + \pi(m + n)) \).

(Here, \( T \) and \( F \) mean \( 0 = 0 \) and \( 0 \neq 0 \), respectively).

(4) \( G^c(a, Q) = F, \) for \( c \geq 0; \ G^c(a, Q) = T, \) for \( c < 0; \)

\( G^c(P, 0) = T, \) for \( c < 0; \)

\( G^c(P, Q) = T, \) for \( c < -\min(m, \pi(m + n)); \)

\( G^c(P, 0) = F, \) for \( c > 0; \)

\( G^c(P, Q) = F, \) for \( c \geq \min(m, \pi(m + n)). \)

(5) \( G^c(P, P') = c = \) the number of roots of \( P. \)

(6) \( G^c(P, P') = c < \) the number of roots of \( P. \)

(7) \( G^c(P, P'Q) = c = \) [the number of roots of \( P \) satisfying \( Q(x) > 0 \)]

- [the number of roots of \( P \) satisfying \( Q(x) < 0 \)]

(8) \( G^c(P, P'Q) \) = (same as (7) except "<" replaces "=").

It follows from (1) and (2), and the fact that \( \text{REM}(-P, Q) = -\text{REM}(P, Q), \) that when the leading coefficients of \( P \) and \( Q \) are nonzero, with \( a_m > 0, \ -m \leq c \leq m, \) and \( \text{Deg} \ Q > \text{Deg} \ P, \) then we have

(i) \( G^c(P, Q) = G^c(P, \text{REM}(Q, P)), \) where \( \text{Deg} \ \text{REM}(Q, P) < \text{Deg} \ P. \)

(ii) \( G^c(P, Q) = G^c(P, \text{REM}(Q, P)), \) where \( \text{Deg} \ \text{REM}(Q, P) < \text{Deg} \ P. \)

Furthermore, given that \( \text{Deg} \ P \geq \text{Deg} \ Q, \) then any descendent \( G \) formula, that is, any \( G \) formula, \( G^d(T, S), \) generated by repeated use of (1) from \( G^c(P, Q), \) will be such that \( \text{Deg} \ T \geq \text{Deg} \ S. \) Thus repeated use of (1) will terminate with \( G \) formulas to which (3) is applicable. Similar remarks hold for \( G^c(P, Q). \) Thus the importance if (1) - (4), (i) and
(ii) is that given a formula containing \( G \) symbols, we can find an equivalent formula not containing \( G \) symbols, in a finite number of steps. The importance of (6) can be seen in the following equivalence:

\[
(9) \quad (\exists x) P(x) = 0 \equiv \left[ a_m = 0 \ & (\exists x) R_d P(x) = 0 \right] \lor \left[ a_m < 0 \ & G^{>0}(P, P') \right] \lor \left[ a_m > 0 \ & G^{>0}(\neg P, \neg P') \right].
\]

Thus, Method A first eliminates the \( G \) symbols from (9), then processes \((\exists x) R_d P(x)\) in the same way. Eventually, we will arrive at \((\exists x) a_0 = 0\), which is equivalent to \( a_0 = 0 \). Thus, given a formula of form (a), we can arrive at an equivalent formula not containing any quantifiers by using (2), (4) and (9), repeatedly. A more refined version of Method A first finds a \( G \)-less formula \( F_1 \) equivalent to \( G^{>0}(P, P') \), using (2) and (4). Then letting \( F_2 \) be the result of replacing each coefficient \( a \) of \( F_1 \) by \(-a\), we can show that \( F_2 \) is a \( G \)-less formula equivalent to \( G^{>0}(\neg P, \neg P') \). Thus to find such a formula, we do not have to use (2) and (4). Another short cut can be used in forming the three disjuncts in (2). Instead of calculating both \( REM(P, Q) \) and \( REM(P, \neg Q) \) on the basis of the definition of \( REM \), we only have to calculate \( REM(P, Q) \) and then note that \( REM(P, \neg Q) = (-1)^{m-n+1} REM(P, Q) \). Thus we can obtain the third \( G \) formula in the right-hand side of (2) without a second use of \( REM \).
3.1.1. The Efficiency of Method A. First, it should be stated that Tarski's method involved a $G$ predicate which generated four new $G$ symbols instead of three. Thus there is some saving here.

Now, one way of measuring the efficiency of Method A is to calculate the number of uses of the remainder operation, REM, required in eliminating "(Ex)." Since REM would take more time to calculate than any other operation in the method, the time required to process formulas of form (a) should be not too large a multiple of the time required by the uses of REM.

Suppose we are given a formula $(\text{Ex})P(x) = 0$. Suppose Deg $P = m$ and all coefficients of $P$ are distinct variables. For each use of (9), we see, by one short cut mentioned above, that we need process only the second $G$ formula by (2). Since the third $G$ formula need not be processed by (2), no uses of REM are required for it. Since Deg $P = m$, there are $m$ such $G$ formulas, namely,

$$G^{>0}(P, P'), \ G^{c_1}(\text{RdP}, (\text{RdP})'), \ldots, \ G^{c_{m-1}}(ax + b, a).$$

For each such $G$ formula, there is a $G$–tree, as Collins has pointed out, whose nodes represent the $G$ symbols as they appear after each use of (2). For each node of the tree, there are three immediately descendent nodes. However, by the other short cut suggested above, we need use REM only once in obtaining these three $G$ formulas. Now the $G$ tree whose first node is $G^{>0}(P, P')$ will have a total
number of nodes equal to $1 + 3 + 3^2 + \ldots + 3^m$. We may disregard the uses of REM for the terminal nodes, since \(\text{REM}(P, a) = 0\). Also the first node does not require REM in its formation. Since only $1/3$ of the remaining nodes require REM, the total number of uses of REM required to eliminate $G > 0(P, P')$ is

$$
\frac{1}{3}(3 + 3^2 + \ldots + 3^{m-1}) = \frac{1}{2}(3^{m-1} - 1) \approx \frac{1}{2}(3^{m-1}).
$$

Since there are $m$ G formulas on our list, each involving polynomials whose degrees go from $m$ to 1, the total number of uses of REM required to process all the G formulas is approximately

$$
\frac{1}{2}(3^{m-1}) + \frac{1}{2}(3^{m-2}) + \ldots + \frac{1}{2} = \frac{1}{2}(1 + 3 + \ldots + 3^{m-1}) \approx \frac{1}{4}(3^m).
$$

Thus our approximate number of uses of REM required by Method A, applied to formulas of form (a) with variable coefficients, is $1/4(3^m)$.

If we consider the case in which all the coefficients are constant, then we see, looking at (9), that only one of the disjuncts needs consideration, since we can evaluate $a_m$ and eliminate the other disjuncts. Thus, at most one G formula needs to be eliminated, and the worst case would be $G > 0(P, P')$, with $a_m \neq 0$. Looking at (2), we see that again we need consider only one disjunct since $b_n$ can be evaluated. (In this case $b_n \neq 0$). Thus, this disjunct contains one G formula whose formation requires one use of REM. If REM($P, P'$) = 0, we are
through; otherwise, we repeat the process. Hence, there are at most \( m - 1 \) nontrivial uses of REM. Thus \( m - 1 \) is the number of uses of REM required to eliminate any arbitrary sentence of form (a), where \( \text{Deg } P = m \). To be more precise, \( m - 1 \) is at most the number of uses. Some sentences of this form will require less uses of REM. If we compare \( m - 1 \) with \( 1/4(3^m) \), we see the great difference it makes when we can evaluate the coefficients of \( P \). Also, the number of uses of REM required when the coefficients are mixed is a number \( k \) such that \( m - 1 \leq k \leq 1/4(3^m) \).

3.2. Method B

This basic method is for formulas of the form (b), that is, \((\text{Ex})(Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)\). Suppose that \( \text{Deg } Q_i = n_i \) and that the leading coefficient of \( Q_i \) is \( q_{i,n_i} \neq 0 \), for \( 1 \leq i \leq r \). Then the following equivalence, due essentially to Tarski, holds:

\[
(10) \quad (\text{Ex})(Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)
\]

\[= [q_{1,n_1} > 0 \& \ldots \& q_{r,n_r} > 0] \]

\[v[(-1)^{n_1}q_{1,n_1} > 0 \& \ldots \& (-1)^{n_r}q_{r,n_r} > 0] \]

\[v(\text{Ex})[(Q_1 \ldots Q_r)'(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0] \].

This is proved by considering the open intervals on which all the \( Q_i \)'s are positive and then showing that the product
of the $Q_i$'s attains a maximum or minimum, given that the other conditions on the coefficients fail. Here we see that formulas of form (b) reduce to form (c). However (10) is not quite enough. For a given formula of form (b), whose coefficients are distinct variables, we have to consider all the cases in which the leading coefficients are zero, the next coefficients are zero, etc. More precisely, given $Q_i$, any coefficient $q_{i,j}$ may serve as a leading nonzero coefficient, with $q_{i,k} = 0$, for $j < k \leq n_i$. The number of possible cases is thus $n = n_1 \ldots n_r$. Now for each case, a formula similar to (10) holds. Thus, for all the cases, there will be generated $n$ new existential formulas of form (c), each disjuncted with the appropriate conditions on the coefficients. Tarski proceeded by this method of generating $n$ such formulas. However, the following simplification was found by Stephen Cook and represents an enormous saving, since only one existential formula of form (c) is generated. Its proof again is similar to the proof of (10), except that cases involving the vanishing of leading coefficients have to be considered.
(11) \((\text{Ex}) (Q_1(x) > 0 \ & ... \ & Q_r(x) > 0)\)

\[= (q_1,n_1 \neq 0 \ & ... \ & q_r,n_r \neq 0 \ & \left[ [q_1,n_1 > 0 \ & ... \ & q_r,n_r > 0 \right] \]

\[
v[(-1)^{n_1}q_1,n_1 > 0 \ & ... \ & (-1)^{n_r}q_r,n_r > 0])]
\]

\[v ... v(q_{i,j} = 0 \ & \left[ [q_{1,0} > 0 \ & ... \ & q_{r,0} > 0 \right] \]

\[v[(-1)^{0}q_{1,0} > 0 \ & ... \ & (-1)^{0}q_{r,0} > 0])]
\]

\[v(\text{Ex}) [(Q_1 \ ... \ Q_r)'(x) = 0 \ & Q_1(x) > 0 \ & ... \ & Q_r(x) > 0],\]

where \(j \geq 1\) and \(i \geq 1\). Note that the dots between "\(v ... v\)" in (11) represent the remaining \((n_1 \ ... \ n_r) - 2\) cases, in each of which a given coefficient serves as a leading nonzero coefficient and all its preceding coefficients are zero.

The only disadvantage with the new existential formula generated by (11) is the fact that any one of the coefficients of \((Q_1 \ ... \ Q_r)'\) may be zero. Now, \(\text{Deg } (Q_1 \ ... \ Q_r)' = n_1 + ... + n_r - 1\). Thus, in processing this formula of form (c), there will be generated \(n_1 + ... + n_r - 1\) new existential formulas, also of form (c), with the degrees of their first polynomials going from \(n_1 + ... + n_r - 2\) to 1. But this is surely better than the \(n_1 \ ... \ n_r\) new existential formulas of form (c), generated by Tarski's equivalence, even though for each such formula we know that the leading coefficient is nonzero.
3.2.1. The Efficiency of Method B. In spite of this enormous saving, (11) shows us an essential difficulty with Method B. Namely, suppose we started off with the formula \((E_y)(E_x)(Q_1(x) > 0 \& \cdots \& Q_r(x) > 0)\). Then after eliminating "(Ex)" via (11) and for purpose of simplification, putting the result in minimal disjunctive normal form, we would have to distribute "(Ey)" among all the disjuncts. Thus we would generate as many new problems of eliminating "(Ey)" as there were disjuncts, and that number would be 
\[
1/k(2(n_1 \cdots n_r) + 1),
\]
where \(k > 0\) is not too large an integer. Note that one of \((Ey)\)-formulas generated would be the formula

\[(E_y)(E_x)((Q_1 \cdots Q_r)'(x) = 0 \& Q_1(x) > 0 \& \cdots \& Q_r(x) > 0).\]

Hence, a repetition of the same trouble. It seems that we cannot avoid this situation in principle, since we can never get rid of zero.

It should be noted that Method B requires no uses of REM, so we cannot measure its efficiency by counting such uses. Rather we may take \((2(n_1 \cdots n_r) + 1)t\) as a measure of efficiency, where \(t\) is the average time it takes to form each disjunct of the right-hand side of (11).

3.3. Method C

This final basic method of Tarski's TDP is for formulas of form (c). Here, however, we must revert back to Tarski's method of processing formulas of form \((E_kx)F_3\).
We note that (c) is equivalent to

\[ \neg (E_0 x) [P(x) = 0 & Q_1(x) > 0 & \ldots & Q_r(x) > 0]. \]

Thus to process formulas of form (c), it is sufficient to process \((E_0 x) F_3\) and then negate the result. Following Tarski closely, we give the following equivalence:

(12) \((E_0 x) [P(x) = 0 & Q_1(x) > 0 & \ldots & Q_r(x) > 0] \equiv (a_m = 0 & (E_0 x) [RdP(x) = 0 & Q_1(x) > 0 & \ldots & Q_r(x) > 0] \lor \sqrt{0 \leq r_1, r_2, r_3 \leq m, r_1 + r_2 - r_3 = 0,}

\[ (a_m \neq 0 & \]

\((E_{r_1} x) [P(x) = 0 & Q_1(x) > 0 & \ldots & Q_{r-2}(x) > 0 & Q_{r-1}^2(x) > 0]) \]

& \((a_m \neq 0 & \]

\((E_{r_2} x) [P(x) = 0 & Q_1(x) > 0 & \ldots & Q_{r-2}(x) > 0 & Q_{r-1}^2(x) > 0]) \]

& \((a_m \neq 0 & \]

\((E_{r_3} x) [P(x) = 0 & Q_1(x) > 0 & \ldots & Q_{r-2}(x) > 0 & Q_{r-1}^2(x) > 0]) \)

, where \(r \geq 2\), and "\(\lor\)" is a metamathematical symbol, not part of the language of Real Algebra, but rather a shorthand device to save us from writing out a great many disjunctions.

We may disregard for the present the first disjunct, in which \(a_m = 0\) and whose existential subformula is again
of the form \((E_0 x)F_3\). We must now give an equivalence similar and complementary to (12):

\[
(13) \quad a_m \neq 0 \& (E x) \left[ \bigwedge_{k} \left[(P(x) = 0 \& \bigwedge_{j \leq k} Q_j(x) > 0) \right] \right]
\]

\[
= \sqrt{0 \leq r_1, r_2, r_3 \leq m, r_1 + r_2 - r_3 = 2k,}
\]

\[
\left[ (a_m \neq 0 \&
\right. \]

\[
(E_{r_1} x) \left[ (P(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_{r-2}(x) > 0 \& Q_{r-1}^2(x) > 0) \right]
\]

\[
\left. & (a_m \neq 0 \&
\right. \]

\[
(E_{r_2} x) \left[ (P(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_{r-2}(x) > 0 \& Q_{r-1}^2(x) > 0) \right]
\]

\[
\left. & (a_m \neq 0 \&
\right. \]

\[
(E_{r_3} x) \left[ (P(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_{r-2}(x) > 0 \& -Q_{r-1}Q_r(x) > 0) \right]
\]

, where \(r \geq 2\).

Now (12) shows us that for those subformulas of the form \(E_{r_1} x)H_1\), \(H_1\) contains one less inequality than the original formula of the form \((E_0 x)F_3\). And (13) shows us that for those subformulas of the form \(E_{r_1} x)H_1\), \(H_1\) contains one less inequality than the original formula of the form \((E_k x)H\). Thus for one use of (12) and repeated use of (13), we will ultimately arrive at formulas of the form \((E_k x)[P(x) = 0 \& Q(x) > 0]\). The following equivalence shows us how to eliminate these.
\[(14) \quad (E_k x)(P(x) = 0 \& Q(x) > 0)\]

\[= (a_m = 0 \& (E_k x)[RdP(x) = 0 \& Q(x) > 0])\]

\[\nu(a_m > 0 \& \sqrt{0 \leq r_1, r_2 \leq m, -m \leq r_3 \leq m, r_1 - r_2 + r_3 = 2k,}
\[\left[G^1_r(P, P') \& G^2(P^2 + Q^2, (P^2 + Q^2)'), \& G^3(P, P'Q)\right])\]

\[\nu(a_m < 0 \& \sqrt{0 \leq r_1, r_2 \leq m, -m \leq r_3 \leq m, r_1 - r_2 + r_3 = 2k,}
\[\left[G^1_r(-P, -P') \& G^2(P^2 + Q^2, (P^2 + Q^2)'), \& G^3(-P, -P'Q)\right])\]

\[(14) \text{ is proved with the help of (7). Here we see that formulas of the form (}E_k x)[P(x) = 0 \& Q(x) > 0]\text{ reduce to } G \text{ formulas, which we already know how to eliminate. Thus, with Method C, we first put formulas of form (c) into the form } \neg(E_0 x)F_3. \text{ We then use (12) on } (E_0 x)F_3, \text{ if } r \geq 2. \text{ We then use (13) repeatedly on formulas of the form } a_m \neq 0 \& (E_f x)H. \text{ And then we use (14) on formulas of the form } (E_k x)(P(x) = 0 \& Q(x) > 0), \text{ disregarding the disjunct in which } a_m = 0. \text{ Finally, we eliminate the } G \text{ formulas by (1) and (3), using a short cut on the third disjunct of (14) analogous to that described for Method A. When } r = 1, \text{ we skip (12) and (13) and go directly to (14) applying it to } (E_0 x)(P(x) = 0 \& Q_1(x) > 0). \text{ For the disjunct in which } a_m = 0, \text{ we use (14) again on } (E_0 x)(RdP(x) = 0 \& Q_1(x) > 0). \text{ By this method we will ultimately arrive at } (E_0 x)(a_0 = 0 \& Q_1(x) > 0), \text{ which is equivalent to } a_0 \neq 0 \text{ v } \neg(Ex)(Q_1(x) > 0). \text{ Thus we have} \]
arrived at a formula of form (b). Returning to the case in which \( r \geq 2 \), the first use of (12) still leaves us with a formula of the form \((E_0x)F_3\). We again use (12) on this formula and repeat the above procedure. After repeated use of (12) and the above procedure, we are finally left with a formula of the form

\[ (E_0x)[a_0 = 0 \land Q_1(x) > 0 \land \ldots \land Q_r(x) > 0]. \]

This is equivalent to

\[ a_0 \neq 0 \lor (\exists x)[Q_1(x) > 0 \land \ldots \land Q_r(x) > 0]. \]

Thus, again we are left with a formula of form (b). But as we saw before, (b) reduces to a formula of form (c), more explicitly, to

\[ (\exists x)[(Q_1 \ldots Q_r)'(x) = 0 \land Q_1(x) > 0 \land \ldots \land Q_r(x) > 0]. \]

Now we seem to have gone in a circle: First we reduced formulas of form (b) to those of form (c), and now we have come back to (b). But the circle is really more like a spiral. For the formula of form (c) which we get from (b) is of a very special kind, and if we begin processing this (c)-formula by the above procedure starting with (12), (or (14)), we see that in considering cases of the coefficients being zero, the following fact saves us from circularity. When

\[ b_j = b_{j-1} = \ldots = b_1 = b_0 = 0, \quad \text{where} \quad \text{Deg } (Q_1 \ldots Q_r)' = j = n_1 + \ldots + n_r - 1, \quad \text{and where} \quad \text{the } b's \text{ are the coefficients of } (Q_1 \ldots Q_r)', \]

it follows that for each \( Q_j \), we have
\[ q_{i,n_i} = q_{i,n_i-1} = \ldots = q_{i,1} = 0. \] Hence \( Q_\perp = q_{i,0} \). Thus, though we start out with

\[
(\exists x)[(Q_1 \ldots Q_r)'(x) = 0 \land Q_1(x) > 0 \land \ldots \land Q_r(x) > 0]
\]

and arrive at \( (\exists x)[Q_1(x) > 0 \land \ldots \land Q_r(x) > 0] \) by Method C, we find that for this case, the formula is equivalent to \((q_{1,0} > 0 \land \ldots \land q_{r,0} > 0)\). "(Ex)" is thus vacuous for this case. Thus, Method C may be summarized as follows: We start off with a formula of form (c), go back to (b), use Method B to come back again to (c), and then use Method C to go back again to vacuous (b). Hence we spiral rather than circle. Thus we see that starting with any formula of form (c), we can find, in a finite number of steps, an equivalent formula without quantifiers by the repeated use of (1), (3), (11), (12), (13) and (14).

### 3.3.1. The Efficiency of Method C

We shall use as an indication of efficiency the total number of uses of REM required by Method C to eliminate "(Ex)" from an arbitrary formula of form (c). However, it should be noted that this number does not really measure the efficiency, but only a fraction of it. Consider the number of instances of formulas of the form \((E_k x)H\) generated by (12) and (13) when applied to a formula of the form

\[
(E_0 x)(P(x) = 0 \land Q_1(x) > 0 \land \ldots \land Q_r(x) > 0), \text{ for } r \geq 2.
\]
Now the number of such distinct formulas is considerably less than the total number of instances of such formulas. Many instances of the same formula must occur. As a special case let us consider the number of instances of formulas of the form \((E_0^x)H\) generated as described above. First, consider the number of \(r_1\)'s, where \(0 \leq r_1 \leq m\), such that there are \(0 \leq r_2, r_3 \leq m\) such that \(r_1 + r_2 - r_3 = 0\). There are \(m + 1\) such \(r_1\)'s. Similarly, there are \((m + 1)\) \(r_2\)'s and \((m + 1)\) \(r_3\)'s. Thus, by the first use of (12), 3\((m + 1)\) instances of formulas of the form \((E_0^x)H\) are generated, though there are only 3 such distinct formulas. Furthermore, any one of these instances will generate, by the use of (13), another 3\((m + 1)\) instances of formulas of the form \((E_0^x)H\), and so forth, until we reach the formulas of the form \((E_0^x)(R(x) = 0 \& Q(x) > 0)\). Moreover, we have yet to consider the other formulas \((E_k^x)H\), where \(k \neq 0\), most of which also generate formulas of the form \((E_0^x)H\).

Now one strategy for improving this situation is the following. For a given formula \(F_1\) which is a subformula of \(F_2\), we would find a simplified quantifier-free formula equivalent to \(F_1\) (using Method C, of course) and then plug this simplified formula into the places where all the other instances of \(F_1\) occur in \(F_2\). Thus we would process by Method C only one instance of \(F_1\) and copy the result for every other instance of \(F_1\). This would save us from duplication. Now supposing this could be accomplished—and
it would be no small programming feat—still, the operation of plugging a formula into a larger formula takes time, however sophisticated our encoding of formulas into the memory of a computer may be. And there would be a very large number of uses of such an operation. And before we could plug in, we would need an equality routine to determine whether two instances of a formula were equal. However, we shall not deal with the problem of estimating the total number of uses of the plugging-in operation and the equality routine required by Method C when it is applied to a formula of form (c). Suffice it to say that this number should also figure in measuring the efficiency of Method C.

We shall now estimate the number of uses of REM required by Method C. Suppose we are given a formula of the form 
\((E_0 x)(P(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)\), for \(r \geq 2\), all of whose coefficients are distinct variables. Suppose \(\text{Deg } P = m\) and \(\text{Deg } Q_i = n_i\). We shall first estimate the number of uses of REM required by one use of (12) on the above formula and repeated use of (13), (14), and (1). Call this number \(\text{NR}_{(m,r)}\). To do this, we must calculate the number of distinct matrices of the form \(P(x) = 0 \& Q(x) > 0\) generated by one use of (12) and repeated use of (13). One use of (12) generates 3 distinct, first-generation matrices. The next use of (13) generates \(3^2\) distinct, second-generation matrices. By induction on \(r\) it can easily be seen that the \(r - 1\)'th generation consists of
3^{r-1} distinct matrices, of the form P(x) = 0 & Q(x) > 0. P is fixed for all these matrices. It is Q which varies. Now we shall calculate the number of distinct G formulas generated by these matrices. Here we have a strong notion of distinct. We shall not regard G^0(P, Q) as distinct from G^d(P, Q), even when 0 ≠ d. The reason for this is that the uses of REM required to process G^d(P, Q) are among the uses of REM required to process G^0(P, Q), although the G-less formula equivalent to G^0(P, Q) will not, in general, be equivalent to the G-less formula equivalent to G^d(P, Q). However, we shall presuppose that we have a method for keeping track of distinct terms of the form REM(P, Q) so that after we have processed such terms once, we may plug in the result, when required, in processing other G formulas. Hence, we are trying to estimate the number of distinct uses of REM. It should be noted that when d = m, the number of uses of REM required to process G^d(P, Q) will be considerably less than that for G^0(P, Q). The reason for this is that G^m(P, Q) will generate G formulas G^{m+c}(S, T), many of which require no uses of REM, since by (3), they are equivalent to falsity. But this situation never occurs for G^0(P, Q), since the G formulas, G^d(S, T), which it generates is such that −m ≤ d ≤ m and Deg S ≥ Deg T. Thus the distinct uses of REM can be found from the G formulas generated by G^0(P, Q).

Since there are 3^{r-1} distinct matrices, there are
3^{r-1} distinct G formulas of the form \( G^0(p^2 + q^2, p^2 + q^2) \) and 3^{r-1} of the form \( G^0(p, p'q) \). There is only one distinct G formula of the form \( G^0(p, p') \), since \( p \) is fixed for these matrices. Thus we may ignore this G formula. It will not add significantly to the number of uses of REM, even when \( r \) is small. We may also ignore those G formulas of the form \( G^0(-p, -p') \) and \( G^0(-p, -p'q) \) generated by (14), since the third disjunct in the right hand side of (14) will be processed by the short-cut mentioned above, and thus no uses of REM will be required here.

Now the uses of REM required to process formulas of the form \( G^0(p, p'q) \) is (as was estimated before) approximately \((1/2)3^{m-1}\). Since there are \( 3^{r-1} \) such distinct formulas, the number of uses of REM required for them is \((1/2)3^{(m+r-2)}\).

The matter is not so simple for formulas of the form \( G^0(p^2 + q^2, (p^2 + q^2)') \), since deg Q varies as Q varies.

Now, the lowest degree Q may attain is \( n_1 + \ldots + n_r \).

Here \( Q = Q_1 \ldots Q_r \). Thus the lowest degree which \( p^2 + q^2 \) may attain is \( 2m + 2(n_1 + \ldots + n_r) \). An upper bound (but by no means close to the least upper bound) for the highest degree Q may attain is \( 2^{r-1}(n_1 + \ldots + n_r) \).

This may be proved by induction on \( r \) and casual glances at (12) and (13). Thus an upper bound for the highest degree which \( p^2 + q^2 \) may attain is \( 2m + 2^r(n_1 + \ldots + n_r) \).

Thus, the number of uses of REM required to process a given formula of the form \( G^0(p^2 + q^2, (p^2 + q^2)') \) is a
number \( k \) such that
\[
\frac{1}{2^3} (2m+2(n_1+\ldots+n_r) - 1) \leq k \leq \frac{1}{2^3} (2m+2^r(n_1+\ldots+n_r) - 1).
\]

Since there are \( 3^{r-1} \) such distinct formulas, the number of such uses of REM is a number \( k \) such that
\[
\frac{1}{2^3} (2m+2(n_1+\ldots+n_r) + r-2) \leq k \leq \frac{1}{2^3} (2m+2^r(n_1+\ldots+n_r) + r-2).
\]

Thus the total number of distinct uses of REM required to process all the distinct \( G \) formulas generated by one use of (12) and repeated use of (13) and (14) is \( NR_{(m,r)} \), where
\[
\frac{1}{2^3} (m+r-2) + \frac{1}{2^3} (2m+2(n_1+\ldots+n_r) + r-2) \leq NR_{(m,r)} \leq \frac{1}{2^3} (m+r-2) + \frac{1}{2^3} (2m+2^r(n_1+\ldots+n_r) + r-2).
\]

But we are far from finished. We must now estimate the uses of REM required by repeated use of (12) and the above procedure. Now the first use of (12) generates the formula \((E_0 x)(RdP(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)\). If we again apply (12) to this formula and then use the above procedure, the number of uses of REM required will be \( NR_{(m-1,r)} \), because \( m \) was arbitrary in the above calculations, and here our formula is the same as the previous \( E_0 \)-formula except that the degree of the first polynomial is \( m-1 \) instead of \( m \). Similarly, repeated use of (12) and the above procedure will yield \( NR_{(m-2,r)}, \ldots, NR_{(1,r)} \). Thus the total number of uses of REM required to process \((E_0 x)(P(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)\) by repeated
use of (12), until \( m = 0 \), is the number \( \text{TNR(m,r)} = \sum_{i=1}^{m} \text{NR}(i, r) \). Hence after a little calculation, we have

\[
\frac{1}{4} \cdot 3^{r-1} (3^m - 1) + \frac{1}{16} \cdot 3^{2r(1 + \ldots + n_r) + r} (2 \cdot 3^{2m} - 1) \leq \text{TNR(m,r)}
\]

\[
\leq \frac{1}{4} \cdot 3^{r-1} (3^m - 1) + \frac{1}{16} \cdot 3^{2r(1 + \ldots + n_r) + r} (3^{2m} - 1).
\]

Now when \( m = 0 \), the resulting \( E_0 \)-formula is

\[(E_0 x)(a_0 = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0),\]

which, as we saw before, is equivalent to

\[a_0 \neq 0 \lor \sim(\exists x)(Q_1(x) > 0 \& \ldots \& Q_r(x) > 0).\]

Thus, we have arrived at a formula of form (b). But Method B gives us back the formula

\[(E_0 x)((Q_1 \ldots Q_r)'(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0),\]

which must again be processed by Method C. Since,

\[\text{Deg } (Q_1 \ldots Q_r)' = n_1 + \ldots + n_r - 1,\]

we have that

\[\text{TNR}(n_1 + \ldots + n_r - 1, r) = \text{the total number of distinct uses of } \text{REM} \text{ required by repeated use of (12) and the above procedure until we arrive at }\]

\[(E_0 x)(b_0 = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0),\]

which is fortunately equivalent to

\[b_0 \neq 0 \lor (q_1,0 > 0 \& \ldots \& q_r,0 > 0).\]

Method C terminates here. Thus the total number of distinct uses of \( \text{REM} \) required by Method C on a formula of form (c) is
\[ WNR_{(m,n_1,\ldots,n_r,r)} = TNR_{(m,r)} + TNR_{(n_1^+\ldots+n_r-1,r)}. \]

After a little calculation, we have

\[ WNR_{(m,n_1,\ldots,n_r,r)} \geq \left( \frac{1}{4} \right)^3 \frac{(n_1^+\ldots+n_r-1)}{3^m + 3} - 2 \]

\[ + \left( \frac{1}{16} \right)^3 \frac{(n_1^+\ldots+n_r-r)}{3^{2m} + 3} - 2, \]

and

\[ WNR_{(m,n_1,\ldots,n_r,r)} \leq \left( \frac{1}{4} \right)^3 \frac{(n_1^+\ldots+n_r-1)}{3^m + 3} - 2 \]

\[ + \left( \frac{1}{16} \right)^3 \frac{(n_1^+\ldots+n_r-r)}{3^{2m} + 3} - 2. \]

Let us mention again that \( r \geq 2 \) here.

Noting that WNR measures only a fraction of the efficiency of Method C, it is obvious that Method C is quite impractical for almost any interesting formula of form (c), with variable coefficients. For concreteness, let us give several examples of formulas of form (c), and then calculate the bounds on WNR for these formulas. WNR may be regarded as a function of the variables \( m, n_1, \ldots, n_r, \) and \( r. \) We shall see what happens to WNR for small and for slightly large values of these variables.

**Example 1.**

\((\text{Ex}) (a_2 x^2 + a_1 x + a_0 = 0 \land b_2 x^2 + b_1 x + b_0 > 0 \land c_2 x^2 + c_1 x + c_0 > 0).\)

Thus, \( m = 2, n_1 = 2 \ (1 \leq i \leq 2), \) and \( r = 2. \) We have
\[
\left(\frac{1}{4}\right) 3 (3^2 + 3^3 - 2) + \left(\frac{1}{16}\right) 3^{10} (3^4 + 3^6 - 2)
\leq \text{WNR} \leq \left(\frac{1}{4}\right) 3 (3^2 + 3^3 - 2) + \left(\frac{1}{16}\right) 3^{18} (3^4 + 3^6 - 2).
\]

After a little calculation, we obtain \(10^6 < \text{WNR} < 10^{11}\).

Thus, between a million and 100 billion uses of REM would be required for this modest formula of Real Algebra. This is within the scope of present-day computers, provided that the average use of REM requires no more than one second. This is another problem in itself.

**Example 2.**

\[
(\text{Ex}) \left( a_{10} x^{10} + \ldots + a_0 = 0 \right) \& \left( b_2 x^2 + b_1 x + b_0 > 0 \right) \& \left( c_2 x^2 + c_1 x + c_0 > 0 \right).
\]

Here, \(m = 10\), \(n_i = 2\) \((1 \leq i \leq 2)\), and \(r = 2\). We then obtain

\[
10^{13} < \frac{1}{16} 3^{30} < \text{WNR} < \frac{1}{8} 3^{38} < 10^{18}.
\]

Here we are passing over the border of practicality.

**Example 3.**

\[
(\text{Ex}) \left( a_{10} x^2 + a_1 x + a_0 = 0 \right) \& \left( b_{10} x^{10} + \ldots + b_0 > 0 \right) \& \left( c_{10} x^{10} + \ldots + c_0 > 0 \right).
\]

Here, \(m = 2\), \(n_i = 10\) \((1 \leq i \leq 2)\), and \(r = 2\). Then

\[
10^{36} < \frac{1}{16} 3^{80} < \text{WNR} < \frac{1}{8} 3^{120} < 10^{58}.
\]
Here we have passed beyond the border of practicality. For even if the average use of REM required only one microsecond, which is asking too much, the time required would be between $10^{-30}$ seconds and $10^{52}$ seconds, or roughly, between $10^{22}$ and $10^{45}$ years. Note that though the degree of the first polynomial is small, WNR is still very large. One reason for this is that the formula in Ex. 3 will reduce to a formula of form (b), which again will reduce to

$$\text{(Ex)} \left( b_{10}x^{10} + \ldots + b_0 \right) \left( c_{10}x^{10} + \ldots + c_0 \right) = 0$$

$$\& \ b_{10}x^{10} + \ldots + b_0 > 0 \ & \ c_{10}x^{10} + \ldots + c_0 > 0$$

where the degree of the first polynomial is now 19.

**Example 4.**

$$\text{(Ex)} \left( a_2x^2 + a_1x + a_0 = 0 \ & \ b_{1,10}x^{10} + \ldots + b_{1,0} > 0$$

$$\& \ \ldots \ & \ b_{10,10}x^{10} + \ldots + b_{10,0} > 0$$

Here, $m = 2, \ n_1 = 10 \ (1 \leq i \leq 10)$, and $r = 10$. Then

$$\text{WNR} > \frac{1}{16^3}408 > 10^{193}$$

We may pass on to the next example without comment.

**Example 5.**

$$\text{(Ex)} \left( a_2x^2 + a_1x + a_0 = 0 \ & \ b_{1,2}x^2 + b_{1,1}x + b_{1,0} > 0$$

$$\& \ \ldots \ & \ b_{10,2}x^2 + b_{10,1}x + b_{10,0} > 0$$

Here, \( m = 2, n_i = 2 \ (1 \leq i \leq 10) \), and \( r = 10 \). Then

\[
\text{WNR} > \frac{1}{10^3} \cdot 88 > 10^{40}.
\]

Again this is beyond practicality. This is an interesting example, because the degree of each polynomial involved is only 2. However, there are just too many such polynomials. This situation may occur quite frequently when TDP deals with formulas of the form \((Ey)(Ex)H\).

**Example 6.**

\[(Ex)(a_{10}x^{10} + \ldots + a_0 = 0 \ & b_{1,2}x^2 + b_{1,1}x + b_{1,0} > 0
\& \ldots \ & b_{10,2}x^2 + b_{10,1}x + b_{10,0} > 0).\]

Here, \( m = 10, n_i = 2 \ (1 \leq i \leq 10) \), and \( r = 10 \). Then

\[
\text{WNR} > \frac{1}{16^3} \cdot 88 > 10^{40}.
\]

This is similar to Ex. 5, except \( m = 10 \). The slightly large value for \( m \) does not affect WNR very much in this case.

**Example 7.**

\[(Ex)(a_{10}x^{10} + \ldots + a_0 = 0 \ & b_{10}x^{10} + \ldots + b_0 > 0
\& \ldots \ & c_{10}x^{10} + \ldots + c_0 > 0).\]

Here, \( m = 10, n_i = 10 \ (1 \leq i \leq 2) \), and \( r = 2 \). Then

\[
\text{WNR} > \frac{1}{16^3} \cdot 80 > 10^{36}.
\]
This is similar to Ex. 3, except that $m = 10$. Here $m$ does affect considerably the value of WNR.

**Example 8.**

$$(a_{10}x^{10} + \ldots + a_0 = 0 \& b_{1,10}x^{10} + \ldots + b_{1,0} > 0$$

$$\& \ldots \& b_{10,10}x^{10} + \ldots + b_{10,0} > 0).$$

Here, $m = 10$, $n_1 = 10$ ($1 \leq i \leq 10$), and $r = 10$. Then

$$\text{WNR} > \frac{10^{408}}{8^{3.4}} > 10^{193}.$$  

This is similar to Ex. 4, except $m = 10$, but again $m$ has very little affect in this case.

The above examples show clearly the general impracticability of Method C. However, formulas such as those in Examples 1 and 2 do have a chance. Moreover, if a multiply-quantified formula of Real Algebra is such that it generates a not too large number of formulas such as in Examples 1 and 2, then it should have a chance also. But it might be difficult to find formulas such as these, which are, at the same time, interesting.

Although we shall not attempt to do so here, it would be interesting to estimate WNR for formulas of form (c), where all the coefficients are constant. Naturally we would expect the values of WNR for these formulas to be considerably less than the values of WNR for the other formulas.
4. HAMBLIN'S METHOD C'

We shall now consider Hamblin's method for processing formulas of form (c). (See [3]). We shall be anxious to know whether it is more efficient than Method C, and, if so, whether it is practical.

We first introduce Hamblin's s function.

Def. 7. \[
\begin{align*}
\text{s}(P; Q_1, \ldots, Q_r; Q) &= \frac{1}{2} [\text{s}(P; Q_1, \ldots, Q_{r-1}; Q_r Q) \\
&\quad + \text{s}(P; Q_1, \ldots, Q_{r-1}; Q_r Q)] \\
&= \text{[the number of roots of } P \text{ satisfying } Q_i(x) > 0 \text{ and } Q(x) > 0] \\
&\quad - \text{[the number of roots of } P \text{ satisfying } Q_i(x) > 0 \text{ and } Q(x) < 0], \text{ where } 1 \leq i \leq r, \text{ and } r \geq 1. \\
\end{align*}
\]

(The supertangle of the polynomials P and Q with respect to the polynomials Q_1, \ldots, Q_r.)

We shall abbreviate \text{s}(P; Q_1, \ldots, Q_0; Q) to \text{s}(P; Q).

Now, according to Hamblin, the following equalities hold for s.

\begin{align*}
(15) \quad &\text{s}(P; Q_1, \ldots, Q_r; Q) = \frac{1}{2} [\text{s}(P; Q_1, \ldots, Q_{r-1}; Q_r Q) \\
&\quad + \text{s}(P; Q_1, \ldots, Q_{r-1}; Q_r Q)] \\
(16) \quad &\text{s}(P; Q_1, \ldots, Q_r; Q) = \text{s}(P; Q_1, \ldots, Q_r; \text{REM}(Q, P)) \\
(17) \quad &\text{s}(P; Q) = \text{the unique } c \text{ such that } G_c(P, P'Q), \text{ if } a_m > 0.
\end{align*}
Clearly (17) follows from (7). Note that \( s(P; Q_1, \ldots, Q_r; Q) \) ranges from \(-m\) to \(m\), where \( \text{Deg } P = m \). We next introduce one of Hamblin's \( S \) predicates.*

**Def. 8.**

a) \( S^c_t(P; Q_1, \ldots, Q_r; Q) = s(P; Q_1, \ldots, Q_r; Q) > c \) &
\( t = \) the number of roots of \( P \) satisfying
\( (Q_1 \ldots Q_r)(x) \neq 0, \) where \( r \geq 1 \).

b) \( S^c_t(P; Q_1, \ldots, Q_r; Q) = a_m > 0 \) & \( G^c(P, P'Q) \),
where \( r = 0 \).

We shall abbreviate \( S^c_t(P; Q_1, \ldots, Q_r; Q) \) to \( S^c_t(P; Q) \).

Now according to Hamblin, the following important equivalence holds:

\[
(18) \quad S^c_t(P; Q_1, \ldots, Q_r; Q) = \bigvee_{\mu \leq d \leq \nu,}
\]

\[
[S^d_t(P; Q_1, \ldots, Q_{r-1}; Q_r^2Q) \& S^c_{t-2c-d}(P; Q_1, \ldots, Q_{r-1}; Q_rQ)],
\]

where \( \mu = \max(-t + 1, 2c - t + 1) \) and \( \nu = \min(t - 1, 2c + t - 1) \).

It can be shown that (18) follows from (15). Thus, given any \( S \) formula, we can find an equivalent formula containing no \( S \) formulas (but only \( G \) formulas and other formulas of Real Algebra) by repeated use of (18) and Def. 8b). It remains only to express formulas of form (c) in terms of \( S \) and \( G \) formulas. Following Hamblin, we have

---

*It is not clear that the \( t \) parameter is needed, as pointed out by S. Cook. Without it, Hamblin's Method would be more efficient in the absolute sense, though the partial measure of efficiency given below is not affected in either case.*
(19) \((\text{Ex}) (P(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)\)

\[= (a_m = 0 \& (\text{Ex}) [RdP(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0])\]

\[\forall (a_m > 0 \& \sqrt{1 \leq t \leq m, [G^t(P, P'Q_1^2 \ldots Q_r^2) \& S^t_{>0}(P; Q_1, \ldots, Q_r; 1)]})\]

\[\forall (a_m < 0 \& \sqrt{1 \leq t \leq m, [G^t(-P, -P'Q_1^2 \ldots Q_r^2) \& S^t_{>0}(-P; Q_1, \ldots, Q_r; 1)]}).\]

Since by repeated use of (18), we can eliminate \(S\) formulas in favor of \(G\) formulas, which we already know how to process, we see how to process formulas of form (c). We first apply (19) to such a formula. We next process the second disjunct in (19), by processing \(G^t(P, P'Q_1^2 \ldots Q_r^2)\), for \(1 \leq t \leq m\), and \(S^t_{>0}(P; Q_1, \ldots, Q_r; 1)\), for \(1 \leq t \leq m\).

After this, we obtain the third disjunct in (19) by minusing all the coefficients in the previous result. Thus no uses of \(\text{REM}\) are required to process the third disjunct. We then apply (19) to

\((\text{Ex}) (RdP(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)\),

and repeat the above procedure. Eventually we will arrive at \((\text{Ex}) (a_0 = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)\), which is equivalent to \(a_0 = 0 \& (\text{Ex}) (Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)\).

Thus, as with Method C, a formula of form (c) reduces to one of form (b), which by Method B reduces again to one of form (c), which finally and fortunately reduces to a vacuously quantified formula of form (b). The method terminates here. One important characteristic of Method C' is that it does not generate formulas of the form \((E_k x) H\).
4.1. The Efficiency of Method C'

Again we shall use as an indication of efficiency the total number of distinct uses of REM required by Method C' to process an arbitrary formula of form (c). We shall call this number WNR'. Again, WNR' is only an indication of efficiency, since, analogous to Method C, a great number of instances of S and G formulas are generated by Method C'. Even if we suppose that we can process distinct G formulas only, still the number of uses of the equality routine and plugging-in operation should be kept in mind in considering the actual efficiency of Method C'.

Now suppose we are given a formula

\[(Ex)(P(x) = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0),\]

where \(\text{Deg } P = m\) and \(\text{Deg } Q_i = n_i\) and where all the coefficients are distinct variables. We shall first calculate \(NR'(m,r)\), which is the number of distinct uses of REM required by one use of (19) and repeated use of (18) and (1) and (2). To do this we must calculate the number of distinct G formulas generated by one use of (19) and repeated use of (18). Again we shall not regard \(G^0(P, Q)\) as distinct from \(G^d(P, Q)\) when \(d \neq 0\). Also we shall ignore the third disjunct in (19). Now, one use of (19) generates \(G^0(P, P'Q_1^2 \ldots Q_r^2)\), which we may ignore, since it will contribute nothing to \(NR'(m,r)\) (this will shortly become obvious), and generates \(m\) distinct S formulas.
However, it can be shown that all the distinct $G$ formulas generated by these $S$ formulas are among those $G$ formulas generated by $S_{m}^{>0}(P; Q_{1}, ..., Q_{r}; 1)$. (Consider (18) and also $u$ and $v$). Thus, it remains to calculate the number of distinct $G$ formulas generated by $S_{m}^{>0}(P; Q_{1}, ..., Q_{r}; 1)$. Now, a typical $G$ formula generated is of the form $G^{>0}(P, P'Q_{1}^{k_{1}} ... Q_{r}^{k_{r}})$, where $k_{i}$ is either 1 or 2. Furthermore, for all possible combinations of the $k_{i}$, the corresponding $G$ formulas are generated. (Thus we see that we were justified in ignoring $G^{t}(P, P'Q_{1}^{2} ... Q_{r}^{2})$).

Hence, there are $2^{r}$ such distinct $G$ formulas. Since $\text{Deg} \ P = m$, each such $G$ formula requires approximately $(1/2)3^{m-1}$ distinct uses of $\text{REM}$. Hence,

$$NR'_{(m,r)} \approx 2^{r}(\frac{1}{2})3^{m-1} = 2^{r-1}3^{m-1}.$$ 

For justification, it should be stated that any term $\text{REM}(R, T)$ generated in processing $G^{>0}(P, P'Q_{1}^{k_{1}} ... Q_{r}^{k_{r}})$ will be distinct from any term $\text{REM}(R_{1}, T_{1})$ generated in processing $G^{>0}(P, P'Q_{1}^{k'_{1}} ... Q_{r}^{k'_{r}})$, where for some $1 \leq i \leq r$, $k_{i} \neq k'_{i}$. This is because $R$ will be distinct from $R_{1}$ or $T$ will be distinct from $T_{1}$; recall that the coefficients are distinct variables.

We must next calculate the number of distinct uses of $\text{REM}$ required by repeated use of (19) and the above procedure until we arrive at
\((\text{Ex})(a_0 = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0)\).

Call this number \(\text{TNR'}(m, r)\). Analogous with TNR we have,

\[
\text{TNR'}(m, r) = \sum_{i=1}^{m} \text{NR'}(i, r) \approx 2^{r-1}(1 + 3 + \ldots + 3^{m-1}) = 2^{r-2}(3^m - 1).
\]

When \(m = 0\), we have

\[(\text{Ex})(a_0 = 0 \& Q_1(x) > 0 \& \ldots \& Q_r(x) > 0),\]

which is equivalent to

\[a_0 = 0 \forall (\text{Ex})(Q_1(x) > 0 \& \ldots \& Q_r(x) > 0).\]

Thus, we have arrived at a formula of form (b), which, by Method B, gives us back a formula of form (c) whose first polynomial is of degree \(n_1 + \ldots + n_r - 1\). Thus \(\text{TNR'}\) for this formula has the following value:

\[
\text{TNR'}(n_1 + \ldots + n_r - 1, r) = \sum_{i=1}^{n_1 + \ldots + n_r - 1} \text{NR'}(i, r)
\approx 2^{r-2}(3^{n_1 + \ldots + n_r - 1} - 1)
\]

Thus for \(\text{WNR'}\) we have

\[
\text{WNR'}(m, n_1, \ldots, n_r, r) = \text{TNR'}(m, r) + \text{TNR'}(n_1 + \ldots + n_r - 1, r)
\approx 2^{r-2}(3^m + 3^{n_1 + \ldots + n_r - 1} - 2).
\]

Note that we do not have the problem of finding bounds here as we did for \(\text{WNR}\). We have a good approximation to \(\text{WNR'}\). Comparing the two, we see that \(\text{WNR'}\) is considerably
less than WNR. In fact, it is considerably less than the lower bound for WNR.

Let us use the same examples as in Sec. 3.3.1. for WNR' in order to have a concrete comparison.

**Example 1.** $m = 2, n_i = 2 \ (1 \leq i \leq 2), \text{ and } r = 2$.

Thus,

$WNR' \approx 3^2 + 3^3 - 2 = 34 \text{ (well within practicality)}$.

**Example 2.** $m = 10, n_i = 2 \ (1 \leq i \leq 2), \text{ and } r = 2$.

So,

$WNR' \approx 3^{10} + 3^3 - 2 \approx 3^{10} \approx 6 \cdot 10^4 \text{ (within practicality)}$.

**Example 3.** $m = 2, n_i = 10 \ (1 \leq i \leq 2), \text{ and } r = 2$.

$WNR' \approx (3^2 + 3^{19} - 2) \approx 3^{19} \approx 10^9 \text{ (on the border)}$.

Here we see how much more efficient WNR' is over WNR. For in this example, WNR > $10^{36}$.

**Example 4.** $m = 2, n_i = 10 \ (1 \leq i \leq 10), \text{ and } r = 10$.

$WNR' \approx 2^8(3^2 + 3^{99} - 2) \approx 2^8 \cdot 3^{99} \approx 3(10^{49}) \text{ (beyond practicality)}$.

Again compare with WNR > $10^{193}$.

**Example 5.** $m = 2, n_i = 2 \ (1 \leq i \leq 10), \text{ and } r = 10$.

$WNR' \approx 2^8(3^2 + 3^{19} - 2) \approx 2^8 \cdot 3^{19} \approx 3 \cdot 10^{11} \text{ (on the border)}$.

Compare with WNR > $10^{40}$. As was noted before this is an interesting example because it may frequently be generated by TDP. With Method C' there is a chance for such a formula whereas with Method C there is not.
Example 6. \( m = 10, n_i = 2 \ (1 \leq i \leq 10) \), and \( r = 10 \).

\[ \text{WNR}' \approx 3 \cdot 10^{11} \] (on the border; similar to Ex. 5).

Example 7. \( m = 10, n_i = 10 \ (1 \leq i \leq 2) \), and \( r = 2 \).

\[ \text{WNR}' \approx 10^9 \] (on the border; similar to Ex. 3).

Example 8. \( m = 10, n_i = 10 \ (1 \leq i \leq 10) \), and \( r = 10 \).

\[ \text{WNR}' \approx 3(10^{49}) \] (beyond practicality; similar to Ex. 4).

Thus we see that as regards uses of REM, Method C' is considerably more efficient than C. Of the eight formulas, six were shown to be beyond practicality for Method C, whereas only two were shown to be so for Method C'. Also, Method C' seems to be more efficient as regards other factors relevant to efficiency (such as the number of uses of the equality routine or the plugging-in operation). But because of these other factors, we should expect that when WNR' is on the border of practicality for a given formula, then the actual efficiency of Method C' for this formula is probably beyond practicality. Similar remarks hold for Method C. Thus the outlook for both methods seems rather grim; neither is really efficient, even for modest formulas of Real Algebra.
5. CONCLUSION

We would recommend against any attempt to program TDP using either Method C or C'. A new method for processing formulas of form (c) must be found. Perhaps a modification of Cohen's TDP will prove efficient. In any case, an attempt should be made to find an efficient method for processing formulas of form (c). Until such a method is found, it will not be practical to computerize Real Algebra. However, there is the possibility that such a method does not exist. That is to say, it may be the case that there is no efficient quantifier-elimination method for processing formulas such as in the above examples, or perhaps slightly "larger" formulas. Thus the most efficient method may still be inefficient. If this is the case, it would be an interesting result if it could be shown. And if it could be shown, we would never again have to look for such a method.
APPENDIX A

The \( G \) predicates can be defined with the help of the auxiliary functions, \( f(P, Q) \) and \( g(P, Q) \).

DEF: \( f(P, Q) = \) The number of roots \( x \) of \( P \) such that

I. Order \((x, P) - \) Order \((x, Q)\) is a positive odd integer.

II. There exists an open interval \((y, z)\) such that the values of \( P \) and \( Q \) in \((y, z)\) have opposite sign and \( x = z \).

DEF. \( g(P, Q) = f(P, Q) - f(P, -Q) \)

Then the \( G \) predicates are defined as follows:

DEF. \( G^C(P, Q) \equiv g(P, Q) = C \)

DEF. \( G^{>C}(P, Q) \equiv g(P, Q) > C \)

*Note that Tarski has "same" instead of "opposite" in his definition of \( f \). But that is because his theorems corresponding to (1)-(8) are slightly different and so require this modification.
REFERENCES


