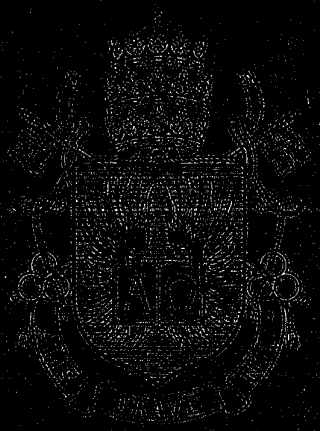


PUC

Series: Monographs in Computer Science
and Computer Applications

Nº 6/72

APPLICATIONS OF QUANTIFIER ELIMINATION TO MECHANICAL THEOREM PROVING

by

Klaus Galda
and
Emmanuel P.L. Passos

Computer Science Department - Rio Datacenter

Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 209 — ZC-20
Rio de Janeiro — Brasil

APPLICATIONS OF QUANTIFIER ELIMINATION TO MECHANICAL THEOREM PROVING

Klaus Galda
Visiting Professor
and
Emmanuel P.L.Passos
Assistent Professor
Computer Science Department
PUC/RJ

This paper was published in the International Congress of Cybernetics and Systems - Oxford - August 28 - September 1st 1972

Series Editor: Prof. A. L. Furtado October /1972

ABSTRACT

The purpose of the paper is to develop algorithms, suitable for computer programming, which apply quantifier elimination methods to prove mathematical theorems. A general discussion attempts to relate the present paper with other work in mechanical theorem proving. A detailed algorithm for the theory of densely ordered sets without first or last element is shown, along with examples of theorems proved with this algorithm. Then the theory of abelian groups with total discrete orderings is considered. Due to the complexity of quantifier elimination in this theory, the algorithm is not presented in complete detail, but an outline showing the principal stages is given. The similarities between the two algorithms are emphasized to show how all the programs can be written within the same framework. A discussion of the actual computer implementation of the algorithm concludes the paper.

APPLICATIONS OF QUANTIFIER ELIMINATION TO MECHANICAL THEOREM PROVING

In this paper we will develop algorithms, based on the method of quantifier elimination for certain first-order theories, and apply them to computer programs for theorem proving. The general method of quantifier elimination is due to Tarski[1]. It was used by him to prove the decidability of the first-order theories of algebraically closed fields and real-closed fields. A number of other mathematically interesting theories have been shown to be decidable by this method. These include the theory of abelian groups[2], elementary Euclidean geometry[1], elementary hyperbolic geometry[3], densely ordered sets[4], and the theory of Boolean algebras[1]. A survey of the area and a relatively complete bibliography are found in Ershov, Lavrov, Taimanov, and Taitslin[5]; the proofs on which our work is based are found in Kreisel and Krivine[6].

We have not found any extensive discussions in the literature on the applicability of quantifier elimination to mechanical theorem proving. Seidenberg[7] makes a few comments on the implementation of his decision method for real-closed fields, which is somewhat different from Tarski's, and, according to Seidenberg, easier to implement. In the case of simpler theories, such as densely ordered sets , it is feasible to construct proofs of short theorems using quantifier elimination and manual computations. However, in the cases of the field theories and the group theory of Example 2 below, the formulas involved in the method become very complex, and computation by hand does not seem reasonable.

The emphasis in the more recent work on mechanical theorem proving has been rather different from the approach taken in this paper. Most of it, like all of the results based on the resolution method of Robinson[8] is done within undecidable theories. Consequently, one could not expect to have a general algorithm that is guaranteed to terminate, stating that a sentence is "true" or "false" in the theory. The principal advantage of the quantifier elimination method is that, within the theories for which the method has been established, it will always terminate for any sentence in the language. In addition to guaranteed termination, we can also give a bound on the number of steps necessary to prove (or disprove) a formula. This bound depends on the individual formula; shorter formulas will, in general, have shorter proofs. (In a later paper, we will attempt to give some precise relations between the structure of a formula and the number of steps required to terminate the algorithms). Another advantage of quantifier elimination is that there are no extraneous formulas and no false starts generated by the algorithms. A simple examination of the algorithms will show that every formula generated is useful in constructing the proof.

There are some obvious disadvantages in the method of eliminating quantifiers. The most serious is its lack of universality; it is intended to be used only for certain decidable first-order theories. However, we can apply the method with partial success to other first-order theories. Suppose that we have a first-order theory T, such that T can be (consistently) extended to a theory T' for which we have a quantifier elimination algorithm. Let A be a formula in the language of T. We can apply T' quantifier elimination to A and have a

partial decision method for A. If A is not a theorem in T' then clearly A could not be a theorem in T. On the other hand, if A is a theorem of T' then we cannot conclude anything about A in T, unless T' has been proven to be a conservative extension of T. Take as an example, the first-order theory of groups which is known to be undecidable. It can be extended to the theory of abelian groups, which permits elimination of quantifiers. There is also a second type of associated partial decision method. If T admits of quantifier elimination and A is a theorem of T, then clearly A must be a theorem of T'. In practice, the first partial decision method is more useful because most interesting theories are not extensions of quantifier elimination theories. Rather, the quantifier elimination theories are generally extensions of the more basic (undecidable) theories.

Another disadvantage of quantifier elimination is that there is no single algorithm that works in all theories for which the elimination method approach to proving decidability is valid. The algorithm depends on the structure of the theory involved. As we will show, we do have the advantage that there are many subroutines common to several or all of the algorithms. These provide a general framework within which individual algorithms are developed. A third disadvantage inherent in quantifier elimination is that it will not, in general, give a short and simple proof of a trivial theorem. This will be seen in the examples given below.

An outline of the manner in which quantifier elimination works follows. We begin with any well-formed sentence in the language of the theory and transform it into prenex normal form. Let the prenex formula be $(Q_1v_1)(Q_2v_2)\ldots(Q_nv_n)M$. Each $Q_iv_i (i = 1,\ldots,n)$ is either a

universal quantifier (v_i) or an existential quantifier (Ev_i) and M is a quantifier-free formula. Each (v_i) is replaced by $\sim(Ev_i)\sim$; any double negations between the quantifiers are immediately eliminated. The formula is now of the form $(\pm Ev_1)\ldots(\pm Ev_n)M'$, where $\pm$ indicates that there may or may not be a negation in front of the quantifier. M' is M if the last quantifier was existential, or M' is $\sim M$ if the last quantifier was universal. M' is now transformed into disjunctive normal form, and the last existential quantifier is distributed over the disjunction. The original formula becomes $(\pm Ev_1)\ldots(\pm Ev_{n-1})\pm((Ev_n)P_1 \vee (Ev_n)P_2 \vee \ldots \vee (Ev_n)P_k)$, where each $P_j (j = 1,\ldots,k)$ is a conjuction of atomic and negations of atomic formulas. In general, each P_j will contain free occurrences of some or all of the $v_i (i = 1,\ldots,n)$.

The problem is to find quantifier-free formulas $P'_1,\ldots,P'_k$ such that $(Ev_n)P_j \iff P'_j$ for all $j = 1,\ldots,k$. This part of the process depends on the particular theory, although there are some steps which are used in several different theories. P'_j will always be free of v_n, but may contain some of the other variables. If some P'_j contain no variables or are equivalent to certain listed tautologies then we may considerably simplify the resulting formula by a truth table analysis. If not, then the original formula is equivalent to $(\pm Ev)\ldots(\pm Ev_{n-1})\ (P'_1 \vee \ldots \vee P'_k)$. The part of the formula following the last existential quantifier is put into disjunctive normal form and we proceed to eliminate Ev_{n-1} in the same way as we eliminated Ev_n. The process will continue until Ev_1 is eliminated, leaving a quantifier-free formula containing no variables which can be evaluated to be true or false in the theory.

We now present two examples of theories which have been shown to be decidable by elimination of quantifiers. For each theory we give in some detail our algorithm.

Example 1: (Theory of dense total orderings without first or last element). Examples of models of this theory are the sets of rational and real numbers with the usual "less than" ordering. The language of the theory contains as primitive symbols a denumerable set of variables $v_1, v_2, \ldots$ and two binary relation symbols $<$, $=$. The axioms are those of the usual predicate logic with equality plus the following special axioms:

A1) $(v_1)\sim(v_1 < v_1)$

A2) $(v_1)(v_2)(v_3)(v_1 < v_2 \wedge v_2 < v_3 \Rightarrow v_1 < v_3)$

A3) $(v_1)(v_2)(v_1 < v_2 \vee v_1 = v_2 \vee v_2 < v_1)$

A4) $(v_1)(v_2)(\exists v_3)(v_1 < v_2 \Rightarrow v_1 < v_3 \wedge v_3 < v_2)$

A5) $(v_1)(\exists v_2)(v_1 < v_2)$

A6) $(v_1)(\exists v_2)(v_2 < v_1)$

Since the elimination of quantifiers algorithm for this theory is simple we present it as a flow chart (Figure 1). The steps of the algorithm are justified by known theorems of the theory[4]. The algorithm we present is a modification of, but much more efficient than, an algorithm found in[4]. First we give a list of formula transformations used in the algorithm.

The following tests and formula transformations are used in the flow chart:

SUBS 0 - Replace each (v_i) by $\sim(Ev_i)\sim$.

SUBS 00 - Eliminate all double negations.

SUBS 01 - Look for tautologies and contradictions and replace by T or $\sim T$,

(a) $v_i = v_i <=> T$ (b) $v_i < v_i <=> \sim T$

SUBS 1 - Simplify the formula by repeatedly applying the following substitutions (also commuted forms of 2-5):

(1) $\sim \sim T$ by T

(2) $Q \vee T$ by T

(3) $Q \vee \sim T$ by Q

(4) $Q \wedge T$ by Q

(5) $Q \wedge \sim T$ by $\sim T$

(6) $Q => T$ by T

(7) $Q => \sim T$ by $\sim Q$

(8) $T => Q$ by Q

(9) $\sim T => Q$ by T

(10) $T <=> Q$ by Q

(11) $\sim T <=> Q$ by $\sim Q$

(12) $(Ev_i)T$ by T

(13) $(Ev_i)\sim T$ by $\sim T$ (where Q is any formula).

TEST 1 - Check if the entire formula is already T or $\sim$T.

SUBS 2 - Eliminate negations by the substitutions

(a) $\sim(v_i = v_j)$ by $v_i < v_j \vee v_j < v_i$,

(b) $\sim(v_i < v_j)$ by $v_i = v_j \vee v_j < v_i$

SUBS DNF - Transform the matrix of the formula under consideration into disjunctive normal form. $(Ev_n)Q$ becomes $(Ev_n)(Q_1 \vee Q_2 \vee \ldots \vee Q_k)$ where Q_i is $\alpha_1 \wedge \ldots \wedge \alpha_{\ell_i}$ with each α_j atomic or negation of atomic.

SUBS 03 - Distribute (Ev_n) over the disjunction.

SUBS 003 - Replace $(Ev_n)(\alpha_1 \wedge \ldots \wedge \alpha_{j-1} \wedge \alpha_j \wedge \alpha_{j+1} \wedge \ldots \wedge \alpha_{\ell_i})$ by

$\alpha_j \wedge (Ev_n)(\alpha_1 \wedge \ldots \wedge \alpha_{j-1} \wedge \alpha_{j+1} \wedge \ldots \wedge \alpha_{\ell_i})$

TEST 2 - Check whether v_n appears on the same side of all inequalities in $(Ev_n)(\alpha_1 \wedge \ldots \wedge \alpha_{\ell_i})$

SUBS 3 - Replace $(Ev_n)(v_{k_1} < v_n \wedge \ldots \wedge v_{k_j} < v_n \wedge v_n < v_{\ell_1} \wedge \ldots \wedge v_n < v_{\ell_m})$ by

$v_{k_1} < v_{\ell_1} \wedge \ldots \wedge v_{k_1} < v_{\ell_m} \wedge v_{k_2} < v_{\ell_1} \wedge \ldots \wedge v_{k_2} < v_{\ell_m} \wedge \ldots \wedge v_{k_j} < v_{\ell_1} \wedge \ldots$

$\ldots \wedge v_{k_j} < v_{\ell_m}$

ALGORITHM FOR DENSELY ORDERED SETS

Figure 1-a

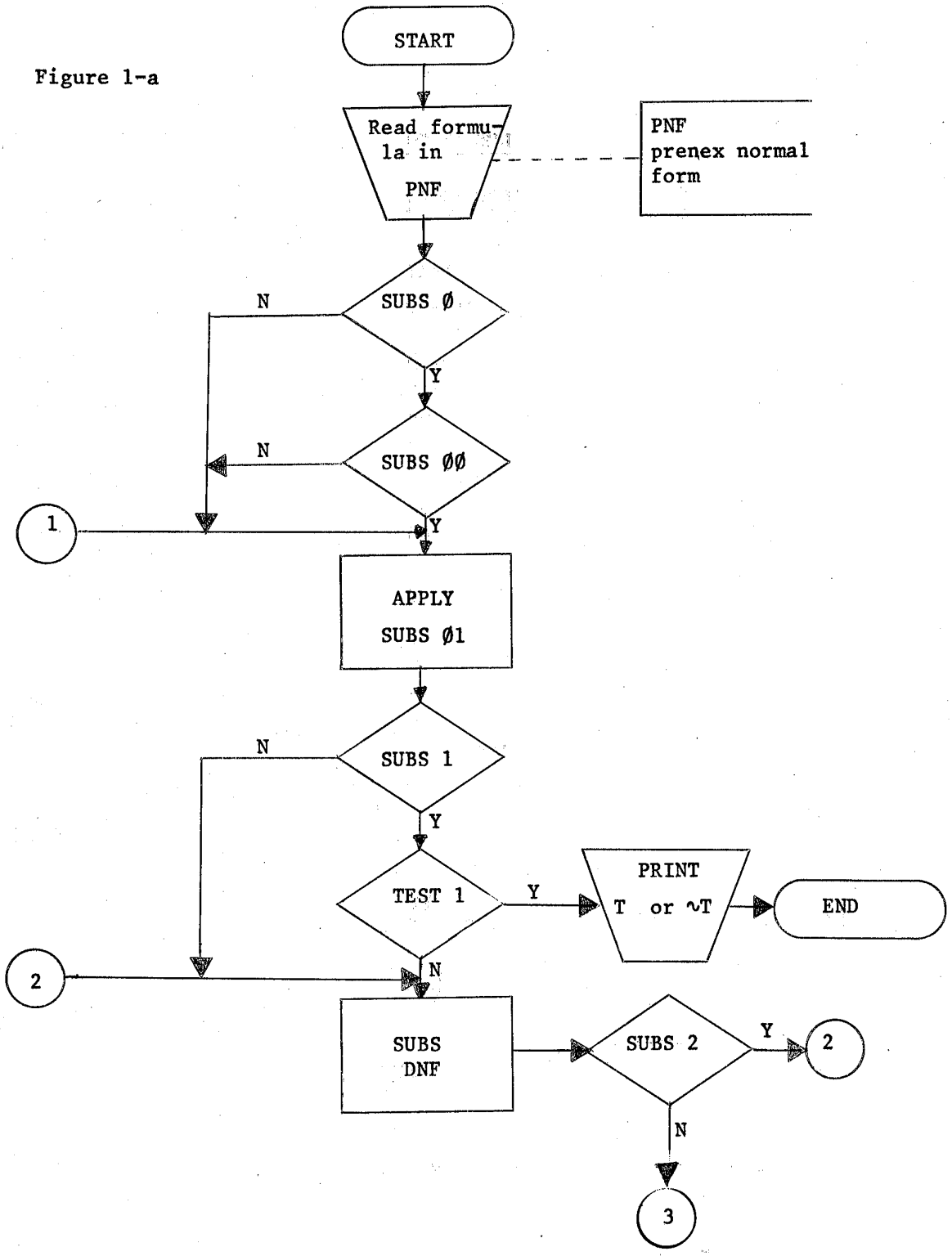


Figure 1-b

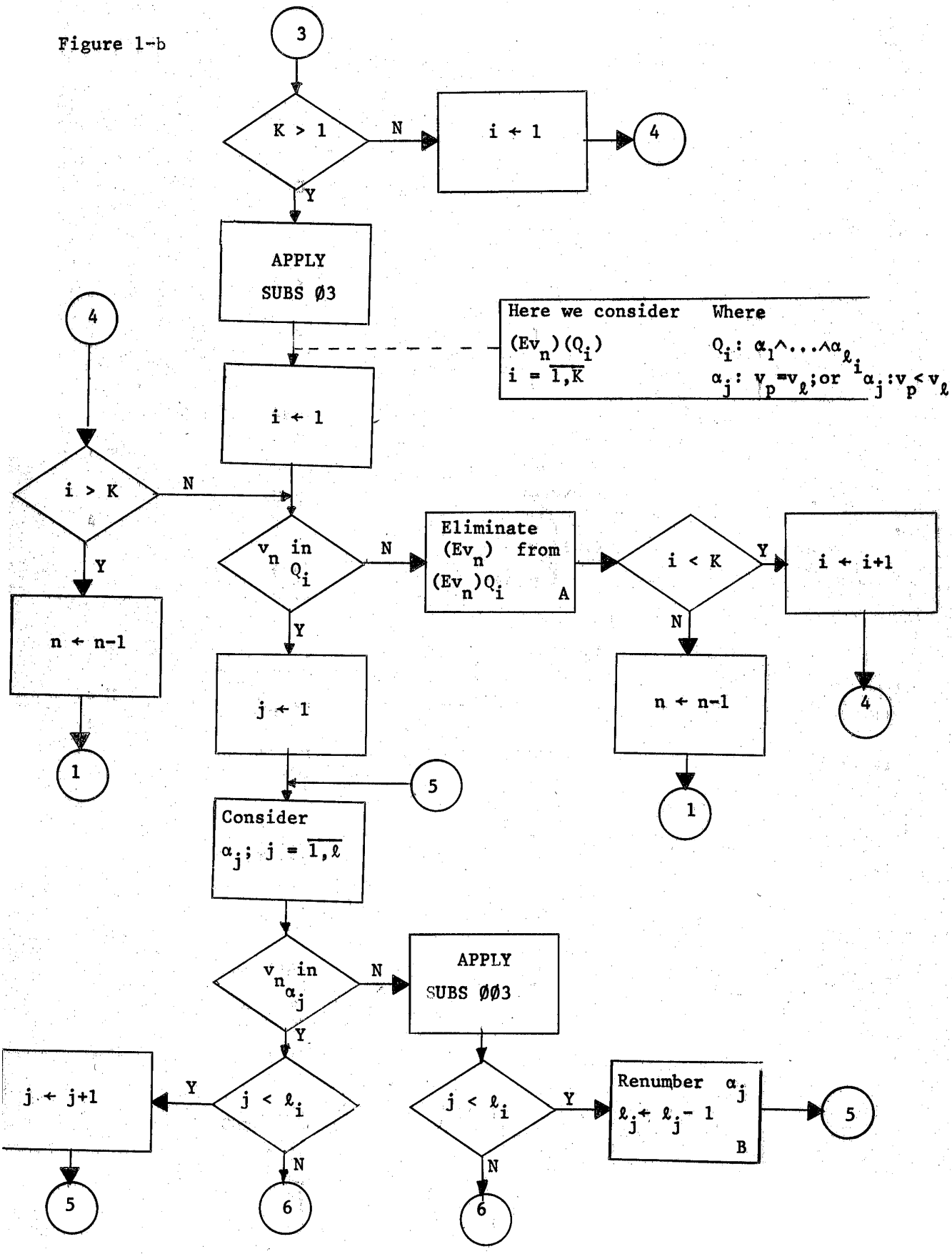

3
K > 1
N
i ← 1
4
Y
APPLY
SUBS Ø3
Here we consider
$(Ev_n)(Q_i)$
$i = \overline{1,K}$
Where
Q_i: $\alpha_1 \wedge \ldots \wedge \alpha_{\ell_i}$
α_j: $v_p = v_\ell$; or α_j: $v_p < v_\ell$
4
i ← 1
i > K
N
Y
v_n in Q_i
N
Eliminate (Ev_n) from $(Ev_n)Q_i$
A
i < K
Y
i ← i+1
n ← n-1
Y
N
n ← n-1
4
1
j ← 1
1
5
Consider
α_j; $j = \overline{1,\ell}$
v_n in α_j
N
APPLY
SUBS ØØ3
Y
j ← j+1
Y
$j < \ell_i$
$j < \ell_i$
Y
Renumber α_j
$\ell_j \leftarrow \ell_j - 1$
B
5
N
N
5
6
6

Figure 1-c

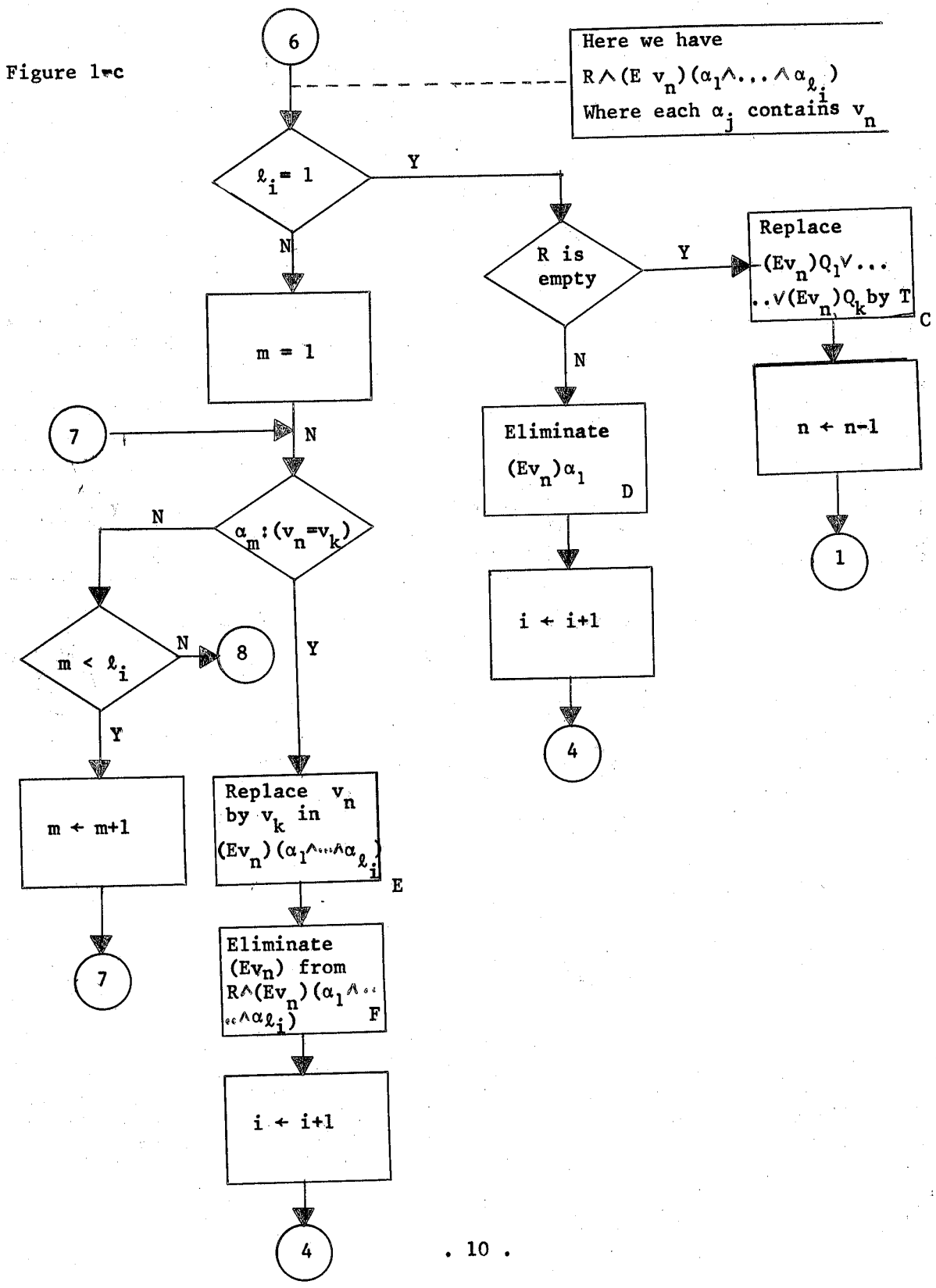
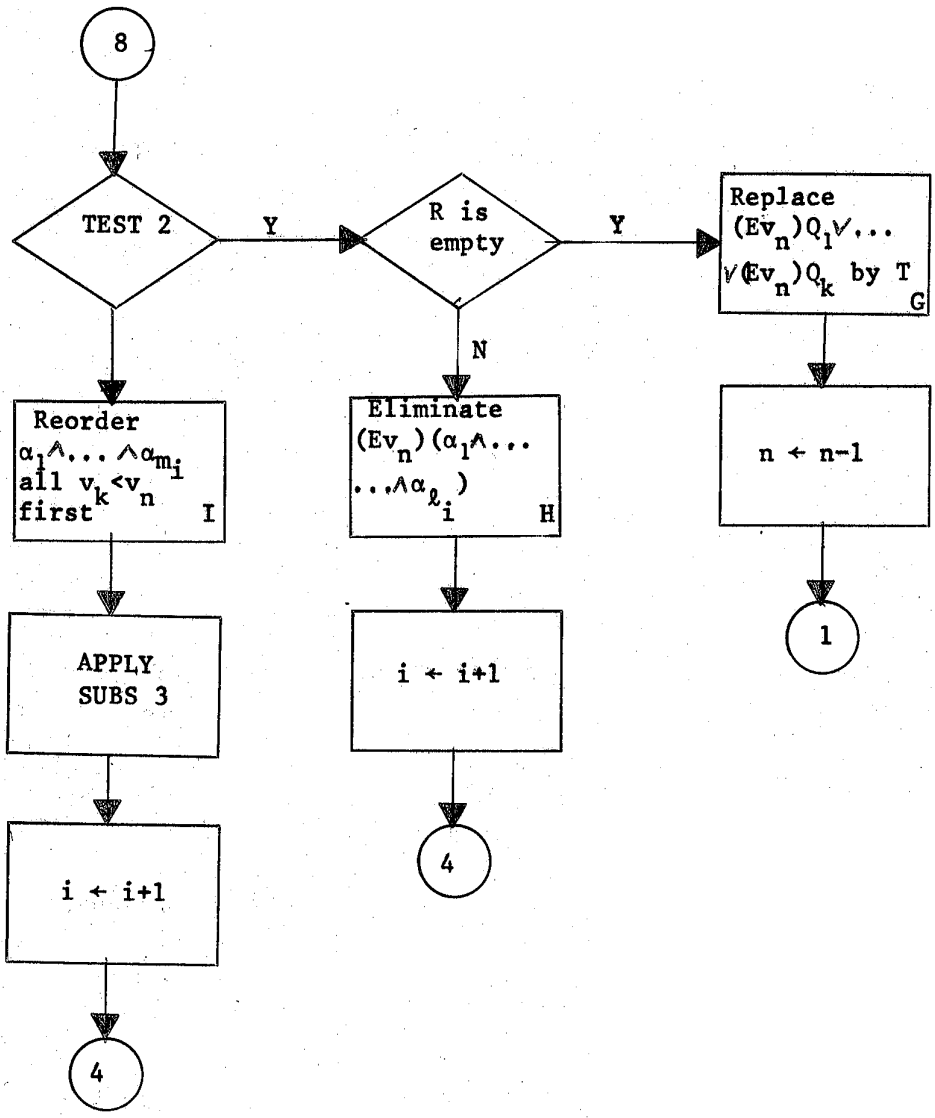

6
Here we have
R∧(E v_n)(α_1∧...∧α_ℓi)
Where each α_j contains v_n
ℓ_i = 1
Y
N
R is empty
Y
Replace
(Ev_n)Q_1∨...
..∨(Ev_n)Q_k by T
C
N
m = 1
Eliminate
(Ev_n)α_1
D
n ← n-1
7
N
1
α_m : (v_n = v_k)
N
Y
m < ℓ_i
N
8
i ← i+1
4
Y
m ← m+1
Replace v_n
by v_k in
(Ev_n)(α_1∧...∧α_ℓi)
E
7
Eliminate
(Ev_n) from
R∧(Ev_n)(α_1∧...
...∧α_ℓi)
F
i ← i+1
4

Figure 1-d


8
TEST 2
Y
R is empty
Y
Replace $(Ev_n)Q_1 \vee \ldots \vee (Ev_n)Q_k$ by T
G
N
Reorder $\alpha_1 \wedge \ldots \wedge \alpha_{m_i}$ all $v_k < v_n$ first
I
Eliminate $(Ev_n)(\alpha_1 \wedge \ldots \ldots \wedge \alpha_{\ell_i})$
H
$n \leftarrow n-1$
1
APPLY SUBS 3
$i \leftarrow i+1$
$i \leftarrow i+1$
4
4

Applications of the algorithm:

(a) Formula to be proven: $(Ev_1)(v_2)(v_1 < v_2)$

$(Ev_1)\sim(Ev_2)\sim(v_1 < v_2)$ (SUBS 0)

$(Ev_1)\sim(Ev_2)\ (v_2 < v_1 \vee v_1 = v_2)$ (SUBS 2)

$(Ev_1)\sim((Ev_2)(v_2 < v_1) \vee (Ev_2)(v_1 = v_2))$ (SUBS 03)

$(Ev_1)\sim(\ T\ )$ (C)

$\sim T$ (SUBS 1-13)

(b) Formula to be proven: $\sim(v_1)(Ev_2)(v_1 < v_2 \wedge v_2 < v_1)$

$(Ev_1)(v_2)\sim(v_1 < v_2 \wedge v_2 < v_1)$ Formula in PNF

$(Ev_1)\sim(Ev_2)\sim\ \sim(v_1 < v_2 \wedge v_2 < v_1)$ (SUBS 0)

$(Ev_1)\sim(Ev_2)\ (v_1 < v_2 \wedge v_2 < v_1)$ (SUBS 00)

$(Ev_1)\sim(v_1 < v_1)$ (SUBS 3)

$(Ev_1)\sim(\sim T)$ (SUBS 01)

$(Ev_1)T$ (SUBS 1-1)

T (SUBS 1-12)

Example 2: Abelian groups with discrete total orderings (e.g. additive group of integers).

The language of this theory consists of the language of Example 1 plus the following:

(i) constant symbols 0,1

(ii) unary function symbol -

(iii) binary function symbol +

We abreviate $s+(-t)$ by $s-t$ (where s,t are arbitrary terms). In addition we introduce the terms $n = 1 + 1 + \ldots + 1$ (n times) and $nt = t + t + \ldots + t$ for any term t. Note that nt does not mean multiplication. We also have for each term $n = 1,2,3,\ldots$ a unary relation $n|$ such that $n|t \iff (Ev_1)(t = nv_1)$.

The axioms of the theory are:

B1) $(v_1)(v_2)(v_3)((v_1 + v_2) + v_3 = v_1 + (v_2 + v_3))$

B2) $(v_1)(v_2)(v_1 + v_2 = v_2 + v_1)$

B3) $(v_1)(v_1 + 0 = v_1)$

B4) $(v_1)(v_1 - v_1 = 0)$

B5) $(v_1)(v_2)(0 < v_1 \wedge 0 < v_2 \Rightarrow 0 < v_1 + v_2)$

B6) $(v_1) \sim (0 < v_1 \wedge 0 < -v_1)$

B7) $(v_1)(v_1 \leq 0 \vee v_1 = 0 \vee -v_1 < 0)$

B8) $(v_1)(0 < v_1 <=> (v_1 = 1 \vee 0 < v_1 - 1))$

B9) $(v_1)(n|v_1 <=> (Ev_2)(v_1 = nv_2))$ for all $n = 1,2,3,\ldots$

B10) $(v_1)(n|v_1 \vee n|v_1 + 1 \vee \ldots \vee n|v_1 + n - 1)$ for all $n = 1,2,3,\ldots$

Since the algorithm is quite complicated we will not present the details, but merely give a brief outline of what it does. After the usual preliminaries (see Ex. 1) we reduce the problem to eliminating the existential quantifier from a formula of the type $(Ev_i)(P_1 \wedge \ldots \wedge P_{k_i})$ with P_j atomic. Negations were eliminated by SUBS 2 and the equivalence $\sim(n|t) <=> (n|t+1) \vee (n|t+2) \vee \ldots \vee (n|t+n-1)$. At first sight there appears to be a problem in representing formulas like $(n|t+1) \vee \ldots \vee (n|t+n-1)$ in the computer. In practice, however, the formulas we consider do not contain variable n, but only fixed integer values, because B9 and B10 are considered infinite sets of axioms.

By algebraic manipulations we can make each P_j equivalent to either $p_j v_i = t_j$ or $p_j v_i < t_j$ or $n_j | p_j v_i + t_j$, where n_j, p_j are fixed integers and t_j is a term not containing v_i. Consequently we are left with a formula

(*) $(Ev_i)(p_1 v_i < t_1 \wedge \ldots \wedge p_k v_i < t_k \wedge q_1 v_i = u_1 \wedge \ldots \wedge q_\ell v_i = u_\ell \wedge n_1 | r_1 v_i + s_1 \wedge \ldots$

$\ldots \wedge n_m | r_m v_i + s_m)$

with p_j, q_j, r_j integers and s_j, t_j, u_j terms free of v_i. Now replace

$n_j|r_jv_i + s_j$ by $(n_j|r_jv_i \wedge n_j|s_j) \vee \ldots \vee (n_j|r_jv_i + n_j - 1 \wedge n_j|s_j + n_j - 1)$.

The new formula is then transformed into conjunctive normal form and the formulas free of v_i are taken out of the scope of the quantifier, leaving (*) with fixed integers s_j.

Reduce (*) as follows:

(1) Define the rank of (*) by

$$h = |p_1| + \ldots + |p_k| + |q_1| + \ldots + |q_\ell| + n_1 + \ldots + n_m + |r_1| + \ldots$$
$$\ldots + |r_m|$$

(2) If $\ell > 1$ then reduce (*) to a formula of lower rank by:

(a) Let $q' = q_1$ if $|q_1| \leq |q_2|$ $\quad q'' = q_2$ if $|q_1| \leq |q_2|$

$= q_2$ otherwise $\quad = q_1$ otherwise

and $u' = u_1$ if $q' = q_1$ $\quad u'' = u_2$ if $q'' = q_2$

$= u_2$ otherwise $\quad = u_1$ otherwise

(b) Substitute $q'v_i = u' \wedge (q'' - q')v_i = u'' - u'$ for

$q_1v_i = u_1 \wedge q_2v_i = u_2$.

(c) Let $q_1 \leftarrow q'$, $q_2 \leftarrow q'' - q'$, $u_1 \leftarrow u'$, $u_2 \leftarrow u'' - u'$.

(d) Replace $0v_i$ (if it occurs) by 0 and more resulting u=0 outside the quantifier and then let $j \leftarrow j-1$ for $j = 2,\ldots,\ell$.

(3) Repeat (2) until there is only one equality left in (*). This will, in general, take fewer than $|q_1| + \ldots + |q_\ell|$ steps.

(4) If $k>1$ then replace (*) by

$((p_2t_1 < p_1t_2 \vee p_2t_1 = p_1t_2) \wedge R^2) \vee (p_1t_2 < p_2t_1 \wedge R^1)$, where R^j is the same as (*) except that $p_jv_i < t_j$ is eliminated, leaving formulas of lower rank.

(5) Repeat (4) on the new formulas until the problem is reduced to eliminating (Ev_i) from 2^{k-1} formulas, each involving only one inequality.

(6) Each formula $(Ev_i)(pv_i < t \wedge qv_i = u \wedge n_1|r_1v_i + s_1 \wedge \ldots \wedge n_m|r_mv_i + s_m)$ is equivalent to either (i) $qt < pu \wedge P$ if $q>0$

or (ii) $pu < qt \wedge P$ if $q<0$,

where P is $q|u \wedge qn_1|r_1u + s_1q \wedge \ldots \wedge qn_m|r_mu + s_mq$.

The above reduction depends on the fact that $k>1$ and $\ell > 1$.

Let us look at the other cases.

Case 1: $k = 1$ and $\ell = 1$. Merely apply step 6.

Case 2: $k = 1$ and $\ell > 1$. Apply steps 2,3 and 6.

Case 3: $k > 1$ and $\ell = 1$. Apply steps 4-6.

Case 4: $k = 0$ and $\ell = 1$. This is the same as Case 1, except the inequalities do not appear.

Case 5: $k = 0$ and $\ell > 1$. This is similar to Case 2.

The cases with $\ell = 0$ use some results from the theory, which we have not discussed but which can be easily programmed. For details see[6]. There are various places in the algorithm where tautologies or contradictions of the forms $t = t$, $t < t$ and $n|n$ are introduced into the formulas. At these places we do a truth analysis similar to that of Example 1, which makes execution of the algorithm much more efficient.

The algorithms are currently being implemented on an IBM/370-165 in a version of LISP 1.5. The formula transformations are special functions of the LISP library which can be used by any of the algorithms. For the two algorithms that we have developed almost half of the special functions are common to both algorithms. We expect most of these to be useful in other algorithms as well.

REFERENCES

1. A Tarski - A Decision Method for Elementary Algebra and Geometry - Berkeley, 1951.

2. W. Szmielew - "Elementary properties of abelian groups" - Fundamenta Mathematicae, vol. 41, 1954, pp. 203-271

3. W. Schwabhauser - "Entscheidbarkeit und Vollständigkeit der elmentaren hyperbolischen Geometrie" - Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, vol. 5, 1959, pp.132-205.

4. A. Margaris - First Order Mathematical Logic - Waltham, Mass., 1967.

5. Y.L. Ershow, I.A. Lavrov, A.D. Taimanov and M.A. Taitslin - "Elementary Theories" - Russian Mathematical Surveys, vol. 20, No.4, 1965 , pp. 35-105.

6. G. Kreisel and J.L. Krivine - Elements of Mathematical Logic: Model Theory, Amsterdam, 1967.

7. A. Seidenberg - "A new decision method for elementary algebra" - Annals of Mathematics, vol. 60, no.2, Sept. 1954 pp. 365-374.

8. J.A. Robinson - "A machine-oriented logic based on the resolution principle", Journal of the Association for Computing Machinery, vol. 12, no.1, Jan. 1965, pp. 23-41.