5

Theorem Proving in Arithmetic without Multiplication

D. C. Cooper
Department of Computer Science
University College of Swansea

INTRODUCTION

A considerable amount of effort has been expended in the last ten years in the field known as 'mechanical theorem proving'. Original motivation in this area was primarily the attempt to prove interesting theorems in mathematics although applications in other areas were considered – see, for example, McCarthy (1958), where application to more general problem solving is made, and W.S.Cooper (1964), where the need for a theorem proving system in information retrieval was considered. More recently other areas have arisen such as robotology, automatic program construction and proofs of correctness of programs. Most work has been done in the first order predicate calculus, but at various times pleas have been made not only for general theorem provers but also for efficient provers in restricted areas. One such is presented in this paper.

In Cooper (1971) a set of programs was described to aid the proving of the convergence and correctness of programs. Part of this package was a theorem prover for the logical system consisting of the integers, integer variables, addition, the usual arithmetical relations and the usual first order logical connectives. This system is commonly referred to as Presburger arithmetic – see Presburger (1929), where a similar system involving only the equality relation was proved decidable. Whilst in such a system one can only state rather simple theorems, yet an efficient algorithm to test for the validity of such formulas is useful in that it can quickly dispose of a host of the simpler formulas arising in some application, leaving only the more complex to be dealt with by some more general theorem prover or by the human. However, the present known algorithm proved impractical even on the formulas produced by the simple program in that paper, due to exponential growth. In this paper we investigate this growth and give a new algorithm which completely eliminates one factor in the growth and considerably mitigates another. It is possible that decision procedures in other areas could be related to that for Presburger arithmetic, thus providing another outlet for the algorithm. One such area is the equivalence of certain special program schemas – see section 4 of Paterson (1972).

THE LOGICAL SYSTEM

A formula of the system is formed from algebraic expressions (only allowing variables, integer constants and addition), the binary relation $<$, the propositional calculus logical connectives and quantification. The domain is that of positive and negative integers, and formulas of the system take their usual meaning. The other arithmetic relations may easily be added – for example $a \geqslant b$ can be defined as $b<a+1$ and $a=b$ as $a<b+1 \wedge b<a+1$ – although it could well lead to a more efficient algorithm if equality were included in the basic system. Subtraction can be allowed: $a-b>c$ is $a>c+b$. The major omission is multiplication, although multiplication by a constant can be included: $3*a$ is $a+a+a$. Such a system is known to be decidable. Examples of simple formulas in the system are:

$$(\forall a)(\forall b)(\exists x)(a<20x \wedge 20x<b)$$

and

$$(\forall a)(\exists b)(a<4b+3a \vee (\neg a<b \wedge a>b+1))$$

For the technique to be used in the decision procedure it is essential to introduce another relation, even though that relation is definable in the system. This relation can be taken to be $\delta \mid \alpha$ (α is exactly divisible by δ) where α is a term but δ must be an integer. This relation can be defined as $(\exists x)(\alpha=\delta * x)$, assuming x does not occur free in α.

THE PRESENT ALGORITHM

Decision algorithms for this system are well known – see, for example, Hilbert and Bernays (1968), or Kreisel and Krivine (1967). These are based on the method known as 'elimination of quantifiers'. Clearly if, given a formula $(\exists x)F$ where F is quantifier free, we can construct an equivalent quantifier-free formula G then we have a decision algorithm: first, close the formula by universally quantifying all free variables, then eliminate quantifiers from the inside thus obtaining a formula without variables which may be evaluated to *true* or *false*. Universal quantifiers may be replaced by existential quantifiers – use $(\forall x)F \equiv \neg(\exists x)\neg F$ – and our basic task is therefore the elimination of a quantifier from a formula of the form $(\exists x)F$ where F is quantifier free but may involve x and other variables. This may be accomplished as follows:

Step 1

Transform F to disjunctive normal form and distribute the quantifier over the disjuncts. As a result we have a number of separate eliminations to perform in each of which F is a conjunct of relations or the negation of relations.

Step 2

Eliminate negation by using $\neg\alpha<\beta$ is $\beta<\alpha+1$ and $\neg(\delta \mid \alpha)$ is $\bigvee_{i=1}^{\delta-1} \delta \mid \alpha+i$.

Step 3

Simplify each relation by collecting the x-terms so that each relation either does not involve x (in which case take it outside the quantifier) or is one of

$$\lambda_i x < \alpha_i$$
$$\beta_i < \mu_i x$$
$$\delta_i \mid \nu_i x + \gamma_i,$$

where λ_i, μ_i, ν_i and δ_i are positive integers and α_i, β_i, γ_i are expressions not involving x.

Step 4

Let δ be the L.C.M. of all the λ_i, μ_i and ν_i. By multiplying both sides of all relations by appropriate constants the coefficient of all x's may be made δ. Now replace $(\exists x)F(\delta x)$ by $(\exists x)(F(x) \wedge \delta \mid x)$. The result will be a conjunct of terms as at the end of Step 3 except that now the coefficients of all x's are unity.

Step 5

The elimination may now be performed by using a generalisation of the equivalence:

$$(\exists x)(\alpha < x \wedge x < \beta \wedge \delta \mid x) \equiv \bigvee_{j=1}^{\delta} (\alpha + j < \beta \wedge \delta \mid \alpha + j).$$

This generalisation is given in full in Cooper (1971) but it will not be given here as the new algorithm is both more efficient and more succinct.

FORMULA EXPANSIONS

There are two main sources of expansion which make the above algorithm unworkable on any but the simplest formulas. First, there is the initial transformation to disjunctive normal form. In some applications this might not be troublesome; however, in the application of Cooper (1971) the formula produced for the algorithm tended to be long but repetitious, with many common sub-expressions. The simplest formula (about 100 characters long) had 55 disjuncts. As the formula was not valid every separate disjunct had to be proved invalid; the variations were primarily in parts of the disjuncts not contributing to the inconsistency, so much work was repeated. A computer program for the algorithm was able to solve this problem, but could not tackle the other formula (up to about 1000 characters long) in any reasonable time because of the very large number of disjuncts. The major feature of the new algorithm is that it is not necessary to have the formula in disjunctive normal form and a program for this new algorithm quickly disposed of all cases.

The second source of expansion occurs in Steps 2 and 5 where new disjuncts are produced whose number of terms depends on the integers occurring in the formula. This can clearly be disastrous, yet in the old algorithm seems to be inevitable. The expansion in Step 2 is avoided altogether in the new algorithm; that in Step 5 remains, but remarks will be made later showing how the effect can be greatly reduced. It should also be noted that in an interesting sub-class of formulas (essentially those in which one only has the successor function, not full addition) the δ of the previous formula is always 1 and this expansion does not occur – this was the case in the formulas occurring in Cooper (1971).

THE NEW ALGORITHM

A new process will now be described for the elimination of an existential quantifier which does not assume the formula is in disjunctive normal form.

Steps 2, 3 and 4 of the previous algorithm did not use the fact that the formula was in disjunctive normal form, and these are again performed. They will not produce any major expansion, apart from the replacement of $\neg(\delta|\alpha)$ in Step 2, and to avoid this we introduce $\nmid$ as a basic relation – $\delta\nmid\alpha$ means that α is not exactly divisible by δ.

The formula is now of the form $(\exists x)F(x)$ where $F(x)$ is formed by conjunction and disjunction (note no negation) of basic relations each of which is one of the forms:

(A) $x<a_i$
(B) $b_i<x$
(C) $\delta_i|x+c_i$
(D) $\varepsilon_i\nmid x+d_i$,

where a_i, b_i, c_i and d_i are expressions not involving x, and δ_i, ε_i are positive integers.

Let δ be the L.C.M. of all δ_i, ε_i and define $F_{-\infty}(x)$ to be $F(x)$ with *true* substituted for all formulas of type (A) and *false* substituted for all formulas of type (B). Clearly, we have:

Lemma. For x sufficiently small $F(x)\equiv F_{-\infty}(x)$.

The following theorem then enables the elimination of the quantifier:

Theorem. With notation and restrictions as above

$$(\exists x)F(x)\equiv\bigvee_{j=1}^{\delta}F_{-\infty}(j)\vee\bigvee_{j=1}^{\delta}\bigvee_{b_i}F(b_i+j)$$

(the second term on the right hand side is omitted if there are no formulas of type (B)).

Proof of theorem. Assume the right hand side is true. Then one of the disjuncts must be true; if it is one of the second term then an x has been found; if $F_{-\infty}(j)$ is true for some j then, by the definition of $F_{-\infty}$, $F_{-\infty}(j)\equiv F_{-\infty}(j-\lambda\delta)$ for any integer λ, and if λ is sufficiently large then $F_{-\infty}(j-\lambda\delta)\equiv F(j-\lambda\delta)$ by the lemma. In either case the left-hand side is true.

Assume the left-hand side is true and let x_0 be such that $F(x_0)$ is true.

Suppose x_0 is of the form b_i+j for some b_i and for some j with $1\leqslant j\leqslant\delta$. Then one of the disjuncts of the second term on the right-hand side is true, hence the right-hand side is true.

Suppose x_0 is not of that form, consider $F(x_0-\delta)$. If $F(x_0-\delta)$ is false, but $F(x_0)$ is true, then at least one basic relation in F of one of the four types must change from true to false as x_0 is changed to $x_0-\delta$ (remember that F does not involve the negation operator). But this is clearly impossible for relations of types (A), (C) and (D), and moreover could only happen in a relation of type (B) if $b_i<x_0$ and $\neg b_i<x_0-\delta$. This latter implies x_0 is of the form b_i+j with $1\leqslant j\leqslant\delta$, contrary to hypothesis for this case.

We can therefore assume $F(x_0-\delta)$ is true. This argument can be repeated with $x_0-\delta$ replacing x_0 until either we find an x of the form b_i+j (making one of the second set of disjuncts true) or until we have an x so small that $F(x)\equiv F_{-\infty}(x)$. By adding a multiple of δ we will prove one of the first set of disjuncts true.

The theorem is now proved and may be used in order to eliminate the quantifier.

PRACTICAL REMARKS ON THE ALGORITHM

If arithmetic relations other than $<$ occur in the formula it is not necessary to formally eliminate them. The only point of interest in applying the theorem is to determine what 'b_i' would occur if the elimination were to be performed. Similarly it is not necessary to eliminate negation: for each relation we determine how many negations control the relation, and if this is an odd number consider the negation of the relation in determining the b_i. Further, instead of decreasing x by δ in the proof, we could increase it, thus obtaining the theorem

$$(\exists x)F(x)\equiv\bigvee_{j=1}^{\delta}F_{\infty}(-j)\vee\bigvee_{j=1}^{\delta}\bigvee_{a_i}F(a_i-j)\quad,$$

where $F_{\infty}(x)$ is $F(x)$ with *false* substituted for formulas of type (A) and *true* for formulas of type (B). Clearly, if there are less 'a_i' than 'b_i' one should use this second approach. These remarks can be summed up as in table 1. The 'A set' and the 'B set' are merely the 'a_i' and the 'b_i' from the formulas of type (A) and type (B), respectively.

Table 1

		positive		*negative*	
	relation	A set	B set	A set	B set
1	$x<\alpha$	α	–	–	$\alpha-1$
2	$\alpha<x$	–	α	$\alpha+1$	–
3	$x\leqslant\alpha$	$\alpha+1$	–	–	α
4	$\alpha\leqslant x$	–	$\alpha-1$	α	–
5	$x=\alpha$	$\alpha+1$	$\alpha-1$	α	α
6	$x\neq\alpha$	α	α	$\alpha+1$	$\alpha-1$
7	$\delta\mid x+\alpha$	–	–	–	–
8	$\delta\nmid x+\alpha$	–	–	–	–

The enlarged algorithm to eliminate the existential quantifier from $(\exists x)F$ becomes

1. Perform Steps 3 and 4 of the old algorithm.
2. For each relation occurring in the new F determine whether it occurs positively or negatively (i.e. whether if the formula were to be put into disjunctive normal form it would occur with a negation sign). Add the appropriate terms to the A set and the B set as indicated in table 1, a dash indicates no addition has to be made.
3. Depending on whether the A set or the B set is smaller use the appropriate formula:

$$(\exists x)F(x)\equiv\bigvee_{j=1}^{\delta} F_{-\infty}(j)\vee\bigvee_{j=1}^{\delta}\bigvee_{\beta\in B}F(\beta+j)$$

or

$$(\exists x)F(x)\equiv\bigvee_{j=1}^{\delta} F_{\infty}(-j)\vee\bigvee_{j=1}^{\delta}\bigvee_{\alpha\in A}F(\alpha-j)\quad .$$

If the arithmetic relations are not eliminated the definitions of $F_{-\infty}$ and F_{∞} must be extended in the obvious way – $F_{-\infty}$ is obtained from F by substituting *true*, *false*, *true*, *false*, *false*, *true* for formulas of type 1 to 6 in table 1, F_{∞} is obtained from F by substituting *false*, *true*, *false*, *true*, *false*, *true* for formulas of type 1 to 6.

Two further remarks. Whilst it can be disastrous to transform the formula into disjunctive normal form, yet it is advantageous to distribute existential quantifiers over disjuncts wherever possible, as this can lead to sub-problems with smaller values of δ and smaller sizes of A sets and B sets. Also, advantage should be taken of equalities by substituting $x=\alpha\wedge P(\alpha)$ for $x=\alpha\wedge P(x)$ wherever it occurs, assuming x is the bound variable being eliminated.

THE δ EXPANSION

The effect of this can be considerably lessened, at least in an important subset of formulas. Assume that the closed formula being tested for validity is in prenex normal form and all the quantifiers are the same, either all existential or all universal. This includes the interesting case of a formula with free variables but no quantifiers, as its closure will have all universal quantifiers at the front. If the quantifiers are universal, the transformation to existential quantifiers will produce a string of existential quantifiers (preceded by a negation), and so in either case the quantifiers have to be eliminated from a formula of the type

$$(\exists x_1)(\exists x_2)\ldots(\exists x_n)F(x_1,x_2,\ldots,x_n)\quad .$$

The steps in the algorithm depend only on the form of the basic relations (assuming negations have been eliminated) and not at all on the form of F. Let us therefore rewrite the formula in the form

$$(\exists x_1)(\exists x_2)\ldots(\exists x_n)F(R_1,R_2,\ldots,R_m)\quad ,$$

where $R_1, \ldots, R_m$ are arithmetical relations involving the variables $x_1, \ldots, x_n$ and F contains no negations. Each elimination produces an operator of the form $\bigvee_{i=1}^{\delta}$, and as this commutes with the existential quantifier we may carry on the procedure without formally expanding this disjunct by merely preserving i as a variable. Let us illustrate this process on a simple example with $n=2$ and $m=3$. *True* and *false* will be denoted by t and f. Starting with:

$$(\exists y)(\exists x)F(x+5y>1,\ 13x-y>1,\ x+2<0) \quad ,$$

replace $13x$ by x:

$$(\exists y)(\exists x)[F(x>13-65y,\ x>y+1,\ x<-26)\wedge 13\,|\,x] \quad .$$

Use the A set to eliminate x:

$$\bigvee_{i=1}^{13}(\exists y)[F(t,t,f)\wedge 13\,|\,-i] \quad \vee$$

$$\bigvee_{i=1}^{13}(\exists y)[F(65y>39+i,\ y<-27-i,\ i>0)\wedge 13\,|\,-26-i] \quad .$$

Replace $65y$ by y in the second disjunct:

$$\bigvee_{i=1}^{13}(\exists y)[F(t,t,f)\wedge 13|-i] \quad \vee$$

$$\bigvee_{i=1}^{13}(\exists y)[F(y>39+i,\ y<-1755-65i,\ i>0)\wedge 13\,|\,-26-i\wedge 65\,|\,y].$$

Trivially eliminate the quantifier from the first disjunct and use the B set in the second disjunct to obtain finally:

$$\bigvee_{i=1}^{13}[F(t,t,f)\wedge 13\,|\,-i] \quad \vee$$

$$\bigvee_{i=1}^{13}\bigvee_{j=1}^{65}[F(f,t,i>0)\wedge 13\,|\,-26-i\wedge 65\,|\,j] \quad \vee$$

$$\bigvee_{i=1}^{13}\bigvee_{j=1}^{65}[F(j>0,66i+j<-1794,i>0)\wedge 13\,|-26-i\wedge 65\,|\,39+i+j].$$

This process is quite general and, starting with

$$(\exists x_1)(\exists x_2)\ldots(\exists x_n)F(R_1, R_2, \ldots, R_m) \quad ,$$

we shall obtain the disjunction of a number of terms each of which is of the form:

$$\bigvee_{i_1=1}^{\delta_1}\bigvee_{i_2=1}^{\delta_2}\cdots\bigvee_{i_n=1}^{\delta_n}[F(S_1, S_2, \ldots, S_m)\wedge \lambda_1\,|\,a_1\wedge\lambda_2\,|\,a_2\wedge\ \ldots\ \wedge\lambda_n\,|\,a_n] \quad ,$$

where $S_1, \ldots, S_m$ are arithmetical relations, $a_1, \ldots, a_n$ are algebraic expressions, and $\lambda_1, \ldots, \lambda_n$ are positive integers – all relations and expressions being within Presburger arithmetic and involving only integers and the variables $i_1, \ldots, i_n$.

This process only involves a moderate expansion which depends on the size of the A and B sets.

In order to evaluate each of these logical expressions it is not necessary to expand the δ disjuncts. In the appendix we show how to find the sets of values of $i_1, \ldots, i_n$ which satisfy the expression $\lambda_1 | a_1 \wedge \ldots \wedge \lambda_n | a_n$; the expansion is then only done over these sets.

Applying this process to the previous examples (in this simple case the answer can be seen immediately) we get our final form:

$$F(t, t, f) \vee F(f, t, t) \vee F(t, f, t) \quad .$$

The original expression is therefore equivalent to this one, only assuming F contains no negations; moreover no drastic expansions occur in obtaining this result. Further simplification is often possible, although not in this example. As F contains no negations we have

$$F(\ldots, f, \ldots) \rightarrow F(\ldots, t, \ldots) \quad ,$$

and therefore, for example,

$$F(f, f, t) \vee F(f, t, t) \quad ,$$

can be replaced by

$$F(f, t, t) \quad .$$

In particular, if $F(t, t, t)$ ever occurs, the process may be stopped with $F(t, t, t)$ as the answer.

If the prenex normal form contains mixed quantifiers then the process is more complicated as, for example, $(\forall y)(\exists x)F$ will become, after elimination of x and transformation of the first quantifier, the disjunct of terms of the form $\neg(\exists y) \bigwedge_{i=1}^{\delta} \neg F$.

There is now no commuting of operators and the A and B sets will in general contain a number of members with i as a variable thus making the size of the sets depend on δ. Hand calculation on simple examples suggests that the method may well still work, but as yet we do not have a simple way to express the algorithm.

CONCLUDING REMARKS

One way to compare the new algorithm with the old is to regard the original transformation to disjunctive normal form method as first finding which sets of truth values of the relations make F true and then testing these required sets of values to determine if they are consistent; on the other hand, the new transformation first finds by direct calculation which sets of truth values for the relations are consistent, and then evaluates F with these consistent values to see if any make F itself true. With examples of the kind which occurred in Cooper (1971) the second method proves much more efficient, as well as being simpler to state and understand. Moreover, the independence of the process on the form of F could lead to an advantage if we had examples with the same relations but different F's, as the above process need only be performed once.

APPENDIX

We wish to find the sets of values of $i_1, \ldots, i_n$ within certain bounds which make true:

$$\bigwedge_{j=1}^{n} \lambda_j \,\Big|\, \sum_{k=1}^{n} \mu_{jk} i_k + \nu_j \quad ,$$

where μ_{jk}, ν_j are integers and λ_j are positive integers. This can be done by a method similar to Gaussian elimination using the two easily proved theorems from the theory of numbers:

Theorem 1

$m \mid ax+b \wedge n \mid \alpha x+\beta$ if and only if
$mn \mid dx+bpn+\beta qm \wedge d \mid \alpha b - a\beta$
where $d = GCD(an, \alpha m)$ and $pan + q\alpha m = d$

(p and q are found as a by-product if Euclid's algorithm is used to find the GCD).

Theorem 2

$m \mid ax+b$ has solutions for x if and only if
$d \mid b$, the solutions are $x = -p(b/d) + t(m/d)$ for all integers t
where $d = GCD(a, m)$ and $d = pa + qm$.

The sets of solutions are found by first using Theorem 1 to reduce the set of divisibility relations to triangular form and then using Theorem 2 and the known bounds on the variables to obtain all the solutions by a back substitution process.

REFERENCES

Cooper, D.C. (1971) Programs for Mechanical Program Verification. *Machine Intelligence* 6, pp. 43–59 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

Cooper, W.S. (1964) Fact Retrieval and Deductive Question-answering information retrieval systems. *J. Ass. Comput. Mach.*, **11**, pp. 117–37.

Hilbert, D. & Bernays, P. (1968) *Grundlagen der Mathematik I* (Zweite Auflage) pp. 366–75. Berlin, Heidelberg, New York: Springer-Verlag.

Kreisel, G. & Krivine, J. L. (1967). *Elements of Mathematical Logic*, pp. 54–7. Amsterdam: North-Holland.

McCarthy, J. (1958) Programs with common sense. *Mechanization of Thought Processes vol. I*, pp. 77–84. Proc. Symp. Nat. Phys. Lab., London.

Paterson, M. (1972) Decision problems in computational models. *Proc. ACM conference on proving assertions about programs*, pp. 74–82. Los Cruces, New Mexico.

Presburger, M. (1929) Uber die Vollstandigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. *Comptes-Rendus du I Congres de Mathematiciens des pays Slaves*, pp. 92–101, 395. Warsaw.