MEMORANDUM
RM-4319-PR
MARCH 1965

ALGEBRAIC TECHNIQUES AND THE MECHANIZATION OF NUMBER THEORY

Stephen A. Cook

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM
RM-4319-PR
MARCH 1965

ALGEBRAIC TECHNIQUES AND THE MECHANIZATION OF NUMBER THEORY

Stephen A. Cook

This research is sponsored by the United States Air Force under Project RAND—Contract No. AF 49(638)-700 monitored by the Directorate of 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.

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90406

Published by The RAND Corporation

PREFACE

Part of the Project RAND research program consists of basic supporting studies in mathematics. This Memorandum shows how some simple algebraic peculiarities of the branch of mathematics known as number theory might be incorporated into a machine for generating proofs in number theory.

SUMMARY

A number of mechanical procedures for generating proofs of theorems of logic have appeared in the literature, all based on the Skolem—Herbrand theorem. Although in principle the procedures can be applied to find proofs of mathematical theorems, the procedures are very inefficient for this purpose because they are designed to operate on too large a domain. This report outlines a general method by which these procedures can be adapted to operate more efficiently on a much more restricted domain: the statements of elementary number theory.

The method is given a theoretical justification, and several specific algorithms are described for realizing the general method. These algorithms are applied to examples. Much more detailed study must be done before the method can be successfully utilized by a computer to generate proofs of difficult number—theoretic theorems.

CONTENTS

ALGEBRAIC TECHNIQUES AND THE MECHANIZATION OF NUMBER THEORY

1. INTRODUCTION

There has been much interest recently in the problem of devising mechanical techniques for generating proofs of theorems. Although ultimately one would like to find feasible procedures for deciding interesting mathematical statements, thus far most of the work has been confined to theorem-proving in pure quantification theory (see [4] for a notable exception). In principle, techniques developed for proving quantificational theorems apply to proving mathematical theorems, since the truth of a mathematical statement S is always equivalent to the validity of a quantificational formula $(A_1 \ \& \ A_2 \ \& \ \ldots \ \& \ A_n) \rightarrow S$, where $A_1, A_2, \ldots, A_n$ are suitable axioms. It is probable, however, that methods so general that they apply to all quantificational formulas will not be nearly efficient enough to prove interesting mathematical theorems. On the contrary, a procedure capable of deciding a large class of interesting mathematical statements will have to be deeply involved with the peculiarities of the relevant branches of mathematics, as well as cognizant of logical deduction. It is the purpose of this paper to show how some simple algebraic peculiarities of number theory might be incorporated in a device to prove theorems in number theory.

We choose to concentrate on number theory because it is an example of a branch of mathematics in which

many interesting statements have simple formulations. In addition, the class of true statements of number theory is highly undecidable (in fact, not even arithmetical). This theoretical result is driven closer to home by the commonplace observation of the many simple open questions in number theory—and closer yet when we notice that there have been many remarks in the literature expressing interest in mechanizing number theory, but as yet not even a convincing description of a possible number-theory machine comparable to the several logic machines or to the Gelernter-Hansen-Loveland geometry machine. This paper does not present such a description, but hopefully brings such a description closer at hand.

Before presenting our approach, we take note of two papers which comment on mechanizing number theory. In [5], Machover and Robinson give a general description of a proof procedure and two examples of proofs which their procedure might generate. Their point of view differs from ours in that they outline a general procedure that is supposed to prove the standard results of elementary number theory, given a list of these results in the right order. (Of course, earlier results may be used in proving later ones.) Our thesis is that if we are to have any success at all, the basic results cannot simply be made available as a list of theorems, but rather they must be thoroughly built into the procedure. For example, if the problem at hand involves verifying a polynomial identity, obviously

no general heuristic device that must select the proper instances of the properties of associativity, commutativity, distributivity, and substitutivity of identity is going to be nearly as effective as the simple algorithms available for checking polynomial identities.

In the second paper [3], Davis and Putnam present a finitely axiomatizable system for number theory, and remark that such a system is necessary for a mechanical theorem prover. But surely one can replace the usual infinite set of induction axioms by a single rule of inference, which a machine can handle as easily as the Putnam-Davis set-theoretic induction axiom. Thus, the plethora of additional symbols necessary for the finite axiomatization seems an unnecessary complication from the point of view of efficient mechanization.

2. THE FORMAL SYSTEM

Our theorem-prover will be cast roughly along the lines of those in [1], [2], [7], [10], [11], [12], and others. Davis [1] provides a well-written elementary description of the general technique. Briefly, if B is the theorem to be proved, then a selection of axioms and relevant theorems A_1, A_2, ..., A_n is made, and the conjunction A_1 & A_2 & ... & A_n & $\neg$ B is put in functional form (i.e., existential variables are replaced by functions of the universal variables governing them). The result is put in conjunctive form, and a list L_1, L_2, ..., L_m of the conjuncts is made, where each L_i is a primary formula (i.e., a disjunction of atomic formulas and negations of atomic formulas). The object is to find a set of substitution instances of the L_i which is truth-functionally inconsistent. The terms used in the substitutions are taken from the Herbrand universe, which consists of all terms built up of the constant symbols and function symbols of the L_i. The Herbrand theorem guarantees that an inconsistent conjunction exists if and only if B is quantificationally implied by A_1, ..., A_n.

Before discussing our proposed modifications of this method, it will be convenient to introduce a slightly non-standard formal system for number theory. The principal unique feature of the system is simply that ordinary polynomial expressions such as xyz + 3 + xz are well-formed

as they stand, without parentheses being required to indicate a particular association. In fact, we shall not allow such parentheses. Thus there is no need for the axioms of associativity for addition and multiplication; indeed, these axioms cannot even be expressed in the system. The reason we take the trouble to formulate the system explicitly is that we wish to note the necessary modifications of a few metamathematical concepts.

The intended range of variables of the system is all integers—positive, negative, and zero. Multiplication is indicated by juxtaposition. The various special function and predicate symbols can stand for such things as the greatest common divisor function, x divides y, etc. The specific interpretation will depend on the context.

The symbols of the system are the following: an infinite ordered list of variables (usually denoted by x, y, z, etc.), an infinite ordered list of special function symbols (f, g, h, ...), an infinite ordered list of special predicate symbols (P, Q, R, ...), the numerals 0, 1, −1, 2, −2, ..., the symbols +, <, =, &, $\vee$, $\neg$, E, comma.

Terms are defined inductively as follows: (i) Sums, products, numerals, variables, and special terms are terms. (ii) Function symbols are special terms. (iii) If $t_1, \ldots, t_n$ $(n \geq 1)$ are terms and f is a function symbol, then $f(t_1, \ldots, t_n)$ is a special term. (iv) If $t_1, \ldots, t_n$ $(n \geq 2)$ are terms none of which is a sum, then $(t_1 + t_2 + \ldots + t_n)$ is a sum. (v) If $t_1, \ldots, t_n$ $(n \geq 2)$

are terms, then $t_1t_2\cdots t_n$ is a product. (vi) Nothing is a sum, product, special term, or term, except as required by (i) - (v).

For example, $(2x + 3 + 5(f((x + y), y)zz + 3))$ is a term, but $(xy)z$ and $((x + y) + z)$ are not.

Atomic formulas are of one of the forms $P(t_1, \ldots, t_n)$ $(n \geq 1)$, $t_1 = t_2$, $t_1 < t_2$, where the t_i are terms and P is a special predicate symbol.

Formulas are defined inductively as follows (all clauses have the condition that no function symbol can appear twice in a formula with different numbers of arguments): (i) Atomic formulas, disjunctions, and conjunctions are formulas. (ii) If $B_1, \ldots, B_n$ $(n \geq 2)$ are formulas, none of which is a disjunction, then $(B_1 \vee B_2 \vee \ldots \vee B_n)$ is a disjunction. (iii) If $B_1, \ldots, B_n (n \geq 2)$ are formulas, none of which is a conjunction, then $(B_1 \,\&\, B_2 \,\&\, \ldots \,\&\, B_n)$ is a conjunction. (iv) If B is a formula and x is a variable, then $\neg B$, $(x)B$, $(Ex)B$ are formulas. (v) Nothing is a formula, conjunction, or disjunction except as required by (i) - (iv).

If a function symbol occurs with no arguments, we shall call it a constant (symbol). Constants will be denoted by a,b,c, etc. We shall use the standard notation for exponents and take other liberties with writing formulas when convenient. The notions of scope and of free and bound variables are defined as usual.

A formula is in rational form if no quantifier is in the scope of an occurrence of $\neg$, and if no two quantifiers operate on different occurrences of the same variable. The functional form fn(B) of B is the formula obtained from B as follows: (1) Put B in rational form (using the standard logical equivalences). (2) If y is an existentially quantified variable, replace each occurrence of y by $f(x_1, \ldots, x_n)$, where f is a new function symbol and $x_1, \ldots, x_n$ are all the universally quantified variables which include (Ey) in the scopes of their quantifiers. If (Ey) is in the scope of no universal quantifier, then y is replaced simply by a new constant symbol a. Distinct existential variables should be replaced by distinct function symbols. (3) Delete existential quantifiers.

This definition of functional form differs in clause (2) from the one given by Davis [1]. Davis includes as arguments for $f(x_1, \ldots, x_n)$ all universal variables which precede (Ey). We include only those whose quantifiers govern (Ey). As is well-known, the Herbrand theorem (Theorem 1) remains valid when the definition given in the preceding paragraph is adopted; in fact, the number of substitution instances necessary to find an inconsistent conjunction is in general reduced.

Modifying Davis' definition [1] slightly, we define the Herbrand universe H(B) of a formula B to be the set of all terms t such that (i) no variables occur in t, and (ii) each function symbol occurring in t must occur in

fn(B) with the same number of arguments.

By a substitution instance of a formula B in rational form, we shall mean the result of substituting terms for the universally quantified variables of B and deleting universal quantifiers. The terms are from H(B), or from the Herbrand universe of the conjunction of all formulas under discussion at the moment. If the term t being substituted is a sum, and the variable x which t replaces is an "addend" of a sum, then the external parentheses of t must be deleted.

The notion of model* for a formula of our system coincides with the usual one, except that a model M with universe U for a set S of formulas (or a single formula B) must include not only a function for each special function symbol of S (or B), but also two additional functions $\oplus$ and $\cdot$, which map $U \times U \longrightarrow U$. We shall require that $+$ and $\cdot$ be associative; i.e. (writing $a \oplus b$ and $a \cdot b$ for $\oplus(a, b)$ and $\cdot(a, b)$, respectively), they must satisfy $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all a, b, c in U. The interpretation of terms must be such that if the terms $t_1, \ldots, t_n$ are assigned objects $\bar{t}_1, \ldots, \bar{t}_n$ in U, then $(t_1 + \ldots + t_n)$ is assigned $\bar{t}_1 \oplus \bar{t}_2 \oplus \ldots \oplus \bar{t}_n$ and $t_1 t_2 \ldots t_n$ is assigned

*The interpretation of = need not be the identity relation in a model.

$\bar{t}_1 \cdot \bar{t}_2 \cdot \ldots \cdot \bar{t}_n$ (under any association). As usual, we say a set S of formulas (or a single formula B) is consistent if S (or B) has a model.

We shall show that the relevant version of the Herbrand theorem holds in our system.

Theorem 1. A formula B is consistent (has a model) if and only if no (finite) conjunction of substitution instances of fn(B) is truth-functionally inconsistent.

Proof. If B has a model, then the standard argument (see, for example, Davis [1]) shows that each conjunction of substitution instances of fn(B) is consistent. Now suppose, conversely, that each conjunction of substitution instances from H(B) is consistent. Then, by the infinity lemma, there is an assignment $\mathcal{T}$ of truth values to every atomic formula of one of the forms $P(t_1, \ldots, t_n)$, $t_1 = t_2$, $t_1 < t_2$, where t_i is in H(B) and P is a predicate symbol occurring in B, which makes each substitution instance of fn(B) come out true. We construct a model M for B as follows: The universe of M is the set H(B). The interpretations of the predicate symbols (including $<$ and $=$) and of the special function symbols are determined by the truth assignment $\mathcal{T}$ in the obvious manner. The two functions $\oplus$ and $\cdot$ are defined by the equations $t_1 \oplus t_2 = (t_1^* + t_2^*)$, and $t_1 \cdot t_2 = t_1 t_2$, where t_1 and t_2 are in H(B), and t_i^* is the same as t_i, except if t_i is a sum, then t_i^* is t_i without the outside parentheses. The functions $\oplus$ and $\cdot$ thus defined are clearly associative. Hence M is completely

defined, and it is easily verified that M is a model for B.

We shall axiomatize our system by listing essentially the axioms for an ordered commutative ring with unity (briefly, an ordered domain), together with an axiom schema for the least number principle. Because we chose to include numerals for every integer in our system, the axioms must be cluttered with schemata defining the numerals. This is perhaps a formal difficulty, but certainly not a computational difficulty, since the defining properties of numerals are always built into computing machines.

We omit universal quantifiers in this list of axioms.

I. Axioms for equality.

I.1 $x = x$.

I.2 $\neg x = y \vee \neg z = y \vee z = x$.

I.3 $\neg x = y \vee (x + z) = (y + z)$.

I.4 $\neg x = y \vee xz = yz$.

I.5 $\neg x = y \vee \neg x < z \vee y < z$.

I.6 $\neg x = y \vee \neg z < x \vee z < y$.

I.7 Axiom schemata for substitution of equals in special function and predicate symbols.

II. Ring axioms.

II.1 $(x + y) = (y + x)$.

II.2 $xy = yx$.

II.3 $x(y + z) = (xy + xz)$.

II.4 $(x + 0) = x$.

II.5 $x1 = x$.

II.6 $(1 + -1) = 0$.

II.7 $\neg 0 = 1$.

III. Order axioms.

III.1 $0 < x \vee 0 = x \vee 0 < -1x$.

III.2 $\neg 0 < x \vee \neg 0 < -1x$.

III.3 $\neg 0 < x \vee \neg 0 < y \vee 0 < (x + y)$.

III.4 $\neg 0 < x \vee \neg 0 < y \vee 0 < xy$.

IV. Definitions of numerals.

IV.1 $\Delta_{n+1} = (\Delta_n + 1)$, where $n \geq 1$ and Δ_n is the numeral standing for n.

IV.2 $\Delta_{n-1} = (\Delta_n + -1)$, where $n \leq 1$, Δ_n as above.

V. Least-number principle.

V.1 $\neg(Ex)(B(x) \& 0 < x) \vee [(Ex)(B(x) \& 0 < x \& (y)(\neg B(y) \vee \neg 0 < y \vee x = y \vee x < y))]$,

where B(x) is a formula with the free variable x and no occurrences of y.

Every ordered domain is a model for axiom groups I, II, III (with I.7 deleted), and conversely, given a model of these axioms, we obtain an ordered domain by passing to the equivalence classes defined by the relation interpreting =. Addition and multiplication are associative in such models by our definition of model. Axiom 1.7 can be made to satisfy any ordered domain R simply by assigning trivial functions and predicates on R to the special function and predicate symbols; group IV is satisfied when numerals are given their standard interpretation in R.

Our system is essentially equivalent to standard systems for elementary number theory in the following

sense. Given a formula A in a standard system, we can change A into a formula B of our system by deleting excess parentheses and relativising quantifiers to nonnegative integers. Then A will be a theorem of the standard system if and only if B is a theorem of our system; that is, if and only if B is true in all models satisfying the axioms of our system. The proof of this statement is straightforward, and will not be presented here.

3. A CONNECTION WITH REAL CLOSED FIELDS

In this section we give a result which suggests a way of excluding from consideration all the axioms except the least-number principle when proving theorems. First, we must set up some terminology.

Group I' shall refer to axiom group I, with I.7 omitted. A closed formula B is valid for ordered domains if B is true in every model for axiom groups I', II, III, IV, no matter how the special function and predicate symbols in B are interpreted.

Lemma 1. If B is closed and has no special function or predicate symbols, then B is valid for ordered domains if and only if B is true in every ordered domain R.

Proof. By "B is true in R" we mean B is true when its variables are interpreted as ranging over R, and when +, juxtaposition, and numerals are given their natural interpretations in R. The lemma follows from the remark immediately following the axioms.

Now suppose C is a formula without quantifiers or variables. A valuation from C to an ordered domain R is a map φ from H(C) to R such that (i) the numerals 0, 1, -1, 2, -2, ... are sent to their natural images in R(i.e., $\varphi(\Delta_n) = n \cdot 1$), (ii) $\varphi((t_1 + t_2 + \ldots + t_n)) = \varphi(t_1) \oplus \varphi(t_2) \oplus \ldots \oplus \varphi(t_n)$, and (iii) $\varphi(t_1 t_2 \ldots t_n) = \varphi(t_1) \cdot \varphi(t_2) \cdot \ldots \cdot \varphi(t_n)$, where $\oplus$ and $\cdot$ denote addition and multiplication in R. If $\mathcal{T}$ is a truth-assignment to the atomic formulas of C, then the valuation

φ is compatible with $\mathcal{T}$, provided that formulas of the forms $t_1 < t_2$, $t_1 = t_2$ come out true under $\mathcal{T}$ if and only if $\varphi(t_1) \prec \varphi(t_2)$, and $\varphi(t_1) = \varphi(t_2)$, respectively, where $\prec$ is the order relation in R.

We shall say the formula C (without variables) is consistent for ordered domains if there is a valuation φ from C to an ordered domain such that φ is compatible with a truth-assignment making C true. For example, the formula $\neg\, f(1 + 1) = f(2)$ is consistent for ordered domains, while the formula $\neg\, 1 + 1 = 2$ is not.

Theorem 2. A closed formula B without special function or predicate symbols is valid for ordered domains if and only if some (finite) conjunction of substitution instances of fn($\neg$ B) is inconsistent for ordered domains.

Proof. Suppose B is not valid for ordered domains. Then by Lemma 1, B is not true in some ordered domain R; i.e., $\neg$ B is true in R. Hence, by use of the Skolem functions for $\neg$ B we can find a valuation φ from fn($\neg$ B) to R and a compatible truth assignment $\mathcal{T}$ such that (i) $\mathcal{T}$ assigns truth values to the atomic formulas composed of terms of H($\neg$ B) and predicate symbols from B, and (ii) $\mathcal{T}$ makes every conjunction of substitution instances of fn($\neg$ B) true. Hence, all such conjunctions are consistent for ordered domains. Suppose, conversely, that B is valid for ordered domains. Let $A_1, A_2, \ldots, A_n$ be the axioms in groups I', II, III, IV, except we include only those (finitely many) instances of IV which are necessary to

define the numerals occurring in B. Then B is true in every model for A_1 & A_2 & ... & A_n, so A_1 & A_2 & ... & A_n & $\neg$ B is (quantificationally) inconsistent. By Theorem 1, there are substitution instances S_1, S_2, ..., S_m of fn(A_1 & A_2 & ... & A_n & $\neg$ B) such that S_1 & S_2 & ... & S_m is truth-functionally inconsistent. We may write S_i in the form $S_i = X_i$ & Y_i, where X_i is a substitution instance of fn(A_1 & ... & A_n) and Y_i is a substitution instance of fn($\neg$ B). We claim that C = Y_1 & Y_2 & ... & Y_m is inconsistent for ordered domains. For suppose, to the contrary, that there is a valuation φ from C to an ordered domain R which is compatible with a truth-assignment $\mathcal{T}$, making C true. We can extend $\mathcal{T}$ to all formulas of one of the forms $t_1 < t_2$, $t_1 = t_2$, t_i in H(B) = H(C), by making $t_1 < t_2$, $t_1 = t_2$ true if and only if $\varphi(t_1) \prec \varphi(t_2)$ in R, $\varphi(t_1) = \varphi(t_2)$ in R, respectively. Thus extended, $\mathcal{T}$ makes each of the X_i true, since the X_i are instantiations of A_1 & A_2 & ... & A_n, and this conjunction contains no special function symbols and is universally true in R. Therefore, X_1 & X_2 &...& X_n & C (i.e., S_1 &...& S_m) is truth-functionally consistent, contrary to assumption. Thus C is indeed inconsistent for ordered domains. This completes the proof of Theorem 2.

Suppose C is a formula without variables or quantifiers. An occurrence of a special term in C is <u>maximal</u> if it is not a proper subterm of a special term. A <u>table</u>

for C is any 1-1 correspondence between variables and the special terms which have maximal occurrences in C. The existential form of C with table T, $E_T(C)$, is the formula obtained from C by replacing all maximal occurrences of special terms in C by the variables specified by T, and then adding a prefix to C consisting of an existential quantifier for each new variable. For example, an existential form of $a^2 + 1 + f(a + 3) = 0$ is $(Ex)(Ey)(x^2 + 1 + y = 0)$.

Lemma 2. Every ordered domain is a subring of a real closed field.

Proof. If R is an ordered domain, then the order relation can be extended to the fraction field K of R by the rule $a/b > 0$ if and only if $ab > 0$. But according to van der Waerden [9], Sec. 72, K can be embedded in a real closed field.

Theorem 3. If C has no quantifiers, variables, or special predicate symbols, then C is consistent for ordered domains if and only if $E_T(C)$ is true in the real numbers.

Proof. Suppose C is consistent for ordered domains. Then C has a valuation φ into an ordered domain R which is compatible with a truth-assignment $\mathcal{T}$ under which C comes out true. $E_T(C)$ is true in R, since φ and T tell how to assign members of R to the variables in $E_T(C)$ in order to satisfy the existential quantifiers (notice that $E_T(C)$ has no special predicate or function symbols). Now, by Lemma 2 we may assume R is a real closed field. Hence

$E_T(C)$ is true in the real numbers, since Tarski's decision procedure for real closed fields [8] shows that an elementary formula is true of a real closed field if and only if it is true of the reals.

Suppose, conversely, that $E_T(C)$ is true in the real numbers. Then there is an assignment of a real number to each variable of $E_T(C)$ which makes $E_T(C)$ come out true, and φ composed with T tells how to transfer the assignment to certain special terms in H(C) (notably those special terms which have maximal occurrences in C). This assignment can be extended to a full valuation φ of C into the reals by setting $\varphi(t) = 0$ for each special term t not "appearing" in T, and then extending to H(C) by the rules $\varphi(\Delta_n) = n$, $\varphi((t_1 + t_2 + \ldots + t_n)) = \varphi(t_1) + \ldots + \varphi(t_n)$, and $\varphi(t_1 \cdots t_n) = \varphi(t_1) \cdot \varphi(t_2) \cdot \ldots \cdot \varphi(t_n)$. Thus φ becomes the required valuation of C into an ordered domain (the real numbers). The truth-assignment $\mathcal{A}$ compatible with φ is the one determined by the ordering and identity relations on the real numbers. C is true under $\mathcal{A}$ precisely because $E_T(C)$ is true in the reals.

Combining Theorems 2 and 3, we obtain the following theorem.

Theorem 4. If B has no special function or predicate symbols, then B is valid for ordered domains if and only if the existential form of some finite conjunction of substitution instances of fn($\neg$ B) is false in the real numbers.

Theorem 4 is relevant to mechanizing number theory because Tarski's decision procedure can, at least in principle, be applied to decide whether the existential form of a substitution instance is true or false in the real numbers. Hence a possible procedure for proving a formula B in the integers is as follows. First, select relevant theorems and instances of the least number principle $A_1, \ldots, A_n$, as before, but do not include the axioms in groups I', II, III, IV. Set $A = fn(A_1 \,\&\, A_2 \,\&\, \ldots \,\&\, A_n \,\&\, \neg B)$. If a proper selection of the A_i's has been made and B is a theorem in our system (i.e., B follows from axiom groups I' – V), then $\neg (A_1 \,\&\, A_2 \,\&\, \ldots \,\&\, A_n) \vee B$ will be valid for ordered domains, and so by Theorem 4 the existential form of some conjunction of substitution instances of A is false in the reals. Thus we may proceed systematically (or heuristically) to form substitution instances of A and to test the existential form of their conjunctions from time to time, using the Tarski procedure.

We do not claim that the above method is efficient or even feasible. The Tarski decision procedure is very complex and difficult to mechanize. However, the alternative of trying to show that a formula is algebraically inconsistent by finding truth-functionally inconsistent instances of the formula and axioms I', II, III, IV (and perhaps some algebraic theorems) is hopelessly inefficient for a theorem-prover that is to have much algebraic sophistication. The unconvinced reader might find it instructive to exhibit explicitly a

conjunction of substitution instances of the axioms which is truth-functionally inconsistent with the formula

$$\neg\ (a + 1)(a + 1) = (aa + 2a + 1).$$

The Tarski decision method required deep insight into the structure of real-closed fields for its formulation, while the method of finding inconsistent substitution instances is so general that it works for systems with no relation even to algebraic rings. Hence, on these general grounds it is highly plausible that the former method is far more efficient at deciding statements about real numbers than the latter, even if the class of statements we are interested in is rather special (namely, the existential forms of conjunctions of subsitution instances of formulas of the form A given above).

In a practical number-theory machine, the algebraic procedure used will probably be more closely related to the integers and be substantially less general than Tarski's procedure. One possibility, for example, is to check the conjunction of substitution instances for consistency in additive number theory rather than for ordered domains. That is, all polynomial terms could be completely multiplied out and simplified, and the resulting formulas put in an existential form similar to the one described above, except multiplication (i.e., juxtaposition) would be regarded as a special function symbol. The Presburger decision procedure [6] for additive number theory would be applied to the result.

Experience has shown that for theorems on the level of difficulty of those in the first chapter of an elementary textbook in number theory, relatively simple algebraic methods suffice. Examples are given in the next section.

In concluding this section, we remark that theorems 1 – 4 can be formulated for much more general objects than ordered domains, and the objects can include special function and predicate symbols in their axiomatizations. The main requirement is that the prenex form of the axioms contain only universal quantifiers.

4. EXAMPLES

In this section we discuss three examples of elementary number-theoretic theorems and how a machine might construct proofs of them. We assume in each case that preliminary steps have been taken so that the machine has at its disposal a list of primary formulas (that is, disjunctions of atomic formulas and negations of atomic formulas), the conjunction of which is to be proved inconsistent in the integers. The conjuncts will include the functional forms of the negation of the statement in question, as well as the functional forms of various theorems that seem to be relevant. The general technique will be to substitute terms for universal variables in the various conjuncts in some order, while at the same time simplifying the resulting expressions by truth-functional manipulation and application of algebraic (and simple number-theoretic) principles.

It will be convenient to define a few notions for the discussion. We shall say two terms t_1 and t_2 are <u>equivalent</u> if they represent the same polynomial; that is, if t_1^* and t_2^* are obtained from t_1 and t_2 by replacing maximal special subterms (see the previous section) by variables in a 1-1 fashion, then t_1 and t_2 are equivalent if and only if

$$(x_1)(x_2)\ldots(x_n)(t_1^* = t_2^*)$$ [redacted]

is true in the integers, where $x_1, x_2, \ldots, x_n$ are the variables occurring in t_1^* and t_2^*. A term is in <u>canonical form</u> if it is written as

a polynomial in some specified standard fashion. This means, roughly speaking, that multiplication has been distributed over addition as much as possible, like products have been combined and simplified, and subterms appear in some standard order as much as possible. The exact specification of canonical form is not important, as long as the following property is satisfied: Every term is equivalent to one and only one term in canonical form. It is not difficult to write a computer program to put terms in canonical form.

The notion of canonical form is useful because it enables us to restrict somewhat the number of terms used in forming substitution instances. For example, evidently Th. 4 still holds if the substitution instances referred to are limited to those formed by substituting only terms in canonical form.

The first example we present is the theorem that every integer greater than one has a prime divisor. The basic ideas of our discussion here were communicated to the author by Professor Hao Wang. In this example, and indeed for most theorems in number theory, it is convenient to use two special predicates: $P(x)$ for x is a prime, and $x \mid y$ for x is a divisor of y. We shall assume that initially the machine is given the following list of conjuncts, which it must find inconsistent:

(1) $1 < a$,

(2) $\neg P(x) \vee \neg x \mid a$,

(3) $\neg 1 < x \vee P(f(x)) \vee \neg x < a$,

(4) $\neg 1 < x \vee f(x) \mid x \vee \neg x < a$.

Formulas (3) and (4) are the result of one application of the least number principle (LNP) to the formula to be proved. The function f(x) is a "Skolem function" resulting from an existential quantifier. Formulas (3) and (4) are more intelligible when written in the form $(1 < x < a) \rightarrow (P(f(x)) \,\&\, f(x) \mid x)$. One application of the LNP or the standard induction axiom to the formula in question is often sufficient for elementary theorems, provided enough number-theoretic devices are built into the machine.

This example seems to refute our thesis that algebra should be built into the machine, since it requires no algebraic facility at all to prove the theorem. This is not surprising, however, since neither + nor · occurs in (1) – (4). When those operations do occur, algebraic facility is required, as the other examples will show. In general, it seems reasonable to admit for substitution only those terms composed of 0, 1, and the function symbols actually occurring in the formulas to be refuted, i.e., (1) – (4). In this case, the admissible terms are thus 0, 1, a, f(0), f(1), f(a), f(f(0)),..., and this is a reasonable order in which to substitute them.

The machine shall proceed as follows. The formulas with (universal) variables are selected sequentially, and the first term in the above list of admissible terms which has not yet been substituted in the selected formula is substituted. Any duplicate disjuncts in the result are eliminated, and any disjuncts to which an application of "cut" applies (with other conjuncts of the list) are eliminated. (The rule of cut allows an atomic formula to be eliminated if its negation appears as a conjunct.) The result of these operations is added to the list of conjuncts, and the augmented list is checked for truth-functional consistency. (An efficient method for checking consistency of formulas in conjunctive form is given in Davis and Putnam [2].) If the new conjunct added is in one of the forms $P(t)$, $\neg P(t)$, then the proper instance of the definition of P, or the negation of the definition, is added as a conjunct. Finally, we assume the machine recognizes that $|$ is reflexive and transitive. We list below the sequence of those conjuncts which contribute to the final contradiction.

(5) $\neg P(a)$ (a for x in (2) and recognition that $a \mid a$ is true).

(6) $b \mid a$ ⎫
(7) $1 < b$ ⎬ Definition of $P(a)$.
(8) $b < a$ ⎭

(9) $P(f(b))$ (b for x in (3) and cut with (7), (8)).

(10) $f(b) \mid b$ (b for x in (4) and cut with (7), (8)).

(11) f(b) | a ((6), (10), and transitivity of |).

(12) $\neg P(f(b))$ (f(b) for x in (2) and recognition that f(b) | f(b) is true).

We see that (9) and (12) form the contradiction.

We note that in the process of exhibiting the definition of $\neg P(a)$, the machine generates a new constant symbol b. This symbol b is fair game for substitutions, and might be placed on the list of terms a few notches after the set of terms needed to generate $\neg P(a)$. Hence, the actual sequence of terms used for substitution might be 0, 1, a, f(0), f(1), f(a), b, f(f(0)), f(f(1)), f(b), with perhaps a few extra terms resulting from exhibiting definitions of $\neg P(t)$ for certain terms t. Since there are only three formulas with variables, i.e., (2), (3), and (4), and they have one variable each, the number of conjuncts generated will be slightly more than three times the number of terms substituted, perhaps about 100. A computer should not have much difficulty checking for consistency 100 conjuncts of the present particularly simple form, using the Davis-Putnam method, because the first step in this method is to eliminate all atomic formulas which do not have both positive and negative occurrences. A little experimentation indicates that very few atomic formulas would survive this initial elimination.

Our next example is the Euclidean Algorithm. Here it is profitable to apply the ordinary induction axiom to the

formula in question, rather than the LNP. Thus the machine is given initially the conjuncts

(1E) $a = qb + r$ (a, b, q, r are constant symbols),

(2E) $0 = r \vee 0 < r$,

(3E) $r < b$,

(4E) $\neg a + 1 = xb + y \vee y < 0 \vee \neg y < b$.

The machine must have some algebraic facility to prove (1) – (4) are inconsistent, and as a start we shall assume the machine places every term in canonical form (see the beginning of this section) immediately upon making every substitution. Since + and · occur explicitly in (1E) – (4E), the set of terms substituted must be expanded to include sums and products. Hence the sequence of terms eligible for substitution will be 0, 1, –1, a, b, q, r, 2, –2, $a + 1$, $b + 1$, $q + 1$, $r + 1$, ab, aq, ... The position of a term in the sequence is determined roughly by the number of symbols composing it, except numerals should be counted for their absolute value and products counted as if the symbol · were there explicitly. Only terms in canonical form should appear in the sequence.

With this apparatus, the Presburger decision procedure for additive number theory (which also takes care of $<$) more than suffices to prove (1E) – (4E) inconsistent. Indeed, the conjuncts

(5) $\neg a + 1 = qb + r + 1 \vee r + 1 < 0 \vee \neg r + 1 < b$

((q, r + 1) for (x,y) in (4)),

(6) $\neg a + 1 = qb + b \vee 0 < 0 \vee \neg 0 < b$

((q + 1, 0) for (x,y) in (4)),

together with (1E), (2E), (3E) are inconsistent in the formal theory of integers which exludes multiplication as a function. In other words, if the only product (notably qb) is replaced by a new constant symbol, then there is no way of assigning integers to the constant symbols so that (1E), (2E), (3E), (5), (6) are simultaneously true. This fact is easily verified by making a few algebraic and truth-functional simplifications.

We remark that an essential application of the canonical form routine was made in forming (6); i.e., the Presburger procedure would be insufficient without this routine.

Although the Presburger procedure is much simpler than the Tarski procedure, it operates naturally on formulas in disjunctive form rather than conjunctive form, and it might be infeasible to apply repeatedly in its full generality. However, an easily mechanized special case which suffices for the present example can be built into the machine.

An atomic formula $t_1 = t_2$ is in <u>reduced form</u> if (i) t_2 is the numeral 0, (ii) t_1 is in canonical form, (iii) the left-most symbol of t_1 (excluding the left parenthesis) is not a numeral for a negative integer, and (iv) the "co-efficients" of the "summands" of t_1 are relatively prime. An atomic formula $t_1 < t_2$ is in <u>reduced form</u> if either

(i) – (iv) above are satisfied, or (i) – (iv) are satisfied with the roles of t_1 and t_2 reversed. It is easily verified that every atomic formula whose predicate is = or $<$ expresses the same algebraic relation as one and only one atomic formula in reduced form.

With this terminology, we can describe the procedure as follows. First, negated inequalities are replaced by a disjunction of two atomic formulas (e.g., $\neg a < b$ by $a = b \vee b < a$), and then all atomic formulas are put in reduced form. Thus (1E) – (4E) become

(1E)' $\quad qb + r - a = 0,$

(2E)' $\quad r = 0 \vee 0 < r,$

(3E)' $\quad 0 < b - r,$

(4E)' $\quad \neg xb + y - a - 1 = 0 \vee y < 0 \vee b - y = 0 \vee b - y < 0.$

Next, if one of the conjuncts is an equation in which a constant symbol c has precisely one occurrence, and that occurrence is a summand (with coefficient ± 1), then the equation is solved for c and the resulting expression substituted for c throughout, thus eliminating c from further consideration. (The original equation is deleted.) Again, all atomic formulas are put in reduced form.

Both a and r are eligible for this operation in (1E)'; it makes no difference which is chosen. If a is chosen, the result is

(1E)" $r = 0 \vee 0 < r$,

(2E)" $0 < b - r$,

(3E)" $\neg xb - qb + y - r - 1 = 0 \vee y < 0 \vee b - y = 0$
$b - y < 0$.

This step is repeated, if possible. It is not possible in the present case.

Now terms are substituted for variables, as before. In addition to truth-functional simplifications on the resulting conjuncts, the machine recognizes when a single formula or a pair of formulas with no variables is inconsistent in the theory of integers without multiplication. This recognition is easily accomplished. If an inconsistency is found, an application of cut is made, if possible. After every manipulation, atomic formulas are put in reduced form. If at any point a conjunct appears which enables a variable to be eliminated by the device mentioned above, the elimination is accomplished.

Here are the steps which lead to a solution of the present example, when this procedure is applied.

(4E)" $b - r - 1 = 0$ ($(q, r + 1)$ for (x,y) in (3E)".
The machine recognizes $\neg 0 = 0$ is false,
$r + 1 < 0$ is inconsistent with (1E)",
$b - r - 1 < 0$ is inconsistent with (2E)", so
these disjuncts are cut.)

Formula (4E)" can be used to eliminate either b or r from consideration. Either choice leads to success. If r is chosen, the list of relevant equations become

(1E)"' $b - 1 = 0 \vee 0 < b - 1$,

(2E)"' $0 < 1$ (redundant, and could be eliminated),

(3E)"' $\neg xb - qb + y - b = 0 \vee y < 0 \vee b - y = 0 \vee b - y < 0$.

Substituting $(q + 1, 0)$ for (x,y) in (3E)"' accomplishes the contradiction, since every disjunct in the result is either inconsistent by itself, or contradicts (1E)"'.

The procedure outlined above was of course closely tailored to fit the present example. However, the procedure can be generalized to include any fraction of the entire Presburger procedure. We have given this example to show how a few simple devices suffice to solve one interesting problem.

Referring back to our list of terms eligible for substitution, we notice that success depends on substituting the thirteenth term in the sequence. Since at any one time there is one equation with two variables available to be replaced, roughly $(13)^2 = 169$ substitution instances would be generated. Actually, 169 is too large, since the variables a and r are eliminated during the procedure, and thereafter the terms composing them would not be eligible for substitution. In any case, the number of conjuncts generated is uncomfortably large, because each new conjunct must be compared in a complicated fashion with all the old ones. Thus, a heuristic to reduce the number of conjuncts is desirable. One possibility is this: if a conjunct

consists of several disjuncts and it is not simplified and does not contribute to a simplification within a specified length of time, it should be eliminated from the list.

As a final example, we present a lemma used in proving the theorem that every positive integer is a sum of four squares. The lemma is that the product of two sums of four squares is again a sum of four squares. The machine need be given only the single equation

$$(1) \quad \neg(a^2 + b^2 + c^2 + d^2)(e^2 + f^2 + g^2 + h^2) = x^2 + y^2 + z^2 + w^2.$$

If the machine can find the right substitution instances for x,y,z,w, the only other facility needed is the ability to put an equation in reduced form and to recognize that $\neg 0 = 0$ is inconsistent. The proper substitutions are

$ae + bf + cg + dh$ for x,

$af - be + ch - dg$ for y,

$ag - ce + df - bh$ for z,

$ah - de + bg - cf$ for w.

This example should be borne in mind by all who think they know a good heuristic for finding substitutions!

5. CONCLUSION

By presenting the theorems in Sec. 3 and discussing a few examples, we have suggested that an algebraic facility could be built into a number-theory machine by a set of little algorithms, such as special cases of the Tarski and Presburger procedures. Of course, proving results in number theory depends not just on algebra, but on mastery of a large variety of special techniques for handling the standard number-theoretic concepts: prime number, division, congruence, etc. A successful machine will have to incorporate these techniques, and such a machine can be built only after detailed analysis of many examples has shown how the incorporation might be successfully accomplished.

REFERENCES

1. Davis, Martin, "Eliminating the Irrelevant From Mechanical Proofs," Proc. of Symp. in Applied Math., Vol. 15, American Mathematical Society, 1963.
2. Davis, Martin, and Hilary Putnam, "A Computing Procedure for Quantification Theory," Jour. Assoc. Comp. Math., 1960, pp. 201–215.
3. Davis, Martin, and Hilary Putnam, A Finitely Axiomatizable System for Elementary Number Theory, AFOSR Report No. TR 59–124, Oct. 1959.
4. Gelernter, H., J. R. Hansen, and D. W. Loveland, "Empirical Explorations of the Geometry Theorem Machine," Proc. of the Western Joint Comp. Conference, Vol. 17, 1960, pp. 143–149.
5. Machover, M., and A. Robinson, On the Mechanization of the Theory of Numbers, U. S. Office of Naval Research Report, September, 1962.
6. Presburger, M., "Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt," Comptes–rendus du I congres des Mathematicians des Pays Slaves, Worszawa, 1929, pp. 92–101, 395.
7. Robinson, J. A., "Theorem–Proving on the Computer," J. ACM, Vol. 10, 1963, pp. 163–174.
8. Tarski, A., and J. C. C. McKinsey, A Decision Method for Elementary Algebra and Geometry, The RAND Corporation, R–109, August, 1948 (revised May, 1951).
9. van der Waerden, B. L., Modern Algebra, Vol. 1, Ungar, revised English edition, 1953.
10. Wang, Hao, "Towards Mechanical Mathematics," IBM J. Res. Develop., Vol. 4, 1960, pp. 2–22.
11. Wang, Hao, "Proving Theorems by Pattern Recognition, I," I. Comm. ACM, Vol. 3, 1960, pp. 220–234.
12. Wang, Hao, "Proving Theorems by Pattern Recognition, II," Bell System Tech. J., Vol. 40, 1961, pp. 1–41.