CANONICAL INFERENCE

by

Dallas S. Lankford

December 1975 ATP-32

This work was supported in part by NSF Grant DCR74-12866

CANONICAL INFERENCE

by

Dallas S. Lankford

Department of Mathematics

Southwestern University

Georgetown, Texas 78626

December 1975

This work was supported in part by Grant #DCR74-12866.

ABSTRACT

We establish some new refutation completeness results for sets of rewrite rules in conjunction with resolution and paramodulation. All results of this paper deal with the case when none of the equations of an equality unsatisfiable set occur in non-unit clauses. When the set of reductions is complete we show that blocked resolution and immediate narrowing are refutation complete. We also show that special paramodulation, which is paramodulation into positions which are not variables, and resolution are refutation complete. Finally, we show that, in the presence of a suitable complexity measure, derived reduction is refutation complete. In addition, we draw a connection between complexity measures and decision procedures for elementary algebra. We also indicate applications of these theoretical results to human-oriented systems of natural deduction.

1. INTRODUCTION

Our primary purpose in this paper is to combine certain algorithms which often decide the word problem for arbitrary abstract algebras with the refutation procedures resolution (14) and paramodulation (13) in a refutationally complete manner. Our point of departure is from a class of decision procedures called complete sets of reductions which were discovered by Knuth and Bendix (10) and independently by Slagle (17) who calls them sets of simplifiers. The central idea behind complete sets of reductions is that equations which axiomatize an algebra are often used in one permanently fixed direction for simplification.

For example, the axioms of a semigroup with unit

1.1 $(x \cdot y) \cdot z = x \cdot (y \cdot z)$,

1.2 $x \cdot 1 = x$, and

1.3 $1 \cdot x = x$

constitute a solution of the word problem for semigroups with no generators and no relations as follows. If the axioms are used for simplification from left to right, then $t = u$ is a consequence of the axioms 1.1 - 1.3 iff t' and u' are identical terms, where t' and u' are the result of simplifying t and u as far as possible, e.g., $(1 \cdot x) \cdot (y \cdot 1) = (x \cdot y) \cdot 1$ because $x \cdot y$ and $x \cdot y$ are identical terms, while $(x \cdot y) \cdot z \neq$

$(x \circ y) \circ (w \circ 1)$ because $x \circ (y \circ z)$ and $x \circ (y \circ w)$ are not identical terms.

For the heuristic of unidirectional substitution of equals to be useful, there must be available some powerful and general methods for detecting when an algebraic theory can be realized by a complete set of reductions. Knuth and Bendix (10) provide such a method which consists of two algorithms: a finite termination property and a unique termination property. Their finite termination property is a complexity measure on terms which often determines when a set of unidirectional rewrite rules always leads to a finite sequence of simplifications, while their unique termination property is a necessary and sufficient criterion based on unification (14) for a set of rewrite rules which necessarily have the finite termination property to have the Church-Rosser property, consult Rosen (15). Their method has been enlarged through the discovery of other complexity measures by Lankford (11). It is not presently known if there is an algorithm which decides unique termination for sets of rewrite rules which do not necessarily have finite termination or if there is an algorithm which decides finite termination.

The unique termination property was originally stated by Knuth and Bendix (10) in terms of a concept they called superposition, which we rephrase using the notion of most general unifier below.

Let $\mathcal{R} = \{L_1 \longrightarrow R_1, \ldots, L_n \longrightarrow R_n\}$ be a finite set of rewrite rules, where L_i and R_i are terms. A special equality inference of $\mathcal{R}$ is an equation $t = u$ which is obtained from two rewrites $L_i \longrightarrow R_i$ and $L_j \longrightarrow R_j$ of $\mathcal{R}$ by replacing one occurrence of $L_i\theta$ in the left side of $L_j\theta = R_j\theta$ by $R_i\theta$ where θ is the most general unifier of L_i and a subterm of L_j which is not a variable.

1.4 The Unique Termination Algorithm If $\mathcal{R}$ is a set of rewrite rules such that each sequence of simplifications by $\mathcal{R}$ is finite, then $\mathcal{R}$ has the unique termination property iff each special equality inference $t = u$ of $\mathcal{R}$ has the property that t and u simplify to identical terms.

For a proof of 1.4 consult Knuth and Bendix (10). To illustrate the unique termination algorithm, let us establish the unique termination of the semigroup axioms 1.1 - 1.3. For the moment let us assume that the rewrite rules

1.5 $(x \cdot y) \cdot z \longrightarrow x \cdot (y \cdot z)$,

1.6 $x \cdot 1 \longrightarrow x$, and

1.7 $1 \cdot x \longrightarrow x$

have the finite termination property. Some of the special equality inferences of 1.5 - 1.7 are (for brevity we do not show all)

1.8 $(w \cdot (x \cdot y)) \cdot z = (w \cdot x) \cdot (y \cdot z)$ by 1.5 and 1.5,

1.9 $x = x$ by 1.6 and 1.6 (or 1.7 and 1.7), 1.10 $y \cdot z = 1 \cdot (y \cdot z)$ by 1.7 and 1.5,

1.11 $x \cdot z = x \cdot (1 \cdot z)$ by 1.6 and 1.5, and

1.12 $x \cdot y = x \cdot (y \cdot 1)$ by 1.6 and 1.5.

Of course the actual forms of the special equality inferences depend upon the formal language used and upon the unification algorithm. When each of the above is simplified as far as possible by the rewrite rules 1.5 - 1.7 (applied in whatever order one wishes) the corresponding sides of the equations become identical, namely $x \cdot (y \cdot (z \cdot w)))$, $y \cdot z$, $x \cdot z$, and $x \cdot y$ (both sides of 1.9 are already identical).

The simplicity of the solution of the unique termination problem which is evident from the preceding discussion stands in sharp contrast to the present state of affairs for the finite termination problem. The partial solutions which have been arrived at by Knuth and Bendix (10) and Lankford (11) do not seem to have been obtained through a deep understanding of the problem. For example, the family of complexity measures of Knuth and Bendix (10) is based primarily on the fact that if t is a term and $n_i(t)$ is the number of occurrences of function symbols of degree i in t then

1.13 $n_0(t) = 1 + n_2(t) + 2n_3(t) + \dots + (j - 1)n_j(t) + \dots$.

Despite its obscure origin, their family of complexity measures handles any associative axiom when expressed as $f(f(x,y),z) \longrightarrow f(x,f(y,z))$, many axioms which decrease length, and certain complexity measures of their family handle axioms which increase length, such as $(x \cdot y)^{-1} \longrightarrow (y^{-1}) \cdot (x^{-1})$.

Briefly, their complexity measures are defined in the usual manner, with a countable number of variable symbols $v_1, v_2, v_3, \ldots$, and a finite number of function symbols $f_1, \ldots, f_N$ of degrees $d_1, \ldots, d_N$. Constants are function symbols of degree 0 . Terms are variables, constants, or (recursively) expressions $f_i(t_1,\ldots,t_{d_i})$ where $t_1, \ldots$, and t_{d_i} are terms. Associated with each function symbol f_i is a non-negative integer w_i called the weight of f_i. The weights of functions satisfy two additional properties:

1.14 (1) each constant has positive weight, and

(2) each function symbol of degree 1 has positive weight, with the possible exception of the last function f_N .

The weight of a term t is defined as

1.15 $w(t) = \mathrm{MIN} \sum n(v_j,t) + \sum w_k n(f_k,t)$

where $n(v_j,t)$ is the number of occurrences of v_j in t , $n(f_k,t)$ is the number of occurrences of f_k in t , and MIN is the minimum of the weights of the constants. An order relation $>$ is defined on terms by

1.16 $t > u$ iff either (1) $w(t) > w(u)$ and $n(v_i,t) \geqq n(v_i,u)$ for all i, or (2) $w(t) = w(u)$ and $n(v_i,t) = n(v_i,u)$ for all i, and either $t = f_N(\ldots(f_N(v_j))\ldots)$, $u = v_j$ where $d_N = 1$, or $t = f_j(t_1,\ldots,t_{d_j})$,

$u = f_k(u_1,\ldots,u_{d_k})$ and either (2a) $j > k$ or (2b) $j = k$ and $t_1 = u_1, \ldots$, and $t_n > u_n$ for some n, $1 \leqq n \leqq d_j$.

By 1.13 and 1.14 it follows that $>$ is a well-ordering on terms without variable symbols and it is also shown by Knuth and Bendix (10) that if $t > u$ then $t\theta > u\theta$ for any substitution θ. It follows at once that if $\mathcal{R}$ is a set of rewrite rules for which each rewrite $L \longrightarrow R$ satisfies $L > R$ then $\mathcal{R}$ has the finite termination property. The finite termination of the axioms of a semigroup 1.5 - 1.7 is now easily settled by letting 1 and $\cdot$ have weight 1.

A striking feature of the approach of Knuth and Bendix (10) is that if a set of rewrite rules does not have the unique termination property then the uniqueness algorithm 1.4 forms the basis of an algorithm which often extends the incomplete set to a complete set. In order to describe this extension algorithm, we first define a simplification algorithm, denoted $*$, to be any algorithm which, given a set of rewrite rules $\mathcal{R}$ with the finite termination property and an expression t, produces a corresponding expression t^* which cannot be further simplified by the rewrites of $\mathcal{R}$. As and example of a simplification algorithm, consider the set of rewrite rules $\mathcal{R}$ as an ordered set, that is a sequence, and assume that the subexpressions of an expression t are ordered by depth first, and when at the same depth by left-most position.

Given the ordering of $\mathcal{R}$ and the ordering of subexpressions, let $*$ be the algorithm which simplifies an expression t by taking the rules of $\mathcal{R}$ in order and attempting to simplify the subexpressions of t in order, beginning with the deepest subexpression. When a simplification is made, $*$ recycles through $\mathcal{R}$, again beginning with the deepest subexpression of the simplified expression. With a given rewrite of $\mathcal{R}$, $*$ must fail to simplify every subexpression before going on to the next rewrite of $\mathcal{R}$. With one simplification algorithm in mind it is clear that by changing the order of $\mathcal{R}$ or the ordering on subexpressions other simplification algorithms can be defined.

1.17 The Knuth and Bendix Extension Algorithm Let $>$ be a complexity measure defined by 1.16, let $\mathcal{R}$ be a set of rewrite rules such that each member $L \longrightarrow R$ of $\mathcal{R}$ satisfies $L > R$, and let $*$ be any simplification algorithm.

(1) Set $i = 0$, $\mathcal{R}_i = \mathcal{R}$.

(2) Let $\mathcal{E}$ be the set of all special equality inferences of $\mathcal{R}_i$.

(3) Let $\mathcal{E}^*$ be the equations of $\mathcal{E}$ which have been completely simplified by $*$ using $\mathcal{R}_i$.

(4) Let $(\mathcal{E}^*)'$ be $\mathcal{E}^*$ minus all equations of the form $t = t$.

(5) If each equation $t = u$ of $(\mathcal{E}^*)'$ does not satisfy one of $t > u$ or $u > t$ then terminate, otherwise let

$\overline{(\mathcal{E}^*)'}$ be the set of rewrite rules obtained from $(\mathcal{E}^*)'$ using the complexity measure $>$.

(6) Set $j = 0$, $\mathcal{S}_j = \mathcal{R}_i \cup \overline{(\mathcal{E}^*)'}$ where $\mathcal{S}_j$ is a sequence of rewrites, $k =$ the number of members of $\mathcal{S}_j$, and $a = 1$.

(7) Select the first member $L \longrightarrow R$ of $\mathcal{S}_j$ and form the equation $L^* = R^*$ where $*$ uses $\mathcal{S}_j - \{L \longrightarrow R\}$ for simplification.

(8) If both L and R were already completely simplified, i.e., if $L^* = L$ and $R^* = R$, then let $\mathcal{S}_{j+1}$ be $\mathcal{S}_j$ modified with the first rewrite placed last, set $a = a + 1$, and set $j = j + 1$, otherwise go to (10).

(9) If $a > k$, set $\mathcal{R}_{i+1} = \mathcal{S}_{j+1}$, set $i = i + 1$, and go to (2), otherw se go to (7).

(10) If L^* and R^* are identical then set $\mathcal{S}_{j+1} = \mathcal{S}_j - \{L \longrightarrow R\}$, set $j = j + 1$, set $k = k - 1$, and go to (7).

(11) If L^* and R^* are $>$-incomparable then terminate.

(12) Now L^* and R^* must be $>$-comparable, i.e., $L^* > R^*$ or $R^* > L^*$. Let $t \longrightarrow u$ be the rewrite that results from the equation $L^* = R^*$, let $\mathcal{S}_{j+1}$ $(\mathcal{S}_j - \{L \longrightarrow R\}) \cup \{t \longrightarrow u\}$ where $t \longrightarrow u$ is the last rewrite in the ordered set $\mathcal{S}_{j+1}$, set $j = j + 1$, set $a = 1$, and go to (7).

One should notice that this algorithm terminates at $\mathcal{R}_T$ only in case either $\mathcal{R}_T$ is a complete set of reductions, or one of the simplified special equality inferences of $\mathcal{R}_T$ is $>$-incomparable, or a $>$-incomparable equation is generated when eliminating "redundancies" in 1.17 (7) - (12). The extension algorithm is amply illustrated with examples by Knuth and Bendix (10), including a derivation of a complete set of reductions for groups with no generators and no relations. Beginning with a minimal axiom set for groups,

1.18 $x \cdot 1 \longrightarrow x$,

1.19 $x \cdot (x^{-1}) \longrightarrow 1$, and

1.20 $(x \cdot y) \cdot z \longrightarrow x \cdot (y \cdot z)$,

an implementation of their algorithm produced the following seven additional rewrite rules in 30 seconds:

1.21 $1 \cdot x \longrightarrow x$,

1.22 $(x^{-1}) \cdot x \longrightarrow 1$,

1.23 $1^{-1} \longrightarrow 1$,

1.24 $(x^{-1})^{-1} \longrightarrow x$,

1.25 $(x \cdot y)^{-1} \longrightarrow (y^{-1}) \cdot (x^{-1})$,

1.26 $x \cdot ((x^{-1}) \cdot y) \longrightarrow y$, and

1.27 $(x^{-1}) \cdot (x \cdot y) \longrightarrow y$.

For the complexity measure $>$, $\cdot$ and -1 were given weight 0 and the constant 1 was given weight 1 .

Because of this and their other examples, one is impressed with the power and efficiency of their approach. For example, in the above process of extending to a decision procedure for groups, their program has established as a byproduct a number of theorems about elementary group theory which in the past have been found difficult for other theorem provers. The major difficulty with their approach is that given an initial set of axioms, there is at present no perscription for selecting a complexity measure which will lead to a set with unique termination. For example, in retrospect it can be seen that a complexity measure which will derive 1.18 - 1.27 must give the function -1 weight 0 ; otherwise, 1.25 will fail to satisfy $(x \cdot y)^{-1} > (y^{-1}) \cdot (x^{-1})$. But the selection of this weight is anything but obvious from inspection of the initial set 1.18 - 1.20, which any assignment of weights will establish. The family of complexity measures of Lankford (11) also suffers a similar defect. Thus an important question is: does there exist an algorithm which, given an axiom set and a family of complexity measures, determines whether or not one or more of the family can establish uniqueness? Another disadvantage of the Knuth and Bendix family defined by 1.16 is that although the distributive rewrite $x \cdot (y + z) \longrightarrow (x \cdot y) + (x \cdot z)$ has finite termination, none of their family will detect this fact.

The family of complexity measures of Lankford (11) contains members which insure the finite termination of these distributive rewrites. Let us briefly summarize his approach below. Recall the term structure of the first order predicate calculus. For each function symbol $f_1, \ldots, f_N$ let $F_1, \ldots, F_N$ be functions from the positive integers to the positive integers such that

1.28 (1) the degree of each F_i is the same as the degree of the corresponding f_i ,

(2) $F_i(x_1,\ldots,x_j,\ldots,x_{d_i}) < F_i(x_1,\ldots,y,\ldots,x_{d_i})$ when $x_j < y$,

and let $\|\cdot\|$ be the function defined on all terms by

(3) $\|v_i\|$ is some fixed positive integer for all i ,

(4) $\|f_i\| = F_i$ when f_i is a constant, and

(5) $\|f_i(t_1,\ldots,t_{d_i})\| = F_i(\|t_1\|,\ldots,\|t_{d_i}\|)$.

It has been shown by Lankford (11) that if $\mathcal{R}$ is a set of rewrite rules and $\|L\theta\| > \|R\theta\|$ for all substitutions θ and all $L \longrightarrow R$ in $\mathcal{R}$ then $\mathcal{R}$ has the finite termination property.

A complexity measure is determined by specifying $F_1,\ldots,F_N$ satisfying 1.28 (1) and (2), selecting a positive integer for 1.28(3) which determines $\|\cdot\|$, and defining

1.29 $t > u$ iff $\|t\theta\| > \|u\theta\|$ for all substitutions θ .

The primary defect with this approach is that the selection of the F_i and the fixed constant for 1.28 (3) must presently be made by trial and error. To illustrate this approach, notice that $F_{\circ}(x,y) = x(1+2y)$, $F_{-1}(x) = x^2$, $F_1 = 2$, and $\|v_i\| = 2$ detect the finite termination of the ten group rewrites 1.18 - 1.27.

Another difficulty is that we know of no algorithmic test for the complexity measures defined by 1.29. However, when the F_i are polynomials a weaker version of 1.29 can be realized by any decision procedure for elementary algebra, such as those of Tarski (18), Seidenberg (16), Cohen (4), and Collins (5), as we show below. Let S be the sentence

<u>1.30</u> $\exists r \forall x_1 \ldots \forall x_n \left(x_1 \geq r \wedge \ldots \wedge x_n \geq r \Longrightarrow \overline{\|t\|} > \overline{\|u\|} \right)$

where $\overline{\|t\|}$ and $\overline{\|u\|}$ are obtained by replacing $\|v_{i_1}\|$, $\ldots$, $\|v_{i_n}\|$ in $\|t\|$ and $\|u\|$ by $x_1, \ldots, x_n$ (the v_{i_j} are the variable symbols that occur in t and u). For the sentences S to faithfully capture 1.29, they must be considered to be sentences interpreted over the integers. Unfortunately, methods used by Davis (6) to show the algorithmic unsolvability of Hilbert's tenth problem can be used to show that there is no algorithm to decide sentences of the form of 1.30.[1] Still, a weaker realization of 1.29 can be obtained by considering S to be a sentence of elementary algebra. In that case the complexity measure defined by

<u>1.31</u> $t > u$ iff S is true, where S is defined by 1.30,

is realized by any decision method for elementary algebra. Collins (5) has reported that an implementation of his decision method will soon be available. We do not know of any implementations of the other decision methods for elementary algebra.

[1] A proof of this fact was given by Martin Davis at the Oberwolfach conference on automatic theorem proving on January 7, 1976, and will be included in a revision of this paper.

As has been said, our primary concern in this paper is to combine complete sets of reductions with the refutation procedures resolution and paramodulation refutationally complete manner. Our approach is straightforward and is based on the simple idea to perform ordinary inferences followed by simplification of the ordinary inferences as far as possible, discarding the partially simplified intermediate steps and saving only the final completely simplified expression. To illustrate this approach let us establish a fragment of a proof of a theorem found in Herstein (7) that H is a subgroup of G iff H is not empty and for each x and y in H, $x \cdot (y^{-1}) \in H$. Let us establish just one part of the above by showing

<u>1.32</u> $c \in H$, and

<u>1.33</u> $x \in H \wedge y \in H \implies x \cdot (y^{-1}) \in H$

imply

<u>1.34</u> $1 \in H$.

We assume the presence of the complete set of reductions for groups, given earlier in 1.18 - 1.27. For this example modus ponens is used to illustrate the natural appearance of canonical inference. By modus ponens with 1.32 and 1.33 the ordinary inference

<u>1.35</u> $c \cdot (c^{-1}) \in H$

is inferred. When 1.35 is simplified as far as possible, 1.34 results. It is easy to see how the other parts would be established.

This paper also deals with sets of rewrite rules which do not have the unique termination property. That such sets exist naturally is a consequence of the unsolvability of word problems. Moreover, there is no algorithm which will decide from the axioms of an algebra whether or not its word problem is solvable, nor is there a partial algorithm which solves the word problem just for those algebras with a solvable word problem, consult Jones (9). In view of these negative results, it would seem that the best use of rewrite rules is while searching for a refutation or proof to simultaneously use the Knuth and Bendix extension algorithm to attempt to find a complete set of reductions. The derived reduction algorithm below does just that. Essentially we have taken 1.17, and when it would normally terminate with a $>$-incomparable equation or be unusable with an initial axiom which is $>$-incomparable, we have continued to form inferences using special paramodulation, which is defined to be ordinary paramodulation with the restriction that substitution into variable positions is not permitted, and special substitution of equals, which is defined to be paramodulation between a rewrite rule and an equation where substitution into a variable position is not allowed, only left sides of rewrite rules are substituted into by an equation, and only left sides of rewrite rules are replaced by right sides when rewrite rules are paramodulated into equations.

<u>1.36</u> <u>The Derived Reduction Algorithm</u> Let $>$ be a complexity measure defined by 1.16 or 1.29, and let $\mathcal{S}$ be a finite equality unsatisfiable set of clauses which contains the trivial reflexive axiom $x \doteq x$ and such that no equation occurs in a non-unit clause.

(1) Set $i = 0$, let $\mathcal{R}_i$ be the equations of $\mathcal{S}$ which can be expressed as rewrites by the complexity measure $>$, let $\mathcal{E}_i$ be the remainder of the equations, and let $\mathcal{S}_i$ be the remainder of $\mathcal{S}$.

(2) By an obvious modification of 1.17 (7) - (12) we may assume $\mathcal{R}_i$ and $\mathcal{E}_i$ to be such that equations of $\mathcal{E}_i$ cannot be further simplified by $\mathcal{R}_i$ and that no rewrite $L \longrightarrow R$ of $\mathcal{R}_i$ can be further simplified by $\mathcal{R}_i - \{L \longrightarrow R\}$.

(3) Reset $\mathcal{S}_i$ to $\mathcal{S}_i^*$, where $*$ uses $\mathcal{R}_i$.

(4) Form all the resolvents R, all the special equality inferences I, all the special paramodulants P, and all the special substitution of equals S, and from $I^* \cup P^* \cup S^*$ put all the $>$-comparable equations as rewrites into $\mathcal{R}$, and all the $>$-incomparable equations into $\mathcal{E}$. Set $\mathcal{R}_{i+1} = \mathcal{R}_i \cup \mathcal{R}$, $\mathcal{E}_{i+1} = \mathcal{E}_i \cup \mathcal{E}$, $\mathcal{S}_{i+1} = \mathcal{S}_i \cup R^*$, $i = i + 1$, and go to (2).

We will presently show that $\square \in \mathcal{S}_k$ for some k. Refutation completeness of 1.36 holds in two interesting degenerate cases: (1) when $\mathcal{R}_0$ is a complete set of reductions and $\mathcal{E}_0$ is empty, and (2) when there is no complexity measure $>$. A less general form of the first degenerate case has been reported by Slagle (17) where he assumes that the input set $\mathcal{S}$ is fully narrowed. The second degenerate case sheds some light on the functional reflexive problem (13). In fact for the general case of 1.36, the functional reflexive axioms are not needed. Recently several researchers have announced that special paramodulation is refutation complete without the functional reflexive axioms.[1] However, this writer has been unable to extend the degenerate case above to the case when equations occur in non-unit clauses, and he is presently unsure of the status of the announced solutions. An algorithm similar to 1.36 has also been reported by Winker (19). An implementation of special paramodulation has been used by Nevins (12) with some impressive successes. A partial implementation of 1.36 by Ballantyne and Lankford in LISP at The University of Texas at Austin substantially improved an example of Nevins (12) that in a group $x^3 = 1$ implies $h(h(x,y),y) = 1$ where $h(x,y) = xyx^{-1}y^{-1}$. Nevins' program took 30 minutes and terminated with a search space of 415 formulas, while Ballantyne and Lankford's program took 30 seconds and terminated with a search space of 11 formulas.

[1] See Resolution and Equality in Theorem Proving, by D. Brand, Dept. of Comp. Sci., Tech. Report # 58, Univ. of Toronto, Nov. 1973, and A Note On The Functional Reflexive Problem, M. Richter, Insbesondere Informatic, Technische Hochschule, Aachen, West Germany.

2. CANONICAL INFERENCE

The terms of the first order logic are constructed in the usual manner from variable, constant, and function symbols. A set of reduction relations is a finite set of objects $L \longrightarrow R$ where L and R are terms and each variable symbol which occurs in R also occurs in L. Each set of reduction relations $\mathcal{R}$ is associated with a corresponding set of equations $E(\mathcal{R})$ by identifying each reduction relation $L \longrightarrow R$ with the equation $L = R$. The term u is an immediate reduction of the term t, denoted $t \longrightarrow u$, in case for some substitution θ, u is the result of replacing one occurrence of $L\theta$ in t by $R\theta$. A set of reduction relations has the finite termination property in case for any term t each sequence $t \longrightarrow t_1 \longrightarrow t_2 \longrightarrow \cdots$ of immediate reductions originating with t terminates after a finite number of steps; that is, some term t_m of the sequence above has no immediate reductions. A set of reductions is a set of reduction relations with the finite termination property. A set of reduction relations has the unique termination property in case for each term t, any two terminating sequences of immediate reductions originating with t terminate with identical terms. A set of reductions with the unique termination property is called a complete set of reductions, which is somewhat more general than the complete set of reductions discussed by Knuth and Bendix (10) and essentially the same as a set of simplifiers described by Slagle (17). Let $\mathcal{R}$ be a complete set of reductions and let $*$ be any algorithm which associates with each term t the corresponding term t^* such that t^* is the last term in a (necessarily terminating) sequence of immediate reductions originating with t . When t has no immediate reductions, t^* is t . We call such terms t^* irreducible with respect to $\mathcal{R}$, and omit reference to $\mathcal{R}$ when ambiguity is unlikely. It may sometimes be convenient to use $\longrightarrow$ to denote a finite (zero or more) sequence of immediate reductions. The operator $*$ and the relation $\longrightarrow$ are extended to predicates, literals, clauses, and sets of clauses in the obvious manner.

While familiarity with the investigations of Knuth and Bendix (10), Lankford (11), and Slagle (17) would be helpful, we have attempted to include the pertinent background. We do assume a thorough knowledge of the basic results about resolution and paramodulation, and especially the excess literal method of Anderson and Bledsoe (1). Our approach to establishing 1.36 is to establish the two degenerate cases first. We begin with an extension of some results reported by Slagle (17).

2.1. BLOCKED RESOLUTION

It might be hoped that complete sets of reductions could be combined directly with resolution; that is, we might conjecture that if S is a set of clauses that contains no equations and $S \cup E(\mathcal{R})$ is equality-unsatisfiable, then $S^* \cup \{\{x = x\}\}$ is unsatisfiable. But let $\mathcal{R}$ be $\{f(g(x,y)) \longrightarrow g(f(x),f(y))\}$ and let S be $\{\{P(f(x))\}, \{\neg P(g(f(a),f(b))\}\}$ and notice that S is irreducible and satisfiable in the presence of $x = x$. While the general conjecture fails, we shall see in Theorem 1 that the corresponding ground conjecture holds. Of course, the counter-example above shows that the ground result cannot be lifted in the usual way. Indeed, examination of this lifting failure will guide us to one solution for the general case. As a necessary preliminary, we first establish the following property of equality-unsatisfiable sets of ground unit clauses.

Lemma 1 If S is a set of ground unit clauses which is closed under paramodulation, contains no complementary pairs, and contains no inequality of the form $t \neq t$, then S has an equality model.

Proof Let T be S together with all ground unit equations of the form $t = t$ where t is any ground term over the Herbrand base of S. Let $P(T)$ be the closure of T under paramodulation. It is clear that $P(T)$ has no complementary pair or inequality of the form $t \neq t$. Let I be the partial interpretation which consists of the positive literals of $P(T)$, and let M be the interpretation obtained by adding to I every negative ground literal over the Herbrand base of S which is not a complement of a member of I. This "most negative" interpretation device was a prominent feature of the maximal model construction of Wos and Robinson (20) which was used to establish the refutation completeness of paramodulation for equality unsatisfiable sets which contain the functional reflexive axioms. It now easily follows that M is an equality model of S.

__Theorem 1__ If $\mathcal{R}$ is a complete set of reductions, S is a set of ground clauses which contain no equations, and $S \cup E(\mathcal{R})$ is equality-unsatisfiable then there is a deduction of $\square$ from $S^* \cup \{\{x = x\}\}$ using resolution.

__Proof__ We induct on the excess literal parameter of S. Throughout, let us depict that R is a resolvent of C and D by the diagram

$$\begin{tikzcd}
C \arrow[dr, dash] & & D \arrow[dl, dash] \\
& R &
\end{tikzcd}$$

and that P is a paramodulant of C by E, where E is the equation of substitution, by the diagram

$$\begin{tikzcd}
E \arrow[r, Rightarrow] & C \arrow[d] \\
& P .
\end{tikzcd}$$

Because of Lemma 1, there must be a complementary pair or an inequality of the form $t \neq t$ which is derivable from S and a finite set of ground instances $E(\mathcal{R})'$ of $E(\mathcal{R})$, when S consists entirely of units. Thus, in the unit case it can be seen that there exists a refutation of $\square$ which has one of two forms:

$$\begin{tikzcd}
t_1 = u_1 \arrow[r, Rightarrow] & C \arrow[d] & D \arrow[d] & v_1 = w_1 \arrow[l, Rightarrow] \\
& P_1 \arrow[d, phantom, "\vdots"] & Q_1 \arrow[d, phantom, "\vdots"] & \\
t_n = u_n \arrow[r, Rightarrow] & P_{n-1} \arrow[d] & Q_{m-1} \arrow[d] & v_m = w_m \arrow[l, Rightarrow] \\
& P_n \arrow[dr, dash] & Q_m \arrow[d, dash] & \\
& & \square &
\end{tikzcd}$$

where C and D are members of S and the equations $t_i = u_i$ and $v_j = w_j$ are inferred from the ground instances $E(\mathcal{R})'$, or

$$\begin{tikzcd}
t_1 = u_1 \arrow[r, Rightarrow] & v \neq w \arrow[d] \\
& v_1 \neq w_1 \arrow[d, phantom, "\vdots"] \\
t_n = u_n \arrow[r, Rightarrow] & v_{n-1} \neq w_{n-1} \arrow[d] \\
x = x \arrow[dr, dash] & t \neq t \arrow[d, dash] \\
& \square
\end{tikzcd}$$

where $v \neq w$ is a member of S and the equations $t_i = u_i$ are inferred from the ground equations $E(\mathcal{R})'$. Let us consider the second form first. It is clear that $v = t$ and $w = t$ are consequences of $E(\mathcal{R})$, and since it has been shown by Knuth and Bendix (10) that the $*$ algorithm is a canonical simplification algorithm for $E(\mathcal{R})$, it follows that v^* and t^* are identical and that w^* and t^* are identical, hence that v^* and w^* are identical. So in this case it follows that S^* contains the inequality $v^* \neq v^*$, and hence $\Box$ is derived by resolving with $x = x$. For the second form we extend the approach used above in the first form. Recall that any literal has the form $X(x_1,\ldots,x_k)$ or $\neg X(x_1,\ldots,x_k)$ where X is a predicate symbol and the x_i , $i = 1 , \ldots , k$, are terms. Consequently, we can represent C , D , P_n , and Q_m by $\pm X_C(c_1,\ldots,c_k)$, $\pm X_D(d_1,\ldots,d_k)$, $\pm X_{P_n}(p_1,\ldots,p_k)$, and $\pm X_{Q_m}(q_1,\ldots,q_k)$. It is clear that the equations $c_i = p_i$, $i = 1 , \ldots , k$, and the equations $q_i = d_i$, $i = 1 , \ldots , k$, are consequences of $E(\mathcal{R})$ and that p_i and q_i , $i = 1 , \ldots , k$, are identical. It follows that c_i^* and d_i^* , $i = 1 , \ldots , k$, are identical. In this case we see that C^* and D^* are complements. This completes the proof of the unit case. The induction step is routine, and so is not presented here.

The direct lifting of this result fails primarily because an instance of an irreducible clause may fail to be irreducible. Therefore, in order for the usual lifting lemma to apply, we must first develop a procedure which given any clause C and any instance C' of C , transforms C into a clause D which has C'^* as an instance. This can be easily done by treating the reductions as equations and allowing paramodulation ontc subterms which are not variables by the left sides of the reductions, followed by reduction of the resulting paramodulant to irreducible form. This kind of restricted paramodulation is called immediate narrowing by Slagle (17). Our discussion is more general here since he considers only sets of reductions which produce only finite sequences of immediate narrowings originating from any term t . For example, any complete set of reductions which contains an associative reduction $f(f(x,y),z) \longrightarrow f(x,f(y,z))$ will produce the infinite sequence of immediate narrowings $f(x_1,x_2)$, $f(x_1,f(x_2,x_3))$, ... , $f(x_1,f(x_2,\ldots f(x_{n-1},x_n)\ldots))$, ... A narrowing is a finite sequence of immediate narrowings. The following lemma was stated without proof by Slagle (17).

Lemma 2 If $\mathcal{R}$ is a complete set of reductions, C is a clause, and C' is an instance of C then there is a narrowing C^N of C which has $(C')^*$ as an instance.

Proof Let θ be the substitution which takes C to C' , and let C'' be the substitution instance of C under θ^* , where θ^* is the substitution which results from θ by applying $*$ to each term of each substitution component of θ . It can be seen that C'' is also the result of applying a finite sequence of immediate reductions to C' , and as such can be thought of as an intermediate step in the construction of $(C')^*$. If C'' is irreducible then we are done. If C'' is not irreducible then let

$C'' \longrightarrow C_1$ be an immediate reduction of C''. Since C'' is an instance of C under an irreducible substitution, the reduction which takes C'' to C_1 must apply to a subterm of C'' which does not correspond to the position of a variable in C. Thus there is a paramodulant of C which has C_1 as an instance, which we denote by P. Let C_1' be the partial reduction of C_1 which is obtained by the corresponding sequence of reductions which takes P to P^*. It can be seen that the immediate narrowing P^* of C has C_1' as an instance under an irreducible substitution. As this process is iterated, we succesively produce ground clauses C_i' which are instances of narrowings of C and which are also intermediate steps in the production of $(C')^*$. Because of finite termination, $(C')^*$ must eventually be one of the C_i'.

Once the appropriate narrowings of a set of clauses are found, the ground refutation can be lifted in the usual way without further need of narrowing. In fact, since the ground refutation is irreducible at each step, the lifted refutation will be such that all resolvents are irreducible, and in addition each most general unifier is irreducible. Slagle (17) has called this kind of deduction blocked resolution. These facts are summarized below.

Theorem 2 If $\mathcal{R}$ is a complete set of reductions, S is a set of clauses which contains $x = x$ and no other equations, and $S \cup E(\mathcal{R})$ is equality-unsatisfiable then there exists a finite set of narrowings S^N of S from which the empty clause can be refuted by blocked resolution and blocked factoring.

Theorem 2 now forms the basis for a refutation complete algorithm for equality unsatisfiable sets $\mathcal{S}$ which contain no occurrences of equations other than units and for which the set of equations $E(\mathcal{S})$ of $\mathcal{S}$ are consequences of some complete set of reductions $\mathcal{R}$.

2.1 (1) Set $\mathcal{S}_0 = \mathcal{S}^*$, from which we may assume all tautologies have been deleted.

(2) Form all blocked resolvents $\mathcal{B}$ of $\mathcal{S}_k$ and all immediate narrowings η of $\mathcal{S}_k$.

(3) Set $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \mathcal{B} \cup \eta$ and return to step (2).

To illustrate this algorithm let us return to the subgroup problem of 1.32 - 1.34. Again we assume the presence of the complete set of reductions for groups. Following 2.1, $\mathcal{S}_0$ consists of

2.2 $c \in H$,

2.3 $x \notin H \vee y \notin H \vee x \cdot (y^{-1}) \in H$, and

2.4 $1 \notin H$.

The only blocled resolvents of $\mathcal{S}_0$ are

2.5 $y \notin H \vee c \cdot (y^{-1}) \in H$ by 2.2 and 2.3, and

2.6 $x \notin H \vee x \cdot (c^{-1}) \in H$ by 2.2 and 2.3.

Some of the immediate narrowings of $\mathcal{S}_0$ are

2.7 $x \notin H \vee 1 \in H$ by 1.19 and 2.3,

2.8 $x^{-1} \notin H \vee 1 \in H$ by 1.22 and 2.3, and

2.9 $x \notin H \vee y^{-1} \notin H \vee x \cdot y \in H$ by 1.24 and 2.3.

On the second round $1 \in H$ is produced by block resolving 2.2 and 2.7, so that $\square$ is produced on the third round.

Notice that since blocked resolution with narrowing is complete, ordinary resolution followed by simplification (with narrowing) is complete. Thus the refutation completeness of 1.36, derived reduction, in the degenerate case when $\mathcal{R}_0$ of 1.36 (1) is a complete set of reductions, is a corollary of the refutation completeness of blocked resolution.

2.2 SPECIAL PARAMODULATION

In this section we establish the refutation completeness of 1.36, the derived reduction algorithm, in the degenerate case when there is no complexity measure. Here we modify the approach used to establish the refutation completeness of blocked resolution. The basic idea of this section is to take the equations of a finite equality unsatisfiable set of ground instances of a general finite equality unsatisfiable set, extend these ground equations to a complete set of reductions, use Theorem 1 to get a ground refutation, and with an analog of Lemma 2 lift the ground result.

Lemma 3 If $\mathcal{R}$ is a set of reduction relations with the finite termination property, then $\mathcal{R}$ has the unique termination property iff the following lattice condition holds:

2.10 if t is any term and u and v are immediate reductions of t, then there exists a term w and two sequences $u = u_0 \longrightarrow \cdots \longrightarrow u_n = w$ and $v = v_0 \longrightarrow \cdots \longrightarrow v_m = w$ of immediate reductions from u and v which terminate with w.

For a proof of Lemma 3 consult Lankford (11).

Lemma 4 If $\mathcal{R}$ is a set of reduction relations with the finite termination property and $*$ is a simplification algorithm, then the lattice condition for $\mathcal{R}$ holds iff each special equality inference $t = u$ of $\mathcal{R}$ has the property that t^* and u^* are identical terms.

<u>Proof</u> ($\Longrightarrow$) Let $t = u$ be a special equality inference of $\mathcal{R}$. This means there are members $L_1 \longrightarrow R_1$ and $L_2 \longrightarrow R_2$ of $\mathcal{R}$, and a most general unifier θ of L_1 and a subterm of L_2 which is not a variable such that $t = (L_2\theta)'$ and $u = R_2\theta$ where $(L_2\theta)'$ is the result of replacing one occurrence of $L_1\theta$ in $L_2\theta$ by $R_1\theta$. Notice that t and u are immediate reductions of $L_2\theta$, and so by the lattice condition with the help of Lemma 3 it follows that t^* and u^* are identical.

($\Longleftarrow$) Let $t \longrightarrow u_0$ and $t \longrightarrow v_0$ be immediate reductions of t by reduction relations $L_1 \longrightarrow R_1$ and $L_2 \longrightarrow R_2$ of $\mathcal{R}$. If L_1 and L_2 do not "interact," then reducing u_0 by $L_2 \longrightarrow R_2$ and v_0 by $L_1 \longrightarrow R_1$ in the corresponding positions that t was reduced produces $u_0 \longrightarrow w$ and $v_0 \longrightarrow w$. If L_1 and L_2 do interact, then without loss of generality assume that $L_1\theta_1$ replaces a subterm of $L_2\theta_2$, where $L_1\theta_1$ is replaced by $R_1\theta_1$ in t to produce u_0 and $L_2\theta_2$ is replaced by $R_2\theta_2$ in t to produce v_0.

If the subterm of $L_2\theta_2$ replaced by $L_1\theta_1$ corresponds to a variable position in L_2, then replace all other occurrences of $L_1\theta_1$ in $L_2\theta_2$ which result from that variable in θ_2. Thus we have $t = (\ldots L_2\theta_2\ldots) \longrightarrow u_0 \longrightarrow \cdots \longrightarrow (\ldots L_2(\theta_2')\ldots)$ where the substitution $\theta_2' = \{t_1/v_{i_1},\ldots,t_j'/v_{i_j},\ldots,t_k/v_{i_k}\}$

is obtained from the substitution $\theta_2 = \{t_1/v_{i_1},\ldots,t_j/v_{i_j},\ldots,t_k/v_{i_k}\}$ by replacing the one corresponding occurrence of $L_1\theta_1$ in t_j by $R_1\theta_1$. Next form the immediate reduction $(\ldots L_2(\theta_2')\ldots) \longrightarrow (\ldots R_2(\theta_2')\ldots) = w$. On the other hand, we have $t = (\ldots L_2\theta_2\ldots) \longrightarrow (\ldots R_2\theta_2\ldots) = v_0$ and by forming a similar sequence of immediate reductions we have $v_0 \longrightarrow \ldots \longrightarrow (\ldots R_2(\theta_2')\ldots) = w$.

If the subterm of $L_2\theta_2$ replaced by $L_1\theta_1$ does not correspond to a variable position, then there is a special equality inference $u = v$ of $L_1 \longrightarrow R_1$ and $L_2 \longrightarrow R_2$ and a substitution θ such that $u_0 = u\theta$ and $v_0 = v\theta$. By assumption u^* and v^* are identical, and by performing the corresponding reductions to those used to obtain u^* and v^*, we get $u_0 \longrightarrow \ldots \longrightarrow w$ and $v_0 \longrightarrow \ldots \longrightarrow w$. This completes the proof of Lemma 4. It should be noticed that Lemma 3 and Lemma 4 constitute a proof of 1.4, the unique termination algorithm.

Theorem 3 Let $>$ be a relation which satisfies

2.11 (1) exactly one of $t > u$, $u > t$, or t and u are identical, for each pair of ground terms t and u,

(2) if t , u and v are ground terms, $t > u$ and w is the result of replacing one occurrence of t in v by u, then $v > w$, and

(3) there is no infinite sequence $t_1 > t_2 > t_3 > \ldots$.

The lexical order (14) and the Knuth and Bendix complexity measures satisfy 2.11 and may be kept in mind as a model for the relation of this theorem. Let $\mathcal{E}$ be a finite set of unit equations and $\mathcal{G}$ a finite set of ground instances of $\mathcal{E}$. Delete all equations of the form $t = t$ from $\mathcal{G}$ and using the relation $>$ express the remainder of $\mathcal{G}$ as a set of rewrites $\mathcal{R}_0$.

2.12 (1) Set $\mathcal{E}_0 =$ the set of triples $(t = u, \theta, L \longrightarrow R)$ where $L \longrightarrow R$ is a rewrite of $\mathcal{R}_0$ and is the substitution instance of $t = u$ under θ. It may happen that $u = t$, instead of $t = u$, is in $\mathcal{E}$ but then $u = t$ can be derived from $\mathcal{E}$ by special paramodulation. So without loss of generality we assume that if $t = u$ is in $\mathcal{E}_k$ then $u = t$ is in $\mathcal{E}_k$.

(2) Form all the special equality inferences S of $\mathcal{R}_k$. Delete from S all equations of the form $t = t$ and divide the remainder into two sets S_1 and S_2, where S_1 is the set of all equations which were obtained by substituting $L_i \longrightarrow R_i$ into a subterm of L_j that corresponds to a variable position in t_j for some $(t_i = u_i, \theta_i, L_i \longrightarrow R_i)$ and $(t_j = u_j, \theta_j, L_j \longrightarrow R_j)$ in $\mathcal{E}_k$, and where S_2 is the set of equations that were obtained by substituting into a position that does not correspond to a variable.

(3) Further simplify each equation $L_j' = R_j$ of S_1 to

$(L_j')' = R_j'$ which is the substitution instance under θ_j' of $t_j = u_j$, where θ_j' is formed like θ_2' in the proof of Lemma 4, and replace S_1 by S_1' which consists of all the corresponding $(L_j')' = R_j'$. Delete from S_1' all equations of the form $t = t$, and express the remainder as rewrites, which are then used to form $\mathcal{E}^1$ the set of all triples $(t_j = u_j, \theta_j', (L_j')' \longrightarrow R_j')$ or $(u_j = t_j, \theta_j', R_j' \longrightarrow (L_j')')$ depending on whether $(L_j')' > R_j'$ or $R_j' > (L_j')'$.

(4) From S_2, because substitution is into a position that does not correspond to a variable, we can form $\mathcal{E}^2$ the set of triples $(v = w, \theta, L \longrightarrow R)$ where $L \longrightarrow R$ is a reduction obtained from S_2 and is the instance of $v = w$ by θ and where $v = w$ or $w = v$ is a special paramodulant of two equations that are first coordinates of two triples of $\mathcal{E}_k$.

(5) Set $\mathcal{E}_{k+1} = \mathcal{E}_k \cup \mathcal{E}^1 \cup \mathcal{E}^2$, $\mathcal{R}_{k+1}$ the third coordinates of $\mathcal{E}_{k+1}$. If $\mathcal{E}_{k+1}$ and $\mathcal{E}_k$ are identical then terminate, otherwise return to (2).

The algorithm 2.12 terminates and the terminal set of reductions $\mathcal{R}_T$ is a complete set of reductions.

<u>Proof</u> If 2.12 did not terminate then by 2.11 (2), it would follow that there is an infinite sequence $t_1 > t_2 > t_3 \quad \ldots$ contradicting 2.11 (3). We establish that $\mathcal{R}_T$ is complete by

showing that the lattice condition holds. Our proof is similar to the ($\Longleftarrow$) part of the proof of Lemma 4. Let $t \longrightarrow u$ and $t \longrightarrow v$ be immediate reductions by $L_1 \longrightarrow R_1$ and $L_2 \longrightarrow R_2$ of $\mathcal{R}_T$. The case when L_1 and L_2 do not interact is obvious. When L_1 and L_2 do interact, consider the triples $(t_1 = u_1, \theta_1, L_1 \longrightarrow R_1)$ and $(t_2 = u_2, \theta_2, L_2 \longrightarrow R_2)$ of $\mathcal{E}_T$, and without loss of generality assume L_1 is the subterm of L_2 that is replaced. If the subterm of L_2 that is replaced corresponds to a variable position in t_2, then form θ_2' (as in the proof of Lemma 4) and perform the corresponding sequence of reductions $t = (\ldots L_2 \ldots) \longrightarrow \ldots \longrightarrow (\ldots L_2' \ldots)$ where $L_2' = t_2(\theta_2')$. On the other hand, we have $t = (\ldots L_2 \ldots) \longrightarrow (\ldots R_2 \ldots) \longrightarrow \ldots \longrightarrow (\ldots R_2' \ldots)$ where $R_2' = u_2(\theta_2')$. If L_2' and R_2' are identical then we are done. Otherwise, one of $(t_2 = u_2, \theta_2', L_2' \longrightarrow R_2')$ or $(u_2 = t_2, \theta_2', R_2' \longrightarrow L_2')$ is in $\mathcal{E}_T$. Thus it is clear that there exists some w such that $u \longrightarrow \ldots \longrightarrow w$ and $v \longrightarrow \ldots \longrightarrow w$. The case when substitution is into a position that does not correspond to a variable is handled similar to the corresponding part of the proof of Lemma 4.

<u>Theorem 4</u> If $\mathcal{S}$ is a finite equality unsatisfiable set of clauses for which no equation occurs in a non-unit clause, then there is a refutation of $\Box$ from $\mathcal{S}$ together with $x = x$ using factoring, resolution, and special paramodulation.

<u>Proof</u> Let $\mathcal{G}$ be a finite equality unsatisfiable set of ground instances of $\mathcal{S}$. Using the lexical order with Theorem 3, form $\mathcal{E}_T$ the terminal set of 2.12. The resulting complete set of reductions $\mathcal{R}_T$ is used to form $\mathcal{G}^*$, where * is any simplification algorithm. By Theorem 1 there is a blocked refutation of □ from $\mathcal{G}^*$. Using the equations of $\mathcal{E}_T$ and special paramodulation,we can derive a set $\mathcal{S}^{SP}$ from $\mathcal{S}$ which has the clauses of $\mathcal{G}^*$ as instances. The proof of this is similar to the proof of Lemma 2 and so is omitted. The ordinary lifting lemma for resolution now lifts the ground refutation from $\mathcal{G}^*$ in the usual manner.

2.3 DERIVED REDUCTION

In this section we establish the refutation completeness of 1.36, derived reduction. Our approach is motivated by Section 2.2 and especially by Theorem 3. Now, however, if we try to duplicate the proof of Theorem 4, beginning with an equality unsatisfiable set of ground instances, and use Theorem 3 to extend to a complete set, several things go wrong. We no longer have only equations at the general level but also reductions. Moreover, the general reductions, equations, and clauses are simplified during each round; so it follows that these simplifications at the general level often force simplifications at the ground level which cannot be duplicated by the original ground instances or their inferences. And in addition, because the general level reductions are determined by a complexity measure, we must find a relation $>$ which satisfies 2.11 and is also compatible with the complexity measure.

Let us consider the complexity measure problem first. A complexity measure is a structure $>$, $\approx$ where

2.13 (1) $>$ is a subset of the Cartesian product of the terms with themselves,

(2) $\approx$ is an equivalence relation on the terms, that is

(a) $t \approx t$ for any term t,

(b) if $t \approx u$ then $u \approx t$, and

(c) if $t \approx u$ and $u \approx v$ then $t \approx v$,

(3) if $t > u$ and $u > v$ then $t > v$,

(4) if $t > u$ $(t \approx u)$ and θ is any substitution then $t\theta > u\theta$ $(t\theta \approx u\theta)$,

(5) if $t > u$ $(t \approx u)$ and w is the result of replacing one occurrence of t in v by u then $v > w$ $(v \approx w)$,

(6) there is no infinite sequence $t_1 > t_2 > t_3 > \dots$.

A complexity measure is said to be ground regular in case

2.14 (1) $t > u$ or $u > t$ or $t \approx u$ for any ground terms t , u ,

(2) $t > u \approx v$ implies $t > v$ for any ground terms t , u , v .

It is now easy to see that

2.15 if a complexity measure is ground regular then exactly one of $t > u$, $u > t$, or $t \approx u$ is true for any ground terms t , u .

The complexity measures 1.16 of Knuth and Bendix with $\approx$ the identity relation and 1.29 with $t \approx u$ defined by $\|t\| = \|u\|$ are ground regular complexity measures. The ground regular complexity measures are those for which we can show derived reduction is refutation complete. We now define a relation $>$ satisfying 2.11 which is compatible with a given ground regular complexity measure. Let R be any relation satisfying 2.11 and

2.16 for each ground term t there are only finitely many u such that $t \, R \, u$,

let C , $\approx$ be a ground regular complexity measure, and for each pair of ground terms t and u let

2.17 (1) $t > u$ if $t \, C \, u$,

(2) $u > t$ if $u \, C \, t$,

(3) $t > u$ if $t \approx u$ and $t \, R \, u$, and

(4) $u > t$ if $t \approx u$ and $u \, R \, t$.

It is easy to show that $>$ defined by 2.16 and 2.17 satisfies 2.11, and 2.17 was designed so that if $t \, C \, u$ then $t > u$.

<u>Theorem 5</u> If $>$ is a ground regular complexity measure then 1.36, the derived reduction algorithm, is refutation complete.

<u>Proof</u> We assume a ground regular complexity measure, $>$, $\approx$ which has been extended by a relation R satisfying 2.11 and 2.16, so that we may assume $>$ satisfies 2.17. Thus we may assume that $>$ satisfies 2.11 and 2.16. We then take a finite equality unsatisfiable set of clauses $\mathcal{S}$ and a finite equality unsatisfiable set of ground instances $\mathcal{G}$. Let $\mathcal{G}$ be divided into the reductions $R(\mathcal{G})_0$ and the remainder $\mathcal{G}_0$. Throughout we assume equations of the form $t = t$ are deleted. Now at the general level by 1.36 we have $\mathcal{R}_k$, $\mathcal{E}_k$, and $\mathcal{S}_k$. At this point each member of $\mathcal{G}_k$ is a substitution instance of a member of $\mathcal{S}_k$ while each member of $R(\mathcal{G})_k$ is an instance of a member of $\mathcal{R}_k \cup \mathcal{E}_k$. We also assume that each $\mathcal{E}_k$ is such that if $t = u$ is in $\mathcal{E}_k$ then its symmetric copy $u = t$ is there also. As redundancies are eliminated from $\mathcal{R}_k$ and $\mathcal{E}_k$ in 1.36 (2) the corresponding simplifications and deletions are made in $R(\mathcal{G})_k$. The ground

reductions that are used in making those simplifications are added to $R(\mathcal{G})_k$. As clauses of $\mathcal{S}_k$ are simplified in 1.36 (3) the corresponding simplifications of clauses of $\mathcal{G}_k$ are made, and the ground reductions that are used in making those simplifications are added to $R(\mathcal{G})_k$. As $\mathcal{R}_{k+1}$, $\mathcal{E}_{k+1}$, and $\mathcal{S}_{k+1}$ are formed in 1.36 (4), $R(\mathcal{G})_{k+1}$ is formed by performing the corresponding inferences at the ground level and in addition adding all those reductions of $R(\mathcal{G})_k$ which correspond to immediately reducing one of the terms of substitution which makes some member of $R(\mathcal{G})_k$ an instance of a member of $\mathcal{R}_k \cup \mathcal{E}_k$ (like was done in the formation of θ_2' in the proof of Lemma 4). Because the complexity of the left sides of each reduction that is added to $R(\mathcal{G})_k$ is less than or equal to some expression in $\mathcal{G}$, it follows that eventually no new additions are made to $R(\mathcal{G})_k$. It also can be shown that the terminal set $R(\mathcal{G})_T$ is a complete set of reductions. Moreover, $\mathcal{G}_T$ can be regarded as an intermediate step in the formation of $\mathcal{G}^*$ where $*$ is a simplification algorithm using $R(\mathcal{G})_T$. To complete the proof we modify the proof of Theorem 4: form $\mathcal{G}_T{}^*$ $(= \mathcal{G}^*)$ while simultaneously forming special inferences at the general level (along the lines of the proof of Lemma 2), so that the clauses of $\mathcal{G}_T{}^*$ are instances of $\mathcal{S}_{T+i}$ for some i, and by Theorem 1 obtain a refutation of $\Box$ from $\mathcal{G}_T{}^*$ by resolution which can be lifted by the ordinary lifting lemma for resolution.

CONCLUSION

We conclude with some questions and remarks which were suggested by the results of this paper.

1. Does there exist an algorithm which will decide whether or not a set of rewrite rules has the finite termination property?
2. If a set of rewrite rules does have the finite termination property, do there exist polynomial functions and a constant so that 1.28 will detect that fact? Is there an algorithm which will construct a collection of such polynomial functions when they exist?
3. Equations whose sides are identical up to permutation of variable symbols, such as commutative axioms, cannot be used as unrestricted rewrites without giving up finite termination. Can the notion of rewrite rule and simplification be enlarged in a non-trivial way to include permutation axioms?
4. Special paramodulation has been announced refutation complete as a positive solution to the functional reflexive problem. The status of the refutation completeness of special paramodulation should be settled at the earliest possible moment.
5. Closely related, is derived reduction refutation complete when equations occur in non-unit clauses?
6. Can one of the decision procedures for elementary algebra be used as an efficient basis for 1.31?

Are there decision procedures for 1.29 when the functions F_i of 1.28 are not polynomials, but from some other specified class?

7. How useful will sets of reductions be as part of a practical theorem prover? Many provers, such as the UT interactive prover of Bledsoe and Tyson (2), have long recognized the value of reduction and used sets of reductions in an ad hoc manner. With the systematic use of reduction we expect to see substantial improvement. Sets of reductions also occur naturally in various approaches to program verification, such as Boyer and Moore (3) and Horwitz and Musser (8). It should be determined if the methods of this paper facilitate these and similar approaches to program verification.

ACKNOWLEDGEMENTS

I would especially like to thank Prof. W. W. Bledsoe for his continued encouragement and support. I gratefully acknowledge the many long conversations with Prof. Dr. M. Richter which were responsible for the inception of many of these ideas. I express my sincere appreciation to Prof. N. Martin for his interest and helpful assistance. Finally, I would like to thank A. M. Ballantyne for introducing me to many of these ideas, and M. Moriconi for more reasons than I can recount.

REFERENCES

1. Anderson, R. and Bledsoe, W. W. A linear format for resolution with merging and a new technique for establishing completeness. JACM 17, 2 (July 1970), 525-534.

2. Bledsoe, W. W. and Tyson, M. The UT interactive theorem prover. Math. Dept. Memo ATP-17, The Univ. of Texas at Austin, May 1975.

3. Boyer, R. S. and Moore, J. S. Proving theorems about LISP functions. JACM 22, 1 (January 1975), 129-144.

4. Cohen, J. P. Decision procedures for real and p-adic fields. Comm. Pure and Appl. Math., XXII (1969), 131-151.

5. Collins, G. E. Quantifier elimination for real closed fields by cylindric algebraic decomposition. Proc. 2nd G I Conference on Automata and Formal Languages, Kaiserslauten, May 1975, Lecture Notes in Computer Science, Springer-Verlag (to appear).

6. Davis, M. Hilbert's tenth problem is unsolvable. Amer. Math. Monthly 80, 3 (March 1973), 233-269.

7. Herstein, I. N. Topics in Algebra, Blaisdell Publishing Co., New York, 1964, 32.

8. Horwitz, E. and Musser, D. R. The synthesis of algebraic specifications of data structures. Unpublished notes, USC Information Sciences Institute, Marina del Rey, Calif., 1975.

9. Jones, J. P. Recursive undecidability - an exposition. Amer. Math. Monthly 81, (Sept. 1974), 724-737.

10. Knuth, D. E. and Bendix, P. B. Simple word problems in universal algebras. Computational Problems in Abstract Algebras, J. Leech, Ed., Pergamon Press, 1970, 263-297.

11. Lankford, D. S. Canonical Algebraic simplification. Math. Dept. Memo ATP-25, The Univ. of Texas at Austin, May 1975.

12. Nevins, A. J. A human oriented logic for automatic theorem proving. JACM 21, 4 (October 1974), 606-621.

13. Robinson, G. A. and Wos, L. Paramodulation and theorem-proving in first-order theories with equality. Machine Intelligence 4, Edinburgh University Press, 1969, 135-150.

14. Robinson, J. A. A machine-oriented logic based on the resolution principle. JACM 12, 1 (Jan. 1965), 23-41.

15. Rosen, B. K. Tree manipulating systems and Church-Rosser theorems. JACM 20, 1 (Jan. 1973), 160-187.

16. Seidenberg, A. A new decision method for elementary algebra and geometry. Ann. of Math., Ser. 2, 60 (1954), 365-374.

17. Slagle, J. R. Automated theorem-proving for theories with simplifiers, commutativity, and associativity. JACM 21, 4 (October 1974), 622-642.

18. Tarski, A. A decision method for elementary algebra and geometry. Univ. of California Press, Berkeley, 1951.

19. Winker, S. Dynamic demodulation. Unpublished notes, Dept. of Comp. Sci., Northern Illinois Univ., August 1975.

20. Wos, L. and Robinson, G. The maximal model theorem. Spring 1968 meeting of Assn. for Symb. Logic.

APPENDIX

Many of the theoretical ideas contained in this paper have been implemented by Nevins (12). In particular, his treatment of equality is for the most part an implementation of derived reduction. The primary difference is that Nevins (12) did not treat associative axioms by reduction, but instead used an associative unification algorithm. We have discussed this difference and we believe that it accounts for much of the improvement in the $x^3 = 1$ group problem mentioned earlier. Another difference is that Nevins (12) did not have a complete set of reductions for groups and in particular used equation 1.25 as a rewrite in the opposite direction. But we believe that most of the improvement reported by Ballantyne and Lankford is due to the treatment of associativity. For the general predicate calculus Nevins (12) used a human-oriented system of natural deduction which incorporated reasoning by cases. We do not know of an analogy to reasoning by cases for resolution for which refutation completeness results are known, nor do we know of any refutation completeness results for reasoning by cases. But the notion of canonical inference, that is ordinary inferences followed by simplification with the intermediate simplifications discarded, is equally applicable to resolution based and natural based deductive systems. Nevins (12) did use canonical inference, and we believe that accounts for a substantial part of the power of his natural deduction program.

POSTSCRIPT

March 1978

Some of the questions we raised at the end of this paper have been subsequently answered. We summarize these recent results below.

1. There is no algorithm which decides finite termination.

   Huet, G. and D. Lankford. On the uniform halting problem for term rewriting systems, preliminary IRIA-Laboria and USC-ISI report, October 1977.

   Lipton, R. and L. Snyder. On the halting of tree replacement systems, Conference on Theoretical Computer Science, Univ. of Waterloo, July 1977.

2. There are rewrite rules with the finite termination property which cannot be detected by polynomials.

   Stickel, M. E. Personal communication.

3. Commutativity can be dealt with.

   Lankford, D. S. and A. M. Ballantyne. Decision procedures for simple equational theories with a commutative axiom: complete sets of commutative reductions, Automatic Theorem Proving Project, Univ. of Texas, Math. Dept., Austin, Texas, report #ATP-35, March 1977.

   Lankford, D. S. and A. M. Ballantyne. Decision procedures for simple equational theories with permutative axioms: complete sets of permutative reductions, report #ATP-37, April 1977.

   Lankford, D. S. and A. M. Ballantyne. Decision procedures for simple equational theories with commutative-associative axioms: complete sets of commutative-associative reductions, report #ATP-39, August 1977.

   Stickel, M. E. and G. Peterson. Complete sets of reductions for equational theories with complete unification algorithms, unpublished paper.

EXTENDED BIBLIOGRAPHY (CHURCH-ROSSER)

1. Newman, M. H. A. On theories with a combinatorial definition of "equivalence." Annals of Math. Vol. 43, No. 2, April 1942, 223-243.

2. Sethi, R. Testing for the Church-Rosser property. JACM 21, 4 (Oct. 1974), 671-679.

3. Staples, J. Church-Rosser theorems for replacement systems. Lecture Notes in Mathematics, A. Dold and B. Eckmann, Eds., Algebra and Logic, Papers from the 1974 Summer Research Institute of the Australian Mathematical Society, Monarch University, Australia, Springer-Verlag, 1975, 293-307.

4. Hindley, R. An abstract form of the Church-Rosser theorem. I The Journal of Symbolic Logic, Vol. 34, No. 4, Dec. 1969, 545-560.

5. Hindley, J. An abstract Church-Rosser theorem. II Applications JSL 39, (1 1974), 1-21.

6. Aho, A., Sethi, R., and Ullman, J. Code optimization and finite Church-Rosser systems. Courant Computer Science Symposium 5, March 1971, Design and Optimization of Compilers, R. Rustin, Ed., Prentice-Hall, Inc., 1972, 88-105.

7. Huet, G. Confluent reductions: abstract properties and applications to term rewriting systems. Proceedings of 18th IEEE Symposium on Foundations of Computer Science, Oct. 1977. (Also as an IRIA Laboria report.)

8. Musser, D. Convergent sets of rewrite rules for abstract data types. USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90291, to appear as an ISI report.

9. Musser, D. A data type verification system based on rewrite rules. Proc. 6th Texas Conference on Computing Systems, Austin, Texas, No. 1977.

10. Bücken, H. Reduction systems and small cancellation theory. Proc. 4th Conference on Automatic Deduction, Feb. 1979, 53-59.

11. Ballantyne, A. and Lankford, D. New decision algorithms for finitely presented commutative semigroups. Louisiana Tech Univ., Math. Dept., Ruston, LA 71272, report MTP-4, May 1979.

12. Bergman, G. The diamond lemma in ring theory. Advances in Math. 29, 1978, 178-218.

EXTENDED BIBLIOGRAPHY (FINITE TERMINATION)

1. Lankford, D. A finite termination algorithm. Southwestern University, Math. Dept., Georgetown, Texas, local report, March 1976.

2. Dershowitz, N. and Manna, Z. Proving termination with multiset orderings. Stanford Artificial Intelligence Laboratory Memo AIM-310 (1978).

3. Manna, Z. and Ness, S. On the termination od Markov algorithms. Proc. 3rd Hawaii International Conference on System Sciences, 1970.

4. Plaisted, D. A recursively defined ordering for proving termination of term rewriting systems. Univ. of Illinois Urbana-Champaign, Urbana, Illinois, 61801, report UIUCDCS-R-78-943, Sept. 1978.

5. Plaisted, D. Well-founded orderings for proving termination of systems of rewrite rules. report UIUCDCS-78-932, July 1978.

6. Dershowitz, N. A note on simplification orderings. Univ. of Illinois at Urbana-Champaign, Dept. of Comp. Sci., Urbana, ILL 61801, April 1979.

7. Dershowitz, N. Orderings for term-rewriting systems. Univ. of Illinois at Urbana-Champaign, Dept. of Comp. Sci., Urbana, ILL 61801, Aug. 1979.

8. Lankford, D. On proving term rewriting systems are Noetherian. Louisiana Tech Univ., Math. Dept., Ruston, LA 71272, MTP-3, May 1979.

9. Goguen, J. How to make rewrite rules converge when they don't want to. Unpublished memo, March 1979.

ADDENDUM *

1/14/80

The proof of Theorem 3 is more complex than is necessary for the following reason. The proof consists of two parts--the completion of a finite set of ground rewrite rules, which therefore decides the uniform word problem for finitely presented algebras, and the lifting of that ground decidability result. At the time "Canonical inference" was written, I had not carefully thought about the completion procedure for finite sets of ground rewrite rules. Much later, I noticed that the superposition step (critical pair step) was not required for ground rewrite rules, but that the redundancy elimination step was enough to derive complete sets of ground reductions. Moreover, if the redundancy elimination procedure alone is used (in conjunction with the lexical order), then it easily follows that the uniform word problem for finitely presented algebras is decidable in constant space. Although I have not checked all the details, it seems to follow that the redundancy elimination procedure also solves the uniform word problem for finitely presented algebras in $O(n^3)$ time.

The decidability of the uniform word problem for finitely presented algebras goes at least back to Ackermann [1954], though he did not give a practical algorithm. Several other algorithms have been given by Kozen [1977] (polynomial time),

* to "Canonical inference"

Nelson and Oppen [1979] ($O(n^2)$ time) and Downey, et al. [1979] ($O(n\log^2 n)$ time). Both algorithms require $O(n)$ space. Shostak [1977] also develops a similar algorithm, but we do not know complexity results for his algorithm. Presumably it is slower than the other two mentioned above.[9]

Experimental evidence for the Nelson and Oppen [1979] procedure (in the Stanford Pascal Verifier) indicates the approach is practical. However, we do not agree with their conclusion that a fast congruence closure algorithm is the best method available for handling equalities in mechanical theorem provers. An equally convincing body of experimental evidence suggests that term rewriting methods are the best methods available for handling equalities in mechanical theorem provers. As we have said, the uniform word problem for finitely presented algebras has a constant space solution by rewrite rule methods. And we think it is fair to say that time is cheaper than space. Another advantage of the rewrite rule methods is that they "lift" to the general level in a particularly nice way, i.e., to Church-Rosser decision algorithms for uniformly terminating term rewriting systems. Also, the term rewriting methods can be combined with inference rules, like resolution, to form refutation complete procedures for the first order predicate calculus with equality. Experiments with this approach by Lankford and Ballantyne [1979] suggest that rewrite rule methods are the best available for treating equality in a

[9] **rumored exponential**

mechanical theorem prover for the first order logic with equality. Finally, the rewrite rule methods are very "natural" in the sense that they closely approximate what a human mathematician might do, i.e., simplify and/or rewrite equations. Thus, mechanical proofs based on rewrite rule methods are easier to read (which must be done to verify that the mechanical proof is indeed a proof).

It may very well be that for finitely presented algebras in particular and ground theories with equality in general the fast congruence closure method is more efficient, especially if its linear space complexity grows slowly. But as a general level method we believe that fast congruence closure is clearly in second place at this time behind term rewriting methods.

It has recently been my good fortune to receive a number of papers and correspondence from Trevor Evans concerning term rewriting methods based on the diamond lemma (Newman [1942]). I had not realized before the debt which the Knuth and Bendix approach owes to Evans [1951a], which contains solutions for the uniform word problem for finitely presented loops and other non-associative algebras by term rewriting methods based on the diamond lemma. In retrospect one can see in Evans [1951a] the inception of the Church-Rosser (critical pairs) algorithm (see the large table of what are more or less the critical pairs for loops). Other solutions given by Evans [1951a] are not the complete sets of reductions which one would get by running the Knuth and Bendix completion procedure on a given presentation where rewrite rules are determined on the basis of some uniform termination test, but are special complete sets of reductions where all ground rules have the form $f(a,b) \longrightarrow c$ where a, b, and c are generators and f is one of the operators. However, we believe that the Knuth and Bendix completion procedure would halt uniformly on finitely presented loops and other non-associative algebras with the lexical order used to determine rewrite rules.

The special rewrite rules mentioned above (when considered as equations) are called closed sets of relations by Evans [1951b], who establishes some very general hypotheses under which the word problem can be solved. To a first approximation, closed sets of rewrite rules are rules of the form $f(t_1,\ldots,t_k) \longrightarrow t$ where $t_1, \ldots, t_k$ and t are constants. For example, the uniform word problem for finitely presented algebras can be solved by complete sets of closed rewrite rules as follows. For each relation of the form $f(t_1,\ldots,t_i) = g(u_1,\ldots,u_j)$ one introduces a new constant c and replaces the relation by the two rewrite rules $f(t_1,\ldots,t_i) \longrightarrow c$ and $g(u_1,\ldots,u_j) \longrightarrow c$. Then for each rewrite rule $f(t_1,\ldots,t_i) \longrightarrow c$ and each argument t_j which is not a constant, one introduces a new constant d and replaces $f(t_1,\ldots,t_i) \longrightarrow c$ by the two

$f(t_1,\ldots,d,\ldots,t_i) \longrightarrow c$ and $t_j \longrightarrow d$. Continuing in this way, one eventually gets a closed set of rewrite rules. The Knuth and Bendix procedure, when applied to this closed set of rewrite rules, (or perhaps we should say the redundancy elimination procedure) terminates with a complete set of reductions that consists of two parts, a complete set of closed reductions and a complete set of reductions all of the form $c_i \longrightarrow d_j$ where c_i and d_j are constants.

Other results based on term rewriting methods include, a solution of the uniform word problem for finitely presented trees, see Evans [1963b] (this also contains some of the early work on equivalence class term rewriting systems); a solution of the uniform word problem for finitely presented Steiner loops, see Treash [1969]; and additional very general hypotheses under which algebras have a solvable word problem, see Evans [1969], Evans, et al. [1975] and Evans [1978a]. Two useful survey papers are Evans [1976] and Evans [1978b].

REFERENCES

Ackermann, W. [1954] Solvable Cases of the Decision Problem, North-Holland, Amsterdam.

Downey, P., Sethi, R. and Tarjan R. Variations on the common subexpression problem. to appear JACM.

Evans, T. [1951a] On multiplicative systems defined by generators and relations, I. Normal form theorems. Proc. Cambridge Philos. Soc. 47, 637-649.

Evans, T. [1951b] The word problem for abstract algebras. J. London Math. Soc. 26, 64-71.

Evans, T. [1963a] The isomorphism problem for some classes of multiplicative systems. Trans. Amer. Math. Soc. 109, 303-312.

Evans, T. [1963b] A decision problem for transformations of trees. Canadian J. Math. 15, 584-590.

Evans, T. [1969] Some connections between residual finiteness, finite embeddability and the word problem. J. London Math. Soc. 2, 1, 399-403.

Evans, T. [1976] Some solvable word problems. Proc. Conference on Decision Problems in Algebra, Oxford, July 1976, to appear North-Holland, Amsterdam.

Evans, T. [1978a] An algebra has a solvable word problem if and only if it is embeddable in a finitely generated simple algebra. Algebra Universalis 8, 197-204.

Evans, T. [1978b] Word problems. Bull. Amer. Math. Soc. 84, 5, 789-802.

Evans, T., Mandelberg, K. and Neff, M. Embedding algebras with solvable word problems in simple algebras-some Boone-Higman type theorems. Proc. Logic Colloq., Univ of Bristol, July 1973, North-Holland, Amsterdam, 1975, 259-277.

Kozen, D. [1977] Complexity of finitely presented algebras. Ninth ACM Symposium on Theory of Computing, 164-177.

Lankford, D. and Ballantyne, A. [1979] The refutation completeness of blocked permutative narrowing and resolution. Fourth Conference on Automated Deduction, Austin, Texas, 168-174.

Nelson, C. and Oppen, D. [1979] Fast decision algorithms based on congruence closure. to appear JACM.

Newman, M. [1942] On theories with a combinatorial definition of "equivalence." Annals of Math. 43, 2, 223-243.

Shostak, R. [1977] An algorithm for reasoning about equality. IJCAI-77, also CACM, July 1978, 583-585.

Treash, C. [1969] Inverse property loops and related Steiner triple systems. Ph.D. thesis, Emory U., Atlanta, GA.

ADDENDUM

2/13/85

As the transparencies fof our talk at the Fourth Workshop on Automated Deduction (Austin, Feb. 1979) were being written, we noticed that the blocking refinement can be further refined as follows, cf. the transparencies for additional details. Let a clause C (literal L, atom A, term T, etc.) be defined as a clause-substitution pair C , θ (L , θ ; A , θ ; T , θ), where initially the substitution is the empty substitution. As clauses are reduced by a complete set during narrowing, or tested for irreducibility during resolution, the current substitutions are checked to see if they are irreducible. If not, the resolvent, narrowing, etc. is not kept. This potentially reduces the search space further. We might call this refinement blocking with history. No computer experiments have been performed with blocking with history. Other variations are also suggested, e.g., blocked completion, etc. See also footnote 7, page 24 of this article.