INFORMATION AND CONTROL 65, 182–215 (1985)

Semantic Confluence Tests and Completion Methods

DAVID A. PLAISTED*

Department of Computer Science, New West Hall 035A, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27514

We present semantic methods for showing that a term-rewriting system is confluent. We also present methods for completing a given term-rewriting system to obtain an equivalent confluent system. These methods differ from the well-known and widely studied Knuth–Bendix method in that they emphasize semantics rather than syntax. Also, they often require more user interaction than the purely syntactic Knuth-Bendix method. The concept of "ground confluence" is discussed; methods for demonstrating ground confluence are also given. We give decision procedures for some sub-problems that arise in this method. © 1985 Academic Press, Inc.

1. INTRODUCTION

The Knuth–Bendix method (Knuth and Bendix, 1970) [24] has become standard for testing confluence of term rewriting systems and for completing term rewriting systems to obtain equivalent confluent systems. Extensions of this method to equational theories have been studied in (Huet, 1980; Jouannaud, 1983; Lankford and Ballantyne, 1977). The REVE term rewriting laboratory (Lescanne, 1983) is largely based on the Knuth–Bendix method and various extensions. Stickel (1984) has recently shown the power of the Knuth–Bendix method for theorem proving in purely equational theories. Hsiang and Dershowitz (1983) have shown how to extend the Knuth–Bendix method to obtain a general first-order predicate calculus theorem prover. For examples of the working of the Knuth–Bendix confluence test and completion method, see (Hullot, 1983; Peterson and Stickel, 1981). However, there are some problems with the Knuth–Bendix approach. The use of equations can be cumbersome unless the equational theory has a decidable unification problem, such as for associative–commutative operators (Fages, 1984; Stickel,1981). This is not

* Much of this work was done while the author was at the Department of Computer Science of the University of Illinois at Urbana-Champaign. This work was partially supported by National Science Foundation Grants MCS 81-09831 and MCS 83-07755.

0019-9958/85 $3.00

true for many cases of practical interest, such as systems containing an associative but not commutative operator; it is known (Plotkin, 1972) that no finite unification algorithm exists for such systems. For a discussion of the extension of the Knuth–Bendix method to equational systems, see (Jouannaud, 1983; Jouannaud and Kirchner, 1984). Furthermore, the extension of the Knuth–Bendix method to conditional rewrite rules is difficult. Some progress in this area has been reported in (Bergstra and Klop, 1982; Kaplan, 1984; Remy, 1982, 1983; Remy and Zhang, 1984). We present a new confluence test which avoids many of these drawbacks. It can be used when no finite unification algorithm exists for the equational part of the theory; in fact, it can even be used when no finite matching algorithm exists for the set of equations. The extension to conditional rewrite rules is straightforward. The method makes use of semantic concepts in addition to syntactic concepts. As a result, more user interaction is required than for the Knuth–Bendix approach, and more computation time may be required with this new method. The method builds on the "partial correctness" and "semantic confluence" ideas presented in (Plaisted, 1980); the partial correctness idea has been further investigated in (Dershowitz, 1983; Thiel, 1984). Also, we distinguish "ground confluence" from "confluence" and give methods for testing ground confluence; we argue that ground confluence is more relevant in some cases. Methods for extending non-confluent term rewriting systems to confluent systems are also described.

2. Definitions

We consider the set of terms composed of variables and operators (function symbols) in some finite set F of function symbols. Let $T(F, X)$ be the set of well-formed terms containing variables in X and function symbols in F. Each function symbol in F has an *arity* which specifies how many arguments it may have. Let R be a set of rewrite rules $\{r_i \rightarrow s_i\}$ indicating that instances of r_i may be replaced by corresponding instances of s_i; for an introduction to the theory of such rules see (Huet and Oppen, 1980). We require that all variables in s_i must also occur in r_i. Many of the following results still are valid if R is a set of *conditional* rewrite rules, that is, rules of the form $r_i \rightarrow s_i$ if C_i. Such a conditional rule may reduce instances of r_i to instances of s_i if the condition C_i is true. Let E be a set of equations. We say term t *rewrites to* u using R, $t \Rightarrow_R u$, if u may be obtained from t by replacing a subterm as specified by R. If t and w are terms, we say $t =_E w$ if the equation $t = w$ is a logical consequence of E. We say t rewrites to u using (R, E), $t \Rightarrow_{R,E} u$, if there exists a term w such that $t =_E w$ and $w \Rightarrow_R u$. Also, (R, E) is *terminating* if there is no infinite sequence

$t_1 \Rightarrow_{R,E} t_2 \Rightarrow_{R,E} t_3 \Rightarrow_{R,E} \cdots$. For methods of proving termination see (Dershowitz, 1982; Jouannaud, Lescanne, and Reinig, 1982; Lankford, 1979; Plaisted, 1978). From now on we often write $\Rightarrow$ instead of $\Rightarrow_{R,E}$. We write $s \Rightarrow *t$ if s rewrites to t using zero or more replacements according to (R, E). The system (R, E) is *confluent* if for all terms t, if $t \Rightarrow *u$ and $t \Rightarrow *w$ then there exist terms u_1 and w_1 such that $u \Rightarrow *u_1$ and $w \Rightarrow *w_1$ and $u_1 =_E w_1$. Let G be the set of ground terms (i.e., terms which contain no variables). The system (R, E) is *ground confluent* if for all ground terms t, if $t \Rightarrow *u$ and $t \Rightarrow *w$ then there exist terms u_1 and w_1 such that $u \Rightarrow *u_1$ and $w \Rightarrow *w_1$ and $u_1 =_E w_1$. Frequently ground confluence is of interest. For example, (R, E) may represent a program, and it may be known that all inputs to this program will be ground terms. Then ground confluence insures that, regardless of the order in which rules are applied, the final result will be the same (up to E equivalence), if (R, E) is terminating. Sometimes we are interested in showing that (R, E) is confluent for a different set E_1 of equations. Therefore we have the following definitions. The system (R, E) is *E_1-confluent* if for all terms t, if $t \Rightarrow *u$ and $t \Rightarrow *w$ then there exist terms u_1 and w_1 such that $u \Rightarrow *u_1$ and $w \Rightarrow w*_1$ and $u_1 =_{E_1} w_1$. The system (R, E) is *E_1-ground confluent* if for all ground terms t, if $t \Rightarrow *u$ and $t \Rightarrow *w$ then there exist terms u_1 and w_1 such that $u \Rightarrow *u_1$ and $w \Rightarrow *w_1$ and $u_1 =_{E_1} w_1$. A term is *reducible* in (R, E) if there exists u such that $t \Rightarrow u$; otherwise t is *irreducible* in (R, E). Note that (R, E) reducibility is partially decidable; if t is reducible, we can enumerate all terms u and eventually find one such that $t \Rightarrow u$, and verify this fact. This verification step requires non-trivial rewriting using E. Also, (R, E) reducibility is still partially decidable for conditional term rewriting systems, since we can verify that appropriate instances of the conditions C_i are true in the theory T (see below). If S is a set of terms, let $R^E(S)$ be $\{t\colon (\exists s \in S) s \Rightarrow *t$ and t is irreducible$\}$. We write $R(S)$ for $R^E(S)$ when E is empty. Let $rts(R, E)$ be the set of terms that are irreducible in (R, E). If (R, E) is terminating and confluent then we say (R, E) is *canonical*. In this case, if s is a term then $R^E(\{s\})$ is an E-equivalence class of terms called the (R, E) *normal form* of s.

Let T be a *theory* ; that is, a set of formulae that are provable in some logical system. Sometimes we refer to the set of formulae that hold in (are true in) a particular model, as a theory. Informally, we say a statement is true in theory T or valid in T if the statement is a member of T. We often represent a theory T (which may be an infinite set of formulae) by a finite set A of axioms. Usually, the theory represented is the set of formulae derivable from A in first-order logic (Manna, 1974). We say M is the *initial* model of A if for all predicates P and ground terms t_i, $P(t_1, ..., t_n)$ holds in M iff $A \models P(t_1, ..., t_n)$. Such a model does not always exist. The *initial* theory of A is the set of formulae that hold in (are satisfied by) the initial model of the axioms, if it exists. We are interested in obtaining term rewriting systems that are decision procedures for various first-order and initial theories. Note that finitely axiomatized first-order theories are recursively enumerable. We assume T is consistent. We say (R, E) is *valid* in T if R and E are subsets of T; that is, every rule $r \to s$ of R, when regarded as the equation $r = s$, and every equation of E, is in the theory T. For conditional rules $r \to s$ if C we require that $C \supset r = s$ be in T. Note that if T is a first-order theory, it is partially decidable whether (R, E) is in T. Also, (R, E) is *ground valid* in T if every ground instance $r \to s$ of a rule of R, when regarded as the equation $r = s$, and every ground instance of an equation of E, is true in T. For conditional rules $r \to s$ if C we require that all ground instances of $C \supset r = s$ be true in T. We say T is *E-separating* on a set S of terms if for all terms t and u in S, $t =_E u$ iff $t = u$ is true in T. If E is empty, we say T is separating or ϕ-separating rather than E-separating. Also, T is *ground E-separating* on a set S of terms if for all ground terms t and u in S, $t =_E u$ iff $t = u$ is in T. We say $t = u$ holds (is satisfied) in a model M if for all assignments of values to variables, t and u have as value the same domain element of the model. If T is E-separating on S, and if t and u are terms which are not E-equal, then there is a model of T in which t and u have distinct interpretations. A similar statement applies to ground E-separation. This is the reason for the term "E-separating." In both cases it is possible that different models of T may be used to give different sets of pairs (t, u) different interpretations.

For conditional rules R, note that R-reducibility is partially decidable assuming membership in T is. Also, R-reducibility is decidable if membership in T is decidable. For equational rewriting, (R, E)-reducibility is partially decidable, assuming membership in T is partially decidable. Also, (R, E)-reducibility is decidable if the E-equivalence classes are finite and membership in T is decidable. To decide if term s is (R, E)-reducible, it suffices to enumerate all terms that are E-equivalent to s and test each one for R-reducibility. Note that E may have an infinite number of finite equivalence classes. In many cases we would like to be able to decide if a condition C is true simply by reducing it using (R, E) and seeing if it reduces to TRUE. We do not have time or space here to discuss when this is complete. However, the following observation frequently is enough:

Suppose R is a conditional term rewriting system. Suppose certain non-primitive operators can always be eliminated by R. That is, any expression containing such an operator can always be reduced. Suppose R-reduction always terminates. Then all occurrences of non-primitive operators in the conditions can be eliminated by rewriting. Thus the conditions can be reduced to simpler form, for which a decision procedure may be available, or for which other methods may be used to show that all T-valid conditions reduce to TRUE.

For example, we shall later discuss a conditional system containing the rules

$$
\begin{aligned}
&\text{MIN(CONS}(A, \text{NIL})) = A\\
&\text{MIN(CONS}(A, L)) \quad = A \text{ IF LESS}(A, \text{MIN}(L))\\
&\text{MIN(CONS}(A, L)) \quad = \text{MIN}(L) \text{ IF NOT LESS}(A, \text{MIN}(L))
\end{aligned}
$$

In this system, all occurrences of MIN in the conditions have arguments that are smaller than the occurrences of MIN in the left-hand sides, in the recursive path ordering (Dershowitz, 1982) [5] with MIN as the maximal operator. Therefore for ground terms these occurrences of MIN may be evaluated recursively using the same set of equations. For ground terms, this reduces the conditions to the form LESS(A, B) or NOT LESS(A, B), where A and B are integers, which can be evaluated by a specialized decision procedure.

There is more than one choice for the definition of reducibility for a canonical term rewriting system, as mentioned in (Brand, Darringer, and Joyner, 1978). It could be, for example, that s reduces to t if condition C_1 is true using one rule, and s reduces to t if condition C_2 is true, using another rule. If $C_1 \vee C_2$ is true in T, does s reduce to t? We have chosen to say no for simplicity, although either viewpoint could be adopted.

3. Demonstrating Confluence

The following results are the basis of our investigation:

THEOREM 3.1. *If (R, E) is terminating and valid in T and T is E_1-separating on* rts(R, E) *then (R, E) is E_1-confluent.*

Proof. Suppose $s \Rightarrow^* t$ and $s \Rightarrow^* u$. Then since (R, E) is terminating there exist irreducible t_1 and u_1 such that $t \Rightarrow^* t_1$ and $u \Rightarrow^* u_1$. Also, since (R, E) is valid in T, $t_1 = u_1$ is true in T. Since T is E_1-separating on rts(R, E), $t_1 =_E u_1$.

COROLLARY. *If R, E, E_1, and T satisfy the hypothesis of theorem, then $s = t$ is true in T iff the (R, E) normal forms s_1 and t_1 of s and t are E_1-equivalent, that is, E_1 implies $s_1 = t_1$. Also, if E_1 equality is decidable, then it is decidable whether $s = t$ in T, and the term rewriting system (R, E) gives such a decision procedure.*

THEOREM 3.2. *If (R, E) is terminating and ground valid in T and T is E_1-separating on $R^E(G)$ then (R, E) is E_1-ground confluent.*

Proof. (Recall that G is the set of ground terms.) Similar to the above, using the fact that if $s \Rightarrow {}^*t$ and s is a ground term, then t is also a ground term, by the restriction on variables on the right-hand side of a rewrite rule.

COROLLARY. *If R, E, E_1, and T satisfy the hypotheses of the theorem, and s and t are ground terms, then $s = t$ is true in T iff the (R, E) normal forms s_1 and t_1 of s and t are E_1-equivalent, that is, E_1 implies $s_1 = t_1$. Also, if E_1 equality is decidable for ground terms, then it is decidable for ground terms s and t whether $s = t$ is in T, and the term rewriting system (R, E) gives such a decision procedure.*

Note that many theories of interest already have decision procedures, at least for the quantifier free part (Nelson and Oppen, 1980; Oppen, 1980); for such theories, checking the validity of (R, E) is much easier. Frequently we are interested in whether a system of rewrite rules is valid in some given theory T; the standard Knuth–Bendix method does not approach this problem directly. It might be interesting to see where the properties of confluence and ground confluence are in the arithmetic hierarchy (Yasuhara, 1971); the same question could be asked for other of the properties we investigate. Our methods is related to the concept of semantic confluence, defined as follows:

DEFINITION. A term rewriting system (R, E) is *semantically confluent* for an interpretation I if for any two terms t_1 and t_2 such that t_1 and t_2 have the same value in I, there are terms u_1 and u_2 such that $t_1 \Rightarrow u_1$ and $t_2 \Rightarrow u_2$ and $u_1 =_E u_2$.

3.1 *Example*

Consider the following example, with arities specified as in OBJ (Goguen, Meseguer, and Plaisted, 1982). (For a more recent version of OBJ and a discussion of term rewriting issues see Futatsugi, Goguen, Jouannaud, and Meseguer, 1985):

```
SORTS
  INT BOOL LIST-OF-INTS
OPS
  ZERO: → INT
  SUCC: INT → INT
  MIN: LIST-OF-INTS → INT
  CONS: INT LIST-OF-INTS → LIST-OF-INTS
  NIL:  → LIST-OF-INTS
  LESS: INT INT → BOOL
```

REDUCTIONS
  MIN(CONS(A, NIL)) = A
  MIN(CONS(A, L)) = A IF LESS(A, MIN (L))
  MIN(CONS(A, L)) = MIN(L) IF NOT LESS(A, MIN(L))

We want to show that this system is ground confluent. This example causes a problem for the conditional approach of Remy and Zhang (1981) [41] because the symbol MIN occurs both in equations and conditions. Intuitively, showing confluence should be easy because MIN of a list reduces to its minimal element, and this minimal element is unique. The semantic confluence approach tries to capture this intuition as to why the system is confluent. We show informally how this example may be treated in the semantic confluence approach.

First, we show that the system (R, E) is terminating, with E empty. To show termination if suffices to find a simplification ordering (Dershowitz, 1982) in which for all rules ($r \rightarrow s$ if C) in R, r is greater than s and r is greater than C. Without going into details, the recursive path ordering of op. cit suffices, with the partial ordering on operators in which MIN is maximal and all other operators unrelated. Next, it is necessary to show that if a ground term s is reduced to an irreducible term t, then t contains no occurrences of MIN. (A nicer result would be to show that if s is MIN(L), where L is of sort LIST-OF-INT, then s reduces to an integer.) To show this, we examine the possible terms containing MIN. We are using techniques from (Dershowitz, 1983; Plaisted, 1980; Thiel, 1984) here. We can assume inductively that MIN does not occur in any subterm of s. Thus s is of the form MIN(L), where L is a list of integers. Now, either L is NIL or L is of the form CONS(A, L). If L is NIL, then MIN(L) does not reduce in the above system. Thus the user would be given the term MIN(NIL) and asked for an irreducible form of it. Suppose the user said "INFINITY." We then modify the above description as follows and start again:

SORTS
  INT+ BOOL LIST-OF-INT+
OPS
  ZERO: → INT+
  SUCC: INT+ → INT+
  INFINITY: → INT+
  MIN: LIST-OF-INT+ → INT+
  CONS: INT+ LIST-OF-INT+ → LIST-OF-INT+
  NIL: → LIST-OF-INT+
  LESS: INT+ INT+ → BOOL
REDUCTIONS
  MIN(NIL) = INFINITY

MIN(CONS(A, NIL)) = A
MIN(CONS(A, L)) = A IF LESS(A, MIN(L))
MIN(CONS(A, L)) = MIN(L) IF NOT LESS (A, MIN(L))

Here INT+ is INT $\cup$ {INFINITY}. The equation MIN(CONS(A, NIL)) = A is now redundant, so we may delete it if desired. By case analysis, we show that all ground terms of form MIN(L) are reducible if L has no occurrences of MIN. Now, any such L will either be NIL or of the form CONS(A, M). If L is NIL then MIN(L) reduces to INFINITY; if L is of the form CONS(A, M) then MIN(L) reduces to A if LESS(A, MIN(M)) and to MIN(M) if NOT LESS(A, MIN(M)). Using a theorem prover, we can verify that the disjunction of the cases LESS(A, MIN(M)) and NOT LESS(A, MIN(M)) is TRUE. Thus all ground terms of the form MIN(CONS(A, M)) reduce. We have done both cases, and so all ground terms of the form MIN(L) reduce if MIN does not occur in L. Therefore MIN does not occur in any irreducible term.

We now need to show that (R, E) is ϕ-separating (separating) on $R^E(G)$, that is, on the set of irreducible ground terms. (Recall that E is empty here.) Let T be the first-order theory of $R \cup E \cup Z$ where Z is an axiomatization of LESS on integers. That is, T is the logical consequences of R, E, and Z, where rules in R are regarded as equations. Let T' be the first-order theory of $R' \cup E' \cup Z$, where R' and E' are R and E with all equations involving MIN omitted. Also, MIN is not an operator of T'. Then T' has no non-trivial equations between terms in $R^E(G)$, and therefore T' is separating on $R^E(G)$. Then we need to show that the definition of MIN is a "proper definition," that is, the definition of MIN does not introduce any new equalities on the domain of T'. From this it follows that any model of T' may be extended to a model of T without introducing any new equalities on the domain of T'. For this, it suffices to show that the definition of MIN is "non-overlapping" in a sense. (For another treatment of the non-overlapping property in term rewriting systems see Hoffman and O'Donnell, 1984.) That is, MIN is not defined twice on the same term, and MIN(t) is always defined in terms of MIN(u), where u is less than t in some well founded ordering. To show the non-overlapping property it is useful to omit the equation MIN(CONS(A, NIL)) = A. This equation can be added later, it turns out, since it is a logical consequence of other equations. Also, to show that the last two equations are non-overlapping, it is necessary to show that the two conditions are mutually exclusive in T', which may be done using a theorem prover. To show that the first equation MIN(NIL) = INFINITY does not overlap with the others, we need to show that NIL and CONS(A, L) are never the same. For this it suffices to take the initial theory for the theory of list structures. That is, we add to Z the infinitely many assertions that $s \neq t$ if s and t are distinct terms built up from CONS and NIL and integers. Without some restriction the semantic confluence property fails (although the system may still be confluent). For example, if NIL = CONS(A, NIL) in some model, then MIN(NIL) reduces both to INFINITY and A in that model. Also, if NIL = CONS(A, NIL) in some model, then the definition of MIN forces INFINITY = A and therefore is not a proper definition if A is not INFINITY. Similarly, we need to know L is less than CONS(A, L) in some ordering to insure that the definitions are not circular. The propriety of definitions could be violated, for example, if we defined LENGTH by

LENGTH(CONS(A, L))1 + LENGTH(L)
LENGTH(NIL) = 0

and in some model CONS(A, L) = L. We would then get the contradiction LENGTH(L) = LENGTH(L) + 1 which means that the definition of LENGTH would eliminate some models and thus would be an improper definition. However, if we take the initial theory of list structure then such circularity is avoided. This completes the demonstration of confluence.

3.2. *General Features of the Method*

We consider some of the characteristics of the semantic confluence method and also compare it to some related approaches, before giving more technical details in following sections.

3.2.1. *Hierarchical Confluence Proofs*

The advantage of hierarchical specifications is now generally recognized. The semantic confluence method permits the proof of confluence to be likewise hierarchical. Furthermore, this method permits proofs of confluence even when parts of a term-rewriting system are only partially specified. Then any sub-system which fits the partial specification will, together with the rest of the system, constitute a confluent term-rewriting system.

For example, suppose that we have a term rewriting system for factoring positive integers. Suppose that all we know about it is that it terminates, is correct with respect to arithmetic, and that, given a composite integer n represented as $S^n(0)$, this integer is reduced to some product of integers (not necessarily prime). Then we know this system is confluent up to associativity and commutativity of multiplication, since (a) it is correct for the theory of arithmetic, (b) any composite integer is reduced, (c) we have the unique factorization theorem for natural numbers, and (d) the system terminates. Actually, we are extending the semantic confluence theory here to show that a term rewriting system is confluent on a subset of the starting terms, in this case on product of integers, but this extension is not difficult to do. Let R be the system; then R is E-separating on S, where S is products of primes and E is associativity and commutativity of multiplication. Thus any term-rewriting system satisfying this incomplete specification is confluent on the specified set of starting terms. In fact, any procedure which takes an input term and produces an output term, consistent with the specification, is similarly confluent. Thus we can prove confluence of procedures which are not expressed as sets of rewrite rules at all. This gives a convenient, semantically well-defined way to combine procedures in some high level language, with rewrite rules. Note that the Knuth–Bendix method, being entirely syntactic in nature, cannot handle sub-systems or procedures which are incompletely specified. The factorization system might be a part of a larger specification.

As another example, we may take a sub-system which sorts a list. Since the sorted form of a list is unique, it does not matter how it is done from an abstract semantic point of view. The semantic confluence method can be used to show systems confluent which have arbitrary sub-systems for sorting. However, Knuth–Bendix needs to know the exact syntactic form of the set of rewrite rules for sorting, and so cannot be used until the subsystem is known exactly.

3.2.2. *Relation to Algebraic Approaches*

The semantic confluence approach has interesting relations to algebraic approaches. For example, the inductionless induction method of (Goguen, 1980; Huet and Hullot, 1982; Musser, 1980) uses confluence to show an equation holds in an initial model; the semantic confluence method uses validity in a theory T to show confluence. There are also relations to "sufficient completeness" (Guttag and Horning, 1978). The fact that certain operators can be eliminated (like MIN above) is similar to sufficient completeness. In fact, the inductionless induction method of (Huet and Hullot, 1982) makes use of concepts of sufficient completeness. We later give some decision procedures that can be applied to the inductionless induction method of ibid, although they may not be very efficient. However, semantic confluence is more general than sufficient completeness; sometimes it not only eliminates certain operators, but eliminates certain combinations of operators. For example, the irreducible terms may be polynomials. The set of polynomials cannot be obtained from the set of arithmetic expressions simply by eliminating certain operators. A more delicate analysis is necessary.

The E-separation property may be given an algebraic interpretation in the following way: Say an algebra B is a *restriction* of algebra A if A and B are the same except that B has fewer operators. In this case, let us say that A is an *enrichment* of B.

PROPOSITION 3.3. *Suppose theory T over set F_1 of operators is E-separating for S, E is valid in T, S is the set of ground terms of T over set F_2 of operators, and $F_2 \subset F_1$. Suppose T has an initial model A, and let B be the restriction of A to the operators in F_2. Then T is ground E-separating for S iff B is an initial algebra for E.*

For example, we may have a theory of lists and natural numbers, in which we may reason about the lengths of lists. If we restrict this theory to omit natural numbers, we get the theory of lists, which is a free theory and is thus initial for E empty, even though the original theory has equations about natural numbers.

An advantage of the semantic confluence method is that the theory T is separated from the term rewriting system R. In the initial algebra approach, T and R are the same. In the semantic confluence approach, one can imagine a specification as having two parts, a theory part and a term-rewriting system part. The theory part would not need to be restricted to equational or even first-order logic, and could contain non-Horn clauses, for example. Also, it is not necessary that T have an initial model. Thus T can contain objects such as infinite sets which may be difficult to represent concretely; R need only contain objects which are necessary for the computation. Furthermore, we shall see below that the "propriety of definition" mentioned above need only hold for T; R may contain rules that violate sufficient conditions for proper definitions, as long as these rules are logical consequences of T. One might use an interactive theorem prover to show that R is valid in T. Such interaction does not seem possible with the Knuth–Bendix method, except in the choice of an ordering on operators for purposes of proving termination. Of course, we must mention that the Knuth–Bendix method is simpler than ours and is more completely automated, and so has important advantages in the cases in which it works.

3.3 *Forbidden Subterms*

In order to use Theorems 3.1 and 3.2, we need methods for establishing E-separation of T. Methods for showing termination of (R, E) and validity in T are well understood, comparatively speaking; the main problem in applying the above theorems is in showing E-separation. In order to make progress, it is useful to consider further the structure of $\mathrm{rts}(R, E)$ and $R^E(G)$. Suppose S is a finite set of terms. Then $\mathrm{fb}(S)$ is defined to be the set of all terms t such that no subterm of t is an instance of a term in S. We think of terms in S as "forbidden terms," hence the name $\mathrm{fb}(S)$. Frequently we can show that $\mathrm{rts}(R, E) \subset \mathrm{fb}(S)$ for some S, and this will help in deciding E-separation of T. For, if $\mathrm{rts}(R, E) \subset \mathrm{fb}(S)$ and R and E are valid in T, then E-separation of T is equivalent to a statement about terms in fb(S). These terms may have some structure which makes E-separation of T easier to decide. For example, we may choose S so that fb(S) consists entirely of constructor terms, which may be known to be distinct in T. Many useful sets of terms can be described as fb(S) for various S. For example, if S contains the term $f(x_1,..., x_n)$ then we know that no term in fb(S) has any occurrences of the operator f. Also, if $+$ and $*$ are associative and commutative binary operators and F is $\{+, *, 0, 1\}$ and S contains the terms $x * (y+z)$, $x * 0$, $x * 1$, and $x+0$, then fb(S) contains terms which are sums of products, that is, polynomials with non-negative integer coefficients, in standard form, except that $3 * x$ may be represented as $x+x+x$, and x^2 as $x * x$, etc. There are many mathematical results known about standard forms for polynomials, and we will show later how these may be used to demonstrate E-separation. The following results help to determine when rts$(R, E) \subset$ fb(S) and when $R^E(G) \subset$ fb(S). Note that these results are more general than those of Thiel (1984), since we do not enforce his restrictions on the structure of R, and we are concerned with more general notions of sufficient completeness than he is. His technique is concerned with showing that all R-irreducible ground terms contain only constructors. However, his result generalizes to equational theories more easily than ours, and is much more efficient.

THEOREM 3.4. *Given R, E, and finite set S of terms, it is partially decidable whether* rts$(R, E) \subset$ fb(S).

Proof. We have that rts$(R, E) \subset$ fb(S) iff every term s in S is (R, E) reducible. However, (R, E) reducibility is partially decidable, by an above remark.

COROLLARY. *Given R, E, and finite set S of terms, if (R, E) reducibility is decidable, then it is decidable whether* rts$(R, E) \subset$ fb(S).

Note that (R, E) reducibility is decidable if the E-equivalence classes are finite. This is sometimes decidable even when no finite E-unification algorithm exists; for example, reducibility for an associative but not commutative operator is decidable, even though no finite unification algorithm exists. We now prove a corresponding result for ground terms, in the case in which E is empty; then we suggest how this may be extended to non-empty E.

THEOREM 3.5. *Given R and finite set S of terms, it is decidable whether* $R(G) \subset$ fb(S).

Proof. We show that if there is a ground term t which is R-irreducible and which is an instance of a term in S, then there is such a ground term t'

of depth bounded by D, where D can be computed. The proof is similar to the proof of the "uvwxy" theorem for context free languages (Hopcroft and Ullman, 1979). If such a term t' does not exist, then $R(G) \subset \text{fb}(S)$. Therefore we can partially decide if $R(G) \subset \text{fb}(S)$ by looking through all ground terms of depth bounded by D and verifying that all of them are reducible. Note that t is R-irreducible if no subterm of t is an instance of the left-hand side of any rule of R.

DEFINITION. Given a term t in $T(F, X)$, the *depth* of t, written depth(t), is defined recursively as follows: If t is a constant or a variable then depth(t) is 1. Also, the depth of $f(t_1,..., t_n)$ is $1 + \max\{\text{depth}(t_1),..., \text{depth}(t_n)\}$.

We write equivalence between terms as $\equiv$. That is, $u \equiv v$ iff u and v are the same term. We consider a term as a mapping from nodes to operators, where nodes are sequences of integers. Let op(t, α) be the operator of t at node α. This is defined recursively by $\text{op}(f(t_1,..., t_n), \alpha) = f$ if α is the empty sequence. Also, $\text{op}(f(t_1,..., t_n), i\alpha) = \text{op}(t_i, \alpha)$. The *domain* of a term t is the set of α such that op(t, α) is defined. We write $t[\alpha]$ for the subterm of t at position α. This is defined recursively by $t[\alpha] = t$ if α is the empty sequence. Also, $f(t_1,..., t_n)[i\alpha] = t_i[\alpha]$. We write $\alpha < \beta$ if α is a proper prefix of β, and $\alpha \leqslant \beta$ if α is a prefix of β, possibly equal to β. For a node α of a term t, we write $|\alpha|$ for the length of α as a sequence of integers. The term $t(\alpha \leftarrow u)$ is defined to be t with the α subterm replaced by u. Formally, $t(\alpha \leftarrow u)$ is defined by the equations $t(\alpha \leftarrow u)[\alpha] = u$ and if β is not a prefix of α then $t(\alpha \leftarrow u)[\beta] = t[\beta]$.

Suppose d is the maximum depth of any left-hand side of a rule in R. Given a term t, let $D(t)$ be $\{(\alpha, \beta): t[\alpha] \equiv t[\beta], |\alpha| \leqslant d, |\beta| \leqslant d\}$. If u is a term, let top(u, n) be defined recursively as follows: $\text{top}(f(\ ...\ ), 1) = f$. If $n > 1$, then $\text{top}(f(t_1,..., t_k), n) = f(u_1,.., u_k)$, where u_i is $\text{top}(t_i, n-1)$. Note that in top(u, n), some operators may have fewer than their usual number of arguments. Thus $\text{top}(f(g(x, y), c), 2)$ is $f(g, c)$. Let top(u) be top(u, d).

DEFINITION. A term t is an *R-redex* if there exist a substitution θ and a rule $r \rightarrow s$ in R such that $r\theta \equiv t$. We often write just redex instead of R-redex when R is understood.

PROPOSITION 3.6. *Suppose t and u are terms and $top(t) = top(u)$ and $D(t) \subset D(u)$. Then if t is a redex, so is u.*

Note that $\{\text{top}(t): t$ is a ground term in $T(F, X)\}$, is finite. Also, $D(T(F, X))$, that is, $\{D(t): t \in T(F, X)\}$, is finite.

DEFINITION. Let $S_\alpha^t(\beta, \pi)$ be $\{\gamma: \text{top}(t[\beta]) = \text{top}(t[\gamma]),\ \beta < \gamma,\ \pi \in$

$D(t(\beta \leftarrow t[\gamma])[\alpha]) - D(t[\alpha])\}$. Intuively, if $t[\gamma]$ replaces $t[\beta]$ then there is a new element π in $D(t[\alpha])$.

PROPOSITION 3.7. *If $\alpha \leqslant \beta$ and $top(t[\beta]) = top(t[\gamma])$ and $t(\beta \leftarrow t[\gamma])[\alpha]$ is a redex and $t[\alpha]$ is not then there exists π such that $\gamma \in S_\alpha^t(\beta, \pi)$.*

Proof. Such a replacement must make two subterms of t identical and thus permit a reduction that was not possible before. This can only happen if some left-hand side of a rule in R has repeated variables. For example, $f(g(a), g(g(a)))$ is not reducible by the rule $f(x, x) \rightarrow x$, but $f(g(a), g(a))$ is. In general, such a $t[\gamma]$ must make two specific subterms of $t(\beta \leftarrow t[\gamma])[\alpha]$ identical, since the "local structure" of $t(\beta \leftarrow t[\gamma])[\alpha]$ is not changed.

DEFINITION. Let π be a pair (π_1, π_2) of nodes and let β be a node. We write $\pi \uparrow \beta$ if $\pi_1 \leqslant \beta$ or $\pi_2 \leqslant \beta$. We write $\pi \downarrow \beta$ if $\beta < \pi_1$ or $\beta < \pi_2$.

PROPOSITION 3.8. *If $\gamma \in S_\alpha^t(\beta, \pi)$ then either $\pi \uparrow \beta$ or $\pi \downarrow \beta$.*

PROPOSITION 3.9. *If $\pi \uparrow \beta$ and γ_1 and γ_2 are in $S_\alpha^t(\beta, \pi)$, then $t[\gamma_1] \equiv t[\gamma_2]$. Thus either $\gamma_1 = \gamma_2$ or γ_1 and γ_2 are incomparable.*

PROPOSITION 3.10. *The set $\{\pi\colon (\exists \alpha \leqslant \beta)(\exists \gamma > \beta)\ \gamma \in S_\alpha^t(\beta, \pi),\ \pi \downarrow \beta\}$ is finite, and bounded in size independent of β, over all terms t. The size bound depends on R.*

DEFINITION. Suppose θ and β are nodes of a term t, and $\beta \leqslant \theta$. Then θ-β is the node θ in the term $t[\beta]$. That is, if we consider nodes as sequences of integers, specifying a path from the root of the term, then θ-β is the suffix ϕ of θ such that $\beta\phi = \theta$. Thus if t is $f(g(d, f(c)))$ and θ is (1 2 1) and β is (1) then $t[\theta]$ is c and $t[\beta]$ is $g(d, f(c))$ and θ-β is (2 1).

DEFINITION. Suppose z is the top of some term. Then a *z-constraint* on a term u with top$(u) = z$ is a predicate $P_{\alpha,\beta}(u)$ of the form $u[\alpha] \equiv u[\beta]$ or a predicate $Q_{\alpha,v}(u)$ of the form $u[\alpha] \equiv v$. Here α and β are nodes of z and v is a term.

Note that z is finite and so the set of such α and β is finite. We now relate constraints to $S_\alpha^t(\beta, \pi)$. Suppose $\gamma \in S_\alpha^t(\beta, \pi)$ and $\pi \downarrow \beta$. There are three cases, one of which must apply: (a) If $\pi_1 > \beta$ and $\pi_2 > \beta$, $t[\gamma][\pi_1 - \beta] \equiv t[\gamma][\pi_2 - \beta]$. (b) If $\pi_1 > \beta$ and not $\pi_2 > \beta$ then $t[\gamma][\pi_1 - \beta] \equiv t[\pi_2]$. (c) If $\pi_2 > \beta$ and not $\pi_1 > \beta$ then $t[\gamma][\pi_2 - \beta] \equiv t[\pi_1]$. This covers all the possibilities if $\pi \downarrow \beta$. Note that in the first case, two distinct occurrences of subterms of $t[\gamma]$ are equal. In the second case, the $\pi_1 - \beta$ subterm of $t[\gamma]$ is equal to some nearby subterm of t. In the third case, the $\pi_2 - \beta$ subterm of $t[\gamma]$ is equal to some nearby subterm of t. In all cases, there is a restriction (constraint) on $t[\gamma]$ that does not apply to $t[\beta]$, since $\gamma \in S_\alpha^t(\beta, \pi)$.

DEFINITION. If C is a set of z-constraints for some z, then the equivalence relation $E(C)$ is defined so that for nodes α and β of z, α and β are equivalent iff for all terms u satisfying C, $u[\alpha] \equiv u[\beta]$. Also, $F(C)$ is defined to be the set of nodes α of z that are fixed by C. That is, $\alpha \in F(C)$ if for all terms u and v satisfying C, $u[\alpha] \equiv v[\alpha]$. Finally, $\#(C)$ is defined to be the number of equivalence classes of $E(C)$ that do not intersect $F(C)$.

DEFINITION. *A z-chain* in t is a sequence $\gamma_1, \gamma_2, ..., \gamma_k$ of nodes of t and *a* sequence $C_1, C_2, ..., C_{k-1}$ of z-constraints such that $\gamma_1 < \gamma_2 < \cdots < \gamma_k$ and such that $\text{top}(t[\gamma_i]) = z$ for all i, and such that $C_i(\text{top}(t[\gamma_j]))$ iff $j > i$. Thus C_1 is true for $t[\gamma_2]$, $t[\gamma_3]$, etc., and C_2 is true for for $t[\gamma_3]$, $t[\gamma_4]$, etc.

THEOREM 3.11. *If z has n nodes then the length of a z-chain is bounded by $n+1$.*

Proof. Note that each C_i is a new constraint that is true of $\text{top}(t[\gamma_{i+1}])$ but not of $\text{top}(t[\gamma_i])$. Each constraint either identifies two subterms that were not already identified, or else fixes a subterm that was not fixed before. In either case, $\#\{C_1, C_2, ..., C_i\}$ decreases by one each time. Since $\# C$ is equal to the number of nodes in z if C is the empty set, and $\# C$ can never be negative, it follows that the length of the z-chain is bounded, that is, $k \leqslant n+1$, where n is the number of nodes in z.

Consider a leaf node θ of t. Then $t[\theta]$ is constant or a variable. Suppose t is R-irreducible, but for any nodes α and β of t with $\alpha < \beta$, $t(\alpha \leftarrow t[\beta])$ is R-reducible. Under this assumption, we construct a tree $T(\theta)$ from nodes β such that $\beta \leqslant \theta$. Also, we add a new root node N to the tree. Edges are added in a manner to be described. We will compute a bound on the size of this tree, which will give a bound on the size of a minimal R-irreducible term t.

The tree consists of the node N, plus a z-subtree T_z for each top z appearing in t. The roots of these z-subtrees are son nodes of N. The z-subtrees are defined inductively. In general, for any subset M of the nodes γ such that $\gamma < \theta$, and such that $\text{top}(t[\gamma]) = z$, we define a z-tree $\text{Tree}(M)$ as follows:

(1) The root of $\text{Tree}(M)$ is the minimal node β in M.

(2) M-$\{\beta\}$ is partitioned into sets $M_1, ..., M_m$ in a manner to be described. The subtrees of $\text{Tree}(M)$ are $\text{Tree}(M_1), ..., \text{Tree}(M_m)$. That is, the roots of these subtrees are son nodes of $\text{Tree}(M)$.

Finally, T_z is $\text{Tree}(M)$, where M is the set of $\gamma \leqslant \theta$ such that $\text{top}(t[\gamma]) = z$.

The partitioning of M into $M_1,..., M_m$ is done as follows: Since t is a minimal R-irreducible term, for all γ in M-$\{\beta\}$ there exists π and $\alpha < \beta$ such that $\gamma \in S^t_\alpha(\beta, \pi)$. If $\pi \uparrow \beta$ then there is at most one γ, and so we let $\{\gamma\}$ be one of the M_i. Such γ we call *super-constrained*, since they are subterms of terms that are constrained to be equal. These will become leaf sons of β. The remaining nodes all must satisfy some constraint that top($t[\beta]$) does not satisfy. Let $\{C_1,..., C_k\}$ be some such set of constraints. We let M_i be some set of nodes such that for all γ in M_i, γ satisfies C_i. Note that if some node γ satisfies more than one constraint, we must choose which of the possible sets M_i to place it in.

We now estimate the size (number of nodes) in this tree, assuming t is minimal R-irreducible. An upper bound on the size of the tree gives an upper bound on the depth of t. Note that any path in $T(\theta)$ corresponds to a z-chain in t, for some z, with possibly the root node N in front and a super-constrained leaf node at the end. Therefore such a path can contain at most $n+1$ nodes other than the root node N and a super-constrained leaf node, where n is the number of nodes in z.

LEMMA. *Suppose π_1 is a node in t. Then the number of nodes π_2 of t such that there exists node α in t and sequences α_1 and α_2 such that $\pi_i = \alpha\alpha_i$ and $0 < |\alpha_i| \leqslant d-1$, is bounded by $(d-1)n'$, where n' is the maximum number of nodes in any top of a subterm.*

Proof. The node α is completely determined by the length of α_1. There are $d-1$ choices for the length of α_1. For each such choice, we may choose π_2 to be any sequence of length in the range $\{1, 2,..., d-1\}$. The number of such sequences is bounded by n', since a top has depth at most d and thus its nodes have lenght bounded by $d-1$.

Informally, we say that π_1 is near π_2 if the conditions of the lemma are true. Now, given a node β of $T(\theta)$ other than N, β can have at most dn'^2 non-leaf sons γ. This is because each son corresponds to a new constraint. Each constraint specifies that one of the subterms of $t[\gamma]$ is identical to a nearby subterm in t, or that two of these n subterms are identical. By the lemma, there are at most $(d-1)n'$ nearby subterms that can be constrained to be equal to some subterm of $t[\gamma]$, leading to about $(d-1)n'^2$ constraints. Also, there are $n(n-1)/2$ ways that two subterms of $t[\gamma]$ may be specified to be identical. Since $n < n'$, the number of ways that a new constraint may be specified is bounded by $d(n')^2$. (The number of constraints is actually infinite, since there are an infinite number of ways that a term may be fixed.) Also, N can have U sons, where U is the number of distinct tops of terms. Thus the total number on non-super-constrained nodes in $T(\theta)$, including the root N, is bounded by $1 + U(1 + d(n')^2 + d^2(n')^4 + \cdots + d^n(n')^{2n'})$, or, $1 + U(d^{n+1}(n')^{2n'+2} - 1)/(d(n')^2 - 1)$.

We now discuss the leaf nodes. We are only concerned with super-constrained leaf nodes. These are nodes γ corresponding to π such that $\pi \uparrow \gamma$. If β is a node in t, β may have a number of super-constrained leaf sons bounded by $(d-1)n'|\beta|$. To see this, if γ is a super-constrained leaf son of β then there exists $\alpha < \beta$ such that $\gamma \in S_\alpha^t(\beta, \pi)$, where $\pi \uparrow \beta$. We know that $\pi_1 \leqslant \beta$ and π_2 is near α, or vice versa. By the lemma, there are at most dn' nodes near α, and at most $|\beta|$ nodes α with $\alpha \leqslant \beta$. Therefore there are at most $(d-1)n'|\beta|$ such pairs π. However, $|\beta|$ depends on how deep β is in t. Thus $|\beta|$ depends indirectly on the size of $T(\theta)$. It is necessary to bound the number of these super-constrained leaf sons somehow to bound the size of $T(\theta)$.

We bound the size of $T(\theta)$ by induction. The idea is to look at the maximum node γ in $T(\theta)$ which is not super-constrained. Let T' be T with this node and all its (leaf) sons deleted. Suppose $|T|$, the number of nodes in T, is m. Then $|\gamma| \leqslant m$ so γ has at most $dn'm$ super-constrained leaf sons. Also, γ may not have any other sons, by the way γ was chosen. Thus $|T| \leqslant dn'|T'|$. In this way we get a recurrence for $|T|$ in terms of the number of non-super-constrained nodes in T. Each non-super-constrained node corresponds to a multiplication of $|T|$ by dn'. Also, we have already bounded the number of non-super-constrained leaf nodes in T. Therefore we may compute a bound on $|T|$, that is, on the depth of t if t is minimal R-irreducible. This is sufficient for our decision procedure. It turns out that this bound is of the form $(dn')^{U*(dn')^{b*n'}}$ for some constant b, where U is the number of tops of terms and n' is the maximum size of a top of a term. Since U is exponential in n', this bound is double exponential in n'. Also, n' is itself exponential in d, the maximum depth of the left-hand side of a rule in R. This gives a bound in depth; the number of terms of depth bounded by k is double exponential in k. So the decision procedure we obtain is quintuple exponential in d. This can be reduced to quadruple exponential in d by representing terms as directed acyclic graphs as indicated below.

To finish the proof of the theorem, we need to show that if there is an R-irreducible term that is an instance of some term u in S, then there is an R-irreducible term of bounded depth that is also an instance of u. To do this, consider a node θ of t such that $t[\theta]$ is a constant or a variable. Consider the path in t to θ starting at the root of t. When constructing the tree $T(\theta)$, we do not include nodes near the top of this path, that is, nodes that are in the domain of u. These nodes are fixed by the requirement that t be an instance of u. Also, these extra nodes influence the number of ways that a term may be super-constrained. In the above proof, if $|\gamma| \leqslant m$, γ has at most $dn'm$ super-constrained leaf sons. Now, some prefix of γ represents nodes in the domain of u. Thus $|\gamma|$-δ represents the length of that portion of γ that is in $T(\theta)$, for δ equal to the depth of u. Clearly γ has at most $dn'((|\gamma| - \delta) + \delta)$ super-constrained leaf sons. Therefore the recurrence

$|T| \leqslant dn'|T'|$ is changed to $|T| \leqslant dn'(|T'| + \delta)$. In this way a bound similar to the above one may be computed. This gives a decision procedure and completes the proof of Theorem 3.5.

There are probably much better methods; the above method is very inefficient since D may be very large. However, in many cases, the analysis is much simpler, since we can often find a small finite set I of instances of s such that each element of I is R-reducible, and such that any ground instance of s is also a ground instance of some element of I. For left linear term-rewriting systems (in which no rule has repeated variables on the left-hand side), the method is much simpler since it is not necessary to worry about equality of subterms of t. In general, it is the number of non-left linear rules that influences the size of D. For general R, the method can be made more efficient by representing t as a directed acyclic graph (Paterson and Wegman, 1978) in which each repeated subterm is represented only once. Then the above bound D can be applied to the total size of t, not its depth. The reason is that a subterm of t can be replaced by *any* subterm of t having the same top, subject to restrictions as in the above proof. Thus we can construct the tree T from *all* the subterms of t, not just those on some path. In this way, the bound on T gives a size on the number of distinct subterms of t, not just the depth of t. Furthermore, if t is represented as a directed acyclic graph, then the size of t will be equal to the number of distinct subterms of t. In this way the decision procedure of Theorem 3.5 may be improved by one exponential.

THEOREM 3.12. *Suppose E consists of the associative axiom or the associative and commutative axioms for one or more operators of R. Suppose that if $r \rightarrow s$ is a rule of R and x is a variable argument to an associative operator in r, then x appears only once in r. Then it is decidable whether $R^E(G) \subset \mathrm{fb}(S)$.*

Proof. Similar to that of Theorem 3.5. The idea is to consider the "flattened" form of t, in which, for example, $f(x, f(y, z))$ is replaced by $f(x, y, z)$ if f is associative. The problem is that there may be infinitely many flattened terms of a bounded depth, since an associative operator may have arbitrarily many arguments in a flattened term. However, we note that if t is (R, E)-irreducible, then so is u obtained from t by deleting some arguments of all associative operators subject to the following conditions: (a) Arguments near the top of t cannot be deleted if this would destroy the property of being an instance of term s in S. (b) Arguments may not be deleted if this makes some subterm of t have a top equal to the top of some instance of the left-hand side of some rule in R. (c) Arguments may not be deleted if this makes two subterms of t identical that were not identical before. We note that (a) only constrains t to have a bounded number of arguments of associative operators near the top of t. Also, (b) may be satisfied if we never reduce the number of arguments of an associative operator to the number appearing in the left-hand side of some rule of R. This is also a finite quantity. Finally, (c) may be satisfied as follows: Suppose that there are n occurences of associative operators in (flattened) t. Let m be the maximum of n and the number of arguments of associative operators in s and left-hand sides of R. Now, all associative operators with more that $2m$ arguments may have arguments deleted so that they have between m and $2m$ arguments, without making any two such occurrences identical. In this way, we may obtain an irreducible term t' in which the number of arguments to associative operators is bounded in terms of the depth of t'. Thus we obtain a finite set of ground terms as before. The commutative axiom by itself can be handled by sorting arguments to commutative operators in t, and by extending R in all possible ways by interchanging arguments of commutative operators in rules of R. As before, it will be necessary to deal with structure near the top of t in a different way.

We extend the previous results to conditional term rewriting systems.

THEOREM 3.13. *Given conditional term rewriting system R, partially decidable theory T, set E of equations, and finite set S of terms, it is partially decidable whether* $\mathrm{rts}(R, E) \subset \mathrm{fb}(S)$.

Proof. Similar to the proof of Theorem 3.4, making use of the fact that reducibility is partially decidable for conditional term rewriting systems.

THEOREM 3.14. *Given conditional term rewriting system R, set E of equations, and finite set S of terms, if validity in T is decidable and if the E equivalence classes are finite then it is decidable whether* $\mathrm{rts}(R, E) \subset \mathrm{fb}(S)$.

Proof. Under the conditions of the theorem, (R, E)-reducibility is decidable.

Unfortunately, we cannot extend Theorem 3.5 to conditional systems, since the height reduction argument does not apply; two ground terms may have the same top but may cause conditions to evaluate differently. Instead, we give a simple sufficient condition for $R(G) \subset \mathrm{fb}(S)$ for unconditional and then for conditional systems.

DEFINITION. A set C of terms is a *covering set* for a term s and a set F of operators if all ground terms t over F which are instances of s are ground instances of some term in C.

For example, in the theory of list structure, if F is {CONS, NIL} then {CONS(CONS(X, Y), Z), CONS(NIL, Z)} is a covering set for the term CONS(X, Y).

DEFINITION. A set C of terms is an *S-covering set* for a term s and a set F of operators if S is a set of terms and all ground terms t over F which are instances of s are either ground instances of some term in C or have a proper subterm which is an instance of some term in S.

For example, if S contains the term APPEND(X, Y) then the following is an S-covering set for APPEND(X, Y) over {APPEND, CONS, NIL}: {APPEND(CONS(X, Y), Z), APPEND(NIL, Z)}. The term APPEND(APPEND(X, Y), Z) need to be included, since APPEND occurs in a proper subterm.

PROPOSITION 3.15. *Suppose C is a covering set for s over F and S is a set of terms over F. Let C' be {u in C: u has a proper subterm which is an instance of some term in S}. Then C-C' is an S-covering set for s over F.*

Proof. Any term t that is an instance of some u in C', has a proper subterm that is an instance of some term in S. Also, any ground term t that is an instance of s is an instance of some term in C, since C is a covering set for s over F. Thus any ground instance of s that does not have a proper subterm that is an instance of some term in S, is an instance of some term in $C - C'$.

For example, let C be the following covering set for APPEND(X, Y) over {APPEND, CONS, NIL}: {APPEND(APPEND(X, Y), Z), APPEND(CONS(X, Y), Z), APPEND(NIL, Z)}. If S contains the term APPEND(X, Y) then C' is the set {APPEND(APPEND(X, Y), Z)} so $C - C'$ is an S-covering set for APPEND(X, Y) over {APPEND, CONS, NIL}. In this way the previous example may be obtained.

THEOREM 3.16. *Given C, S, s, and F, it is decidable whether C is an S-covering set for s over F.*

Proof. For simplicity, assume that none of the terms in $C \cup S$ are variables. For each operator f, let f' be a new operator of the same arity. Let s' be s with its top-level operator g replaced by g'. Let C' be C with each term u in C having its top level operator replaced in the same way. Then C is an S-covering set for s over F iff every ground instance of s' has a subterm that is an instance of some term in $S \cup C'$. This property may be decided as in Theorem 3.5.

COROLLARY. *Given C, s, and F, it is decidable whether C is an s-covering set over F.*

THEOREM 3.17. *Given unconditional term rewriting system (R, E) over F and finite set S of terms, $R^E(G) \subset \text{fb}(S)$ if for every s in S there is an S-covering set C for s and F such that every term t in C is (R, E)-reducible.*

Proof. Suppose u is in $R^E(G)$. Suppose u has a subterm v that is an instance of some s in S. Let v be some minimal such subterm of u, so that v does not itself have proper subterms that are instances of any term in S. Consider the S-covering set C for s and F. By definition of an S-covering set, v is an instance of some term in C. By the theorem, every term in C is (R, E)-reducible. Therefore u is (R, E)-reducible, contracting the statement that u is in $R^E(G)$. Therefore $R^E(G) \subset \text{fb}(S)$.

DEFINITION. Suppose (R, E) is a conditional term rewriting system and t is a ground term. Then t is *disjunctively reducible in* (R, E) *and theory* T if there is a set $\{r_i \rightarrow s_i \text{ if } D_i\}$ of *ground instances* of rules in R, such that

(a) every r_i is a subterm of a term E-equivalent to t and

(b) $\vee\ _iD_i$ is valid in T.

THEOREM 3.18. *Suppose* (R, E) *is a conditional term rewriting system over* F *and* S *is a finite set of terms. Suppose all conditions* C *of the rules satisfy the property that if* C_1 *is a ground instance of* C *then either* C_1 *or* $\neg C_1$ *is valid in* T. *Suppose that for every* s *in* S *there is an* S*-covering set* C *of* s *over* F *such that every term* t *in* C *is disjunctively reducible in* (R, E) *and* T. *Then* $R^E(G) \subset \text{fb}(S)$.

Proof. The conditions guarantee that every ground instance of s is reducible. The condition on C_1 guarantees that if a disjunction of ground conditions is true, then one of the disjuncts must be true, so a reduction can occur.

To show that a term is disjunctively reducible, we need to show that $\vee\ _iD_i$ is valid in T. This requires some kind of a theorem prover. We envisage that an interactive theorem prover would be used to prove this and other theorems of T. This has the disadvantage of requiring more user interaction but has the advantage of possibly increasing the power of the system over Knuth–Bendix, which does not permit much user interaction.

3.4 *Completing a Non-confluent System*

Using the above techniques, an (R, E) irreducible term not in fb(S) can often be exhibited if it exists. Then a user would typically be able to give a term it should reduce to. In this way a non-confluent system could be completed, with user interaction. However, for theoretical reasons, we also give completely automated methods for completing non-confluent term rewriting systems.

If $>$ is a partial ordering on terms, then we say (R, E) is $>$ *decreasing* if for all terms t and u, if $t \Rightarrow u$ then $t > u$. Note that if $>$ is well founded and

(R, E) is $>$ decreasing, then (R, E) is terminating. The technique of simplification orderings (Dershowitz, 1982) may be used to show termination when E is empty; an extension of this technique to associative–commutative operators is given in (Bachmair and Plaisted, 1985; Dershowitz *et al.*, 1983). We say an ordering $>$ is *computable* if, given terms t and u, it is decidable whether $t > u$. We say an ordering $>$ is *E-respecting* if $t > u$ and $t =_E w$ and $u =_E v$ implies $w > v$.

THEOREM 3.19. *Given finitely axiomatized first-order theory T, E valid in T, finite set S of terms, and computable ordering $>$ which is E-respecting, it is partially decidable whether there exists R such that R is valid in T, (R, E) is $>$ decreasing, and* $\text{rts}(R, E) \subset \text{fb}(S)$. *Moreover, such an R can be constructed, if it exists.*

Proof. For each term s in S, we enumerate terms u until we find one such that $s > u$ and $s = u$ is valid in T. Then the rule $s \to u$ is added to R.

THEOREM 3.20. *Given first-order finitely axiomatized theory T, finite set S of terms, and computable ordering $>$, it is partially decidable whether there exists R such that R is valid in T and R is $>$ decreasing and* $R(G) \subset \text{fb}(S)$. *Moreover, such an R can be constructed, if it exists.*

Proof. The conditions on R are all decidable or partially decidable.

Theorem 3.20 can be extended to systems (R, E) in which E consists of associative and/or commutative axioms for some of the operators.

These methods are not necessarily efficient; however, the efficiency can be improved by generating u such that $s > u$ and $s = u$ in a more careful way. For example, if T is equational, we can use Lankford's (1975) method, starting with the equation $s' \neq x$, where s' is s with all variables replaced by new constants. This will generate contradictions by finding u such that $s' = u$ is a logical consequence of T. Furthermore, such u will be generated in a way that respects $>$, that is, not necessarily all such u will be generated, but only those that are simplified with respect to $>$.

3.5. *Proper Definitions*

We now give syntactic methods of showing that (R, E) is E_1-separating on $\text{fb}(S)$ for various theories T. Later we shall give other methods based on linear algebra, polynomials, and related results.

PROPOSITION 3.21. *The empty theory T is separating on any set of terms.*

PROPOSITION 3.22. *The first-order theory of E is E-separating on any set of terms.*

PROPOSITION 3.23. *If the initial theory of A is E-separating on a set U of terms and A_1 is a set of assertions that are true in the initial model of A then the initial theory of $A \cup A_1$* is E-separating *on U.*

PROPOSITION 3.24. *If T is E-separating on set U of ground terms and T has an initial model, then the initial theory of T is ϕ-separating on set U of terms.*

This proposition is not true for non-ground terms. There are often equations true in the initial theory of T that are not true in T. Therefore initial models are most useful for showing ground confluence.

PROPOSITION 3.25. *If (R, E) is a possibly conditional term rewriting system and (R, E) is terminating and E_1 confluent, then the first-order theory of $R \cup E$ is E_1-separating on* $\mathrm{rts}(R, E)$.

Thus, one may be able to show that a set of rules is confluent using the Knuth–Bendix method, and then extend the theory in various other ways that maintain E_1-separation.

DEFINITION. Suppose T is a theory and T_1 is obtained from T by adding new formulae D that define a new operator or predicate, and also including in T_1 all the logical consequences of $T \cup D$. Then we say the definition D is a *proper definition* for T if every model M of T may be extended to a model M_1 of T_1 by interpreting the new operator or predicate appropriately.

THEOREM 3.26. (*Properties of proper definitions*). *Suppose T is E-separating on a set U of terms. Suppose T_1 is obtained from T by adding a proper definition. Then T_1 is also E-separating on U.*

THEOREM 3.27. *Suppose A is a set of axioms whose first-order theory T is E-separating on a set U of terms. Suppose A has an initial model. Suppose D is a definition which is proper for the initial theory of A. Then the first-order theory of $A \cup D$ is E-separating of U.*

Proof. There is a model of $A \cup D$ which extends the initial model of A. The initial model of A is E-separating on U. Therefore the extension of this model is E-separating on U. Since the initial model is a model of A, the extension is a model of $A \cup D$. Therefore the first-order theory of $A \cup D$ is E-separating on U.

We note that many specifications are essentially built up by a series of definitions which can be shown syntactically to be proper. Thus E-separation should be easy to establish in many cases. We give some examples of syntactic criteria for proper definitions. It should be easy to extend these methods to higher order logic. Theorem 3.27 may be used to show E-separation for improper definitions when T has an initial model. Note that if U is expressed as fb(S) for some S, then U also depends on the set of operators in T. If T is extended, new terms may have to be added to S.

Suppose a new operator f is defined by a series of equations of the form

$$\begin{array}{ll} r_1 = s_1 & \text{if } C_1 \\ r_2 = s_2 & \text{if } C_2 \\ \vdots & \\ r_k = s_k & \text{if } C_k \end{array}$$

where the r_1 are terms having f at the top level. For example, append can easily be defined in this way with C_i being TRUE by defining append(cons$(x, y), z)$ and append(NIL, z) separately. Also, max(x, y) can be so defined, using the conditions C_i. Let F_1 be the set of subterms of r_i, s_i, and C_i with f at the top level, excluding the subterm r_i itself. We say such a definition is *syntactically well founded* if t is smaller than r_i in some well-founded ordering, for all terms t in F_1. The well-founded ordering must satisfy the condition $u < v \supset u\theta < v\theta$. Such a definition is *syntactically non-overlapping* if none of the r_i have common instances. Let args(t) for term $t = f(t_1, ..., t_n)$ be defined to be the tuple $\langle t_1, ..., t_n \rangle$. Also, for a gound term t, let t^M be the interpretation of t in model M. Such a definition is *semantically well founded* for a theory T if in any model M of T, there is a well-founded ordering $<$ on tuples of elements of the domain of M such that for all t in F_i, for all subsitutions θ replacing all relevant variables by ground terms, if $C_i\theta$ is valid in T then $(\text{args}(t)\theta)^M < (\text{args}(r_i)\theta)^M$. Such a definition is *semantically non-overlapping* for T if for all r_i and r_j with i, j distinct, for all θ_i and θ_j replacing relevant variables by ground terms, and for all models M of T, if $C_i\theta_i$ and $C_j\theta_j$ are true in M $(\text{args}(r_i)\theta_i)^M$ and$(\text{args}(r_j)\theta_j)^M$ are distinct. Also, we require that there be no "self-overlap," that is, for all r_i, for all θ_i and θ_i' replacing relevant variables by ground terms, and for all models M of T, if $C_i\theta_i$ and $C_i\theta_i'$ are true in M and $(s_i\theta_i)^M$ and $(s_i\theta_i')^M$ are distinct then $((\text{args}(r_i)\theta_i)^M$ and $(\text{args}(r_i)\theta_i')^M$ are distinct. Note that if a definition is semantically non-overlapping and the conditions are all TRUE then the definition is also syntactically non-overlapping.

THEOREM 3.28. *Suppose a definition of f in a theory T is semantically* *non-overlapping and semantically well founded. Then this definition is proper for T.*

Proof. By induction on the well-founded ordering on tuples of elements of the domain of M, we show that f may be given an interpretation consistent with M.

PROPOSITION 3.29. *Suppose a definition of f in a theory T is syntactically well founded and has no semantic overlap. Then the definition is also semantically well founded.*

This proposition is useful when syntactic well-foundedness is easier to demonstrate than semantic well-foundedness. Note that a definition may be incomplete, and still be proper. Thus it might be possible to consider incompletely specified theories and use them to show confluence. Also, note that although the definition needs to be non-overlapping to guarantee that it is proper, the term-rewriting system may overlap.

3.6. *An Example*

We now show how some of the above techniques may be used to demonstrate E-separation. Suppose we are given the following specification for list structures and integers:

```
SORTS
  LIST INT BOOLEAN
OPS
  NIL: → LIST
  CONS: LIST LIST → LIST
  0: → INT
  S: INT → INT
  P: INT → INT
REDUCTIONS
  S(P(X)) = X
  P(S(X)) = X
```

We can take the theory simply to be the two equations, together with the usual theory of Boolean connectives and TRUE, FALSE. This may be shown confluent using the traditional Knuth–Bendix method. Alternatively, we may show confluence since $\text{rts}(R) \subset \text{fb}(\text{“}S(P(X))\text{”}, \text{“}P(S(X))\text{”})$. Thus $\text{rts}(R)$ consisis of list structures together with expressions of the form $S^n(0)$ and $P^n(0)$ and we know that these are all distinct in the theory of arithmetic. Next, we can take the *initial* theory of these two equations. This preserves ϕ-separation on the set $G \cap \text{fb}(\text{“}S(P(X))\text{”}, \text{“}P(S(X))\text{”})$, by

Proposition 3.24. Next, we may add some proper definitions, such as the following:

$\text{APPEND}(\text{NIL}, Z) = Z$
$\text{APPEND}(\text{CONS}(X, Y), Z) = \text{CONS}(X, \text{APPEND}(Y, Z))$
$\text{LENGTH}(\text{NIL}) = 0$
$\text{LENGTH}(\text{CONS}(X, Y)) = S(\text{LENGTH}(Y))$
$\text{PLUS}(0, X) = X$
$\text{PLUS}(S(X), Y) = S(\text{PLUS}(X, Y))$
$\text{PLUS}(P(X), Y) = P(\text{PLUS}(X, Y))$
$\text{GT}(S(X), X) = \text{TRUE}$
$\text{GT}(S(X), Y) = \text{TRUE IF GT}(X, Y)$
$\text{GT}(X, X) = \text{FALSE}$
$\text{GT}(P(X), Y) = \text{FALSE IF NOT GT}(X, Y)$
$\text{MAX}(X, Y) = X \text{ IF GT}(X, Y)$
$\text{MAX}(X, Y) = Y \text{ IF NOT GT}(X, Y)$

These definitions are proper since we are considering the initial theory. For example, the first two rules are non-overlapping because in the initial model, CONS(X, Y) and NIL are distinct for all X, Y. Since these definitions are proper, the resulting theory is still ϕ-separating on the same set of terms. However, because new operators have been added, this set of terms must now be expressed as $G \cap \text{fb}($"$P(S(X))$," "$S(P(X))$," "PLUS(X, Y)," "MAX(X, Y)," "GT(X, Y)," "LENGTH(X)," "APPEND(X, Y)"). Let us call this set U. Finally, we can give the following ground confluent term rewriting system

$S(P(X)) = X$
$P(S(X)) = X$
$\text{APPEND}(\text{NIL}, Z) = Z$
$\text{APPEND}(\text{CONS}(X, Y), Z) = \text{CONS}(X, \text{APPEND}(Y, Z))$
$\text{APPEND}(\text{APPEND}(X, Y), Z) = \text{APPEND}(X, \text{APPEND}(Y, Z))$
$\text{LENGTH}(\text{NIL}) = 0$
$\text{LENGTH}(\text{CONS}(X, Y)) = S(Z) \text{ IF LENGTH}(Y) = Z$
$\text{PLUS}(0, X) = X$
$\text{PLUS}(S(X), Y) = S(\text{PLUS}(X, Y)$
$\text{LENGTH}(\text{APPEND}(X, Y)) = \text{PLUS}(\text{LENGTH}(X), \text{LENGTH}(Y))$
$\text{MAX}(X, Y) = X \text{ IF NOT GT}(Y, X)$
$\text{MAX}(X, Y) = Y \text{ IF NOT GT}(X, Y)$

We know this is ground confluent since all equations are valid in the given theory, and the theory is ϕ-separating on U. Also, we can show that $\text{rts}(R) \subset U$. We are assuming GT(X, Y) is decided by some decision procedure. We have not shown this system confluent, however. Note that this term-rewriting system is overlapping. For an example of a ground confluent system in which E and E_1 of the definition of ground confluence differ, consider the above system in which the reduction APPEND(APPEND(X, Y), Z) = APPEND(X, APPEND(Y, Z)) is considered as an equation rather than a reduction. Then this system is still ground confluent, by the same reasoning. Then the system (R, E) with R as follows and E the associativity axiom may be shown ground confluent:

APPEND(NIL, Z) = Z
APPEND(CONS(X, Y), Z) = CONS(X, APPEND(Y, Z))
LENGTH(NIL) = 0
LENGTH(CONS(X, Y)) = $S(Z)$ IF LENGTH(Y) = Z
LENGTH(APPEND(CONS(X, NIL), Z)) = S(LENGTH(Z))
LENGTH(APPEND(Z, CONS(X, NIL))) = S(LENGTH(Z))

In order to do this, it is necessary to prove that the system is terminating relative to the associativity axiom. Note that normal forms of ground terms are unique; this system is ground confluent relative to the empty set of equations even though E is not empty. For another example, if we have a specification of sets of integers, there may not be unique normal forms for sets but there may be for integers. Therefore the confluence properties may differ for different sorts.

4. Showing E-Separation

We give additional methods for showing that theories are E-separating on various sets of ground and non-ground terms. These involve establishing that some first-order formula is in T. Since T is assumed to be a partially decidable theory, such a formula can be proven, possibly with help from human interaction. A number of algebraic results involving properties of polynomials are also used to show E-separation. We use AC as an abbreviation for associative–commutative.

Theorem 4.1. *Suppose S contains all terms of the form $f(x_1,..., x_n)$ for $f \in NC$, the set of non-constructor functions. Let C, the set of "constructor functions," be F-NC. Suppose that the following assertions are valid in T:*

$$f(\bar{x}) \neq g(\bar{y}) \qquad \textit{for distinct } f, g \textit{ in } C$$
$$f(\bar{x}) = f(\bar{u}) \supset x_i = y_i \qquad \textit{for } f \textit{ in } C$$

Then T is ϕ-separating on fb(S). *That is, T is E-separating for E the empty set of equations.*

Proof. All elements of fb(S) are terms containing only constructors. The given assertions imply that distinct such terms are unequal in T.

THEOREM 4.2. *Suppose $R^E(G)$ consists entirely of ground terms of the form $c+c+c+ \ldots +c$, where c is a constant and $+$ is some AC operator, and the term 0, where 0 satisfies $c+0=c$. Suppose the assertions $x+c \neq 0$ and $x+c=y+c \supset x=y$ are valid in T. Then T is E-separating on $R^E(G)$, where E consists of the associative and commutative equations for $+$.*

Proof. Straightforward.

THEOREM 4.3. *Suppose $R^E(G)$ consits entirely of ground terms of the form $c_1+c_2+ \cdots +c_k$, where the c_i are constants and $+$ is some AC operator, and of the form 0, where 0 satisfies $x+0=x$. Let $d_1,\ldots, d_p$ be the p distinct constants. Suppose T also has a multiplicative operator $*$ satisfying $n*x=x+x+ \cdots +x$ (n times) for natural numbers n, and $0*x=0$. Suppose the assertion $\sum_i m_i\, d_i = \sum_i n_i\, d_i \supset (\forall j) m_j = n_j$ is valid in T. Note that this is a first-order assertion in T. Then T is E-separating on $R^E(G)$ for E consisting of the associative and commutative equations for $+$.*

Proof. Straightforward.

THEOREM 4.4. *Let R, E, and T be as above, and suppose that* rts(R, E) *consists of terms which are sums of variables and the constants d_i. Suppose the same assertions as above are valid in T, plus the assertion $x+z=y+z \supset x=y$. Suppose T has a constant 0 satisfying $x+0=x$. Then T is E-separating on* rts(R, E), *where E consists of the associative and commutative equations for $+$.*

Proof. Let A and B be two terms in rts(R, E). We can set all variables in A and B to 0, so their constant parts must be equal. By linear independance of the constants, these constant parts must be identical (up to associativity and commutativity of $+$). By the equation $x+z=y+z \supset x=y$, the variable parts must be equal. Note that it follows from linear independence that d_j and 0 are unequal for all j. By replacing each variable in turn by d_1 and the others by 0, using linear independence, we can verify that A and B have the same number of occurrences of each variable.

It would be interesting to extend the previous result to the case in which rts(R, E) may also contain constructor functions applied to arguments of the same form, that is, sums of variables and constructor terms. Note that one way to demonstrate the "cancellation law" $x+z=y+z \supset x=y$ is to obtain some expression Expr such that Expr$(x+y, y)=x$. If there is an additive inverse, then Expr(w, y) can be $w-y$.

4.1. *Fields*

We now make use of some properties of fields to show E-separation of T when rts(R, E) is a set of polynomials. For example, if rts(R, E) consists of sums of products of variables and the additive identity 1, then terms in rts(R, E) can be regarded as polynomials with natural number coefficients. Thus $x * x + x + x + 1$ can be regarded as the polynomial $x^2 + 2x + 1$. If there is also a constant -1 then sums of products of variables, 1, and -1 can be regarded as polynomials over the integers. Therefore results about polynomials can be used to show E-separation on such sets of terms if $*$ and $+$ are AC operators satisfying certain axioms. These results can sometimes apply even when the theory T is not a theory of polynomials, but instead has such a subtheory. Also, sometimes the existence of additive inverses can be replaced by other assumptions. From now on assume that $+$ and $*$ are AC operators, that 0 is an additive identity, that 1 is a multiplicative identity, that $x * 0 = 0$, and that $x * (y + z) = x * y + x * z$ is valid in T. Also, there is an element -1 such that $1 + (-1) = 0$. Thus we have a commutative ring. For a discussion of the theory of rings and fields see (Birkhoff and MacLane, 1965; Lang, 1970). A commutative ring is called an *integral domain* if there are no x and y distinct from zero such that $x * y = 0$. A *field* is an integral domain together with a division operator x/y defined when $y \neq 0$, satisfying the property $y * (x/y) = x$. The *characteristic* of a field is the number of distinct multiples 1, $1 + 1$, $1 + 1 + 1$,..., of 1. Often "characteristic 0" is used to denote an infinite number of such multiples. The following results (Lang, 1970) are basic:

THEOREM 4.5. *In a field, two polynomials in one variable of degree p, identical on $p + 1$ values, are identical.*

THEOREM 4.6. *In a field, let q be the smallest value such that $x^q = x$ for all x, if such a q exists, else infinity. Then two polynomials in several variables, whose exponents of variables are all less than q, are identical, if they are identical as functions for all values of the variables.*

One can show that a field has characteristic infinity by showing the existence of a partial ordering $<$ satisfying $0 < 1$ and $x < y \supset x + z < y + z$. Then any two polynomials in several variables, which are equal as functions, are identical. Otherwise, we show that $x^q = x$ and that for all n, if $0 < n < q$ then there exists y such that $y^n \neq y$. Let p be the characteristic of the field. If is known that if p is finite then p must be a prime, and if q is also finite then q is a power of p. The fact that the characteristic is p may be established by the first-order assertion that $1 + 1 + 1 + \cdots + 1 = 0$, where 1 is repeated p times, and the assertions that $1 + 1 + \cdots + 1 \neq 0$, where 1 is repeated n times, for $0 < n < p$. The equation $x + x + \cdots + x = 0$ has the effect of reducing all coefficients of the polynomial to elements of the underlying field. Then if the equation $x^q = x$ is added to the term-rewriting system, and the equation $x + x + \cdots + x = 0$ is added, where x is repeated p times, the irreducible forms of polynomials will still be identical if they are equal. Note that the assertions about q and p are first-order assertions, hence can be verified by a first-order proof checker. We therefore get the following results.

THEOREM 4.7. *Suppose* rts(R, E) *consists of a set of polynomials, that is, sums of products of variables and integers* 1 *and* -1, *with* $x * (y+z)$, $x * 0$, $x+0$, $x * 1$, $(-1) * (-1)$, $1 + (-1)$, *and* $x + (-1) * x$ *as forbidden subterms. Suppose* T *contains the field axioms given above for* $+$ *and* $*$, 0, 1, *and* -1, *and that* E *contains only the associative and commutative equations for* $+$ *and* $*$. *Suppose there is a partial ordering* $<$ *such that* $0 < 1$ *and* $x < y$ *implies* $x + z < y + z$ *in* T. *Then* T *is* E*-separating on* rts(R, E).

THEOREM 4.8. *As above, except that instead of the partial ordering we have the first-order assertions given earlier about* q *and* p, *and also the rules* $x^q \rightarrow x$ *and* $x + x + \cdots + x \rightarrow 0$ *(for* x *repeated* p *times) in* R. *Then* T *is* E*-separating on* rts(R, E).

A *boolean ring* is a ring with the axiom $x * x = x$ added; it can be shown (Stone, 1936) that in a Boolean ring $x * y = y * x$ and $x + x = 0$. Note that the subset $\{0, 1\}$ of a Boolean ring is a field of characteristic 2. It follows that two polynomials over a Boolean ring that are equal for all values of the variables, are identical, if no variable has exponent higher than one. We thus obtain the following result:

THEOREM 4.9. *Suppose* rts(R, E) *consists of a set of polynomials in several variables. Suppose* R *contains the equations* $x * x \rightarrow x$ *and* $x + x \rightarrow x$. *Suppose there is a theory* T *such that* R *and* E *are valid in* T, *and* $+$ *and* $*$ *form a Boolean ring in* T. *Then* T *is* E*-separating on* rts(R, E), *where* E *consists of the associative and commutative equations for* $+$ *and* $*$.

There are other ways of showing theories to be E-separating on sets of polynomials, even if an additive inverse is not explicitly given. We define a *difference operator* $D(x, y)$ to satisfy the following axioms:

$$D(x, x) = 0$$

$$D(y, z) = 0 \supset y = z$$

$$D(x * y, x * z) = x * D(y, z)$$

$$D(x + y, u + v) = D(x, u) + D(y, v)$$

$$D(x * y, u * v) = x * D(y, v) + v * D(x, u)$$

For example, in a ring we may define $D(x, y) = x + (-1) * y$.

PROPOSITION 4.10. *Suppose* rts(R, E) *consists of a set of polynomials in several variables. Suppose* (R, E) *are valid in a theory* T *containing an additive identity* 0 *and a multiplicative identity* 1. *Also, suppose that in* T, *addition and multiplication are AC and multiplication distributes over addition. Suppose that* $0 \neq 1$ *in* T, *and that* T *has no zero divisors* (*i.e., if* $x * y = 0$ *then* $x = 0$ *or* $y = 0$). *Suppose that a difference operator* D *as above may be defined in* T. *Let* E *consists of the associative and commutative equations for addition and multiplication. Then* T *is E-separating on* rts(R, E).

Proof. The idea of the proof is to show that if $p(x)$ is a polynomial then $D\ (p(x+1), p(x))$ is a polynomial of lower degree, and if $p(x)$ is identically zero, then $D(p(x+1), p(x))$ is also identically zero. By induction one can show that the coefficient in $p(x)$ of the largest power of x is zero, hence all coefficients are zero. This argument extends to polynomials in several variables. Now, if we have $p(x) = q(x)$ then $r(x) = D(p(x), q(x))$ is identically zero. By a similar argument, one can show that the largest coefficients of $p(x)$ and $q(x)$ must agree; hence all coefficients must agree.

The following result is similar, but is instead based on partial order arguments showing that the largest coefficient of a polynomial must dominate all others in value.

PROPOSITION 4.11. *Suppose* rts(R, E) *consists of a set of polynomials in several variables. Suppose* (R, E) *are valid in a theory* T *containing an additive identity* 0 *and a multiplicative identity* 1. *Also, suppose that in* T, *addition and multiplication are AC and multiplication distributes over addition. Suppose that in* T, $x + z = y + z$ *implies* $x = y$, *and that* $0, 1, 1+1, 1+1+1, \ldots$, *are all distinct in* T. *Suppose there is a norm function* $|\ \ |$ *in* T *mapping to the real numbers, such that* $|x * y| = |x| * |y|$, $|x| \geqslant 0$ *for all* x, $|x+y| \leqslant |x| + |y|$, $|x+y| \geqslant |x| - |y|$, *and* $|n| = n$ *for natural numbers* n. *Let* E *consist of the associative and commutative equations for addition and multiplication. Then* T *is E-separating on* rts(R, E).

Proof. The idea is to show that if $p(x)$ is a polynomial, there are arbitrarily large values of x which cause the highest power of x in $p(x)$ to dominate everything else, so that if $p(x)$ is identically zero then all coefficients of $p(x)$ are zero. The same argument extends to polynomials in several variables. Also, if $p(x) = q(x)$ for all x, then if the coefficients of the largest power of x in p and q are not identical, say the coefficient in $p(x)$ is larger, then for large enough x, $|p(x)| > |q(x)|$, contradiction. The same argument can then be repeated for smaller coefficients since $x + z = y + z$ implies $x = y$. A similar argument applies to polynomials in several variables.

It would be interesting to find out if there are general results subsuming all of these results, which could be used to demonstrate E-separating of T on sets of polynomials.

Acknowledgments

Comments by N. Dershowitz, E. Kounalis, J.-P. Jouannaud, D. Musser, and the referees contributed to this paper.

Received August 23, 1984; accepted January 31, 1985

References

Bachmair, L., and Plaisted, D. (1985), Associative path orderings, *in* "First International Conference on Rewriting Techniques and Applications," Dijon, France, May.

Bergstra, J., and Klop, J. (1982), Conditional rewrite rules: Confluence and termination, Department of Computer Science preprint, Mathematisch Centrum, Amsterdam.

Birkhoff, G., and MacLane, S. (1965), "A Survey of Modern Algebra," MacMillan Co., New York, 1965.

Brand, D., Darringer, J., and Joyner, W. (1978), "Completeness of Conditional Recutions," IBM Research Report RC 7404.

Dershowitz, N. (1982), Orderings for term-rewriting systems, *Theoret. Comput. Sci.* **17**, 279–301.

Dershowitz, N. (1983), "Computing with Rewrite Systems," Report No. ATR-83 (8478)-1, The Aerospace Corp., El Segundo, California.

Dershowitz, N., Hsiang, J., Josephson, A., and Plaisted, D. (1983), Associative–commutative rewriting, *in* "Proc, 10th International Joint Conference on Artificial Intelligence."

Fages, F. (1984), Associative–commutative unification, *in* "Proceedings of the Seventh Conference on Automated Deduction," pp. 194–208.

Futatsugi, K., Goguen, J., Jouannaud, J., and Meseguer, J. (1985), Principles of OBJ 2, *in* "Conference Record of the Twelfth Annual ACM Symposium on Principles of Programming Languages," New Orleans, pp. 52–66.

Goguen, J. (1980), How to prove algebraic inductive hypotheses without induction, with application to the correctness of data type implementation *in* "Proc. Fifth Conference on Automated Deduction."

Goguen, J., Meseguer, J., and Plaisted, D. (1982), Programming with parameterized abstracts objects in OBJ, *in* "Theory and Practise of Software Technology" (Ferrari, D., Bolognani, M. and Goguen, J., Eds.), North-Holland, Amsterdam.

Guttag, J. V., and Horning, J. J. (1978), The abstract specification of abstract data types, *Acta Inform.* **10**, 27–52.

Hoffmann, C., and O'Donnell, M. (1984), Implementation of an interpreter for abstract equations, *in* "Eleventh Annual ACM Symposium on the Principles of Programming Languages," pp. 111–121.

Hopcroft, J., and Ullman, J. (1979), "Introduction to Automata Theory, Languages, and Computation," Addison–Wesley, Reading, Mass.

Hsiang, J., and Dershowitz, N. (1983), Rewrite methods for clausal and non-clausal theorem proving, *in* "Proc. 10th European Assoc. Theoret. Comput. Sci. Intl. Colloq. on Automata, Languages, and Programming," Barcelona, Spain.

Huet, G. (1980), Confluent reductions: abstract properties, and applications to term rewriting systems, *J. Assoc. Comput. Mach.* **27**, 797–821.

Huet, G., and Hullot, J. (1982), Proofs by induction in equational theories with constructors, *J. Comput. System Sci.* **25**, 239–266.

Huet, G., and Oppen, D. (1980), Equations and rewrite rules: A survey, *in* "Formal Languages: Perspectives and Open Problems" (R. Book, Ed.), Academic Press, New York.

Hullot, J. (1980), A catalogue of canonical term rewriting systems, Report CSL-113, SRI International.

Jouannaud, J. (1983), Confluent and coherent sets of reductions with equations, *in* "Application to proofs in Data Types, Proc. 8th Colloquium on Trees in Algebra and Programming," Geneva.

Jouannaud, J., and Kirchner, H. (1984), Completion of a set of rules modulo a set of equations, *in* "Eleventh Annual ACM Symposium on the Principles of Programming Languages," pp. 83–92.

Jouannaud, J., Lescanne, P., and Reinig, F. (1982), Recursive decomposition ordering, *in* "Proceedings 2nd Internat. Federation Inform. Process. Workshop on Formal Description of Programming Concepts," Garmish Partenkirchen, W. Germany.

Kaplan, S. (1984), Conditional rewrite rules, *Theoret. Comput. Sci.* **33**, 175–193.

Knuth, D., and Bendix, P. (1970), Simple word problems in universal algebras, *in* "Computational Problems in Abstract Algebra" (J. Leech, Ed.), pp. 263–297, Pergamon, Oxford.

Lang, S. (1970), "Algebra." Addison–Wesley, Reading, Mass.

Lankford, D. (1975), "Canonical Algebraic Simplification in Computational Logic," Report ATP-25, University of Texas at Austin.

Lankford, D. (1979), "On proving Term Rewriting Systems are Noetherian," Louisiana Tech U., Math Dept. Report MTP-3.

Lankford, D., and Ballantyne, A. (1977), "Decision Procedures for Simple Equational Theories with Permutative Axioms: Complete Sets of Permutative Reductions," Report ATP-37, Department of Computer Science, University of Texas at Austin.

Lescanne, P. (1983), Computer experiments with the REVE term rewriting system generator, *in* "10th ACM Sympos. on Principles of Programmining Languages."

Manna, Z. (1974), "Mathematical Theory of Computation," McGraw–Hill, New York.

Musser, D. (1980), On proving inductive properties of abstract data types, *in* "Proc. 7th ACM Sympos. on Principles of Programming Languages."

Nelson, G., and Oppen, D. (1980), Fast decision procedures based on congruence closure, *J. Assoc. Comput. Mach.* **27**, 356–364.

Oppen, D. (1980), Reasoning about recursively defined data structures, *J. Assoc. Comput. Mach.* **27**, 403–411.

Paterson, M. and Wegman, M. (1978), Linear unification, *J. Comput. System Sci.* **16**, 158–167.

Peterson, G., and Stickel, M. (1981), Complete sets of reduction for some equational theories, *J. Assoc. Comput. Mach.* **28**, 233–264.

Plaisted, D. (1980), Partial correctness and semantic confluence of term-rewriting systems, unpublished, Department of Computer Science, University of Illinois, Urbana, Ill.

Plaisted, D. (1978), "A Recursively Defined Ordering for Proving Termination of Term-rewriting Systems," Dept. of Computer Science Report No. 943, University of Illinois at Urbana-Champaign.

Plotkin, G. (1972), Building in equational theories, *Mach. Intell.* **7**, 73–90.

REMY, J. (1982), "Etude des systèmes de réecriture conditionnels et application aux spécifications algébriques de types abstraits," Thèse d'Etat, University Nancy, Nancy, France.

REMY, J. (1983), Proving conditional identities by equational case reasoning, rewriting, and normalization, Actes seminaire laboratoire informatique theorique de Paris, France.

REMY, J., AND ZHANG, H. (1984), REVEUR 4: A system to proceed experiments on conditional term rewriting systems, unpublished, Centre de Recherche en Informatique de Nancy, Nancy, France.

STICKEL, M. (1981), A unification algorithm for associative-commutative functions, *J. Assoc. Comput. Mach.* **28**, 423–434.

STICKEL, M. (1984), A case study of theorem proving by the Knuth–Bendix Method: Discovering that $x^3=x$ implies ring commutativity, *in* "7th Internat. Conference on Automated Deduction," pp. 248–258.

STONE, M. (1936), The theory of representations for Boolean algebra, *Trans. Amer. Math. Soc.* **40**, 37–111.

THIEL, J. (1984), Stop losing sleep over incomplete data type specifications, *in* "Eleventh Annual ACM Symposium on Principles of Programming Languages," pp. 76–82.

YASUHARA, A. (1971), "Recursive Function Theory and Logic," Academic Press, New York.