SIAM J. COMPUT.
Vol. 17, No. 4, August 1988

001

EXISTENCE, UNIQUENESS, AND CONSTRUCTION OF REWRITE SYSTEMS*

NACHUM DERSHOWITZ†, LEO MARCUS‡, AND ANDRZEJ TARLECKI§

Abstract. The construction of term-rewriting systems, specifically by the Knuth–Bendix completion procedure, is considered. We look for conditions that might ensure the existence of a finite canonical rewriting system for a given equational theory and that might guarantee that the completion procedure will find it. We define several notions of equivalence between rewriting systems in the ordinary and modulo case, and examine uniqueness of systems and the need for backtracking in implementing completion.

Key words. term-rewriting systems, termination, Knuth–Bendix completion procedure, equational theories, rewriting modulo a congruence

AMS(MOS) subject classifications. 68Q50, 68T15, 08B05, 03C05, 03B35

1. Introduction. Much is known about the theory of term-rewriting systems (see [15] and [10] for surveys). However, there have been many unanswered questions about the existence of rewrite systems and the applicability of the Knuth–Bendix completion procedure (KB) [14], [19] and its extensions [16], [21], [26], [3] for finding them. In certain cases KB generates a "canonical" term-rewriting system that decides validity in a given equational theory. In general, the procedure takes a finite set of equations and an ordering on terms, and either halts with success, aborts (fails), or loops infinitely. KB completion is being used in many program-specification and theorem-proving applications (see, for example, [8], [9], [12], [13], [15], [23], [18], and [6]).

Two basic questions arise: when does a given decidable equational theory have a decision procedure in the form of a canonical rewrite system, and when does the Knuth–Bendix procedure generate such a rewrite system for a given equational theory? As a partial answer, the following facts are discussed in this paper:[1]

1. There are decidable equational theories which require an expanded language in order to obtain a canonical rewrite system to decide them, and others which do not have canonical rewrite systems in any language (Examples 1 and 2).
2. For a given reduction ordering KB cannot terminate successfully with two different systems ([6], [24], Theorem 5).
3. Conditions are given which imply the uniqueness of a term-rewriting system modulo a congruence (§ 4). If any one of these conditions is relaxed, uniqueness is lost (Corollary 12, Examples 6–11).
4. For a given theory and ordering on terms, KB cannot both succeed and loop, depending on the choice of equation to orient into a rule ([8], Theorem 17).
5. Given a theory with an equivalent rewrite system, R, and the ordering induced by R, KB will not abort on the first step, but can abort on a later step (Theorem 2, Example 3).
6. KB can abort with different results (Example 12).

* Received by the editors August 26, 1985; accepted for publication (in revised form) July 13, 1987. This research was partially supported by the Aerospace Corporation.

† Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801.

‡ Computer Science Laboratory, The Aerospace Corporation, P.O. Box 92957, Los Angeles, California 90009.

§ Institute of Computer Science, Polish Academy of Sciences, 00-901 Warsaw, Poland.

[1] The numbers refer to theorems and examples in the sections that follow.

7. KB can succeed or abort, depending on choice of representation for the theory (Example 4).
8. KB can succeed or abort, depending on the choice of equation to orient, thus backtracking is necessary (Example 14).
9. Any total monotonic well-ordering on terms allows KB to succeed (without backtracking) on a ground (variable-free) theory ([20], Theorem 18).
10. Success of KB cannot be guaranteed by adding function symbols to the language (Example 13).

These results have important implications for the implementation of the Knuth–Bendix procedure in general, and in the modulo case, for associative commutative theories.

This paper expands on [11], which was originally motivated by a seminar on term-rewriting systems given by the first author at The Aerospace Corporation in the summer of 1981.

2. Definitions and notation. If L is an alphabet (denumerable set of function symbols and constants, signature), $T(L, X)$ denotes the (countably infinite) set of (first-order) terms constructed from symbols in L and variables from a (fixed denumerable) set X. It is well known that $T(L, X)$ with the operations defined in the natural way constitutes a free L-algebra over X, which we denote by $T(L, X)$ as well. Terms are usually denoted by small letters of the English alphabet: l, r, s, t, etc. A *term-rewriting* system (or simply *rewrite system*) R is a set of ordered pairs of terms called *rules*, and is written $l \to r$. (We do not insist a priori that all variables in r are also in l.)

If $\sigma: X \to T(L, X)$ is a substitution of terms for variables, then $l\sigma$ denotes the term l with each occurrence of a variable x that appears in l replaced by the term $x\sigma$ to which x is mapped under σ. We use the notation $s = c[u]$ to indicate that a term s contains a subterm u within context c; then we refer to the term s with that occurrence of u replaced by v as $c[v]$. If a term s contains an instance $l\sigma$ of the left-hand side of a rule $l \to r \in R$, then $s = c[l\sigma]$ *rewrites* to $t = c[r\sigma]$ and we write $s \xrightarrow[R]{1} t$. Clearly, $\xrightarrow[R]{1}$ is closed with respect to substitution and subterm replacement, i.e., $s \xrightarrow[R]{1} t$ implies $c[s\sigma] \xrightarrow[R]{1} c[t\sigma]$, for any context c and substitution σ. A *derivation* in R is a sequence $t_0 \xrightarrow[R]{1} t_1 \xrightarrow[R]{1} t_2 \xrightarrow[R]{1} \cdots$. The notation $s \xrightarrow[R]{*} t$ indicates that there is a (perhaps empty) derivation in R from s to t, and $s \xrightarrow[R]{+} t$ indicates that there is a nonempty derivation in R from s to t. The notation $s \overset{*}{\underset{R}{\leftrightarrow}} t$ indicates that there is a sequence of applications of $\xrightarrow[R]{}$ and $\xleftarrow[R]{}$ between s and t. (In other words, $\xrightarrow[R]{*}$, $\xrightarrow[R]{+}$ and $\overset{*}{\underset{R}{\leftrightarrow}}$ are the reflexive-transitive, transitive, and symmetric-reflexive-transitive closures, respectively, of $\xrightarrow[R]{1}$.)

A rewrite system R is (*finitely*) *terminating* if there is no infinite R-derivation. Note that a system containing a rule $l \to r$ with a variable in r but not in l is nonterminating. A term s is in (R-) *normal form* if there is no t (other than perhaps s) such that $s \xrightarrow[R]{1} t$. (The standard definition of normal form is slightly stronger in that it does not allow a normal form to reduce to itself.) We write $s \xrightarrow[R]{!} t$ if $s \xrightarrow[R]{*} t$ and t is in normal form; $R(s)$ denotes the set of normal forms of s. Two terms r and s are *confluent* in R if there is a t such that $r \xrightarrow[R]{*} t$ and $s \xrightarrow[R]{*} t$, written $r \downarrow_R s$. If R is finite and terminating, then the relation $\downarrow_R$ is recursive. A rewrite system R is *confluent* if for every r and s, $r \overset{*}{\underset{R}{\leftrightarrow}} s$ implies that r and s are confluent in R.

Let E be an equational theory, presented as a set of equations between terms in $T(L, X)$ of the form $s \equiv t$. Thus, $s \equiv t \in E$ means that the given equation actually appears in the set E, while $E \vdash s \equiv t$ means that $s \equiv t$ is valid in E, i.e., it follows from E by equational logic (substitution of equals for equals). Two theories are *equivalent* if they give the same set of valid equations. A theory is *finitely based* if it is presentable as a finite set of equations. The alphabet $L(E)$ (or $L(R)$) is the set of function symbols actually appearing in E (or R). A system R *recognizes* the equational theory E if $E \vdash s \equiv t$ if and only if $s \downarrow_R t$; R recognizes E *in an expanded language* $L(R) \supseteq L(E)$ if for all $s, t \in T(L(E), X)$, $E \vdash s \equiv t$ if and only if $s \downarrow_R t$.

A system R is *canonical* if it is terminating and confluent. It is canonical *for* a theory E if it is terminating and recognizes E. Note that if R recognizes E, then it must be confluent. If R recognizes E and is finite and terminating, then R decides E.

Let $\sim$ be an arbitrary congruence on $T(L, X)$, closed with respect to substitution. A set of rules R is a rewrite system *modulo* a congruence relation $\sim$ if the left and right sides of the rules in R are $\sim$-congruence classes. Operationally, we let the left and right sides be terms, but allow a term u to reduce to v (mod$\sim$) if $u \sim u'[l\sigma]$ and $u'[r\sigma] \sim v$ for some $l \to r \in R$.[2] (If $l \sim l'$, $r \sim r'$, then $l \to r$ and $l' \to r'$ are considered to be the same rule.) Thus, for the modulo case, $\xrightarrow[R]{1}$ is a binary relation on the quotient algebra $T(L, X)/\sim$, closed with respect to substitution and subterm replacement. If $\sim$ is the identity relation, then R modulo $\sim$ reduces to the case of an ordinary rewrite system. All the previous definitions, viz. termination, normal form, and confluence, generalize to modulo systems R.

A prime use for rewrite systems modulo a congruence is in rewrite systems for associative-commutative theories (see [26]). Notice that in the modulo case, if $s \sim t$ and $t \xrightarrow[R]{*} u$, then $s \xrightarrow[R]{*} u$; in particular, if $s \sim t$ then $s \xrightarrow[R]{*} t$.

3. Equational theories and rewrite systems. There are inherent limitations to rewriting. Not all classes of algebras can be described equationally (e.g., fields). Not all equational theories are finitely based (e.g., the equational theory of the algebra over $T(\{\cdot, 0, 1, 2\}, X)$ with binary function $\cdot$ satisfying $0 \cdot x \equiv x \cdot 0 \equiv 0$, $1 \cdot 1 \equiv 0$, $1 \cdot 2 \equiv 1$, $2 \cdot 1 \equiv 2 \equiv 2 \cdot 2$: see [28]). Not all finitely based equational theories are decidable (e.g., some Thue systems; see [7]). Not every decidable equational theory is amenable to canonical (modulo identity) rewriting (e.g., the commutativity axiom). See [5], [17] and [25] for related theorems and examples pertaining to string rewriting. Work on related questions appears in [2].

Example 1. Let $E = \{fg^ihx \equiv fg^jhx\colon 0 \leqq i, j < \omega\}$ be a theory over $T(\{f, g, h, a\}, \{x\})$. Then there is no finite rewrite system recognizing E (since all rules must be of the form $fg^mhx \to fg^nhx$, all but finitely many fg^khx would be irreducible), but there is a finite rewrite system that recognizes E in an expanded language: $R = \{fgx \to fg'x$, $g'gx \to gg'x$, $g'hx \to hx\}$ in the language $\{f, g, h, g', a\}$ $(fg^ihx \xrightarrow[R]{*} fhx)$.

Example 2. Not all decidable equational theories have finite recognizing rewrite systems in extended languages. For example, let E be the theory of one commutative binary function symbol f. Assume R is system in an extended language that recognizes E. Let fxy and fyx reduce to some normal form N. Then a derivation of N from fyx can be obtained from the derivation of N from fxy by reversing the roles of x and y. Thus, N cannot contain the variables x or y. (Either x appears "before" y in N or it does not.) Therefore N is also derivable from fuv, for any variables (or terms) u, v. But this represents a new equality not valid in E, namely $fxy = fuv$, a contradiction.

This raises the open question: For which decidable equational theories are there extended canonical rewrite systems?

Of course, by encoding or ordering variables we may transform a decidable equational theory to a Turing machine, and that Turing machine to a rewrite system.

[2] A weaker system is obtained by just allowing a rule $l \to r$ to be applicable to any term $l' \sim l\sigma$, yielding a term $r' \sim r\sigma$; the congruence cannot be applied in the rest of u. See [26] and [16].

But that rewrite system does not serve as a canonical system for the original theory. Next we give some examples of the relation between an equational theory and a finite canonical rewrite system that recognizes it. First, a lemma in which E and R need not be related.

LEMMA 1. *Let R be any rewrite system and E any equational theory. If $E \vdash u \equiv v$ for some term v in R-normal form and any other term u, $u \neq v$, then there is an axiom $s \equiv t \in E$ such that s or t is in R-normal form.*

Proof. Let $u = w_0 \equiv w_1 \equiv \cdots \equiv w_n = v$ $(n \geqq 1)$ be an equational proof of $u \equiv v$ in E. That is, for each i $(1 \leqq i \leqq n)$, $w_{i-1} = c[s_i\sigma]$ and $w_i = c[t_i\sigma]$, for some context c, substitution σ, and axiom $s_i \equiv t_i$ (or $t_i \equiv s_i$) in E. Since $v = w_n$ is in R-normal form, t_n must also be. Hence $s_n \equiv t_n$ (or $t_n \equiv s_n$) is an axiom of E, one side of which (t_n) is in R-normal form. □

THEOREM 2. *If R is a canonical rewrite system for an equational theory E, and E is nontrivial, then there is $s \equiv t \in E$ such that $s \xrightarrow[R]{+} t$ (or $t \xrightarrow[R]{+} s$).*

Proof. $R \neq \varnothing$ since E is not trivial. Therefore, there is $s' \rightarrow t' \in R$ such that t' is in R-normal form; otherwise R would not be terminating. Now, $E \Vdash s' \equiv t'$, and so by Lemma 1, there is t in R-normal form and s such that $s \equiv t \in E$. But then it follows that $s \xrightarrow[R]{+} t$. □

The above theorem however does not generalize to more than one equation. Instead, we have that even if R is a finite canonical rewrite system for $E' \cup R'$ (interpreting the rules in R' as equations), every term appearing in E' is in R'-normal form, E' is nontrivial, and $R' \subseteq R$, there still may be no $s \equiv t \in E'$ such that $s \xrightarrow[R]{+} t$ or $t \xrightarrow[R]{+} s$.

Example 3. Let $R' = \{1x \rightarrow x,\ x1 \rightarrow x,\ xx^{-1} \rightarrow 1,\ x^{-1}x \rightarrow 1,\ x^{-1-1} \rightarrow x,\ x(x^{-1}y) \rightarrow y,\ x^{-1}(xy) \rightarrow y,\ (xy)^{-1} \rightarrow y^{-1}x^{-1},\ (xy)z \rightarrow x(yz)\}$, and $R = R' \cup \{1^{-1} \rightarrow 1\}$. Then R is a finite canonical rewrite system for groups (see [15]) and is equivalent to $R' \cup \{1^{-1}x \equiv x1^{-1}\}$. (It is sufficient to show that $R' \cup \{1^{-1}x \equiv x1^{-1}\} \vdash 1^{-1} = 1$.) The terms $1^{-1}x$ and $x1^{-1}$ are in R'-normal form, but $1^{-1}x \mathrel{\not\xrightarrow[R]{+}} x1^{-1}$ and $x1^{-1} \mathrel{\not\xrightarrow[R]{+}} 1^{-1}x$. (Of course, both terms rewrite in R to x.)

Even if $E = \{s_1 \equiv t_1, \cdots, s_n \equiv t_n\}$ has a finite canonical rewrite system, and the equations in E are independent, there need not be a finite canonical S for E such that *for every i, $s_i \rightarrow t_i \in S$* or $t_i \rightarrow s_i \in S$.

Example 4. Let E be the axiomatization $\{x(yz) \equiv (xy)z,\ xx^{-1} \equiv 1,\ 1x \equiv x,\ x1 \equiv 1x\}$ of group theory. It can be shown that E is independent, but there can be no finite canonical rewrite system in which $x1 \rightarrow 1x$ or vice versa.

Even if E has a finite canonical rewrite system, $E \vdash s_i \equiv t_i$ for $i = 1, \cdots, n$, and $\{s_1 \rightarrow t_1, \cdots, s_n \rightarrow t_n\}$ is terminating, there need not be a canonical R for E with $s_i \rightarrow t_i \in R$ for all i.

Example 5. Let $E = \{a \equiv b\}$, in which case $E \vdash fa \equiv fb,\ gb \equiv ga$. The system $R = \{fa \rightarrow fb,\ gb \rightarrow ga\}$ is terminating, but cannot be extended to include either $a \rightarrow b$ or $b \rightarrow a$ without engendering nontermination.

LEMMA 3. *Let S be a finite canonical rewrite system which recognizes an equational theory E and let R be any system with $\xrightarrow[R]{+} \subseteq \xrightarrow[S]{+}$ that does not recognize E. Then there are terms s and t in R-normal form such that $E \vdash s \equiv t$ and $s \xrightarrow[S]{+} t$.*

Proof. If R is terminating (since S is) and does not recognize E, then there must exist two distinct R-normal forms, u and v, such that $E \vdash u \equiv v$. Let t be the common S-normal form of u and v. Then t is also in R-normal form. Since $u \neq v$, we have $u \xrightarrow[S]{+} t$ or $v \xrightarrow[S]{+} t$ (or both). □

4. Uniqueness. In this section we examine various criteria for similarity of rewrite systems with the goal of characterizing varieties of uniqueness.

First, let us give some informal examples which answer some of the obvious questions. Must two rewrite systems that decide the same theory be identical? No; consider $R=\{fx \to a\}$, $S=\{fy \to a\}$. Must they be the same up to variable renaming? No; consider $R=\{a \to b\}$, $S=\{b \to a\}$. What if $R \cup S$ has no cycles? No: $R=\{a \to b, b \to c\}$, $S=\{a \to c, b \to c\}$. What if the right-hand sides are irreducible? No: $R=\{fx \to gx,\ fa \to c,\ ga \to c\}$, $S=\{fx \to gx,\ ga \to c\}$. Assume in addition that the left-hand sides reduce to only one term. Then we do get that R and S have the same "semantics," i.e., $\xrightarrow[R]{1} = \xrightarrow[S]{1}$ (compare Theorem 10). But we still have the example of $R=\{fa \to fb,\ a \to b\}$, $S=\{a \to b\}$. If no rule contains an instance of another, then $R=S$ up to renaming of variables (see Theorem 5).

Let us illustrate additional problems we have to cope with in the case of modulo systems.

Example 6. The following two different modulo systems decide the same theory over $T(\{f, g, a, b\}, \{x\})$ and satisfy the conditions mentioned above for ordinary (modulo identity) rewrite systems: $R=\{a \to b\}$, $S=\{ga \to gb\}$, and $\sim$ is presented by $\{fgx \sim x\}$.

Example 7. What if no left-hand side is congruent to its own subterm? Still the systems can be different: Take $fghx \sim fx$, $R=\{fx \to a\}$, $S=\{fgx \to a\}$.

As we will see, if no nontrivial instance of a left-hand side is a proper subterm of a congruent term, then the systems are the same up to renaming and congruence (see Corollary 12 and Lemma 13).

Now more formally, given rewrite systems R and S, some notions of similarity are:

(1) If R and S satisfy for all $l \to r \in R$ there is a renaming of variables θ such that $l\theta \to r\theta \in S$, we say that R is *syntactically contained in* S. If S is also syntactically contained in R, then they are *isomorphic*, or *syntactically equivalent.*

(2) If R and S satisfy for all $l \to r \in R$, $l \xrightarrow[S]{1} r$, we say that R is *semantically contained in* S. If S is also semantically contained in R, we say they are *semantically equivalent.*

(3) If R and S satisfy for all $l \to r \in R$, $l \xrightarrow[S]{*} r$ (or equivalently: for all l and r, if $l \xrightarrow[R]{*} r$, then $l \xrightarrow[S]{*} r$), we say that R is *derivationally contained in* S. If also S is derivationally contained in R, we say they are *derivationally equivalent.*

(4) If R and S satisfy for all terms s and t, if $s \downarrow_R t$, then $s \downarrow_S t$, we say that R is *operationally contained in* S. If S is also operationally contained in R, then they are *operationally equivalent.*

(5) If R and S satisfy for all terms s and t, if $s \overset{*}{\underset{R}{\leftrightarrow}} t$, then $s \overset{*}{\underset{S}{\leftrightarrow}} t$, we say that R is *deductively contained in* S. If S is also deductively contained in R, then they are *deductively equivalent.*

These criteria can be compactly defined as:

(1)	$R \subseteq S$ (up to renaming)	syntactic containment,
(2)	$\xrightarrow[R]{1} \subseteq \xrightarrow[S]{1}$	semantic containment,
(3)	$\xrightarrow[R]{*} \subseteq \xrightarrow[S]{*}$	derivational containment,
(4)	$\downarrow_R \subseteq \downarrow_S$	operational containment,
(5)	$\overset{*}{\underset{R}{\leftrightarrow}} \subseteq \overset{*}{\underset{S}{\leftrightarrow}}$	deductive containment,
(6)	$R=S$	isomorphism,
(7)	$\xrightarrow[R]{1} = \xrightarrow[S]{1}$	semantic equivalence,

(8) $\xrightarrow[R]{*} = \xrightarrow[S]{*}$ derivational equivalence,

(9) $\downarrow_R = \downarrow_S$ operational equivalence,

(10) $\overset{*}{\underset{R}{\leftrightarrow}} = \overset{*}{\underset{S}{\leftrightarrow}}$ deductive equivalence.

These criteria all apply to rewriting modulo a congruence $\sim$ as well (in which case if $l\theta \sim l'$ and $r\theta \sim r'$ for some variable renaming θ, then $l \to r$ and $l' \to r'$ are considered the same).

LEMMA 4. *The above criteria satisfy*: $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4) \Rightarrow (5)$, *and hence* $(6) \Rightarrow (7) \Rightarrow (8) \Rightarrow (9) \Rightarrow (10)$.

Proof. All clear, except perhaps $(4) \Rightarrow (5)$, for which we note that $\overset{*}{\underset{R}{\leftrightarrow}}$ and $\overset{*}{\underset{S}{\leftrightarrow}}$ are the transitive closures of $\downarrow_R$ and $\downarrow_S$, respectively. □

The reverse implications $(5) \Rightarrow (4) \cdots$ do not hold. The obvious question is: Under what (interesting) additional conditions do they hold?

For ordinary rewrite systems (mod identify) we say that R is *reduced* if all right-hand sides are in R-normal form and all left-hand sides are in normal form with respect to all other rules in R. Any ordinary canonical system can be reduced to a deductively equivalent one ([24]; see also [2]).

The following is known.

THEOREM 5 (due to M. Ballantyne; mentioned without proof in [6]; a proof is given in [24]). *If two reduced systems R and S both recognize the same equational theory E, and if for some strict partial order* $>, \xrightarrow[R]{1}, \xrightarrow[S]{1} \subseteq >$, *then R and S are isomorphic.*

We include the proof here for motivation, even though this is a special case of Corollary 12.

Proof. Let R and S be two canonical rewrite systems that recognize E, and for which $\xrightarrow[R]{+}, \xrightarrow[S]{+} \subseteq >$. Assume $l \to r \in R$. Since $E \vdash l \equiv r$ and S recognizes E, there is a term u such that $l \xrightarrow[S]{*} u$ and $r \xrightarrow[S]{*} u$. Since $E \vdash r \equiv u$, R recognizes E, and R is reduced, $u \xrightarrow[R]{!} r$. Thus we have that $r \geqq u$ and $u \geqq r$ ($\geq$ is the reflective closure of $>$), and so $r = u$. It follows that $l \xrightarrow[S]{+} r$.

Similarly, $l \to r \in S$ implies $l \xrightarrow[R]{+} r$, and thus $s \xrightarrow[S]{1} t$ implies $s \xrightarrow[R]{+} t$.

Assume now that $l \to r \in R$ but $l \to r \notin S$ (even up to a renaming of variables). If $l \xrightarrow[S]{1} t \xrightarrow[S]{+} r$ for some t, then $l \xrightarrow[R]{+} t \xrightarrow[R]{+} r$. But this contradicts the premise that no rule in R other than $l \to r$ can reduce l. So it must be that $l \xrightarrow[S]{1} r$ by a single application of a rule $l' \to r' \in S$ such that $l \neq l'$ (even up to a renaming of variables). But then, too, $l' \xrightarrow[R]{+} r'$ would contradict R being reduced. □

We can extend and split the definition of reduced for R mod$\sim$by using the following definitions:

(i) R is *rhs-reduced* if the right-hand side of each rule is in R-normal form mod$\sim$;

(ii) R is *lhs-reduced* if $l \xrightarrow[R]{1} u \xrightarrow[R]{*} r$ implies $r \sim u$ for any left-hand side l of a rule $l \to r \in R$;

(iii) R is *reduced* if it is both rhs-reduced and lhs-reduced.

But the uniqueness theorem does not extend immediately.

Example 8. The following reduced, canonical systems R and S mod$\sim$for the same theory over $T(\{f, g, c, d\}, \varnothing)$ are not isomorphic: Let $\sim$ be presented by $fgc \sim c$, $fgd \sim d$, and take $R = \{c \to d\}$, $S = \{gc \to gd\}$.

THEOREM 6. *Let R be a rewrite system* mod$\sim$ *deductively contained in a confluent system S* mod$\sim$. *Then R is operationally contained in S.*

Proof. Let $s \downarrow_R t$. Then $s \overset{*}{\underset{S}{\leftrightarrow}} t$ by deductive containment and $s \downarrow_S t$ by confluence of S. □

COROLLARY 7. *For confluent systems R and S* mod $\sim$, *R and S are operationally equivalent if and only if* $l \downarrow_S r$ *for all* $l \to r \in R$ *and* $l \downarrow_R r$ *for all* $l \to r \in S$.

Proof. If $l \to r \in R$ implies $l \downarrow_S r$, then $s \xrightarrow[R]{1} t$ implies $s \downarrow_S t$ and R is deductively contained in S. □

We say that two rewrite systems mod $\sim$, R and S, are *noninverting* if for all terms s and t, $s \xrightarrow[R]{*} t \xrightarrow[S]{*} s$ implies $s \sim t$.

LEMMA 8. *Let R and S be two operationally equivalent systems* mod $\sim$. *If R and S are noninverting, then* $\xrightarrow[R]{!} = \xrightarrow[S]{!}$.

Proof. We show only one direction; the other follows from symmetry. Let $s \xrightarrow[R]{!} t$. By operational equivalence there is a u such that $s \xrightarrow[S]{*} u$ and $t \xrightarrow[S]{*} u$. Again, by operational equivalence, $t \xrightarrow[R]{*} v$, $u \xrightarrow[R]{*} v$ for some v. Since t is an R-normal form, we have $u \xrightarrow[R]{*} t$. By the noninverting property, it must then be that $u \sim t$, and therefore $s \xrightarrow[S]{*} t$. Suppose $t \xrightarrow[S]{+} w$ for some w. As before, we must have $w \sim t$. But then t is an S-normal form, since it only reduces to congruent terms. □

Example 9. The following pair, R and S, of operationally equivalent rewrite systems modulo the identity are not derivationally equivalent: $R = \{a \to b, c \to b\}$, $S = R \cup \{a \to c\}$.

THEOREM 9. *Let R and S be two rewrite systems* mod $\sim$ *for which R is operationally contained in S, R is rhs-reduced, and every R-normal form is an S-normal form. Then R is derivationally contained in S.*

Note that if $\xrightarrow[R]{!} \subseteq \xrightarrow[S]{!}$, then trivially any R-normal form is an S-normal form.

Proof. Let $l \to r \in R$. Then $l \downarrow_S r$. But since R is rhs-reduced and all R-normal forms are S-normal forms, it follows that $l \xrightarrow[S]{*} r$. □

Example 10. The following pair, R and S, of derivationally equivalent systems mod $\sim$ over $T(\{f, g, a, b\}, \emptyset)$ are not semantically equivalent: Let $\sim$ be presented by $\{fga \sim a\}$, $R = \{a \to b, fgb \to b\}$, $S = \{ga \to gb, fgb \to b\}$.

THEOREM 10. *Let R and S be two reduced rewrite systems* mod $\sim$ *which are derivationally equivalent. Then R and S are semantically equivalent.*

Proof. We shall prove that $l \to r \in R$ implies $l \xrightarrow[S]{1} r$. Let $l \to r \in R$. Since $l \not\sim r$ and R is derivationally contained in S, $l \xrightarrow[S]{+} r$. Now assume that $l \xrightarrow[S]{1} u \xrightarrow[S]{*} r$, $l \not\sim u$. Since S is derivationally contained in R, there is some v such that $l \xrightarrow[R]{1} v \xrightarrow[R]{*} u \xrightarrow[R]{*} r$. R is lhs-reduced so $v \sim r$, and so $r \xrightarrow[R]{*} u$. Now using the fact that R is rhs-induced, $r \sim u$ and thus $l \xrightarrow[S]{1} r$. □

Were we to weaken the definition of reduced to allow trivial rules $l \to r$, where $l \sim r$, then we would only get $\xrightarrow[R]{0,1} = \xrightarrow[S]{0,1}$, which is slightly weaker than semantic equivalence.

Example 11. The following pair, R and S, of systems mod $\sim$ over $T(\{f, g, c\}, \{x\})$ are semantically equivalent, but not isomorphic: Let $\sim$ be presented by $fx \sim fghx$, and take $R = \{fx \to c\}$, $S = \{fgx \to c\}$.

A rewrite system R is *minimal* if no left-hand side l of a rule in R contains a proper instance of a left-hand side l' of a (not necessarily different) rule in R, that is, $l \not\sim c[l'\sigma]$ for any nonempty context c or substitution σ that is more than a renaming of variables in l'.

THEOREM 11. *Suppose R and S are semantically equivalent rewrite systems mod* $\sim$. *If R and S are both minimal, then R and S are isomorphic.*

Proof. Let $l \to r \in R$. Then $l \sim u[l'\sigma] \xrightarrow[S]{1} u[r'\sigma] \sim r$ for some $l' \to r' \in S$. Similarly, $l' \sim v[\bar{l}\mu] \xrightarrow[R]{1} v[\bar{r}\mu] \sim r'$ for some $\bar{l} \to \bar{r} \in R$. That is, $l \sim u[v[\bar{l}\mu]\sigma]$ and $r \sim u[v[\bar{r}\mu]\sigma]$. By minimality, u and v are empty and σ and μ are no more than renamings (on the appropriate sets of variables, μ on variables of $\bar{l}$, σ on variables of $\bar{l}\mu$). Hence, $l \sim u[l'\sigma] = l'$ and $r \sim u[r'\sigma] = r'$ as desired up to renaming. □

COROLLARY 12. *Let R and S be deductively equivalent confluent rewrite systems* mod$\sim$. *If R and S are noninverting and R and S are each reduced and minimal, then R and S are isomorphic.*

This extends a result of Lankford and Ballantyne [22] in which they assume that the $\sim$-congruence classes are finite. None of the conditions can be omitted.

LEMMA 13. *A rewrite system R* mod$\sim$ *is minimal if no left-hand side l is reducible by another rule, and if $l \not\sim c[l\sigma]$ for any nonempty context c or substitution σ that is more than renaming.*

Proof. Straightforward from the definition of minimality. □

LEMMA 14. *A terminating rewrite system R* mod$\sim$ *is reduced if it is minimal and no rule applies to either the left- or right-hand side of another rule.*

Proof. Let $l \to r \in R$. We are given that no other rule applies to l and by minimality, $\sim$ does not create any nontrivial way for l to apply to itself. Furthermore, the right-hand side r must be a normal form, since no other rule applies, and were l to apply, the system would not be terminating. Hence R is reduced, □

COROLLARY 15. *A rewrite system R* mod$\sim$ *is minimal if no left-hand side l is reducible by another rule and $\sim$ generates only finite congruence classes.*

Proof. Were $l \sim c[l\sigma]$ for nonempty context c, then there would be an infinite congruence class $l \sim c[l\sigma] \sim c[c[l\sigma]\sigma] \sim \cdots$. Suppose, then, that $l \sim l\sigma$. If either l or $l\sigma$ contains a variable x not in the other, then by substituting terms for x we would obtain an infinite set of congruent terms. So were σ not a renaming it would have to map some variable to a nonvariable, in which case $l \sim l\sigma \sim l\sigma\sigma \sim \cdots$ would be an infinite congruence class. □

It is however undecidable if $\sim$-congruence classes are finite, even if $\sim$ is presented by a finite, canonical rewrite system. If, however, $\sim$ is presented by a finite set of ground equations, then finiteness is decidable (see [27]).

5. Completion. In this section we consider the relationship between the Knuth-Bendix completion procedure [19] and the existence of a finite canonical rewrite system for an equational theory. The procedure has been extended to the modulo $\sim$ case by [16], [21], [26], [3], and [4], although we shall not consider that more complicated procedure here.

Let $l \to r$ and $l' \to r'$ be two (not necessarily distinct) rules in R whose variables have been renamed, if necessary, so that they are distinct. Assume l' overlaps l, that is, $l = c[v]$ for some context c and nonvariable subterm v such that $v\sigma = l'\sigma$ for some (most general) substitution σ for the variables of l and l'. Then the overlapped term $l\sigma$ $(= c\sigma[l'\sigma])$ can be rewritten as either $r\sigma$ or $c\sigma[r'\sigma]$. These two possibilities are called a *critical pair.*

The *Knuth-Bendix completion procedure* (KB$(E, >)$ or just KB) takes as input a finite set E of equations and (a program to compute) a well-founded ordering $>$ that is closed with respect to substitution and subterm replacement, i.e., a well-founded partial order on $T(L(E), X)$ such that $c[s\sigma] > c[t\sigma]$where $s > t$, for any context c, substitution σ, and terms s and t.

The procedure consists of the following steps:

Repeat as long as equations are left in E. If none remain, terminate successfully.

(1) Remove an equation $s \equiv t$ (or $t \equiv s$) from E such that $s > t$. If none exists, terminate with failure (abort).
(2) Add the rule $s \to t$ to R.
(3) Use R to reduce the right-hand sides of existing rules.

(4) Add to E all critical pairs formed using the new rule.
(5) Remove from R all other rules whose left-hand side contains an instance of s.
(6) Use R to reduce both sides of equations in E. Remove any equation whose reduced sides are identical.

We use the notation $R \cup E \xrightarrow[\text{KB}]{} R' \cup E'$ to indicate the effect of one iteration of the above procedure. For an abstract version see [4]. The procedure can be optimized in various ways that do not concern us here.

THEOREM 16 [19], [14]. *If* KB *terminates successfully for input equations E, then it returns as output a finite canonical system R for E.*

The completion procedure may fail in one of two fashions: it may be unable to add any rule because s and t are incomparable under the given ordering $>$ for all equations $s \equiv t$ in E; or it may go on generating an infinite number of new rules without ever finding a finite canonical system. In the former case, we will say that the procedure *aborts*, in the latter that it *loops*. Clearly, KB$(E, >)$ only generates rules $l \rightarrow r$ that are true in the theory E, i.e., $E \vdash l \equiv r$, and for which $l > r$.

THEOREM 17 ([8], based on [14]). *Let R be a reduced finite canonical rewrite system for a finite equational theory E, and let* KB *be implemented by a "fair" scheduler, i.e., no critical pair that persists is ignored indefinitely. Then* KB$(E, \xrightarrow[R]{+})$ *will generate R if it does not abort.*

Proof. Assume KB does not abort. Then in the limit (i.e., considering only those rules which persist from some point on) a reduced canonical system S will be generated for E (see [14] and [3]). By Theorem 5, $R = S$. KB terminates once R is generated, since critical pairs must now reduce to triviality. □

Even if R is a reduced finite canonical rewrite system for E, KB$(E, \xrightarrow[R]{+})$ need not generate R since it might abort. By Theorem 2, KB$(E, \xrightarrow[R]{+})$ will not abort on the first step.

Example 12. Despite the existence of $R = \{m \rightarrow c,\ n \rightarrow c,\ fc \rightarrow c\}$, KB$(\{fn \equiv c, fm \equiv m,\ m \equiv n\}, \xrightarrow[R]{+})$, aborts on all paths (see also [1].)

Example 13. As pointed out in [19], varying the procedure by expanding the language (adding rules $s \rightarrow h(\bar{x})$ and $t \rightarrow h(\bar{x})$, where h is a new function symbol and $\bar{x}$ are the variables in an unorientable equation $s \equiv t$) may sometimes circumvent the abort case of completion. However, the following example shows that such expansions may also cause KB to become nonterminating: $\{fm \equiv m,\ fn \equiv c,\ m \equiv n\} \xrightarrow[\text{KB}]{} \{fh \equiv h, fh \equiv c,\ m \rightarrow h,\ n \rightarrow h\} \xrightarrow[\text{KB}]{} \{fh \rightarrow h,\ h \equiv c,\ m \rightarrow h,\ n \rightarrow h\} \xrightarrow[\text{KB}]{} \{fk \equiv k,\ c \rightarrow k,\ h \rightarrow k, m \rightarrow k,\ n \rightarrow k\} \xrightarrow[\text{KB}]{} \cdots \xrightarrow[\text{KB}]{} \{fl \equiv l,\ k \rightarrow l, \cdots\} \xrightarrow[\text{KB}]{} \cdots$ (using the subterm ordering for $>$).

We now show that backtracking is sometimes necessary, that is, there are E and $>$ for which KB$(E, >)$ can either abort or succeed depending on the sequence in which the equations are chosen for orienting.

Example 14. Given the theory

$$E = \{k \equiv m,\ k \equiv n,\ fm \equiv m,\ fk \equiv c\}$$

and the ordering

$$k > m, n,$$

$$m, n, fm, fn, fc > c,$$

$$fm > m,$$

$$fn > n,$$

$\#(m, n, fc)$, $\#(k, fm, fn, fc)$, etc. ($\#$designates pairwise incomparability)

Knuth-Bendix can either succeed and find:

$$E \xrightarrow[\mathrm{KB}]{} \{k \to m, m \equiv n, fm \equiv m, fm \equiv c\} \xrightarrow[\mathrm{KB}]{} \{k \to m, m \equiv n, fm \to m, m \equiv c\}$$
$$\xrightarrow[\mathrm{KB}]{} \{k \to c, n \to c, fc \to c, m \to c\}$$

or else it can fail (abort) along the path:

$$E \xrightarrow[\mathrm{KB}]{} \{n \equiv m, k \to n, fm \equiv m, fn \equiv c\} \xrightarrow[\mathrm{KB}]{} \cdots \xrightarrow[\mathrm{KB}]{} \{n \equiv m, k \to n, fm \to m, fn \to c\}.$$

Hence, KB should backtrack to try alternative choices, even for ground systems. Note that if in the previous example n and k were incomparable or the given order were extended to include either $m > n$ or $n > m$, then KB would succeed along every path.

In general, we have Theorem 18.

THEOREM 18. *If the set of terms $T(L, X)$ is totally ordered by $>$, then* KB$(E, >)$ *does not need to backtrack.*

Proof. Obviously, KB cannot abort, and we have already seen (Theorem 17) that KB cannot both succeed and loop. □

As pointed out in [20], there is always such a total ordering for ground systems. As pointed out in [15], there need not be such an ordering for nonground systems.

6. Conclusion. We have discussed the question of when a given decidable equational theory possesses a canonical rewrite system in the original or expanded language. We have pointed to some inherent limitations in the ability of the Knuth-Bendix completion procedure to discover appropriate rewrite systems, even if the procedure is extended to backtrack upon failure or to introduce new function symbols. These limitations can be partially circumvented by allowing more deduction in the procedure (see [20], [3], [4]). In practice completion is frequently quite effective. We have shown that barring failure (and under reasonable conditions) the procedure will find the same rewrite system regardless of the choices made.

One of the remaining open questions is: Suppose there exists some finite canonical system for an equational theory E. Must there exist an ordering $>$ for which the completion algorithm, given E and $>$, has a successful outcome. Another area worth investigating is the extent to which systems that rewrite modulo a congruence (whose classes are all finite) are sure to exist for decidable E.

Acknowledgments. The authors gratefully acknowledge discussions with Dallas Lankford and Pierre Lescanne.

REFERENCES

[1] J. AVENHAUS, *On the termination of the Knuth-Bendix completion algorithm*, 120/84, Universität Kaiserslautern, Kaiserslautern, West Germany, 1985.

[2] ———, *On the descriptive power of term rewriting systems*, J. Symbolic Computation, 2 (1986), pp. 109–122.

[3] L. BACHMAIR, *Proof methods for equational theories*, Ph.D. thesis, University of Illinois, Urbana, IL, 1986.

[4] L. BACHMAIR, N. DERSHOWITZ, AND J. HSIANG, *Orderings for equational proofs*, Proc. Symposium on Logic in Computer Science, Cambridge, MA, June, 1986, pp. 346–357.

[5] R. BOOK, *Thue systems as rewriting systems*, Proc. First International Conference on Rewriting Methods and Techniques, Dijon, France, 1985.

[6] G. BUTLER AND D. LANKFORD, *Experiments with computer implementations of procedures which often derive decision algorithms for the word problem in abstract algebras*, MTP-7, Lousiana Tech. University, Ruston, LA, August, 1980.

[7] M. DAVIS, *Computability and Unsolvability*, McGraw-Hill, New York, London, Toronto, 1958.

[8] N. DERSHOWITZ, *Applications of the Knuth-Bendix completion procedure*, Proc. Seminaire d'Informatique Théoretique, Univ. Paris VII, Paris, 1983.

[9] ———, *Computing with rewrite systems*, Inform. and Control, 65 (1985), pp. 122-157.

[10] N. DERSHOWITZ AND J.-P. JOUANNAUD, *Rewrite systems*, in Handbook of Theoretical Computer Science, A. Meyer, M. Nivat, M. Paterson, and M. Perrin, eds., North-Holland, Amsterdam, to appear.

[11] N. DERSHOWITZ AND L. MARCUS, *Existence and construction of rewrite systems*, ATR-82(8478)-3, The Aerospace Corporation, December, 1982.

[12] F. FAGES, *Le système KB: manual de référence: présentation et bibliographie, mise en oeuvre*, R. G. 10.84, Greco de Programmation, C.N.R.S., Bordeaux, France, 1984.

[13] S. GERHART, D. MUSSER, D. THOMPSON, D. BAKER, R. BATES, R. ERICKSON, R. LONDON, D. TAYLOR, AND D. WILE, *An overview of* AFFIRM. *A specification and verification system*, in Information Processing, IFIP, North-Holland, Amsterdam, 1980.

[14] G. HUET, *A complete proof of correctness of the Knuth-Bendix completion algorithm*, J. Comput. System Sci., 23 (1981), pp. 11-21.

[15] G. HUET AND D. C. OPPEN, *Equations and rewrite rules: a survey*, in Formal Language Theory: Perspectives and Open Problems, R. Book, ed., Academic Press, New York, 1980, pp. 349-405.

[16] J.-P. JOUANNAUD AND H. KIRCHNER, *Completion of a set of rules modulo a set of equations*, SIAM J. Comput., 15 (1986), pp. 1155-1194.

[17] D. KAPUR AND P. NARENDRAN, *A finite Thue system with decidable word problem and without equivalent finite canonical system*, Theoret. Comput. Sci., 35 (1985), pp. 337-344.

[18] D. KAPUR AND G. SIVAKUMAR, *Experiments with and architecture of* RRL, *a rewrite rule laboratory*, Proceedings of an NSF Workshop on the Rewrite Rule Laboratory, 1983. Available as Report 84GEN008, General Electric Research and Development, Schenectady, NY, April 1984.

[19] D. KNUTH AND P. BENDIX, *Simple Word Problems in Universal Algebras*, in Computational Problems in Abstract Algebra, J. Leech, ed., Pergamon Press, Oxford, 1970.

[20] D. LANKFORD, *Canonical inference*, ATP-32, University of Texas, Austin, TX, December, 1975.

[21] D. LANKFORD AND M. BALLANTYNE, *Decision procedures for simple equational theories with permutative axioms: complete sets of permutative reductions*, ATP-37, Departments of Mathematics and Computer Sciences, University of Texas, Austin, TX, April, 1977.

[22] ———, *On the uniqueness of term rewriting systems*, unpublished manuscript, 1983.

[23] P. LESCANNE, *Computer experiments with the* REVE *term rewriting system generator*, Proc. Tenth ACM Symposium on Principles of Programming Languages, Austin, TX, 1983, pp. 99-108.

[24] Y. METIVIER, *About the rewriting systems produced by the Knuth-Bendix completion algorithm*, Inform. Process. Lett., 16 (1983), pp. 31-34.

[25] F. OTTO, *Finite complete rewriting systems for the Jantzen monoid and the Greendlinger group*, Theoret. Comput. Sci., 32 (1984), pp. 249-260.

[26] G. E. PETERSON AND M. E. STICKEL, *Complete sets of reduction for some equational theories*, J. Assoc. Comput. Math., 28 (1981), pp. 233-264.

[27] J.-C. RAOULT, *Finiteness results on rewriting systems*, RAIRO Theoretical Informatics, 15 (1981), pp. 373-391.

[28] W. TAYLOR, *Equational logic*, in Universal Algebra by G. Gratzer, 2nd ed., Springer-Verlag, Berlin, New York, Heidelberg, 1979, pp. 378-400.