EQUATIONAL INFERENCE, CANONICAL PROOFS, AND PROOF ORDERINGS*

Leo Bachmair
Department of Computer Science
State University of New York at Stony Brook
Stony Brook, New York 11794, U.S.A.

Nachum Dershowitz
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801, U.S.A.

April 14, 1992

Abstract

We describe the application of proof orderings—a technique for reasoning about inference systems—to various rewrite-based theorem-proving methods, including refinements of the standard Knuth-Bendix completion procedure based on critical pair criteria; Huet's procedure for rewriting modulo a congruence; ordered completion (a refutationally complete extension of standard completion); and a proof by consistency procedure for proving inductive theorems.

*This is a substantially revised version of the paper, "Orderings for equational proofs," co-authored with J. Hsiang and presented at the *Symp. on Logic in Computer Science* (Boston, Massachusetts, June 1986). It includes material from the paper "Proof by consistency in equational theories," by the first author, presented at the *Third Annual Symp. on Logic in Computer Science* (Edinburgh, Scotland, July 1988). This research was supported in part by the National Science Foundation under grants CCR-89-01322, CCR-90-07195, and CCR-90-24271.

1. Introduction

Applications of equational reasoning to computer science comprise the use of equations within programming languages and for specifications of abstract data types. When equations are interpreted as programs, they are used in one direction only, to rewrite an expression to a simpler one. A *rewrite system* is a set of such one-way rules. For example, the following system serves as a program for adding and multiplying Peano numbers (0, $0'$, $0''$, etc.):

$$\begin{array}{rcl} x+0 & \rightarrow & x \\ x+y' & \rightarrow & (x+y)' \\ x\times 0 & \rightarrow & 0 \\ x\times y' & \rightarrow & (x\times y)+x \end{array}$$

Equational programs such as this one are usually considered to be nondeterministic: an expression can be rewritten by any equation, the left side of which matches a subterm. For example, one possible computation proceeds as follows:

$$\begin{array}{lcll} 0''\times 0'' & \rightarrow & (0''\times 0')+0'' & \rightarrow \\ ((0''\times 0)+0'')+0'' & \rightarrow & (((0''\times 0)+0'')+0')' & \rightarrow \\ (((0''\times 0)+0')'+0')' & \rightarrow & ((0+0')'+0')' & \rightarrow \\ ((0+0')'+0)'' & \rightarrow & ((0+0)''+0)'' & \rightarrow \\ (0''+0)'' & \rightarrow & 0'''' & \end{array}$$

Here "two times two" reduces, by a sequence of rewrites, to the value "four."

Many rewrite systems satisfy the Church-Rosser, or "confluence," property: whenever two terms are equivalent, they can be rewritten to a common form. This guarantees that the "normal forms" (unrewritable terms) computed by a program are unique. Another important property is "termination": there are no infinite sequences of rewrites. Termination ensures that there is at least one normal form for every term. A system with both these properties is called "convergent" (or "complete"). A finite convergent system is a decision procedure for the (uniform) free word problem of the underlying equational theory: an identity holds (in all models of the axioms) if and only if both its sides can be rewritten to the identical normal form. The first rewriting-based decision procedure (for loops) was given by Evans [22], in 1951. See [19] for a survey of the field.

In 1970, Knuth and Bendix [42] proposed a procedure that attempts to construct a convergent rewrite system from a finite axiomatization of an equational theory. This "completion" procedure must be supplied with an ordering that is used to determine in which direction a derived equation $s \approx t$ is to be oriented into a one-way rule, $s \rightarrow t$ or $t \rightarrow s$. It deduces new equations by a process (involving unification) called "superposition": When the left-hand side of an instance of a rule can be rewritten by another rule, the two possible results of rewriting the left-hand side instance form what is called a "critical pair." (Precise definitions will be given in Section 2.3.)

The following version of completion [16] takes as input a set of equations E_0 and a well-founded ordering $\succ$. It generates all new critical pairs at once; more practical versions (notably the one in [33]) do this incrementally.

> Let E be E_0 and R be the empty set. Then repeat the following six steps as long as equations are left in E; if none remain, terminate successfully:

1. Remove an equation $s \approx t$ (or $t \approx s$) from E such that $s \succ t$. If none exists, terminate with failure.
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. (Optional) Remove all old rules from R whose left-hand side contains an instance of s.
6. Use R to reduce both sides of equations in E to normal forms. Remove any equation whose reduced sides are identical.

The first example on which Knuth and Bendix tried completion was the following axiomatization of free groups:

$$\begin{array}{rcl} e \cdot x & \approx & x \\ x^{-} \cdot x & \approx & e \\ (x \cdot y) \cdot z & \approx & x \cdot (y \cdot z). \end{array}$$

For example, if the associativity axiom is the first equation examined in step 1 of the procedure, then with an ordering that assigns greater weight to the left argument of a product, associativity may be oriented by step 2 into a rule $(x \cdot y) \cdot z \to x \cdot (y \cdot z)$. There being no other rules, step 3 does nothing. In step 4, this rule produces one critical pair, $(x_1 \cdot (x_2 \cdot y)) \cdot z \approx (x_1 \cdot x_2) \cdot (y \cdot z)$, which equates the results of alternative ways of applying associativity to the term $((x_1 \cdot x_2) \cdot y) \cdot z$. Step 6 reduces both sides of the critical pair to the same term, $x_1 \cdot (x_2 \cdot (y \cdot z))$, leaving R with one rule and E empty.

Continuing in this way, and with an appropriate ordering, the following ten rules can be derived:

$$\begin{array}{rclcrcl} e \cdot x & \to & x & \qquad & x \cdot e & \to & x \\ x^{-} \cdot x & \to & e & & x \cdot x^{-} & \to & e \\ (x \cdot y) \cdot z & \to & x \cdot (y \cdot z) & & x^{--} & \to & x \\ e^{-} & \to & e & & (x \cdot y)^{-} & \to & y^{-} \cdot x^{-} \\ x^{-} \cdot (x \cdot y) & \to & y & & x \cdot (x^{-} \cdot y) & \to & y \end{array}$$

With this convergent system, any identity of group theory reduces to a trivial equation.

Since certain (even finitely-based) equational theories (such as combinatory logic) are undecidable, not every equational theory can be represented as a finite convergent rewrite system. Hence, a completion procedure can have any of three outcomes: it may (i) succeed in constructing a finite convergent system, (ii) fail, or (iii) not terminate and instead compute successive approximations R_n of an infinite convergent system R.

An intriguing feature of the completion procedure is that every rewrite rule that is generated can be used to simplify other, already deduced equations (steps 3, 5, and 6). Thus terms can be kept in fully simplified form and redundant equations may be discarded. For example, the rule $x^{--} \cdot e \to x$ may be deleted during completion of group theory, since its left-hand side is simplifiable by the rule $x \cdot e \to x$. While such optimizations account for the efficiency of completion, it may considerably complicate the task of verifying that a procedure is "correct," in the sense that the potentially infinite set of generated, but not discarded, rules forms a convergent system. This notion of correctness is crucial for the use of completion as a (relatively efficient) semi-decision procedure for validity in equational theories, as proposed by Lankford [46] and Huet [33].

Any such correctness proof has to address three problems: first, that the deduction mechanism is general enough for construction of convergent systems (which essentially amounts to proving a Church-Rosser property); second, that simplification and deletion terminate (which can be done by constructing suitable well-founded orderings); third, and most difficult, that simplification and deletion are compatible with deduction (that is, discarded equations are all redundant and any two terms that ever had a common form will have one using persisting, undiscarded rules). In the past, Church-Rosser properties for rewrite systems have been proved by induction with respect to various orderings on terms. Some particularly elegant proofs are given by Huet in [32]. An intricate and rather complicated proof of correctness of a specific completion procedure can be found in [33].

We believe that simplification and deletion mechanisms are best discussed in a proof-theoretic setting, where correctness of completion can be formulated as a proof normalization property. Classical inference systems work from the axioms, "expanding" the set of established formulae by inferring new formulae from old; in this paper, we develop a proof-theoretic formalism for inference systems that also "contract" the set of established formulae via simplification and deletion. We design suitable orderings on *proofs* to establish proof normalization properties (and hence correctness of completion). The advantage of this approach is not only that it is conceptually simpler than previous arguments, but that the correctness results cover a wide variety of different completion procedures. We specify precise and effective conditions which ensure that a control strategy yields a correct completion procedure. As we will see, too much simplification can make a procedure incorrect!

We reformulate the Knuth-Bendix completion method as an equational inference system and demonstrate that it can be viewed as a process of proof simplification, the goal of which is to deduce enough rewrite rules so that any equational proof can be transformed to a "normal-form proof," that is, a proof that two terms (containing variables) are equal by virtue of their both rewriting to the same term. The proof transformations induced by the inference rules can themselves be thought of as (conditional) rewrite rules on equational proofs. Discussing completion at this abstract level allows us to separate the logical, or proof theoretical, component of the method from issues pertaining to strategy and control.

We develop techniques, based on well-founded orderings on proofs, for reasoning about completion and related rewrite methods. To establish the correctness of a completion procedure, for instance, one first has to show that each sequence of proof transformations terminates and produces a minimal proof. We prove this property for arbitrary completion procedures by constructing an appropriate proof ordering. Secondly, in order to guarantee that every minimal proof is in the desired normal-form, a procedure must satisfy a certain "fairness" requirement which ensures that necessary inferences are eventually performed and transformation of non-normal proofs eventually becomes possible. We express fairness in general terms, so that our correctness results apply to a broad class of completion procedures. In particular, generating critical pairs, by overlapping left-hand sides of undeleted rules, satisfies this requirement. Fairness forms the interface between logic and control, in that it specifies under which conditions a control strategy is correct.

We outline our approach by applying it, in Section 3, to the original Knuth-Bendix completion method, and then formulate various refinements, variants, and extensions of standard completion within the same framework. In Section 4, we discuss the concept of "critical pair criterion" for weakening the conditions for fairness to require that only a subset of the critical deductions be performed. In Section 5, we formalize Huet's [32] method for handling equations, such as commutativity, that can not be oriented.

Then, in Section 6, we describe a method, called "ordered completion," that extends standard completion to a complete equational proof method. For example, the following system for Boolean rings defines unique normal forms for ground (variable-free) terms:

$$
\begin{array}{rcl@{\qquad}rcl}
x+0 & \to & x & 0+x & \to & x \\
x+x & \to & 0 & x+(x+y) & \to & y \\
x\cdot 0 & \to & 0 & 0\cdot x & \to & 0 \\
x\cdot 1 & \to & x & 1\cdot x & \to & x \\
x\cdot x & \to & x & x\cdot(x\cdot y) & \to & x\cdot y \\
x+y & \leftrightarrow & y+x & x\cdot y & \leftrightarrow & y\cdot x \\
(x+y)+z & \to & x+(y+z) & (x\cdot y)\cdot z & \to & x\cdot(y\cdot z) \\
y+(x+z) & \leftrightarrow & x+(y+z) & y\cdot(x\cdot z) & \leftrightarrow & x\cdot(y\cdot z) \\
x\cdot(y+z) & \to & (x\cdot y)+(x\cdot z) & (y+z)\cdot x & \to & (y\cdot x)+(z\cdot x)
\end{array}
$$

The rules with double-headed arrows are used only in the direction that results in a decrease in a specific ordering $\succ$ (in this case, the lexicographic path ordering with a total ordering of the operators and constants; see [17]). Ordered completion, starting with this system and (the Skolemized negation of) a theorem in first-order predicate calculus with equality (taking $+$ to be exclusive-or and $\cdot$ to be conjunction), can be used (by Herbrand's Theorem) as the basis of a refutationally complete prover (cf. [30]).

Finally, in Section 7, we present, within the same proof-transformation framework, a method for proving inductive theorems, due originally to Musser [49], based on the concept of "proof by consistency." With this method, it is easy to prove automatically from the definition of multiplication given at the outset that $0 \times x \approx 0$. We end with a brief conclusion.

2. Equational Proofs

Theorem provers can often be viewed profitably as proof normalization procedures. Such a proof procedure is said to be "correct" if enough consequences can be deduced so that any arbitrary proof can be transformed to a normal-form proof. We study rewrite-based equational reasoning methods from this point of view: the set of theorems corresponds to some congruence relation on terms; proofs are finite sequences of equational replacements; and normal-form proofs are proofs in which equations are used in a specified direction, as one-way rewrite rules.

Accordingly, we will be concerned with binary relations on (first-order) *terms* (over some set of operator symbols $\mathcal{F}$ and some set of variables $\mathcal{V}$), and on (annotated) finite sequences of terms (representing proofs). We assume that there is at least one constant; thus, the set of *ground* terms, that is, terms containing no variables, is non-empty. For example, if $+$ is a binary operator, $-$ is a unary operator, and 0 and 1 are constants, then $(-x+y)+0$ is non-ground, while $1+0$ is ground.

The symbols s, t, and u will be used to denote terms. In our examples, we will use x, y, and z for variables. The expression $t|_p$ denotes the subterm of t at position p. Positions may, for instance, be represented in Dewey decimal notation. So, if $p' = pq'$, then $t|_{p'}$ is a subterm of $t|_p$. We use Λ for the top-most position ($t|_\Lambda = t$). A subterm of a term t is called *proper* if it is distinct from t. A term t with a distinguished position p will be called a *context*, denoted $t[\cdot]_p$. By $t[s]_p$ we indicate the term obtained from that context by placing term s at position p, or—equivalently—the result of replacing the subterm $t|_p$ of t by s.

By $t\sigma$ we denote the result of applying the substitution σ to the term t, and call $t\sigma$ an *instance* of t. An instance s of t is *proper* if t is not an instance of s. For example, if σ is the substitution $\{x \mapsto y\}$, then $f(x, y)\sigma = f(y, y)$ is a proper instance of $f(y, x)$. Two terms are said to be *literally similar* if they are instances of each other. For instance, $f(x, y)$, $f(y, x)$, and $f(y, z)$ are all literally similar.

2.1. Rewrite Rules

A binary relation $\rightarrow$ is called *terminating* if there exists no infinite sequence $t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow \cdots$. The symbols $\rightarrow^+$, $\rightarrow^*$ and $\leftrightarrow$ denote the transitive, transitive-reflexive, and symmetric closure of $\rightarrow$, respectively. The inverse of $\rightarrow$ is denoted by $\leftarrow$. A binary relation $\rightarrow$ on terms is called a *rewrite relation* if $s \rightarrow t$ implies $u[s\sigma]_p \rightarrow u[t\sigma]_p$, for all contexts $u[\cdot]_p$, terms s and t, and substitutions σ.

An *equation* is an ordered pair of terms, written $s \approx t$; when we wish to refer to either of $s \approx t$ and $t \approx s$, we will write $s \simeq t$. If E is a set of equations, we use E^{-1} for the set $\{t \approx s \mid s \approx t \in E\}$. Given a set of equations E, we denote by $\rightarrow_E$ the smallest rewrite relation containing the set of equations E. That is, $v \rightarrow_E w$ if and only if v is $u[s\sigma]_p$ and w is $u[t\sigma]_p$, for some context $u[\cdot]_p$, substitution σ, and equation $s \approx t$ in E. By $\leftrightarrow_E$ we denote the symmetric closure of $\rightarrow_E$. The relation $\leftrightarrow_E^*$ will be referred to as the *equational theory* of E; it is the reflexive-symmetric-transitive closure of $\rightarrow$. For example, if E is the associativity axiom, then $(x_1 \cdot (x_2 \cdot y)) \cdot z \leftrightarrow_E^* (x_1 \cdot x_2) \cdot (y \cdot z)$. In some applications it will be useful to attach labels to equations. We may write $s \approx_n t$ to denote such labelled equations, but usually we use different relation symbols for different types of equations. We identify literally similar equations.

A set of equations R will be called a *rewrite system* if we are primarily interested in the asymmetric rewrite relation $\rightarrow_R$ rather than the equational theory $\leftrightarrow_R^*$. In that case, the equations in R are also called *rewrite rules* and are written $s \rightarrow t$. We also say that u *rewrites* to v if $u \rightarrow_R v$. A term that can not be rewritten is said to be *irreducible* by R. A *normal form* of t (with respect to R) is any irreducible term u for which $t \rightarrow_R^* u$; in other words, repeatedly rewriting t until no longer possible, gives a normal form of t. We write $t \rightarrow_R^! u$ if u is a normal form of t.

A rewrite system R is said to be *Church-Rosser* (or *confluent*) if, for all terms s and t with $s \leftrightarrow_R^* t$, there exists a term v, such that $s \rightarrow_R^* v$ and $t \rightarrow_R^* v$. A rewrite system R is said to *terminate* if the relation $\rightarrow_R$ is terminating, that is, if $\rightarrow_R^+$ is a well-founded (strict) partial ordering. Terminating Church-Rosser systems are called *convergent* (or *complete*) and define unique normal forms for all terms. A terminating rewrite system R will be called *ground convergent* if, for all *ground* terms s and t such that $s \leftrightarrow_R^* t$, we have $s \rightarrow_R^* v \leftarrow_R^* t$.

By a *rewrite ordering* we mean an irreflexive and transitive rewrite relation on terms. Terminating orderings are *well-founded*; well-founded rewrite orderings are called *reduction orderings*. Evidently, a rewrite relation $\rightarrow_R$ terminates if and only if R is contained in some reduction ordering. In an untyped terminating system all variables appearing on the right-hand side of a rule must also appear on the corresponding left-hand side. Lexicographic and multiset orderings are particularly useful in proving termination of rewrite relations:

- If $\succ_1$ and $\succ_2$ are orderings, then their *lexicographic combination* is defined on pairs by: $(s, t) \succ_{lex} (s', t')$ if either $s \succ_1 s'$ or $s = s'$ and $t \succ_2 t'$. The lexicographic combination of more than two orderings can be defined in a similar way. A lexicographic ordering is well-founded if and only if its component orderings are well-founded.

- A *multiset* over a set S is a mapping M from S to the natural numbers. Informally, $M(x)$ specifies the number of occurrences of x in M. A multiset M is *finite* if $M(x) > 0$ for a finite number of $x \in S$. For simplicity, we use a set-like notation to describe (finite) multisets. For example, $\{x, x, x\}$ denotes the multiset M for which $M(x) = 3$ and $M(y) = 0$, for $y \neq x$. Any ordering $\succ$ on a set S can be extended to an ordering $\succ_{mul}$ on (finite) multisets over S as follows: $M \succ_{mul} N$ if and only if (i) $M \neq N$ and (ii) whenever $N(x) > M(x)$ then $M(y) > N(y)$, for some y such that $y \succ x$. In other words, according to the multiset ordering, any element of a multiset can be replaced by any finite number of elements that are smaller with respect to $\succ$. The multiset ordering $\succ_{mul}$ is well-founded (on finite multisets) if and only if the ordering $\succ$ is well-founded [20].

See [17] for a survey of termination.

2.2. Rewrite Proofs

A *(equational) proof step* is an expression $s \leftrightarrow^p_e t$, where s and t are terms, e is an equation $u \approx v$, and p is a position in s, such that $s = w[u\sigma]_p$ and $t = w[v\sigma]_p$, for some context $w[\cdot]_p$ and substitution σ. We say that $s \leftrightarrow^p_{u\approx v} t$ is a *proof step in* E if either $u \approx v$ or $v \approx u$ is an equation in E. We sometimes omit the position from a proof step and just write $s \leftrightarrow_{u\approx v} t$. We say that $s \leftrightarrow_e t$ for an application of equation e. By definition, there is a proof step $s \leftrightarrow^p_{u\approx v} t$ in E if and only if $s \leftrightarrow_E t$.

A *(equational) proof* (of $t_0 \approx t_n$) is any finite sequence of proof steps

$$t_0 \xleftrightarrow[e_1]{p_1} t_1,\ t_1 \xleftrightarrow[e_2]{p_2} t_2,\ \ldots,\ t_{n-1} \xleftrightarrow[e_n]{p_n} t_n,$$

usually written in abbreviated form as

$$t_0 \xleftrightarrow[e_1]{p_1} t_1 \xleftrightarrow[e_2]{p_2} t_2 \xleftrightarrow[e_3]{p_3} \cdots t_{n-1} \xleftrightarrow[e_n]{p_n} t_n.$$

The empty sequence, abbreviated as a single term t, serves as a proof of an identity $t \approx t$.

We will use the letters P and Q to denote proofs. We say that P is a *proof in* E if each proof step of P is in E. Note that an equation $s \approx t$ is provable in E if and only if $s \leftrightarrow^*_E t$. Equational replacement is complete, that is, $E \models s \approx t$, for a set of equations E, if and only if $s \leftrightarrow^*_E t$.

For simplicity, we often denote by $s \rightarrow_E t$ any proof step $s \leftrightarrow^p_{u\approx v} t$, where $u \approx v$ is an equation in E; if $v \approx u$ is the equation in E, we write instead $s \leftarrow_E t$; by $s \leftrightarrow_E t$ we denote a proof step $s \rightarrow_E t$ or $s \leftarrow_E t$ which uses one equation in *either* direction. We also write $s \leftrightarrow^*_E t$ to denote arbitrary proofs in E. If P is a proof

$$t_0 \xleftrightarrow[u_1\approx v_1]{p_1} t_1 \xleftrightarrow[u_2\approx v_2]{p_2} t_2 \leftrightarrow \cdots \leftrightarrow t_{n-1} \xleftrightarrow[u_n\approx v_n]{p_n} t_n,$$

we denote by P^{-1} the inverted proof:

$$t_n \xleftrightarrow[v_n\approx u_n]{p_n} t_{n-1} \leftrightarrow \cdots \leftrightarrow t_2 \xleftrightarrow[v_2\approx u_2]{p_2} t_1 \xleftrightarrow[v_1\approx u_1]{p_1} t_0;$$

by $P\sigma$ we denote the instantiated proof:

$$t_0\sigma \xleftrightarrow[u_1\approx v_1]{p_1} t_1\sigma \xleftrightarrow[u_2\approx v_2]{p_2} t_2\sigma \leftrightarrow \cdots \leftrightarrow t_{n-1}\sigma \xleftrightarrow[u_n\approx v_n]{p_n} t_n\sigma;$$

and by $u[P]_q$, where q is a position in u, the same proof applied to the subterm of u at q:

$$u[t_0]_q \xleftrightarrow[u_1\approx v_1]{qp_1} u[t_1]_q \xleftrightarrow[u_2\approx v_2]{qp_2} u[t_2]_q \leftrightarrow \cdots \leftrightarrow u[t_{n-1}]_q \xleftrightarrow[u_n\approx v_n]{qp_n} u[t_n]_q.$$

If P is a proof (in E), then P^{-1}, $P\sigma$, and $u[P]_p$ are also proofs (in E). By a *subproof* of P we mean any proof

$$t_i \xleftrightarrow[u_{i+1}\approx v_{i+1}]{p_{i+1}} t_{i+1} \leftrightarrow \cdots \leftrightarrow t_{j-1} \xleftrightarrow[u_j\approx v_j]{p_j} t_j,$$

where $0 \le i \le j \le n$. We write $P[Q]$ to indicate that P contains Q as a subproof, and denote by $P[Q']$ the proof obtained from P by replacing Q by Q'.

Let $E \cup R$ be a given set of equations, some of which (those in R) are designated rules and are written as $s \to t$, while non-rules (those in E) are written $s \approx t$. A proof step $s \leftrightarrow_E t$ is called an *equality step*; a step $s \to_R t$ or $s \leftarrow_R t$ is called a *rewrite step*; a (sub)proof $s \leftarrow_R u \to_R t$ is a *peak*. We usually abbreviate a proof of the form $t_0 \to_R \cdots \to_R t_n$ by $t_0 \to_R^* t_n$, and call a proof $t_0 \to_R^* t_k \leftarrow_R^* t_n$ a *rewrite proof*.

A *proof transformation relation* is any binary relation $\Rightarrow$ on proofs such that (i) $P \Rightarrow Q$ implies $u[P\sigma]_q \Rightarrow u[Q\sigma]_q$, for all proofs P and Q, substitutions σ, terms u, and positions q in u; (ii) $Q \Rightarrow Q'$ implies $P[Q] \Rightarrow P[Q']$, for all proofs $P[Q]$, Q, and Q'; and (iii) $P \Rightarrow Q$ only if P and Q are proofs of the same equation. A *proof (reduction) ordering* $\gg$ is a well-founded ordering on proofs that satisfies (i) and (ii). Terminating proof transformation relations will be called *proof normalization relations*; any such is contained in some proof ordering.

We will be looking in subsequent sections at proof normalization relations that take arbitrary equational proofs to rewrite proofs.

2.3. Critical Pairs

When there is no rewrite proof for a given peak, it will be necessary to deduce new equations to facilitate construction of a rewrite proof. Certain equational consequences called "critical pairs" are of central importance in this regard.

Let $s \approx t$ and $u \approx v$ be equations in E (they may be the same equation) with no variables in common (the variables of one equation are renamed if necessary) and suppose that, for some position p, the term $s|_p$ is not a variable and is unifiable with u, σ being the most general unifier. Then the *superposition* of $u \approx v$ on $s \approx t$ at position p determines a *critical pair* $t\sigma \approx s\sigma[v\sigma]_p$. The proof $t\sigma \leftarrow^p_{s\approx t} s\sigma \to^p_{u\approx v} s\sigma[v\sigma]_p$ is called a *critical overlap*; the term $s\sigma$, the *overlapped term*; the position p, the *critical pair position*. For example, the rule $x \cdot x^- \to e$ can be superposed on $(x \cdot y)^- \to y^- \cdot x^-$ to yield a critical pair $x^{--} \cdot x^- \approx e^-$. The corresponding critical overlap is $x^{--} \cdot x^- \leftarrow_R (x \cdot x^-)^- \to_R e^-$. By $CP(E)$ we denote the set of all critical pairs between equations in E, formed from overlapping left-hand sides. The set of critical pairs $CP(E)$ is finite whenever E is finite.

LEMMA 2.1 (Critical Pair Lemma [42]). *Let E be a set of equations $\{e, e'\}$. If P is a proof $s \leftarrow_e u \to_{e'} t$, then either there exists a rewrite proof $s \to_E^* v \leftarrow_E^* t$ or else $P = u[Q\tau]_p$, for some context $u[\cdot]_p$, substitution τ, and critical overlap Q (between the equations in E), and $s \leftrightarrow_{CP(E)} t$.*

SKETCH OF PROOF. If the two positions p and p' at which the equational steps take place in a peak $s \leftarrow_e^p u \rightarrow_{e'}^{p'} t$ are disjoint, the peak is a *non-overlap.* However, if p is above p', that is, if p is a prefix of $p' = pq'$, and, furthermore, q' is a position in u at which a variable occurs or q' is a position not in u above which a variable occurs, then the peak is called *nested.* In the case of a non-overlap or nested peak, there exists a term v, such that $s \rightarrow_E^* v \leftarrow_E^* t$. Any other overlap is critical, and can be written as $u[Q\tau]_p$, where Q is the critical overlap (assuming p is above p'). In this case, $s \leftrightarrow_c^p t$, where c is a critical pair of e and e'. For details, see [42, 32]. □

3. Equational Inference

Knuth and Bendix [42] presented a procedure that attempts to construct a convergent rewrite system for a given equational theory. This completion procedure has been reformulated as an equational inference system in which new equations and new rewrite rules are derived, while old ones are simplified and/or deleted. The application of completion to a set of equations is viewed as a process of proof simplification, the goal of which is the derivation of rewrite proofs. A detailed exposition of this approach is contained in the first author's dissertation [2]; see also [3].

What distinguishes completion from ordinary logical inference is the incorporation of rules for removing redundant items from the set of inferred equations. In this context, an inference rule is a binary relation on sets of equations. A rule that adds a consequence to the set is called an *expansion* rule; one that deletes a redundant equation is called a *contraction* rule.

In its most general form, expansion for equations is expressed in the following inference rule:

$$\text{EXPANSION:} \qquad \frac{E}{E \cup \{s \approx t\}} \qquad \text{if } s \leftrightarrow_E^* t$$

Were a prover omniscient, then one step with this inference rule would suffice to prove any theorem. Realistic provers limit expansion to smaller steps, requiring a sequence of expansions to prove most theorems. We must ensure, however, that the expansion rules used are powerful enough to prove any theorem.

Before formulating contraction rules, we need a notion of redundancy. To this end, we use a proof transformation relation $\Rightarrow$ that captures what it means for one proof to be "better" than another. In its most general form, contraction is expressed in the following inference rule:

$$\text{CONTRACTION:} \qquad \frac{E \cup \{s \approx t\}}{E} \qquad \text{if } s \leftrightarrow_{s\approx t}^{\Lambda} t \Rightarrow^* s \leftrightarrow_E^* t$$

Since we only delete an equation when any proof step using it can be replaced by a proof that is no worse, vis-à-vis the proof relation $\Rightarrow$, contraction preserves completeness of the inference system, and we say the system is "sound."

The goal of an inference sequence is to produce the formulae necessary for a normal-form proof either of a given theorem or of all theorems in a given theory. In our case, rewrite proofs are in normal form, and we need to design the proof normalization relation $\Rightarrow$ so that non-rewrite proofs can be transformed into rewrite proofs. We also insist that $\Rightarrow$ be terminating, so that (among other considerations) contracting a finite set of equations takes only a finite number of steps.

For example, if equations are used to rewrite only in a direction that decreases the term in some reduction ordering, then we could use a transformation of the form

$$s \underset{E}{\longleftarrow} u \underset{E}{\longrightarrow} t \quad \Rightarrow \quad s \underset{E}{\overset{*}{\longrightarrow}} v \underset{E}{\overset{*}{\longleftarrow}} t$$

which replaces peaks by "valleys."

We write $E \vdash E'$ to indicate that the set of equations E' can be obtained from E by one application of an inference rule. A (possibly infinite) sequence $E_0 \vdash E_1 \vdash E_2 \vdash \cdots$ is called a *derivation* from E_0. The (lower) *limit* $\bigcup_i \bigcap_{j \geq i} E_j$ of a sequence of equations $E_0, E_1, \ldots$, contains all equations that persist from some point in the sequence on, and is denoted by E_∞.

The following is straightforward:

LEMMA 3.1 (Reflection). *If P is a proof in E and $E \vdash E'$ is an application of an expansion or contraction inference rule, then there exists a proof Q in E' such that $P \Rightarrow^* Q$.*

Since $\Rightarrow$ is terminating, we also have:

COROLLARY 3.2 (Persistence). *Let $E_0 \vdash E_1 \vdash E_2 \vdash \cdots$ be a (finite or infinite) derivation. If P is a proof in $\bigcup_i E_i$, then there exists a proof Q in E_∞ such that $P \Rightarrow^* Q$.*

In particular, every identity in E_0 eventually has a persisting proof.

PROOF. By reflection and the properties of proof normalization relations, for every derivation $E_0 \vdash E_1 \vdash \cdots$ and proof or subproof P_i in E_i, there exists a corresponding proof transformation sequence $P_i \Rightarrow P_{i+1} \Rightarrow \cdots$, where P_j is a proof in E_j, for all $j \geq i$. Since the relation $\Rightarrow$ is terminating, we have $P_n = P_{n+1} = \cdots$, for some $n \geq i$, so that P_n is a proof in E_∞. □

To show that an inference system is complete and all theorems are provable, we need to establish a lemma of the form:

LEMMA 3.3 (Existence). *If a proof P in E is not in normal form, then there exists an expansion inference $E \vdash E'$ and a proof Q in E' such that $P \Rightarrow^+ Q$.*

If, for example, we would require that for a proof of an equation e to be deemed "normal form" it be a direct application of the axiom e, then the Existence Lemma would trivially hold, since one expansion is all that is needed to generate e from E.

It follows from Existence and the well-foundedness of $\Rightarrow^+$ that for any equation provable in E_0, there is some derivation $E_0 \vdash E_1 \vdash E_2 \vdash \cdots$ giving a normal-form proof of the same equation in the limit set E_∞. Of course, not all derivations give normal-form proofs, since a derivation could completely ignore some equations. To characterize those derivations that do enough expansions for normal-form proofs to eventually become available, we introduce a notion of "fairness":

DEFINITION 3.4 (Fairness). A derivation $E_0 \vdash E_1 \vdash E_2 \vdash \cdots$ is *fair* (with respect to a proof reduction relation $\Rightarrow$ and a set of expansion rules $\mathcal{E}$) if for any proof P in E_∞ that is not in normal form, and for which there is an expansion inference $E_\infty \vdash E'_\infty$ (in $\mathcal{E}$) and a proof P' in E'_∞ such that $P \Rightarrow^+ P'$, there also exists a proof Q in $\bigcup_i E_i$ such that $P \Rightarrow^+ Q$.

This does not mean that fair derivations contain only expansions; rather, it requires that derivations do not forever ignore those expansions that are needed to derive normal-form proofs.

Combining fairness with existence and persistence, we have:

THEOREM 3.5 (Normalization). *Let $E_0 \vdash E_1 \vdash E_2 \vdash \cdots$ be a fair derivation. If an equation is provable in E_0, then it has a normal-form proof in E_∞.*

In the remainder of this paper, we look at various normal forms for proofs, and at restrictions of expansion and combinations of specific expansions and contractions that guarantee that those proofs can be found. To establish a Reflection Lemma, we will need only show that the inference rules are special cases of expansion and contraction. To show persistence, we will need to prove that the transformation relation is terminating. For each specific notion of normal form and transformation relation, we will have to prove an Existence Lemma. Finally, we will endeavor to give sufficient, practical conditions for fairness.

3.1. Standard Completion

Let $\succ$ be a reduction ordering on terms. The inference system $\mathcal{C}$, which was introduced in [2, 3] and which we call *standard completion* because it is inspired by Knuth and Bendix [42], contains six rules operating on mixed sets $E \cup R$ of rules R and equations E, where R is contained in $\succ$. Normal-form proofs are rewrite proofs that use rules in R only. Hence, a non-normal proof either uses an equation from E or has a peak formed from applications of rules in R.

The first inference rule in $\mathcal{C}$ is used to expand the set of equations with critical pairs obtained by rewriting an instance of a left-hand side of a rule in two different ways:

$$\text{DEDUCTION:} \qquad \frac{E \cup R}{E \cup \{s \approx t\} \cup R} \qquad \text{if } s \approx t \in CP(R)$$

Such new equations serve to eliminate critical peaks that do not have an alternative rewrite proof. For example, the two rules $e \cdot x \to x$ and $(x \cdot y) \cdot z \to x \cdot (y \cdot z)$ define a peak $y \cdot z \leftarrow_R (e \cdot y) \cdot z \to_R e \cdot (y \cdot z)$, so that the critical pair $y \cdot z \approx e \cdot (y \cdot z)$ can be deduced. Deduction is reflected on the proof level by transformations of the form

$$s \underset{R}{\longleftarrow} u \underset{R}{\longrightarrow} t \quad \Rightarrow \quad s \underset{CP(R)}{\longleftrightarrow} t$$

To show that that this expansion rule is sufficient, we add a proof transformation for non-critical peaks:

$$s \underset{R}{\longleftarrow} u \underset{R}{\longrightarrow} t \quad \Rightarrow \quad s \overset{*}{\underset{R}{\longrightarrow}} v \overset{*}{\underset{R}{\longleftarrow}} t$$

and apply the Critical Pair Lemma, which asserts that all peaks can be replaced with a valley or with a critical-pair step. This is not quite enough to establish the requisite Existence Lemma, however, since we have no way of eliminating proof steps involving E.

Standard completion makes a strong distinction between equations and rules. Equations may be eliminated from non-normal proofs by turning them into one-way rewrite rules. Equations are oriented according to the given reduction ordering $\succ$ to ensure that rewrite systems derived by completion are terminating:

$$\text{ORIENTATION:} \qquad \frac{E \cup \{s \simeq t\} \cup R}{E \cup R \cup \{s \to t\}} \qquad \text{if } s \succ t$$

(Recall that $s \simeq t$ denotes either $s \approx t$ or $t \approx s$.) For instance, the equation $e^- \approx e$ can be oriented (with respect to any ordering that includes the proper subterm relation) into a rule $e^- \to e$. Orientation should be viewed as an expansion followed by a contraction: the rule $s \to t$ is added, making the equation $s \simeq t$ redundant. Rules are preferred over equations, since they are used in rewrite proofs:

$$s \underset{s \approx t}{\longleftrightarrow} t \quad \Rightarrow \quad s \longrightarrow_{s \to t} t \qquad \text{where } s \succ t$$

These two inference rules provide an Existence Lemma for some, but not all, proofs, since not all equations can be oriented. For example, there is no way to make a rule out of commutativity, $x + y \approx y + x$, and preserve termination. We say that a derivation $E_0 \vdash E_1 \vdash E_2 \vdash \cdots$ *fails* if some equation persists, that is if $E_\infty \neq \emptyset$. Thus, the Normalization Theorem only holds for fair, unfailing derivations. Two ways of circumventing such failures will be described in Sections 5 and 6.

A simple contraction rule deletes trivial equations:

$$\textsc{Deletion:} \qquad \frac{E \cup \{s \approx s\} \cup R}{E \cup R}$$

For this to fit the contraction scheme, we need a proof transformation of the form:

$$s \overset{\Lambda}{\underset{s \approx s}{\longleftrightarrow}} s \quad \Rightarrow \quad s$$

Since proof transformation relations are closed under instantiation and application in context, this means that any superfluous step $u \leftrightarrow^p_{s \approx s} u$ can be deleted from a proof.

Another contraction rule, uses rules to simplify deduced equations:

$$\textsc{Simplification:} \qquad \frac{E \cup \{s \simeq t\} \cup R}{E \cup \{u \approx t\} \cup R} \qquad \text{if } s \to_R u$$

This inference is also a combination of expansion (adding $u \approx t$), followed by contraction (removal of the now redundant $s \simeq t$). For this contraction to be valid, we need to be able to replace any proof step using the removed equation by a simpler proof:

$$s \underset{s \approx t}{\longleftrightarrow} t \quad \Rightarrow \quad s \longrightarrow_{r} u \longleftrightarrow_{u \approx t} t$$

where r is any rewrite rule. For efficiency reasons, implementations of completion procedures usually include further simplification mechanisms which can be described by the following two inference rules:

$$\textsc{Composition:} \qquad \frac{E \cup R \cup \{s \to t\}}{E \cup R \cup \{s \to u\}} \qquad \text{if } t \to_R u$$

$$\textsc{Collapse:} \qquad \frac{E \cup R \cup \{s \to t\}}{E \cup \{u \approx t\} \cup R} \qquad \begin{array}{l}\text{if there is a proof } s \to^p_{v \to w} u \\ \text{in } R\text{, where } s \rhd v\end{array}$$

The symbol $\rhd$ denotes the *encompassment ordering* (called "containment" in [33]): $s \rhd v$ *if some subterm of* s *is an instance of* v*, but not vice versa.* Composition allows simplification of right-hand sides of rewrite rules; collapsing simplifies left-hand sides. While composition produces another rewrite rule, the equation obtained by collapsing a rule need not necessarily be orientable.

The composition and collapse rules are reflected by proof transformations of the form:

$$s \xrightarrow[s \to t]{} t \;\Rightarrow\; s \rightarrow_{s \to u} u \leftarrow_{r} t$$

$$s \rightarrow^{\Lambda}_{s \to t} t \;\Rightarrow\; s \rightarrow^{q}_{v \to w} u \longleftrightarrow^{\Lambda}_{u \approx t} t$$

where $s \rhd v$ and r is any rewrite rule.

As before, we write $E \vdash E'$ to indicate that the set of equations E' can be inferred from E by an application of one of the inference rules in $\mathcal{C}$. Completion is *sound* in that the congruence relations $\leftrightarrow^*_{E \cup R}$ and $\leftrightarrow^*_{E' \cup R'}$ are the same, whenever $E \cup R \vdash E' \cup R'$. Furthermore, if $E \cup R \vdash E' \cup R'$ and the reduction ordering $\succ$ contains R, then $\succ$ also contains R'. Consequently, the system R_∞ is terminating for any derivation for which the initial rewrite system R_0 (which is usually empty) is contained in the reduction ordering $\succ$. The above inference rules never decrease the strength of rewriting, that is, if $E \cup R \vdash E' \cup R'$, then any term t that is reducible by R is also reducible by R'.

By a *(standard) completion procedure*, we mean a program that accepts as input a set of equations $E_0 \cup R_0$ and a reduction ordering $\succ$ containing R_0, and uses some strategy to apply the inference rules of $\mathcal{C}$ to generate a derivation from $E_0 \cup R_0$. We say that a completion procedure *succeeds* for a given input if no unoriented equation persists forever ($E_\infty = \emptyset$) and the set of persisting rules (R_∞) is convergent. A procedure *fails* for an input if $E_\infty \neq \emptyset$. Similarly, we distinguish between failing and non-failing derivations depending on whether or not E_∞ is empty. A completion procedure is called *correct* if R_∞ is convergent, whenever $E_\infty = \emptyset$. In other words, correctness asserts that all non-failing derivations result in a convergent system.

The procedure given in Section 1 can be viewed as an exemplar of standard completion. Step 2 represents an application of orientation; step 3, repeated application of composition; step 4, repeated deduction; step 6, repeated simplification and deletion. Step 4, in combination with step 5, implicitly uses collapse, for whenever a rule $s \to t$ can be collapsed to $u \approx t$, then $u \approx t$ is a critical pair in $CP(R)$. In the above procedure, the equation $u \approx t$ is first deduced in step 4, but may be deleted in step 5. Fairness of the procedure, that is, fairness of all its derivations, depends on the order in which, in step 1, equations are removed from E.

Observe that equations in E are kept in fully simplified form (step 6). Consequently, whenever an equation $s \approx t$ is selected in step 1, both s and t are irreducible with respect to the current rewrite system R. This guarantees that R will never contain two rules with literally similar left-hand sides and also that the restrictions imposed on collapse inferences are satisfied.

3.2. Proof Transformation

By a *(proof) transformation system* we mean a binary relation $\mathcal{R}$ on equational proofs. Elements of $\mathcal{R}$ are called proof transformation rules and are written $P \Rightarrow Q$. As far as proof transformation is concerned we do not distinguish a proof P from the inverse proof P^{-1}. We associate with $\mathcal{R}$, a proof transformation relation $\Rightarrow$ which is the smallest proof transformation relation that contains $\mathcal{R}$ and for which $P^{-1} \Rightarrow Q^{-1}$, whenever $P \Rightarrow Q$. We also say that a proof P can be *transformed* to Q, whenever $P \Rightarrow^* Q$.

Let $\mathcal{R}$ denote the set of all the above proof transformation rules, and let $\Rightarrow$ be the corresponding proof transformation relation. Since each inference rule is a combination of basic expansions and contractions, the Reflection Lemma asserts that every application of an inference rule in $\mathcal{C}$ is reflected on the proof level by transformations in $\mathcal{R}$.

We take a "two-dimensional" view of proofs (larger terms above smaller ones), and illustrate the proof normalization process with the axioms of group theory:

$$\begin{array}{rcl} e \cdot x & \approx & x \\ x^- \cdot x & \approx & e \\ (x \cdot y) \cdot z & \approx & x \cdot (y \cdot z). \end{array}$$

The equation $z^{--} \cdot e \approx z$ is provable:

$$\begin{array}{rcl} z^{--} \cdot e & \leftrightarrow_E & z^{--} \cdot (z^- \cdot z) \\ & \leftrightarrow_E & (z^{--} \cdot z^-) \cdot z \\ & \leftrightarrow_E & e \cdot z \\ & \leftrightarrow_E & z. \end{array}$$

Let $\succ$ be the recursive path ordering (see [17]) corresponding to a precedence ordering in which $\cdot$ is greater than $^-$, which is greater than e. By repeated application of orientation we obtain the following rewrite system R_1 from the above equations:

$$\begin{array}{rcl} e \cdot x & \rightarrow & x \\ x^- \cdot x & \rightarrow & e \\ (x \cdot y) \cdot z & \rightarrow & x \cdot (y \cdot z). \end{array}$$

These inference steps are reflected by a sequence of proof transformations $P_0 \Rightarrow^+ P_1$, where P_0 is the above proof and P_1 is

$$\begin{array}{ccccccccc} & & & & (z^{--} \cdot z^-) \cdot z & & & & \\ & & & \swarrow_{R_1} & & \searrow_{R_1} & & & \\ & & z^{--} \cdot (z^- \cdot z) & & & & e \cdot z & & \\ & \swarrow_{R_1} & & & & & & \searrow_{R_1} & \\ z^{--} \cdot e & & & & & & & & z. \end{array}$$

The middle two steps of this proof form a peak which is an instance of a peak

$$x^- \cdot (x \cdot y) \leftarrow_{R_1} (x^- \cdot x) \cdot y \rightarrow_{R_1} e \cdot y$$

between the second and third rule of R_1. We can deduce the critical pair $x^- \cdot (x \cdot y) \approx e \cdot y$ and obtain a new proof P_2:

$$\begin{array}{ccccccc} & & z^{--} \cdot (z^- \cdot z) & \leftrightarrow_{E_2} & e \cdot z & & \\ & \swarrow_{R_2} & & & & \searrow_{R_2} & \\ z^{--} \cdot e & & & & & & z, \end{array}$$

where $R_2 = R_1$ and E_2 contains the critical pair. Again, $P_1 \Rightarrow^+ P_2$. The equation $x^- \cdot (x \cdot y) \approx e \cdot y$ can be simplified to $x^- \cdot (x \cdot y) \approx y$, replacing one step of P_2 with two steps in P_3:

$$\begin{array}{ccccccccc} & & & & & & e \cdot z & & \\ & & & & & \swarrow_{R_3} & & \searrow_{R_3} & \\ & & z^{--} \cdot (z^- \cdot z) & \leftrightarrow_{E_3} & z & & & & z \\ & \swarrow_{R_3} & & & & & & & \\ z^{--} \cdot e & & & & & & & & \end{array}$$

Orienting the new equation and moving it from E_3 to R_4, induces a transformation $P_3 \Rightarrow^+ P_4$, where P_4 is

$$\begin{array}{ccccccccc} & & z^{--} \cdot (z^{-} \cdot z) & & & & e \cdot z & & \\ & \swarrow_{R_4} & & \searrow_{R_4} & & \swarrow_{R_4} & & \searrow_{R_4} & \\ z^{--} \cdot e & & & & z & & & & z. \end{array}$$

The trivial peak $z \leftarrow_{R_4} e \cdot z \rightarrow_{R_4} z$ can be transformed away, so that we end up with a proof

$$\begin{array}{ccccc} & & z^{--} \cdot (z^{-} \cdot z) & & \\ & \swarrow_{R_4} & & \searrow_{R_4} & \\ z^{--} \cdot e & & & & z. \end{array}$$

We deduce the critical pair $x^{--} \cdot e \approx x$ and turn that into a rule $x^{--} \cdot e \rightarrow x$. The initial equation $z^{--} \cdot e \approx z$ can now be deduced immediately, via a one-step rewrite proof. However, the set of derived rules

$$\begin{array}{rcl} e \cdot x & \rightarrow & x \\ x^{-} \cdot x & \rightarrow & e \\ (x \cdot y) \cdot z & \rightarrow & x \cdot (y \cdot z) \\ x^{-} \cdot (x \cdot y) & \rightarrow & y \\ x^{--} \cdot e & \rightarrow & x \end{array}$$

is not yet convergent. For instance, the equation $e^{-} \cdot y \approx y$ is provable, but not by a rewrite proof. If we continue the completion process, we eventually obtain the ten-rule convergent system shown in Section 1.

3.3. Proof Normalization

The proof transformations reflecting completion are simplifying, so that any arbitrary sequence of proof transformations (applied to a finite proof) has to be finite:

LEMMA 3.6. *The proof transformation relation $\Rightarrow$ is terminating.*

SKETCH OF PROOF. We define a measure of the complexity of an equational proof by assigning a certain cost to each single proof step. The cost of an equational proof step $s \rightarrow^{p}_{u \approx v} t$ or $s \leftarrow^{p}_{u \approx v} t$ is the triple $(\{s, t\}, u, t)$. The cost of a rewrite proof step $s \rightarrow^{p}_{u \rightarrow v} t$ or $s \leftarrow^{p}_{u \rightarrow v} t$ is $(\{s\}, u, t)$. The complexity of a proof is the multiset of the costs of its proof steps.

Proof steps are compared with each other according to their cost, using the lexicographic combination of the multiset extension $\succ_{mul}$ of the reduction ordering $\succ$ in the first component, the specialization ordering $\rhd$ in the second component, and the reduction ordering $\succ$ in the last component. Proofs are compared according to their complexity, using the extension of the above lexicographic ordering to multisets. This ordering, which is a proof reduction ordering, contains the proof transformation relation $\Rightarrow$. For details of a similar proof, see [2, 3]. $\square$

It remains to be shown that every identity eventually has a (persisting) rewrite proof. If no unoriented equations persist, that is, if the procedure does not fail, this is the same as saying that the result R_∞ is Church-Rosser. For this, it suffices to show that for each proof P_i in $E_i \cup R_i$ there is a suitable proof transformation sequence for which the minimal proof P_n is a rewrite proof in R_∞. In short, we interpret completion as a process of proof normalization, the goal of which is the derivation of rewrite proofs. Computation of critical pairs is a sufficient condition for fairness:

PROPOSITION 3.7. *A derivation in $\mathcal{C}$ is fair (with respect to $\Rightarrow$) if the set $CP(R_\infty)$ of all critical pairs between persisting rules is a subset of the set $\bigcup_k E_k$ of all deduced equations.*

THEOREM 3.8 (Correctness). *The limit R_∞ of a non-failing fair derivation is convergent.*

PROOF. As all rewrite systems R_i in a derivation are contained in the given reduction ordering $\succ$, the limit R_∞ is terminating. To prove that R_∞ is Church-Rosser, we need to show that whenever an equation $s \approx t$ is provable in R_∞, then it is provable by a rewrite proof. Let P be a minimal proof of $s \approx t$ in R_∞ with respect to the (well-founded) proof ordering $\Rightarrow^*$. Suppose P is not a rewrite proof. Then, by existence (in the non-failing case) and fairness, there exists a proof P' of $s \approx t$ in $\bigcup_i (E_i \cup R_i)$, such that $P \Rightarrow^+ P'$. But then (by persistence) there also exists a proof P'' of $s \approx t$ in $E_\infty \cup R_\infty$, such that $P' \Rightarrow^* P''$. Since $E_\infty = \emptyset$, this P'' would be a strictly smaller proof in R_∞. This contradicts the assumption that P is minimal. Hence, P must be in normal form. $\square$

It is tempting to strengthen the collapse rule, so that one can reduce the left-hand side of a rule by another rule with the same left side. This, however, can not always be done without losing correctness. Consider the following initial system of oriented equations:

$$\begin{array}{rcl} c & \rightarrow & a \\ g(x) & \rightarrow & x \\ f(x,b) & \rightarrow & x \\ f(x,g(y)) & \rightarrow & f(g(x),y) \\ f(b,z) & \rightarrow & c \end{array}$$

along with a lexicographic path ordering (see [17]) in which the function symbols in order of decreasing precedence are f, c, g, b, a, and in which the first argument of f is more significant than the second. Deduction generates the rules $c \rightarrow b$ and $f(g(b),z) \rightarrow c$. The latter (unreduced) produces $c \rightarrow g(b)$ and $f(g(g(b)),z) \rightarrow c$. Now, $c \rightarrow g(b)$ can be used to collapse $c \rightarrow b$ away (with the proposed lax inference rule) and $g(x) \rightarrow x$ simplifies the other one away. So their critical pairs do not persist. The new ones, however, generate $c \rightarrow g(g(b))$ and $f(g(g(g(b))),z) \rightarrow c$, and so on, never reaching a convergent system. This, despite the fact that there is a finite one:

$$\begin{array}{rcl} f(x,a) & \rightarrow & x \\ f(a,z) & \rightarrow & a \\ b & \rightarrow & a \end{array}$$

On the proof level, the proof of $b \approx a$ gets stretched out, more and more.

It is all right to use older rules to collapse newer ones with the same left-hand side. To allow for this, we should compare left-hand sides using a relation $\rhd'$ that includes encompassment and makes newer rules bigger, instead of just encompassment. This change should be made in the inference rule, transformation rule, and proof ordering; see also Section 6.

Correctness, in our sense, of a specific completion procedure was first proved in [33]. By formulating completion as an equational inference system and deriving corresponding completeness results, we obtain correctness of a wide class of different completion procedures. The proof ordering approach is comparatively simple and intuitive, especially for dealing with simplification inferences. The intrinsic difficulty of applying our approach consists in finding a suitable ordering for proving normalization. Once an appropriate proof ordering has been found, the remaining verification steps are straightforward. We use multisets of the individual costs of steps, since that allows one proof step to be replaced by any number of smaller ones. In particular, proof transformations that delete steps will always result in a decrease in a multiset ordering. For standard completion, steps are ordered by triples, $(\{s,t\},u,t)$ or $(\{s\},u,t)$, each component of which is designed to handle some of the transformations. Were we to only have the first four inference rules, the first component alone would suffice: The cost $\{s\},\{s\}$ of the two steps in a peak $u \leftarrow s \rightarrow v$ would be replaced by the cost $\{u,v\}$ of an application of a critical pair; orienting reduces the cost of a step from $\{u,v\}$ to just $\{u\}$ or $\{v\}$; and simplifying equations reduces the cost of a step from $\{s,t\}$ to $\{s,u\}$, for some u smaller than t. But once we include composition and collapsing, this ordering must be refined: the second component was designed to handle the two steps that replace a collapsed step, while the third component makes using a composed rule smaller than using the original. Küchlin [45] has also discussed completion from the point of view of simplification.

3.4. Completeness of Completion

We can show the following:

THEOREM 3.9. *Suppose there exists some finite convergent rewrite system that is contained in a reduction order $\succ$ and which has the same equational theory as E_0. Then, any fair, unfailing derivation from E_0, using $\succ$, will reach a convergent system (not necessarily the same one) after a finite number of steps.*

PROOF. Let S be the system in the supposition and let S' be a system with the same left-hand sides for each rule, but with right-hand sides replaced by their normal-form in S. The system S' is also convergent: It is terminating, since $\rightarrow_{S'} \subseteq \rightarrow_S^*$. It is Church-Rosser, since for any proof $s \leftrightarrow_{S'}^* t$ there is also a proof $s \leftrightarrow_S^* t$. Since S is convergent, there is also a proof $s \rightarrow_S^! v \leftarrow_S^! t$. Moreover, all ways of rewriting must lead to the same normal form v; in particular, we can apply rules in S to mimic S'. Hence, $s \rightarrow_{S'}^* v \leftarrow_{S'}^* t$.

We know that each rule $s \rightarrow t$ in S' is a theorem in E_0. Hence, in any fair derivation $E_0 \vdash E_1 \vdash E_2 \vdash \cdots$, each rule will have a persisting rewrite proof after a finite number of steps. In fact, these rewrite proofs must be of the form $s \rightarrow_{R_i}^* t$, since t can not be reducible without violating the fact that both S' and R_i are contained in the same irreflexive ordering $\succ$. Let R be the finite set of persisting rules needed for rewrite proofs of all rules in S'.

We show that R is Church-Rosser. Let $s \leftarrow_R u \rightarrow_R t$ be a peak in R. By soundness, there is a proof $s \leftrightarrow_{S'}^* t$ in S' (since there is one in E_0) and by confluence of S' there is a valley proof $s \rightarrow_{S'}^* v \leftarrow_{S'}^*$. We have $\rightarrow_{S'} \subseteq \rightarrow_R^*$. Therefore, there is also a proof $s \rightarrow_R^* v \leftarrow_R^*$. □

A rewrite system R is *reduced* if, for each rule $l \rightarrow r$ in R, the right-hand side r is irreducible by R and no term s less than l in the encompassment ordering is reducible [19]. It has been shown [14, 48] that there is exactly one (finite or infinite) reduced convergent rewrite system (up to renaming of variables) contained in any given reduction ordering. The contraction rules we have suffice for reducing rules in this way. We can accordingly modify our notion of "normal-form proof" to insist that all rules used in valley proofs be reduced. It would then follow that whenever a finite rewrite system R exists for a given ordering $\succ$, any fair unfailing derivation will discover it.

The extensive use of simplification, composition, and collapse is typical of completion procedures. The procedure of Section 1 keeps rules and equations reduced. That procedure may generate critical pairs from non-persisting rules that are not necessary. On the other hand, not all critical pairs between persisting rules are generated either. For, if the right-hand side of an old rewrite rule is simplified in step 3, no critical pairs are computed with the newly simplified rule. This optimization can be viewed as an application of a "critical pair criterion." The fairness of the procedure can be proved using the techniques described in the next section.

4. Critical Pair Criteria

The efficiency of the completion process depends primarily on the number of critical pairs generated. Simplification can be a very effective mechanism for eliminating superfluous equations. For instance, a critical pair is redundant if it can be reduced to an existing critical pair. Redundancy of a critical pair can often be determined by examining the structure of the associated critical overlap.

A terminating rewrite system R will be called *convergent for* a set of equations E if for every equation $s \approx t$ in E we have $s \rightarrow_R^* v \leftarrow_R^* t$. By a *critical pair criterion* CPC we mean a mapping from sets of equations to sets of equations, where $CPC(E)$ is meant to indicate which critical pairs are deemed redundant. A criterion CPC is *sound* with respect to a reduction ordering $\succ$ if, whenever a rewrite system R is contained in $\succ$ and is convergent for $CP(R) \setminus CPC(R)$, R is Church-Rosser. In other words, a sound criterion provides a characterization of the Church-Rosser property. To be of use in practice, a criterion also has to be compatible with the simplification mechanism employed by completion.

A derivation $E_0 \cup R_0 \vdash E_1 \cup R_1 \vdash E_2 \cup R_2 \vdash \cdots$ is called *fair with respect to* CPC if for every critical overlap P associated with a critical pair $s \approx t$ in $CP(R_\infty) \setminus \bigcup_i CPC(E_i \cup R_i)$ there exists a proof Q in $\bigcup_i(E_i \cup R_i)$, such that $P \Rightarrow^+ Q$. We say that a criterion CPC is *correct* if any non-failing derivation is fair whenever it is fair with respect to CPC. Clearly, if a non-failing derivation is fair with respect to a correct criterion CPC, then its limit R_∞ is convergent.

Every correct criterion is sound [3] but not vice versa. For instance, Zhang and Kapur [59] suggest a criterion PCP, where $PCP(R)$ consists of all critical pairs of a rule $u \rightarrow v$ on another rule $s \rightarrow t$ at a position pq, such that $p, q \neq \Lambda$ and the overlapped term $s\sigma$ is reducible at position p. This criterion can be proved to be sound without much difficulty, but is not correct in this sense.[1] For instance, the derivation

$$\begin{aligned}\{-0 \rightarrow 0,\ \sqrt{-x+x} \rightarrow 0,\ -0+0 \rightarrow 0\} & \\ \vdash\ \ & \{-0 \rightarrow 0,\ \sqrt{-x+x} \rightarrow 0,\ 0+0 \approx 0\} \\ \vdash\ \ & \{-0 \rightarrow 0,\ \sqrt{-x+x} \rightarrow 0,\ 0+0 \rightarrow 0\}\end{aligned}$$

is fair with respect to the criterion, but the final rewrite system is not Church-Rosser, as the term $\sqrt{-0+0}$ has two different normal forms $\sqrt{0}$ and 0. The problem with the criterion is that a critical pair may be redundant at some stage during the completion process, but non-redundant at a later stage. (Note, in particular, that the critical pair $0 \approx \sqrt{0+0}$ obtained by superposing $-0 \rightarrow 0$ on

[1] Zhang and Kapur [59] attribute the criterion to Winkler and Buchberger [58], but the latters' actual, more subtle criterion is a specific instance of the composite criterion described below and is correct.

$\sqrt{-x+x} \to 0$ is in $PCP(E_0 \cup R_0)$ and therefore is superfluous according to the criterion.) All of the correct criteria that have been suggested in the literature can be viewed as applications of compositeness, a concept that is the analogue, on the *proof level*, of the *term-based* notion of "connectedness" [12].

A peak P of the form $s \leftarrow_E u \rightarrow_E t$ is called *composite* with respect to a proof ordering $\gg$ if there exist a sequence of proofs $Q_1, \ldots, Q_n$ in E, where Q_i is a proof of $u_{i-1} \approx u_i$, such that $u_0 = s$, $u_{n+1} = t$, $u \succ u_i$, and $P \gg Q_i$. We also call $Q_1, \ldots, Q_n$ a *decomposition* of the peak P. Note that it is possible that the composition $Q_1 \cdots Q_n$ of all proofs Q_i is bigger than P with respect to a proof ordering $\gg$. A critical pair is called composite if its corresponding critical overlap is.

Let $\gg$ be a proof ordering like the one used for standard completion, except that we add a component to the cost of a step: the cost of an equational proof step $s \rightarrow^p_{u\approx v} t$ (or $s \leftarrow^p_{u\approx v} t$) is now $(\{s,t\}, s|_p, u, t)$, and for a rewrite proof step $s \rightarrow^p_{u\rightarrow v} t$ (or $s \leftarrow^p_{u\rightarrow v} t$) is now $(\{s\}, s|_p, u, t)$. Proof steps are compared as before, using the proper subterm ordering for the new second component. By $CCP(E)$ we denote the set of all critical pairs in $CP(E)$ that are composite with respect to $\gg$.

THEOREM 4.1. *The composite criterion CCP is correct.*

PROOF. Let $E_0 \cup R_0 \vdash E_1 \cup R_1 \vdash E_2 \cup R_2 \vdash \cdots$ be a fair derivation with respect to CCP, where $E_\infty = \emptyset$. We use induction on $\gg$ to show that every proof P in $\bigcup_i (E_i \cup R_i)$ can be transformed to a rewrite proof in R_∞.

By persistence, if P is not a proof in R_∞ or contains a non-proper overlap, then there is some proof Q with $P \Rightarrow^+ Q$. On the other hand, if P is a proof in R_∞, then any proper overlap $s \leftarrow_{R_\infty} u \rightarrow_{R_\infty} t$ can be written as $v[P'\sigma]$, where P' is a critical overlap and $s' \approx t'$ the corresponding critical pair in $CP(R_\infty)$.

If the critical pair $s' \approx t'$ is contained in some set $CCP(E_k \cup R_k)$, then there exists a decomposition of P' and consequently also a decomposition $P_1, \ldots, P_n$ of $v[P'\sigma]$, where $P \gg P_i$, for all i with $1 \le i \le n$. We may use the induction hypothesis to infer that each proof P_i can be transformed to a rewrite proof Q_i in R_∞. Let Q' be the composition $Q_1 \cdots Q_n$ of all these rewrite proofs. Since all terms in Q' are strictly smaller than u, we have $P \gg Q'$ and hence may apply the induction hypothesis again, to conclude that there is a rewrite proof of $s \approx t$ in R_∞. Therefore there exists a proof Q with $P \Rightarrow^+ Q$.

If the critical pair $s' \approx t'$ is not contained in any set $CCP(E_k \cup R_k)$, then by fairness there exists a proof Q with $P \Rightarrow^+ Q$.

In sum, any proof P that is not a rewrite proof in R_∞ can be transformed to a simpler proof Q. By the induction hypothesis, the proof Q (and hence P) can be transformed to a rewrite proof. $\square$

Various techniques have been used in practice to check for compositeness. The basic idea is to check whether, in a critical overlap $P = t\sigma \leftarrow_R s\sigma[u\sigma]_p \rightarrow_R s\sigma[v\sigma]_p$ (of $u \to v$ on $s \to t$), the overlapped term $s\sigma[u\sigma]_p$ can be reduced in other ways than indicated by the overlap.

Suppose $s\sigma$ can be rewritten to a term w by applying a rule $s' \to t'$ at a position p'. If $p' = pq$ for some position q, such that either $q \neq \Lambda$ or else both $s \rhd s'$ and $u \rhd s'$, then P can be decomposed into two peaks $P_1 = t\sigma \leftarrow_R s\sigma \rightarrow_R w$ and $P_2 = w \leftarrow_R s\sigma \rightarrow_R s\sigma[v\sigma]_p$. (It can easily be checked that $P \gg P_1$ and $P \gg P_2$.) Since the rule $s \to t$ can be collapsed if $s \rhd s'$, it is sufficient to require only $q \neq \Lambda$ in completion procedures that construct reduced rewrite systems. This special case of compositeness was introduced by Kapur, Musser, and Narendran [40]. For example, if R

contains rewrite rules $(x^- \cdot y)^- \to y^- \cdot x^{--}$, $x \cdot x^- \to e$, and $x^{--} \to x$, then the critical overlap

$$x^{---} \cdot x^{--} \leftarrow_R (x^- \cdot x^{--})^- \to_R e^-$$

(between the first two rules) is composite, because the subterm x^{--} in $x^- \cdot x^{--}$ is reducible.

If the position p' is not below p, additional information is required to ensure compositeness. Suppose $p' \neq \Lambda$ and there exists a proof P_1 of $t\sigma \approx w$, such that $P \succ\!\succ P_1$. By taking P_2 to be the peak $w \leftarrow_R s\sigma \to_R s\sigma[v\sigma]$, we obtain a (binary) decomposition of the original peak. The existence of a suitable proof P_1 is difficult to determine, in general, but is guaranteed if the critical pair between rules $s \to t$ and $s' \to t'$ has already been computed. Techniques for keeping track of previously computed critical pairs have been described by Küchlin [43], for instance.

We conclude this section by using compositeness to establish a new Church-Rosser result. Suppose $u \to v$ and $u' \to v'$ can both be superposed on the same rule $s \to t$, at positions p and p', respectively. Let $t\sigma \approx s\sigma[v\sigma]_p$ and $t\sigma' \approx s\sigma'[v'\sigma']_{p'}$ be the corresponding critical pairs. We say that the first critical pair is *subsumed* by the second if $p' \neq \Lambda$ and there exists a substitution τ, such that $x\sigma = (x\sigma')\tau$, for all variables x in s. A set of critical pairs S, with $S \subseteq CP(R)$, is called *complete* if each critical pair in $CP(R)$ is subsumed by some critical pair in S.

THEOREM 4.2. *A terminating rewrite system R is Church-Rosser if and only if it is convergent for some complete subset of $CP(R)$.*

PROOF. The only-if direction is trivial. For the other direction, let $\succ$ be a reduction ordering containing R. We prove that whenever S is a complete subset of $CP(R)$, then all critical pairs in $CP(R) \setminus S$ are composite with respect to $\succ\!\succ$. Let $t\sigma \approx s\sigma[v\sigma]$ be such a critical pair and P be the corresponding overlap $t\sigma \leftarrow_R s\sigma[u\sigma] \to_R s\sigma[v\sigma]$. Since S is complete, this critical pair is subsumed by some critical pair $t\sigma' \approx s\sigma'[v'\sigma']_{p'}$ in S, where $p' \neq \Lambda$ and $(t\sigma')\tau = t\sigma$ and $(s\sigma'[v'\sigma'])\tau = s\sigma[v'\sigma]$, for some substitution τ. Since R is convergent for S, there is a rewrite proof of $t\sigma' \approx s\sigma'[v'\sigma']$. Let P_1 be a corresponding rewrite proof of $t\sigma \approx s\sigma[v'\sigma]$ and P_2 be the proof $s\sigma[v'\sigma] \leftarrow_R s\sigma \to_R s\sigma[v\sigma]$. The proof P_2 differs from P in that its first rewrite step is simpler: $\{(\{s\sigma\}, s\sigma, s, t\sigma)\} \succ\!\succ \{(\{s\sigma\}, u'\sigma, u', t\sigma)\}$ (note that subsumption implies that $u'\sigma$ is a proper subterm of $s\sigma$). Since we also have $P \succ\!\succ P_1$, the two proofs P_1 and P_2 form a decomposition of P. We conclude that all critical pairs are composite, which, by the correctness of the composite criterion, implies that R is Church-Rosser. $\square$

This theorem has applications to rewrite systems R containing rules $t[s,s] \to u$ with multiple occurrences of the same subterm on the left-hand side. For each critical pair obtained by superposing on one occurrence of s, there is a corresponding critical pair obtained by superposing on another occurrence of s. As all these critical pairs subsume each other, it is sufficient to compute just one of them.

5. *Completion Modulo a Congruence*

Some equations induce non-terminating rewrite relations. For example, the commutativity axiom $x + y \approx y + x$ enables a rewrite $t + t \to t + t$. Standard completion fails for such problematic equations, but it is often possible to construct rewrite systems for which normal forms are unique up to a simple equivalence relation.

Let A be a set of equations, assumed to be symmetric for simplicity. A rewrite system R is called *Church-Rosser modulo* A if, for all terms s and t with $s \leftrightarrow^*_{A\cup R} t$, there are terms u and v, such that $s \to^*_R u \leftrightarrow^*_A v \leftarrow^*_R t$. A proof of the form $s \to^*_R u \leftrightarrow^*_A v \leftarrow^*_R t$ is called a *rewrite proof modulo* A.

The rewrite system R/A consists of all rules $s \to t$, for which $s \leftrightarrow^*_A u \to_R v \leftrightarrow^*_A t$. It represents the rewrite relation induced by R on equivalence classes of A. We say that a reduction ordering $\succ$ is *compatible* with A if $s \succ t$ implies $u \succ v$, for all terms s, t, u, and v with $u \leftrightarrow^*_A s$ and $t \leftrightarrow^*_A v$. A system R/A is terminating if and only if R is contained in a reduction ordering $\succ$ that is compatible with A.

Consider, for example the set A of associativity and commutativity axioms

$$\begin{array}{rcl} x+y & \approx & y+x \\ x+(y+z) & \approx & (x+y)+z \end{array} \quad \begin{array}{rcl} x\times y & \approx & y\times x \\ x\times(y\times z) & \approx & (x\times y)\times z \end{array}$$

and the set R [32] of rules

$$\begin{array}{rcl} x+0 & \to & x \\ 0+x & \to & x \\ f(0) & \to & 1 \end{array} \quad \begin{array}{rcl} x\times 1 & \to & x \\ 1\times x & \to & x \\ f(x+y) & \to & f(x)\times f(y) \end{array}$$

The system R/A is terminating, for if ground terms are evaluated by

$$\begin{array}{rcl} \nu(0) & = & 2 \\ \nu(1) & = & 2 \\ \nu(s+t) & = & \nu(s)+\nu(t) \\ \nu(s\times t) & = & \nu(s)+\nu(t) \\ \nu(f(s)) & = & \nu(s)^2 \end{array}$$

then any application of a rule decreases the value of a ground term, while equivalent terms with respect to A have the same value. The rewrite system R is also Church-Rosser modulo A; a test for this property is described below.

Following the techniques outlined in previous sections, we first describe Huet's method for constructing convergent rewrite systems modulo A from the perspective of proof normalization. In this context, a normal-form proof is a rewrite proof modulo A by R. As before, by a peak we mean a proof $s \leftarrow_R u \to_R t$. By a *cliff* we mean a proof $s \leftrightarrow_A u \to_R t$ or $s \leftarrow_R u \leftrightarrow_A t$. A rewrite system R is Church-Rosser modulo A if and only if there is a rewrite proof modulo A in R for every equation provable in $A \cup R$. A proof in $A \cup R$, on the other hand, is a rewrite proof modulo A if and only if it contains no peak or cliff.

The Critical Pair Lemma indicates that every peak can be replaced by a rewrite proof unless it is a proper overlap. Let P be a cliff $s \leftrightarrow_A u \to_R t$. If P is a non-overlap, then a rewrite proof modulo A can be obtained by commuting the two proof steps. If P is a proper overlap, then by the Critical Pair Lemma $s \leftrightarrow_{CP(A\cup R)} t$. Nested cliffs can be problematic. For example, in

$$a\times b \leftrightarrow_A a\times a \to_R a$$

the application of an equation $a \approx b$ is nested within the rule $x\times x \to x$. The cliff can be replaced by a proof

$$a\times b \leftrightarrow_A b\times b \to_R b \leftrightarrow_A a$$

(which contains a similar nested cliff!) but not by a rewrite proof modulo A. This example involves a *non-left-linear* rewrite rule. A cliff in which the application of an equation of A is nested within the application of a *left-linear* rule can always be replaced by a rewrite proof modulo A. Therefore the Critical Pair Lemma also provides an Existence Lemma for rewrite proofs modulo A, provided rewrite rules are left-linear.

Let $\succ$ be a reduction ordering that is compatible with A. The inference system $\mathcal{L}$ contains all inference rules of $\mathcal{C}$, but restricted so that all sets of equations are of the form $A \cup E \cup R$, with A being fixed. In addition, $\mathcal{L}$ contains two new inference rules:

Extension: $$\frac{A \cup E \cup R}{A \cup E \cup \{s \approx t\} \cup R} \quad \text{if } s \approx t \in CP(A \cup R) \setminus CP(A)$$

Deletion: $$\frac{A \cup E \cup \{s \approx t\} \cup R}{A \cup E \cup R} \quad \text{if } s \leftrightarrow^*_A t$$

which are reflected by proof transformation rules

$$\begin{array}{ccc} s \underset{A}{\longleftrightarrow} u \underset{R}{\longrightarrow} t & \Rightarrow & s \longleftrightarrow_E t \\ s \longleftrightarrow_E t & \Rightarrow & s \longleftrightarrow^*_A t \end{array}$$

(The first kind of transformation covers not only extension, but also elimination of non-overlaps and certain nested overlaps between A and R.)

Lemma 5.1. *The proof transformation relation $\Rightarrow$ for $\mathcal{L}$ is terminating.*

Proof. We define a well-founded ordering using a suitable measure of the complexity of a proof. The cost of a proof step $s \leftrightarrow^p_e t$ is defined to be the triple $(\{s, \bot\}, \bot, \bot)$, if e is an equation in A; $(\{s\}, u, \{t\})$, if e is a rule $u \to v$; and $(\{t\}, v, \{s\})$ if e is a rule $u \leftarrow v$. If e is neither an equation in A nor a rule, the cost of the proof step is $(\{s\}, \top, \{t\})$, if $s \succ t$; $(\{t\}, \top, \{s\})$ if $t \succ s$; and $(\{s, t\}, \bot, \bot)$, otherwise. The complexity of a proof is the multiset of all costs of its proof steps. Proof steps are compared using the multiset extension of the reduction ordering $\succ$ in the first component, the encompassment ordering $\rhd$ in the second component, and the reduction ordering $\succ$ in the last component. The symbols $\top$ and $\bot$ are assumed to be maximum and minimum elements, respectively, in any of these orderings. We denote this ordering on triples by $\succ_{lex}$. Proofs are compared using the multiset extension of this ordering. This proof ordering $\succ\!\!\succ$ is well-founded and contains the transformation relation $\Rightarrow$ based on the old and new transformation rules. We show some representative cases in detail:

Extension. We have $(s \leftrightarrow_A u \to_R t) \succ\!\!\succ (s \leftrightarrow_E t)$, because $(\{s, \bot\}, \bot, \bot) \succ_{lex} (\{s\}, \top, t)$.

Orientation. $(s \leftrightarrow_E t) \succ\!\!\succ (s \to_R t)$, because $(\{s\}, \top, t) \succ_{lex} (\{s\}, u, t)$, for any term u.

Deletion. $(s \leftrightarrow_E t) \succ\!\!\succ (s \leftrightarrow^*_A t)$, because $\{s, t\} \succ_{mul} \{u, \bot\}$, for every term u with $s \leftrightarrow^*_A u$. (Note that $s \leftrightarrow^*_A t$ implies $s \not\succ t$ and $t \not\succ s$.)

Simplification. If $s \succ t$, then $(s \leftrightarrow_E t) \succ\!\!\succ (s \to_R u \leftrightarrow_E t)$, because $s \succ u$ and $(\{s\}, \top, t) \succ_{lex} (\{s\}, v, u)$, for any term v.

If $t \succ s$, then $(s \leftrightarrow_E t) \succ\!\!\succ (s \to_R u \leftrightarrow_E t)$, because $(\{t\}, \top, s) \succ_{lex} (\{s\}, v, u)$ and $(\{t\}, \top, s) \succ_{lex} (\{t\}, \top, u)$.

If neither $s \succ t$ nor $t \succ s$, then $(s \leftrightarrow_E t) \succ\!\!\succ (s \to_R u \leftrightarrow_E t)$, because $s \succ u$ and $(\{s, t\}, \bot, \bot) \succ_{lex} (\{s\}, v, u)$. ∎

Using induction on $\Rightarrow^+$ we can prove:

THEOREM 5.2 ([32]). *Let R be a rewrite system and A be a set of equations, such that R/A terminates. The system R is Church-Rosser modulo A if and only if, for all terms s and t with $s \leftarrow_R u \rightarrow_{A \cup R} t$, there are terms v and w, such that $s \rightarrow_R^* v \leftrightarrow_A^* w \leftarrow_R^* t$.*

Again, computation of critical pairs suffices for fairness:

PROPOSITION 5.3. *A derivation $A \cup E_0 \cup R_0 \vdash A \cup E_1 \cup R_1 \vdash \cdots$ in $\mathcal{L}$ is fair if the set of all persisting critical pairs $CP(A \cup R_\infty) \setminus CP(A)$ is a subset of $\bigcup_k E_k$.*

We obtain the following proof normalization result:

THEOREM 5.4. *Let R_0 be a rewrite system and $\succ$ be a reduction ordering that contains R_0 and is compatible with A. If $E_0 \cup R_0 \vdash E_1 \cup R_1 \vdash \cdots$ is a fair unfailing derivation in $\mathcal{L}$, such that R_∞ is left-linear, then R_∞ is convergent modulo A.*

PROOF. Let $A \cup E_0 \cup R_0 \vdash A \cup E_1 \cup R_1 \vdash \cdots$ be a fair derivation as indicated. We show that whenever a proof P in $A \cup E_\infty \cup R_\infty$ contains a peak $s \leftarrow_{R_\infty} u \rightarrow_{R_\infty} t$, a cliff $s \leftrightarrow_A u \rightarrow_{R_\infty} t$, or a cliff $s \leftarrow_{R_\infty} u \leftrightarrow_A t$, then there is a proof Q in $A \cup E_j \cup R_j$, for some $j \geq i$, such that $P \Rightarrow^+ Q$. By the Critical Pair Lemma, any peak or cliff in P which is a non-overlap or nested can be replaced by a rewrite proof modulo A. (Left-linearity of R_∞ guarantees that there are no problematic nested cliffs.) By fairness, every proper overlap can also be replaced by a simpler proof. Since the derivation is unfailing, $E_\infty = \emptyset$, and we may conclude that any non-normal persisting proof can be transformed to a simpler proof. The assertion follows by persistence. □

The above completion method has been suggested by Huet [32]. Its main drawback is the restriction of left-linearity. (Note that a non-left linear rule can be inferred from left-linear ones.) A different approach to rewriting modulo a congruence employs a stronger rewrite relation R_A that is based on A-matching and in which every problematic nested cliff $s \leftrightarrow_A u \rightarrow_R t$ can be regarded as a single rewrite step $s \rightarrow_{R_A} t$. Then a rewrite system R has to be constructed so that R_A is Church-Rosser modulo A. This eliminates the problem with nested cliffs, but introduces a new problem of handling more general overlaps involving the rewrite relation R_A. Such overlaps can be effectively dealt with if a finite, complete A-unification algorithm is given. An associative-commutative completion procedure based on this approach has been described by Peterson and Stickel [53]; for a formulation within the inference rule *cum* proof normalization approach, see [7]

6. Ordered Completion

Standard completion fails whenever an equation $s \approx t$ between persistently irreducible, yet incomparable terms s and t is generated. Commutativity, $x \times y \approx y \times x$, is an example of such an unorientable equation, as the two terms $x \times y$ and $y \times x$ are incomparable with respect to any reduction ordering. The strategy used by a completion procedure to construct a derivation may determine whether or not an unorientable equation is generated [21]. To avoid failure a procedure may have to systematically enumerate all possible derivations (for example, via backtracking). In some cases, standard completion is bound to fail even with backtracking. In fact, the method may fail even when it is supplied with a reduction ordering suitable for a convergent system that does exist.

For example, if the initial set of equations E_0 is

$$\begin{aligned} 1 \times (-x + x) &\approx 0 \\ 1 \times (x + -x) &\approx x + -x \\ -x + x &\approx y + -y \end{aligned}$$

then standard completion fails, regardless of which reduction ordering is supplied as input! The only inference rule that can possibly be applied to the above set of equations is orientation, which may result in two rules:

$$\begin{aligned} 1 \times (-x + x) &\to 0 \\ 1 \times (x + -x) &\to x + -x \end{aligned}$$

(no other orientation is possible). These two rules do not overlap and constitute a convergent rewrite system. The third equation is unorientable and both its sides are irreducible by the above two rules. As a consequence, no inference rules are applicable. Nonetheless, there exists a convergent system

$$\begin{aligned} -x + x &\to 0 \\ x + -x &\to 0 \\ 1 \times 0 &\to 0 \end{aligned}$$

for the given equational theory.

Unorientable equations, in which one side contains a variable not occurring in the other side, can sometimes be dealt with by introducing new function symbols, a technique suggested in [42]. For instance, if the equation $-x + x \approx y + -y$ is replaced by two rules $-x + x \to c$ and $y + -y \to c$, where c is a new (minimal) constant, then completion succeeds in constructing a convergent system of four rules:

$$\begin{aligned} -x + x &\to c \\ x + -x &\to c \\ 1 \times c &\to c \\ 0 &\to c \end{aligned}$$

which represents a conservative extension of the original equational theory (and, hence, also provides a decision procedure). The preceding example indicates an inherent inadequacy of standard completion, however, and very often this technique just leads to the introduction of ever more new function symbols.

We pursue a different approach for dealing with unorientable equations, called ordered completion, which is a refutationally complete theorem prover for equational theories. The method requires neither backtracking, nor introduction of new function symbols. As an equational theorem prover, it has the advantage over paramodulation [57] that equations can always be kept in fully simplified form and fewer equational consequences need to be considered, since the ordering supplied to the procedure gives some measure of direction to the prover.

A convergent rewrite system defines unique normal forms for *all* terms. In applications such as (refutational) theorem proving uniqueness of *ground* normal forms is sufficient. In this section, we present an inference system, called "ordered completion," for the construction of systems of equations that define unique ground normal forms.

Let E be a set of equations E and $\succ$ be a reduction ordering. We write $s \rightarrow_{E^\succ} t$ (or $t \leftarrow_{E^\succ} s$) if $s \leftrightarrow_E t$ and $s \succ t$. We say that E is *ground convergent* (with respect to $\succ$) if for all ground terms s and t with $s \leftrightarrow_E^* t$, there exists a ground term v, such that $s \rightarrow_{E^\succ}^* v \leftarrow_{E^\succ}^* t$. A ground convergent system defines unique ground normal forms.

By a *rewrite proof* (with respect to $\succ$) we mean a proof of the form $s \rightarrow_{E^\succ}^* v \leftarrow_{E^\succ}^* t$. A set of equations E is ground convergent if and only if there exists a ground rewrite proof for every equation between ground terms that is provable in E. A ground proof is either a rewrite proof or else contains a peak $s \leftarrow_{E^\succ} u \rightarrow_{E^\succ} t$ or an unorientable equality step $s \leftrightarrow_E t$ (where neither $s \succ t$ nor $t \succ s$). If the reduction ordering is total on ground terms, unorientable equality steps cannot occur, though.

A reduction ordering $\succ$ is said to be *complete* (with respect to E) if, whenever s and t are distinct ground terms (and $s \leftrightarrow_E^* t$), then either $s \succ t$ or else $t \succ s$. A ground rewrite proof with respect to a complete reduction ordering $\succ$ is simply a (ground) proof containing no peak $s \leftarrow_{E^\succ} u \rightarrow_{E^\succ} t$. Proof normalization corresponds to elimination of these peaks which can be achieved by computation of suitable critical pairs.

Let $s \approx t$ and $u \approx v$ be equations with no variables in common, and suppose that some non-variable subterm $s|_p$ of s is unifiable with u, σ being their most general unifier. We say that the *superposition* of $u \approx v$ on $s \approx t$ at position p determines a *(ordered) critical pair* $t\sigma \approx s\sigma[v\sigma]_p$ (with respect to the ordering $\succ$) if there exists a (ground) substitution τ, such that $(s\sigma)\tau \succ (t\sigma)\tau$ and $(s\sigma)\tau \succ (s\sigma[v\sigma]_p)\tau$. As before, the peak $t\sigma \leftarrow_{t\approx s}^{\Lambda} s\sigma \rightarrow_{u\approx v}^{p} s\sigma[v\sigma]_p$ is a *critical overlap.* By $CP^\succ(E)$ we denote the set of all ordered critical pairs (with respect to $\succ$) between equations in $E \cup E^{-1}$.

For example, the two equations $(x \cdot y) \cdot (z \cdot w) \approx (x \cdot z) \cdot (y \cdot w)$ and $(x \cdot y) \cdot x \approx x$ overlap in $((u \cdot v) \cdot u) \cdot (v' \cdot v) \leftrightarrow_E ((u \cdot v) \cdot v') \cdot (u \cdot v) \leftrightarrow_E u \cdot v$ and define an ordered critical pair $((u \cdot v) \cdot u) \cdot (v' \cdot v) \approx u \cdot v$ with respect to the lexicographic path ordering.

An ordered critical pair of an equation on itself at the top need not be trivial. For instance, superposing the equation $a^- \approx x \cdot a$ on itself at the top may yield a non-trivial equation $x \cdot a \approx y \cdot a$.

The computation of ordered critical pairs with respect to a reduction ordering $\succ$ requires that one be able to decide, given terms s, t, u, and v, whether there exists a ground substitution σ, such that $s\sigma \succ t\sigma$ and $u\sigma \succ v\sigma$. This question is decidable, for instance, if $\succ$ is a path ordering based on a total precedence [15, 37]. If inequations can not be solved in general for the given ordering $\succ$, or if the decision procedure is prohibitively expensive, then completion may have to deduce more equations than are actually necessary to ensure fairness. However, the computation of standard critical pairs always suffices for fairness, as $CP(E \cup E^{-1})$ is a superset of the set of ordered critical pairs $CP^\succ(E)$.

The Critical Pair Lemma can be adapted to ordered critical pairs without much difficulty:

LEMMA 6.1 (Ordered Critical Pair Lemma [46]). *Let $\succ$ be a complete reduction ordering with respect to E. For all ground terms s, t, and u with $s \leftarrow_{E^\succ} u \rightarrow_{E^\succ} t$, there either exists a ground term v, such that $s \rightarrow_{E^\succ}^* v \leftarrow_{E^\succ}^* t$, or else $s \leftrightarrow_{CP^\succ(E)} t$.*

PROOF. The proof is similar to that of the Critical Pair Lemma. Non-overlaps and proper overlaps, in particular, can be dealt with in the same way. Any nested overlap $s \leftarrow_{E^\succ} u \rightarrow_{E^\succ} t$ can be replaced by a proof of the form $s \rightarrow_{E^\succ}^* v \leftrightarrow_E w \leftarrow_{E^\succ}^* t$, which is a ground rewrite proof, as the completeness of $\succ$ implies that the proof step $v \leftrightarrow_E w$ is either of the form $v \rightarrow_{E^\succ} w$ or $v \leftarrow_{E^\succ} w$. ∎

For the purpose of normalizing ground proofs, the following expansion rule is sufficient:

$$\text{DEDUCTION:} \quad \frac{E}{E \cup \{s \approx_n t\}} \quad \text{if } s \approx t \in CP^\succ(E)$$

where we also assume that a label n (typically a non-negative number) is assigned to each newly deduced equation. We also assume that a well-founded ordering on labels is supplied. These labels will be used in formulating suitable contraction rules, as discussed next.

We shall not design specific combinations of expansion and contraction for ordered completion, but only specify the proof transformation relation with respect to which contraction may be applied. We first refine the proof ordering for standard completion. Let the cost of a single proof step $s \leftrightarrow^p_{u \approx_n v} t$ be the quadruple $(\{s\}, u, n, t)$, if $s \succ t$; $(\{t\}, u, n, s)$, if $t \succ s$; and $(\{s,t\}, u, n, s)$, otherwise. Proof steps are compared as before (see Lemma 3.6), using the given ordering on labels for the new third component. The corresponding proof ordering is denoted by $\succ\!\!\succ$. By $\Rightarrow$ we denote the proof transformation relation induced by this proof ordering: $P \Rightarrow Q$ if and only if P and Q are proofs of the same equation for which $P \succ\!\!\succ Q$. Observe that whenever P is a proof $s \leftarrow_{E^\succ} u \rightarrow_{E^\succ} t$ and Q is a corresponding rewrite proof $s \rightarrow^*_{E^\succ} v \leftarrow^*_{E^\succ} t$, then $P \Rightarrow Q$.

By an *ordered completion system* we mean any inference system consisting of a version of expansion that includes at least the above deduction rule, and any version of contraction based on the above proof transformation relation. Ordered completion systems are *sound*: whenever $E \vdash E'$, the two congruence relations $\leftrightarrow^*_E$ and $\leftrightarrow^*_{E'}$ are the same.

Various simplification techniques can be covered with the above version of contraction. For instance, the inference rules of standard completion are derived inference rules of ordered completion. The strong distinction between equations and rules that is essential in standard completion, can be viewed as a specific way of assigning labels: all equations get the same label, say 1, while rules get a different label, say 0. We may refine this scheme by assigning to all equations a maximum label, say $\top$, and assigning to rules the index of the set R_i in which it is included as a result of orientation. Orientation thus corresponds to decreasing the label of an equation. Note that this refined labelling scheme allows for collapse of a rule by an older rule with a literally similar left-hand side. Formally, the inference rules, orientation, simplification, collapse, and composition, can be represented as suitable applications of expansion followed by contraction.

Computation of ordered critical pairs between persisting equations ensures fairness.

THEOREM 6.2. *A derivation in ordered completion is fair if the set of ordered critical pairs $CP^\succ(E_\infty)$ is a subset of the set of all derived equations $\bigcup_k E_k$.*

Fair derivations, on the other hand, always succeed:

THEOREM 6.3. *Let $\succ$ be a complete reduction ordering with respect to E, and let E_∞ be the limit of a fair derivation from E. Then E_∞ is ground convergent with respect to $\succ$.*

PROOF. Let $E_0 \vdash E_1 \vdash \cdots$ be a derivation as indicated, and let $\succ$ be a complete reduction ordering with respect to E_0. By completeness of $\succ$, if P is a non-normal ground proof in E_∞, then it has to contain a peak $s \leftarrow_{E^\succsim_\infty} u \rightarrow_{E^\succsim_\infty} t$. If the peak is a non-overlap or nested overlap, then it can be transformed into a ground rewrite proof. In the case of a proper overlap, we have $s \leftrightarrow_{E_i} t$, for some i. Thus, the derivation is fair, and we conclude that P can be transformed to a normal-form proof. This indicates that E_∞ is ground convergent. □

The theorem applies to reduction orderings that are total on equivalent ground terms. All general-purpose term orderings used in practice can be extended to complete orderings. For instance, any ordering based on polynomial interpretations [46, 47] can be extended to a complete ordering by combining it with a well-founded ordering to distinguish ground terms having the same interpretations.

As an example of ordered completion consider the equational theory of the entropic groupoid defined by the two axioms

$$
\begin{aligned}
(x \cdot y) \cdot (z \cdot w) &\approx (x \cdot z) \cdot (y \cdot w) \\
(x \cdot y) \cdot x &\approx x.
\end{aligned}
$$

The first equation is *permutative* and cannot be oriented in any reduction ordering. Standard completion will fail for this set of equations, whereas with ordered completion we can obtain a set of equations

$$
\begin{aligned}
(x \cdot y) \cdot z &\approx (x \cdot w) \cdot z \\
(x \cdot y) \cdot x &\rightarrow x \\
x \cdot (y \cdot z) &\rightarrow x \cdot z \\
((x \cdot y) \cdot z) \cdot w &\rightarrow x \cdot w
\end{aligned}
$$

[31], which is ground convergent with respect to the lexicographic path ordering, and therefore provides a decision procedure for the word problem in the above theory.

It can be shown that, under certain reasonable assumptions, ordered completion actually succeeds in constructing a *convergent* rewrite system, if such a system exists. For details see [8].

Ordered completion is a refutationally complete theorem proving method for equational theories. Let E be a set of equations and $s \approx t$ be an equation provable in E, that is, $s \leftrightarrow_E^* t$. Let $\widehat{s} \approx \widehat{t}$ be a Skolemized version of $s \approx t$ (that is, all variables are replaced by unique Skolem constants). Evidently, we also have $\widehat{s} \leftrightarrow_E^* \widehat{t}$.

Let now $E_0 \vdash E_1 \vdash \cdots$ be any fair derivation in ordered completion, where $\succ$ may be any complete reduction ordering. Since E_∞ is ground convergent with respect to $\succ$, there is a ground rewrite proof $\widehat{s} \rightarrow_{E_i^\succ}^* v \leftarrow_{E_i^\succ}^* \widehat{t}$, for some $i \geq 0$, such that v is irreducible by $E_\infty^\succ$. In other words, the two terms $\widehat{s}$ and $\widehat{t}$ can be reduced to a common normal form by some set $E_i^\succ$, whenever $\widehat{s} \leftrightarrow_E^* \widehat{t}$. By soundness, $\widehat{s}$ and $\widehat{t}$ can only be reduced to a common normal form if $\widehat{s} \leftrightarrow_E^* \widehat{t}$. In sum, ordered completion provides a semi-decision procedure for the validity problem in equational theories.

Ordered completion is in essence a restricted version of paramodulation, enriched by contraction rules (including simplification by rewriting). Computation of equational consequences from unorientable equations already appears in the work of Brown [11] and Lankford [46] on integrating resolution and simplification by rewriting. Peterson [52] proved the refutation completeness of an inference system combining resolution, paramodulation, and simplification with respect to orderings on ground terms that are order-isomorphic to the natural numbers. (This class of orderings excludes many important orderings, though, such as most path orderings.) Fribourg [23] proved the completeness of a restricted version of paramodulation with locking resolution. Hsiang and Rusinowitch [31] used their transfinite semantic tree method to prove the refutation completeness of a restricted version of ordered completion, albeit with weaker simplification than even standard completion, as with semantic trees it is difficult to account for simplification rules depending on the encompassment ordering. For example, in their inference system the equation $(x \cdot y) \cdot (y^- \cdot z) \approx x \cdot z$ can not be simplified by $x \cdot (y \cdot z) \to (x \cdot y) \cdot z$.

Implementations of ordered completion procedures have been reported in [50, 51]. Experiments with a procedure that combines ordered completion with associative-commutative completion are described in [1].

7. *Proof by Consistency*

In many applications, such as algebraic data type specifications and equational programming, equations are intended to define a certain standard model, called the "initial model." Reasoning about algebraic data types and equational programs thus requires proof methods that reflect this initial algebra semantics. Such proof methods typically employ some induction scheme, such as induction on the structure of terms. We shall discuss an alternative approach—proof by consistency—that can be applied to equational theories that are presented as ground convergent rewrite systems.

7.1. *Ground Reducibility*

An equation $s \approx t$ is said to be an *inductive theorem* of E if $s\sigma \leftrightarrow_E^* t\sigma$ for all ground equations $s\sigma \approx t\sigma$. For example, let R_0 be the rules given at the beginning of Section 1 defining addition and multiplication in terms of zero and successor. Associativity and commutativity of $+$ and $\times$ are inductive theorems of R_0, but *not* equational theorems, as there are non-standard models in which the functions denoted by $+$ and $\times$ are not associative and commutative.

Dershowitz [16] has pointed out that an equation $s \approx t$ is an inductive theorem of a (ground) convergent rewrite system R if and only if no equation $u \approx v$ between distinct ground terms irreducible by R, is provable by $R \cup \{s \approx t\}$. Thus, if $s \approx t$ is an inductive theorem of R, then any ground convergent rewrite system for $R \cup \{s \approx t\}$ defines the same ground normal forms as R.[2] This observation is the basis for the proof by consistency method.

For the remainder of this section, let R be a ground convergent rewrite system. Furthermore, let $\succ$ be a reduction ordering containing R. (The results below can also be adapted to reduction orderings for which the transitive closure of the union with R is well-founded.)

The fact that two sets of equations R and $R \cup \{s \approx t\}$ define the same initial algebra can be expressed in proof-theoretic terms. An equation $s \approx t$ is called *consistent with* R if, for every ground equation $s\sigma \approx t\sigma$, the two terms $s\sigma$ and $t\sigma$ can be reduced to identical normal forms by R. Otherwise, $s \approx t$ is said to be *inconsistent* with R (since it equates two distinct elements of the initial algebra of R). A set of equations C is said to be consistent with R if all its equations are consistent with R; and inconsistent, otherwise.

THEOREM 7.1 ([16]). *A set of equations C is consistent with a ground convergent rewrite system R if and only if all equations in C are inductive theorems of R.*

[2] If R is ground convergent, then the algebra defined on the set of ground normal-form terms is an initial model of R [28]. It is isomorphic to the quotient of the set of ground terms by the congruence $\leftrightarrow_R^*$.

Consistency in this sense is not decidable (not even semi-decidable). However, we will show that any inconsistent set of equations can be transformed so that its inconsistency can be verified.

First observe that an inconsistency is indicated by a ground proof of the form

$$s' \overset{!}{\underset{R}{\longleftarrow}} s\sigma \overset{\Lambda}{\underset{s\approx t}{\longleftrightarrow}} t\sigma \overset{!}{\underset{R}{\longrightarrow}} t'$$

where $s \simeq t$ is an equation in C and s' and t' are distinct irreducible terms. We call such a ground proof an *inconsistency proof* for $s \approx t$. An inconsistency proof is said to be in *normal form* if (i) $s' = s\sigma$ and $t' = t\sigma$, or (ii) $s \succ t$ and $s' = s\sigma$, or (iii) $t \succ s$ and $t' = t\sigma$. In other words, a normal-form proof is one in which rewrite steps can be applied only to the smaller term of $s\sigma$ and $t\sigma$. An equation $s \approx t$ is said to be *verifiably inconsistent* if there is a normal inconsistency proof for it.

This case of inconsistency is indeed decidable. The key here is the notion of "ground reducibility." A term t is called *ground reducible* by R if all its ground instances are reducible by R (see [36]). For example, every term in which the function symbol $+$ or $\times$ occurs is ground reducible by the system R_0, since in any ground term the subterm rooted at the rightmost occurrence of $+$ or $\times$ can be rewritten by one of the four rules.

LEMMA 7.2. *An equation $s \approx t$ is verifiably inconsistent with R if and only if (i) $s \succ t$ and s is not ground reducible by R, or (ii) $t \succ s$ and t is not ground reducible by R, or (iii) $s\#t$ is not ground reducible by $R \cup \{x\#x \to \top\}$, where $\#$ and $\top$ are new function symbols.*

For example, the equation $x'' \approx x'$ is verifiably inconsistent, as there is a ground instance $0'' \approx 0'$, in which both terms are irreducible, yet distinct; and hence $0''\#0'$ is not ground reducible by $R \cup \{x\#x \to \top\}$.

Ground reducibility is decidable for finite rewrite systems [54, 39], but, according to [38] is in exponential time even for left-linear rewrite systems. Algorithms for deciding ground-reducibility with respect to left-linear rewrite systems have been described in [41, 36]. In theories with free constructors ground reducibility is trivially decidable: a term is ground reducible if and only if it contains a non-constructor symbol (cf. the discussion of constructors in [34, 36, 24]). In the example above, the function symbols 0 and $'$ are constructors: all ground normal-form terms are built from these two symbols only. The constructors are also free: no two ground terms built solely from these two symbols are equivalent.

To sum up, we have the following result:

THEOREM 7.3. *It is decidable whether an equation $s \approx t$ is verifiably inconsistent with a given ground convergent rewrite system R.*

7.2. Proofs of Inconsistency

The design of inference rules for normalizing inconsistency proofs can be approached as usual. Since a non-normal inconsistency proof contains a subproof $u \leftarrow_R s\sigma \leftrightarrow_C t\sigma$ or $u \leftarrow_R t\sigma \leftrightarrow_C s\sigma$, where $s \not\succ t$, proof transformations of the form

$$u \underset{R}{\longleftarrow} s\sigma \underset{C}{\longleftrightarrow} t\sigma \quad \Rightarrow \quad u \underset{C}{\longleftrightarrow} t\sigma$$

provide a suitable basis for such proof normalization. Moreover, we know that in inconsistency proofs the equation in C is applied at the top, so that we need to consider only proper overlaps and nested peaks. Proper overlaps can be eliminated via computation of suitable critical pairs. We denote by $CP_R(C)$ the set of all critical pairs of rules in R on an equation $s \approx t$ in $C \cup C^{-1}$, where $t \not\succeq s$.

These considerations lead to the following expansion rule:

$$\textsc{Deduction:} \qquad \frac{C}{C \cup \{s \approx t\}} \qquad \text{if } s \approx t \in CP_R(C)$$

The equations in C will also be called *conjectures*. Deduction already suffices to derive a verifiably inconsistent equation from any inconsistent set of initial conjectures. In addition, we shall use the following contraction rules:

$$\textsc{Deletion:} \qquad \frac{C \cup \{t \approx t\}}{C}$$

$$\textsc{Simplification:} \qquad \frac{C \cup \{s \simeq t\}}{C \cup \{u \simeq t\}} \qquad \text{if } s \leftrightarrow^{+}_{R \cup L} u \text{ and } s \succ u$$

where L may be any set of inductive theorems ("lemmas") of R.

Deletion is the same as in standard completion, but simplification is more general: a term s can be simplified to any smaller equivalent term u, where equivalence is to be established with respect to $R \cup L$. The ground theory of L is, by assumption, contained in the ground theory of R; but the equational theory presented by L need not be contained in the equational theory presented by R. Rewriting with respect to $R \cup L$ is thus more powerful, in general, than rewriting by R. For instance, if associativity and commutativity are known to be inductive theorems of R, then conjectures can be simplified by associative-commutative rewriting (that is, rewriting with associative-commutative matching), while there is no need to employ associativity and commutativity for deduction. Moreover, simplification does not require every associative-commutative rewrite step to be reducing, but only that the final term in the rewrite sequence be smaller than the initial term.

This possibility of incorporating associativity and commutativity is in marked contrast with the approach, advocated by Jouannaud and Kounalis [36], of extending associative-commutative completion by a suitable inconsistency test. The latter approach mandates the use of associative-commutative unification for deduction, but allows only for weaker simplification in which every single rewrite step is reducing. The notion of ground reducibility needs to be generalized and reduction ordering are limited to associative-commutative orderings. The approach is failure-prone as it has to abort if an equation can not be oriented with respect to the given associative-commutative ordering. Proof by consistency is thus the preferable method, in particular because in applications to abstract data types, associativity and commutativity are usually not part of a data type specification, but rather arise as inductive theorems.

Simplification of conjectures is similar to standard completion:

$$\textsc{Simplification:} \qquad \frac{C \cup \{s \simeq t\}}{C \cup \{u \simeq t\}} \qquad \begin{array}{l}\text{if there is a proof} \\ s \leftrightarrow^{p}_{v \approx w} u \text{ where } v \succ w \\ \text{and } s \rhd v\end{array}$$

The conditions can be slightly relaxed in various ways. For instance, it is sufficient to require $s \succ u$ whenever s is not an instance of v. Further improvements may be gained from labelling conjectures, as we outlined for ordered completion.

By $\mathcal{P}$ we denote the inference system consisting of the above inference rules. This inference system is sound in the following sense:

THEOREM 7.4 (Soundness). *Let R be a ground convergent system. If $C \vdash C'$, then C is consistent with R if and only if C' is.*

Let now the complexity of a proof $u \leftarrow_R^* s\sigma \leftrightarrow_{s \approx t}^{\Lambda} t\sigma \rightarrow_R^* v$ be the triple $(\{s\sigma\}, s, t\sigma)$ if $s \succ t$; $(\{t\sigma\}, t, s\sigma)$ if $t \succ s$; and $(\{s\sigma, t\sigma\}, \bot, \bot)$, otherwise. Let $\succ_{sub}$ be the smallest ordering that contains $\succ$ and such that $t[s] \succ_{sub} s$, for all terms t and proper subterms s of t. The ordering $\succ_{sub}$ is well-founded, but not necessarily a rewrite relation.[3] Let $\succ\!\!\succ$ be the lexicographic combination of the multiset extension of $\succ_{sub}$ in the first component, the encompassment ordering $\rhd$ in the second component, and the ordering $\succ_{sub}$ in the last component. By $\Rightarrow$ we denote the restriction of $\succ\!\!\succ$ to inconsistency proofs: $P \Rightarrow Q$ if and only if P and Q are inconsistency proofs with $P \succ Q$. We emphasize that $\Rightarrow$ is not a proof transformation relation in our original sense, as $P \Rightarrow Q$ does not imply that P and Q are proofs of the same equation. The ordering $\succ\!\!\succ$ is well-founded, but is not a proof ordering in our sense, since, for instance, it is not closed under context application. However, not all properties of a proof ordering are required, since we consider only inconsistency proofs. The relation $\Rightarrow$ reflects all of the above inference rules:

LEMMA 7.5 (Reflection). *If $C \vdash C'$ and P is an inconsistency proof for an equation in C, then there exists an inconsistency proof P' for an equation in C', such that $P \Rightarrow^* P'$.*

PROOF. Suppose $C \vdash C'$ and let P be an inconsistency proof $s' \leftarrow_R^! s\sigma \leftrightarrow_{s \approx t}^{\Lambda} t\sigma \rightarrow_R^! t'$, where $s \simeq t$ is an equation in C.

If $s \approx t$ is an equation in C', let P' be P. On the other hand, suppose C' is $(C \setminus \{s \approx t\}) \cup \{u \approx t\}$, where $s \succ u$ and either $s \leftrightarrow_{R \cup L}^{+} u$, or else $s \leftrightarrow_{v \approx w}^{p} u$ where $v \approx w$ is an equation in C, such that $s \rhd v$ and $v \succ w$. Let u' be the normal form of $u\sigma$. Since $s\sigma \leftrightarrow_{s \approx t}^{\Lambda} t\sigma \succ\!\!\succ u\sigma \leftrightarrow_{u \approx t}^{\Lambda} t\sigma$, a suitable proof P' exists if $t' \neq u'$. Let us therefore assume that $u' = t'$, and hence $u' \neq s'$.

This case only arises if $s \leftrightarrow_{v \approx w}^{p} u$ for some equation $v \approx w$ in C, such that $s \rhd v$ and $v \succ w$. Let τ be a substitution, such that $s|_p = v\tau$ and $u|_p = w\tau$, and let v' and w' be the (ground) normal forms of $v\tau\sigma$ and $w\tau\sigma$, respectively. Note that $s' \neq u'$ implies $v' \neq w'$.

Let P' be the proof $v' \leftarrow_R^* v\tau\sigma \leftrightarrow_{v \approx w}^{\Lambda} w\tau\sigma \rightarrow_R^* w'$. Since $v \succ w$, the complexity of P' is $(\{v\tau\sigma\}, v, w\tau\sigma)$. The complexity of P is $(\{s\sigma\}, s, t\sigma)$, if $s \succ t$; $(\{t\sigma\}, t, \sigma)$, if $t \succ s$; and $(\{s\sigma, t\sigma\}, \bot, \bot)$, otherwise. Considering the different cases and using the fact that $v\tau\sigma$ is a subterm of $s\sigma$ and $s \rhd v$, we conclude that $P \succ\!\!\succ P'$. ∎

If a derivation from an initial inconsistent set of conjectures is "fair" (in a sense to be made precise), then eventually a verifiably inconsistent equation is generated. In this sense the inference system $\mathcal{P}$ may be called *refutationally complete*. If the inital set is consistent, all derived equations in $\bigcup_i C_i$ are inductive theorems of R. Thus if the derivation is finite one has successfully proved the initial conjectures.

[3]The subterm relation is not a rewrite relation. For example, x is a subterm of x^-, but $x + y$ is not a subterm of $x^- + y$.

For example, from the above rewrite system R and initial conjecture $x+y \approx y+x$ we can derive new equations $0+x \approx x$ and $x'+y \approx (x+y)'$. The set of all derived equations can be shown to be consistent (as we shall describe below). Thus all equations are inductive theorems of R.

We say that C' is a *cover set* for C (or C' *covers* C) with respect to R and $\succ$ if for every non-normal inconsistency proof P for an equation in C there exists an inconsistency proof P' for an equation in $C \cup C'$, such that $P \ggcurly P'$. (In other words, C' covers C if for every *minimal* non-normal inconsistency proof for C there is a simpler inconsistency proof for C'.) By this definition: the empty set covers any consistent set C; if C' covers C, then any superset of C' covers any subset of C; and if C_1' covers C_1 and C_2' covers C_2, then $C_1' \cup C_2'$ covers $C_1 \cup C_2$.

THEOREM 7.6. *If $C_0 \vdash C_1 \vdash \cdots$ is a (finite or infinite) derivation, such that the initial set C_0 is inconsistent and the set $\bigcup_i C_i$ of all deduced equations covers the set C_∞ of all persisting conjectures, then some set C_i is verifiably inconsistent.*

Fair derivations always exist:

PROPOSITION 7.7. *The set $CP_R(C)$ covers C.*

PROOF. A minimal non-normal inconsistency proof P for C has to contain a subproof $u \leftarrow_R s\sigma \leftrightarrow_C t\sigma$ or $u \leftarrow_R t\sigma \leftrightarrow_C s\sigma$. Thus there exists an inconsistency proof Q for $u \approx t\sigma$ or $u \approx s\sigma$, which is also an inconsistency proof for $CP_R(C)$. We have $P \ggcurly Q$, which completes the proof. □

Covering sets need not be based on critical pairs, though. In general, $s \approx t$ covers any equation $u[s]_p \approx u[t]_p$, where $p \neq \Lambda$. For whenever $u\sigma[s\sigma]_p \approx u\sigma[t\sigma]_p$ is an inconsistent ground instance, so is $s\sigma \approx t\sigma$, and moreover $u\sigma[s\sigma]_p \leftrightarrow^{\Lambda}_{u[s]\approx u[t]} u\sigma[t\sigma]_p \succ_{\mathcal{P}} s\sigma \leftrightarrow^{\Lambda}_{s\approx t} t\sigma$. In a similar vein, in theories with free constructors the set of equations $\{s_1 \approx t_1, \ldots, s_n \approx t_n\}$ can be shown to be a cover set for any equation $f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n)$ for which f is a constructor symbol (cf. [34]). For if f is a free constructor, then a ground instance $f(s_1\sigma, \ldots, s_n\sigma) \approx f(t_1\sigma, \ldots, t_n\sigma)$ is inconsistent if and only if one of the equations $s_i\sigma \approx t_i\sigma$ is inconsistent. Moreover, in such theories any equation $f(s_1, \ldots, s_m) \approx g(t_1, \ldots, t_n)$, where f and g are different constructors, is inconsistent.

Another useful technique for constructing cover sets is based on the following notion: We say that p is an *inductive position* in a term t (with respect to R) if $t|_p$ is not a variable and each ground term $t\sigma|_p$ for which σ is irreducible, is an instance of some left-hand side of R [24]. In other words, if p is an inductive position in t, then each ground instance $t\sigma$ can be rewritten either within the variable part of t or at position p. If p is an inductive position in a term s, then the set of all critical pairs obtained by superposing rules in R on the equation $s \approx t$ at position p in s covers $s \approx t$ (whenever $t \not\succeq s$). The notion of inductive positions can obviously be generalized to sets of positions. A set $\{p_1, \ldots, p_n\}$ of non-variable positions in a term t is said to be *inductive* if every ground instance $t\sigma$ is reducible by R at some position p_i, whenever σ is irreducible (cf. [44, 13]).

For example, in $x+(y+z)$ the subterm $y+z$ is rooted at an inductive position (the rightmost occurrence of $+$ or $\times$ is reducible by R, as we have pointed out above). Suppose now that we wish to prove that $x+(y+z) \approx (x+y)+z$ is an inductive theorem. We obtain a cover set for $x+(y+z) \approx (x+y)+z$ by superposing at the inductive position in the left-hand side, obtaining two critical pairs, $x+y \approx (x+y)+0$ and $x+(y+(z)') \approx (x+y)+(z)'$, the first of which can be simplified to a trivial equation $x+y \approx x+y$, while the second can be simplified to the equation $(x+(y+z))' \approx ((x+y)+z)'$, which is covered by $x+(y+z) \approx (x+y)+z$. That is, there is a finite fair derivation where no verifiably inconsistent equation has been derived; hence all derived equations are inductive theorems. The proof in this example in essence corresponds to an "induction on the variable z." As Fribourg [24] points out, choosing an inductive position in a term essentially corresponds to selecting an "induction schema" for a proof by induction.

Let R_0 be the rewrite system for addition and multiplication, and $\succ$ be the recursive path ordering for the operators $\times$, $+$, $'$, and 0 (listed in decreasing precedence). Furthermore let L_0 be the set of lemmas

$$\begin{array}{rcl} x+y & \approx & y+x \\ x+(y+z) & \approx & (x+y)+z \end{array}$$

We prove that the set of conjectures

$$\begin{array}{rcl} x\times(y+z) & \approx & x\times y+x\times z \\ x\times y & \approx & y\times x \\ x\times(y\times z) & \approx & (x\times y)\times z \end{array}$$

is consistent with R_0.

First observe that $x\times(y+z)\succ x\times y+x\times z$. Hence, we obtain a cover set for distributivity by superposing on the inductive position in its left-hand side. There are two critical pairs, $x\times y+x\times 0\approx x\times y$ and $x\times y+x\times z'\approx x\times(y+z)'$, the first of which can be simplified to a trivial equation $x\times y\approx x\times y$ and deleted. Since $x\times y+x\times z'\rightarrow_{R_0/L_0}(x\times y+x\times z)+x$ and $x\times(y+z)'\rightarrow_{R_0} x\times(y+z)+x\rightarrow_C (x\times y+x\times z)+x$, the second equation can also be simplified and deleted.

Superposition on (either side of) the commutativity axiom $x\times y\approx y\times x$ yields two critical pairs, $0\times x\approx 0$ and $y'\times x\approx x\times y+x$, both of can be used as rewrite rules. Computation of cover sets for these two rules results in four new equations, three of which can be simplified to trivial equations. The remaining equation, $y'\times x+y'\approx x'\times y+x'$, can be simplified by $\rightarrow^+_{R_0/L_0}$ to $(x\times y+(x+y))'\approx(y\times x+(x+y))'$, an equation which is already covered by the commutativity axiom.

Finally, to deal with the associativity axiom $x\times(y\times z)\approx(x\times y)\times z$, we superpose at the inductive position in its right-hand side and obtain a cover set of two equations, $x\times y\approx x\times(y\times 0)$ and $(x\times y)\times z+(x\times y)\approx x\times(y\times z')$. The first equation can be simplified to a trivial equation $x\times y\approx x\times y$; the second, to $(x\times y)\times z+(x\times y)\approx x\times(y\times z)+(x\times y)$. The latter equation is covered by the associativity axiom.

In conclusion, we have obtained a fair derivation without any verifiably inconsistent equation. Therefore all deduced equations, including the initial conjectures, are inductive theorems.

A characteristic of the proof by consistency method presented above is that new equations are deduced by superposing rules from the initial (fixed) rewrite system R on conjectures. Critical pairs between conjectures need not be considered. This also distinguishes the inference system $\mathcal{P}$ from *inductive completion* procedures, which are essentially standard completion procedures augmented by some inconsistency test and compute *all* critical pairs in $CP(R\cup C)$.

Musser [49] was the first to describe an inductive completion procedure. His procedure applies to abstract data type specifications, where an equality predicate *eq* is associated with each data type and the specification is "sufficiently complete," so that each ground expression $eq(s,t)$ can be reduced to the Boolean constant *true* or *false*. The equation *true* $\approx$ *false* indicates an inconsistency. Various improvements of the basic scheme have been suggested. The various approaches mainly differ in the respective notions of consistency they employ.

Huet and Hullot [34] studied the case of theories with (free) constructors, in which case an inconsistency is signified by an equation between two distinct ground terms built solely from constructor symbols. In contrast with Musser's method, an explicit axiomatization of equality is not required in this context.

Dershowitz [16] and Jouannaud and Kounalis [36] designed inductive completion procedures based on ground reducibility of terms. An inconsistency is signified by a rewrite rule $u \to v$ the left-hand side of which is not ground reducible. In a similar procedure [41], "test sets" are used to check for consistency.

It can be argued that inductive completion attempts to solve *all* possible induction schemes—and fails to terminate if one induction schema diverges, while the inference rules of $\mathcal{P}$ can be directed at one specific induction schema via the use of inductive positions. As a consequence, there are finite derivations from consistent sets of conjectures for which inductive completion fails, see [24]. On the other hand, it is also possible that inductive completion deduces equations that are useful for simplification but can not be obtained by the more restrictive deduction rule of $\mathcal{P}$.

Consider the following example [44]: Let R be the ground convergent rewrite system

$$\begin{array}{rcl} app(nil, x) & \to & x \\ app(cons(x, y), z) & \to & cons(x, app(y, z)) \\ rev(nil) & \to & nil \\ rev(cons(x, y)) & \to & app(rev(y), cons(x, nil)) \end{array}$$

and $rev(rev(x)) \approx x$ be a conjectured inductive theorem of R. The last rule in R can be superposed on the given conjecture, resulting in a critical pair

$$rev(app(rev(y), cons(x, nil))) \approx cons(x, y).$$

If we superpose the initial conjecture $rev(rev(x)) \approx x$ on this new conjecture, we obtain a critical pair

$$rev(app(y, cons(x, nil))) \approx cons(x, rev(y))$$

which can be oriented into a rule

$$rev(app(y, cons(x, nil))) \to cons(x, rev(y))$$

(with respect to a suitable lexicographic path ordering). The first critical pair can now be simplified and deleted. The remaining set of rewrite rules is ground convergent, which implies that $rev(rev(x)) \approx x$ is an inductive theorem of R.

On the other hand, with a linear deduction strategy the above rewrite rule cannot be deduced and an infinite derivation

$$\begin{array}{rcl} rev(app(rev(y), cons(x, nil))) & \approx & cons(x, y) \\ rev(app(app(rev(z), cons(y, nil)), cons(x, nil)) & \approx & cons(x, cons(y, z)) \\ & \cdots & \end{array}$$

may be produced.

The main problem with inductive completion is that, like standard completion, it must be supplied with an ordering on terms which is used to orient equations into rewrite rules and may fail if an equation is generated that cannot be oriented in the given ordering. An important advantage of our approach is the capability of handling unorientable equations and the flexibility in the choice of a reduction ordering. The inference system $\mathcal{P}$ also provides a complete method for disproving inductive theorems, and thus solves a problem posed by Jouannaud in [35]. Finally, let us point out that the ground convergence of the given theory R is only needed for the purpose of refutation completeness. The same techniques can also be applied to non-convergent sets of equations. Gramlich [29] describes a more refined proof by consistency system along lines similar to ordered completion. The relation of completion-based approaches to well-founded induction has been studied in [56].

8. Conclusion

We have represented completion and related rewrite methods as equational inference systems and have described techniques, based on proof reduction orderings, for reasoning about such proof systems. We have outlined the application of this approach to standard completion and refinements based on critical pair criteria, extended completion, ordered completion, and proof by consistency. An important class of procedures we have not discussed are completion procedures based on semantic unification. For a formalization of such procedures within the proof ordering framework, see [7].

The inference system *cum* proof complexity approach is by no means limited to purely equational theories. In [27, 9, 18], completion procedures for conditional equations (Horn clauses) have been described in this framework. Applications to first-order theorem proving (with or without equality) are described in [6].

The techniques underlying ordered completion have also been applied to other forms of unification of importance in theorem provers based on equational matings; see [26, 25]. For another interesting application, to unification in Boolean rings and Abelian groups, see [10]. Our approach was found to be of advantage in describing rewrite techniques for program synthesis [55]. Many of the rewrite techniques, which we have discussed in the context of equational theories, can also be applied in the more general setting of first-order clausal theorem proving, as described in [4, 5].

Acknowledgements

We thank Jieh Hsiang for thinking with us about these issues and Jean-Pierre Jouannaud for his enthusiasm and clarifying comments.

References

[1] S. Anantharaman and J. Hsiang. Automated proofs of the Moufang identities in alternative rings. *J. Automated Reasoning*, 6:79–109, 1990.

[2] L. Bachmair. *Proof methods for equational theories.* PhD thesis, University of Illinois, Urbana-Champaign, 1987.

[3] L. Bachmair and N. Dershowitz. Critical pair criteria for completion. *J. Symbolic Computation*, 6:1–18, 1988.

[4] L. Bachmair and H. Ganzinger. On restrictions of ordered paramodulation with simplification. In *Proc. 10th Int. Conf. on Automated Deduction*, Lect. Notes in Comput. Sci., pages 427–441, Berlin, 1990. Springer-Verlag.

[5] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. Technical Report MPI-I-91-208, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 1991. To appear in *Journal of Logic and Computation*.

[6] Leo Bachmair and Nachum Dershowitz. Inference rules for rewrite-based first-order theorem proving. In *Proceedings of the Second IEEE Symposium on Logic in Computer Science*, pages 331–337, Ithaca, NY, June 1987.

[7] Leo Bachmair and Nachum Dershowitz. Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 67(2 & 3), October 1989.

[8] Leo Bachmair, Nachum Dershowitz, and David A. Plaisted. Completion without failure. In H. Aït-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures 2: Rewriting Techniques*, chapter 1, pages 1–30. Academic Press, New York, 1989.

[9] H. Bertling and H. Ganzinger. Completion-time optimization of rewrite-time goal solving. In *Proc. 3rd Int Conf. Rewriting Techniques and Applications*, Lect. Notes in Comput. Sci., Berlin, 1989. Springer-Verlag.

[10] A. Boudet, J. P. Jouannaud, and M. Schmidt-Schauß. Unification in free extensions of Boolean rings and Abelian groups. In *Proc. Third Annual Symp. on Logic in Computer Science*, pages 121–130, Edinburgh, Scotland, 1988.

[11] T. Brown. *A structured design-method for specialized proof procedures.* PhD thesis, California Institute of Technology, Pasadena, 1975.

[12] B. Buchberger. A critical-pair/completion algorithm for finitely generated ideals in rings. In *Logic and Machines: Decision Problems and Complexity*, Lect. Notes in Comput. Sci., pages 137–161, Berlin, 1984. Springer-Verlag.

[13] R. Bündgen and Wolfgang Küchlin. Computing ground reducibility and inductively complete positions. In *Proc. 3rd Int Conf. Rewriting Techniques and Applications*, Lect. Notes in Comput. Sci., pages 59–75, Berlin, 1989. Springer-Verlag.

[14] George Butler and Dallas S. Lankford. Experiments with computer implementations of procedures which often derive decision algorithms for the word problem in abstract algebras. Memo MTP-7, Department of Mathematics, Louisiana Tech. University, Ruston, LA, August 1980.

[15] H. Comon. Solving inequations in term algebras. In *Proc. Fifth Annual Symp. on Logic in Computer Science*, pages 62–69, Philadelphia, Pennsylvania, 1990.

[16] Nachum Dershowitz. Applications of the Knuth-Bendix completion procedure. In *Proceedings of the Seminaire d'Informatique Theorique*, pages 95–111, Paris, France, December 1982.

[17] Nachum Dershowitz. Termination of rewriting. *J. of Symbolic Computation*, 3(1&2):69–115, February/April 1987. Corrigendum: *4*, 3 (December 1987), 409–410.

[18] Nachum Dershowitz. Ordering-based strategies for Horn clauses. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 118–124, Sydney, Australia, August 1991.

[19] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science B: Formal Methods and Semantics*, chapter 6, pages 243–320. North-Holland, Amsterdam, 1990.

[20] Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, August 1979.

[21] Nachum Dershowitz, Leo Marcus, and Andrzej Tarlecki. Existence, uniqueness, and construction of rewrite systems. *SIAM J. on Computing*, 17(4):629–639, August 1988.

[22] T. Evans. On multiplicative systems defined by generators and relations, I. *Proceedings of the Cambridge Philosophical Society*, 47:637–649, 1951.

[23] L. Fribourg. A superposition oriented theorem prover. *Theoretical Computer Science*, 35:129–164, 1985.

[24] L. Fribourg. A strong restriction of the inductive completion procedure. *J. Symbolic Computation*, 8:253–276, 1989.

[25] J. Gallier, W. Snyder, P. Narendran, and D. Plaisted. Rigid E-unification is NP-complete. In *Proc. Third Annual Symp. Logic in Computer Science*, pages 218–227, Edinburgh, Scotland, 1988.

[26] J. Gallier, W. Snyder, and S. Raatz. Rigid E-unification and its application to equational matings. In H. Aït-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures (Vol. 1: Algebraic techniques)*, pages 151–216. Boston, Academic Press, 1989.

[27] H. Ganzinger. A completion procedure for conditional equations. *J. Symbolic Computation*, 1988. To appear.

[28] J. A. Goguen. How to prove algebraic inductive hypotheses without induction. In W. Bibel and R. Kowalski, editors, *Proc. 5th Conf. Automated Deduction*, Lect. Notes in Comput. Sci., pages 356–373, Berlin, 1980. Springer-Verlag.

[29] B. Gramlich. Completion based inductive theorem proving: An abstract framework and its applications. In *Proc. ECAI-90*, pages 314–319, Stockholm, Sweden, 1990.

[30] Jieh Hsiang and Nachum Dershowitz. Rewrite methods for clausal and non-clausal theorem proving. In *Proceedings of the Tenth International Colloquium on Automata, Languages and Programming*, pages 331–346, Barcelona, Spain, July 1983. European Association of Theoretical Computer Science. Vol. 154 of *Lecture Notes in Computer Science*, Springer, Berlin.

[31] Jieh Hsiang and Michaël Rusinowitch. On word problems in equational theories. In T. Ottmann, editor, *Proceedings of the Fourteenth EATCS International Conference on Automata, Languages and Programming*, pages 54–71, Karlsruhe, West Germany, July 1987. Vol. 267 of *Lecture Notes in Computer Science*, Springer, Berlin.

[32] G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. of the Association for Computing Machinery*, 27:797–821, 1980.

[33] G. Huet. A complete proof of correctness of the Knuth and Bendix completion algorithm. *J. Computer and System Sciences*, 23:11–21, 1981.

[34] G. Huet and J.-M. Hullot. Proofs by induction in equational theories with constructors. *J. Computer and System Sciences*, 25:239–266, 1982.

[35] Jean-Pierre Jouannaud. A set of eleven important open problems in term rewriting based theorem proving. *Bulletin of the European Association for Theoretical Computer Science*, (31):272–273, February 1987.

[36] Jean-Pierre Jouannaud and Emmanuel Kounalis. Automatic proofs by induction in equational theories without constructors. *Information and Computation*, 81(1):1–33, 1989.

[37] Jean-Pierre Jouannaud and Mitsuhiro Okada. Satisfiability of systems of ordinal notations with the subterm property is decidable. In *Proceedings of the Eighteenth EATCS Colloquium on Automata, Languages and Programming*, pages 455–468, Madrid, Spain, July 1991. Vol. 510 in *Lecture Notes in Computer Science*, Springer, Berlin.

[38] D. Kapur, P. Narendran, D. J. Rosenkrantz, and H. Zhang. Sufficient-completeness, quasi-reducibility and their complexity. Tech. Rep., Dept. of Computer Science, SUNY at Albany, 1987.

[39] D. Kapur, P. Narendran, and H. Zhang. On sufficient-completeness and related properties of term rewriting systems. *Acta Informatica*, 24:395–415, 1987.

[40] Deepak Kapur, D. R. Musser, and P. Narendran. Only prime superpositions need be considered for the Knuth-Bendix procedure. *J. Symbolic Computation*, 4:19–36, August 1988.

[41] Deepak Kapur, Paliath Narendran, and Hantào Zhang. Automating inductionless induction using test sets. *J. Symbolic Computation*, 11:83–112, 1991.

[42] Donald E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, U. K., 1970. Reprinted in *Automation of Reasoning 2*, Springer, Berlin, pp. 342–376 (1983).

[43] W. Küchlin. A confluence criterion based on the generalised Newman lemma. In B. Caviness, editor, *Proc. Eurocal '85*, Lect. Notes in Comput. Sci., pages 390–399, Berlin, 1985. Springer-Verlag.

[44] W. Küchlin. Inductive completion by ground proof transformation. In H. Aït-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures (Vol. 2: Rewriting Techniques)*, pages 211–244. Boston, Academic Press, 1989.

[45] Wolfgang Küchlin. *Equational completion by proof transformation.* PhD thesis, Department of Mathematics, Swiss Federal Institute of Technology, Zurich, Switzerland, June 1986.

[46] D. Lankford. Canonical inference. Technical Report ATP-32, Dept. of Mathematics and Computer Science, University of Texas, Austin, 1975.

[47] D. Lankford. On proving term rewriting systems are Noetherian. Technical Report MTP-3, Mathematics Department, Louisiana Tech. Univ., Ruston, 1979.

[48] Yves Métivier. About the rewriting systems produced by the Knuth-Bendix completion algorithm. *Information Processing Letters*, 16(1):31–34, January 1983.

[49] D. R. Musser. On proving inductive properties of abstract data types. In *Proc. 7th ACM Symp. on Principles of Programming Languages*, pages 154–162, Las Vegas, Nevada, 1980.

[50] J. Mzali. *Methodes de filtrage equationnel et de preuve automatique de theoremes.* PhD thesis, Université de Nancy, 1986.

[51] A. Ohsuga and K. Sakai. Metis: A term rewriting system generator. Technical report, ICOT Research Center, Tokyo, Japan, 1986.

[52] G. E. Peterson. A technique for establishing completeness results in theorem proving with equality. *SIAM J. Comput.*, 12:82–100, 1983.

[53] Gerald E. Peterson and Mark E. Stickel. Complete sets of reductions for some equational theories. *J. of the Association for Computing Machinery*, 28(2):233–264, April 1981.

[54] D. A. Plaisted. Semantic confluence tests and completion methods. *Information and Computation*, 65:182–215, 1985.

[55] U. Reddy. Rewriting techniques for program synthesis. In *Proc. 3rd Int Conf. on Rewriting Techniques and Applications*, Lect. Notes in Comput. Sci., pages 388–403, Berlin, 1989. Springer-Verlag.

[56] U. Reddy. Term rewriting induction. In *Proc. 10th Int. Conf. on Automated Deduction*, Lect. Notes in Comput. Sci., pages 162–177, Berlin, 1990. Springer-Verlag.

[57] G. Robinson and L. Wos. Paramodulation and theorem-proving in first order theories with equality. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 135–150. Edinburgh University Press, Edinburgh, Scotland, 1969.

[58] F. Winkler and B. Buchberger. A criterion for eliminating unnecesssary reductions in the Knuth-Bendix algorithm. In *Proc. Coll. on Algebra, Combinatorics and Logic in Computer Science*, Györ, Hungary, 1983.

[59] H. Zhang and D. Kapur. Consider only general superpositions in completion procedures. In *Proc. 3rd Int Conf. on Rewriting Techniques and Applications*, Lect. Notes in Comput. Sci., pages 513–527, Berlin, 1989. Springer-Verlag.