PROOF METHODS FOR EQUATIONAL THEORIES

BY

LEO BACHMAIR

Dipl. Ing., Johannes Kepler Universität Linz, 1982
M.S., University of Illinois, 1985

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1987

Urbana, Illinois

ABSTRACT

In this thesis we study the application of rewrite techniques to equational reasoning. We present various rewrite-based proof methods and formalize them on an abstract level as equational inference systems. We also introduce techniques, based on the concept of proof orderings, for reasoning about such inference systems.

We describe the standard Knuth-Bendix completion method in our formalism and establish its correctness. Our correctness proofs are comparatively simple and apply to a large class of specific versions of completion. The notion of critical pair criterion can also be conveniently formalized in our framework.

We further discuss completion for rewriting modulo a congruence and present methods that are more general in scope than other completion procedures. We present, for instance, a completion procedure that can be applied to equational theories with infinite congruence classes; a case that can not be handled by any other method.

We also describe an extension of standard completion, completion without failure, that often succeeds in constructing a canonical system when standard completion fails. Unfailing completion is also a refutationally complete theorem prover for purely equational theories.

Finally, we describe some techniques for proving the termination of rewrite systems.

TO KARIN

ACKNOWLEDGEMENTS

I sincerely thank Nachum Dershowitz, my thesis advisor, for his consistent encouragement and guidance. Many of the ideas presented in this thesis originated from discussions with him.

Thanks are also due to David Plaisted for his valuable comments and suggestions. I am grateful to Steven Greenbaum, Jieh Hsiang, Alan Josephson, Yuh-Jeng Lee, and G. Sivakumar for their comments on various versions of parts of this thesis.

I would like to express my gratitude to Bruno Buchberger for his crucial advice and encouragement.

I would like to thank my wife and my parents for their patience and understanding.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1. Equations and Rewrite Rules

Equational reasoning methods have been applied to many problems in automated theorem proving, program verification, program synthesis, logic programming, and symbolic computation. Reasoning about equations may, for example, involve deciding whether an equation is a logical consequence of a given set of equations. Dealing effectively with the equality predicate within automated theorem provers is notoriously difficult. Simply adding equality axioms almost invariably leads to unacceptable inefficiencies. Instead, a number of special methods have been devised for reasoning about equality.

Within resolution-based provers, "paramodulation" (Robinson and Wos, 1969) is frequently employed. Paramodulation is a complete method for handling equations but is difficult to control and usually produces a large number of irrelevant or redundant formulas. More restrictive inference rules have been proposed, e.g. "demodulation" (Wos, et al., 1967), which consists of using equations in only one direction to rewrite terms to a "simpler" form. Demodulation is, in general, an incomplete, ad-hoc method. In this thesis we describe complete proof methods that are based on the concept of rewriting.

We consider purely *equational theories,* that is, theories that can be axiomatized by a set of equations. The *validity problem* in such theories is, of course,

semi-decidable: an equation $s = t$ is true in all models of a (countable) set of equations E, if t can be obtained from s by using the axioms of E to replace "equals by equals." However, systematically deriving new equations by substitution and replacement of equals by equals until $s = t$ has been proved is impractical for all but the simplest problems. The advantage of rewrite methods is that they provide some means of guidance and drastically limit the search space of equational consequences to be considered.

Rewrite systems are collections of directed equations (rules) used to compute by repeatedly substituting equal terms in a given expression until a simplest form possible (normal form) is obtained. Many formula manipulation systems, such as REDUCE or MACSYMA use equations in this manner. Canonical, i.e. terminating Church-Rosser, rewrite systems have the property that two terms are equivalent if and only if they simplify to an identical normal form. Deciding validity in theories for which canonical systems are known (e.g. group theory) is thus easy and reasonably efficient. A large number of canonical systems have been derived using the *Knuth-Bendix completion method* (Knuth and Bendix, 1970). Completion has been applied to a variety of problems including the word problem in universal algebra (Knuth and Bendix, 1970), proofs of inductive properties of data types (Musser, 1980; Huet and Hullot, 1982), and equational programming (O'Donnell, 1985; Dershowitz, 1985a).

Completion is supplied with a set of equations and a well-founded ordering on terms (a reduction ordering) and generates rewrite rules by orienting equations according to the given ordering. The only new equations that have to be derived are those obtained by paramodulating left-hand sides of rules into other left-hand sides. However, no paramodulations have to take place within the variable part of a rule. Mutual simplification of rules typically results in the deletion of redundant rules. The practicality and efficiency of completion crucially depend on this

use of simplification. On the other hand, the completeness of proof methods that contain such simplification mechanisms is difficult to establish, in general, and has required rather intricate arguments (e.g. Huet, 1981).

In the main part of this thesis we develop a formalism for reasoning about rewrite methods. We introduce the concept of *proof ordering* and demonstrate its usefulness for formalizing the essential properties of completion or similar proof methods. Proof orderings facilitate comparatively simple and intuitive proofs of correctness. In Chapter 2, we introduce proof ordering techniques by applying them to the standard Knuth-Bendix completion procedure. We formulate completion in an abstract framework and prove correctness of a large class of completion procedures, not just a specific version as in Huet (1981). We also utilize proof orderings in the design and verification of mechanisms—critical pair criteria—for sorting out redundant equations. Such criteria permit direct control over the number of equations generated and may considerably improve the efficiency of completion. The criteria presented here subsume those proposed by Winkler and Buchberger (1983), Küchlin (1985), and Kapur, et al. (1985).

Equational theories that can not be represented as canonical systems include, for instance, theories with commutativity. Such problematic axioms can often be built into the completion procedure itself, by employing more general matching and unification algorithms and using rules for *rewriting modulo a congruence.* Completion procedures for rewriting modulo a congruence were described by Lankford and Ballantyne (1977a, b, c), Peterson and Stickel (1981), Huet (1980), Jouannaud (1983), and Jouannaud and Kirchner (1986). In Chapter 3, we present new methods for rewriting modulo a congruence that subsume these procedures. We describe a method that allows construction of fully reduced canonical systems, which is not possible with other methods, in general. We also present a method that may be applied to equational theories with infinite

congruence classes, a case that can not be handled by any other completion procedure.

Completion must be supplied with a reduction ordering, a requirement that can not always be easily fulfilled. Unfortunately, even when an appropriate ordering is chosen, completion may fail to find any canonical system, though one exists (Dershowitz, et al., 1986). In Chapter 4, we address this problem by presenting an "unfailing" extension of completion. For a large class of orderings, including those most often used in practice (recursive path orderings and polynomial interpretations), this *unfailing completion* method is guaranteed to succeed in constructing a canonical system, whenever one exists. The method is refutationally complete for theorem proving in equational theories, but has the advantage over paramodulation that terms can always be kept in fully-simplified form and that fewer equational consequences need to be considered.

Finally, in Chapter 5, we study the problem of proving termination of rewrite systems and outline the use of *transformation* mappings in defining reduction orderings. In particular, we describe orderings for rewriting modulo a congruence, among them a class of orderings for associative-commutative rewriting.

1.2. Definitions

Most of the terminology we use is standard and we refer the reader to Huet (1980) for a more detailed exposition. The basic notions concerning equations, rewrite rules, and equational proofs are reviewed below.

Let $\mathbf{T}$ be the set of *terms* over some set of operator symbols F and some set of variables V. We use s, t, u, $\cdots$ to denote terms; f, g, h, $\cdots$ to denote operator symbols; and x, y, z, $\cdots$ to denote variables. We assume

that F contains at least one constant. Thus the set of *ground* terms, i.e. terms containing no variables, is non-empty. For example, if + is a binary operator, – a unary operator, and 0 and 1 are constants, then $0+(-x+y)$ is a non-ground term and $1+0$ is a ground term.

Let t be a term. A *subterm* of t is called *proper* if it is distinct from t. The expression t/p denotes the subterm of t at position p (positions may, for instance, be represented as sequences of integers). We write $s[t]$ to indicate that a term s contains t as a subterm and (ambiguously) denote by $s[u]$ the result of replacing a particular occurrence of t by u. If necessary, the position p of the replacement may be indicated by writing $s[p \leftarrow u]$. We will write $s[t_1, \ldots, t_n]$ if s contains subterms $t_1, \ldots, t_n$.

A binary relation $\rightarrow$ on T is *monotonic* (with respect to the term structure) if $s \rightarrow t$ implies $u[s] \rightarrow u[t]$, for all terms s, t and u. It is *stable* (under substitution) if $s \rightarrow t$ implies $s\sigma \rightarrow t\sigma$, for any substitution σ of terms in T for variables in s and t. The symbols $\rightarrow^+$, $\rightarrow^*$ and $\leftrightarrow$ denote the transitive, transitive-reflexive, and symmetric closure of $\rightarrow$, respectively. The inverse of $\rightarrow$ is denoted by $\leftarrow$. We call $\rightarrow$ a (strict partial) *ordering* if it is irreflexive and transitive. A relation $\rightarrow$ is *Noetherian* if there is no infinite sequence $t_1 \rightarrow t_2 \rightarrow t_3 \cdots$. A transitive Noetherian relation is called *well-founded.* A *reduction ordering* is a well-founded ordering that is stable and monotonic.

An *equation* is a pair (s, t), written $s = t$, where s and t are terms. For any set of equations E, $\leftrightarrow_E$ denotes the smallest symmetric relation that contains E and is stable and monotonic. That is, $s \leftrightarrow_E t$ if and only if, for some term c and some substitution σ, $s = c[u\sigma]$ and $t = c[v\sigma]$, where $u \doteq v$ is in E ($u \doteq v$ denotes, ambiguously, $u = v$ or $v = u$). The relation $\leftrightarrow_E^*$ is the smallest stable congruence that contains E; a congruence is, by definition, monotonic.

Directed equations are also called *rewrite rules* and are written $s \to t$. A *rewrite system* is any set R of rewrite rules, in which all variables appearing on a right-hand side also appear on the corresponding left-hand side. The *reduction relation* $\to_R$ is the smallest stable and monotonic relation that contains R. That is, $s \to_R t$ (s *reduces* or *rewrites to* t) if and only if $s = c[l\sigma]$ and $t = c[r\sigma]$, for some rewrite rule $l \to r$ in R, term c, and substitution σ. A term t is *irreducible* in R if there is no term u such that $t \to_R u$. An irreducible term t' is called a *normal form* of t in R if $t \to_R^* t'$.

We use the notation $s \downarrow_R t$ to indicate that there exists a term u such that $s \to_R^* u \leftarrow_R^* t$. A rewrite system R is *Church-Rosser* if $s \leftrightarrow_R^* t$ implies $s \downarrow_R t$, for all terms s and t. This property is equivalent to *confluence:* $s \leftarrow_R^* u \to_R^* t$ implies $s \downarrow_R t$, for all terms s, t, and u. A rewrite system R *terminates* if $\to_R$ is Noetherian. Thus, a rewrite system terminates if and only if it is contained in some reduction ordering. A rewrite system is called *canonical* if it is terminating and Church-Rosser. In a canonical system every term has a unique normal form. A rewrite system R is *reduced* if, for every rule $l \to r$ in R, r is irreducible in R and l is irreducible in $R - \{l \to r\}$. Reduced canonical systems are unique: if R and R' are both canonical and reduced and define the same set of irreducible terms, then they are the same up to renaming of variables.

Let E be a set of equations and R be a rewrite system. A *proof* of $s = t$ in $E \cup R$ (or a proof of $s \leftrightarrow_{E \cup R}^* t$) is a sequence $P = (s_0, \ldots, s_n)$, such that s_0 is s, s_n is t and, for $0 < i \leq n$, one of $s_{i-1} \leftrightarrow_E s_i$, $s_{i-1} \to_R s_i$, or $s_{i-1} \leftarrow_R s_i$ holds. Every single proof step (s_{i-1}, s_i) has to be *justified* by an equation $u_i = v_i$, a substitution σ_i, and a position p_i, such that $s_{i-1}/p_i = u_i\sigma_i$, $s_i = s_{i-1}[v_i\sigma_i]$, and $u_i \doteq v_i$ is in $E \cup R$. We say, for instance, that a proof step (s_{i-1}, s_i) applies at position p_i. The justification may be (partially) indicated by writing the proof

as, for instance, $s_0 \leftrightarrow_E s_1 \rightarrow_R \cdots \leftarrow_R s_n$, etc. A proof step $s \leftrightarrow_E t$ is called an *equality step*; a step $s \rightarrow_R t$ a *rewrite step*. A proof of the form $s \leftarrow_R u \rightarrow_R t$ is called a *peak*; etc. Usually, we abbreviate a proof of the form $s_0 \rightarrow_R \cdots \rightarrow_R s_n$ by $s_0 \rightarrow_R^* s_n$. A proof $s_0 \rightarrow_R^* s_k \leftarrow_R^* s_n$ is called a *rewrite proof*. By definition, $P = (s)$ is a proof of $s = s$. A *subproof* of P is any proof $(s_i, \ldots, s_j)$, where $0 \leq i \leq j \leq n$. The notation $P[P']$ indicates that P contains P' as a subproof.

For example, if E contains the commutativity axiom $x+y = y+x$ and the associativity axiom $x+(y+z) = (x+y)+z$, and R contains rules $x+0 \rightarrow x$ and $x+(-x) \rightarrow 0$, then the following is a proof of $x+(-x+y) = y$ in $E \cup R$:

$$x+(-x+y) \leftrightarrow_E (x+(-x))+y \rightarrow_R 0+y \leftrightarrow_E y+0 \rightarrow_R y.$$

A *proof pattern* in $E \cup R$ is a schema for a class of proofs; it describes proofs that share a common structure. For example, the pattern $s \rightarrow_R t$, where s and t are metavariables denoting arbitrary terms and R denotes an arbitrary rewrite system, characterizes all single step rewrite proofs in R; $s \rightarrow_R^* u \leftarrow_R^* t$ describes all rewrite proofs in R; $s \leftarrow_R u \rightarrow_R t$, all peaks, etc. An *instance* of a pattern is any specific proof of the given structure.

CHAPTER 2

THE KNUTH-BENDIX COMPLETION METHOD

In this chapter we describe the Knuth-Bendix completion method (Knuth and Bendix, 1970) and introduce a new technique—orderings on equational proofs—for proving the correctness of completion. We also utilize proof orderings for designing criteria for completion that may be used to reduce the number of equational consequences that have to be generated.

2.1. Standard Completion

We first present an abstract formulation of the *Knuth-Bendix completion method* for constructing a canonical rewrite system R for a given set of equations E. If R is finite and canonical, and the congruence relations $\leftrightarrow_E^*$ and $\leftrightarrow_R^*$ are the same, then R may be used as a *decision procedure* for the *validity problem* in E, since then two terms s and t are equivalent in E if and only if they reduce to identical normal forms in R. In particular, canonical systems may be used for solving word problems in equational theories. The unsolvability of the word problem for certain (finitely-based) equational theories implies that the construction of a canonical system R is not always possible. Thus, a completion procedure may terminate either with success or failure, or it may not terminate and instead compute successive approximations R_n of an infinite canonical system R. The latter case can provide a *semi-decision procedure* for the validity problem in E (as pointed out in Huet, 1981).

We will formalize the notion of completion within the framework of an equational inference system. Since we distinguish between equations and rewrite rules, the objects of this inference system are pairs (E,R), where E is a set of equations and R is a set of rules. The main steps of completion are (a) turning equations into rules, (b) generating equational consequences from rules, and (c) simplifying equations.

Let $>$ be a reduction ordering on terms. The system **BC** (*basic completion*) consists of the following *inference rules*, where R is any rewrite system contained in $>$:

C1) Orienting an equation.

$$\frac{(E \cup \{s \doteq t\},R)}{(E,R \cup \{s \to t\})} \qquad \text{if } s > t$$

C2) Adding an equational consequence.

$$\frac{(E,R)}{(E \cup \{s = t\},R)} \qquad \text{if } s \leftarrow_R u \to_R t$$

C3) Simplifying an equation.

$$\frac{(E \cup \{s \doteq t\},R)}{(E \cup \{u \doteq t\},R)} \qquad \text{if } s \to_R u$$

C4) Deleting a trivial equation.

$$\frac{(E \cup \{s = s\},R)}{(E,R)}$$

We write $(E,R) \underset{\mathrm{I}}{\vdash} (E',R')$ if (E',R') can be obtained from (E,R) by (one or more) applications of inference rules of the inference system **I**. A (possibly infinite) sequence $(E_0,R_0), (E_1,R_1), \cdots$ is called a *derivation* in **I** if $(E_{i-1},R_{i-1}) \underset{\mathrm{I}}{\vdash} (E_i,R_i)$, for all $i > 0$. The *limit* of a derivation is the pair (E^∞,R^∞), where E^∞ is the set of all *persisting equations*, i.e. the set

$\cup_{i \geq 0} \cap_{j \geq i} E_j$, and R^∞ is the set of all *persisting rules,* i.e. the set $\cup_{i \geq 0} \cap_{j \geq i} R_j$.

Basic completion is *sound*:

Lemma 2.1. *If* $(E,R) \underset{\mathbf{BC}}{\vdash} (E',R')$, *then the congruence relations* $\overset{*}{\leftrightarrow}_{E \cup R}$ *and* $\overset{*}{\leftrightarrow}_{E' \cup R'}$ *are the same.*

In addition, we have

Lemma 2.2. *If* $(E,R) \underset{\mathbf{BC}}{\vdash} (E',R')$ *and the reduction ordering* $>$ *contains* R, *then* $>$ *also contains* R'.

Consequently, if $(E_0,R_0), (E_1,R_1), \cdots$ is a derivation in **BC**, then all systems R_i, for $i \geq 0$, as well as the limit R^∞, are contained in $>$ and therefore terminate.

We are interested in derivations $(E_0,R_0), (E_1,R_1), \cdots$ for which the limit R^∞ is canonical. If the equational theory of $E_0 \cup R_0$ can be represented by a canonical system R^∞, then there is, for any valid equation in $E_0 \cup R_0$ a rewrite proof in R^∞. In general, a rewrite proof in $E \cup R$ can be characterized as a proof that contains no "critical" subproof of the form $s \leftrightarrow_E t$ or $s \leftarrow_R u \rightarrow_R t$. We denote the set of these two patterns by $\mathrm{N_C}$. Application of an inference rule of completion has the effect that certain instances of critical proof patterns can be replaced by "simpler" proofs. Thus the application of inference rules is reflected on the proof level by a reduction relation on proofs. We will formalize this aspect of completion below.

A relation $\Rightarrow$ on proofs is *monotonic* if $P \Rightarrow P'$ implies $Q[P] \Rightarrow Q[P']$, for all proofs P, P', and Q. It is *stable* if

$$P = (s, \ldots, u_i, \ldots, t) \Rightarrow (s, \ldots, v_j, \ldots, t) = P'$$

implies

$$(c\,[s\,\sigma], \ldots, c\,[u_i\,\sigma], \ldots, c\,[t\,\sigma]) \Rightarrow (c\,[s\,\sigma], \ldots, c\,[v_j\,\sigma], \ldots, c\,[t\,\sigma]),$$

for all proofs P and P', terms c, and substitutions σ. A stable and monotonic ordering on proofs is called a *proof (reduction) ordering* if it is well-founded.

An *elimination pattern* is a pair of proof patterns. If S is a set of elimination patterns, then $\Rightarrow_S$ denotes the smallest stable and monotonic ordering on proofs that contains any instance $P \Rightarrow P'$ of an elimination pattern of S. (In other words, $\Rightarrow_S$ is a rewrite relation on proofs.) We say that a set S of elimination patterns is *compatible* with an inference system I if $(E,R) \vdash_I (E',R')$ implies that for every proof P in $E \cup R$ there is a proof P' in $E' \cup R'$, such that $P \Rightarrow_S^* P'$.

For basic completion we have the following set S_{BC} of elimination patterns, where R and R' are contained in the given reduction ordering $>$ (see Figure 2.1):

$$\begin{array}{rcl} s \leftrightarrow_E t & \Rightarrow & s \rightarrow_{R'} t \\ s \leftrightarrow_E t & \Rightarrow & s \rightarrow_{R'} u \leftrightarrow_{E'} t \\ s \leftrightarrow_E s & \Rightarrow & s \\ s \leftarrow_R u \rightarrow_R t & \Rightarrow & s \leftrightarrow_{E'} t \\ s \leftarrow_R u \rightarrow_R t & \Rightarrow & s \rightarrow_{R'}^* v \leftarrow_{R'}^* t \end{array}$$

The first three patterns are called *equality patterns*, the remaining two, *overlap patterns*. They may be used to eliminate equality steps and peaks, respectively. The corresponding proof relation is denoted by $\Rightarrow_{BC}$.

Lemma 2.3. *The set* S_{BC} *is compatible with basic completion.*

Proof. It can easily be seen that the inference rules C1, C3, and C4 correspond to the equality patterns; inference rule C2 corresponds to the first overlap pattern. ∎

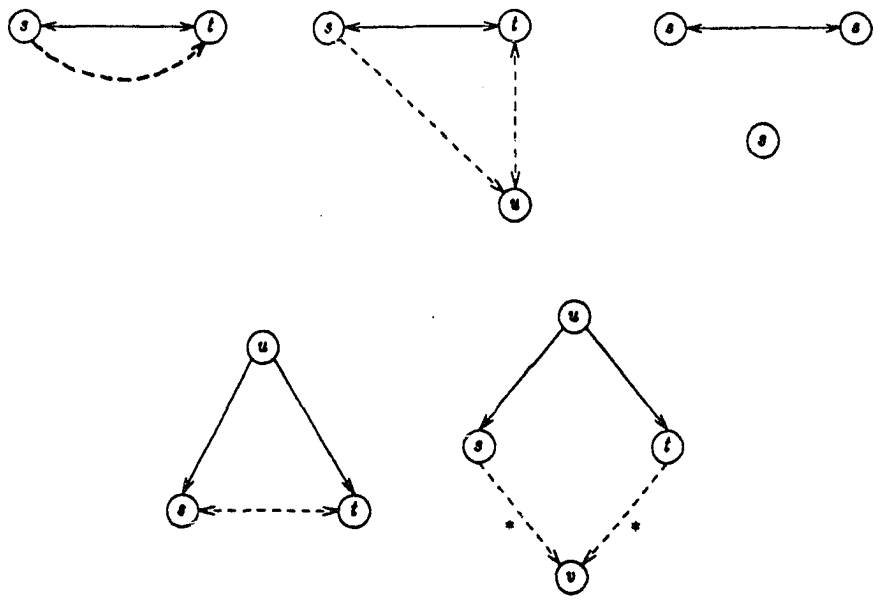


Figure 2.1. *Equality and overlap patterns*

Example 2.1. Let E be the following set of equations characterizing *group theory:*

$$\begin{array}{rcl} 0+x & = & x \\ -x+x & = & 0 \\ (x+y)+z & = & x+(y+z) \end{array}$$

The equation $--x+0=x$ is valid in this theory as the following proof P_0 shows:

$$\begin{array}{lll} --x+0 & \leftrightarrow_E & --x+(-x+x) \\ & \leftrightarrow_E & (--x+-x)+x \\ & \leftrightarrow_E & 0+x \\ & \leftrightarrow_E & x \end{array}$$

If we repeatedly apply inference rule C1 to orient the equations in E, then we obtain the following rewrite system R_1:

$$\begin{array}{rcl} 0+x & \to & x \\ -x+x & \to & 0 \\ (x+y)+z & \to & x+(y+z) \end{array}$$

The corresponding set E_1 is empty. We orient equations by comparing terms by their size. Terms of the same size are ordered by lexicographically comparing their (top-level) subterms from left to right. To ensure stability under substitution, we restrict $s > t$ so that no variable appears more often in t than in s. Replacing all equality steps in P_0 by rewrite steps, we get the following proof P_1:

$$\begin{array}{lll} --x+0 & \leftarrow_{R_1} & --x+(-x+x) \\ & \leftarrow_{R_1} & (--x+-x)+x \\ & \to_{R_1} & 0+x \\ & \to_{R_1} & x \end{array}$$

The proof P_1 is simpler than P_0 in the sense that $P_0 \Rightarrow_C P_1$. It is not a rewrite proof, since it contains a peak

$$--x+(-x+x) \leftarrow_{R_1} (--x+-x)+x \to_{R_1} 0+x,$$

which is an instance of

$$-x+(x+y) \leftarrow_{R_1} (-x+x)+y \to_{R_1} 0+y.$$

The latter peak reflects an *overlap* between the third and second rule of R_1. By inference rule C2, we may add the equation $-x+(x+y)=0+y$ to E_1, obtaining a new set E_2 and a proof P_2:

$$\begin{array}{lll} --x+0 & \leftarrow_{R_2} & --x+(-x+x) \\ & \leftrightarrow_{E_2} & 0+x \\ & \to_{R_2} & x \end{array}$$

where R_2 is R_1. Again, we have $P_1 \Rightarrow_C P_2$. By inference rule C3, the equation $-x+(x+y)=0+y$ can be simplified to $-x+(x+y)=y$, which is reflected on the proof level by a reduction $P_2 \Rightarrow_C P_3$, where P_3 is

$$\begin{array}{lll} --x+0 & \leftarrow_{R_3} & --x+(-x+x) \\ & \leftrightarrow_{E_3} & x \\ & \leftarrow_{R_3} & 0+x \\ & \rightarrow_{R_3} & x \end{array}$$

and R_3 is R_2. This proof contains a "trivial peak"

$$x \leftarrow_{R_3} 0+x \rightarrow_{R_3} x$$

which can be eliminated by first adding the equation $x=x$, using C2, and then deleting it, using C4. Thus E_3 and R_3 do not change, but we get a simpler proof

$$\begin{array}{lll} --x+0 & \leftarrow_{R_3} & --x+(-x+x) \\ & \leftrightarrow_{E_3} & x \end{array}$$

The equation $-x+(x+y)=y$, may be oriented into a rule $-x+(x+y)\rightarrow y$, resulting in a pair (E_4,R_4), where E_4 is empty, and R_4 is R_3 plus the new rule above. This leads to a proof

$$\begin{array}{lll} --x+0 & \leftarrow_{R_4} & --x+(-x+x) \\ & \rightarrow_{R_4} & x \end{array}$$

which is an instance of the peak

$$--x+0 \leftarrow_{R_4} -x+(x+y) \rightarrow_{R_4} x.$$

By inference rule C2, we can now generate $--x+0=x$, and, by C1, turn it into a rule $--x+0\rightarrow x$. We now have derived a rewrite system R:

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

that contains enough rules to prove $--x+0=x$ by simple rewriting. This system is not canonical, however. For example, there is no rewrite proof of $-0+y=y$, even though the equation is provable:

$$-0+y \leftarrow_R -0+(0+y) \rightarrow_R y.$$

The following canonical and reduced system for group theory was derived by Knuth and Bendix (1970) with a more powerful completion method that is described below:

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

Proof orderings are the key to our approach of studying completion methods. The proof orderings we design are based on the given reduction ordering and employ information contained in the justification of a proof. The concept of *multiset orderings* is of particular importance in this context. A *multiset* is an unordered collection of elements in which elements may appear more than once. If $>$ is a partial ordering on a set S, then the corresponding multiset ordering $\gg$ on the set of all finite multisets of elements in S is the smallest transitive relation such that

$M \cup \{x\} \gg M \cup \{y_1, \ldots, y_n\}$, whenever $n \geq 0$ and $x > y_i$, for $1 \leq i \leq n$.

According to this ordering an element of a multiset can be replaced by any finite number of elements that are smaller in $>$.

PROPOSITION 2.1. (Dershowitz and Manna, 1979) *The multiset ordering $\gg$ is well-founded if and only if $>$ is well-founded.*

We specify a proof ordering by a complexity measure c of single proof steps and a corresponding ordering $>^c$. The *complexity* $M(P)$ of a proof $P = (s_0, \ldots, s_n)$ is the multiset $\{c_1, \ldots, c_n\}$, where $c_i = c(s_{i-1}, s_i)$ is the

complexity of the i-th proof step in P. The ordering $>_I$ corresponding to c and $>^c$ is defined by: $P >_I P'$ *if and only if* $M(P) \gg^c M(P')$, where $\gg^c$ is the multiset ordering corresponding to $>^c$. The ordering $>_I$ is monotonic, by definition. It is stable if $c(s,t) >^c c(s',t')$ implies $c(u[s\sigma],u[t\sigma]) >^c c(u[s'\sigma],u[t'\sigma])$, for any proof step $s \leftrightarrow_{E \cup R} t$, term u, and substitution σ. By Proposition 2.1, $>_I$ is well-founded if and only if $>^c$ is well-founded.

The ordering $>_{BC}$ is defined by the following complexity measure c_{BC} and ordering $>^c_{BC}$:

if $s \rightarrow_R t$, then $c_{BC}(s,t)$ is $\{s\}$;

if $s \leftarrow_R t$, then $c_{BC}(s,t)$ is $\{t\}$;

if $s \leftrightarrow_E t$, then $c_{BC}(s,t)$ is $\{s,t\}$;

the ordering $>^c_{BC}$ is the multiset ordering $\gg$ corresponding to the given reduction ordering $>$. The ordering $>_{BC}$ is well-founded, since the reduction ordering $>$ is. In addition, the monotonicity and stability (under substitution) of $>$ imply stability of $>_{BC}$. Therefore, $>_{BC}$ is a proof ordering.

Lemma 2.4. *The relation* $\Rightarrow_{BC}$ *is a proof ordering.*

Proof. Since $>_{BC}$ is a proof ordering, it suffices to show that $>_{BC}$ contains any instance of an elimination pattern for basic completion.

a) $(s \leftrightarrow_E t) >_{BC} (s \rightarrow_{R'} t)$, since $\{s,t\} \gg \{s\}$;

b) $(s \leftrightarrow_E t) >_{BC} (s \rightarrow_{R'} u \leftrightarrow_{E'} t)$,
since $\{s,t\} \gg \{s\}$ and $\{s,t\} \gg \{u,t\}$.

c) $(s \leftrightarrow_E s) >_{BC} (s)$, since $\{\{s,s\}\} \gg \emptyset$;

d) $(s \leftarrow_R u \rightarrow_R t) >_{BC} (s \leftrightarrow_{E'} t)$, since $\{u\} \gg \{s,t\}$;

e) $(s \leftarrow_R u \rightarrow_R t) >_{BC} (s \rightarrow^*_{R'} v \leftarrow^*_{R'} t) = P'$,
since $\{u\} \gg \{w\}$, for any term w in P'. •

Let N be a set of proof patterns and S be a set of elimination patterns compatible with an inference system I.

Definition 2.1. A derivation $(E_0,R_0), (E_1,R_1), \cdots$ in I is *fair relative to* S and N if, for any proof P in $E_i \cup R_i$ that contains an instance of a pattern in N, there is a proof P' in $E_j \cup R_j$, for some $j \geq i$, such that $P \Rightarrow_S P'$.

THEOREM 2.1. *Let* I *be an inference system,* N *be a set of proof patterns, and* S *be a set of elimination patterns, such that* S *is compatible with* I *and* $\Rightarrow_S$ *is a proof ordering. If a derivation* $(E_0,R_0), (E_1,R_1), \cdots$ *in* I *is fair relative to* S *and* N, *then there is, for any proof* P *in* $E_i \cup R_i$, $i \geq 0$, *a proof* P' *in* $E^\infty \cup R^\infty$ *such that* $P \Rightarrow_S^* P'$ *and* P' *contains no instance of a pattern in* N.

Proof. By induction on $\Rightarrow_S$. Let P be a proof in $E_i \cup R_i$. If P is not a proof in $E^\infty \cup R^\infty$, then it must use some non-persisting equation or rule. Hence, by compatibility of S with I, there is a proof Q in $E_j \cup R_j$, for some $j \geq i$, such that $P \Rightarrow_S Q$. On the other hand, if P is a proof in $E^\infty \cup R^\infty$, but contains an instance of a proof pattern in N, then, by fairness, there is a proof Q in $E_j \cup R_j$, such that $P \Rightarrow_S Q$. The induction hypothesis may then be applied to Q to yield the desired conclusion. •

Before we apply Theorem 2.1 to completion, we extend the basic inference system **BC** by introducing additional inference rules for simplification of rewrite rules. These rules are essential for efficiency and also allow the construction of reduced systems, which is not possible in general with basic completion. The inference system **C** (*standard completion*) consists of the inference rules in **BC** plus the following *simplification rules*:

S1) Simplifying the right-hand side of a rewrite rule

$$\frac{(E, R \cup \{s \to t\})}{(E, R \cup \{s \to u\})} \quad \text{if } t \to_R u$$

S2) Simplifying the left-hand side of a rewrite rule

$$\frac{(E, R \cup \{s \to t\})}{(E \cup \{u = t\}, R)} \quad \text{if } s \to_R u \text{ at a position } p \text{ not at the top,}$$

$$\frac{(E, R \cup \{s \to t\})}{(E \cup \{u = t\}, R)} \quad \text{if } s \to_R u \text{ by } l \to r \text{ and } s \rhd l.$$

The symbol $\rhd$ denotes the subsumption ordering: $s \rhd l$ *if and only if* s *is a proper instance of* l. For example, $f(z, g(z))$ and $f(z, z)$ are proper instances of $f(x, y)$, but $f(x, z)$ is not.

Both Lemma 2.1 (soundness) and Lemma 2.2 generalize to standard completion. Let S_C be the set S_{BC} plus the two additional patterns

$$s \to_R t \quad \Rightarrow \quad s \to_{R'} u \leftarrow_{R'} t$$
$$s \to_R t \quad \Rightarrow \quad s \to_{R'} v \leftrightarrow_{E'} t$$

where R and R' are contained in the given reduction ordering $>$, $s \to t$ is a rule in R, $s \to u$ is a rule in R', and $s \to_{R'} v$ is by application of a rule $l \to r$ such that s either is a proper instance of l or contains an instance of l (see Figure 2.2). The corresponding proof relation is denoted by $\Rightarrow_C$. The patterns above are called *simplification patterns*. The following lemma can easily be derived from the definition above.

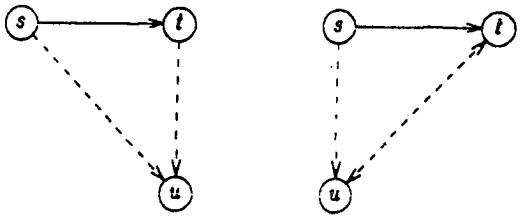


Figure 2.2. *Simplification patterns*

Lemma 2.5. *The set of elimination patterns* S_C *is compatible with standard completion.*

In order to prove that $\Rightarrow_C$ is a proof ordering, we introduce an ordering $>_C$ using the following complexity measure c and ordering $>^c$:

if $s \rightarrow_R t$ by $l \rightarrow r$ at position p, then $c(s,t)$ is $(\{s\},s/p,l,t)$;

if $s \leftarrow_R t$ by $l \rightarrow r$ at position p, then $c(s,t)$ is $(\{t\},t/p,l,s)$;

if $s \leftrightarrow_E t$, then $c(s,t)$ is $(\{s,t\},-,-,-)$

Only the first component is relevant in the last case. The ordering $>^c$ is the lexicographic combination of the multiset extension of the reduction ordering $>$, the proper subterm ordering, the proper subsumption ordering $\rhd$, and the reduction ordering $>$. The complexity measure c extends the complexity measure c_{BC} for basic completion. The inference rules for simplification are reflected by the additional components of $c(s,t)$. Both the reduction ordering $>$ and the proper subterm ordering are well-founded, monotonic, and stable (under substitution); the subsumption ordering is well-founded. From this one can readily infer that $>_C$ is a proof ordering.

Lemma 2.6. *The relation* $\Rightarrow_C$ *is a proof ordering.*

Proof. We show that $>_C$ contains $\Rightarrow_C$. By the same arguments as in the proof of Lemma 2.4, it can be shown that $>_C$ contains any instance of an equality or a overlap pattern. For simplification patterns we have:

a) $\{(\{s\},s,s,t)\} \gg^c \{(\{s\},s,s,u),(\{t\},t/p,l,u)\}$,

since $s > t$ and $t > u$;

b) $\{(\{s\},s,s,t)\} \gg^c \{(\{s\},s/p,l,v),(\{t,u\},-,-,-)\}$,

since $s > t$, $s > u$, and either s/p is a strict subterm of s or $s \rhd l$.

•

Remark. Inference rule S2 can not be used to simplify the left-hand side of a rule $l \to r$ by another rule $l \to r'$. This is no restriction in practice, since such simplifications are not necessary if equations are kept simplified. They may be permitted if the "age" of a rule, i.e. when it was generated, is included in the complexity measure of proof steps.

We will next derive sufficient conditions for the fairness of a derivation. Let us consider the possibilities for eliminating equality steps $s \leftrightarrow_E t$ and peaks $s \leftarrow_R u \to_R t$. Equality steps can be eliminated as a result of orienting, simplifying, or deleting an equation. To eliminate peaks it suffices to generate certain equational consequences called *critical pairs.*

Let $s \to t$ and $l \to r$ be rules in R with no variables in common (the variables of one rule are renamed if necessary) and suppose that, for some position p, s/p is not a variable and is unifiable with l, σ being the most general unifier, i.e. $s\sigma/p = l\sigma$. Then the *superposition* of $l \to r$ on $s \to t$ at position p determines a *critical pair* $c = d$, where c is $t\sigma$ and d is $s\sigma[p \leftarrow r\sigma]$. The proof $c \leftarrow_R s\sigma \to_R d$ is called a *critical (rule) overlap*; the term $s\sigma$, the *overlapped term*; the position p, the *critical pair position.*

Lemma 2.7. (Critical Pair Lemma, Knuth and Bendix, 1970; see also Huet, 1980) *If $s \leftarrow_R u \to_R t$, then either $s \downarrow_R t$ or, for some critical pair $c = d$ between rules in R, $s = v[c\sigma]$ and $t = v[d\sigma]$.*

Sketch of proof. Let P be an overlap $s \leftarrow_R u \to_R t$. If s and t are identical, then obviously $s \downarrow_R t$. Let us assume that s and t are distinct. We distinguish three cases.

a) Suppose both proof steps apply at disjoint positions, i.e. they do *not overlap.* Then P can be written as

$$s = u[r, l'] \leftarrow_R u[l, l'] \to_R u[l, r'] = t,$$

where $l \to_R r$ and $l' \to_R r'$, and we have (see Figure 2.3):

$$s = u[r, l'] \to_R u[r, r'] \leftarrow_R u[l, r'] = t.$$

b) We speak of a *variable overlap* if one proof step applies in the variable part of the other step. Then P may be written as

$$s = u[r[l', \ldots, l']] \leftarrow_R u[l[l', \ldots, l']] \to_R u[l[r', l', \ldots, l']] = t$$

and can be replaced by (see Figure 2.4):

$$s = u[r[l', \ldots, l']] \to_R^* u[r[r', \ldots, r']]$$
$$\leftarrow_R u[l[r', \ldots, r']] \leftarrow_R^* u[l[r', l', \ldots, l']] = t.$$

c) The only remaining possibility is a *critical overlap*, where one proof step applies below the other, but not in the variable part. In this case $s = t$ contains an instance of a critical pair. • The considerations above lead to

Lemma 2.8. *A derivation* $(E_0, R_0), (E_1, R_1), \cdots$ *in* C *is fair relative to* S_C *and* N_C *if (a)* $E^\infty = \emptyset$ *and (b) all critical pairs between rules in* R^∞ *are contained in* $\cup_k E_k$.

Proof. Let $(E_0, R_0), (E_1, R_1), \cdots$ be a derivation satisfying (a) and (b), and P be a proof in $E_i \cup R_i$, for some $i \geq 0$. Suppose that P is not a rewrite proof.

If P contains an equality step $s \leftrightarrow_{E_i} t$, where an equation $u = v$ is used, then, by property (a), the equation $u = v$ will eventually be formed into a rewrite rule, simplified, or deleted (C1, C3, C4). By Lemma 2.3, there is a proof P' in $E_j \cup R_j$, for some $j > i$, such that $P \Rightarrow_C P'$. Likewise, if P contains a rewrite step $s \to_{R_i} t$, where a non-persisting rule is used, then simplification of this rule by S1 or S2 will result in a proof P', such that $P \Rightarrow_C P'$.

If P is a proof in R^∞, i.e. uses only persisting rules, then it must contain a peak $Q = (u \leftarrow_{R_i} v \to_{R_i} w)$. By the Critical Pair Lemma, if Q is not a critical overlap, then there is a rewrite proof $Q' = (u \to_{R_i}^* v' \leftarrow_{R_i}^* w)$, and therefore $Q \Rightarrow_C Q'$. If Q is a critical overlap, then $u = w$ must contain an instance of a

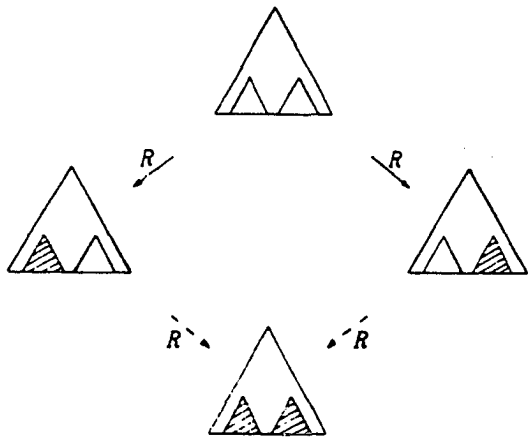


Figure 2.3. *No overlap*

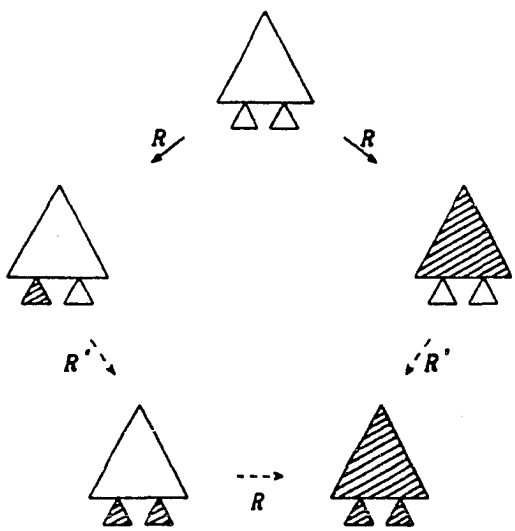


Figure 2.4. *Variable overlap*

critical pair $c = d$, which will be computed eventually because of property (b). Thus there is a proof Q' in $E_j \cup R_j$, for some $j \geq i$, such that $Q \Rightarrow_C Q'$. In either case, we may conclude that $P \Rightarrow_C P'$.

In summary, if P is not a rewrite proof, then there is a proof P' in $E_j \cup R_j$, for some $j \geq i$, such that $P \Rightarrow_C P'$. Therefore the given derivation is fair relative to S_C and N_C. •

We call a derivation *fair (for completion)* if it satisfies properties (a) and (b) of Lemma 2.8.

A *completion procedure* is any procedure that accepts as input a set of equations E, a rewrite system R, and a reduction ordering $>$ containing R; and, using applications of the inference rules of **C** as the only elementary computation steps, generates a derivation $(E_0,R_0), (E_1,R_1), \cdots$, where E_0 is E and R_0 is R. Since a fair derivation may not be possible from an arbitrary pair (E_i,R_i), or may require backtracking (see Dershowitz, Marcus, and Tarlecki, 1987), we have to allow for the possibility of *failure* for certain inputs E, R and $>$. A completion procedure is called *fair* if it generates only fair derivations unless it fails. Theorem 2.1 and Lemma 2.7 together yield:

THEOREM 2.2. (Huet, 1981) *If a completion procedure is fair and does not fail for inputs E, R and $>$, then R^∞ is canonical.*

COROLLARY 2.1. *Let C be a fair completion procedure. If $s \leftrightarrow^*_{E \cup R} t$ and C does not fail for inputs E, R and $>$, then it will generate a pair (E_i,R_i) such that $s \downarrow_{R_i} t$.*

If R^∞ is finite, then it may be used as a *decision procedure* for the *validity problem* in $E \cup R$; if it is infinite, completion still provides a *semi-decision procedure.*

Any particular completion procedure has to specify in which order the inference rules of C are to be applied to given sets of equations and rules. We call such a selection strategy fair if it gives rise only to fair or failing derivations. By Theorem 2.2, any implementation using a fair selection strategy is guaranteed to construct a canonical system, provided it does not fail. Such an implementation is therefore called *correct*. The correctness—in this sense—of a specific completion procedure was first proved by Huet (1981). Huet's proof requires intricate arguments using induction on certain orderings on *terms*. One of the main differences with our approach is that we use orderings on *proofs*. The use of multisets of terms, as in Jouannaud and Kirchner (1986), may be regarded as a simple instance of a proof ordering that makes no use of the information contained in the proof steps. The full potential of proof orderings is only realized when this information is utilized.

The notion of completion as we have formalized it above covers a wide variety of specific completion procedures, including those given in Knuth and Bendix (1970) and Huet (1981). These versions of completion permit application of inference rules in C only in a systematically restricted way. For example, computation of a critical pair $c = d$, i.e. application of C2, is usually followed by the normalization of the critical pair, that is, simplification of $c = d$ by repeated application of C3. Keeping the sets of equations and rules fully simplified tends to improve the efficiency of the completion process and guarantees that the final system will be reduced.

Proof orderings provide a convenient tool for proving the correctness of specific implementations of completions. In addition, enhancements and improvements of standard completion can easily be shown to be correct. For instance, since fairness requires only computation of critical pairs between persisting rules, simplifying a rule may be combined with deleting critical pairs that originated

from this rule. However, only those critical pairs may be deleted that have not yet been turned into rules and used for simplification.

Equations that contain instances of other equations can be deleted. Formulated as an inference rule:

C5) Deleting a subsumed equation

$$\frac{(E \cup \{s \doteq t,\ u[s\sigma] = u[t\sigma]\}, R)}{(E \cup \{s = t\}, R)}$$

Furthermore, orientable instances of equations can be used for simplification. That is, if $u = v$ is an equation and $u\sigma > v\sigma$, then the rule $u\sigma \rightarrow v\sigma$ can be used for simplification. We do not list the corresponding inference rules, since they are essentially the same as C3, S1 and S2. The associated elimination patterns define a proof ordering (that is not contained in $>_C$, though).

Failure is a major drawback of standard completion. Part (b) of the fairness requirement will always be satisfied, provided critical pairs are selected properly for application of C2. Therefore failure can only be the result of violating part (a). Whenever an equation $s = t$ is generated such that s and t are irreducible in $\cup_{i \geq 0} R_i$ and neither $s > t$ nor $t > s$ hold, then no inference rule is ever applicable to $s = t$, hence no fair derivation possible. Conversely, if the reduction ordering $>$ is total on each congruence class $[t]$ in $E \cup R$, then C1 is applicable to any equational consequence $s = t$ of $E \cup R$. This requirement is too strong, though, and such an ordering need not exist for an arbitrary pairs (E, R). For example, the terms $f(x, y)$ and $f(y, x)$ are incomparable in any reduction ordering. Therefore, if E contains a commutativity axiom, no reduction ordering can be total on all congruence classes, and completion may fail in such an instance. This problem can be partially remedied by building certain axioms, such as commutativity, directly into the completion procedure. We will describe

this approach in detail in the next chapter. A different approach, that eliminates the possibility of failure at all, will be described in Chapter 4.

2.2. Critical Pair Criteria

The efficiency of the completion process depends on the number of rules and critical pairs that are generated. Simplification is important because it allows deletion of redundant rules and critical pairs. For instance, whenever a critical pair $c = d$ is generated, both c and d can be reduced to normal forms c' and d'. If c' and d' are identical, then the equation $c' = d'$ can be deleted, indicating that the original equation $c = d$ was redundant. Normalizing an equation can be costly, whereas redundancy of a critical pair $c = d$ can often be determined by looking at the structure of its associated critical overlap $c \leftarrow_R u \rightarrow_R d$. Characterizations of redundant critical pairs are commonly called critical pair criteria. We utilize proof orderings both in formalizing and verifying such criteria.

A critical pair can be considered redundant if its associated critical overlap can be eliminated without computing the critical pair itself. We say that a set S_{CPC} of elimination patterns of the form

$$s \leftarrow_R u \rightarrow_R t \Rightarrow s \leftrightarrow^*_{E \cup R} t,$$

where R is contained in the given reduction ordering $>$, specifies a *critical pair (elimination) criterion* CPC. By $\Rightarrow_{CPC}$ we denote the corresponding ordering on proofs. A critical pair is redundant according to criterion CPC, if its associated critical overlap can be eliminated by $\Rightarrow_{CPC}$. In other words, $\Rightarrow_{CPC}$ can be used to sort out redundant critical pairs. Intuitively, a criterion is "correct" if it is "compatible" with completion and if $\Rightarrow_{CPC}$ is a proof ordering. More formally,

Definition 2.2. A critical pair criterion CPC is *correct* if the ordering induced by $S_C \cup S_{CPC}$ is well-founded.

Let $CPC_{E,R}$ be the set of all critical pairs $c = d$, such that there is no proof P' in $E \cup R$ for which $P \Rightarrow_{CPC} P'$, where P is the critical overlap $c \leftarrow_R u \rightarrow_R d$ associated with $c = d$. If a criterion is correct, then critical pairs not in $CPC_{E,R}$ may be ignored by completion.

Definition 2.3. A derivation $(E_0,R_0), (E_1,R_1), \cdots$ is *fair relative to a critical pair criterion CPC* if (a) $E^\infty = \emptyset$, and (b) CPC^∞ is contained in $\cup_k E_k$, where CPC^∞ is $\cup_i \cap_{j \geq i} CPC_{E_j,R_j}$.

THEOREM 2.3. *Let CPC be a correct critical pair criterion and C be a completion procedure that is fair relative to CPC. If C does not fail for inputs E, R and >, then R^∞ is canonical.*

Proof. The proof is by induction on $\Rightarrow_C \cup \Rightarrow_{CPC}$. Let CPC be a correct criterion and suppose that the derivation $(E_0,R_0), (E_1,R_1), \cdots$ is fair relative to CPC. We show that whenever a proof P in $E_i \cup R_i$ is not a rewrite proof, then there is a proof P' in $E_j \cup R_j$, for some $j \geq i$, such that $P \Rightarrow_C P'$ or $P \Rightarrow_{CPC} P'$. If P uses an equation or a non-persisting rule, then, by fairness, $P \Rightarrow_C P'$, for some proof P'. Suppose that P contains a peak $s \leftarrow_{R_i} u \rightarrow_{R_i} t$, where both rewrite steps persist. If this peak is not a critical overlap, then there is a rewrite proof $s \rightarrow_{R_i}^* v \leftarrow_{R_i}^* t$, and hence $P \Rightarrow_C P'$, for some P'. Otherwise, s is $u[c\sigma]$ and t is $u[d\sigma]$, for some critical pair $c = d$. If $c = d$ is in CPC^∞, then, by fairness, $c = d$ is in E_j, for some j, and therefore $P \Rightarrow_C P'$, for some proof P' in $E_j \cup R_j$. If $c = d$ is not in CPC^∞, then it is not contained in CPC_{E_j,R_j}, for some $j \geq i$. By correctness of CPC, there exists a proof P' in $E_j \cup R_j$, such that $P \Rightarrow_{CPC} P'$. •

A criterion can considerably decrease the total number of critical pairs generated by completion. This advantage may be offset, however, by the additional

cost of checking whether the criterion applies to a given critical pair. Critical pair criteria have also been applied to testing the Church-Rosser property.

Definition 2.4. A criterion CPC is *sound* if, for any terminating rewrite system R, the following property holds:

R is Church-Rosser if and only if, for all critical pairs $c = d$ not in $CPC_{\emptyset,R}$, there is a term v, such that $c \rightarrow_R^* v \leftarrow_R^* d$.

A sound criterion, whose applicability can be effectively tested, can be used for testing the Church-Rosser property of terminating systems. Soundness of a criterion can usually be established without difficulty. Verifying correctness is considerably more difficult.

Lemma 2.9. *Any correct criterion is sound.*

Proof. Let R be a rewrite system and CPC a correct criterion. In addition, suppose that, for all critical pairs $c = d$ not in $CPC_{\emptyset,R}$, there is a term v with $c \rightarrow_R^* v \leftarrow_R^* d$. We have to show that R is Church-Rosser. Any proof P in R that is not a rewrite proof must contain a peak $s \leftarrow_R u \rightarrow_R t$. If this peak is not a critical overlap, then $P \Rightarrow_C P'$, for some proof P'. Otherwise, $s = t$ must contain an instance of a critical pair $c = d$. If $c = d$ is not in $CPC_{\emptyset,R}$, then, by assumption, there is a term v with $c \rightarrow_R^* v \leftarrow_R^* d$, i.e. $P \Rightarrow_C P'$, for some P'. If $c = d$ is in $CPC_{\emptyset,R}$, then $P \Rightarrow_{CPC} P'$, by the definition of $CPC_{\emptyset,R}$. The assertion follows by induction on $\Rightarrow_C \cup \Rightarrow_{CPC}$. •

Formalizing critical pair criteria by using proof orderings greatly facilitates the task of proving correctness. We will present correctness proofs for all known criteria, including those for which correctness had not been established previously.

2.3. Connectedness

Several critical pair criteria have been proposed that are based on the concept of *connectedness.*

Definition 2.5. Let R be a rewrite system and $>$ be a reduction ordering. Two terms s and t are *connected* (in $E \cup R$) below u (relative to $>$) if $s = u_0 \leftrightarrow_{E \cup R} u_1 \cdots \leftrightarrow_{E \cup R} u_n = t$, for some terms $u_0, u_1, \ldots, u_n$ with $u > u_i$, for $0 < i < n$.

This concept was introduced in a more restricted form by Buchberger (1984) and can be readily utilized for a critical pair criterion. Completion can be viewed as a process of establishing, for every critical overlap $c \leftarrow_R u \rightarrow_R d$, the connectedness of c and d below the overlapped term u. For instance, adding the critical pair $c = d$ as an equation is one possible way of establishing connectedness. Conversely, if c and d are already connected, then the critical pair $c = d$ is redundant. Thus, we define the set $\mathbf{S}_{CCP}$ as consisting of all elimination patterns, for $n \geq 1$, of the form

$$s \leftarrow_R u \rightarrow_R t \Rightarrow s \leftrightarrow_{E \cup R} u_1 \leftrightarrow_{E \cup R} \cdots u_{n-1} \leftrightarrow_{E \cup R} t,$$

where $>$ contains R and $u > u_i$, for $1 \leq i < n$. The set $CCP_{E,R}$ contains all critical pairs $c = d$ that are not connected below the associated overlapped term u.

PROPOSITION 2.2. *Criterion CCP is correct for completion.*

Proof. It suffices to prove that $\Rightarrow_{CCP}$ is contained in the proof ordering $>_C$. Suppose that $P \Rightarrow_{CPC} P'$, i.e. P is $s \leftarrow_R u \rightarrow_R t$ and P' is $u_0 \leftrightarrow_{E \cup R} \cdots \leftrightarrow_{E \cup R} u_n$, where u_0 is s, u_n is t, and $u > u_i$, for $1 \leq i < n$. The first component of the quadruple $c(s, u)$ is $\{u\}$, and the first component of $c(u_{i-1}, u_i)$ is $\{u_{i-1}, u_i\}$, $\{u_{i-1}\}$, or $\{u_i\}$. Since $u > u_i$, for $0 \leq i \leq n$, we have

$c(s,u) >^c c(u_i, u_{i+1})$, which implies $P >_C P'$. •

A criterion based on connectedness was first formulated by Buchberger (1979) for a completion-like algorithm for constructing canonical bases for polynomial ideals. Similar criteria for completion have been described by Winkler and Buchberger (1983), Winkler (1984, 1985), and Küchlin (1985, 1986a). These criteria differ in the respective tests that are used for checking whether a critical pair is connected relative to the ordering $\rightarrow_R^+$ induced by R. Let us sketch the basic scheme. Suppose $c \leftarrow_R u \rightarrow_R d$ is a critical overlap and u is reducible to a term v at a position strictly below the critical pair position p. Thus, we have overlaps $c \leftarrow_R u \rightarrow_R v$ and $v \leftarrow_R u \rightarrow_R d$. If $c \leftarrow_R u \rightarrow_R v$ is a variable overlap, then c and v are connected below u. Otherwise, $c = v$ must contain an instance of a critical pair $c' = d'$. If this critical pair has already been generated, then c and v are also connected below u. Similar arguments apply to v and d. Thus, connectedness can often be verified by checking whether certain critical pairs have already been computed. Various book-keeping mechanisms have been proposed for that purpose (e.g. Küchlin, 1985, 1986a; Winkler, 1985). The test described by Winkler (1985) restricts the position at which the rewrite step $u \rightarrow_R v$ may apply. No such restriction is imposed by Küchlin (1985, 1986a).

The emphasis in the papers cited above is on soundness and practicality. Winkler (1985) and Küchlin (1986a) also show the correctness of specific versions of completion that incorporate tests for connectedness. Winkler's proof is similar to the proof of correctness of standard completion in Huet (1981); Küchlin's proof is based on multiset induction. Both proofs are quite complicated. Our correctness proof, besides being considerably simpler, applies to a large class of completion procedures. The flexibility of our approach should be particularly helpful in

establishing the correctness of other implementations of completion procedures and criteria.

2.4. Compositeness

A different type of criterion was suggested by Kapur, Musser, and Narendran (1985).

Definition 2.6. Let R be a rewrite system and $c \leftarrow_R u \rightarrow_R d$ be a critical overlap in R. The critical pair $c = d$ is called *composite* if the overlapped term u is reducible in R strictly below the critical pair position p.

Let S_{PCP} be the set of elimination patterns

$$s \leftarrow_R u \rightarrow_R t \quad \Rightarrow \quad s \leftarrow_R u \rightarrow_R v \leftarrow_R u \rightarrow_R t$$

where the rewrite step $u \rightarrow_R v$ applies strictly below $u \rightarrow_R s$, and $u \rightarrow_R s$ applies below $u \rightarrow_R t$. The proof relation $\Rightarrow_{PCP}$ induced by S_{PCP} can be used to eliminate overlaps corresponding to composite critical pairs. The set $PCP_{E,R}$ contains all non-composite critical pairs of R. Criterion PCP is sound:

Lemma 2.10. (Kapur, Musser, and Narendran, 1985) *A terminating rewrite system R is Church-Rosser if and only if, for every non-composite critical pair $c = d$, there is a term v, such that $c \rightarrow_R^* v \leftarrow_R^* d$.*

We prove the correctness of PCP by constructing a well-founded ordering $>_{PCP}$ that contains $\Rightarrow_{PCP}$ and $\Rightarrow_C$. Let P be an overlap $s \leftarrow_R u \rightarrow_R t$, and P' be $s \leftarrow_R u \rightarrow_R v \leftarrow_R u \rightarrow_R t$, where $u \rightarrow_R v$ by $l'' \rightarrow r''$ at a position q strictly below p (see Figure 2.5). Since both P and P' contain the proof steps $u \rightarrow_R s$ and $u \rightarrow_R t$, we have $P' >_C P$! However, including a measure of the overlap between successive proof steps in the complexity of a proof allows us to distinguish between occurrences of these single proof steps in P and P',

respectively, so that we can design a proof ordering $>_{PCP}$ wherein $P >_{PCP} P'$.

Let P be a proof $s_0, \ldots, s_n$ and p_i be the position of the i-th proof step (s_{i-1}, s_i). We define the complexity measure c_P by:

if $s_{i-1} \rightarrow_R s_i$ by $l \rightarrow r$, then $c_P(s_{i-1}, s_i, P)$ is $(\{s_{i-1}\}, s_{i-1}/p_i, l, s_i, s_{i-1}/p_{i-1})$, where s_{i-1}/p_{i-1} is $-$, if $i = 1$;

if $s_{i-1} \leftarrow_R s_i$ by $l \rightarrow r$, then $c_P(s_{i-1}, s_i, P)$ is $(\{s_i\}, s_i/p_i, l, s_{i-1}, s_i/p_{i+1})$, where s_i/p_{i+1} is $-$, if $i = n$;

if $s_{i-1} \leftrightarrow_E s_i$, then $c_P(s_{i-1}, s_i, P)$ is $(\{s_{i-1}, s_i\}, -, -, -, -)$.

The first four components of c_P are the same as for the complexity measure c. The additional fifth component reflects the amount of the overlap of a rewrite step with its neighboring step. The ordering $>_P^c$ is the lexicographic combination of the multiset extension of the reduction ordering $>$, the proper subterm ordering, the proper subsumption ordering $\rhd$, the reduction ordering $>$, and the proper subterm ordering. This ordering is well-founded and stable, but not monotonic. Let $>_{PCP}$ be the ordering on proofs corresponding to c_P and $>_P^c$. This ordering is well-founded.

Lemma 2.11. *The ordering* $>_{PCP}$ *contains* $\Rightarrow_C$.

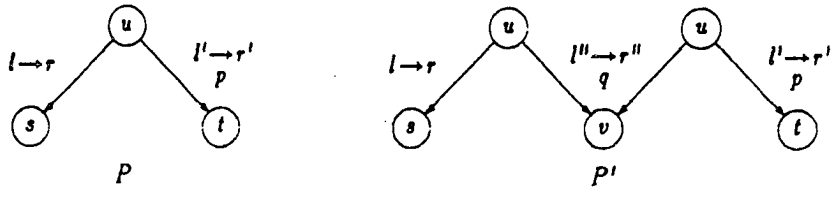


Figure 2.5.

Proof. By similar arguments as in the proof of Lemma 2.6, we can show that $P >_{PCP} P'$, for any instance $P \Rightarrow P'$ of any elimination pattern for standard completion. (In addition, the multisets $M(P)$ and $M(P')$ have no elements in common.) However, since $>_{PCP}$ is not monotonic, we can not immediately conclude that $Q[P] >_{PCP} Q[P']$. Let Q be a proof $(s_0, \ldots, s_i, P, s_j, \ldots, s_n)$ and Q' be $Q[P']$. Since $c_P(s_{k-1}, s_k, Q) = c_P(s_{k-1}, s_k, Q[P'])$, for $k < i$ and $k > j$, we have $Q >_{PCP} Q'$ if and only if $(s_i, P, s_j) >_{PCP} (s_i, P', s_j)$. Let us assume that Q is (s', P, t') and P and P' are proofs of $s = t$. Since the complexity of a proof step, may depend on its neighboring step, replacing the subproof P by P' may change the complexity of the first and last proof step in Q, though these proof steps themselves remain unchanged. Thus, if the first proof step of Q is $s' \leftarrow_R s$, then its complexity depends on the first step in P. Similarly, if the last step is $t \rightarrow_R t'$, then its complexity depends on the last step in P. We consider these problematic cases for all instances $P \Rightarrow P'$ of elimination patterns in $\mathbb{S}_C$.

a) If (P, P') is an instance of an equality pattern, then P is $s \leftrightarrow_E t$. If $s' \leftarrow_R s$, then $c_P(s, t, Q) >_P^c c_P(s', s, Q')$. If $t \rightarrow_R t'$, then $c_P(s, t, Q) >_P^c c_P(t, t', Q')$. Thus, we have $Q >_{PCP} Q'$.

b) If (P, P') is an instance of an overlap pattern, then P is $s \leftarrow_R u \rightarrow_R t$. If $s' \leftarrow_R s$, then $c_P(s, u, Q) >_P^c c_P(s', s, Q')$. If $t \rightarrow_R t'$, then $c_P(u, t, Q) >_P^c c_P(t, t', Q')$. Again, we obtain $Q >_{PCP} Q'$.

c) If (P, P') is an instance of a simplification pattern, then P is $s \rightarrow_R t$ and P' is either $s \rightarrow_{R'} u \leftarrow_{R'} t$ or $s \rightarrow_{R'} u \leftrightarrow_{E'} t$. The conditions on P' guarantee that the position of $s \rightarrow_{R'} u$ is below (though not necessarily strictly below) the position of $s \rightarrow_R t$. Consequently, we have $c_P(s', s, Q) \geq_P^c c_P(s', s, Q')$. In addition, $c_P(s, t, Q) >_P^c c_P(t, t', Q')$, which implies $Q >_{PCP} Q'$. •

PROPOSITION 2.3. *Criterion PCP is correct.*

Proof. By the above Lemma, $>_{PCP}$ contains $\Rightarrow_C$. We show that it also contains $\Rightarrow_{PCP}$. Suppose $P \Rightarrow_{PCP} P'$, i.e. P contains an overlap $s \leftarrow_R u \rightarrow_R t$ that can be decomposed into $s \leftarrow_R u \rightarrow_R v \leftarrow_R u \rightarrow_R t$, as shown in Figure 2.5. The proof P has complexity

$$M(P) = N \cup \{(\{u\}, u, l, s, u/p), (\{u\}, u/p, l', t, u)\},$$

whereas P' has complexity

$$M(P') = N \cup \{(\{u\}, u, l, s, u/q), (\{u\}, u/q, l'', v, u),$$
$$(\{u\}, u/q, l'', v, u/p), (\{u\}, u/p, l', t, u/q)\},$$

where the position q is strictly below p. Since the first three elements in $M(P')$ are smaller than the first element in $M(P)$, and the last element is smaller than the second element in $M(P)$, we have $P >_{PCP} P'$. •

As a special case of non-compositeness we have *blocking*, a concept that has been used in theorem proving and rewriting by Slagle (1974) and Lankford and Ballantyne (1979).

Definition 2.7. Let $c = d$ be a critical pair corresponding to rules $s \rightarrow t$ and $l \rightarrow r$ in R and substitution σ. Then $c = d$ is called *blocked* if $x\sigma$ is irreducible, for all variables x in c or d. Otherwise, it is called *unblocked.*

An unblocked critical pair is composite. The set S_{BCP} is obtained by imposing on the elimination patterns

$$s \leftarrow_R u \rightarrow_R t \Rightarrow s \leftarrow_R u \rightarrow_R v \leftarrow_R u \rightarrow_R t,$$

of S_{PCP} the additional constraint that the rewrite step $u \rightarrow_R v$ apply at a position q such that either u/q is a variable or q is not a position in u at all. Let $\Rightarrow_{BCP}$ be the corresponding relation on proofs. By definition, $\Rightarrow_{BCP}$ is contained in $\Rightarrow_{PCP}$. The set $BCP_{E,R}$ consists of all blocked critical pairs of R.

COROLLARY 2.2. *Criterion BCP is correct.*

Proof. The lemma is an immediate consequence of Proposition 2.3. •

The composite criterion PCP or the blocked criterion BCP can be included in any completion procedure without difficulty. A critical pair $c = d$ is composite if and only if some proper subterm of the overlapped term u is reducible. It is unblocked if, for some variable x in c or d, the term $x\sigma$ is reducible, where σ is the (most general) unifier associated with $c = d$. Composite and connectedness criteria can be combined.

PROPOSITION 2.4. *The combination of PCP and CCP, i.e. the critical pair criterion corresponding to* $S_{PCP} \cup S_{CCP}$, *is correct.*

Proof. The orderings $\Rightarrow_C$, $\Rightarrow_{CCP}$ and $\Rightarrow_{PCP}$ are all contained in $>_{PCP}$; hence their union is well-founded. •

Experimental results that give some indication of the practicality of critical pair criteria have been reported by Kapur, Musser and Narendran (1985)—for compositeness—and by Küchlin (1985)—for connectedness.

CHAPTER 3

REWRITING MODULO A CONGRUENCE

Equational theories that can not be characterized by canonical rewrite systems include, for instance, many theories that contain commutativity axioms. Commutativity can not be used as a rewrite rule since this would destroy the termination property. Instead, such problematic axioms can often be handled by generalizing the notions of rewriting, matching, and unification, defining them with respect to a given congruence. A number of approaches have been suggested for "rewriting modulo a congruence." Lankford and Ballantyne (1977a, b, c) present Church-Rosser theorems for sets of permutativity (e.g. associativity and commutativity) axioms; Peterson and Stickel (1981) describe a completion method for associative-commutative rewriting; Huet (1980) studies left-linear rewrite systems; Jouannaud (1983) and Jouannaud and Kirchner (1986) formulate completion procedures for sets of equations A that generate finite congruence classes. In this chapter, we present new completion methods for rewriting modulo a congruence.

3.1. Completion for Left-linear Rewrite Systems

Let A be a set of equations. 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 $s \rightarrow^*_R u \leftrightarrow^*_A v \leftarrow^*_R t$. A proof of the form $s \rightarrow^*_R u \leftrightarrow^*_A v \leftarrow^*_R t$ is called a *rewrite proof modulo* A. The rewrite system R/A (R *mod* A) consists of all

rewrite rules $l \to r$ such that $l \leftrightarrow_A^* u \to_R v \leftrightarrow_A^* r$. Evidently, R/A is Church-Rosser modulo A if and only if it is Church-Rosser.

A reduction ordering $>$ is *compatible* with A if $s > t$ implies $u > v$, 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 there is a reduction ordering $>$ that contains R and is compatible with A. We say that R is *canonical modulo* A if R/A is terminating and R is Church-Rosser modulo A.

Let A be a set of equations. For simplicity, we assume that A is symmetric. A rewrite system R is Church-Rosser modulo A if there is a rewrite proof modulo A for any valid equation in $A \cup R$. A rewrite proof modulo A, on the other hand, is a proof that contains no peak $s \leftarrow_R u \to_{A \cup R} t$. Such peaks can be reduced by the elimination pattern

$$s \leftarrow_R u \to_{A \cup R} t \quad \Rightarrow \quad s \to_R^* v \leftrightarrow_A^* w \leftarrow_R^* t.$$

By $\Rightarrow_{RA}$ we denote the proof relation induced by this pattern.

Let $>$ be a reduction ordering that is compatible with A. Thus, $>$ induces an ordering on congruence classes of $\leftrightarrow_A^*$. We introduce a proof ordering $>_{RA}$ by defining a complexity measure c and ordering $>^c$ as follows:

if $s \leftrightarrow_A t$, then $c(s,t)$ is $\{s, min\}$,

if $s \to_R t$, then $c(s,t)$ is $\{s\}$,

where min is a new constant with $t > min$, for all terms t. The ordering $>^c$ is the multiset extension of $>$. (We assume that terms equivalent under A are considered identical when compared in the ordering $>$. This is permissible, since $>$ is compatible with A.) The ordering $>_{RA}$ is well-founded, monotonic, and stable; hence a proof ordering.

Lemma 3.1. *If R/A terminates, then $\Rightarrow_{RA}$ is a proof ordering.*

Proof. Let $>$ be the ordering $\rightarrow_{R/A}^{+}$. If R/A terminates, then $>$ is a reduction ordering. We show that $>_{RA}$ contains $\Rightarrow_{RA}$. Suppose that $P \Rightarrow_{RA} P'$.

a) If P is $s \leftarrow_R u \rightarrow_R t$ and P' is $s \rightarrow_R^* v \leftrightarrow_A^* w \leftarrow_R^* t$, then $P >_{RA} P'$ since $u > u'$, for any term u' in P'.

b) If P is $s \leftarrow_R u \leftrightarrow_A t$, then P' must be $s \rightarrow_R^* v \leftrightarrow_A^* w \leftarrow_R^+ t$. (The terms t and w must be distinct, for otherwise R/A would be non-terminating.) The term t is the biggest term in P' as well as in P. In P it appears in an equality step, in P', in a rewrite step. Therefore, $P >_{RA} P'$. •

THEOREM 3.1. (Huet, 1980) *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 every proof $s \leftarrow_R u \rightarrow_{A \cup R} t$ is reducible by $\Rightarrow_{RA}$.*

Proof. For the only-if direction, suppose that R is Church-Rosser modulo A. If P is $s \leftarrow_R u \rightarrow_{A \cup R} t$, then, by the Church-Rosser property, $s \rightarrow_R^* v \leftrightarrow_A^* w \leftarrow_R^* t$. In other words, there is a proof P' with $P \Rightarrow_{RA} P'$. For the if-direction, we have to show that there is a rewrite proof for any valid equation in $A \cup R$. Let P be a proof of $s = t$ in $A \cup R$. If P is not a rewrite proof modulo A, then it must contain a subproof $v \leftarrow_R u \rightarrow_{A \cup R} w$. By the assumption, P can be reduced by $\Rightarrow_{RA}$, i.e. there is a proof P', such that $P \Rightarrow_{RA} P'$. The assertion follows by induction on $\Rightarrow_{RA}$. •

Standard completion can be extended in a natural way to rewriting modulo a congruence. The main difference between the extension and the standard method is that for the former critical pairs have to be computed both with rules in R and with equations in A. (A critical pair with an equation $u = v$ in A is a critical pair with $u \rightarrow v$ or $v \rightarrow u$.) Unfortunately, the resulting method works only for left-linear systems. (A rewrite rule $l \rightarrow r$ is *left-linear* if no variable

occurs more than once in l. A rewrite system is left-linear if all its rules are left-linear.)

Let A be a symmetric set of equations and $>$ be a reduction ordering that is compatible with A. The inference system M (*completion for rewriting modulo a congruence*) consists of the following inference rules, where E may be any set of equations and R any rewrite system contained in $>$:

M1) Orienting an equation.

$$\frac{(E \cup \{s \doteq t\}, R)}{(E, R \cup \{s \to t\})} \qquad \text{if } s > t$$

M2) Adding an equational consequence.

$$\frac{(E, R)}{(E \cup \{s = t\}, R)} \qquad \text{if } s \overset{*}{\leftarrow}_{R \cup A} u \overset{*}{\rightarrow}_{R \cup A} t$$

M3) Simplifying an equation.

$$\frac{(E \cup \{s \doteq t\}, R)}{(E \cup \{u = t\}, R)} \qquad \text{if } s \to_{R/A} u$$

M4) Deleting an equation.

$$\frac{(E \cup \{s \doteq t\}, R)}{(E, R)} \qquad \text{if } s \overset{*}{\leftrightarrow}_A t$$

Basic completion is a specific instance of M, for $A = \emptyset$. Note that equations may be simplified by R/A, not just by R. The inference system L consists of M plus the following rules for simplification:

L1) Simplifying a right-hand side of a rule

$$\frac{(E, R \cup \{s \to t\})}{(E, R \cup \{s \to u\})} \qquad \text{if } t \to_{R/A} u$$

L2) Simplifying a left-hand side of a rule

$$\frac{(E, R \cup \{s \to t\})}{(E \cup \{u = t\}, R)} \qquad \text{if } s \to_R u \text{ at a position } p \text{ not at the top}$$

$$\frac{(E, R \cup \{s \to t\})}{(E \cup \{u = t\}, R)} \qquad \text{if } s \to_R u \text{ by } l \to r \text{ and } s \rhd l$$

where $\rhd$ denotes the proper subsumption ordering.

Right-hand sides of rules may be simplified by R/A, whereas simplification of left-hand sides is confined to rewriting in R. The inference system **L** is obviously sound. Corresponding elimination patterns can be constructed easily. We have *equality patterns*

$$\begin{aligned} s \leftrightarrow_E t &\Rightarrow s \to_{R'} t \\ s \leftrightarrow_E t &\Rightarrow s \to_{R'/A} u \leftrightarrow_{E'} t \\ s \leftrightarrow_E t &\Rightarrow s \leftrightarrow_A^* t \end{aligned}$$

overlap patterns

$$\begin{aligned} s \leftarrow_R u \to_{A \cup R} t &\Rightarrow s \to_R^* v \leftrightarrow_A^* w \leftarrow_R^* t \\ s \leftarrow_R u \to_R t &\Rightarrow s \leftrightarrow_{E'} t \\ s \leftarrow_R u \leftrightarrow_A t &\Rightarrow s \leftarrow_{R'} t \end{aligned}$$

and *simplification patterns*

$$\begin{aligned} s \to_R t &\Rightarrow s \to_{R'} u \leftarrow_{R'/A} t \\ s \to_R t &\Rightarrow s \to_{R'} v \leftrightarrow_{E'} t \end{aligned}$$

where R and R' are contained in $>$ and, in the last two patterns, $s \to_R t$ is by application of $l \to r$ at position p; $s \to_{R'} u$ by application of $l \to r'$ at position p; and $s \to_R v$ by application of a rule $l' \to r'$ below p (if the rule applies at position p, then $l \rhd l'$).

Let $\mathbf{S_L}$ be the set of all these patterns and $\Rightarrow_{\mathbf{L}}$ the corresponding proof relation. By $\mathbf{N_L}$ we denote the set of proof patterns $s \leftrightarrow_E t$ and $s \leftarrow_R u \to_{A \cup R} t$.

Lemma 3.2. *The set of elimination patterns* $\mathbf{S_L}$ *is compatible with the inference system* **L**.

Proof. We have to prove that whenever $(E,R)\vdash_{\overline{L}}(E',R')$ and P is a proof in $E\cup R$, there is a proof P' in $E'\cup R'$ such that $P\Rightarrow_L^* P'$. Applications of inference rules M1, M3, and M4 are covered by the equality patterns; L1 and L2, by the simplification patterns. For inference rule M2 the assertion holds trivially, since then R and R' are the same and E is contained in E', which implies that any proof in $E\cup R$ is also a proof in $E'\cup R'$. The overlap patterns can be used to eliminate peaks: the first pattern, to eliminate non-overlaps or variable overlaps; the second, to eliminate critical overlaps between rules of R; the third, to eliminate critical overlaps between R and A. The third overlap pattern can only be applied when the critical pair has been turned into a rule. •

For proving that the ordering $\Rightarrow_L$ is well-founded, we use the following complexity measure c_L:

if $s\leftrightarrow_E t$, then $c_L(s,t)$ is $(\{s,t\},-,-,-)$,

if $s\leftrightarrow_A t$, then $c_L(s,t)$ is $(\{s,min\},-,-,-)$,

if $s\rightarrow_R t$ by $l\rightarrow r$ at position p, then $c_L(s,t)$ is $(\{s\},s/p,l,t)$,

if $s\leftarrow_R t$ by $l\rightarrow r$ at position p, then $c_L(s,t)$ is $(\{t\},t/p,l,s)$.

This complexity measure is similar to the one for standard completion. The ordering $>_L^c$ is the lexicographic combination of the multiset ordering $\gg$, the proper subterm ordering, the proper subsumption ordering $\rhd$, and the reduction ordering $>$. By $>_L$ we denote the ordering specified by c_L and $>_L^c$. This ordering is stable, monotonic, and well-founded; hence a proof ordering.

Lemma 3.3. *The ordering $\Rightarrow_L$ is a proof ordering.*

Proof. It can easily be seen that $>_L$ contains any instance of an elimination pattern of S_L. For equality and simplification patterns the proof is essentially the same as for Lemma 2.6. For overlap patterns the same line of reasoning as in Lemma 3.1 can be adopted. •

We will next derive sufficient conditions for the fairness of a derivation in L relative to S_L and N_L. In the previous chapter we have shown that computation of critical overlaps of R suffices for eliminating peaks $s \leftarrow_R u \rightarrow_R t$. Now, we also have to consider peaks $s \leftarrow_R u \leftrightarrow_A t$. Let P be such a peak.

If P is not an overlap, then a simpler proof can be obtained by commuting the two proof steps: $s \leftrightarrow_A v \leftarrow_R t$.

If P is a critical overlap, then, by the Critical Pair Lemma, $s = t$ contains an instance of a critical pair $c = d$ of $A \cup R$. Just adding the equation $c = d$ does not produce a simpler proof. We even have $(s \leftrightarrow_{E'} t) >_L (s \leftarrow_R u \leftrightarrow_A t)$! However, since the equation $c = d$ was obtained from an overlap $c \leftarrow_R u \leftrightarrow_A d$, it is orientable: $d > c$. Adding the rule $d \rightarrow c$ results in a proof $P' = (s \leftarrow_{R'} t)$ for which we have $P \Rightarrow_L P'$. An essential difference between equations and rules is in the respective simplification steps that can be applied to them. The equation $c = d$ could be reduced to a trivial equation $c = c$, since $c \leftarrow_{R/A} d$. This reduction by R/A takes place at the *top*, and therefore is ruled out as a consequence of turning $c = d$ into a rule $d \rightarrow c$.

Variable overlaps can be problematic. If the reduction step applies in the variable part of the equality step, then there is a proof $P' = (s \rightarrow_R^* v \leftrightarrow_A w \leftarrow_R^* t)$ for which $P \Rightarrow_L P'$. If the equality step applies in the variable part of the reduction step, then there also exists a proof $P' = (s \leftrightarrow_A^* v \leftarrow_R w \leftrightarrow_A^* t)$. Unfortunately, the proof $P = (s \leftarrow_R u \leftrightarrow_A t)$ can not be reduced to P' by $\Rightarrow_L$, in general. More precisely, if $u \rightarrow_R s$ by application of a *left-linear* rule, then $P' = (s \leftrightarrow_A^* v \leftarrow_R t)$, for which we have $P \Rightarrow_L P'$ (see Figure 3.1). On the other hand, if a non-left-linear rule is used in $u \rightarrow_R s$, then P and P' may be incomparable (see Figure 3.2).

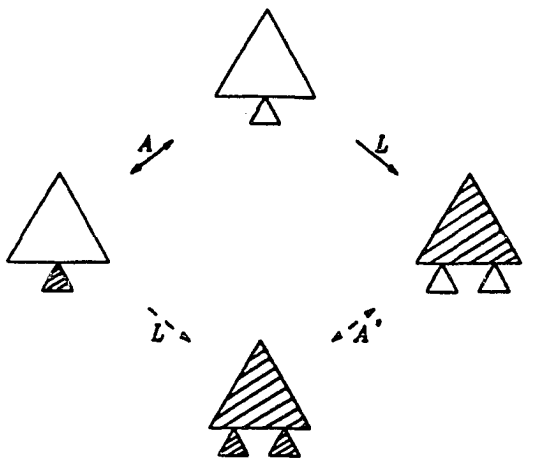


Figure 3.1.

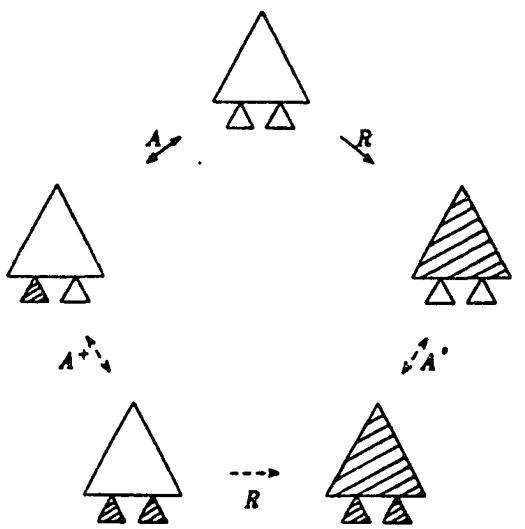


Figure 3.2.

For example, with $A = \{x + y = y + x\}$ and $R = \{f\ (x * x) \rightarrow x\}$ we can construct two incomparable proofs

$$((x+y)^*(y+x)) \leftrightarrow_A f((x+y)^*(x+y)) \rightarrow_R (x+y)$$

and

$$f((x+y)^*(y+x)) \leftrightarrow_A f((y+x)^*(y+x)) \rightarrow_R (y+x) \leftrightarrow_A (x+y).$$

In summary, only if R is left-linear does computation of critical pairs of $A \cup R$ guarantee that critical proof patterns of N_L can be eliminated. The considerations above lead to

Lemma 3.4. *A derivation* $(E_0,R_0), (E_1,R_1), \cdots$ *in* L *is fair relative to* S_L *and* N_L *if* $E^\infty=\emptyset$, R^∞ *is left-linear, all critical pairs of* R^∞ *are contained in* $\cup_k E_k$, *and all critical pairs between* R^∞ *and* A *are contained in* $\cup_k R_k$.

A completion procedure based on L is fair if it generates only fair derivations (in the sense of Lemma 3.4) unless it fails. As an immediate consequence of Theorem 2.1 and Lemma 3.4, we have

THEOREM 3.2. *Let* A *be a set of equations,* R *a rewrite system,* $>$ *a reduction ordering that contains* R *and is compatible with* A, *and* C *a fair* L*-completion procedure. If* C *does not fail for inputs* E, R *and* $>$, *then* $E^\infty=\emptyset$ *and* R^∞ *is Church-Rosser modulo* A.

3.2. Completion Based on A-unification

Variable overlaps of equations in A on non-left-linear rules in R can not always be readily eliminated. Thus, the completion method described above applies only to left-linear systems. A possible remedy for this problem consists of extending the notion of rewriting so that variable overlaps can be regarded as single rewrite steps.

Let A be a set of equations and R be a rewrite system. The rewrite system R/A consists of all rules $l \rightarrow r$ such that $l \leftrightarrow_A^* u\sigma$ and $r = v\sigma$, for some rule

$u \to v$ in R and some substitution σ. For example, if A consists of the associativity and commutativity axioms for addition, and R contains rules $-x+x \to 0$ and $f(x,x) \to g(x)$, then $f(x+y,y+x)$ is irreducible in R, but reduces to $g(x+y)$ in $R \cdot A$. The term $-x+(x+y)$ is irreducible in $R \cdot A$, whereas it reduces to $0+y$ in R/A. A rewrite step in $R \cdot A$ corresponds to the application of a rule in R using *A-matching*, i.e. matching with respect to the congruence $\leftrightarrow_A^*$.

Any variable overlap $s \leftrightarrow_A u \to_R t$ of A on R can be regarded as a single rewrite step in $R \cdot A$. The rewrite relation $R \cdot A$ therefore obviates the problem with variable overlaps, but introduces a new problem of eliminating more complicated peaks. We will see that such peaks can be efficiently resolved when there exists a finite complete unification algorithm for A.

We will study rewrite systems R that are partitioned into two sets L and N, where L contains only left-linear rules, and corresponding rewrite relations $R^A = L \cup N \cdot A$. In other words, A-matching is restricted to rules in N. Thus, $s \to_{R^A} t$ abbreviates $s \to_L t$ or $s \to_{N \cdot A} t$.

The inference system M can be used as a basis for constructing, for given sets of equations A and E, a rewrite system $R = L \cup N$ such that $L \cup N \cdot A$ is canonical modulo A and the congruence relations $\leftrightarrow_{A \cup E}^*$ and $\leftrightarrow_{A \cup R}^*$ are the same. Let $>$ be a reduction ordering that is compatible with A. The inference system A consists of M plus the following inference rules for simplification, where E may be any set of equations and R any rewrite system contained in $>$:

A1) Simplifying a right-hand side

$$\frac{(E, R \cup \{s \to t\})}{(E, R \cup \{s \to u\})} \qquad \text{if } t \to_{R/A} u$$

A2) Simplifying a left-hand side

$$\frac{(E, R \cup \{s \to t\})}{(E \cup \{u = t\}, R)} \qquad \text{if } s \to_{R/A} u \text{ at a position not at the top,}$$

$$\frac{(E, R \cup \{s \to t\})}{(E \cup \{u = t\}, R)} \qquad \text{if } s \to_R u \text{ by } l \to r \text{ and } s \rhd l$$

where $\rhd$ denotes the proper subsumption ordering.

We implicitly assume that R and R' are partitioned into $L \cup N$ and $L' \cup N'$, respectively, where L and L' are left-linear. If a rule $s \to t$ is in L, then simplification of its right-hand side by A1 yields a rule $s \to u$ in L'; if $s \to t$ is in N, then $s \to u$ is in N'.

Application of A1 and A2 can be described by the *simplification patterns*

$$\begin{aligned} s \to_R t \quad &\Rightarrow \quad s \to_{R'} u \leftarrow_{R'/A} t \\ s \to_R t \quad &\Rightarrow \quad s \to_{R'/A} v \leftrightarrow_{E'} t \\ s \to_R t \quad &\Rightarrow \quad s \to_{R'} w \leftrightarrow_{E'} t \end{aligned}$$

where $s \to_R t$ is by application of a rule $l \to r$ at position p; $s \to_{R'} u$ is by $l \to r'$ at position p; $s \to_{R'/A} v$ is by application of a rule strictly below p; and $s \to_{R'} w$ is by a rule $l' \to r'$ with $l \rhd l'$ at position p.

A proof in $A \cup E \cup R$ is a rewrite proof modulo A, relative to R^A, if and only if it contains no equality step $s \leftrightarrow_E t$ and no peak $s \leftarrow_{A \cup R^A} u \to_{R^A} t$. Any peak $s \leftarrow_{A \cup R^A} u \to_{R^A} t$ contains a simpler peak $s \leftarrow_{A \cup R} u \to_{R^A} t$. (Recall that $l \to_{R^A} r$ abbreviates $l \to_L r$ or $l \to_{N \cdot A} r$, and that $l \to_{N \cdot A} r$ abbreviates $l \leftrightarrow_A^* l' \to_N r$.) We will derive patterns for eliminating such peaks between $A \cup R$ and R^A.

Non-overlaps can be eliminated by the usual pattern

$$s \leftarrow_{A \cup R} u \to_{R^A} t \quad \Rightarrow \quad s \to_{R^A} v \leftarrow_{A \cup R} t$$

where the two proof steps on the left-hand side apply at disjoint positions p and

q, and are simply commuted to obtain the pattern on the right-hand side.

For variable overlaps we have patterns

$$\begin{aligned} s \leftarrow_R u \rightarrow_{R^A} t &\Rightarrow s \rightarrow^*_{R^A} v \leftarrow_R w \leftarrow^*_{R^A} t \\ s \leftrightarrow_A u \rightarrow_L t &\Rightarrow s \rightarrow^*_L v \leftrightarrow_A w \leftarrow^*_L t \\ s \leftarrow_L u \leftrightarrow_A t &\Rightarrow s \leftrightarrow^*_A v \leftarrow_L t \\ s \leftrightarrow_A u \rightarrow_{N\cdot A} t &\Rightarrow s \rightarrow^*_{N\cdot A} v \leftrightarrow_A w \leftarrow^*_{N\cdot A} t \end{aligned}$$

where all left-hand sides denote variable overlaps with the second step below the first (strictly below in the last pattern) and right-hand sides denote the corresponding rearrangements of peaks (see also Lemma 2.7). There are no variable overlaps of A on $N \cdot A$.

Overlaps can be effectively eliminated if a finite, complete unification algorithm for the theory A is known. Two terms s and t are *A-unifiable* if there exists a substitution (an *A-unifier*) σ, such that $s\,\sigma \leftrightarrow^*_A t\,\sigma$. A set Σ of A-unifiers of s and t is *complete* if for any A-unifier τ of s and t there exists a substitution ρ, such that $x\,\tau \leftrightarrow^*_A (x\,\sigma)\rho$, for all variables x. We will assume, from now on, that finite complete sets of unifiers for A exist and that an algorithm for computing them is given. Finite, complete unification algorithms are known for many theories of practical importance, including commutativity (Plotkin, 1972), associativity and commutativity (Stickel, 1981; Fages, 1984), and associativity, commutativity and identity (Fages, 1984). If A is the empty set, then the set consisting of the (unique) most general unifier of s and t is complete. For an overview on unification, see Siekmann (1984).

Let $u \rightarrow v$ and $l \rightarrow r$ be rules in R and R', respectively, with no variables in common (the variables of one rule are renamed if necessary). Let p be a non-variable position in u, such that u/p and l are A-unifiable with a complete set of unifiers Σ. For any σ in Σ, the proof $v\,\sigma \leftarrow_R u\,\sigma \rightarrow_{R'\cdot A} u\,\sigma[p/r\,\sigma]$ is called an

A -critical overlap of R' on R. The equation $v\sigma = u\sigma[p/r\sigma]$ is called an *A-critical pair* of $l \to r$ on $u \to v$ at position p (or an A-critical pair of R' on R). An A-critical pair between R and R' is either an A-critical pair of R on R' or of R' on R. If A is empty, then we speak of *critical pairs.*

Lemma 3.5. (Extended Critical Pair Lemma; Jouannaud, 1983). *Let A be a set of equations for which there exists a finite complete unification algorithm. Let $u \to v$ and $l \to r$ be rules and p a position in u, such that u/p is not a variable and is A-unifiable with l, Σ being a complete set of A-unifiers. Then there exist, for any overlap $v\tau \leftarrow_R u\tau \rightarrow_{R'} u\tau[p/r\tau]$, substitutions σ and ρ, σ in Σ, such that $x\tau \leftrightarrow_A^* (x\sigma)\rho$, for all variables x in $u \to v$ or $l \to r$. Consequently, there exists an A-critical pair $c = d$, such that $v\tau \leftrightarrow_A^* c\rho$ and $u\tau[p/r\tau] \leftrightarrow_A^* d\rho$. If v is not a variable, then no equation in $v\tau \leftrightarrow_A^* c\rho$ applies at the top.*

This lemma is the basis for the *overlap patterns*

$$
\begin{array}{rcl}
s \leftarrow_R u \rightarrow_L t & \Rightarrow & s \leftrightarrow_{E'} t \\
s \leftrightarrow_A u \rightarrow_L t & \Rightarrow & s \rightarrow_{R'} t \\
s \leftarrow_L u \leftrightarrow_A t & \Rightarrow & s \leftarrow_{R'} t \\
s \leftarrow_R u \rightarrow_{N \cdot A} t & \Rightarrow & s \leftrightarrow_A^* v \leftrightarrow_{E'} w \leftrightarrow_A^* t \\
s \leftrightarrow_A u \rightarrow_{N \cdot A} t & \Rightarrow & s \leftrightarrow_A^* v \rightarrow_{R'} w \leftrightarrow_A^* t
\end{array}
$$

where all left-hand sides denote overlaps with the second step applying below the first step, and all proof steps on right-hand sides apply below the position of $u \rightarrow_{A \cup R} s$. In addition, in the last pattern, the positions of $u \rightarrow_{N \cdot A} t$ and of all steps in $s \leftrightarrow_A^* v$ are strictly below the position of $u \rightarrow_A s$. The equality steps in $s \leftrightarrow_A^* v$ and $w \leftrightarrow_A^* t$ reflect the fact that an overlap between $N \cdot A$ and $A \cup R$ need not contain an instance of an A-critical pair, but only an equation equivalent to such an instance.

By S_A we denote the set consisting of the above simplification patterns and elimination patterns for peaks and overlaps, and the equality patterns from the previous section. If the set A satisfies certain conditions, then the ordering $\Rightarrow_A$ is a proof ordering. By N_A we denote the set of proof patterns $s \leftrightarrow_E t$ and $s \leftarrow_{A \cup R^A} u \rightarrow_{R^A} t$. We obviously have

Lemma 3.6. *The set of elimination patterns* S_A *is compatible with the inference system* A.

We next introduce a complexity measure c_A that is needed for proving well-foundedness of $\Rightarrow_A$. Let P be a proof $(t_0, \ldots, t_n)$ in $A \cup E \cup R$ and let p_i be the position of the i-th proof step (t_{i-1}, t_i). The complexity $M_A(P)$ of P is the multiset $\{c_A(t_0, t_1, P), \ldots, c_A(t_{n-1}, t_n, P)\}$, where $c_A(t_{i-1}, t_i, P)$ is

$$
\begin{array}{ll}
(\{t_{i-1}, t_i\}, -, -, -, -) & \text{if } t_{i-1} \leftrightarrow_E t_i \\
(\{t_{i-1}\}, t_{i-1}/p_i, max, l, t_i) & \text{if } t_{i-1} \leftrightarrow_A t_i \text{ by an equation } l = r \\
(\{t_{i-1}\}, t_{i-1}/p_i, e(t_{i-1}, t_i, P), l, t_i) & \text{if } t_{i-1} \rightarrow_R t_i \text{ by a rule } l \rightarrow r
\end{array}
$$

and $e(t_{i-1}, t_i, P)$ is the multiset $\{t_{i-k+1}/p_{i-k+1}, \ldots, t_{i-1}/p_{i-1}\}$, with k being the largest index for which $(t_{i-k}, \ldots, t_{i-1})$ is a proof of the form $t_{i-k} \leftrightarrow_A^* t_{i-1}$ or $t_{i-k} \leftarrow_R t_{i-k+1} \leftrightarrow_A^* t_{i-1}$. The multiset $e(t_{i-1}, t_i, P)$ encodes information about the "environment" of a rewrite step. The symbol *max* denotes a new constant.

The ordering $>_A^c$ is the lexicographic combination of the extension to multisets of the reduction ordering $>$, the proper subterm ordering modulo A, the extension to multisets of the proper subterm ordering modulo A, the subsumption ordering, and the reduction ordering $>$. (The constant *max* is assumed to be maximal.) This ordering is well-founded if and only if the proper subterm ordering modulo A is well-founded. We define $>_A$ by: $P >_A P'$ *if and only if* $M_A(P) \gg_A^c M_A(P')$.

Lemma 3.7. *If the proper subterm ordering modulo A is well-founded, then $\Rightarrow_A$ is a proof ordering.*

Proof. We show that $\Rightarrow_A$ is contained in $>_A$ and therefore is well-founded (by assumption the proper subterm ordering modulo A is well-founded). Let P and P' be proofs for which $P \Rightarrow_A P'$.

If P' is obtained from P by application of an equality pattern, then an equality step $s \leftrightarrow_E t$ is replaced by a number of proof steps that are smaller in the first component of the complexity measure c_A. As a result of the replacement, the complexity of the neighboring steps of $s \leftrightarrow_E t$ may change in the third component. Such changes only pertain to rewrite steps $u \rightarrow_R v$ for which $s \leftrightarrow_A^* u$ or $t \leftrightarrow_A^* u$. Consequently, these proof steps are strictly smaller than $s \leftrightarrow_E t$, and any increase in the third component of their complexity is offset by the deletion of the equality step.

If P' is obtained from P by application of a simplification pattern, then similar arguments as in Lemma 2.6 can be used to show that a rewrite step $s \rightarrow_R t$ is replaced by a sequence of smaller proof steps. (The second, fourth, and fifth components of the complexity measure c_A are needed for this case.) Now, consider a neighboring (rewrite) step $u \rightarrow_R v$ of $s \rightarrow_R t$. If $u \leftrightarrow_A^* t$, then the neighboring step is smaller than $s \rightarrow_R t$ and a possible increase in the complexity of the former is offset by deletion of the latter. On the other hand, if $u \leftrightarrow_A^* s$, then the third component of $c_A(u, v, P)$ can not increase. (The rewrite step $s \rightarrow_R t$ is either replaced by a rewrite step that applies at the same position, or by a sequence of proof steps $s \leftrightarrow_A^* s' \rightarrow_R w$ all of which apply at lower positions.)

Finally, suppose that P' is obtained from P by eliminating a peak. It can easily be checked that the elimination patterns for non-overlaps and for variable

overlaps simplify a proof. The (first and fourth) elimination patterns for overlaps $s \leftarrow_R u \rightarrow_{R'} t$ replace an overlap by a sequence of proof steps in which all terms are strictly smaller than u. In addition, any rewrite step whose complexity changes in the third component must apply to terms smaller than u. Thus, these elimination patterns simplify a proof. There remain the (second, third, and fifth) overlap patterns. (a) A rewrite step $s \rightarrow_R t$ is always smaller (in the second or third component) than an equality step $s \leftrightarrow_A u$ that applies at the same or a higher position. Therefore, in the second overlap pattern, $s \rightarrow_{R'} t$ is smaller than $s \leftrightarrow_A u \rightarrow_L t$. (b) The fifth overlap pattern allows us to replace $s \leftrightarrow_A u \rightarrow_{N \cdot A} t$ by $s \leftrightarrow_A^* v \rightarrow_{R'} w \leftrightarrow_A^* t$. The condition on the latter proof guarantee that all its proof steps are smaller than the equality step $s \leftrightarrow_A u$ (all equality steps in $s \leftrightarrow_A^* v$ apply at positions strictly below $s \leftrightarrow_A u$). (c) In the third pattern, $s \leftarrow_L u \leftrightarrow_A t$ is replaced by $s \leftarrow_{R'} t$. The rewrite steps have equivalent left hand sides (with respect to $\leftrightarrow_A^*$) and furthermore apply at the same position. But the third component of $c_A(s,t,P)$ contains an additional element (corresponding to the equality step $u \leftrightarrow_A t$). Therefore, $u \rightarrow_L s$ is bigger than $t \rightarrow_{R'} s$ in the third component of c_A. (The third component of c_A is only needed for this case, in fact.) •

Lemma 3.8. *Let A be a set of equations such that the proper subterm ordering modulo A is well-founded. A derivation $(E_0,R_0), (E_1,R_1), \cdots$ in A is fair relative to S_A and N_A if (a) $E^\infty=\emptyset$, (b) all critical pairs of L^∞ on R^∞ and all A-critical pairs of N^∞ on R^∞ are contained in $\bigcup_k E_k$, and (c) all critical pairs between L^∞ and A and all A-critical pairs of N^∞ on A are contained in $\bigcup_k R_k$.*

Proof. Let P be a proof in $A \cup E_i \cup R_i$ that contains an instance of a pattern in N_A, i.e. a subproof of the form $s \leftrightarrow_{E_i} t$ or $s \leftarrow_{R_i'} u \rightarrow_{A \cup R_i'} t$. We have to show that P is reducible by $\Rightarrow_A$. If P contains an equality step $u \leftrightarrow_{E_i} v$, then, by

part (a) of the fairness requirement, eventually one of the inference rules M1, M3, or M4 has to be applied, resulting in a proof Q in $A \cup E_j \cup R_j$, for some $j \geq i$, such that $P \Rightarrow_A Q$. Similarly, if P uses a non-persisting rewrite step, application of A1 or A2 will yield a simpler proof Q. Finally, suppose that P uses only persisting rules or equations in A, i.e. is a proof in $A \cup R^\infty$, but contains a peak $v \leftarrow_{A \cup R_i} u \rightarrow_{R_i^A} w$. If this peak is a variable overlap, or not an overlap at all, then it can be eliminated by rearranging its proof steps, as described in the corresponding elimination patterns. On the other hand, parts (b) and (c) of the fairness hypothesis guarantee that all critical pairs necessary for elimination of overlaps are computed. In either case, there is a proof Q in $A \cup E_j \cup R_j$, for some $j \geq i$, such that $P \Rightarrow_A Q$. •

A completion procedure based on A is called fair if it generates only fair derivations (in the sense of Lemma 3.8) unless it fails. As an immediate consequence of Theorem 2.1 and Lemma 3.8, we obtain

THEOREM 3.3. *Let A be a set of equations with a finite complete unification algorithm for which the proper subterm ordering modulo A is well-founded. Let E be a set of equations, $R = L \cup N$ be a rewrite system, and $>$ be a reduction ordering that is compatible with A and contains R. If C is a fair completion procedure and does not fail for inputs E, R and $>$, then $E^\infty = \emptyset$ and $(R^\infty)^A$ is canonical modulo A.*

A system R^A is called *reduced* if, for every rule $l \rightarrow r$ in R, l is irreducible in $(R - \{l \rightarrow r\})^A$ and r is irreducible in R^A. Completion procedures based on the inference system A do not allow construction of reduced systems, in general, since simplification of left-hand sides by $N \cdot A$ at the *top* is not permitted. Thus, a final (canonical) system R^∞ may contain two rules $l \rightarrow r$ and $u \rightarrow v$, for which $l \leftrightarrow_A^* u\sigma$, for some substitution σ. But a reduced canonical system can be easily

obtained (see also Jouannaud and Kirchner, 1986):

PROPOSITION 3.1. *Let R be a (finite) rewrite system and A be a set of equations, such that R^A is canonical modulo A. Let R' be the system obtained from R by deleting one by one any rule $l \to r$ for which there is a rule $u \to v$ in N, distinct from $l \to r$, such that $l \leftrightarrow_A^* u\,\sigma$, for some substitution σ. Then $(R')^A$ is canonical modulo A.*

In contrast with standard rewriting, reduced systems that are canonical modulo a congruence need not be unique with respect to a reduction ordering $>$ when A allows infinite congruence classes. A system R is called *minimal*, if, whenever $l \leftrightarrow_A^* u\,[s\,\sigma]$, for any two (not necessarily distinct) rules $l \to r$ and $s \to t$ in N, then u is a variable and σ is a renaming of variables.

THEOREM 3.4. (Dershowitz, Marcus, and Tarlecki, 1986) *Let A be a set of equations, R and R' be minimal rewrite systems, and $>$ be a reduction ordering that is compatible with A and contains R and R'. If R^A and $(R')^A$ are reduced and canonical modulo A, and the congruence relations $\leftrightarrow_{A \cup R}^*$ and $\leftrightarrow_{A \cup R'}^*$ are the same, then R and R' are identical up to renaming of variables.*

3.3. Protected and Extended Rules

A major limitation of the proof system **A** is that it does not permit simplifications of left-hand sides by $N \cdot A$ at the top. This limitation can be bypassed to a certain extent (at the cost of imposing other restrictions).

Let $v\,\sigma \leftrightarrow_A u\,\sigma \to_{N \cdot A} u\,\sigma[p/r\,\sigma]$ be an A-critical overlap of $l \to r$ on $u = v$ at position p. In the completion procedure proposed by Jouannaud and Kirchner (1986), not the A-critical pair $v\,\sigma = u\,\sigma[r\,\sigma]$ is generated, but rather an equation $w = u\,\sigma[r\,\sigma]$, if there exists a term w such that $v\,\sigma \to_{R^A} w$, or an *extended rule*

$u[p/l] \to u[p/r]$, if $v\sigma$ is irreducible in R^A. This more intricate scheme of applying inference rule M2 can also be viewed as combining creation of a rule $v\sigma \to u\sigma[r\sigma]$ with the simplification (possibly by $N \cdot A$ at the top) of its left-hand side. If possible, a term w is chosen that can be obtained by reducing a proper subterm of $v\sigma$ in R/A, or by reducing $v\sigma$ at the top in R. If $v\sigma$ can only be reduced by $N \cdot A$ at the top, using a rule $l' \to r'$ in N, say, then $l' \to r'$ has to be *protected* from simplification on the left-hand side. An extended rule $u[l] \to u[r]$ is used to reduce $v\sigma$ to $u\sigma[r\sigma]$ in $N \cdot A$; hence has to be protected, too.

Let $R = L \cup N$ be a rewrite system, where some of the rules may be protected. The above schema of applying M2 can be described by an overlap pattern

$$s \leftrightarrow_A u \to_{N \cdot A} t \quad \Rightarrow \quad s \to_{N \cdot A} v \leftrightarrow_E t$$

where the left-hand side denotes an overlap with the second proof step strictly below the first, and $s \to_{N \cdot A} v$ is by application of a protected rule at a position below $s \leftrightarrow_A u$. The set S_P is obtained from S_A by adding this overlap pattern and replacing the simplification patterns by more restricted versions

$$\begin{aligned} s \to_R t \quad &\Rightarrow \quad s \to_{R'} u \leftarrow_{R'/A} t \\ s \to_R t \quad &\Rightarrow \quad s \to_{R'/A} v \leftrightarrow_{E'} t \\ s \to_R t \quad &\Rightarrow \quad s \to_{R'} w \leftrightarrow_{E'} t \end{aligned}$$

in which $s \to_R t$ is by application of an unprotected rule $l \to r$ at position p; $s \to_{R'} u$ is by $l \to r'$ at position p; $s \to_{R'/A} v$ is by application of a rule strictly below p; and $s \to_{R'} w$ is by a rule $l' \to r'$ with $l \rhd l'$ at position p.

The ordering $\Rightarrow_P$ induced by S_P is well-founded if the proper subterm ordering modulo A is. Let P be a proof in $A \cup E \cup R$. So far, we have associated with each proof step in P a measure $c_A(s,t,P)$ of its complexity. Let

$R = L \cup N$ be a rewrite system, N' be the set of all protected rules in N, and R' be $R - N'$. We can translate P into a proof P' in $A \cup E \cup R' \cup N' \cdot A$. Thus, a rewrite step $s \rightarrow_{N' \cdot A} t$ represents a sequence $s \leftrightarrow_A^* s' \rightarrow_{N'} t$ in P, but a single proof step in P'. (We assume that P' contains no subproof $s \leftrightarrow_A u \rightarrow_{N' \cdot A} t$ for which $s \rightarrow_{N' \cdot A} t$. Also, when translating a sequence $s \leftarrow_{N'} u \leftrightarrow_A^* v \rightarrow_{N'} t$ in which all equality steps are below at least one of the rewrite steps, as many equality steps as possible are associated with the rewrite step at a lower position. If both rewrite steps apply at the same position, then all equality steps are associated with the left rewrite step.) The complexity measure c_P is defined with respect to this alternative interpretation of proofs. This implies that certain equality steps $s \leftrightarrow_A t$ in P do not directly contribute to the complexity of a proof.

Let P be a proof in $A \cup E \cup R$ and $P' = (t_0, \ldots, t_n)$ be the corresponding proof in $A \cup E \cup R' \cup N' \cdot A$. The complexity $M_P(P)$ of P is the multiset $\{c_P(t_0, t_1, P'), \ldots, c_P(t_{n-1}, t_n, P')\}$, where $c_P(t_{i-1}, t_i, P')$ is

$$
\begin{array}{ll}
(\{t_{i-1}, t_i\}, -, -, -, -) & \text{if } t_{i-1} \leftrightarrow_E t_i \\
(\{t_{i-1}\}, t_{i-1}/p_i, max, l, t_i) & \text{if } t_{i-1} \leftrightarrow_A t_i \text{ by an equation } l = r \\
(\{t_{i-1}\}, t_{i-1}/p_i, a(t_{i-1}, t_i, P'), l, t_i) & \text{if } t_{i-1} \rightarrow_R t_i \text{ by a rule } l \rightarrow r
\end{array}
$$

Here p_i denotes the position of the i-th proof step in P' and $a(t_{i-1}, t_i, P')$ is defined by: (i) if $t_{i-1} \rightarrow_{N' \cdot A} t_i$ (i.e. by a protected rule) and $t_{i-1} \rightarrow_{N' \cdot A} t_{i-2}$ applies at the same position, then $a(t_{i-1}, t_i, P') = \emptyset$; otherwise $a(t_{i-1}, t_i, P')$ is the multiset $\{(t_{i-1}/p_{i-1}, \delta_{i-1}), (t_{i1}/p_{i1}, 1), \ldots, (t_{ii_k}/p_{ii_k}, 1)\}$, where $(t_{i1}, \ldots, t_{ii_k})$ is the proof in P represented by $t_{i-1} \rightarrow_{N' \cdot A} t_i$ and δ_i is 0, if the i-th proof step in P' is a rewrite step, and 1, otherwise; (ii) if $t_{i-1} \rightarrow_{R'} t_i$ (i.e. by an unprotected rule), then $a(t_{i-1}, t_i, P')$ is the multiset $\{(t_{i-k+1}/p_{i-k+1}, \delta_{i-k+1}), \ldots, (t_{i-1}/p_{i-1}, \delta_{i-1})\}$, where k is the largest index for which $(t_{i-k}, \ldots, t_{i-1})$ is a

proof of the form $t_{i-k} \leftrightarrow_A^* t_{i-1}$, or $t_{i-k} \leftarrow_{R'} t_{i-k+1} \leftrightarrow_A^* t_{i-1}$, or $t_{i-k} \leftarrow_{N' \cdot A} t_{i-k+1} \leftrightarrow_A^* t_{i-1}$.

The ordering $>_A^c$ (with appropriate modifications in the third component) is used to compare elements $c_P(s, t, P)$. We define $>_P$ *by:* $P >_P P'$ *if and only if* $M_P(P) \gg_A^c M_P(P')$.

Lemma 3.9. *If the proper subterm ordering modulo A is well-founded, then $\Rightarrow_P$ is a proof ordering.*

Proof. It can be proved that $P \Rightarrow_P Q$ implies $P >_P Q$, for all proofs P and Q in $A \cup E \cup R$. The complexity of proofs P and Q depends on their respective translations into proofs P' and Q' in $A \cup E \cup R' \cup N' \cdot A$. The assertion can be proved separately for the various elimination patterns. For the equality and simplification patterns the proof is the same as for Lemma 3.7. Elimination of peaks $s \leftarrow_R u \rightarrow_{R^A} t$ is also similar to Lemma 3.7. With elimination of $s \leftrightarrow_A u \rightarrow_{R^A} t$ we distinguish whether $s \leftrightarrow_A t$ is part of a rewrite step $s' \leftarrow_{N' \cdot A} u$ in P', or is a proof step by itself. In the latter case one can proceed as in Lemma 3.7. As an example of the former case, suppose the fifth overlap pattern of S_A is used to replace $s \leftrightarrow_A u \rightarrow_{N \cdot A} t$ by $s \leftrightarrow_A^* v \rightarrow_{R'} w \leftrightarrow_A^* t$. Thus, in P the subproof $s' \leftarrow_{N' \cdot A} u \rightarrow_{N \cdot A} t$ is replaced by a proof $s' \leftarrow_{N' \cdot A} v \rightarrow_{R'} w \leftrightarrow_A^* t$. The sequence $w \leftrightarrow_A^* t$ is smaller than $u \rightarrow_{N \cdot A} t$, since all its terms are smaller than u. Let p be the position of $s \leftrightarrow_A u$. The restrictions on the overlap pattern guarantee that the position q of $v \rightarrow_{R'} w$ is below p, and that all equality steps in $s \leftrightarrow_A^* v$ are strictly below p. (Of course, p must be below the position of $s' \leftarrow_{N' \cdot A} u$.) The rewrite step $s' \leftarrow_{N' \cdot A} v$ is obtained from $s' \leftarrow_{N' \cdot A} u$ by replacing one of its equality steps by a sequence of equality steps that apply at strictly lower positions. Therefore, the former proof step is smaller than the latter in the third component. (The positions of the

respective neighboring steps may also change. But since q is below p this has no effect.) Finally, the rewrite step $v \rightarrow_{R'} w$ is smaller than $s' \leftarrow_{N' \cdot A} u$ either in the second or in the third component. Therefore, $P >_{\mathrm{P}} Q$. Similar reasoning applies to the other cases. •

Lemma 3.10. *Let A be a set of equations for which the proper subterm ordering modulo A is well-founded. A derivation in* **A** *is fair relative to* $\mathrm{S_P}$ *and* $\mathrm{N_A}$ *if (a) $E^\infty = \emptyset$, (b) all critical pairs of L^∞ on R^∞ and all A-critical pairs of N^∞ on R^∞ are contained in $\bigcup_k E_k$, (c) all critical pairs between L^∞ and A are contained in $\bigcup_k R_k$, (d) whenever $l = r$ is an A-critical pair of N^∞ on A, then there is some j, such that either $l \rightarrow r$ is contained in R_j, or $l' = r$ is contained in E_j, where $l \rightarrow_{N_j \cdot A} l'$ by application of a protected rule, and (e) left-hand sides of protected rules are not simplified.*

The proof is essentially the same as for Lemma 3.8. Theorem 1 is valid with the above characterization of fairness.

The completion procedure described in Jouannaud and Kirchner (1986) is a specific version of the proof system **A**. Jouannaud and Kirchner (1986) prove the correctness of this procedure under the assumptions that congruence classes generated by A are finite and that the proper subsumption ordering modulo A is well-founded. The first assumption implies that the subterm ordering modulo A is well-founded, whereas the second is not needed at all.

Rewrite relations other than $R \cdot A$ have also been used for rewriting modulo a congruence. Pedersen (1985) introduces a rewrite relation RA that consists of all pairs (u, v) such that, for some substitution σ and some rewrite rule $l \rightarrow r$ in R, $u \leftrightarrow_A^* l\sigma \rightarrow_R r\sigma = v$, where each equality step applies in the variable part of l. For example, if A is $\{a = b\}$, and R is $\{f(x, x) \rightarrow g(x)\}$, then $f(a, b) \rightarrow_{RA} g(a)$. The relation RA is less general than $R \cdot A$. For instance, if

A is $\{a=b\}$ and R is $\{f(a,a)\rightarrow g(a)\}$, then $f(a,b)$ is irreducible in RA, whereas it reduces to $g(a)$ in $R \cdot A$. Variable overlaps $s \leftrightarrow_A u \rightarrow_R t$ can be regarded as single rewrite steps in RA. Pedersen (1985) hints at a completion procedure for RA in which A-unification is not needed.

3.4. Completion for the Infinite Congruence Class Case

Extended rules were introduced by Peterson and Stickel (1981) in the context of associative-commutative rewriting. Jouannaud and Kirchner (1986) generalized the concept to rewriting modulo a congruence, in general. In this section we present a completion method, based on the systematic use of extended rules, that can be applied to any set of equations A with a finite complete unification algorithm. In particular, it can be used for equational theories that generate infinite congruence classes, e.g. theories with identity, $f(x,e)=x$, or equipotency, $f(f(x))=x$. Such theories can not be handled by any other method.

Let $R=L \cup N$ be a rewrite system, where L contains only left-linear rules. A rule $l \rightarrow r$ in N and equation $u=v$ in A determine an extended rule $u[l] \rightarrow u[r]$, if l is A-unifiable with some proper (non-variable) subterm u/p of u.

Left-hand sides of extended rules must not be simplified. This may preclude construction of fully reduced systems. But extended rules do have advantages. Consider an A-critical overlap $v\sigma \leftrightarrow_A u\sigma \leftrightarrow_A^* u\sigma[l\sigma] \rightarrow_R u\sigma[r\sigma]$ of $l \rightarrow r$ on $u=v$ at position p. The term $v\sigma$ reduces to $u\sigma[r\sigma]$ by application of $u[l] \rightarrow u[r]$. In other words, in the presence of an extended rule, the A-critical pair $v\sigma = u\sigma[r\sigma]$ simplifies to a trivial equation $u\sigma[r\sigma] = u\sigma[r\sigma]$; hence need not be computed in the first place. This argument applies to any A-critical pair of $l \rightarrow r$ on $u=v$ at position p. Therefore, it suffices to compute a single extended

rule, instead of a possibly large set of A-critical pairs. In addition, it is usually more efficient to computed an extended rule than an A-critical pair, since only a test for A-unifiability is required.

Extended rules can be used within the inference system A. But completion procedures based on A are confined to equational theories A for which the proper subterm ordering modulo A is well-founded. We present a different completion method that imposes no such requirement. This method generalizes the associative-commutative completion procedure by Peterson and Stickel (1981) to arbitrary equational theories A.

Let A be a symmetric set of equations and $>$ be a reduction ordering compatible with A. We assume that all rewrite systems R are partitioned into two sets L and N, where L contains only left-linear rules. Rewrite rules (in L or N) may be protected (from simplification on the left-hand side). The inference system E consists of M plus the following simplification rules:

E1) Simplifying a right-hand side of a rule

$$\frac{(E, R \cup \{s \to t\})}{(E, R \cup \{s \to u\})} \qquad \text{if } t \to_{R/A} u$$

E2) Simplifying a left-hand side of a rule

$$\frac{(E, R \cup \{s \to t\})}{(E \cup \{u = t\}, R)} \qquad \text{if } s \to t \text{ is not protected and } s \to_{R/A} u \text{ by a rule } l \to r \text{ with } s \rhd l$$

Here $\rhd$ may be any well-founded ordering on terms, e.g. terms may be compared by their size. The corresponding simplification patterns are

$$s \to_R t \quad \Rightarrow \quad s \to_{R'} u \leftarrow_{R'/A} t$$
$$s \to_R t \quad \Rightarrow \quad s \to_{R'/A} v \leftrightarrow_{E'} t$$

where $s \to_R t$ is by application of a rule $l \to r$ at position p; $s \to_{R'} u$ is by

$l \to r'$ at position p; and $s \to_{R'/A} v$ is by a rule $l' \to r'$ with $l \rhd l'$ at a position below p.

The set S_E is obtained from S_A by replacing the simplification patterns by the patterns above, and replacing the overlap patterns by patterns

$$\begin{array}{rcl}
s \leftarrow_R u \to_L t & \Rightarrow & s \leftrightarrow_{E'} t \\
s \leftrightarrow_A u \to_L t & \Rightarrow & s \to_{R'} t \\
s \leftarrow_L u \leftrightarrow_A t & \Rightarrow & s \leftarrow_{R'} t \\
s \leftarrow_R u \to_{N \cdot A} t & \Rightarrow & s \leftrightarrow_A^* v \leftrightarrow_{E'} w \leftrightarrow_A^* t \\
s \leftrightarrow_A u \to_{N \cdot A} t & \Rightarrow & s \to_{N \cdot A} v \leftrightarrow_A^* t
\end{array}$$

where all left-hand sides denote overlaps with the second step applying below the first step, and all proof steps on right-hand sides apply below the position of $u \to_{A \cup R} s$. In addition, the rewrite steps $s \to_{R'} t$, in the second pattern, $s \leftarrow_{R'} t$, in the third pattern, and $s \to_{N \cdot A} v$, in the last pattern, are by application of a protected rule.

Lemma 3.11. *The set of elimination patterns* S_E *is compatible with the inference system* E.

We next prove that the ordering $\Rightarrow_E$ induced by S_E is well-founded. Let $R = L \cup N$ be a rewrite system. Let N' be the set of all protected rules in N, and R' be $R - N'$. Any proof in $A \cup E \cup R$ can be interpreted as a proof in $A \cup E \cup R' \cup N' \cdot A$. In other words, $s \to_{N' \cdot A} t$ is considered as a single rewrite step, not as an abbreviation for a sequence $s \leftrightarrow_A^* s' \to_{N'} t$. (In case of doubt, equality steps are always associated with a rewrite step that applies at the lower position, or with the rewrite step to the left.) Let P be a proof in $A \cup E \cup R' \cup N' \cdot A$. The ordering $>_E$ is based on the following complexity measure c_E:

if $s \leftrightarrow_E t$, then $c_E(s, t, P)$ is $(\{s, t\}, -, -)$,

if $s \leftrightarrow_A t$, then $c_E(s,t,P)$ is $(\{s\},-,max)$,

if $s \rightarrow_{R^A} t$ by a protected rule $l \rightarrow r$ at position p,

then $c_E(s,t,P)$ is $(\{s\},-,n(s,t,P))$,

if $s \rightarrow_{R^A} t$ by an unprotected rule $l \rightarrow r$ at position p,

then $c_E(s,t,P)$ is $(\{s\},l,t)$,

where $n(s,t,P)$ is the number of equality steps in $s \rightarrow_{R^A} t$, unless the neighboring step of $s \rightarrow_{R^A} t$ in P is of the form $s \rightarrow_{N^I \cdot A} u$ (i.e. by a protected rule in N) and applies at the same or a higher position, in which case $n(s,t,P)=0$. Elements $c_E(s,t,P)$ are compared lexicographically in the multiset ordering $\gg$, the ordering $\rhd$, and a combination of the reduction ordering $>$ and the usual greater-than ordering on natural numbers (more precisely, $max > t > n$, for all terms t and natural numbers n). The ordering $>_E$ is well-founded.

Lemma 3.12. *The relation $\Rightarrow_E$ is a proof ordering.*

Proof. It suffices to prove that $\Rightarrow_E$ is contained in $>_E$. This can be done in a similar way as for Lemma 3.9. We prove the assertion for a representative case, namely for the overlap pattern

$$s \leftrightarrow_A u \rightarrow_{N \cdot A} t \quad \Rightarrow \quad s \rightarrow_{N^I \cdot A} v \leftrightarrow_A^* t .$$

Suppose that P' can be obtained from P by application of this elimination pattern. First note that all terms in $v \leftrightarrow_A^* t$ are smaller than u. Therefore, the sequence $v \leftrightarrow_A^* t$ is smaller than the rewrite step $u \rightarrow_{N \cdot A} t$. If $s \leftrightarrow_A u$ contributes to the complexity of P, then it is bigger than $s \rightarrow_{N^I \cdot A} v$, implying $P >_E P'$. On the other hand, suppose that $s \leftrightarrow_A u$ is part of a rewrite step, i.e. P contains a subproof $w \leftarrow_{N^I \cdot A} u \rightarrow_{N \cdot A} t$. The respective subproof in P' is $w \leftarrow_{N^I \cdot A} s' \rightarrow_{N^I \cdot A} v \leftrightarrow_A^* t$, for some s' with $s' \leftrightarrow_A^* s$. Now, $w \leftarrow_{N^I \cdot A} s'$ is smaller than $w \leftarrow_{N^I \cdot A} u$, since it contains fewer equality steps. Since the rewrite step $s \rightarrow_{N^I \cdot A} v$ applies below $w \leftarrow_{N^I \cdot A} s$, we have $n(w,s,P')=0$. Therefore,

$s \rightarrow_{N' \cdot A} v$ is also smaller than $w \leftarrow_{N' \cdot A} u$, which again implies that P' is simpler than P. •

Lemma 3.13. *A derivation* $(E_0,R_0), (E_1,R_1), \cdots$ *in* E *is fair relative to* S_E *and* N_A *if (a)* $E^\infty=\emptyset$, *(b) all critical pairs of* L^∞ *on* R^∞ *and all A -critical pairs of* N^∞ *on* R^∞ *are contained in* $\bigcup_k E_k$, *(c) all extended rules of* N^∞ *on A and all critical pairs between* L^∞ *and A are contained in* $\bigcup_k R_k$ *and are protected, and (d) left-hand sides of protected rules are never simplified.*

The lemma can be proved in the same way as Lemma 3.8. A completion procedure based on E is fair if it generates only fair derivations (in the sense of Lemma 3.13) unless it fails. As an immediate consequence of Theorem 2.1 and Lemma 3.13 we obtain

THEOREM 3.5. *Let A and E be sets of equations, R be a rewrite system, and* $>$ *be a reduction ordering that contains R and is compatible with A. If C is a fair completion procedure based on* E *and does not fail for inputs E, R and* $>$, *then* $E^\infty=\emptyset$ *and* $(R^\infty)^A$ *is Church-Rosser modulo A.*

The associative-commutative completion procedure by Peterson and Stickel (1981) can be formulated within the inference system E. This procedure applies to sets AC of associativity and commutativity axioms and employs the rewrite relation $R \cdot AC$. For simplification of left-hand sides an ordering $\rhd$ is used in which terms are first compared by size, then with respect to the proper subsumption ordering modulo AC. The only extended rules, originating from rules $f(s,t) \rightarrow u$ with an AC-operator f as outermost symbol on the left-hand side, are $f(x,f(s,t)) \rightarrow f(x,u)$ and $f(f(s,t),x) \rightarrow f(u,x)$, where x is a new variable not appearing in s, t, or u. (Since both rules are equivalent, only one is actually needed. Extensions of extended rules are superfluous.) A large number of canonical systems have been derived with this completion method (e.g. Hullot

1980).

3.5. Examples

Abelian group theory (Peterson and Stickel, 1981). The axioms for free abelian groups are

$$\begin{aligned} x+0 &= x \\ x+(-x) &= 0 \\ x+(y+z) &= (x+y)+z \\ x+y &= y+x \end{aligned}$$

Let R be the system:

$$\begin{aligned} x+0 &\rightarrow x \\ (x+0)+y &\rightarrow x+y \\ -0 &\rightarrow 0 \\ -(-x) &\rightarrow x \\ -(x+y) &\rightarrow (-x)+(-y) \\ x+(-x) &\rightarrow 0 \\ (x+(-x))+y &\rightarrow y \end{aligned}$$

where the operator $+$ is in AC. Then $R_{\cdot AC}$ is canonical modulo AC. (A proof of termination of R/AC is outlined in the next chapter.) The second and the last rule are extended rules.

The following slightly different system was given by Jouannaud and Kirchner (1986):

$$\begin{aligned} x+0 &\rightarrow x \\ 0+x &\rightarrow x \\ -0 &\rightarrow 0 \\ -(-x) &\rightarrow x \\ -(x+y) &\rightarrow (-x)+(-y) \\ x+(-x) &\rightarrow 0 \\ (x+(-x))+y &\rightarrow y \end{aligned}$$

Let L be the set of the first five rules, N the set of the remaining rules. The last rule is an extended rule. Then $L \cup N_{\cdot AC}$ is canonical modulo AC.

Neither of the above systems is reduced. Instead, let R be

$$\begin{array}{rcl} x+0 & \to & x \\ 0+x & \to & x \\ -0 & \to & 0 \\ -(-x) & \to & x \\ -(x+y) & \to & (-x)+(-y) \\ x+(-x) & \to & 0 \\ (x+y)+(-y) & \to & x \end{array}$$

L consist of the first five rules above, and N of the last two rules. The system $L \cup N \cdot AC$ is reduced and canonical.

3.6. Critical Pair Criteria

In the preceding chapter we have described critical pair criteria for standard completion. Similar criteria can be applied to rewriting modulo a congruence. In this section, we generalize the concept of *compositeness* (Kapur, Musser, and Narendran, 1985). For related work see Küchlin (1986b).

Definition 3.1. (a) A critical pair $c = d$ of $l \to r$ on $u \to v$ at position p, with corresponding unifier σ, is called *composite*, if $u\sigma$ is reducible by R/A at a position strictly below p. (b) An A-critical pair $c = d$ of $l \to r$ on $u \to v$ at position p, with corresponding A-unifier σ, is called *composite*, if one of the terms $u\sigma$ or $u\sigma[p/l\sigma]$ is reducible by R/A at a position strictly below p.

Composite critical pairs are redundant for (certain) completion procedures based on the inference system A. Consider, for example, an A-critical overlap

$$P = (v\sigma \leftarrow_R u\sigma \rightarrow_{N \cdot A} u\sigma[p/r\sigma]).$$

If $u\sigma$ reduces to s, at a position strictly below critical pair position p, then the overlap P can be decomposed into two peaks

$$Q = (v\sigma \leftarrow_R u\sigma \rightarrow_{R/A} s \leftarrow_{R/A} u\sigma \rightarrow_{N \cdot A} u\sigma[p/r\sigma]).$$

Now, the first proof step has smaller complexity in Q than in P, because its neighboring rewrite step applies at a lower position. A similar argument applies to the last proof step. The additional proof steps in Q apply at lower positions; hence are less complex. Therefore, we have $P >_A Q$. In other words, any overlap corresponding to a composite A-critical pair of N on R can be simplified. Similar arguments apply to other critical overlaps, but *not* to overlaps involving extended rules. (A different complexity measure is used for extended rules!)

An special case of compositeness is blocking. An A-critical pair $c = d$ is called *blocked* if $x\sigma$ is irreducible in R/A, for all variables x, σ being the A-unifier corresponding to $c = d$. Unblocked critical pairs are composite; hence redundant. Kapur, Musser and Narendran (1985) report that the application of blocking to the associative-commutative completion method of Peterson and Stickel (1981) typically results in considerable savings of computation time. Associative-commutative completion is based on extended rules, however, and the correctness of blocked (or composite) criteria for this case is an open problem.

CHAPTER 4

COMPLETION WITHOUT FAILURE

A completion procedure may fail for various reasons. Any completion procedure must fail for input E, if E can not be represented by a canonical system. Even if E can be characterized by a canonical system R, completion may fail whenever the given reduction ordering $>$ does not contain R, i.e. if l and r are incomparable with respect to $>$, for some rule $l \rightarrow r$ in R. More disturbingly, standard completion may also fail when a reduction ordering containing R is provided as input (see Dershowitz, Marcus, and Tarlecki, 1987). Failure can be avoided by choosing as input a reduction ordering $>$ that is total on congruence classes of E. This requirement is too strong in general, and no such ordering exists, for instance, if E contains a commutativity axiom. In this chapter we show that if standard completion is extended appropriately, then a more reasonable condition on the given reduction ordering suffices for establishing a Church-Rosser property on *ground* (i.e. variable-free) terms. This extension, called "unfailing" completion, can be applied to the construction of canonical rewrite systems and to refutational theorem proving.

4.1. Unfailing Completion

Let E be a set of equations, R be a rewrite system, and $>$ be a reduction ordering containing R. A *ground rewrite proof* relative to $>$ in $E \cup R$ is a proof $u_0 \leftrightarrow_{E \cup R} u_1 \leftrightarrow_{E \cup R} \cdots \leftrightarrow_{E \cup R} u_n$, such that $u_0 > \cdots > u_k < \cdots < u_n$, for

some k, $0 \leq k \leq n$. A ground rewrite proof is a rewrite proof in $R_E \cup R$, where R_E consists of all "orientable instances" $u\sigma \rightarrow v\sigma$ (i.e. $u\sigma > v\sigma$) of equations $u \doteq v$ in E. We say that (E,R) is *ground Church-Rosser relative to* $>$ if, for all ground terms s and t with $s \overset{*}{\leftrightarrow}_{E \cup R} t$, there exists a ground rewrite proof relative to $>$ in $E \cup R$ of the equation $s = t$.

A reduction ordering $>$ is called a *ground reduction ordering for* E if it is total on ground terms equivalent in E. That is, whenever s and t are distinct ground terms such that $s \overset{*}{\leftrightarrow}_E t$, then $s > t$ or $t > s$.

Let $>$ be a ground reduction ordering for E. A ground rewrite proof relative to $>$ can be characterized as a ground proof that contains no subproof of the form $s \leftrightarrow_{E \cup R} u \leftrightarrow_{E \cup R} t$, for which $u > s$ and $u > t$. We denote the set consisting of this "critical" pattern by N_U and use the following overlap (elimination) pattern to simplify such proofs:

$$P = (s \leftrightarrow_{E \cup R} u \leftrightarrow_{E \cup R} t) \Rightarrow (s \overset{*}{\leftrightarrow}_{E' \cup R'} t) = P',$$

where $u > s$, $u > t$, $P >_C P'$, and R and R' are contained in $>$. This overlap pattern allows us, for example, to replace a ground proof $s \leftrightarrow_{E \cup R} u \leftrightarrow_{E \cup R} t$ by an equality step $s \leftrightarrow_{E'} t$ or by a ground rewrite proof $s \overset{*}{\leftrightarrow}_{E' \cup R'} t$. Let S_U be the set consisting of this overlap pattern plus the equality and simplification patterns for standard completion. The corresponding proof relation is denoted by $\Rightarrow_U$.

Lemma 4.1. *The relation* $\Rightarrow_U$ *is a proof ordering.*

Proof. It suffices to prove that $>_C$ contains any instance of an elimination pattern in S_U. For the overlap pattern this follows immediately from the definition; for the equality and simplification patterns, from Lemma 2.6. •

Let $>$ be a reduction ordering. The inference system U (*unfailing completion*) consists of the following *inference rules*, where R is any set of rules contained in $>$:

U1) Orienting an equation.

$$\frac{(E \cup \{s \doteq t\}, R)}{(E, R \cup \{s \to t\})} \qquad \text{if } s > t$$

U2) Adding an equational consequence.

$$\frac{(E, R)}{(E \cup \{s = t\}, R)} \qquad \text{if } s \leftrightarrow_{E \cup R} u \leftrightarrow_{E \cup R} t$$

U3) Simplifying an equation.

$$\frac{(E \cup \{s \doteq t\}, R)}{(E \cup \{u \doteq t\}, R)} \qquad \text{if } s \to_R u$$

U4) Deleting a trivial equation.

$$\frac{(E \cup \{s = s\}, R)}{(E, R)}$$

S1) Simplifying the right-hand side of a rewrite rule

$$\frac{(E, R \cup \{s \to t\})}{(E, R \cup \{s \to u\})} \qquad \text{if } t \to_R u$$

S2) Simplifying the left-hand side of a rewrite rule

$$\frac{(E, R \cup \{s \to t\})}{(E \cup \{u = t\}, R)} \qquad \text{if } s \to_R u \text{ at a position not at the top}$$

$$\frac{(E, R \cup \{s \to t\})}{(E \cup \{u = t\}, R)} \qquad \text{if } s \to_R u \text{ by } l \to r, \text{ where } s \rhd l$$

The difference with standard completion is that both equations *and* rules, and not just rules, may be used to generate equational consequences. The inference system U depends on the reduction ordering $>$, whereas N_U has been defined relative to a *ground* reduction ordering $\succ$. We will therefore assume that unfailing completion is used in combination with a reduction ordering $>$ that can be

extended to a ground reduction ordering $\succ$. Orderings based on polynomial interpretations (Lankford, 1975, 1979) satisfy this requirement. Furthermore, any partial ordering on the set of operator symbols (a *precedence* ordering), can be extended to a ground reduction ordering on terms by way of a *recursive path ordering* (Dershowitz, 1982). (If the precedence ordering is total, then the corresponding recursive path ordering is total on ground terms.)

Let U be the inference system for a ground reduction ordering $>$. Let $\succ$ be a ground reduction ordering containing $>$, and N_U and S_U be the sets associated with $\succ$. We have

Lemma 4.2. *The set of elimination patterns* S_U *is compatible with unfailing completion.*

Lemma 4.3. *A derivation* $(E_0,R_0), (E_1,R_1), \cdots$ *in* U *is fair relative to* S_U *and* N_U *if any critical pair* $c = d$ *of* $E^\infty \cup R^\infty$ *is contained in* $\cup_k E_k$.

Proof. Let $(E_0,R_0), (E_1,R_1), \cdots$ be a derivation in U. We have to show that whenever P is a proof in $E_i \cup R_i$ that contains a critical pattern of N_U, then there exists a proof P' in $E_j \cup R_j$, $j \geq i$, such that $P \Rightarrow_U P'$. Let P be a proof in $E_i \cup R_i$. If P uses a non-persisting equation or rule, then $P \Rightarrow_U P'$, for some proof P'. Since $\Rightarrow_C$ is contained in $\Rightarrow_U$, we have $P \Rightarrow_U P'$. Next suppose that P contains a ground proof $Q = (s \leftrightarrow_{E^\infty \cup R^\infty} u \leftrightarrow_{E^\infty \cup R^\infty} t)$, where $u \succ s, t$. If Q is not a critical overlap, then, by the Critical Pair Lemma, there is a ground rewrite proof Q' (relative to $\succ$) for $s = t$. Thus, $Q \Rightarrow_U Q'$ and, by monotonicity, $P \Rightarrow_U P'$, for some proof P'. If Q is a critical overlap, then $s = t$ must contain an instance of a critical pair $c = d$ of $E^\infty \cup R^\infty$. By fairness, $c = d$ is in E_k, for some k, thus $Q' = (s \leftrightarrow_{E_k} t)$ is a proof in E_k. Again, we have $Q \Rightarrow_U Q'$, and therefore $P \Rightarrow_U P'$, for some appropriate P'. •

The characterization of fairness above can be refined. Since critical pairs of $E^{\infty} \cup R^{\infty}$ are needed for eliminating overlaps $s \leftrightarrow_{E \cup R} u \leftrightarrow_{E \cup R} t$ with $u \succeq s$ and $u \succeq t$, any critical pair coming from an overlap $s \leftrightarrow_{E \cup R} u \leftrightarrow_{E \cup R} t$ for which $s > u$ or $t > u$ can be ignored.

A **U-*completion procedure*** is any procedure that accepts as input a set of equations E, a rewrite system R, and a reduction ordering $>$ containing R. Using the inference rules of unfailing completion as elementary computation steps, it generates a derivation $(E_0,R_0), (E_1,R_1), \cdots$, where E_0 is E, and R_0 is R. A U-completion procedure is fair if it generates only fair derivations. It can easily be seen from Lemma 4.3 that fair derivations can be constructed from any arbitrary pair (E_i,R_i).

THEOREM 4.1. *Let C be a fair* **U-*completion procedure***, *E be a set of equations, R be a rewrite system, and $>$ be a reduction ordering that contains R and can be extended to a ground reduction ordering $\succeq$ for E. Then C, for inputs E, R and $>$, will generate a derivation such that (E^{∞},R^{∞}) is ground Church-Rosser relative to $\succeq$.*

Proof. Let P be any ground proof in $E^{\infty} \cup R^{\infty}$. By Theorem 2.1 and Lemma 4.3, there is a proof P', such that $P \Rightarrow_U^* P'$ and P' contains no critical pattern of N_U. If P' contains no such critical pattern, then it must be a ground rewrite proof. •

4.2. Construction of Canonical Rewrite Systems

We next turn to the problem of constructing reduced canonical rewrite systems. Recall that a canonical rewrite system R is reduced if, for every rewrite rule $l \rightarrow r$ in R, the term r is irreducible in R and l is irreducible in

$R - \{l \to r\}$. Any two reduced canonical systems R and R' that are contained in the same reduction ordering are identical up to renaming of variables—provided, of course, that the congruence relations $\leftrightarrow_R$ and $\leftrightarrow_{R'}$ are the same (e.g. Metivier, 1983).

Let R be a reduced canonical system and $>$ be a reduction ordering containing R. If $l \to r$ is a rule in R, then all proper subterms of l and all terms of which l is a proper instance are irreducible in R. A term t is irreducible in R if and only if it is minimal with respect to $>$ in its congruence class. Every congruence class contains exactly one irreducible (minimal) term. The systems constructed by completion need not be reduced, in general. We may enforce deletion of redundant rewrite rules by imposing a stronger fairness requirement.

Definition 4.1. A derivation $(E_0,R_0), (E_1,R_1), \cdots$ in **U** is *fair for simplification* if (a) all critical pairs of $E^\infty \cup R^\infty$ are contained in $\cup_k E_k$, (b) R^∞ is reduced, (c) whenever $u = v$ is contained in E^∞, then u and v are incomparable with respect to $>$ and irreducible in R^∞.

THEOREM 4.2. *Let R be a reduced canonical system for E, $>$ be a reduction ordering containing R, and C be a* **U***-completion procedure that is fair for simplification. If $>$ can be extended to a ground reduction ordering for E, then C will generate a derivation for inputs E and $>$, such that $E^\infty = \emptyset$ and R^∞ and R are the same up to renaming of variables.*

Proof. Let R be a reduced canonical system for E, and $\succ$ be a ground reduction ordering for E containing $>$. By $\bar{t}$ we denote the skolemized version of a term t. The set $\bar{R}$ consists of all rules $\bar{l} \to \bar{r}$, where $l \to r$ is in R. We extend the ordering $\succ$ to terms containing Skolem constants of $\bar{R}$ by defining: $s \succ t$ *if and only if either* $s\nu \succ t\nu$ *or* $s\nu = t\nu$ *and* $s >' t$, where ν maps all (new) Skolem

constant to some (fixed) constant c of R and $>'$ is an arbitrary ground reduction ordering. A term t is irreducible in R if and only if $\bar{t}$ is. Thus, t is minimal with respect to $\succ$ if and only $\bar{t}$ is.

Now let (E_0,R_0), (E_1,R_1), $\cdots$ be the derivation generated by C for inputs E and $>$, i.e. $E_0=E$ and $R_0=\emptyset$, and let $l\to r$ be a rule in R. We will prove, by induction on $\Rightarrow_U$, that there is a ground rewrite proof of $\bar{l}=\bar{r}$ wherein the first step is a rewrite step. Since $\bar{l}=\bar{r}$ is valid in E, by Theorem 4.1, there is a ground rewrite proof $P=(\bar{l},u_1,\ldots,u_{n-1},\bar{r})$ in $E^\infty\cup R^\infty$, i.e. $\bar{l}\succ\cdots\succ u_k\prec\cdots\prec\bar{r}$. Since $\bar{r}$ is minimal, we actually have $\bar{l}\succ\cdots\succ\bar{r}$. Since all proper subterms of $\bar{l}$ are minimal, the first step in P is by application of an equation $u=v$ in $E^\infty\cup R^\infty$, at the *top*, that is, $\bar{l}$ is $u\sigma$. If $\bar{l}$ is an instance of u, so is l. If l were a proper instance, then u would be minimal. This would imply $v>u$, which contradicts $u\sigma\succ v\sigma$. Therefore, l and u have to be the same up to renaming of variables and we may assume without loss of generality that they are identical. In other words, the system $E^\infty\cup R^\infty$ contains an equation $l=v$.

If l and v are comparable, i.e. $l>v$, then, by fairness for simplification, $l\to v$ must be in R^∞. Thus, the first step in P is a rewrite step. This is in particular true if P consists of only one proof step, since then v is equal to r. If the first proof step is not a rewrite step, then l and v must be incomparable and P must contain at least two proof steps. Since the first proof step applies at the top, the first two steps must overlap. Thus there is a critical pair $c=d$ of $E^\infty\cup R^\infty$, which, by fairness, is contained in $\cup_k E_k$. Then there is a proof P', such that $P\Rightarrow_U P'$ and, consequently, also a ground rewrite proof P'' with $P'\Rightarrow_U P''$. By the induction hypothesis, some ground rewrite proof of $\bar{l}=\bar{r}$ begins with a rewrite step. As we have shown above, this implies that R^∞

contains a rule $l \to v$.

Since R^∞ contains a rule $l \to v$ in R^∞, for every left-hand side l of a rule in R, any term reducible in R is reducible in R^∞. This implies that $E^\infty = \emptyset$ and, since R^∞ is reduced, that $R^\infty = R$. •

There are reduction orderings that can not be extended to a ground reduction ordering. Unfortunately, these also include reduction orderings induced by a canonical reduced rewrite system. For example, the rewrite system $R = \{f(h(x)) \to f(i(x)),\ g(i(x)) \to g(h(x)),\ h(a) \to c,\ i(a) \to c\}$ is canonical and reduced. Any ground reduction ordering for R must contain $h(a) > i(a)$ or $i(a) > h(a)$. If $h(a) > i(a)$, then, by monotonicity, $g(h(a)) > g(i(a))$; but from the second rule in R we infer, by stability, $g(i(a)) > g(h(a))$. A similar contradiction can be derived from the assumption $i(a) > h(a)$.

We will next describe a class of rewrite systems that are contained in a ground reduction ordering. A *reduction sequence* (of length n) is any sequence $t_0 \to_R t_1 \to_R \cdots \to_R t_n$. If R is finite and terminating, then (by König's Lemma) there are only finitely many reduction sequences from any given term t. A reduction sequence is called *innermost* if, for $1 \leq i \leq n$, the reduction step $t_{i-1} \to_R t_i$ applies at a position p_i, where each proper subterm of t_{i-1}/p_i is irreducible in R. If R is finite and reduced, then we denote by $I(t)$ the length of the shortest innermost reduction sequence from t to its normal form t', and define the ordering $>_R^i$ by: $s >_R^i t$ *if and only if* $s \leftrightarrow_R^* t$ *and either* $I(s) > I(t)$ *or* $I(s) = I(t)$ *and* $s \succ t$, where $>$ is the usual "greater-than" relation on the natural numbers and $\succ$ is any ground reduction ordering for R.

Lemma 4.4. *The ordering* $>_R^i$ *is monotonic and contains* R.

Proof. Since R is reduced, we have $I(l)=1$ and $I(r)=0$, for any rule $l \to r$ in R. Therefore, $>_R^i$ contains R. Now suppose that $s >_R^i t$, and let s' be the normal form of s and t in R. Any shortest innermost reduction sequences of $u[s]$ can be rearranged so that s is reduced to s' before any other rewrite steps are applied. In other words, $u[s] \to_R^* u[s'] \to_R^* u'$ is a shortest innermost sequence. Since the corresponding innermost sequence $u[t] \to_R^* u[s'] \to_R^* u'$ is shorter, we have $u[s] >_R^i u[t]$. •

The ordering $>_R^i$ is not stable under substitution. For example, if R is $\{f(x) \to g(x,x,x),\ a \to b\}$, then $f(x) >_R^i g(x,x,x)$, but $f(a) \not>_R^i g(a,a,a)$.

PROPOSITION 4.1. *Let R be a reduced canonical system wherein no instance of a right-hand side is reducible and no variable appears more often in a right-hand side than in the respective left-hand side. Then R is contained in some ground reduction ordering.*

Proof. Let $>$ be the transitive closure of the union of the reduction ordering $\to_R^+$ induced by R and the restriction of $>_R^i$ to ground terms. This ordering is obviously stable and monotonic. For well-foundedness we prove that the restriction of $\to_R^+$ to ground terms is contained in $>_R^i$. It suffices to show that $l\sigma >_R^i r\sigma$, for every rule $l \to r$ in R and every ground substitution σ. Let σ be a ground substitution. Its normalized version σ' assigns to each variable x the normal form of $x\sigma$. Since no variable appears more often in a right-hand side of a rule than in the corresponding left-hand side, no shortest innermost reduction sequence $r\sigma \to_R \cdots \to_R r\sigma'$ can be longer than a shortest innermost sequence $l\sigma \to_R \cdots \to_R l\sigma'$. The term $r\sigma'$ is irreducible, by assumption, whereas $l\sigma'$ is reducible. Thus, $I(l\sigma) > I(r\sigma)$ which implies $l\sigma >_R^i r\sigma$. •

For arbitrary canonical systems we have the following result:

THEOREM 4.3. *Let R be a reduced canonical system for E and $>$ be any reduction ordering contained in $>_R^i$. Then any fair* U*-completion procedure will generate a derivation such that R is contained in $E^\infty \cup R^\infty$.*

Proof. Let $\succ$ be the union of $>$ and the restriction of $>_R^i$ to ground terms. This ordering is a ground reduction ordering. Let $l \to r$ be a rule in R. We show that any proof P of $\bar{l} = \bar{r}$ in $E^\infty \cup R^\infty$ consists of a single step only, or otherwise can be simplified with respect to $\succ_C$. The assertion then follows by induction on $\succ_C$. We may assume that P is a (ground) proof $(\bar{l}, u_1, \ldots, u_{n-1}, \bar{r})$, wherein $\bar{l} \succ \cdots \succ \bar{r}$. The first proof step is by application of an equation $u = v$ at the top, i.e. $\bar{l} = u\sigma$. Consequently, l is an instance of u. First suppose that $I(u) \geq I(v)$. (This we have in particular if $u > v$.) If $I(u) = I(v) = 0$, then both u and v would be irreducible in R, which is impossible. Thus $I(u) > 0$, which implies that u and l are reducible by some rule $l' \to r'$ in R. Since R is reduced, l and l' must be identical. Thus, u is the same as l and v the same as r, up to renaming of variables. In other words, $E^\infty \cup R^\infty$ contains (a variant of) the equation $l = r$. On the other hand, if $I(v) > I(u)$, then v is reducible in R. Therefore P must contain at least two proof steps, of which the first two must overlap. Then P can be simplified with respect to $\succ_C$. •

4.3. Refutational Theorem Proving in Equational Theories

Completion procedures have been primarily used for constructing canonical rewrite systems. Huet (1981) suggested the use of completion for theorem proving in equational theories, and proved that standard completion is a semi-decision procedure for validity in purely equational theories, in cases where it does not fail. We will prove that unfailing completion is a refutationally complete proof

method for equational theories.

Let E be a set of equations and $s = t$ be an equation. Let E^* be the set $E \cup \{eq(x,x) = true,\ eq(\bar{s},\bar{t}) = false\}$, where *true* and *false* are new constants, eq is a new binary operator, and $\bar{s}$ and $\bar{t}$ are skolemized versions of s and t, respectively. The equation $s = t$ is valid in E if and only if $true = false$ is valid in E^*. The inference system U is *refutationally complete* in the sense that a refutation $true = false$ can be derived in U from E^*, whenever $s = t$ is valid in E.

THEOREM 4.4. *Let C be a fair* U*-completion procedure, E be a set of equations, and $>$ be a reduction ordering that can be extended to a ground reduction ordering for E. If $s = t$ is valid in E, then C will generate a refutation $true = false$ for inputs E^* and $>$.*

Proof. We first show that the reduction ordering $>$ can be extended to a ground reduction ordering for E^*. By assumption, $>$ can be extended to a ground reduction ordering for E. We denote this extension by $>$, too, and assume, without loss of generality, that it is also defined on terms containing Skolem constants of $\bar{s}$ and $\bar{t}$. We then extend $>$ to an ordering $\succ$ by defining, for all terms s, t, u, and v not containing the symbols eq, or *true*, or *false*:

(a) $eq(s,t) \succ u \succ true \succ false$; and

(b) $eq(s,t) \succ eq(u,v)$ if and only if $\{s,t\} \gg \{u,v\}$.

Now, $\succ$ is a ground reduction ordering for E^*. Since $true = false$ is valid in E^*, there is, by Theorem 4.1, a ground rewrite proof relative to $\succ$ of $true = false$ in $E_i \cup R_i$, for some i. Since no term is smaller than *true* or *false*, this proof can only be of the form $true \leftrightarrow_{E_i \cup R_i} false$, which implies that the equation $true = false$ is contained in $E_i \cup R_i$. •

Example 4.1. (J. Hsiang). Suppose E consists of

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

To prove that $(x-y)+z=x$ is an equational consequence of E, we derive a refutation from E^*, where E^* is E plus the equations $eq(x,x)=true$ and $eq((a-b)+c,a)=false$. For orienting equations we use the recursive path ordering (see next chapter) corresponding to a precedence ordering in which $-$ is smaller than $+$. In this ordering the second and third equation in E are orientable:

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

The first rule and the remaining equation of E produce a critical pair $(x+z)-y=(x-y)+(x-y)$, which can be simplified to $(x+z)-y=(x+(x-y))-y$. This new equation and the second rule yield a critical pair $x=(x+(x-y))-y$, which can be turned into a rule, and then used for simplifying the original equation to $(x+z)-y=x$. We obtain a new rule $(x+z)-y\rightarrow x$ that can be used to simplify $eq((a+c)-b,a)=false$ to $eq(a,a)=false$. The critical pair of this equation with $eq(a,a)=true$ is a refutation $true=false$.

Unfailing completion is refutationally complete when used with a reduction ordering that is contained in some ground reduction ordering. A corollary of Theorem 4.4 is the refutation completeness of paramodulation, since the latter essentially corresponds to unfailing completion with the "empty" ordering. The advantages of using a non-trivial ordering, as compared to paramodulation, are that (a) equations may be turned into rules and used for simplification, and (b) equational consequences from rules are restricted to those obtained by overlapping left-hand sides. The systematic use of rewriting may thus considerably

reduce the search space of a proof procedure, without destroying refutation completeness.

Though the power of simplification is undisputed, rewrite techniques have not yet been widely employed in (first-order) theorem proving with equality. First attempts at integrating resolution and simplification by rewriting were made by Brown (1975) and Lankford (1975). Peterson (1982) proved completeness of an inference system combining resolution, paramodulation, and simplification with respect to orderings isomorphic to ω on ground terms. Hsiang and Rusinowitch (1986a) show that oriented paramodulation, an inference rule similar to C5 in that it excludes paramodulants obtained by replacing a term by a more complex term, is complete for various resolution strategies. Hsiang and Rusinowitch (1986b) use semantic trees to establish the refutation completeness of an unfailing completion method that extends basic completion, but does not include any of the simplification rules of standard completion. (The completeness proof can be generalized to cover simplification; J. Hsiang, personal communication.) Implementations of completion without failure have been reported by Hsiang and Rusinowitch (1986b) and Ohsuga and Sakai (1986).

CHAPTER 5

TRANSFORMATION ORDERINGS

A rewrite system R that is Church-Rosser and terminating provides a decision procedure for an equational theory E if it is finite, and a semi-decision procedure if it is infinite. In the preceding chapters we have implicitly assumed termination (by using reduction orderings for orienting equations) and showed how to establish the Church-Rosser property. In this chapter, we will deal with the problem of (orderings for) termination in more detail.

While termination of rewrite systems is undecidable, in general (e.g. Huet and Lankford, 1978), a number of techniques have been developed to prove termination of specific rewrite systems, among them the Knuth-Bendix ordering (Knuth and Bendix, 1970), monotonic interpretations (Manna and Ness, 1970), polynomial interpretations (Lankford, 1975, 1979), path of subterms ordering (Plaisted, 1978), recursive path ordering (Dershowitz, 1982), lexicographic and semantic path ordering (Kamin and Levy, 1980), recursive decomposition ordering (Lescanne, 1982), and associative path ordering (Bachmair and Plaisted, 1985). For a survey, with an extensive list of references, see Dershowitz (1985b). We will describe termination methods that are based on transformation techniques, and in particular outline the design of reduction orderings for rewriting modulo a congruence.

5.1. Transformation

A rewrite system is terminating if and only if it is contained in some reduction ordering. Thus, most methods for proving termination consist of finding or constructing an appropriate reduction ordering. It is frequently convenient to separate a reduction ordering $>$ into two parts: a "termination function" τ that maps terms in T to a set W, and a "standard" well-founded ordering $\succ$ on W. In this section, we will consider specific termination functions, called *transforms*, that map terms to terms and can be represented by canonical rewrite systems.

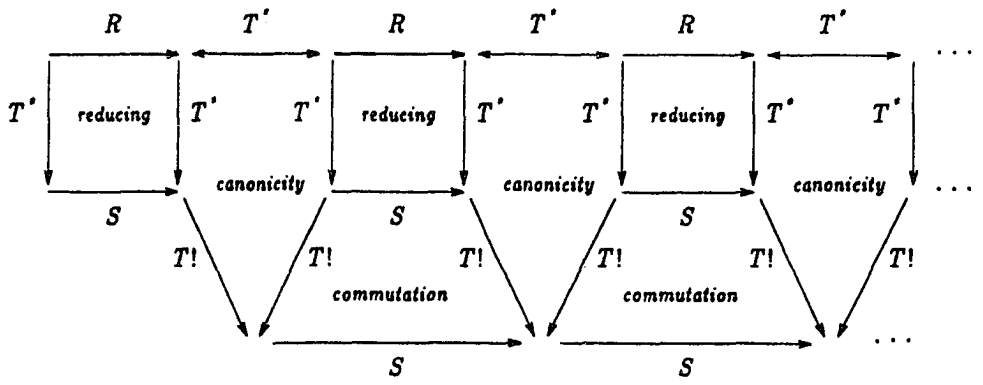
The termination function described by a canonical system T maps a term t to its (unique) normal form t' in T. The system consisting of all rules $t \rightarrow t'$ is denoted by $T!$. We also assume that the ordering $\succ$ is a reduction ordering, and thus can be characterized by some (possibly infinite) rewrite system S. We will derive sufficient conditions for the combination of S and T to be a reduction ordering. For the sake of readability, we will use the symbols R, R^* and R^+ to ambiguously denote the relations $\rightarrow_R$, $\rightarrow_R^*$ and $\rightarrow_R^+$, respectively, and $R^{\leftrightarrow}$ to denote $\leftrightarrow_R^*$.

We say that two rewrite systems R and S *commute* if $\leftarrow_R \circ \rightarrow_S$ is contained in $\rightarrow_S \circ \leftarrow_R$. A system R is *reducing relative to S and T* if it is contained in $T^* \circ S \circ (T^*)^{-1}$, i.e. if $l \rightarrow_T^* \circ \rightarrow_S \circ \leftarrow_T^* r$, for every rule $l \rightarrow r$ in R.

THEOREM 5.1. *Let R, S, and T be rewrite systems such that T is canonical, S terminates, and S and T! commute. If R is reducing relative to S and T, then* $R / T^{\leftrightarrow}$ *terminates.*

Proof. If $R / T^{\leftrightarrow}$ is not terminating, then there is an infinite sequence $t_1 \rightarrow_R t_2 \leftrightarrow_T^* t_3 \rightarrow_R t_4 \leftrightarrow_T^* \cdots$. Using the facts that R is reducing, T is canonical, and S and $T!$ commute, we can construct an infinite sequence

$u_1 \to_S u_2 \to_S u_3 \to_S \cdots$:



This contradicts the fact that S is terminating. •

COROLLARY 5.1. *Let R, S, T, and T' be rewrite systems such that T is canonical, S terminates, and S and $T!$ commute. If T' is contained in $T^{\leftrightarrow}$ and R is reducing relative to S and T, then R/T' terminates.*

Let S and T be rewrite systems, such that S terminates and T is canonical. The *transformation ordering* $>_T^S$ is defined by: $u >_T^S v$ *if and only if* $u \to_{T!} \circ \to_S^+ \circ \leftarrow_{T!} v$. *It can easily be seen that* $>_T^S$ *is transitive and irreflexive, and hence a (strict partial) ordering. The ordering is well-founded, since S is terminating.*

Lemma 5.1. *If S and $T!$ commute, then $>_T^S$ is a reduction ordering.*

Proof. Let $>$ denote the transformation ordering $>_T^S$. For monotonicity and stability it suffices to show that $s > t$ implies $c[s\sigma] > c[t\sigma]$, for all terms s, t, and c, and all substitutions σ. Suppose that $s > t$, i.e. $s \to_{T!} u \to_S^+ v \leftarrow_{T!} t$. Then, for any substitution σ and term c,

$$c[s\sigma] \to_T^* c[u\sigma] \to_S c[v\sigma] \leftarrow_T^* c[t\sigma].$$

Denoting by u' and v' the normal forms in T of $c[s\sigma]$ and $c[t\sigma]$, we have

$c\,[u\,\sigma]\to_{T!}u'$ and $c\,[v\,\sigma]\to_{T!}v'$. Since S and $T!$ commute, so do S^+ and $T!$. Thus we have $u'\to_S^+ v'$, which implies $c\,[s\,\sigma]>c\,[t\,\sigma]$. •

Commutation is essential for $>_T^S$ to be a reduction ordering. On the other hand, we have

Lemma 5.2. *If T is canonical and $>_T^S$ is a reduction ordering containing S, then S^+ and $T!$ commute.*

Proof. Suppose that $u\leftarrow_{T!}c\,[s\,\sigma]\to_S c\,[t\,\sigma]$, where $s\to t$ is in S. From $s>_T^S t$ we may infer, by monotonicity and stability, $c\,[s\,\sigma]>_T^S c\,[t\,\sigma]$. That is, we have $c\,[s\,\sigma]\to_{T!}u\to_S^+ v\leftarrow_{T!}c\,[t\,\sigma]$, which implies commutation of S^+ and $T!$. •

Let us consider specific transforms. For termination proofs *symbolic interpretations* of operators are often useful. These consist of a single rewrite rule $f\,(x_1,\ldots,x_n)\to t\,[x_1,\ldots,x_n]$, where t contains all variables $x_1,\ldots,x_n$, but not f. Such transforms are canonical. They may be used, for instance, to declare two operators equivalent (for the purpose of proving termination). The *T-normalized version* R_T of R consists of all rules $l'\to r'$, where l' and r' are normal forms of l and r in T, respectively, for some rule $l\to r$ in R.

Lemma 5.3. *Let R be a rewrite system, T be a symbolic interpretation, and R_T be the T-normalized version of R. Then R is reducing relative to R_T and T, and R_T^+ and $T!$ commute.*

Proof. By the definition of R_T, R is reducing relative to R_T and T. For commutation, suppose that $c\,[l\,\sigma]\to_{R_T}c\,[r\,\sigma]$, for some rule $l\to r$ in R_T, some substitution σ and some term c. We have to show that $u\to_{R_T}^+ v$, where u and v are normal forms of $c\,[l\,\sigma]$ and $c\,[r\,\sigma]$ in T, respectively. Let σ' be the "normalized version" of σ, i.e. $x\,\sigma$ is the normal form of $x\,\sigma'$, for all variables x, and let

$d[x, \ldots, x]$ be the normal form of $c[x]$ in T. A normal form of $c[l\sigma]$ can be computed as follows:

$$c[l\sigma] \to_T^* c[l\sigma'] \to_T^* d[l\sigma', \ldots, l\sigma'] = u.$$

Similarly, we have

$$c[r\sigma] \to_T^* c[r\sigma'] \to_T^* d[r\sigma', \ldots, r\sigma'] = v.$$

Neither u nor v contain the operator f, hence are irreducible. In addition, we have $u \to_{R_T}^+ v$, which completes the proof. •

Combining Theorem 5.1 and Lemma 5.3, we obtain

PROPOSITION 5.1. *Let R be a rewrite system and T be a symbolic interpretation. The system $R/T^{\leftrightarrow}$ is terminating if R_T is.*

Example 5.1. Let R be

$$\begin{array}{rcl} g(x,y) & \to & h(x,y) \\ h(f(x),y) & \to & f(g(x,y)) \end{array}$$

We use the first rule as a transform T and let R' be the second rule. The T-normalized version R_T' of R' is

$$h(f(x),y) \to f(h(x,y))$$

R_T' terminates, since it decreases the summed length of all the terms with outermost operator h. By Proposition 5.1, R'/T is terminating. Since T is also terminating, so is $R = R' \cup T$.

The termination methods outlined above may be applied to rewrite systems R/E by using transforms T, such that E is contained in $T^{\leftrightarrow}$.

If I consists of the axioms for identity, $f(x,e) = x$ and $f(e,x) = x$, then we may use a transform $T_I = \{f(x,e) \to x, f(e,x) \to x\}$. This transform is canonical, and $T_I^{\leftrightarrow}$ contains I. Let Σ_I be the set of all substitutions σ, such that $x\sigma$ is x or e, for all variables x. Note that the composition of two

substitutions in Σ_I again yields a substitution in Σ_I.

Let R be a rewrite system and R' be the set of all rules $l\sigma \to r\sigma$, where $l \to r$ is in R and σ is in Σ_I. We define R_I' as the T_I-normalized version of R_I'. If $l \to r$ is in R_I' and σ is in Σ_I, then the T_I-normalized version of $l\sigma \to r\sigma$ is also in R_I'. Now, let R_I be R_I' plus the following additional rules (necessary for commutation of R_I and T_I!): for every rule $e \to r$ in R_I', where $r \neq e$, rules $x \to f(x,r)$ and $x \to f(r,x)$; for every rule $l \to e$ in R_I', where $l \neq e$, rules $f(x,l) \to x$ and $f(l,x) \to x$; and the rule $x \to x$, if $e \to e$ is in R_I'.

Lemma 5.4. *Let R be a rewrite system and T_I and R_I be as defined above. Then R is reducing relative to T_I and R_I, and R_I and T_I! commute.*

Proof. R is reducing relative to T_I and R_I, since R_I contains all T_I-normalized versions of rules in R. For commutation, suppose that $c[l\sigma] \to_{R_I} c[r\sigma]$, for some rule $l \to r$ in R_I, some substitution σ, and some term c. Let u and v be normal forms in T_I of $c[l\sigma]$ and $c[r\sigma]$, respectively. We have to show that $u \to_{R_I} v$.

We may assume, without loss of generality, that $c[x]$ is irreducible and that σ is normalized, i.e. $x\sigma$ is irreducible, for all variables x. Let σ' be a substitution in Σ_I such that $x\sigma'$ is e if $x\sigma$ is e, and $x\sigma'$ is x, otherwise. Then the T_I-normalized version $l' \to r'$ of $l\sigma' \to r\sigma'$ is contained in R_I. Also, σ is the composition of σ' and some substitution ρ, for which $x\rho \neq e$, for all variables x. Since neither l' nor r' contain a subterm $f(x,e)$ (or $f(e,x)$), $l'\rho$ and $r'\rho$ contain no such subterm either. In other words, both $l'\rho$ and $r'\rho$ are irreducible in T_I. Now, if both $c[l'\rho]$ and $c[r'\rho]$ are irreducible, then $u = c[l'\sigma]$ and $v = c[r'\sigma]$, and since $l' \to r'$ is in R_I, we have $u \to_{R_I} v$. If $c[l'\rho]$ is reducible, then l' must be e and $c[x]$ can be written as $d[f(x,t)]$ (or $d[f(t,x)]$, where t is some term different from e. A similar argument applies to $c[r'\rho]$. Let us

consider the various possibilities.

(a) If $l' = e = r'$, then R_I contains a rule $x \to x$. Since $c[l'\sigma]$ and $c[r'\sigma]$ are identical, so are their respective normal forms u and v, and we have $u \to_{R_I} v$.

(b) If $l' = e \neq r'$, then R_I contains a rule $x \to f(r', x)$ (or $x \to f(x, r')$). Applying this rule we obtain $u = d[t] \to_{R_I} d[f(r', s)] = v$.

A similar argument applies if $l' \neq e = r'$. •

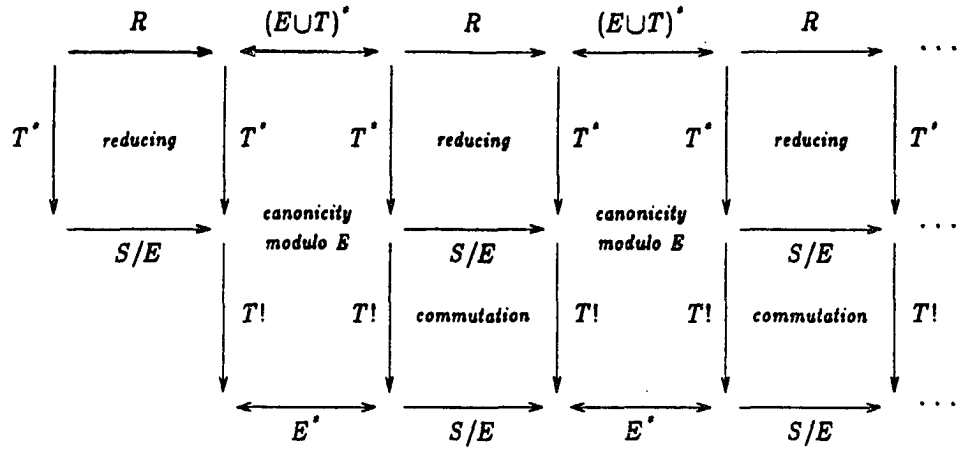
PROPOSITION 5.2. *An equational system R/I terminates if and only if R_I terminates.*

Proof. The if-direction follows from Theorem 5.1 and Lemma 5.4. The only-if-direction holds because R_I is contained in R/I. •

Transforms need not necessarily be canonical, but may be canonical modulo some set of equations E. Such transforms are particularly useful for, but not limited to, proving termination of systems R/E. Recall that if R is canonical modulo E, then two terms are equivalent in $E \cup R$ if and only if their respective normal forms in R are equivalent in E.

THEOREM 5.2. *Let R, S, and T be rewrite systems and E be an equational theory, such that T is canonical modulo E, S/E is terminating, and S and $T!$ commute. If R is reducing relative to S/E and T, then $R/(E \cup T^{\leftrightarrow})$ terminates.*

Proof. Let $t_1 \to_R t_2 \overset{*}{\leftrightarrow}_{E \cup T} t_3 \to_R t_4 \overset{*}{\leftrightarrow}_{E \cup T} \cdots$ be an infinite sequence. If T is canonical modulo E, and S and $T!$ commute, then S/E and $T!$ also commute. An infinite sequence of S/E reduction steps can be constructed as follows:



This contradicts termination of S/E. •

In the next section we will consider transforms for associative-commutative rewrite systems in depth.

5.2. Transforms Based on Distributivity

Equational rewrite systems R/E, where E is a set of associativity and commutativity axioms, are of particular importance in practice. We will apply the transformation techniques outlined above to the termination problem for such systems (*AC termination*).

Let f be some operator symbol in F. An *associativity* axiom is an equation of the form $f(x,f(y,z))=f(f(x,y),z)$ or $f(f(x,y),z)=f(x,f(y,z))$. A *commutativity* axiom is an equation $f(x,y)=f(y,x)$. An equational rewrite system R/E is called *associative-commutative* if E contains only associativity and commutativity axioms. From now on let AC denote a set of associativity and commutativity axioms for which any associative operator is also commutative and vice versa. We say that f is in AC to indicate that f is an

associative-commutative operator.

A reduction ordering $>$ is compatible with AC if $s \leftrightarrow^*_{AC} u > v \leftrightarrow^*_{AC} t$ implies $s > t$, for all terms s, t, u, and v. A rewrite system R/AC terminates if and only if there is a reduction ordering $>$ that is compatible with AC and contains R. We will show how to combine a standard ordering—the recursive path ordering—with an appropriate transform to obtain a reduction ordering that is compatible with AC.

The *permutation congruence* $\sim$ is the smallest stable congruence such that $f(X,u,Y,v,Z) \sim f(X,v,Y,u,Z)$. Let $>$ be an ordering, called a *precedence*, on the set of operator symbols F. The *recursive path ordering* $>_{rpo}$ (Dershowitz, 1982) is defined as follows:

$$s = f(s_1, \ldots, s_m) >_{rpo} g(t_1, \ldots, t_n) = t$$

if

(a) $s_i >_{rpo} t$ or $s_i \sim t$, for some i, $1 \leq i \leq m$; or

(b) $f = g$ and $\{s_1, \ldots, s_m\} \gg_{rpo} \{t_1, \ldots, t_n\}$; or

(c) $f > g$ and $s >_{rpo} t_j$, for all j, $1 \leq j \leq n$.

Lemma 5.5. (Dershowitz, 1982) *Let $>$ be a precedence ordering on the set of operator symbols F. The recursive path ordering is well-founded if and only if $>$ is well-founded.*

Unfortunately, the recursive path ordering $>_{rpo}$ is not compatible with AC. For example, if f is in AC and $a > b$, then

$$f(a,f(b,b)) \leftrightarrow_{AC} f(f(a,b),b) >_{rpo} f(a,f(b,b)),$$

but $f(a,f(b,b)) >_{rpo} f(a,f(b,b))$ is false.

If we have a transform that selects a unique representative (canonical form) from each AC-congruence class, then we can easily get a compatible ordering by

comparing canonical representatives instead of terms themselves. A natural choice for such a canonical representation are "flattened" terms.

A *flattening rule* for an operator f is a rule $f(X,f(Y),Z)\rightarrow f(X,Y,Z)$, where Y denotes a sequence of variables $y_1,\ldots,y_n$ of length $n\geq 2$, and X and Z are sequences of variables of length k and l, respectively, where $k+l\geq 1$. When using flattening rules, we have to regard operators as *varyadic*. However, flattening rules do not apply to unary operators or constants. For example, $f(x,f(y,z))\rightarrow f(x,y,z)$ is a flattening rule, but $f(f(x))\rightarrow f(x)$ is not. Let F_L be a set of operators and L be the set of all flattening rules for operators in F_L. The system L is canonical; terms irreducible in L are called *flattened.*

Consider the transformation ordering $>_L$, defined by: $s>_L t$ *if and only if* $s\rightarrow_L{}!u>_{rpo} v\leftarrow_L{}!t$. This ordering is not monotonic, as the following example illustrates. If f is in AC and $f>g$, then $f(a,b)>_L g(a,b)$. The term $f(f(a,b),c)$ flattens to $f(a,b,c)$ and $f(g(a,b),c)$ is already flattened. We have $f(g(a,b),c)>_L f(a,b,c)$, instead of the opposite!

We have to enrich our transform in order to get a reduction ordering. It suffices, in fact, to include *distributivity rules*

$$f(X,g(Y),Z)\rightarrow g(f(X,y_1,Z),\cdots,f(X,y_n,Z)),$$

for certain operators f and g (Y is a sequence $y_1,\ldots,y_n$ of length $n\geq 1$). For instance, $x*(y+z)\rightarrow x*y+x*z$ and $-(x+y)\rightarrow(-x)+(-y)$ are distributivity rules. Sets of distributivity rules are not canonical, in general, but only under certain conditions. For example, if f distributes over both g and h, then the term $f(g(x),h(y))$ can be transformed to $g(h(f(x,y)))$ or $h(g(f(x,y)))$.

Let F be a given set of operators and F_D be a subset of F that contains all AC operators and no constants. Let $>$ be a precedence ordering on F; D be the set of all distributivity rules for operators f and g in F_D with $f>g$; and

L be the set of all flattening rules for operators in F_D. We call the rewrite system $T = L \cup D$ the *A-transform* corresponding to $>$ and F_D.

Definition 5.1. A precedence $>$ satisfies the *associative path condition* with respect to F_D if, for every operator f in F_D,

(a) $f > g$ implies that g is in F_D;

(b) at most two operators are smaller than f;

(c) $f > g$ and $f > h$ implies $g > h$ or $h > g$;

(d) $f > g > h$ implies that g is unary;

(e) for all g and h in F_D, $g > f$ and $h > f$ implies $g > h$ or $h > g$.

For example, if f, g, and h are in F_D, then the precedence ordering in Figure 5.1(a) satisfies the associative path condition. The precedence in Figure 5.1(b) only satisfies the condition if g is unary. The orderings in Figures 5.1(c) and (d) violate the requirements (c) and (e), respectively.

If a precedence satisfies the associative path condition, then F_D can be partitioned into (totally ordered) sets $F_1, \ldots, F_n$, such that any two operators from different sets are incomparable. Conditions (b), (c) and (d) guarantee that the

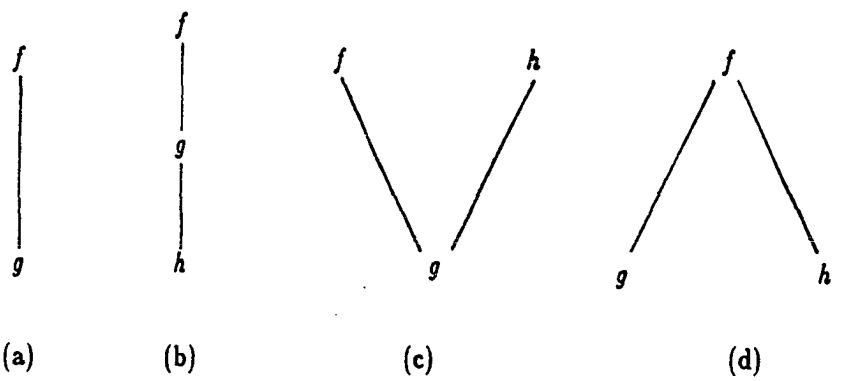


Figure 5.1.

transform is canonical modulo $\sim$. Condition (a) is necessary for monotonicity of the corresponding transformation ordering; condition (e), for stability.

The congruence generated by AC is contained in $\sim \cup T^{\leftrightarrow}$, since (i) $f(x,y) \sim f(y,x)$ and (ii) both $f(x,f(y,z))$ and $f(f(x,y),z)$ flatten to $f(x,y,z)$. Furthermore, we have

Lemma 5.6. *If the precedence $>$ satisfies the associative path condition with respect to F_D, then the corresponding A-transform T is canonical modulo $\sim$.*

Proof. Let $>$ be a precedence that satisfies the associative path condition with respect to F_D. The system $T/\sim$ is terminating, since the recursive path ordering $>_{rpo}$ contains T and is compatible with the permutation congruence. Thus, to prove that T is Church-Rosser modulo $\sim$ it suffices to show that, for every critical overlap $c \leftarrow_T u \rightarrow_T d$ or $c \leftarrow_T u \sim d$, there exist terms v and w, such that $c \rightarrow_T^* v \sim w \leftarrow_T^* d$. For overlaps between T and $\sim$ this holds since T and $\sim$ commute, i.e. $c \leftarrow_T u \sim d$ implies $c \sim v \leftarrow_T d$. Let us consider overlaps between rules of T. For instance, we have an overlap

$$c = g(f(X,h(Y),Z)) \leftarrow_T f(X,g(h(Y)),Z)$$
$$\rightarrow_T f(X,h(g(y_1,\ldots,g(y_n)),Z) = d,$$

if f, g and h are in F_D with $f > g > h$ (because of the associative path condition, g must be unary). This overlap can be resolved since both c and d reduce to $h(g(f(X,y_1,Z)),\ldots,g(f(X,y_n,Z)))$ in T. Other overlaps can be handled similarly. •

Definition 5.2. Let $>$ be a well-founded precedence that satisfies the associative path condition with respect to F_D, and let T be the A-transform corresponding to $>$ and F_D. The *associative path ordering* $>_{apo}$ is defined by:

$$s >_{apo} t \text{ if and only if } u >_{rpo} v,$$

where u and v are normal forms of s and t in T, respectively.

Lemma 5.7. *The associative path ordering is monotonic with respect to the term structure.*

Proof. We have to show that $s >_{apo} t$ implies $c[s] >_{rpo} c[t]$, for all terms s, t, and c. We may assume without loss of generality that s, t, and all proper subterms of $c[x]$ are irreducible in T, and that $c[x]$ is of the form $f(\cdots x \cdots)$. Let s be $g(s_1, \ldots, s_m)$ and t be $h(t_1, \ldots, t_n)$ and let u and v denote $c[s]$ and $c[t]$, respectively.

a) If f is not in F_D, then both $f(\cdots s \cdots)$ and $f(\cdots t \cdots)$ are in normal form, and $u >_{apo} v$ follows from the monotonicity of the recursive path ordering.

b) If f is a minimal non-unary operator in F_D, then no distributivity rule applies to u or v. Therefore the respective flattened versions $\bar{u}$ and $\bar{v}$ are irreducible in T. If $g \neq f$, then $\bar{u}$ is $\bar{c}[s]$ and, since $\bar{c}[s] >_{rpo} \bar{c}[t] \geq_{rpo} \bar{v}$, we have $u >_{apo} v$. If $g = f$, then $\bar{u}$ is $f(\ldots, s_1, \ldots, s_m, \ldots)$. Now, if $g \not\geq h$, then $s_i >_{rpo} t$, for some i, $1 \leq i \leq m$. Since any term with operator symbol f must have at least two subterms, i.e. $m \geq 2$, we obviously have $\bar{u} >_{rpo} \bar{v}$. On the other hand, if $g = h$, then $\{s_1, \ldots, s_m\} \gg_{rpo} \{t_1, \ldots, t_n\}$. Thus

$$\bar{u} = f(\ldots, s_1, \ldots, s_m, \ldots) >_{rpo} f(\ldots, t_1, \ldots, t_m, \ldots) = \bar{v}.$$

Since g is minimal, $g > h$ is impossible.

c) Next suppose that f is a unary operator. We certainly have $f(s) >_{apo} f(t)$ if $f(s)$ is irreducible. Now, the term $f(s)$ is reducible only if $f > g$. Then, because of the associative path condition, g must be in F_D and minimal. Thus $f(s)$ has the normal form $u' = g(f(s_1), \ldots, f(s_m))$ in T. If $s_i \geq_{rpo} t$, for some i, then $u' >_{rpo} f(s_i) \geq_{rpo} f(t)$, which implies $u >_{apo} v$. If $g = h$, then $\{s_1, \ldots, s_m\} \gg_{rpo} \{t_1, \ldots, t_n\}$ and $f(t)$ reduces to

$g(f(t_1), \ldots, f(t_n))$. Since $\{f(s_1), \ldots, f(s_m)\} \gg_{rpo} \{f(t_1), \ldots, f(t_n)\}$, we have $u >_{apo} v$.

d) Finally, suppose that f is neither minimal nor unary. We prove this case by induction on the combined size of $c[x]$, s, and t. First suppose that $c[x] = f(\cdots x \cdots)$ is reducible in T. If the flattened version $\bar{c}[x]$ is shorter than $c[x]$, we may apply the induction hypothesis to $\bar{c}[x]$, s and t and obtain $\bar{c}[s] >_{apo} \bar{c}[t]$, which proves this case. Next suppose that $c[x]$ contains a top-level subterm $f'(u_1, \ldots, u_k)$, where $f > f'$. Then the term

$$c[f'(u_1, \ldots, u_k), x] = f(\ldots, f'(u_1, \ldots, u_k), \ldots, x, \ldots)$$

reduces to

$$f'(c[u_1, x], \ldots, c[u_k, x]).$$

Applying the induction hypothesis to $c[u_i, x]$, s, and t, we obtain

$$c[u_i, s] >_{apo} c[u_i, t],$$

for all i, $1 \leq i \leq k$. Because of the associative path condition, f' must be unary or minimal. Thus, by (b) and (c) above,

$$f'(c[u_1, s], \ldots, c[u_k, s]) >_{apo} f'(c[u_1, t], \ldots, c[u_k, t]),$$

which proves this case.

The only remaining case is that $c[x]$ is irreducible. The assertion trivially holds if $c[s]$ is irreducible. Let us therefore assume that $f \geq g$.

(i) If $g \not\geq h$, then $s_i \geq t$, for some i, $1 \leq i \leq n$. By induction, we get $c[s_i] >_{apo} c[t]$. Now, if $f = g$, then $c[s]$ reduces to $f(\ldots, s_1, \ldots, s_m, \ldots)$ in T Since $m \geq 2$, we have $c[s] >_{apo} c[s_i]$, and thus $c[s] >_{apo} c[t]$. If $f > g$, then $c[s]$ reduces to $g(c_1, \ldots, c_m)$, where c_i is the normal form of $c[s_i]$ in T. Thus we have $c[s] >_{apo} c[s_i] \geq_{apo} c[t]$.

(ii) If $g = h$, then $\{s_1, \ldots, s_m\} \gg_{rpo} \{t_1, \ldots, t_n\}$. If $f = g$, then $\bar{u} = f(\ldots, s_1, \ldots, s_m, \ldots) >_{rpo} f(\ldots, t_1, \ldots, t_m, \ldots) = \bar{v}$ and, since both terms are irreducible, we obtain $u >_{apo} v$. If $f > g$, then $c[s]$ reduces to

$g(c_1, \ldots, c_m)$, where c_i is the normal form of $c[s_i]$ in T, and $c[t]$ reduces to $g(c_1', \ldots, c_n')$, where c_j' is the normal form of $c[t_j]$. Now, if $s_i >_{rpo} t_j$, then $c[s_i] >_{apo} c[t_j]$, by the induction hypothesis applied to $c[x]$, s_i, and t_j. Thus $\{c_1, \ldots, c_m\} \gg_{rpo} \{c_1', \ldots, c_n'\}$, which implies $c[s] >_{apo} c[t]$.

(iii) Finally, suppose that $g > h$ and $s >_{rpo} t_i$, for $1 \leq i \leq n$. Then $c[t]$ reduces to $h(f(\ldots t_1 \ldots), \ldots, f(\ldots t_n \ldots))$. Applying the induction hypothesis to $c[x]$, s, and t_i, we get $u >_{apo} f(\cdots t_i \cdots)$, for $1 \leq i \leq n$. Let u' be the normal form of u in T. Since $op(u') \geq g > h$, we conclude that $u' >_{apo} h(f(\ldots t_1 \ldots), \ldots, f(\ldots t_n \ldots))$, i.e. $u >_{apo} v$. •

Lemma 5.8. *The associative path ordering is stable under substitution.*

Proof. We have to show that $s >_{apo} t$ implies $s\sigma >_{apo} t\sigma$, for all substitutions σ. We may assume that s and t are irreducible and that, for some variable x, $x\sigma$ is $h(x_1, \ldots, x_n)$, where $k \geq 0$ and $x_1, \ldots, x_k$ are new variables, and $y\sigma$ is y, for $y \neq x$. (Any arbitrary substitution can be composed of substitutions such as σ.) The proof is by induction on the combined size of s and t. Let s be $f(s_1, \ldots, s_m)$ and t be $g(t_1, \ldots, t_n)$.

a) If $f \not\geq g$, then $s_i \geq_{rpo} t$, for some i, $1 \leq i \leq n$. By the induction hypothesis, $s_i\sigma \geq_{apo} t\sigma$. The assertion then follows from the following subterm property: $f(\cdots s \cdots) >_{apo} s$, *for all terms* $f(\cdots s \cdots)$ *and* s. The proof of this property is not difficult, but rather technical, and is omitted.

b) If $f = g$, then $M = \{s_1, \ldots, s_m\} \gg_{rpo} \{t_1, \ldots, t_n\} = N$. We may assume that N is $(M - \{s_i\}) \cup \{s_{i1}, \ldots, s_{ik}\}$, where $k \geq 0$ and $s_i >_{rpo} s_{ij}$, for $1 \leq j \leq k$. (The general case can be obtained by repeated application of the elementary one.) By the induction hypothesis, $s_i\sigma >_{rpo} s_{ij}\sigma$, for $1 \leq j \leq k$. If $k = 1$, then $s\sigma >_{apo} t\sigma$ follows by the monotonicity of the associative path ordering. Let us assume that $k > 1$. Then f must not be unary. If $op(s_i) > f$, then

$s_i >_{rpo} f\ (s_{i1}, \ldots, s_{ik})$ and, by induction, $s_i\sigma >_{rpo} f\ (s_{i1}\sigma, \ldots, s_{ik}\sigma)$. Again, the assertion follows by monotonicity. Finally, suppose that $op(s_i) \not> f$. Since s is irreducible, we have $f \geq op(s_i)$. Because of the associative path condition, f and $op(s_i)$, and consequently f and $op(s_i\sigma)$, have to be incomparable. Therefore, transforming $s\sigma$ does not change $s_i\sigma$, so that we can infer without much difficulty that $s\sigma >_{apo} t\sigma$.

c) If $f > g$, then $s >_{rpo} t_j$, for all j, $1 \leq j \leq n$. By the induction hypothesis, $s\sigma >_{apo} t_j\sigma$, for $1 \leq j \leq m$. If $op(s\sigma) > g$, then $s\sigma >_{apo} g(t_1\sigma, \ldots, t_n\sigma) >_{apo} t\sigma$. Let us therefore assume that $g \geq h = op(s\sigma)$. Thus, f, g, and h are all in F_D and $f > g \geq h$. There are various possibilities, according to the associative path condition.

1) First assume that f is not unary and g and h are unary. Since $f \not\geq op(s_i)$, for $1 \leq i \leq m$, the variable x must occur as a top-level subterm of s, for otherwise $op(s\sigma) = f$. Also, $x\sigma$ is $h(x_1)$. Now, $s\sigma$ has a normal form $u = h(\cdots h(f\ (u_1, \ldots, u_m)) \cdots)$, where u_i is the normal form of $s_i\sigma$, if $s_i \neq x$, and x_1, otherwise. Let u' denote $f\ (u_1, \ldots, u_m)$. The term $t = h(t_1)$ has a normal form $h(v_1)$, where v_1 is the normal form of $t_1\sigma$. By our assumptions, $u >_{rpo} v_1$ and $op(v_1) \geq h$. If $op(v_1) \neq h$, then $u' >_{rpo} v_1$. If $op(v_1) = h$, then t_1 is either x or $f\ (t_1^1, \ldots, t_l^1)$. In the first case, $u' >_{rpo} v_1$. In the second case, $v_1 = h(\cdots h(f\ (v_1^1, \ldots, v_l^1)) \cdots)$, where v_i^1 is the normal form of $t_i^1\sigma$, if $t_i^1 \neq x$, and x_1, otherwise. Let v_1' denote $f\ (v_1^1, \ldots, v_l^1)$. From $s >_{rpo} t_1$ and $u >_{rpo} v_1$ we can deduce $u' >_{rpo} v_1'$, which also implies $u' >_{rpo} h(v_1)$. In summary, $u >_{rpo} v$.

2) Next suppose that f, g, and h are all non-unary. Because of the associative path condition, g and h must be identical and minimal. Also, $x\sigma$ is $h(x_1, \ldots, x_k)$. As before, s must contain x as a top-level subterm. The normal form u of $s\sigma$ is $h(u_1, \ldots, u_l)$, where $u_1, \ldots, u_l$ are all possible terms

$f(w_1, \ldots, w_m)$ in which w_i is the normal form of $s_i\sigma$, if $s_i \neq x$, and any one of the variables $x_1, \ldots, x_k$, otherwise. Let v_j be the normal form of $t_j\sigma$, for $1 \leq j \leq n$. If $op(v_j) \neq h$, then we define n_j to be 1 and v_1^j to be v_j. Otherwise, v_j can be written as $h(v_1^j, \ldots, v_{n_j}^j)$. The normal form v of $t\sigma$ is $h(v_1^1, \ldots, v_{n_1}^1, \ldots, v_1^n, \ldots, v_{n_n}^n)$. By our assumptions, $u >_{rpo} v_j$ and $op(v_j) \geq h$, for $1 \leq j \leq n$. If $op(v_j) \neq h$, then $u_i >_{rpo} v_j$, for some i, $1 \leq i \leq l$. If $op(v_j) = h$, then we can deduce from $s >_{rpo} t_j$ and $u >_{apo} v_j$ that each term t_l^j is strictly smaller than some term u_i. Consequently, we have $u >_{rpo} v$.

3) If $f > g > h$, then g must be unary. The normal form u of $s\sigma$ is as described in (2) above. Let v and v_1 be the normal forms of t and t_1, respectively. If $op(v_1) \neq h$, then $v = g(v_1)$ and, for some i, $u_i >_{rpo} v_1$. Since $op(u_i) = f$, we also have $u_i >_{rpo} v$ and hence $u >_{rpo} v$. On the other hand, if $v_1 = h(v_1^1, \ldots, v_n^1)$, then $v = h(g(v_1^1), \ldots, g(v_n^1))$. Each term v_j^1 is strictly smaller than some term u_i. Since $op(u_i) = f$, $g(v_j^1)$ is also smaller than u_i. Thus, $u >_{rpo} v$.

4) The only remaining case is that f is unary. Then g and h must be identical. Similar arguments as in (2) above can be applied in this case.
In summary, $s\sigma >_{apo} t\sigma$, which concludes case (c). •

Combining the above two lemmata we obtain

THEOREM 5.3. *If $>$ is a well-founded precedence that satisfies the associative path condition, then the corresponding associative path ordering $>_{apo}$ is a reduction ordering.*

Now, we can apply associative path orderings to AC-termination:

THEOREM 5.4. *Let $>$ be a precedence ordering that satisfies the associative path condition with respect to F_D. Let T be the corresponding A-transform and*

R and R' be rewrite systems, such that R' is contained in $T^{\leftrightarrow}$. If $l > _{apo} r$, for every rule $l \rightarrow r$ in R, then $(R/AC)/(R'/AC)$ terminates.

Proof. Let S be the restriction of the recursive path ordering to terms irreducible in T. By Lemma 5.6, T is canonical modulo $\sim$. Moreover, S/E is terminating and, since the associative path ordering is a reduction ordering, S and $T!$ commute. Now, if $l > _{apo} r$, for every rule $l \rightarrow r$ in R, then R is reducing relative to S and T. Thus, by Theorem 5.2, $R/(\sim \cup T^{\leftrightarrow})$ terminates. Both R' and AC are contained in $\sim \cup T^{\leftrightarrow}$. Therefore $R/(R' \cup AC)$ and, consequently, $(R/AC)/(R'/AC)$ are terminating. •

Theorem 5.3 shows how termination of $(R \cup R')/AC$ may be reduced to proving termination of R/AC and R'/AC separately. The systems R and R' usually contain no varyadic operators. In practice, R' often is a set of "standard" distributivity rules of the form $f(x, g(y)) \rightarrow g(f(x, y))$ or $f(x, h(y, z)) \rightarrow h(f(x, y), f(x, z))$, where operators have fixed arity (f and h are binary and g is unary). In that case, R' is contained in T. Furthermore, we have

Lemma 5.9. *Let $>$ be a well-founded precedence that satisfies the associative path condition and R be a corresponding set of standard distributivity rules. Then R/AC is terminating.*

Proof. For the proof one can use a polynomial interpretation P over the integers greater than 1, where f_P is $\lambda xy.\, 2x+2$, if f is a unary operator in F_D; $\lambda xy.x+y+5$, if f is a minimal non-unary operator in F_D; and $\lambda xy.x \times y$, if f is a non-minimal non-unary operator in F_D (e.g. Peterson and Stickel, 1981). •

Transformation techniques for AC termination were first suggested by Dershowitz, et al. (1983). Associative path orderings were suggested by Plaisted

(1984) and formalized in Bachmair and Plaisted (1985). The difference with our ordering is in the associative path condition. We allow precedence orderings with $f > g > h$ (provided g is unary), but not orderings with $f > h$ and $g > h$, where f and g are incomparable (f , g and h in F_D). Bachmair and Plaisted (1985) rule out the former case, but not the latter. Orderings defined by such a precedence need not be stable, however. Suppose, for example, that f , g and h are in F_D with $f > h$ and $g > h$. Let a and b be constants, with $a > b$. Then $s = f(g(a,x),a) >_{apo} f(f(g(b,x),g(b,x)),a) = t$ (the second term flattens to $f(g(b,x),g(b,x),a)$). If we substitute $h(y,z)$ for x, then the first term reduces to $h(f(g(a,y),a),f(g(a,z),a))$, whereas the second reduces to $h(\cdots f(g(b,y),g(b,z),a)\cdots)$. Thus, $s\sigma >_{apo} t\sigma$ is false in this case. Since stability is not satisfied, rather complicated "lifting" schemes are necessary to compare terms containing variables (see Bachmair and Plaisted, 1986). Our ordering, on the other hand, can be implemented easily and, in addition, is more efficient. The associative path condition is not overly restrictive and, as the examples below indicate, many theories of practical interest allow precedence orderings that satisfy the condition.

The A -transform can also be used in combination with a lexicographic path ordering $>_{lpo}$. In other words, operators not in AC may be given lexicographic status, i.e. some positions in a term may be given more significance than others (see Kamin and Levy, 1980). For example, if $a > b$, then we have $f(a,b) >_{lpo} f(b,a)$, if the operator f has left-to-right status, and $f(b,a) >_{lpo} f(a,b)$, if it has right-to-left status.

5.3. Examples

Most of the rewrite systems below were constructed using the rewrite rule laboratory RRL (Kapur and Sivakumar, 1984). Further examples of canonical systems can be found in Hullot (1980).

Example 5.2. *Boolean algebra.* The following canonical rewrite system R for boolean algebra is due to Hsiang (1985):

$$\begin{array}{rcl} x \oplus false & \rightarrow & x \\ x \wedge false & \rightarrow & false \\ x \wedge true & \rightarrow & x \\ x \wedge x & \rightarrow & x \\ (x \oplus y) \wedge z & \rightarrow & (x \wedge z) \oplus (y \wedge z) \\ x \oplus x & \rightarrow & false \\ x \vee y & \rightarrow & (x \wedge y) \oplus (x \oplus y) \\ x \supset y & \rightarrow & (x \wedge y) \oplus (x \oplus true) \\ x \equiv y & \rightarrow & (x \oplus y) \oplus true \\ \neg x & \rightarrow & x \oplus true \end{array}$$

The operators denote the usual boolean connectives; $\oplus$ denotes exclusive disjunction. The operators $\oplus$ and $\wedge$ are in AC. We first apply a symbolic interpretation $false \rightarrow true$ to R. For termination of the resulting system, we use the associative path ordering corresponding to $F_D = \{\wedge, \oplus\}$ and the precedence ordering shown in the Hasse diagram in Figure 5.2. This precedence satisfies the associative path condition relative to F_D. The fifth rule of R is a distributivity rule and is placed in R'. For all other rules, we have $l >_{apo} r$. Thus, by Theorem 5.4 and Lemma 5.9, R/AC terminates.

Example 5.3. *Abelian group theory.* The axioms for free abelian groups are

$$\begin{array}{rcl} x+0 & = & x \\ x+(-x) & = & 0 \\ x+(y+z) & = & (x+y)+z \\ x+y & = & y+x \end{array}$$

The following system R, where $+$ is in AC, is canonical modulo AC for this

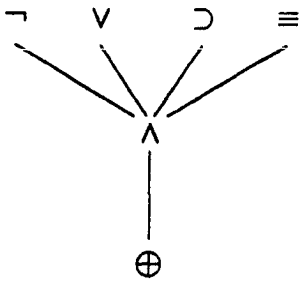


Figure 5.2.

theory:

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

To prove termination of R/AC, we use an associative path ordering corresponding to $F_D = \{+\}$ and a precedence ordering $>$, where $+$ and 0 are smaller than $-$. The precedence $>$ satisfies the associative path condition relative to F_D. Since F_D contains only one operator, the only transformation rules are flattening rules. We have $l >_{apo} r$ for all rules $l \rightarrow r$ in R. Thus, by Theorem 5.4, R/AC terminates.

If R is a rewrite system, then the corresponding *ground system* R_G consists of all rules $l\sigma \rightarrow r\sigma$, where $l \rightarrow r$ is in R and σ is a ground substitution. It can easily be proved that if R contains at least one constant, then it terminates if and only R_G terminates.

Example 5.4. *Rings.* The axioms for rings are the axioms for abelian groups, plus the following two distributivity rules:

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

A canonical system R is

$$\begin{array}{rcl} x+0 & \to & x \\ x+(-x) & \to & 0 \\ -0 & \to & 0 \\ -(-x) & \to & x \\ -(x+y) & \to & (-x)+(-y) \\ x*(y+z) & \to & (x*y)+(x*z) \\ (x+y)*z & \to & (x*z)+(y*z) \\ x*0 & \to & 0 \\ 0*x & \to & 0 \\ x*(-y) & \to & -(x*y) \\ (-x)*y & \to & -(x*y) \end{array}$$

The first five rules form a canonical system for abelian groups. To prove termination of R/AC, an associative path ordering may be used with $F_D = \{+\}$ and a precedence $>$, where $*$ is bigger than $-$, and $-$ is bigger than 0 and $+$.

A ring is associative (commutative) if $*$ is associative (commutative). The following is a canonical system for associative-commutative rings with unit:

$$\begin{array}{rcl} x+0 & \to & x \\ x+(-x) & \to & 0 \\ -0 & \to & 0 \\ -(-x) & \to & x \\ -(x+y) & \to & (-x)+(-y) \\ x*(y+z) & \to & (x*y)+(x*z) \\ x*0 & \to & 0 \\ x*(-y) & \to & -(x*y) \\ x*1 & \to & x \end{array}$$

For termination of R/AC, we use an associative path ordering for $F_D = \{+, *, -\}$ (F_D must contain all AC-operators). We can not use the same precedence as above, since 0 must not be smaller than any operator in F_D. Let us use a precedence $>$, where $*$ is bigger than $-$, and $-$ is bigger than $+$. This precedence satisfies the associative path condition relative to F_D. The fifth, sixth, and

eighth rule are distributivity rules. For all other rules, except the second, we have $l >_{apo} r$. Since $\mathbf{0}$ is minimal, ground instance $l\sigma \to r\sigma$ of the second rule is contained in $>_{apo}$. Thus $>_{apo}$ contains R_G. Consequently, R_G/AC is terminating, which implies that R/AC is also terminating.

Example 5.5. *A-Modules.* Let A be an associative ring with unit. A (left) A-module M over A is an algebraic structure consisting of operations $\oplus : M \times M \to M$ and $\cdot : A \times M \to M$, such that M with $\oplus$ is an abelian group (the identity of the group is denoted by Ω, the inverse to $\oplus$ by I), and the following identities hold:

$$\begin{array}{rcl} \alpha\cdot(\beta\cdot x) & = & (\alpha * \beta)\cdot x \\ 1\cdot x & = & x \\ (\alpha+\beta)\cdot x & = & (\alpha\cdot x) \oplus (\beta\cdot x) \\ \alpha\cdot(x \oplus y) & = & (\alpha\cdot x) \oplus (\alpha\cdot y) \end{array}$$

For the sake of readability we use Greek letters for variables ranging over elements of A, and Roman letters for variables ranging over elements of M. Terms are not typed, though. The following is a canonical system for A-modules, where A is an associative-commutative ring with unit and $+$, $*$, and $\oplus$ are in AC:

$$
\begin{array}{rcl}
\alpha+0 & \rightarrow & \alpha \\
\alpha+(-\alpha) & \rightarrow & 0 \\
-0 & \rightarrow & 0 \\
-(-\alpha) & \rightarrow & \alpha \\
-(\alpha+\beta) & \rightarrow & (-\alpha)+(-\beta) \\
\alpha*(\beta+\gamma) & \rightarrow & (\alpha*\beta)+(\alpha*\gamma) \\
\alpha*0 & \rightarrow & 0 \\
\alpha*(-\beta) & \rightarrow & -(\alpha*\beta) \\
\alpha*1 & \rightarrow & \alpha \\
x \oplus \Omega & \rightarrow & x \\
\alpha\cdot(\beta\cdot x) & \rightarrow & (\alpha*\beta)\cdot x \\
1\cdot x & \rightarrow & x \\
(\alpha+\beta)\cdot x & \rightarrow & (\alpha\cdot x) \oplus (\beta\cdot x) \\
\alpha\cdot(x \oplus y) & \rightarrow & (\alpha\cdot x) \oplus (\alpha\cdot y) \\
(-\alpha\cdot x) \oplus (\alpha\cdot x) & \rightarrow & \Omega \\
(-1\cdot x) \oplus x & \rightarrow & \Omega \\
0\cdot x & \rightarrow & \Omega \\
\alpha\cdot\Omega & \rightarrow & \Omega \\
I(x) & \rightarrow & (-1)\cdot x
\end{array}
$$

The first nine rules form a canonical system for the ring A, and the remaining rules describe the module structure. (A different system for A-modules has been given by Hullot, 1980.) To prove termination of R/AC we use the associative path ordering corresponding to $F_D = \{+, *, -, \oplus\}$ and the precedence ordering $>$ shown in the Hasse diagram in Figure 5.3. The operator $\cdot$ has (right to left) lexicographic status. The fifth, sixth and eighth rules are distributivity rules. All ground instances of other rules are contained in the associative path ordering $>_{apo}$, which implies termination of R/AC.

Example 5.6. *A-Bimodules.* Analogous to left A-modules, one can define right A-modules as algebraic structures consisting of operations $\oplus : M \times M \rightarrow M$ and $\circ : M \times A \rightarrow M$, such that M with $\oplus$ is an abelian group, and the following axioms hold:

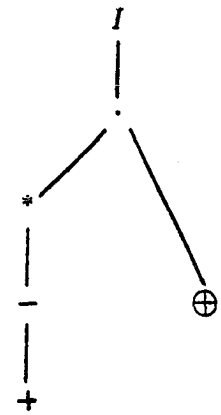


Figure 5.3.

$$\begin{array}{rcl} (x \circ \alpha) \circ \beta & = & x \circ (\alpha^* \beta) \\ x \circ 1 & = & x \\ x \circ (\alpha + \beta) & = & (x \circ \alpha) \oplus (x \circ \beta) \\ (x \oplus y) \circ \alpha & = & (x \circ \alpha) \oplus (y \circ \alpha) \end{array}$$

If M is a left and right A-module, then it is called an A-bimodule if, in addition, the following axiom holds:

$$(\alpha \cdot x) \circ \beta \quad = \quad \alpha \cdot (x \circ \beta)$$

A possible canonical system for A-bimodules over an associative-commutative ring with unit A consists of the canonical system for left A-modules from the preceding examples, plus the following rules:

$$\begin{array}{rcl} \Omega \circ \alpha & \rightarrow & \Omega \\ x \circ 1 & \rightarrow & x \\ x \circ 0 & \rightarrow & \Omega \\ (x \circ \alpha) \circ \beta & \rightarrow & x \circ (\alpha^* \beta) \\ x \circ (\alpha + \beta) & \rightarrow & (x \circ \alpha) \oplus (x \circ \beta) \\ (x \oplus y) \circ \alpha & \rightarrow & (x \circ \alpha) \oplus (y \circ \alpha) \\ x \circ (-\alpha) & \rightarrow & (-1) \cdot (x \circ \alpha) \\ (\alpha \cdot x) \circ \beta & \rightarrow & \alpha \cdot (x \circ \beta) \end{array}$$

where $+$, $*$, and $\oplus$ are in AC. This system is simpler than the one given in

Hullot (1980). For termination of R/AC, we use the set $F_D = \{+, *, -, \oplus\}$ from the preceding example, and extend the precedence ordering $>$ in the Hasse diagram in Figure 5.3 by choosing $\circ$ to be bigger than $\cdot$. The operator $\circ$ has (left to right) lexicographic status.

Example 5.7. *A-Rings.* Let M, with $\cdot$ and $\circ$, be an A-bimodule and, with $\oplus$ and $\otimes$, a ring. M is called an A-ring if the following equations hold:

$$\begin{array}{rcl} (\alpha\cdot x)\otimes y & = & \alpha\cdot(x\otimes y) \\ (x\circ\alpha)\otimes y & = & x\otimes(\alpha\cdot y) \\ x\otimes(y\circ\alpha) & = & (x\otimes y)\circ\alpha \end{array}$$

A canonical system for A-rings over an associative-commutative ring with unit A consists of the canonical system for an A-bimodule above, plus the following rules:

$$\begin{array}{rcl} x\otimes\Omega & \to & \Omega \\ \Omega\otimes x & \to & \Omega \\ x\otimes(y\oplus z) & \to & (x\otimes y)\oplus(x\otimes z) \\ (x\oplus y)\otimes z & \to & (x\otimes z)\oplus(y\otimes z) \\ x\otimes(-\alpha\cdot y) & \to & (-1)\cdot(x\otimes(\alpha\cdot y)) \\ (\alpha\cdot x)\otimes y & \to & \alpha\cdot(x\otimes y) \\ (x\circ\alpha)\otimes y & \to & x\otimes(\alpha\cdot y) \\ x\otimes(y\circ\alpha) & \to & (x\otimes y)\circ\alpha \\ x\otimes(\alpha\cdot(y\circ\beta)) & \to & (x\otimes(\alpha\cdot y))\circ\beta \end{array}$$

where $+$, $*$, and $\oplus$ are in AC. The first five of these additional rules describe the structure of the ring M. The remaining rules describe the A-ring structure. For termination of R/AC, one may use an associative path ordering for $F_D = \{+, *, -, \oplus\}$ and the precedence $>$ from the previous example, extended by defining $\otimes$ to be bigger than $\circ$. The operator $\otimes$ has lexicographic status (left to right).

Example 5.8. *A-Algebras.* Let A be an associative-commutative ring with unit. An A-ring M is called an A-algebra, if

$$\alpha \cdot x \quad = \quad x \circ \alpha$$

for all elements x and α. A canonical system for A-algebras consists of the canonical system for a left A-module from Example 5.8, plus the following rules:

$$\begin{array}{rcl} x \circ \alpha & \rightarrow & \alpha \cdot x \\ x \otimes \Omega & \rightarrow & \Omega \\ \Omega \otimes x & \rightarrow & \Omega \\ x \otimes (y \oplus z) & \rightarrow & (x \otimes y) \oplus (x \otimes z) \\ (x \oplus y) \otimes z & \rightarrow & (x \otimes z) \oplus (y \otimes z) \\ (\alpha \cdot x) \otimes y & \rightarrow & \alpha \cdot (x \otimes y) \\ x \otimes (\alpha \cdot y) & \rightarrow & \alpha \cdot (x \otimes y) \end{array}$$

where $+$, $*$, and $\oplus$ are in AC. For termination of R/AC, the associative path ordering from the preceding example may be used.

In the examples above we have used A-transforms for termination of associative-commutative systems. They may also be used for proving termination of ordinary rewrite systems.

Example 5.9. *Associativity and endomorphism.* Let R be the following rewrite system (Cherifa and Lescanne, 1986):

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

Let T' be the first rule of R and R' be $R - R'$. For termination of R, it suffices to prove termination of R'/T' and T', separately. T' is terminating. Let T be the A-transform corresponding to $F_D = \{f, \cdot\}$ and a precedence ordering $>$, where $\cdot$ is bigger than f. Then we have $l >_{apo} r$, for both rewrite rules $l \rightarrow r$ in R'. Since T' is contained in $T^{\leftrightarrow}$, R'/T' is terminating.

CHAPTER 6

SUMMARY

We have presented various rewrite-based proof methods—standard completion, completion for rewriting modulo a congruence, completion without failure—and have introduced new concepts—proof orderings—for reasoning about them. Our approach differs from previous work in this area in that we do not present specific versions of completion, but formulate completion on a more abstract level, using equational inference rules.

This approach of representing completion as an inference system has several advantages. Various notions such as fairness and correctness (of completion), that otherwise have to be defined with respect to a particular version of completion, can be formalized in more general terms that apply to a wide spectrum of completion procedures. Correspondingly, our correctness results also apply to a large class of completion procedures. This aspect greatly simplifies the task of establishing the correctness of an implementation. Observations pertaining to one method can often be carried over to a related method, e.g. subsumption can be safely used with standard as well as with unfailing completion.

The key concept of our approach are proof orderings. The essential properties of an inference system, in particular, the relationship between the individual inference rules, are closely reflected in the complexity measures used for proof orderings. The complexity measure for standard and unfailing completion indicates, for instance, that simplification rules are only loosely connected with the

other inference rules. Consequently, correctness can be proved without imposing any restrictions on simplification rules. On the other hand, proof orderings also illuminate the specific difficulties with rewriting modulo a congruence and simplification of rules.

Reduction orderings are an important component of proof orderings. They are crucial for the completion process itself. Any completion procedure has to be supplied with a reduction ordering; success or failure often depending on this ordering. Appropriate orderings are often difficult to find. A number of schemes have been devised for constructing reduction orderings for ordinary rewrite systems (see the survey of Dershowitz, 1985). There are only a few such methods for rewriting modulo a congruence. We have discussed one technique, transformation orderings.

Rewrite techniques are not restricted to purely equational theories, but have been successfully integrated in theorem provers for first-order predicate logic, as exemplified by the work of Hsiang (1985). We have undertaken research in adapting proof ordering techniques to such methods.

REFERENCES

[1] Bachmair, L., and Dershowitz, N. (1986). Critical pair criteria for the Knuth-Bendix completion method. *Proc. Symp. on Symbolic and Algebraic Computation,* B. W. Char, ed., Waterloo, Canada, 215-217. (Revised version to appear in *J. Symbolic Computation.*)

[2] Bachmair, L., and Dershowitz, N. (1986). Commutation, transformation, and termination. *Proc. 8th Int. Conf. on Automated Deduction,* J. H. Siekmann, ed., Lect. Notes in Comp. Science 230, Springer-Verlag, Berlin, 5-20.

[3] Bachmair, L., and Dershowitz, N. (1987). Completion for rewriting modulo a congruence. To appear in *Proc. Second Int. Conf. on Rewriting Techniques and Applications,* Springer-Verlag.

[4] Bachmair, L., Dershowitz, N., and Hsiang, J. (1986). Orderings for equational proofs. *Proc. IEEE Symp. Logic in Computer Science,* Cambridge, Massachussetts, 346-357.

[5] Bachmair, L., Dershowitz, N., and Plaisted, D.A. (1987). Completion without failure. To appear in *Proc. Coll. on Resolution of Equations in Algebraic Structures,* Lakeway, Texas.

[6] Bachmair, L., and Plaisted, D.A. (1985). Termination orderings for associative-commutative rewriting systems. *J. of Symbolic Computation* 1, 329-349.

[7] Brown, T. (1975). A structured design-method for specialized proof procedures. Ph.D. thesis, California Institute of Technology, Pasadena, Calif.

[8] Buchberger, B. (1979). A criterion for detecting unnecessary reductions in the construction of Gröbner bases. *Proc. EUROSAM '79,* W. Ng, ed., Lect. Notes in Comp. Science 72, Springer-Verlag, Berlin, 3-21.

[9] Buchberger, B. (1984). A critical-pair/completion algorithm for finitely generated ideals in rings. *Proc. Symp. Rekursive Kombinatorik,* E. Boerger, et al., eds., Lect. Notes in Comp. Science 171, Springer-Verlag, Berlin, 137-161.

[10] Cherifa, A. B., and Lescanne, P. (1986). A method for proving termination of rewriting systems based on elementary computations on polynomials. *Proc. 8th Int. Conf. on Automated Deduction,* J. H. Siekmann, ed., Lect. Notes in Comp. Science 230, Springer-Verlag, Berlin,

[11] Dershowitz, N. (1982). Orderings for term-rewriting systems. *Theoretical Computer Science* 17, 279-301.

[12] Dershowitz, N. (1985). Computing with rewrite systems, *Information and Control* 64, 122-157.

[13] Dershowitz, N. (1985). Termination. *Rewriting Techniques and Applications,* J.-P. Jouannaud, ed., Lect. Notes in Comp. Sci. 202, Springer-Verlag, Berlin, 180-224. (Revised version to appear in *J. of Symbolic Computation.*)

[14] Dershowitz, N., Hsiang, J., Josephson, N.A., and Plaisted, D.A. (1984). Associative-commutative rewriting. *Proc. 8th IJCAI,* Karlsruhe, 940-944.

[15] Dershowitz, N., and Manna, Z. (1979). Proving termination with multiset orderings. *Comm. ACM* 22, 465-476.

[16] Dershowitz, N., Marcus, L., and Tarlecki, A. (1987). Existence, uniqueness, and construction of rewrite systems. To appear in *SIAM J. Computing*.

[17] Fages, F. (1984). Associative-commutative unification. *Proc. 7th Conf. Automated Deduction,* R. E. Shostak, ed., Lect. Notes in Comp. Science 170, Springer-Verlag, Berlin, 194-208.

[18] Hsiang, J. (1985). Refutational theorem proving using term-rewriting systems. *Artificial Intelligence* 25, 255-300.

[19] Hsiang, J., and Rusinowitch, M. (1986). A new method for establishing refutational completeness in theorem proving. *Proc. 8th Int. Conf. on Automated Deduction,* J. H. Siekmann, ed., Lect. Notes in Comp. Science 230, Springer-Verlag, Berlin,

[20] Hsiang, J., and Rusinowitch, M. (1986). On word problems in equational theories. Tech. Rep. 86/29, SUNY at Stony Brook.

[21] Huet, G. (1980). Confluent reductions: abstract properties and applications to term rewriting systems. *J. ACM* 27, 797-821.

[22] Huet, G. (1981). A complete proof of correctness of the Knuth and Bendix completion algorithm. *J. Comp. and System Sciences* 23, 11-21.

[23] Huet, G. and Hullot, J.M. (1982). Proofs by induction in equational theories with constructors. *J. Comp. and System Sciences* 25, 239-266.

[24] Huet, G. and Lankford, D. S. (1978). On the uniform halting problem for term rewriting systems. Rapport Laboria 359, INRIA, Le Chesnay, France.

[25] Hullot, J.M. (1980). A catalogue of canonical term rewriting systems. Tech. Rep. CSL-113, SRI International, Menlo Park, California.

[26] Jouannaud, J.-P. (1983). Confluent and coherent equational term rewriting systems: Application to proofs in abstract data types. *Proc. 8th Coll. on Trees in Algebra and Programming, CAAP '83,* G. Ausiello and M. Protasi, eds., Lect. Notes in Comp. Sci. 59, Springer-Verlag, Berlin, 269-283.

[27] Jouannaud, J.-P., and Kirchner, H. (1986). Completion of a set of rules modulo a set of equations. *SIAM J. Computing* 15, 1155-1194.

[28] Kamin, S., and Levy, J.J. (1980). Two generalizations of the recursive path ordering. Unpublished manuscript, Univ. of Illinois at Urbana-Champaign.

[29] Kapur, D., Musser, D.R., and Narendran, P. (1985). Only prime superpositions need be considered in the Knuth-Bendix procedure. Unpublished manuscript, Computer Science Branch, Corporate Research and Development, General Electric, Schenectady, New York.

[30] Kapur, D., and Sivakumar, G. (1984). Architecture of and experiments with RRL, a rewrite rule laboratory. *Proc. NSF Workshop on the Rewrite Rule Laboratory,* Rensellaerville, New York, 33-56.

[31] Knuth, D., and Bendix, P. (1970). Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra,* J. Leech, ed., Pergamon Press, 263-297.

[32] Küchlin, W. (1985). A confluence criterion based on the generalised Newman lemma. *Proc. Eurocal '85,* B. Caviness, ed., Lect. Notes in Comp. Sci. 204, Springer-Verlag, Berlin, 390-399.

[33] Küchlin, W. (1986). A generalized Knuth-Bendix algorithm. Report 86-01, Dept. of Mathematics, ETH Zürich, Switzerland.

[34] Küchlin, W. (1986). Equational Completion by Proof Simplification. Report 86-02, Dept. of Mathematics, ETH Zürich, Switzerland.

[35] Lankford, D. (1975). Canonical inference. Memo ATP-32, Dept. of Mathematics and Computer Science, University of Texas, Austin, Texas.

[36] Lankford, D.S. (1979). On proving term rewriting systems are Noetherian. Memo MTP-3, Mathematics Department, Louisiana Tech. Univ., Ruston, Louisiana.

[37] Lankford, D., and Ballantyne, A. (1977). Decision procedures for simple equational theories with commutative axioms: Complete sets of commutative reductions. Memo ATP-35, Dept. of Mathematics and Computer Science, University of Texas, Austin, Texas.

[38] Lankford, D., and Ballantyne, A. (1977). Decision procedures for simple equational theories with permutative axioms: Canonical sets of permutative reductions. Memo ATP-37, Dept. of Mathematics and Computer Science, University of Texas, Austin, Texas.

[39] Lankford, D., and Ballantyne, A. (1977). Decision procedures for simple equational theories with associative-commutative axioms: Complete sets of associative-commutative reductions. Memo ATP-39, Dept. of Mathematics and Computer Science, University of Texas, Austin, Texas.

[40] Lankford, D., and Ballantyne, A. (1979). The refutation completeness of blocked permutative narrowing and resolution. *Proc. Fourth Workshop on Automated Deduction,* W. H. Joyner, Jr., ed., Austin, Texas.

[41] Le Chenadec, P. (1986). *Canonical forms in finitely presented algebras.* Lect. Notes in Theoretical Computer Science, Pitman-Wiley.

[42] Lescanne, P. (1982). Some properties of decomposition ordering, a simplification ordering to prove termination of rewriting systems. *R.A.I.R.O. Theoretical Informatics* 16, 331-347.

[43] Manna, Z., and Ness, S. (1970). On the termination of Markov algorithms. *Proc. Hawaii Int. Conf. on System Science,* Honolulu, Hawaii, 789-792.

[44] Metivier, Y. (1983). About the rewriting systems produced by the Knuth-Bendix completion algorithm. *Information Proc. Letters* 16, 31-34.

[45] Musser, D.L. (1980). On proving inductive properties of abstract data types. *Proc. 7th ACM Symp. on Principles of Programming Languages,* Las Vegas, 154-162.

[46] O'Donnell, M.J. (1985). *Equational logic as a programming language.* MIT Press, Cambridge, Massachusetts.

[47] Ohsuga, A., and Sakai, K. (1986). Metis: A term rewriting system generator - An inference engine for equations and inequations. ICOT Research Center, Tokyo, Japan.

[48] Pedersen, J. (1985). Obtaining complete sets of reductions and equations without using special unification algorithms. *Proc. Eurocal '85,* B. Caviness, ed., Lect. Notes in Comp. Sci. 204, Springer-Verlag, Berlin, 422-423.

[49] Peterson, G. (1983). A technique for establishing completeness results in theorem proving with equality. *SIAM J. Computing* 12, 82-100.

[50] Peterson, G., and Stickel, M. (1981). Complete sets of reductions for some equational theories. *J. ACM* 28, 233-264.

[51] Plaisted, D.A. (1978). A recursively defined ordering for proving termination of term rewriting systems. Report R-78-943, Department of Computer

Science, University of Illinois, Urbana, Illinois.

[52] Plaisted, D.A. (1984). Associative path orderings. *Proc. NSF Workshop on the Rewrite Rule Laboratory,* Rensellaerville, New York, 123-126.

[53] Plotkin, G. (1972). Building-in equational theories. *Machine Intelligence* 7, B. Meltzer and D. Michie, eds., American Elsevier, New York, 73-90.

[54] Robinson, G.A., and Wos, L. (1969). Paramodulation and theorem proving in first order theories with equality. *Machine Intelligence* 4, B. Meltzer and D. Michie, eds., American Elsevier, New York, 135-150.

[55] Siekman, J. (1984). Universal unification. *Proc. 7th Conf. Automated Deduction,* R. E. Shostak, ed., Lect. Notes in Comp. Science 170, Springer-Verlag, Berlin, 1-42.

[56] Slagle, J. R. (1974). Automated theorem proving for theories with simplifiers, commutativity, and associativity. *J. ACM* 21, 622-642.

[57] Stickel, M. E. (1981). A complete unification algorithm for associative-commutative functions. *J. ACM* 28, 423-434.

[58] Winkler, F. (1984). The Church-Rosser property in computer algebra and special theorem proving: An investigation of critical-pair/completion algorithms. Dissertation, University Linz, Austria.

[59] Winkler, F. (1985). Reducing the complexity of the Knuth-Bendix completion algorithm: a 'unification' of different approaches. *Proc. Eurocal '85,* Lect. Notes in Comp. Science 204, B. Caviness, ed., Springer-Verlag, Berlin, 378-389.

[60] Winkler, F., and Buchberger, B. (1983). A criterion for eliminating unnecesssary reductions in the Knuth-Bendix algorithm. *Proc. Coll. on*

Algebra, Combinatorics and Logic in Computer Science, Györ, Hungary.

[61] Wos, L.T., Robinson, G.A., Carson, D.F., Shalla, L. (1967). The concept of demodulation in theorem proving. *J. ACM* 14, 698-709.

VITA

Leo Bachmair was born in Ried im Innkreis, Austria, in 1959. He obtained a diploma in mathematics from Johannes Kepler Universität Linz, Austria, in 1982 and a M.S. in computer science from the University of Illinois at Urbana-Champaign in 1985. He is currently an assistant professor in the Department of Computer Science at the State University of New York at Stony Brook.