MAX-PLANCK-INSTITUT FÜR INFORMATIK

Ordered Chaining for Total Orderings

Leo Bachmair
Harald Ganzinger

MPI–I–93–250 December 1993

INFORMATIK

Im Stadtwald
D 66123 Saarbrücken
Germany

Addresses

Leo Bachmair, Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY 11794, U.S.A, leo@sbcs.sunysb.edu
Harald Ganzinger, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, hg@mpi-sb.mpg.de

Acknowledgements

The research described in this paper was supported in part by the German Science Foundation (Deutsche Forschungsgemeinschaft) under grant Ga 261/4-1, by the German Ministry for Research and Technology (Bundesministerium für Forschung und Technologie) under grant ITS 9102/ITS 9103 and by the ESPRIT Basic Research Working Group 6028 (Construction of Computational Logics). The first author was also supported by the Alexander von Humboldt Foundation.

Abstract

We design new inference systems for total orderings by applying rewrite techniques to chaining calculi. Equality relations may either be specified axiomatically or built into the deductive calculus via paramodulation or superposition. We demonstrate that our inference systems are compatible with a concept of (global) redundancy for clauses and inferences that covers such widely used simplification techniques as tautology deletion, subsumption, and demodulation. A key to the practicality of chaining techniques is the extent to which so-called variable chainings can be restricted. Syntactic ordering restrictions on terms and the rewrite techniques which account for their completeness considerably restrict variable chaining. We show that variable elimination is an admissible simplification techniques within our redundancy framework, and that consequently for dense total orderings without endpoints no variable chaining is needed at all.

Keywords

Automated Theorem Proving, First-Order Logic, Total Orderings, Chaining, Variable Elimination, Term Rewriting

Contents

1 Introduction

The axioms of the theories of partial and total orderings are extremely prolific in the context of resolution-based theorem proving. Many theorem provers build in transitivity by so-called chaining rules, which allow one to derive $(C \vee D \vee u < v)\sigma$ from premises $C \vee u < s$ and $D \vee t < v$ after unifying s and t by σ. Even though chaining is a considerable improvement over naive resolution, in this general form it still generates a huge search space. First, chaining inferences are always possible if s or t is a variable. For instance, the totality axiom $x < y \vee y \leq x$ can always be applied in four different ways, leading to the derivation of many equivalent variants of each clause. The situation is even worse for total orderings with additional structure, and in particular dense orderings with no endpoints. The clauses which express density and the absence of endpoints can also be chained with any other clause. For example, the clause $x < rx$ expressing the non-existence of a right endpoint, generates infinitely many inequalities $x < r^n x$ just by chaining with itself. Secondly, the same clause $C \vee u < s$ can be used as the first or as the second premise of a chaining inference, so that one needs to chain with both terms, u and s. Third, and related to the first two points, chaining is designed to generate the full (possibly infinite) transitive closure of a given set of inequalities, even though for any particular refutation a finite subset will be sufficient. This lack of goal-orientedness forms another obstacle to any efficient proof search. These problems apply to transitive relations in general; solutions have been proposed for certain theories.

Bledsoe and Hines (1980) have investigated dense total orderings without endpoints and have developed techniques for eliminating certain occurrences of variables from formulas. In their inference system, no chaining through variables is performed and no explicit inferences with transitivity, density, totality and the "no endpoints" axioms are computed. Completeness results for particular such systems of restricted chaining are proved by Bledsoe, Kunen and Shostak (1985) and Hines (1992). Theorem provers developed from these theoretical investigations have performed successfully in proving theorems such as the continuity of the sum of two continuous functions or the intermediate value theorem; see Bledsoe and Hines (1980), Hines (1988) or Hines (1990).

Ordered paramodulation (Hsiang and Rusinowitch 1991) and superposition (Bachmair and Ganzinger 1990) are chaining-based inference systems for equality relations (the congruence properties of which also require chaining through subterms). Paramodulation into or below variables was first shown to be unnecessary by Brand (1975). Various syntactic restrictions avoid that equalities are always applied in both directions. In many cases even an infinite transitive closure of a given set of equalities can be represented by a finite convergent rewrite system. Only that system, and not the complete transitive closure of the given set of equalities (or clauses), is computed by the ordered variants of paramodulation and superposition.

Our aim is to combine the two approaches, keeping their best features, but avoiding their drawbacks. For that purpose we adapt the term rewriting techniques for arbitrary transitive relations described in Bachmair and Ganzinger (1993) to total orderings.

1.1 Results

We present refutationally complete inference systems for total orderings, in which chaining is restricted by syntactic ordering constraints in much the same way as we know it from superposition calculi. That is, chaining inferences as above are performed only if the term $s\sigma$ is maximal in both (instances by σ of the) parent clauses. These ordering constraints immediately rule out many forms of chaining through variables. For instance, a term $x\sigma$ cannot be maximal if the variable x is shielded in a clause, that is, occurs as an argument of a function symbol. We also go beyond simple chaining in that we eliminate all occurrences of the maximal term in one single inference that combines several chaining steps. The effect is similar to hyper-resolution, in that the results of intermediate chaining steps need not be explicitly generated. Explicit inferences with transitivity and totality are shown to be redundant. This inference system applies to arbitrary total orderings and avoids most, though not all, variable chainings. The ordering constraints, like in the equational case, derive from a presentation of the full transitive closure of the given binary relation by an appropriate rewrite closure.

We prove refutational completeness of our inference systems in the presence of a general notion of redundancy for clauses and inferences by which most of the commonly applied simplification and elimination techniques (e.g., tautology elimination or subsumption) can be justified. For total orderings in particular we show that variable elimination, as proposed by Bledsoe and Hines (1980), is a simplification rule in our sense: a clause becomes redundant once variable elimination has been applied to it. In other words, variable elimination can be made mandatory. Ordering constraints for inferences and mandatory elimination of unshielded variables together achieve that chaining through variables can be completely avoided for dense total orderings with no endpoints. Variable elimination is also excluded in the inference systems proposed by Bledsoe, Kunen and Shostak (1985) and Hines (1992), though these calculi, unlike ours, are not compatible with tautology deletion. Our proofs are comparatively simple and in particular profit from our ability of treating simplification techniques such as variable elimination as a separate issue.

The first of the two calculi which we present in this paper lacks an efficient treatment of equality. Like in previous approaches (Bledsoe and Hines 1980, Bledsoe, Kunen and Shostak 1985, Hines 1992) an equality $s \approx t$ is represented by two inequalities $s \leq t$ and $t \leq s$. The disadvantage of this approach is that the implicitly specified equality relation requires additional congruence axioms for each function symbol, while such powerful simplification mechanisms as demodulation cannot be used. Therefore we describe another system, in which we combine chaining with superposition (Bachmair and Ganzinger 1990). Equality is thus built into the inference rules; explicit inferences with congruence axioms or functional reflexive axioms and superposition into or below variables are not needed. The extended inference system also considerably improves earlier chaining systems with paramodulation (Slagle 1972), and for dense orderings without endpoints superposition from variables is not needed either. In addition, when superposing into inequalities only the maximal term of the inequality needs to be replaced. As before, the completeness result allows to discard redundant clauses and inferences and admits simplifications such as demodulation or condensement.

We have implemented most the techniques investigated here as an extension of the Saturate system (Nivela and Nieuwenhuis 1993), and have obtained promising experimental results.

2 Preliminaries

2.1 Orderings

A (strict) *partial ordering* is a transitive and irreflexive binary relation; a *quasi-ordering* a reflexive and transitive binary relation. The reflexive closure of a strict ordering is a quasi-ordering. On the other hand, if $\leq$ is a quasi-ordering, then its *strict part* $<$, defined by: $x < y$ if and only if $x \leq y$ but not $y \leq x$, is a strict ordering. An ordering $<$ is said to be *total* if $x < y$ or $y < x$, whenever x and y are distinct. The ordering is *dense* if for all x and y with $x < y$, there exists an element z, such that $x < z$ and $z < y$. An ordering $<$ has no *left* (resp. *right*) *endpoint* if for every x there exists a y such that $y < x$ (resp. $x < y$). By an ordering *without endpoints* we mean one that has neither a left nor a right endpoint. For instance, the usual less-than relation on the natural numbers is a total ordering with a left, but no right, endpoint. The less-than relation on the real numbers is a dense total ordering without endpoints.

2.2 Predicate logic

We consider first-order predicate logic with equality; more specifically, first-order languages with function symbols, variables, and (interpreted) predicate symbols $<$, $\leq$, and $\approx$.[1] A *term* is an expression $f(t_1, \ldots, t_n)$ or x, where f is a function symbol of arity n, x is a variable, and $t_1, \ldots, t_n$ are terms. An *atomic formula* (or simply *atom*) is an expression $s \approx t$, called an *equality*, or an expression $s < t$ or $s \leq t$, called a strict or non-strict *inequality*, respectively, where s and t are terms. A *literal* is an expression A (a *positive* literal) or $\neg A$ (a *negative* literal), where A is an atomic formula. We also write $s \not\approx t$ instead of $\neg(s \approx t)$ and use a similar notation for inequalities. When we speak of the *priority* of a predicate symbol, we take $<$ to have the highest and $\approx$ the lowest priority. The symbol $\lhd$ is used to denote (strict or non-strict) inequalities; the symbol $\unlhd$ to denote equalities and inequalities. A *clause* is a finite multiset of literals. We write a clause by listing its literals, $\neg A_1, \ldots, \neg A_m, B_1, \ldots, B_n$; or as a disjunction $\neg A_1 \vee \cdots \vee \neg A_m \vee B_1 \cdots \vee B_n$; or as a sequent $A_1, \ldots, A_m \rightarrow B_1, \ldots, B_n$. An expression is said to be *ground* if it contains no variables.

By a (*Herbrand*) *interpretation* we mean a set I of ground atomic formulas. We say that an atom A is *true* (and $\neg A$, *false*) *in* I if $A \in I$; and that A is *false* (and $\neg A$, *true*) in I if $A \notin I$. A ground clause is *true* in an interpretation I if at least one of its literals is true in I; and is *false* otherwise. In general, a clause is said to be true in I if all its ground instances are true. The *empty clause* is false in every interpretation. We say that I is a *model* of a set of clauses N (or that N is *satisfied* by I) if all elements of N are true in I. Occasionally, a model of N will also be called an N-*interpretation*. A set N is *satisfiable* if it has a model, and *unsatisfiable* otherwise. For instance, any set containing the empty clause is unsatisfiable.

We intend $\approx$ to be interpreted as an equality relation[2] and $<$ as a strict ordering. Also, $x \leq y$ is meant to be interpreted as $x < y \vee x \approx y$. (From a logical point of view, the symbol $\leq$ is therefore superfluous. But for theorem proving purposes shorter formulas are generally preferable, and it is better to avoid replacing a non-strict inequality by a disjunction.) The most problematic properties of these relations—for an automated

[1] Uninterpreted predicate symbols can be encoded in a many-sorted language with function symbols and equality, by representing a formula $P(t_1, \ldots, t_n)$ as an equality $f_P(t_1, \ldots, t_n) \approx true_P$.

[2] An equality relation is a congruence on ground terms.

theorem prover—are the transitivity properties. By a *transitivity interpretation* we mean a model of the set TR of all *transitivity axioms*

$$x \unlhd_1 y, y \unlhd_2 z \quad \rightarrow \quad x \unlhd z$$

where $\unlhd$ is the highest-priority symbol of $\unlhd_1$ and $\unlhd_2$. For example, $x \not< y \vee y \not\approx z \vee x < z$ is a transitivity axiom. The set PO consisting of the axioms in TR and the axioms

$$\begin{array}{rcl} x < y & \rightarrow & x \leq y \\ & \rightarrow & x \leq x \\ x < x & \rightarrow & \end{array}$$

encodes that $<$ is a strict ordering and $\leq$ a quasi-ordering (with $<$ contained in its strict part). Interpretations satisfying PO are called partial orderings for short. If in addition the totality axiom

$$x < y \vee y \leq x$$

is satisfied, the interpretation is called a total ordering. In that case, $<$ coincides with the strict part of $\leq$.

2.3 Chaining

Many theorem provers build in transitivity by so-called *chaining rules*,

$$\frac{C \ , u < s \quad D \ , t < v}{C\sigma \ , D\sigma \ , u\sigma < v\sigma}$$

where σ is a most general unifier of s and t; see Slagle (1972). For instance, with chaining we may deduce $s < t$ from $s < u$ and $u < t$. Specific variants of this inference rule can be found in calculi for dense total orderings (Bledsoe and Hines 1980, Bledsoe, Kunen and Shostak 1985) and in paramodulation calculi for equality (Robinson and Wos 1969). In its full generality, the chaining rule is not practical, as the search space spanned by the inferences may be huge. *Variable chainings*, that is, chainings with a premise $C \vee u < x$ or $D \vee y < v$, are particularly prolific. Fortunately, they can be completely excluded in the case of dense total orderings without endpoints (Bledsoe, Kunen and Shostak 1985) and considerably restrained in the case of paramodulation (Brand 1975 was the first to prove that paramodulation into a variable is unnecessary).

Chaining essentially generates the transitive closure of a given binary relation. For example, from $a < b$ and $b < c$ and $c < d$ we may deduce $a < c$ and $b < d$ and $a < d$. In other words, whenever there is a (finite) chain of equalities and/or inequalities

$$x_1 \unlhd_1 x_2 \unlhd_2 \cdots \unlhd_{n-1} x_n$$

the equality or inequality $x_1 \unlhd x_n$ can be deduced, where $\unlhd$ is the highest-priority predicate among all $\unlhd_i$. The basic idea of the term rewriting approach is to consider not arbitrary chains, but only those in which the intermediate terms $x_2, \ldots, x_{n-1}$ are in a certain sense simpler than the endpoints x_1 and x_n. Moreover, the equality or inequality $x_1 \unlhd x_n$ is not necessarily deduced, but instead the corresponding chain may be implicitly represented by a rewrite system. We employ these ideas for the design of refutationally complete chaining systems in which various constraints are imposed on inference rules. Two concepts are of importance in this context: rewrite systems and simplification orderings.

2.4 Simplification orderings

An ordering $\succ$ is *well-founded* if there is no infinite sequence $t_1 \succ t_2 \succ \cdots$. An ordering on ground terms is called a *reduction ordering* if it is well-founded and satisfies the following property: whenever $s \succ t$ then $u[s] \succ u[t]$, for all terms u, s, and t. A reduction ordering is called a *simplification ordering* if it also satisfies the *subterm property*: $A[s] \succ s$, for all expressions A and proper subexpressions s of A. A reduction ordering is *complete* if it is total on ground terms. (Complete reduction orderings are simplification orderings.)

Later on, we shall extend the term ordering $\succ$ to an ordering on literals. Also, any ordering $\succ$ on a set S can be extended to an ordering on finite multisets over S (which for simplicity we also denote by $\succ$) as follows: $M \succ N$ if (i) $M \neq N$ and (ii) whenever $N(x) > M(x)$ then $M(y) > N(y)$, for some y such that $y \succ x$. An ordering on literals can thus be extended to an ordering on clauses. If $\succ$ is a total (resp. well-founded) ordering, so is its multiset extension.

Finally, we extend an ordering $\succ$ to non-ground expressions by defining: $E \succ E'$ if and only if $E\sigma \succ E'\sigma$, for all ground instances $E\sigma$ and $E'\sigma$. Thus, we have $E \not\succeq E'$ if $E'\sigma \succ E\sigma$, for some ground instances $E\sigma$ and $E'\sigma$. We say that a literal L is *maximal* in a clause C if $L' \succ L$, for no literal L' in C; and that L is *strictly maximal* in C if $L' \succeq L$, for no L' in C.[3]

For a survey on (reduction) orderings see Dershowitz (1987).

2.5 Rewrite systems

Let I be a set of ground atomic formulas and $\succ$ be a simplification ordering. We use the equalities and inequalities in I as *rewrite rules*. More precisely, we write $u \lhd_I v$ if $u \lhd v$ is an inequality in I and write $u \approx_I v$ if $u = w[s]$ and $v = w[t]$, for some term w and equality $s \approx t$ or $t \approx s$ in I. Furthermore, we write $u \unlhd_I^L v$, if $u \succ v$, and $u \unlhd_I^R v$, if $v \succ u$. The subscripts are dropped if I is clear from the context.

By a *proof* (in I) we mean a finite sequence $u_0 \unlhd_1 u_1 \unlhd_2 u_2 \ldots u_{n-1} \unlhd_n u_n$ (in I), where $n \geq 0$. More specifically, we speak of a proof of $u_0 \leq u_n$. We speak of a proof of $u_0 < u_n$ if in addition at least one of the symbols $\unlhd_i$ is $<$ (and hence $n \geq 1$); and of a proof of $u_0 \approx u_n$ if *all* symbols $\unlhd_i$ are $\approx$. (If $n = 0$, we have a proof of $u_0 \approx u_0$.) By the *transitive closure* of I we mean the set of all atoms $u \unlhd v$ provable in I. A *rewrite proof* is a sequence

$$u_0 < u_1$$

or

$$u_0 \unlhd_1^L \cdots \unlhd_{m-1}^L u_m \unlhd_m^R \cdots \unlhd_n^R u_n.$$

By the *rewrite closure* of I we mean the set of all atoms $u \unlhd v$ that are provable by a rewrite proof in I.

In proving the completeness of chaining systems we construct Herbrand models for certain clause sets, describing interpretations by rewrite closures. A key question is under which circumstances the rewrite closure of a set I is a transitivity interpretation; a question, it turns out, that is related to commutation properties of the rewrite relations $\unlhd_I$.

[3]Depending on the ordering on ground terms, this extension to non-ground expressions may or may not be decidable. In the latter case one will have to employ a safe and decidable approximation.

If a sequence $u_0 \trianglelefteq_1 \cdots \trianglelefteq_n u_n$ is not a rewrite proof then either (i) there is a subsequence $u_{i-1} \trianglelefteq_i^R u_i \trianglelefteq_{i+1}^L u_{i+1}$, or (ii) u_{i-1} and u_i are identical or incomparable (i.e., $u_{i-1} \not\succ u_i$ and $u_i \not\succ u_{i-1}$), for some i. A subsequence $u \trianglelefteq_1^R t \trianglelefteq_2^L v$, called a *peak*, is said to *commute* if $u \trianglelefteq v$ is provable by a rewrite proof, where $\trianglelefteq$ is the highest-priority symbol of $\trianglelefteq_1$ and $\trianglelefteq_2$. Commutation allows us to deal with case (i). If $\succ$ is a complete simplification ordering, then case (ii) applies only if $u_{i-1} = u_i$.

For example, if I contains the atoms $a < b$ and $b < c$ and $c < d$, and if $a \succ b \succ c \succ d$, then $a <^L b$ and $b <^L c$ and $c <^L d$. The relation $<^L$ evidently commutes with the (empty) relation $<^R$. If we choose the ordering differently, $b \succ c \succ d \succ a$, there is a peak, $a <^R b <^L c$, that does not commute. There are rewrite proofs of $a < b$ and $b < c$, but not of $a < c$.

The following lemma relates commutation and transitivity.

Lemma 1 *Let $\succ$ be a complete simplification ordering and I be a Herbrand interpretation that contains no strict inequality $t < t$. The rewrite closure of I is a transitivity interpretation if and only if all peaks in I commute.*

Proof. Suppose that all peaks in I commute. It is sufficient to show that whenever $u \trianglelefteq v$ is provable in I, then there also exists a rewrite proof. If a sequence $u = u_0 \trianglelefteq_1 \cdots \trianglelefteq_k u_k = v$ contains peak, then by commutation the peak can be replaced by a rewrite proof. Moreover, any proof step $u_i \approx u_{i+1}$ or $u_i \leq u_{i+1}$, where $u_i = u_{i+1}$, can be eliminated from a proof. (Strict inequalities $t < t$ cannot always be eliminated, which is the reason why we have to exclude them.) The multiset of terms $\{u'_0, \ldots, u'_n\}$ used in the resulting new proof $u = u'_0 \trianglelefteq_1 \cdots \trianglelefteq_n u'_n = v$ is smaller than the multiset $\{u_0, \ldots, u_k\}$ with respect to the (multiset extension of the) simplification ordering $\succ$. Since the ordering $\succ$ is well-founded, we may conclude that every sequence of rewrites can be transformed to a rewrite proof.

For the converse observe that by transitivity of the rewrite closure of I any peak must have a rewrite proof. □

The lemma provides the starting point for our investigation of chaining techniques for total orderings, as peaks can be made to commute by applying suitable chaining inferences. (In the above example the peak $a <^R b <^L c$ commutes once the inequality $a < c$ has been deduced from $a < b$ and $b < c$ by chaining.) Similar, so-called "critical pair lemmas" form the basis of all completion procedures; Levy and Agustí (1993) appear to have been the first to apply these techniques to non-symmetric rewrite relations. We go beyond usual completion procedures in that we consider general clauses, and thus have to deal with negative literals and disjunctions of literals. Particular emphasis will be given to the question of (the necessity of) variable chainings. For a discussion of the general aspects of these questions we refer to our work on rewrite techniques for transitive relations (Bachmair and Ganzinger 1993). In this paper we look at total orderings, primarily dense total orderings without endpoints.

3 Maximal chaining

3.1 Inference rules

The predicate $\leq$ for non-strict inequality can be defined in terms of strict inequality and equality. It is also possible, though, to express equality in terms of inequality, representing $u \approx v$ by (the conjunction of) two inequalities $u \leq v$ and $v \leq u$. This, indeed, is the framework in which previous chaining inference systems have been formalized, the advantage being that no specific inference mechanisms for equality are needed. The necessary properties of the implicit equality relation are specified by the set $EE_{\mathcal{F}}$ of clauses

$$\bigvee_{1\le i\le n} x_i < y_i \vee \bigvee_{1\le i\le n} y_i < x_i \vee f(x_1,\ldots,x_n) \le f(y_1,\ldots,y_n)$$

where f ranges over all function symbols in $\mathcal{F}$; see Bledsoe, Kunen and Shostak (1985). The necessity of including these axioms is a disadvantage. Later on, we will build equality into the inference mechanism via paramodulation, thereby obviating any explicit equality axioms. (Another advantage of equational inference systems is that further simplification techniques such as demodulation become available, which are indispensable in actual implementations.)

In this section all atomic formulas are assumed to be (strict or non-strict) inequalities. Totality can be expressed as a clause $x < y \vee y \le x$ or $x \le y \vee y \le x$. By TA we denote the set of these two clauses. They can be used to transform negative inequalities into positive ones: replace $x \not< y$ by $y \le x$ and $x \not\le y$ by $y < x$. Thus, we only need to consider disjunctions of positive inequalities.

We build the irreflexivity and transitivity properties directly into the inference mechanism.

Irreflexivity Resolution:

$$\frac{C\ ,\ s < t}{C\sigma}$$

where σ is the most general unifier of s and t and $s\sigma < t\sigma$ is maximal in $C\sigma$.

Maximal Chaining:

$$\frac{C\ ,\ u_1 \lhd_1 s_1, \ldots, u_m \lhd_m s_m \quad D\ ,\ t_1 \lhd'_1 v_1, \ldots, t_n \lhd'_n v_n}{C\sigma\ ,\ D\sigma\ ,\ \bigvee_{1\le i\le m, 1\le j\le n} u_i\sigma \lhd_{i,j} v_j\sigma}$$

where (i) σ is the most general unifier of $s_1, \ldots, s_m, t_1, \ldots, t_n$ and $\lhd_{i,j}$ is the highest-priority symbol of $\lhd_i$ and $\lhd'_j$, (ii) $u_i\sigma \not\succeq s_1\sigma$, for all $1 \le i \le m$, and $u\sigma \not\succeq s_1\sigma$, for all terms u in C, and (iii) $v_i\sigma \not\succeq t_1\sigma$, for all $1 \le i \le n$, and $v\sigma \not\succ t_1\sigma$, for all terms v in D, and $t_1\sigma$ occurs in $D\sigma$ only in inequalities $v \lhd t_1\sigma$.

For example, from $x < f(a) \vee b < f(x)$ and $f(y) < c \vee f(a) < d$ we may deduce $a < c \vee a < d \vee b < c \vee b < d$ (assuming $f(a)$ is the maximal term in the clause).

The conclusion of a maximal chaining inference can also be obtained by a sequence of $m \times n$ ordinary chaining inferences, interspersed with factoring inferences; but an important difference is that with maximal chaining the intermediate clauses need not be deduced. In this regard, the effect of maximal chaining is similar to that of hyper-resolution. Chaining inferences are restricted to the maximal terms in a clause; maximal chaining is designed to eliminate these terms.

The inference system $MC^{\succ}$ consisting of irreflexivity resolution and maximal chaining represents our basic inference mechanism for total orderings. (We also assume that the premises of an inference have no variables in common; if necessary, the variables in one premise are renamed.) Below we shall outline further improvements to the calculus for richer structures, such as dense total orderings without endpoints. These improvements ultimately derive from the concept of redundancy (of clauses and inferences) discussed in the next section.

3.2 Redundancy

Simplification and deletion techniques, such as subsumption, tautology deletion, condensement, demodulation, contextual rewriting, etc., have proved to be indispensable for actual implementations of theorem provers. We have developed a framework, based on the notion of redundancy, for formalizing and reasoning about such techniques (Bachmair and Ganzinger 1994). The main ideas are sketched below.

Let $\succ$ be a well-founded ordering on clauses and N and S be sets of clauses. We say that a ground instance $C\sigma$ of some clause C (which need not be an element of N) is *S-redundant* with respect to N if there exist ground instances $C_1\sigma_1, \ldots, C_k\sigma_k$ of N such that $C\sigma$ is true in every S-model of $C_1\sigma_1, \ldots, C_k\sigma_k$ and $C\sigma \succ C_j\sigma_j$, for all j with $1 \leq j \leq k$. The clause C is S-redundant if all its ground instances are.

Tautologies are S-redundant in this sense, for any set S, and most cases of proper subsumption are also covered by this notion of redundancy.[4] According to the definition, the axioms in S are all S-redundant. We are interested in PO-redundancy. (Since these axioms are built into the chaining mechanism, we expect them to be redundant.)

If an inference system is in a certain sense compatible with this notion of redundancy, then redundant formulas can be ignored by inferences. A ground inference (i.e., an inference in which the premises and the conclusion are ground clauses) is called *simplifying* (with respect to $\succ$) if its conclusion is smaller than its maximal premise. A non-ground inference is simplifying if all its ground instances are. We next define a suitable well-founded ordering on ground (instances of) clauses, with respect to which irreflexivity resolution and maximal chaining are simplifying. It is not difficult to find such an ordering; the reason we present one that is relatively complicated is that we apply the same ordering to variable elimination and superposition inferences.[5]

The clause ordering is based on an ordering on literals. Let L be a literal $u \trianglelefteq v$ or $\neg(u \trianglelefteq v)$ in a ground instance $C\sigma$ of a clause C. We have to distinguish between positive and negative literals; let s_L be 1 if L is a negative literal, and 0 otherwise. We shall compare literals primarily by looking at their terms; let max_L be the maximal term, and min_L the minimal term of u and v. The argument position of the maximal term is also of importance; let p_L be 1 if L is an (strict or non-strict) inequality with $max_L = u$, and 0 otherwise. We need to distinguish between the different predicates; let r_L be 2 if L is a strict inequality, 1 if L is a non-strict inequality, and 0 otherwise. Finally, certain characteristics of the clause C are crucial for dealing with variable elimination. Let n_C be the number of different variables occurring as an argument of an equality or inequality in C. For example, if C is the clause $x < f(y) \vee z \leq f(z)$, then $n_C = 2$, as both x and z (but not y) qualify. We define $n_L = -1$, if C is the totality axiom; $n_L = 0$, if $\trianglelefteq$ is $\approx$; and $n_L = n_C$, otherwise.[6]

Now, with each literal L we associate the tuple $(max_L, s_L, p_L, n_L, r_L, min_L)$, and compare such tuples using the ordering $\succ$ in the first and last component, and the usual ordering on natural numbers in the other components. The corresponding ordering on literals is also denoted by the symbol $\succ$. Its multiset extension is an ordering on ground (instances of) clauses.

For example, if $s \succ t$, we associate with an occurrence of a literal $s < t$ in a ground

[4] The definition of redundancy does not, for instance, cover cases where two clauses subsume each other.

[5] This ordering is the key to our refutational completeness proofs.

[6] We shall see that the number n_C decreases under variable elimination. The totality axiom is needed to justify variable elimination; its special role is reflected in the ordering.

instance of C the tuple $(s, 0, 1, n_C, 2, t)$; with $t \not< s$ the tuple $(s, 1, 0, n_C, 2, t)$; and with $s \leq s$ the tuple $(s, 0, 1, n_C, 1, s)$.

It can easily be checked that irreflexivity resolution or maximal chaining inferences are simplifying with respect to this clause ordering. (In the case of irreflexivity resolution, one literal is removed and the number n_C can only decrease. In the case of maximal chaining, the maximal term occurs as the first argument of an inequality in the maximal premise, whereas in the conclusion it may occur only as the second argument of an inequality, if at all.)

A ground inference with conclusion B and maximal premise C is called *S-redundant* with respect to N if either some premise is S-redundant, or else there exist ground instances $C_1, \ldots, C_k$ of N such that B is true in every S-model of $C_1, \ldots, C_k$ and $C \succ^c C_j$, for all j with $1 \leq j \leq k$. A non-ground inference is called S-redundant if all its ground instances are S-redundant.

We say that a set of clauses N is *saturated up to S-redundancy* if all inferences from N are S-redundant. Since the inferences in $MC^{\succ}$ are simplifying, they can be rendered redundant by adding the conclusion to the given set of clauses. Thus, computing the closure of a clause set under these inferences yields a saturated set.

3.3 Refutational completeness

We claim that if a set N is saturated up to PO-redundancy under $MC^{\succ}$, then it either contains the empty clause or has a total ordering as model. Our approach for proving this claim is straightforward. We define a (transfinite) sequence of "partial" interpretations, one for each ground instance of a clause in N, beginning with the "empty" interpretation and considering clauses in increasing order according to the given well-founded ordering $\succ$. Whenever a clause C is false in the current partial interpretation I_C, we extend I_C by adding an inequality of C that extends the ordering described by I_C in a least restrictive way. The clause C is satisfied in the extended interpretation.

Let us first formally define which inequality is chosen from C to extend an interpretation I. Let C be a ground clause with maximal term s and I be an interpretation. By $P_I(C)$ we denote some inequality $s \lhd t$ or $t \lhd s$ in C, such that (i) $P_I(C)$ is an inequality $s \lhd t$, if C contains any inequality $s \lhd' t'$ at all; (ii) if $P_I(C)$ is an inequality $s \lhd t$, there is no inequality $s \lhd' t'$ in C, such that either $t < t'$ is true in I or else $t \leq t'$ is true in I and $\lhd'$ has lower priority than $\lhd$; (iii) if $P_I(C)$ is an inequality $t \lhd s$, there is no inequality $t' \lhd' s$ in C, such that either $t' < t$ is true in I or else $t' \leq t$ is true in I and $\lhd'$ has lower priority than $\lhd$.

For example, suppose $a \succ b \succ c$ and let C be the clause $a < b \vee a < c$. If $b < c$ is true in I, then $P_I(C)$ is $a < c$. Extending I by $a < b$ yields a *non-minimal* extension of I to a model of C, as $a < c$ follows from $a < b$ and $b < c$ by transitivity.

Now, for every ground instance C of a clause in N, let R_C be the set $\bigcup_{C \succ D} E_D$ and I_C the rewrite closure of R_C. If C is false in I_C and does not contain $s < s$, where s is the maximal term in C, then $E_C = \{P_{I_C}(C)\}$. In all other cases, $E_C = \emptyset$. We say that C *produces* $P_{I_C}(C)$, and call C *productive*, if $E_C = \{P_{I_C}(C)\}$. By I^C we denote the rewrite closure of $R_C \cup E_C$. Finally, by R we denote the set $\bigcup_C E_C$ and by I the rewrite closure of R.

Evidently, a productive clause C is false in I_C, but true in I^C. The key properties of the interpretations I_C are summarized in the following lemma.

Lemma 2 *Let N be a set of clauses that is saturated up to PO-redundancy but does not contain the empty clause. Let I be the interpretation constructed from all ground instances of $N \cup TA$. Then for every ground instance C of a clause in $N \cup TA$ we have:*

(1) if C is productive it is non-redundant;

(2) I^C satisfies PO; and

(3) C is true in I^C.

Proof. We use induction on $\succ$. Let C be a ground instance of a clause in N, such that the assertion holds for all smaller ground instances of N.

(1) The induction hypothesis may be used to show that I_C is a partial ordering and all ground instances of N and the totality axioms that are smaller than C are true in I_C. Therefore, if C were PO-redundant it would have to be true in I_C. But as a productive clause C is false in I_C.

(2) To prove that I^C is a transitivity interpretation, it suffices, by Lemma 1, to show that all peaks in I^C commute. By the induction hypothesis, peaks in I_C commute. Suppose there is a peak $u_1 \lhd_1' s \lhd_1 v_1$ in I^C that is not also a peak in I_C. Then $s \lhd_1 v_1$ must be produced by C, so that C can be written as $C' \vee s \lhd_1 v_1 \vee \cdots \vee s \lhd_n v_n$, where C' contains no inequality $s \lhd' v$ and $v_j \leq v_i$ is true in I_C, for all i and j with $i < j$. The inequality $u_1 \lhd_1' s$ on the other hand must be produced by some smaller clause D of the form $D' \vee u_1 \lhd_1' s \vee \cdots \vee u_m \lhd_m' s$, where the maximal term s does not occur in D' and $u_i \leq u_j$ is true in I_C, whenever $i < j$. We have to show that there is a rewrite proof of $u_1 \lhd v_1$ in I^C, where $\lhd$ is the highest-priority symbol of $\lhd_1'$ and $\lhd_1$.

Let us first assume that C is a ground instance $v_1 \leq s \vee s < v_1$ or $v_1 < s \vee s \leq v_1$ of a totality axiom. We consider the first case, the second is similar. Take the smaller ground instance $v_1 \leq u_1 \vee u_1 < v_1$ of the totality axiom, which by the induction hypothesis is true in I_C. We know that $u_1 \leq s$ is true in I^D, and hence in I_C. If $v_1 \leq u_1$ were true in I_C, then $v_1 \leq s$ would also have to be true in (the partial ordering) I_C, which contradicts the assumption that C, as a productive clause, is false in I_C. In short, we may infer that $u_1 < v_1$ is true in I_C.

Let us therefore assume that C is not an instance of TA. Since D cannot be an instance of TA either, the clause

$$C'' = C' \vee D' \vee \bigvee_{1 \leq i \leq m, 1 \leq j \leq n} u_i \lhd_{i,j} v_j,$$

where $\lhd_{i,j}$ is the highest-priority symbol of $\lhd_i'$ and $\lhd_j$, is the conclusion of a maximal chaining from ground instances of N. By saturation up to redundancy and part (1), C'' must follow from PO and ground instances of N smaller than C. By the induction hypothesis, all of these clauses, and hence C'', are true in I_C. On the other hand, C is productive, so that C and C' are false in I_C. Similarly, D' is false in I_D and remains false in I_C. (Inequalities produced by a clause $\tilde{D} \succ D$ have a maximal term $\tilde{s} \succeq s$ and therefore cannot contribute to any rewrite proof of an inequality in D'.) We may thus infer that some disjunct $u_i \lhd_{i,j} v_j$ is true in I_C. Moreover, both $u_1 \leq u_i$ and $v_j \leq v_1$ are true in I. Thus, if $\lhd$ is $\leq$ or if $\lhd_{i,j}$ is $<$, we are done. Suppose $\lhd$ is $<$ (so that either $\lhd_1$ or $\lhd_1'$ is $<$) and $\lhd_{i,j}$ is $\leq$ (so that both $\lhd_i'$ and $\lhd_j$ are $\leq$). We distinguish two subcases.

If $u_1 < s$ is true in I_C, then we may use (part (iii) of) the definition of $P_{I_C}(C)$ to infer that $u_i \leq u_1$ is false in I_C. By the induction hypothesis, the ground instance $u_1 < u_i \vee u_i \leq u_1$ of TA is true in I_C, which implies that $u_1 < u_i$, and hence $u_1 < v_1$, is true in I_C.

If $s < v_1$ is true in I_C, then we may use (part (ii) of) the definition of $P_{I_C}(C)$ to infer that $s \leq v_j$ is false in I_C. The ground instance $v_j < s \vee s \leq v_j$ of TA is smaller than C,[7] so that by the induction hypothesis we may infer that $v_j < s$ is true in I_C. Again this implies that $u_1 < v_1$ is true in I_C.

To show irreflexivity of $<$ let us assume that u is the smallest term for which $u < u$ is true in I^C. Since $u < u$ cannot be produced by any clause, there must exist a rewrite proof $u \lhd^L \cdots \lhd^L v \lhd^R \cdots \lhd^R u$ in I^C. We may use transitivity to infer that $v < v$ is also true in I^C, but as $u \succ v$ this contradicts the assumption that u is the smallest term with this property.

The remaining axioms of PO are satisfied for any rewrite closure.

(3) If C is redundant, it follows from PO and smaller ground instances of N. By the induction hypothesis, all these clauses are true in I_C, hence C is true in I_C (and I^C). If C is productive, it is evidently true in I^C, as $E_C \subseteq I^C$. If C is not productive, it is either true in I_C, or else is of the form $C' \vee s < s$, where s is the maximal term in C. Since N is saturated up to redundancy, the irreflexivity resolution inference

$$\frac{C' \ , s < s}{C'}$$

is redundant. In other words, C' follows from PO and clauses smaller than C. We may use the induction hypothesis to infer that C' is true in I_C. □

We now obtain the following refutational completeness result.

Theorem 1 *Let N be a set of clauses with subset $EE_{\mathcal{F}}$, where $\mathcal{F}$ is the given set of function symbols. If N is saturated up to PO-redundancy with respect to MC, then it has a total ordering as model if and only if it does not contain the empty clause.*

Proof. If N contains the empty clause, it has no model. Suppose N does not contain the empty clause. Let I be the interpretation constructed from all ground instances of $N \cup TA$. We may use Lemma 2 to show that I is a model of $N \cup PO \cup TA$, and hence satisfies all clauses specifying a total ordering. □

This theorem applies to arbitrary total orderings. The chaining mechanism can be improved for more specific theories, such as dense total orderings without endpoints.

3.4 Variable elimination

A *variable chaining* is a maximal chaining in which one of the terms s_i or t_j is a variable. Variable chaining can be quite prolific, as the unification of terms required for an inference always succeeds if the terms are variables. Fortunately, the ordering constraints in conditions (ii) and (iii) considerably cut down on the number of variable chainings. More specifically, the ordering constraints can only be satisfied if each term s_i or t_j is either a non-variable or an *unshielded variable*; that is, a variable that does not occur in a subterm $f(\ldots, x, \ldots)$. For example, the variable x is unshielded in $a < x \vee x < b$. Certain unshielded variables can be eliminated in any total ordering. More occurrences of unshielded variables can be eliminated in total orderings without endpoints. If the ordering is also dense *all* unshielded variables can be eliminated.

[7] The fourth component n_L in the tuples associated with literals is crucial at this point.

Variable Elimination:

$$\frac{C \ , \ u_1 \lhd_1 x, \ldots, u_m \lhd_m x \ , \ x \lhd'_1 v_1, \ldots, x \lhd'_n v_n}{C \ , \ \bigvee_{1\le i\le m, 1\le j\le n} u_i \lhd_{i,j} v_j}$$

where x is an unshielded variable not occurring in C, u_i and v_i, and where $\lhd_{i,j}$ is the lowest-priority symbol of $\lhd_i$ and $\lhd'_j$.

This inference rule is sound for total dense orderings without endpoints. It has been used by Bledsoe, Kunen and Shostak (1985) in their chaining calculus. Weaker variable elimination rules can be applied to non-dense orderings. For example, in any total ordering $a < b$ is equivalent to $a < x \vee x < b$. In an ordering without left endpoint a disjunction $C \vee \bigvee_i t_i < x$, where x does not occur in C or any term t_i, is equivalent to C. In a total, "discrete" ordering, in which $s(x)$ denotes the "successor" of x, the disjunction $a \le x \vee x \le b$ is equivalent to $a \le b \vee a \approx s(b)$. This can also be generalized to a variable elimination rule. (The equation $a \approx s(b)$ can be translated into a conjunction of inequations or, alternatively, one may use the equational inference systems described in the next section.)

Lemma 3 *(i) A clause $C \vee x \le x$ is PO-redundant with respect to any set of clauses.*

(ii) A clause $C \vee x < x$ is PO-redundant with respect to any set of clauses that contains C.

(iii) Let C' be a clause $C \vee \bigvee_i u_i \lhd_i x \vee \bigvee_j x \lhd_j v_j$ with an unshielded variable x not occurring in C, u_i, v_i, and let C'' be $C \vee \bigvee_{i,j} u_i \lhd_{i,j} v_j$, where $\lhd_{i,j}$ is the lowest-priority symbol of $\lhd_i$ and $\lhd'_j$. Then C' is either a totality axiom or else is PO-redundant with respect to any set N that contains C'' and the totality axioms.

Proof. (i) A clause $C \vee x \le x$ is satisfied in any partial ordering and hence is *PO*-redundant.

(ii) Observe that $C \vee x < x$ is true in any model of C and that C is smaller than $C \vee x < x$.

(iii) We sketch the main ideas. Take ground instances $D' = D \vee \bigvee_i a_i \lhd_i c \vee \bigvee_j c \lhd_j b_j$ and $D'' = D \vee \bigvee_{i,j} a_i \lhd_{i,j} b_j$ of C' and C'', respectively. Since C'' does not contain the unshielded variable x, but otherwise contains the same terms as C', we have $n_{C'} > n_{C''}$. Consider the maximal literal L'' in D''. Then there is an inequality L' in D' with the same or bigger maximal term occurring in the same argument position, so that the tuple associated with L' is bigger than the tuple associated with L'' in one of the first four components. In short, we have $D' \succ D''$.

Let T_a be the set of all disjunctions $c < a_i \vee a_i \le c$ and $a_i < c \vee c \le a_i$; and let T_b be the set of all disjunctions $c < b_j \vee b_j \le c$ and $b_j < c \vee c \le b_j$. All of these disjunctions are ground instances of the totality axioms. It can be shown that D' is true in every model of $TR \cup A \cup \{D''\}$ and in every model of $TR \cup B \cup \{D''\}$. Also, if none of the terms a_i or c is maximal in D', then clearly all clauses in T_a are smaller than D'. On the other hand, if some term a_i or c is maximal in D', then that term occurs as the first argument of an inequality in D', which we may use to infer that all clauses in T_b are smaller than D'. In sum, D' is TR-redundant with respect to $TA \cup \{D''\}$. □

The lemma not only indicates that variable elimination is simplifying, but that in the presence of the totality axioms the premise in any of the elimination rules is rendered redundant by its conclusion. The variable elimination rules may therefore be called *simplification* rules: their premises can be *replaced* by the respective conclusion. In other words, variable elimination can be made *mandatory*, in that no other inference rule needs to be applied to a clause with an unshielded variable, which also means that variable chainings are not needed. The compatibility of variable elimination with other chaining systems has been shown by Richter (1984) and Hines (1992).

By $CV^{\succ}$ we denote the calculus consisting of irreflexivity resolution and maximal chaining plus variable elimination. Also, let $DO(d, l, r)$ denote the set PO plus all clauses

$$\begin{aligned} x < y &\rightarrow x < d(x,y) \\ x < y &\rightarrow d(x,y) < y \\ &\rightarrow l(x) < x \\ &\rightarrow x < r(x) \end{aligned}$$

which encodes the properties of a dense ordering without endpoints.

Lemma 4 *Let N be a set of clauses that is saturated up to PO-redundancy with respect to MC and contains $EE_{\mathcal{F}}$ as a subset, where $\mathcal{F}$ is the given set of function symbols. If N contains no unshielded variables, then $N \cup DO(d, l, r)$ is also saturated up to PO-redundancy, where d, l and r are function symbols not in $\mathcal{F}$.*[8]

Proof. The only possible inferences between clauses in N and $DO(d, l, r)$ would be variable chainings, of which there are none, as there are no clauses with unshielded variables. Inferences from PO are trivially PO-redundant. The only other inference from $DO(d, l, r)$ which is the chaining of the two axioms for d produces a tautology and is therefore redundant too. Since N is assumed to be saturated, inferences from N are also redundant in N and hence in $N \cup DO(d, l, r)$. In short, $N \cup DO(d, l, r)$ is saturated. □

The significance of this lemma rests on the fact that with variable elimination it is possible to eliminate clauses with unshielded variables. Clauses $C \vee x \leq x$ are redundant, whereas a clause $C \vee x < x$ with unshielded variable x becomes redundant once C has been deduced by irreflexivity resolution. (This is the only situation in which irreflexivity resolution needs to be applied to a clause $C \vee s < t$, where s or t is a variable.) All other unshielded variables can be eliminated by variable elimination. In the next section we briefly discuss the main aspects of the saturation process; for further details see (Bachmair and Ganzinger 1994).

3.5 Saturation

Theorem proving may be viewed as a process of saturating a given set of clauses. A *theorem proving derivation* is a (possibly infinite) sequence $S_0, S_1, \ldots$ of clause sets, such that each set S_{i+1} is obtained from its predecessor S_i either by deleting some redundant clauses, or by adding (one or more) conclusions of inferences, the premises of which are in S_i. The *limit* S_∞ of such a derivation is the set $\bigcup_i \bigcap_{j \geq i} S_j$ of all *persisting* clauses.

A theorem proving derivation is said to be *fair* if the every conclusion of an inference, the premises of which are in S_∞, is contained in some set S_i. A fair derivation can be constructed by systematically applying inferences to persisting formulas.

[8]The function symbols d, l, and r are Skolem symbols for which no equality axioms are needed (Bledsoe, Kunen and Shostak 1985).

Note that fairness imposes no restriction on deletion of redundant formulas. If variable elimination is applied exhaustively, no clause with an unshielded variable persists. In other words, the limit of such a derivation contains no clauses with unshielded variables.

We have the following theorem, cf. (Bachmair and Ganzinger 1994).

Theorem 2 *The limit of a fair theorem proving derivation is saturated up to redundancy.*

In sum, any theorem prover based on maximal chaining is refutationally complete, provided the inference rules are applied in a fair way, as defined above.

4 Chaining with superposition

The chaining calculus described above requires explicit equality axioms. Equality can also be built in via paramodulation (originally introduced by Robinson and Wos 1969), which in fact may be seen as a form of "subterm chaining" (as discussed in Bachmair and Ganzinger 1993). We now allow all three symbols $\approx$, $<$ and $\leq$ to occur in clauses. This is logically redundant as $\leq$ can be expressed as a disjunction in $\approx$ and $<$, but from a practical point of view, and as was confirmed by our experimentation with the Saturate system, it is more appropriate to handle the three relations simultaneously and specifically so as to avoid duplication of terms as much as possible.[9] For the same reason we allow negative equalities $s \not\approx t$ that would otherwise have to be transformed into disjunctions $s < t \vee t < s$. Thus, clauses may contain positive and negative equalities in addition to positive inequalities.

Totality can be expressed as a clause TE

$$x < y \vee y < x \vee x \approx y$$

and the anti-symmetry axiom AS

$$x \leq y \,,\, y \leq x \quad \rightarrow \quad x \approx y$$

specifies the relationship between equality and inequality.

The calculus CS consists of three parts: a chaining calculus for inequalities, a superposition calculus for equalities, and further chaining rules that connect equalities with inequalities. Superposition is a form of paramodulation in which ordering constraints are imposed on the inference rules. We use the superposition calculus of Bachmair and Ganzinger (1994). Symmetry of equality is built in as an inference rule

$$\frac{C \,,\, u \approx v}{C \,,\, v \approx u}$$

where u or v is a maximal term in the premise. (In practice, equations are usually considered as unordered pairs to avoid unnecessary duplication of formulas.)

[9] Testing the solvability of ordering constraints for the lexicographic path ordering is NP-complete in the number of terms that are involved in an inference (Nieuwenhuis 1993) and may easily become a bottleneck in practice.

Reflexivity Resolution:

$$\frac{u \not\approx v \ , C}{C\sigma}$$

where σ is a most general unifier of u and v and $u\sigma \not\approx v\sigma$ is a maximal literal in $C\sigma$.

Superposition:

$$\frac{C \ , t \approx s \quad D \ , L'[s']}{C\sigma \ , D\sigma \ , L[t]\sigma}$$

where (i) σ is a most general unifier of s and s'; (ii) $t\sigma \approx s\sigma$ is strictly maximal in $C\sigma$ and $t\sigma \not\preceq s\sigma$; (iii) L' is either an equality $u[s] \approx v$ or $v \approx u[s]$, or the negation thereof, such that $v\sigma \not\preceq u\sigma$ and $L'\sigma$ is strictly maximal in $D\sigma$, if it is a positive equality, and just maximal in $D\sigma$ if it is negative; and (v) s' is not a variable.

Equality Factoring:

$$\frac{C \ , t \approx s \ , t' \approx s'}{C\sigma \ , t\sigma \not\approx t'\sigma \ , t'\sigma \approx s'\sigma}$$

where (i) σ is a most general unifier of s and s', (ii) $t\sigma \not\preceq s\sigma$, and (iii) $s\sigma \approx t\sigma$ is maximal in $(C \vee s \approx t \vee s' \approx t')\sigma$.

Superposition is a restricted form of subterm chaining applied to equations. We need a similar form of "maximal subterm chaining" with equations into inequations. These inference rules are designed to reduce maximal terms.

Equality Chaining Left:

$$\frac{C \ , u \approx s \quad D \ , t_1[s_1]_p \lhd_1 v_1, \ldots, t_n[s_n]_p \lhd_n v_n}{C\sigma \ , D\sigma \ , t_1[u]_p \lhd_1 v_1, \ldots, t_n[u]_p \lhd_n v_n}$$

where (i) $\sigma = \rho\tau$ with ρ the most general unifier of $s, s_1, \ldots, s_n$, and τ the most general unifier of $t_1[s_1]\rho, \ldots, t_n[s_n]\rho$, (ii) $u\sigma \not\preceq s\sigma$, and $u\sigma \approx s\sigma$ is strictly maximal in $C\sigma$, (iii) $v_i\sigma \not\preceq t_1\sigma$, for all $1 \leq i \leq n$, and $v\sigma \not\prec t_1\sigma$, for all terms v in D, and $t_1\sigma$ occurs in $D\sigma$ only in equalities or inequalities $v \trianglelefteq t_1\sigma$, and (iv) none of the s_i is a variable.

Equality Chaining Right:

$$\frac{C \ , u \approx s \quad D \ , v_1 \lhd_1 t_1[s_1]_p, \ldots, v_n \lhd_n t_n[s_n]_p}{C\sigma \ , D\sigma \ , v_1 \lhd_1 t_1[u]_p, \ldots, v_n \lhd_n t_n[u]_p}$$

where (i) $\sigma = \rho\tau$ with ρ is the most general unifier of $s, s_1, \ldots, s_n$, τ the most general unifier of $t_1[s_1]\rho, \ldots, t_n[s_n]\rho$, (ii) $u\sigma \not\preceq s\sigma$, and $u\sigma \approx s\sigma$ is strictly maximal in $C\sigma$, (iii) $v_i\sigma \not\preceq t_1\sigma$, for all $1 \leq i \leq n$, and $v\sigma \not\prec t_1\sigma$, for all terms v in D, and $t_1\sigma$ occurs in $D\sigma$ only in equalities, and (iv) none of the s_i is a variable.

Finally, we also need to be able to deduce equalities implicitly specified by inequalities, so that the anti-symmetry axiom is satisfied. For that purpose we modify the maximal chaining rule as follows.

Inequality Chaining:

$$\frac{C \ , u_1 \lhd_1 s_1, \ldots, u_m \lhd_m s_m \quad D \ , t_1 \lhd'_1 v_1, \ldots, t_n \lhd'_n v_n}{C\sigma \ , D\sigma \ , \bigvee_{1 \le i \le m, 1 \le j \le n} ineq(i,j)}$$

where (i) σ is the most general unifier of $s_1, \ldots, s_m, t_1, \ldots, t_n$, (ii) $ineq(i,j)$ is the strict inequality $u_i\sigma < v_j\sigma$ if $\lhd_i$ or $\lhd'_j$ is $<$, and the disjunction $u_i\sigma < v_j\sigma \vee u_i\sigma \approx t_1\sigma$ otherwise,[10] (iii) $u_i\sigma \not\succeq s_1\sigma$, for all $1 \le i \le m$, and $u\sigma \not\succeq s_1\sigma$, for all terms u in C, and (iv) $v_i\sigma \not\succeq t_1\sigma$, for all $1 \le i \le n$, and $v\sigma \not\succ t_1\sigma$, for all terms v in D, and $t_1\sigma$ occurs in $D\sigma$ only in inequalities $v \lhd t_1\sigma$.

For example, from $a \le f(x)$ and $f(a) \le b \vee f(y) < c$ we may deduce $a < b \vee f(a) \approx a \vee a < c$ (assuming $f(a)$ is the maximal term in the clause). Inequality chaining is like maximal chaining, except that when two non-strict inequalities are chained we split the resulting non-strict inequality into a strict inequality and an equality.

The calculus CS consists of the above inference rules plus irreflexivity resolution. (Maximal chaining is replaced by inequality chaining.) It can easily be shown that all inference rules in CS are simplifying. To prove the refutational completeness proof of CS we use the same approach as for MC, but have to modify the definition of the interpretations I_C and I. Let I be an interpretation. A ground term t is said to be *reducible* with respect to I if we have $t \approx_I^L s$ (or equivalently $s \approx_I^R t$), for some term s. Terms that are not reducible are said to be *irreducible*.

Let C be a ground clause with maximal term s and I be an interpretation. Let $P_I(C)$ be an inequality $s < t$ if C contains $s \lhd t$ and $t < t'$ is false in I, for any inequality $s \lhd t'$ in C; and let $P_I(C)$ be $t < s$ if C contains $t \lhd s$, but no inequality $s \lhd u$, and $t' < t$ is false in I, for any inequality $t' \lhd s$ in C.

Let N be a set of ground clauses and C be a clause in N with maximal term s. As before, let R_C be the set $\bigcup_{C \succ D} E_D$ and I_C the rewrite closure of R_C. The set E_C is defined as follows:

- If s occurs in a negative literal in C or else is reducible with respect to I_C, then $E_C = \emptyset$. (For the remaining cases, we assume that s is irreducible and occurs only in positive literals.)

- If C is false in I_C and contains some inequality $s \lhd t$ or $t \lhd s$, but not $s < s$, then $E_C = \{P_{I_C}(C)\}$.

- If C is false in I_C and can be written as $C' \vee s \approx t$, where $s \approx t$ is a strictly maximal literal and C' is false in the rewrite closure of $R_C \cup \{s \approx t\}$, then $E_C = \{s \approx t\}$.

- Otherwise, $E_C = \emptyset$.

Again, we call C *productive* if $E_C \ne \emptyset$. By I^C we denote the rewrite closure of $R_C \cup E_C$. Finally, by R we denote the set $\bigcup_C E_C$ and by I the rewrite closure of R.

The key properties of the interpretation I are summarized in the following lemma.

Lemma 5 *Let N be a set of clauses that is saturated up to PO-redundancy under CS and contains EE as a subset, but does not contain the empty clause. Let I be the interpretation constructed from all ground instances of $N \cup TE$. Then for every ground instance C of a clause in N we have:*

[10]Multiple copies of an equation $u_i\sigma \approx t_1\sigma$ resulting from different inequalities $t_j < v_j$ and $t_k < v_k$ can of course be merged into one.

(1) if C is productive it is non-redundant;

(2) I^C satisfies PO;

(3) C is true in I^C.

Furthermore, I is a model of the anti-symmetry axiom.

Proof. We use induction on $\succ$. Let C be a ground instance of a clause in N, such that the assertion is satisfied for all smaller ground instances of N.

(1) This part is the same as for Lemma 2.

(2) This part is similar to Lemma 2, except that we have to use inequality chaining instead of maximal chaining. We outline the differences between the two cases.

Transitivity, as we have seen, reduces to a commutation property. Let C be a clause $C' \vee s \lhd_1 v_1 \vee \cdots \vee s \lhd_n v_n$, where C' contains no inequality $s \lhd' v$ and $v_j \leq v_i$ is true in I_C, for all i and j with $i < j$; and let D be a clause $D' \vee u_1 \lhd_1' s \vee \cdots \vee u_m \lhd_m' s$, where D' contains no inequality $s \lhd u$ or $u \lhd s$ and $u_i \leq u_j$ is true in I_C, whenever $i < j$. We assume that C produces $s < v_1$ and D produces $u_i < s$, and have to show that there is a rewrite proof of $u_1 \lhd v_1$ in I^C, where $\lhd$ is $<$ if $\lhd_1'$ or $\lhd_1$ is $<$, and $\leq$ otherwise. Note that the term s must be irreducible with respect to I_C (for otherwise C cannot be productive).

We know that D is not an instance of TE. Let us first assume that C is not an instance of TE either. Then the clause

$$C'' = C' \vee D' \vee \bigvee_{1 \leq i \leq m, 1 \leq j \leq n} u_i < v_j \vee \bigvee_{k \in K} u_k \approx s$$

is the conclusion of an inequality chaining inference from ground instances of N. (Here $K = \emptyset$, if all symbols $\lhd_i'$ are $<$, and K is the set of all indices k for which $\lhd_k$ is $\leq$, otherwise.) The equations $u_k \approx s$ are all false in I_C, as s is irreducible and $s \succ u_k$, for all k. Since C' and D' are also false in I_C, some strict inequation $u_i < v_j$ is true in I_C, which implies that $u_1 < v_1$ is true in I_C.

Suppose C is a ground instance $v_1 < s \vee s < v_1 \vee s \approx v_1$ of a totality axiom in TE. By the induction hypothesis, the smaller ground instance $v_1 < u_1 \vee u_1 < v_1 \vee u_1 \approx v_1$ is true in I_C. Since $u_1 < s$ is true, and $v_1 < s$ false in I_C, we may infer that both $v_1 < u_1$ and $u_1 \approx v_1$ are false in I_C. Consequently, $u_1 < v_1$ must be true in I_C.

The remaining axioms in PO can be shown to be true in I_C in the same way as in the proof of Lemma 2.

(3) If C is productive, it is evidently true in I^C. Suppose C is non-productive. If C is false in I_C, then it must violate some other condition imposed on productive clauses. We distinguish according subcases.

(3.1) Suppose C is an instance $\hat{C}\sigma$ of a clause $\hat{C}$ in N, such that $x\sigma$ is reducible with respect to I_C, for some variable x in $\hat{C}$. Let t be a ground term, such that $x\sigma \approx_{I_C}^L t$, and let τ be the same substitution as σ, except that $x\tau = t$. Since $\hat{C}\sigma \succ \hat{C}\tau$, we may use the induction hypothesis to infer that $C' = \hat{C}\tau$ is true in $I_{C'}$, and hence in I_C. But since $x\sigma \approx t$ is also true in I_C, and C' differs from C only in that some occurrences of $x\sigma$ are replaced by t, the clause C is also true in I_C.

For the remaining subcases we assume that C is a *reduced* ground instance $\hat{C}\sigma$ of a clause in N in the sense that $x\sigma$ is irreducible for all variables x in $\hat{C}$.[11]

(3.2) If C is of the form $C' \vee s < s$, the same arguments as in Lemma 2 apply.

[11] This assumption is needed for lifting subterm chaining inferences.

(3.3) If C is of the form $C' \vee s \not\approx s$, we may adapt case (3.2) by using reflexivity resolution instead of irreflexivity resolution.

(3.4) If s is reducible with respect to I_C, we may apply superposition or equality chaining. We discuss one representative case.

Suppose C is of the form $C' \vee s \not\approx t$, where $s \not\approx t$ is a maximal literal in C, $s \succ t$, and s contains a subterm u, for some equation $u \approx v$, with $u \succ v$, produced by a clause $D = D' \vee u \approx v$ (smaller than C). By superposition we obtain $C'' = C' \vee D' \vee s[v] \not\approx t$ from C and D. Since C and D are reduced ground instances of clauses $\hat{C}$ and $\hat{D}$, respectively, the clause C'' is a ground instance of the conclusion of a superposition inference from $\hat{C}$ and $\hat{D}$. By saturation up to PO-redundancy, C'' must logically follow from PO and ground instances of N smaller than C. Since all of these clauses are true in I_C, we may infer that C'' is true in I_C, which implies that C is true in I_C.

(3.5) The only remaining case not yet covered is of a clause $C = C'' \vee s \approx t' \vee s \approx t$, where s is irreducible by R_C and $t \approx t'$ is true in I_C. By equality factoring we obtain the clause $C'' \vee t \not\approx t' \vee s \approx t'$, which by saturation up to PO-redundancy is true in I_C. Consequently, C is also true in I_C, which completes the proof of part (3).

Finally, it is straightforward to show that all ground instances $u \leq v \vee v \leq u \rightarrow u \approx v$ are true in I. Suppose $u \leq v$ and $v \leq u$ are true in I, hence provable by rewrite proofs. Since R contains only equations or strict inequalities, each rewrite proof is either a proof of an equation or of a strict inequality. Since $u < v$ is false (for otherwise irreflexivity is violated), we may conclude that there is a proof of $u \approx v$. □

Using this lemma, we obtain the following completeness result.

Theorem 3 *Let N be a set of clauses that is saturated up to PO-redundancy (with respect to CS). Then N has a total ordering with equality as a model if and only if it does not contain the empty clause.*

The variable elimination rule applies as previously, but the presence of equality complicates matters somewhat. The elimination of literals $x < x$, $x \leq x$, $x \approx x$, and $x \not\approx x$ is obvious. To eliminate an unshielded variable x from a clause D we translate all (positive and negative) equalities with x into inequalities, and then apply the usual variable elimination rule. More precisely, a clause

$$D = C \vee \bigvee_{1 \leq i \leq m} x \not\approx u_i \vee \bigvee_{1 \leq j \leq n} x \approx v_j$$

where x is an unshielded variable, is transformed to the formula

$$C \vee \bigvee_{1 \leq i \leq m} (x < u_i \vee u_i < x) \vee \bigvee_{1 \leq j \leq n} (x \leq v_j \wedge v_j \leq x)$$

in which x does not occur in an equality. The formula can be translated into 2^n clauses, to which the usual variable elimination rule can be applied. Let us denote by VE_D the set of all these clauses, with x eliminated. Unfortunately, some ground instances of clauses in VE_D may be larger than the corresponding ground instance of $D\sigma$ with respect to our clause ordering. Consequently, clauses are not necessarily rendered PO-redundant by variable elimination. However, those ground instances of D, in which x is instantiated to the maximal term in the clause, are PO-redundant.

Lemma 6 *Let D be a clause with an unshielded variable x of the above form and σ be a substitution such that $x\sigma$ is a maximal term in $D\sigma$. Then $D\sigma$ is redundant with respect to any set N that contains the clauses in VE_D and the totality axioms.*

To sum up these considerations, a clause with an unshielded variable x is not necessarily redundant, but inferences involving this variable (chaining through x, superposition from x, [ir]reflexivity resolution applied to [in]equalities with x) are redundant. (The ordering constraints for such inferences require x to be instantiated with the maximal term, in which case the corresponding instances of the C are redundant, as indicated by the lemma.) The lemma thus indicates that variable chaining is not needed.

5 Summary

We have presented chaining-based inference systems for total orderings that extend previous work in several respects: we impose ordering constraints on chaining inferences; we build equality directly into the inference mechanism; and we establish refutational completeness in the presence of a notion of redundancy that covers such important simplification techniques as tautology deletion, subsumption, condensement, and demodulation. The completeness proofs of our results are conceptually simple. We deal with variable elimination as a simplification rule, separately from the chaining rules; an approach that better clarifies the connection between these essential components of chaining systems.

The improvements are not only of theoretical significance, as experimental evidence indicates that equational inference mechanisms, such as superposition and demodulation, are preferable to explicit axiomatizations of equality. Some form of equational reasoning appears to be used in the provers described by Bledsoe and Hines (1980), Hines (1988) and Hines (1990), and inference systems with equational reasoning capabilities may provide a better approximation to actual implementation practice than other chaining systems.

The superposition calculus, as described above, contains an explicit factoring rule for equalities, whereas maximal chaining implicitly encodes factoring for inequalities. An alternative to maximal chaining would be ordered chaining (as in Bachmair and Ganzinger 1993) with inequality factoring.

In algebraic structures such as ordered rings one may have axioms

$$\begin{aligned} x < y &\rightarrow x + z < y + z \\ x < y &\rightarrow -y < -x \end{aligned}$$

that express monotonicity or anti-monotonicity properties of an ordering with respect to certain functions. The techniques we have discussed should be extended to such cases. An important question in this context is to what extent chaining through or below variables is necessary. Our techniques may also be useful in the context of chaining and variable elimination for set theory, an application that has been studied by Hines (1990).

References

L. Bachmair and H. Ganzinger, 1990. On Restrictions of Ordered Paramodulation with Simplification. In M. Stickel, editor, *Proc. 10th Int. Conf. on Automated Deduction, Kaiserslautern*, Lecture Notes in Computer Science, vol. 449, pp. 427–441, Berlin, Springer-Verlag.

L. Bachmair and H. Ganzinger, 1993. Rewrite Techniques for Transitive Relations. Technical Report MPI-I-93-249, Max-Planck-Institut für Informatik, Saarbrücken.

L. Bachmair and H. Ganzinger, 1994. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, Vol. 4, No. 3, pp. 1–31. Revised Version of MPI-I-91-208, to appear.

W. Bledsoe, K. Kunen and R. Shostak, 1985. Completeness results for inequality provers. *Artificial Intelligence*, Vol. 27, pp. 255–288.

W. W. Bledsoe and L. M. Hines, 1980. Variable Elimination and Chaining in a Resolution-based Prover for Inequalities. In Wolfgang Bibel, Robert Kowalski, editors, *5th Conference on Automated Deduction*, LNCS 87, pp. 70–87, Les Arcs, France, Springer-Verlag.

D. Brand, 1975. Proving theorems with the modification method. *SIAM Journal of Computing*, Vol. 4, pp. 412–430.

N. Dershowitz, 1987. Termination of Rewriting. *Journal of Symbolic Computation*, Vol. 3, pp. 69–116.

L. M. Hines, 1988. Hyper-chaining and knowledge-based theorem proving. In *Proc. 9th Int. Conf. on Automated Deduction*, Lecture Notes in Computer Science, pp. 469–486, Berlin, Springer-Verlag.

L. Hines, 1990. Str$\dot{+}$ve$\subseteq$: The Str$\dot{+}$ve-based Subset Prover. In Mark E. Stickel, editor, *10th International Conference on Automated Deduction*, LNAI 449, pp. 193–206, Kaiserslautern, FRG, Springer-Verlag.

L. M. Hines, 1992. Completeness of a Prover for Dense Linear Logics. *Journal of Automated Reasoning*, Vol. 8, pp. 45–75.

J. Hsiang and M. Rusinowitch, 1991. Proving refutational completeness of theorem proving strategies: The transfinite semantic Tree method. *Journal of the ACM*, Vol. 38, No. 3, pp. 559–587.

J. Levy and J. Agustí, 1993. Bi-rewriting, a Term Rewriting Technique for Monotonic Order Relations. In C. Kirchner, editor, *Rewriting Techniques and Applications*, Lecture Notes in Computer Science, vol. 690, pp. 17–31, Berlin, Springer.

Robert Nieuwenhuis, 1993. Simple LPO constraint solving methods. *Information Processing Letters*, Vol. 47, pp. 65–69.

P. Nivela and R. Nieuwenhuis, 1993. Saturation of first-order (constrained) clauses with the Saturate system. In *Proc. 5th Int. Conf. on Rewriting Techniques and Applications*, Lecture Notes in Computer Science, vol. 690, pp. 436–440, Berlin, Springer-Verlag.

Michael M. Richter, 1984. Some reordering properties for inequality proof trees. Lecture Notes in Computer Science, vol. 171, pp. 183–197, Berlin, Springer-Verlag.

G.A. Robinson and L. T. Wos, 1969. Paramodulation and theorem proving in first order theories with equality. In B. Meltzer, D. Michie, editors, *Machine Intelligence 4*, pp. 133–150. American Elsevier, New York.

J. R. Slagle, 1972. Automatic theorem proving for theories with Built-in Theories Including Equality, Partial Orderings, and Sets. *Journal of the ACM*, Vol. 19, pp. 120–135.